# Introduction aux CRF via l'annotation par des modèles graphiques

Isabelle Tellier

LIFO, Université d'Orléans

# Plan

1. Annoter pour quoi faire
2. Apprendre avec un modèle graphique
3. Annnoter des chaînes avec un HMM
4. Les CRF et leur application aux chaînes
5. CRF sur les arbres
6. Conclusion

# 1. Annoter pour quoi faire

Qu'est-ce qu'annoter ?

- les données de départ peuvent être des textes ou des arbres ou...
  - texte = séquence d'items
  - arbre = structure hiérarchique d'items

  pris dans un vocabulaire fini
- annotation : l'association des données avec d'autres items pris dans un autre vocabulaire fini
- ici : les données et les annotation auront la même structure mais ce n'est pas obligatoire

# 1. Annoter pour quoi faire

Exemples d'annotations sur des textes

- étiquetage POS ("part of speech") : item = "mot", annotation = catégorie syntaxique (Det, Nom, etc.) dans le texte
- reconnaissance des entités nommées, EI : item = "mot", annotation = position de l'EN (B : "Begin", I : "In", O : "Out")

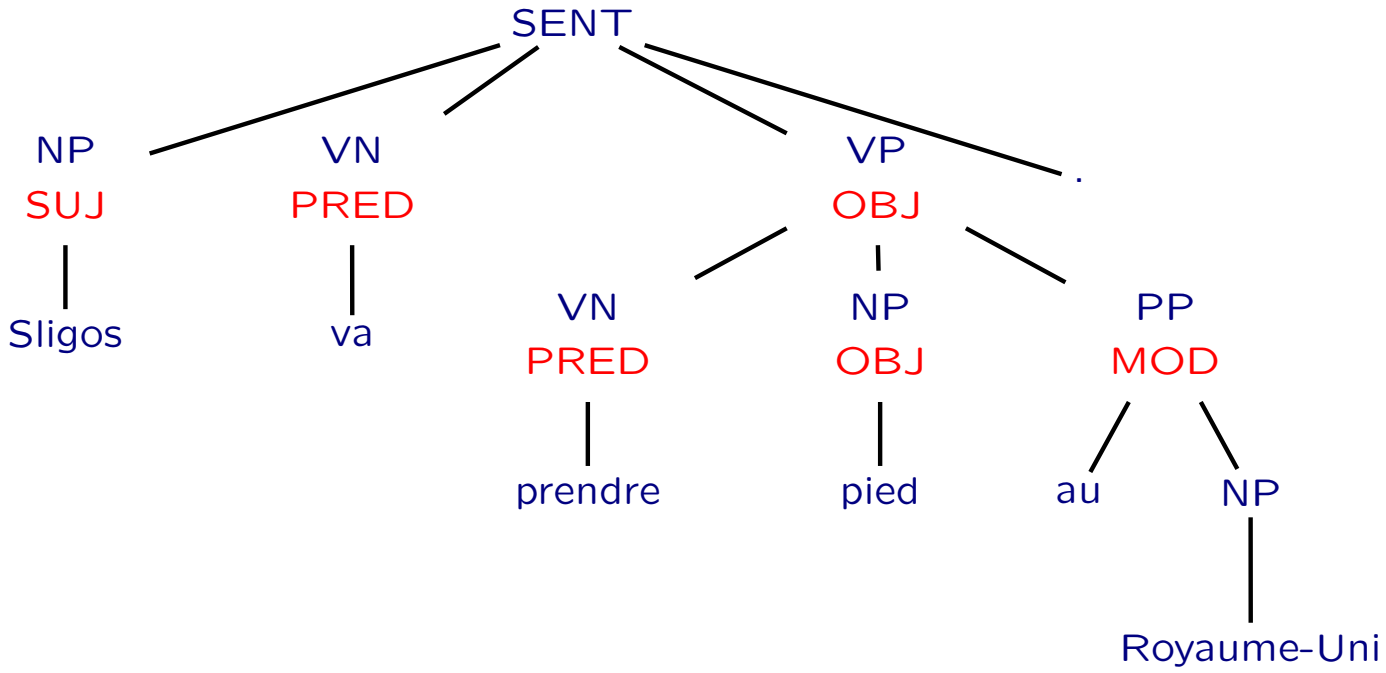
| En | 2008 | les | Jeux | Olympiques | ont | eu | lieu | à | Pékin |
|---|---|---|---|---|---|---|---|---|---|
| O | B | O | B | I | O | O | O | O | B |

- segmentation d'un texte en "chunks", en "syntagmes"...
- alignement de phrases : item = "mot", annotation = le(s) mot(s) correspondant(s) dans une autre phrase (par exemple pour la traduction automatique)
- annotation de phrases : item = "phrase", annotation = "classe"...

# 1. Annoter pour quoi faire

Exemples d'annotations sur des arbres

– étiquetage fonctionnel d'arbres syntaxiques

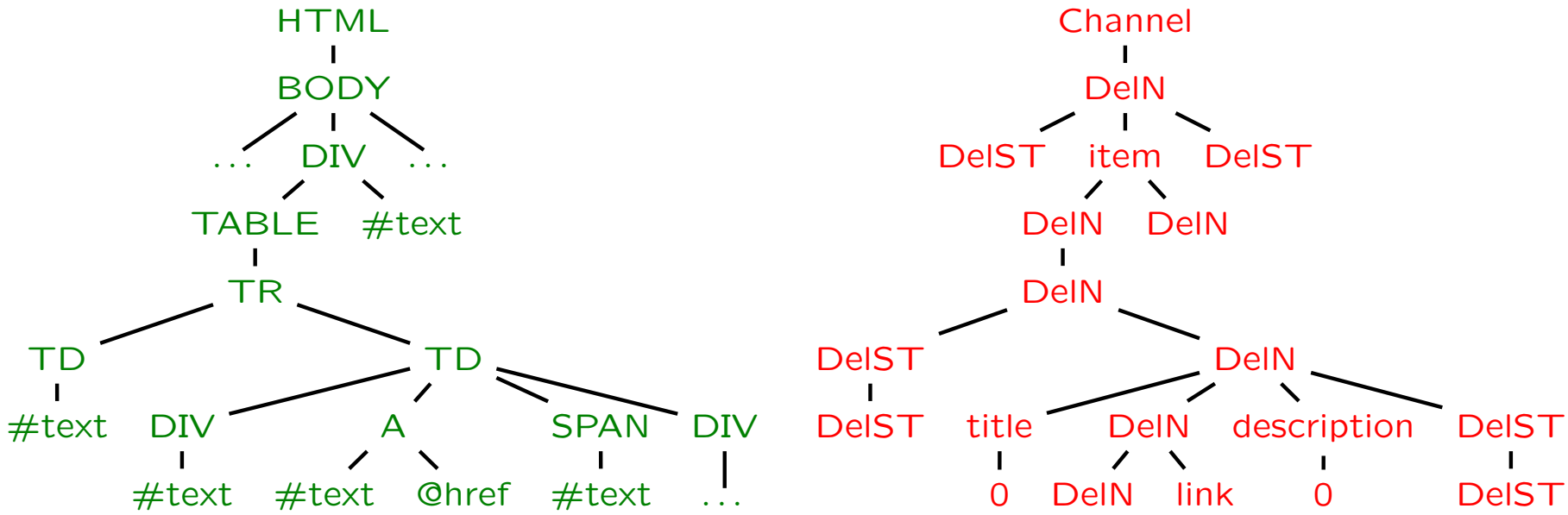



– étiquetage en rôles thématiques/sémantiques d'arbres syntaxiques : idem mais avec annotation agent, patient, etc.

– extraction d'information sur le Web ou les documents XML

Exemples d'annotations sur des arbres (suite)

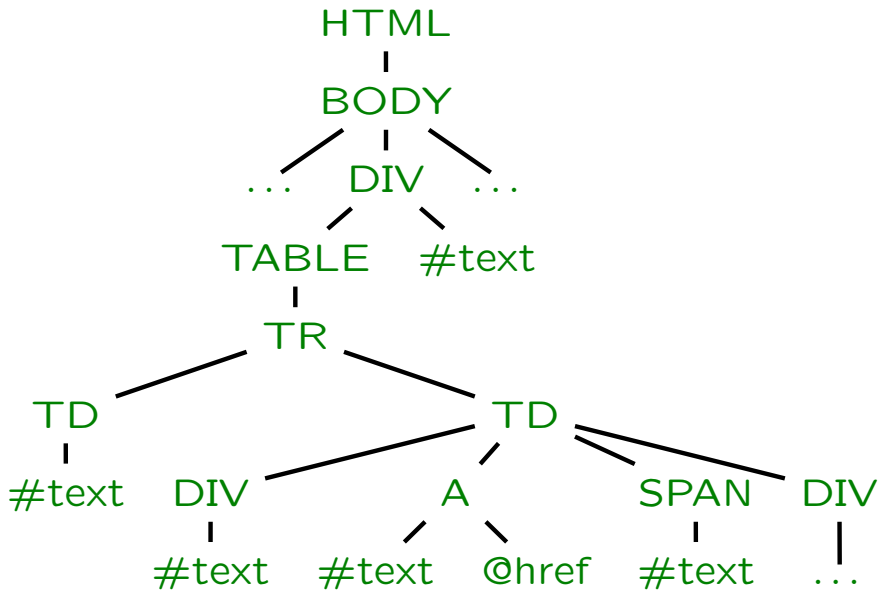
- transformation d'un arbre en un autre

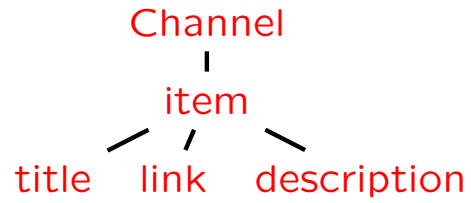



- à gauche : un arbre HTML
- à droite : une annotation avec des opérations d'édition
  - DelN, DelST : suppression de nœud/sous-arbre
  - channel, item, title, link, description : renommage de nœuds

Exemples d'annotations sur des arbres (suite)

- exécution des opérations d'édition




- application implémentée : génération de flux RSS à partir de pages HTML
- passage d'une DTD à une autre

# 1. Annoter pour quoi faire

Synthèse

- de nombreuses tâches peuvent se formuler comme des tâches d'annotation
- chaque tâche requiert de spécifier :
  - la nature des items
  - les relations entre items : séquence, ordres dans un arbre...
  - la nature des annotations et leur interprétation
  - les relations entre annotations
  - les relations entre les items et leur annotation
- pré-traitements et post-traitements souvent nécessaires

# Plan

1. Annoter pour quoi faire
2. Apprendre avec un modèle graphique
3. Annnoter des chaînes avec les HMM
4. Les CRF et leur application aux chaînes
5. CRF sur les arbres
6. Conclusion

# 2. Apprendre avec un modèle graphique

Apprendre à annoter : pourquoi ?

- ne requiert pas de ressources externes (dictionnaires, listes)
- requiert (en principe) moins de travail
- requiert (en principe) moins de compétences en programmation
- le même programme s'adapte aux données, à la langue...
- en étant plus robuste aux données bruitées

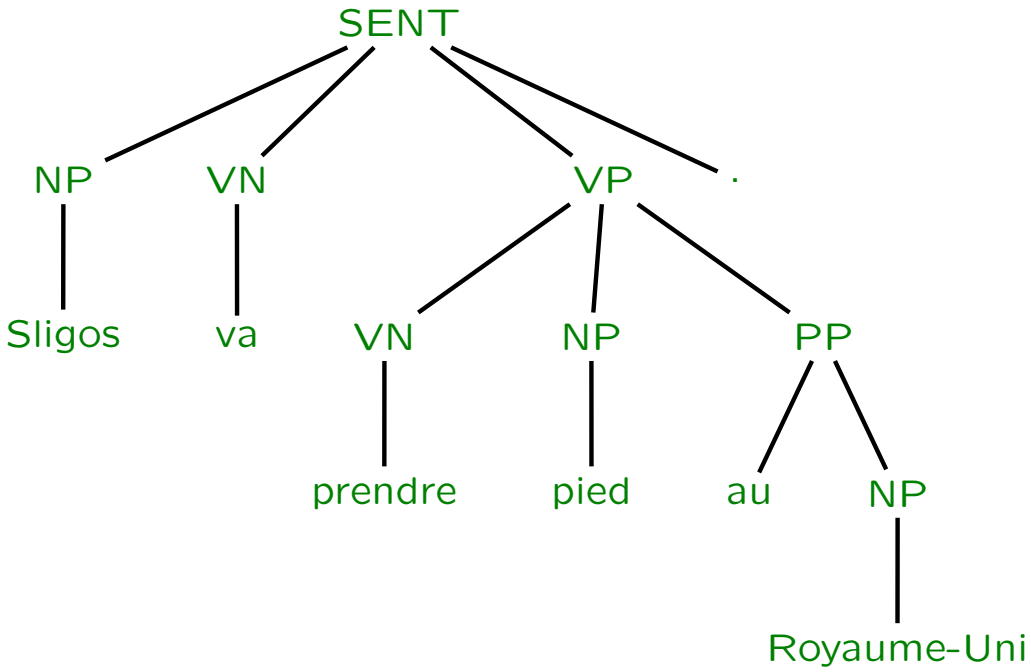
A condition...

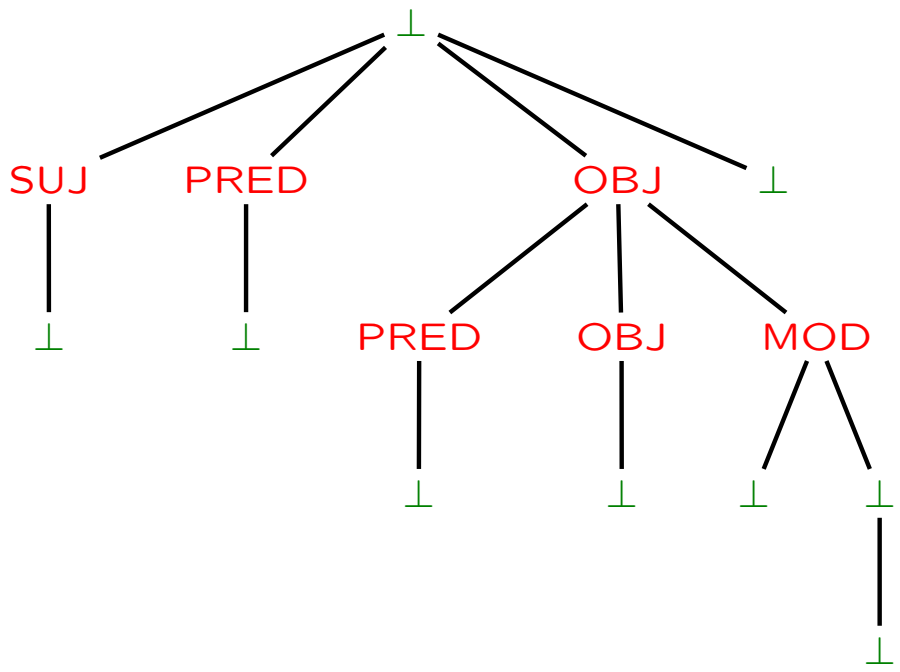
- de disposer d'exemples annotés du domaine

# 2. Apprendre avec un modèle graphique

## Notations de base

- notations classiques : $x$ est une donnée, $y$ est une annotation
- on supposera ici que $x$ et $y$ ont la même structure
- ex. sur les séquences : $x = un\ chat\ dort$ et $y =$Det Nom Vintr
- ex. sur les arbres : $x$ à gauche, $y$ à droite



# 2. Apprendre avec un modèle graphique

Apprendre à annoter : protocole classique

- on dispose d'un ensemble d'exemples annotés $S = \{(x, y)\}$
- on sépare $S$ en deux sous-ensembles : $S = A \cup T$ avec
  - $A$ : ensemble d'apprentissage
  - $T$ : ensemble de test
- avec $A$, on apprend un étiqueteur $f$, qui prédit $f(x) = \widehat{y}$
- $\forall (x, y) \in T$, on compare $f(x) = \widehat{y}$ avec $y$ avec des mesures
  - précision de l'étiquette E : $p_E = \frac{|y{=}E \ \ ET \ \ \widehat{y}{=}E|}{|\widehat{y}{=}E|}$
  - rappel de l'étiquette E : $r_E = \frac{|y{=}E \ \ ET \ \ \widehat{y}{=}E|}{|y{=}E|}$
  - F1-mesure : $\frac{2pr}{p+r}$
- si $S$ est trop limité ou s'il y a des risques de biais dans le découpage $S = A \cup T$ : validation croisée

# 2. Apprendre avec un modèle graphique

Apprentissage automatique statistique

- on suppose qu'il existe une distribution de probabilité $p(y|x)$
- la forme du modèle $p$ est fixée, à des paramètres près
- les deux problèmes qui se posent :
  - apprentissage : fixer les paramètres du modèle $p$ à l'aide des couples $(x, y)$
  - annotation : une fois $p$ fixé, pour tout nouveau $x$, trouver l'annotation $y$ la plus probable, c'est-à-dire $\hat{y} = argmax_y \ p(y|x)$

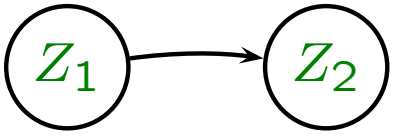
# 2. Apprendre avec un modèle graphique

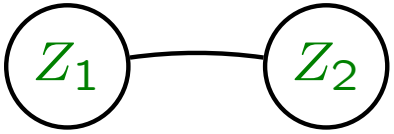
Apprentissage automatique statistique

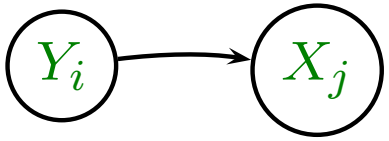
- une variable aléatoire est une variable pouvant prendre plusieurs valeurs données (cf. le dé...)
- on décompose $x$ et $y$ en des ensembles de variables aléatoires : $X = \{X_1, X_2, ..., X_n\}$ et $Y = \{Y_1, Y_2, ...Y_n\}$
- ex : les $x$ sont des séquences de mots, les $y$ leur étiquetage POS :
  - $X_1$ : variable dont les valeurs sont les 1ers mots des séquences $x$
  - $Y_1$ : variable dont les valeurs sont les 1ères étiquettes POS des séquences $y$, etc.
- intuition : il y a des dépendances entre les variables :
  - ex : si $X_i = le$, alors $Y_i$ =Det ou $Y_i$ =Pro
  - si en plus $Y_{i+1}$ =Nom alors $Y_i$ =Det...

# 2. Apprendre avec un modèle graphique

## Modèles graphiques

- un modèle graphique définit les dépendances entre variables aléatoires par un graphe
- les variables aléatoires sont les nœuds du graphe
- 

: la valeur de $Z_2$ dépend de celle de $Z_1$
- 

: dépendances mutuelles entre $Z_1$ et $Z_2$
- dans un modèle graphique génératif, il y a des dépendances dirigées 

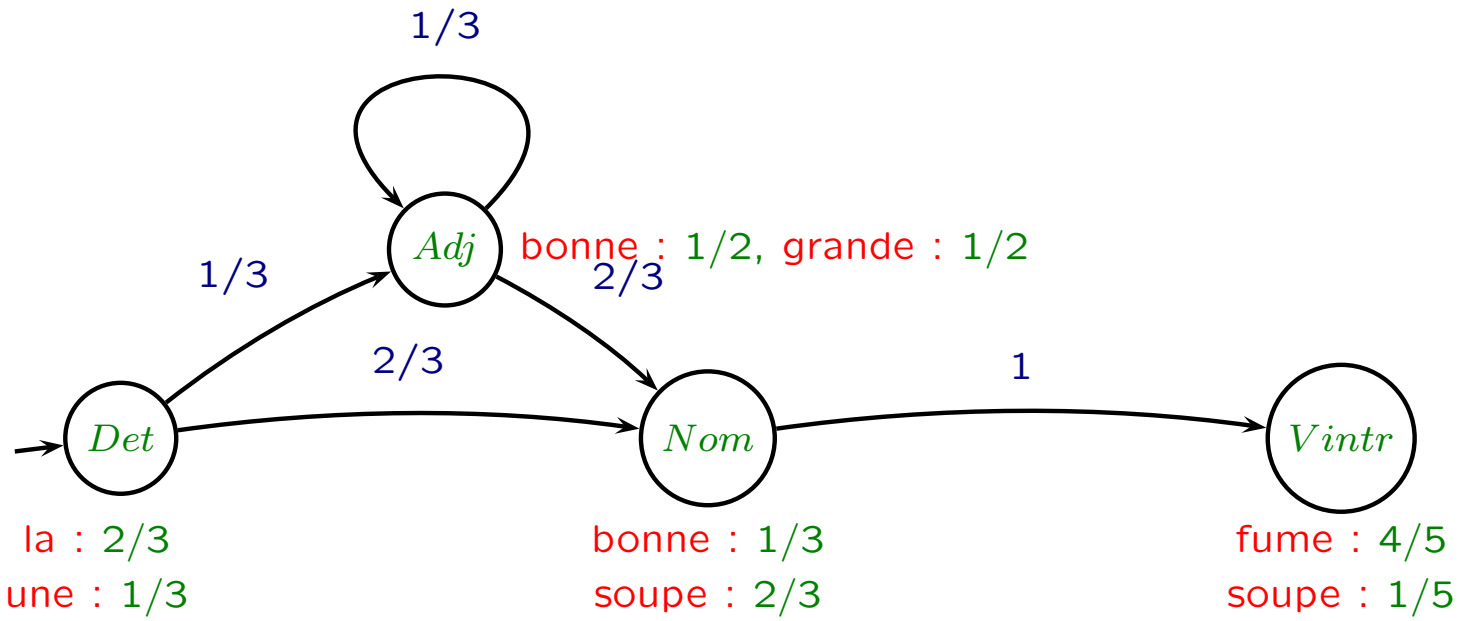
- exemples de modèle graphique génératif : les HMM, les PCFG
- dans ce cas, on utilise : $p(y|x) = \frac{p(x,y)}{p(x)}$

# Plan

1. Annoter pour quoi faire
2. Apprendre avec un modèle graphique
3. Annnoter des chaînes avec les HMM
4. Les CRF et leur application aux chaînes
5. CRF sur les arbres
6. Conclusion

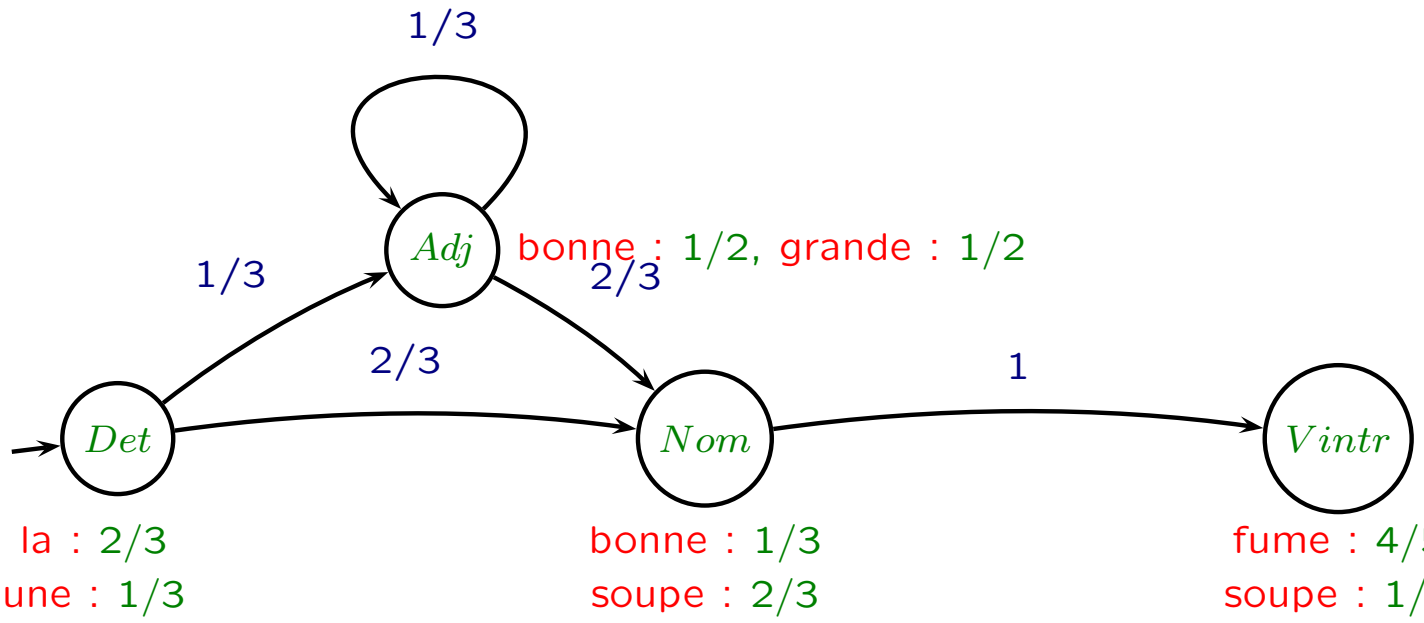
# 3. Annnoter des chaînes avec les HMM

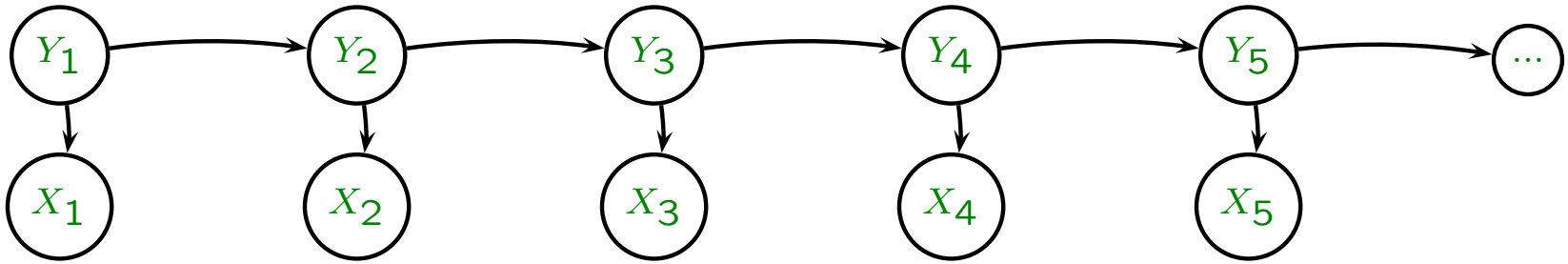
## Un HMM



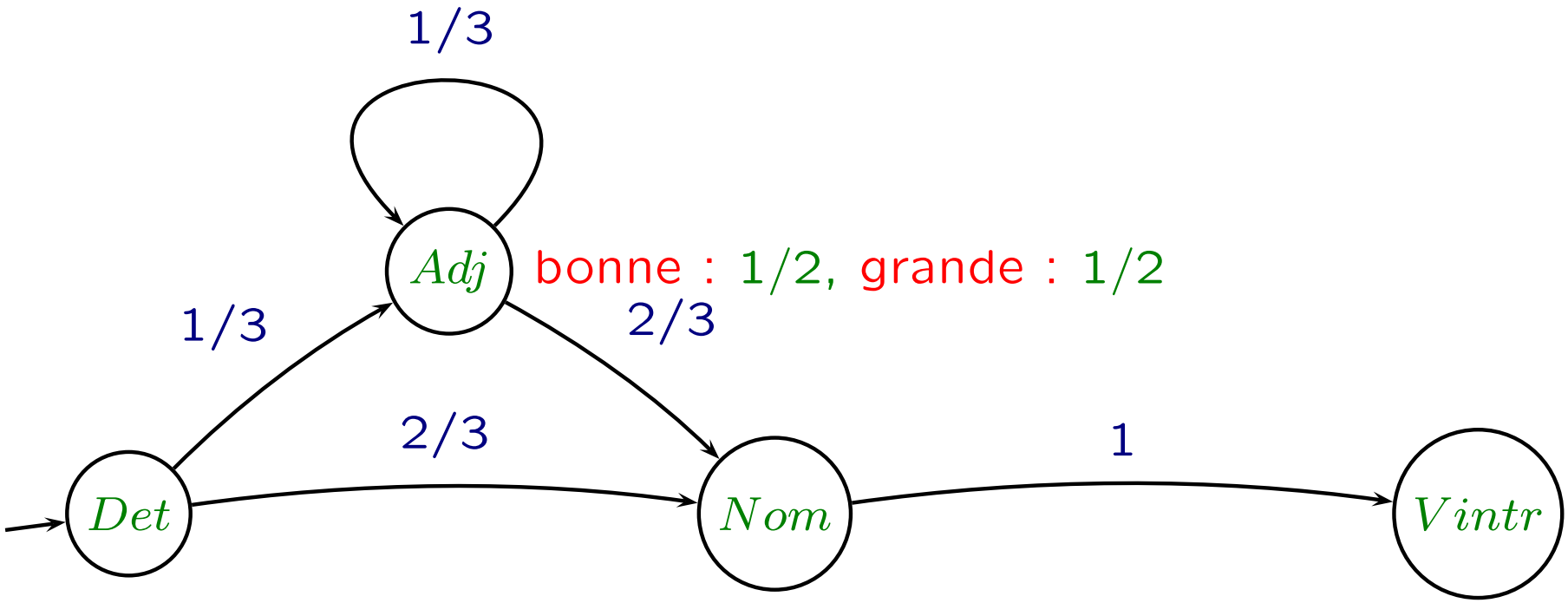
# 3. Annnoter des chaînes avec les HMM

## Un HMM et le graphe correspondant



# 3. Annnoter des chaînes avec les HMM




$p(x = \text{la bonne soupe}, y = Det\ Adj\ Nom)$

$= p(Det : \text{la})p(Det \to Adj)p(Adj : \text{bonne})p(Adj \to Nom)p(Nom : \text{soupe})$

$p(x = \text{la bonne soupe}, y = Det\ Adj\ Nom) = 2/3{*}1/3{*}1/2{*}2/3{*}2/3 = 4/81$

$p(x = \text{la bonne soupe}, y = Det\ Nom\ Vintr) = 2/3{*}2/3{*}1/3{*}1{*}1/5 = 4/135$

$p(x = \text{la bonne soupe}) = \sum_y p(x, y) = 4/81 + 4/135$

# 3. Annnoter des chaînes avec les HMM

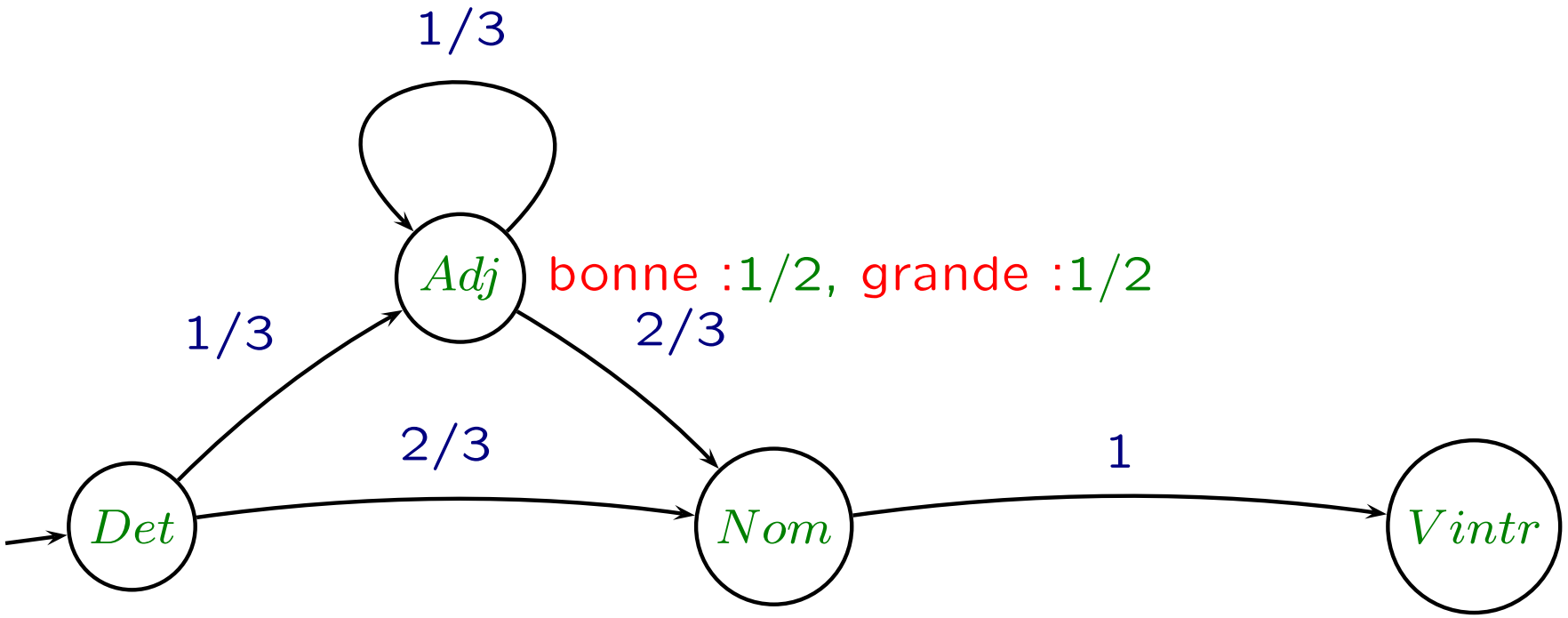
## Hypothèses préliminaires

- la structure du HMM est supposée connue (ou hypothèse que tous les états sont connectés)
- des exemples annotés $(x, y)$ sont disponibles

## Les deux problèmes classiques

- apprentissage : trouver les paramètres, c'est-à-dire les probabilités d'émissions/de transitions à partir des $(x, y)$
- annotation : utiliser le modèle $p$ appris sur une nouvelle donnée $x$, c'est-à-dire trouver l'annotation $\widehat{y}$ qui maximise $p(y|x) = \frac{p(x,y)}{p(x)}$ : $\widehat{y} = argmax_y \ p(x, y)$

# 3. Annnoter des chaînes avec les HMM




$p(x =$une bonne soupe$, y = Det\ Adj\ Nom) = 1/3{*}1/3{*}1/2{*}2/3{*}2/3 = 2/81$

$p(x =$une bonne soupe$, y = Det\ Nom\ VIntr) = 1/3{*}2/3{*}1/3{*}1{*}1/5 = 2/135$

$\Longrightarrow$ la meilleure annotation de $x =$une bonne soupe est $y = Det\ Adj\ Nom$

# 3. Annnoter des chaînes avec les HMM

Méthodes classiques

- pour évaluer les probabilités d'émissions/de transitions à partir des exemples $(x, y)$ : Baum-Welsh (bibliothèques disponibles)
- pour trouver la meilleure annotation $y$ associée à un nouveau $x$ : programmation dynamique (idem)

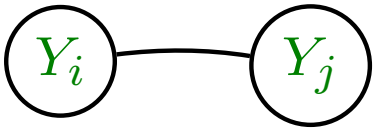
Problèmes avec les HMMs

- le modèle est génératif : on modélise comment obtenir $x$ alors que les données $x$ sont disponibles...
- les dépendances sur l'observation sont très limitées
- problème du "biais de labels"

# Plan

1. Annoter pour quoi faire
2. Apprendre à annoter : pourquoi, comment ?
3. Apprendre avec un modèle graphique
4. Annnoter des chaînes avec les HMM
5. Les CRF et leur application aux chaînes
6. CRF sur les arbres
7. Conclusion

## Premières propriétés des CRF

- modèle markovien : $\forall i, p(Y_i|X, \{Y_{j\neq i}\}) = p(Y_i|X, \{$  $\})$

- dans ce cas, on a (Hammersley-Clifford 71) :

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \psi_c(y_c, x)$$

  - $\mathcal{C}$ est l'ensemble des cliques du graphe sur $Y$
  - $y_c$ : valeurs des variables de $y$ sur la clique $c$
  - $Z(x)$ un coefficient de normalisation
  - chaque $\psi_c(y_c, x)$ est une fonction de potentiels
- chaque CRF est un modèle graphique markovien non dirigé (donc non génératif)
- chaque $Y_i$ est toujours reliée à tous les $X_j$ dans le graphe
- il reste à définir un graphe sur les annotations $Y_i$

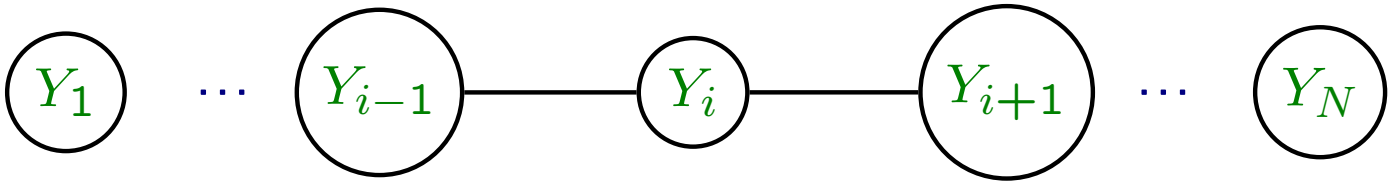
## La formule

- proposition de (Lafferty, McCallum et Pereira 01) :

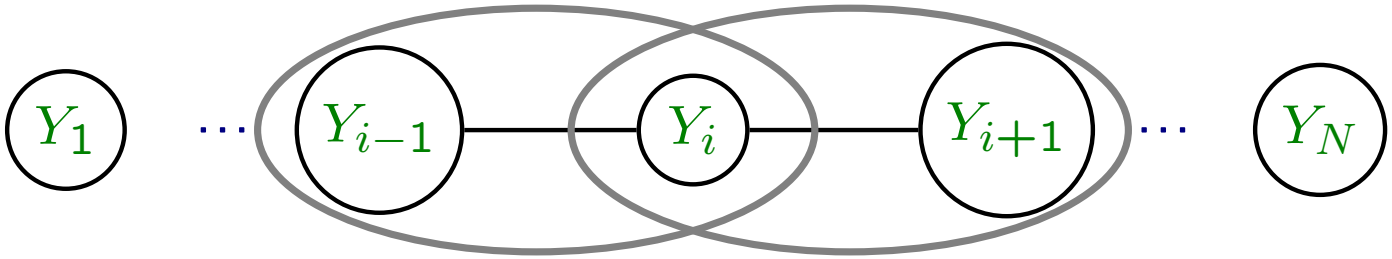
$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \exp\left( \sum_k \lambda_k f_k(y_c, x, c) \right)$$

- chaque $f_k$ est une fonction "feature" donnée par l'utilisateur
- c'est le même ensemble de $f_k$ qui sert pour chaque clique
- chaque $\lambda_k$ est un poids initialement inconnu associé à $f_k$

## Les CRF "linéaires"

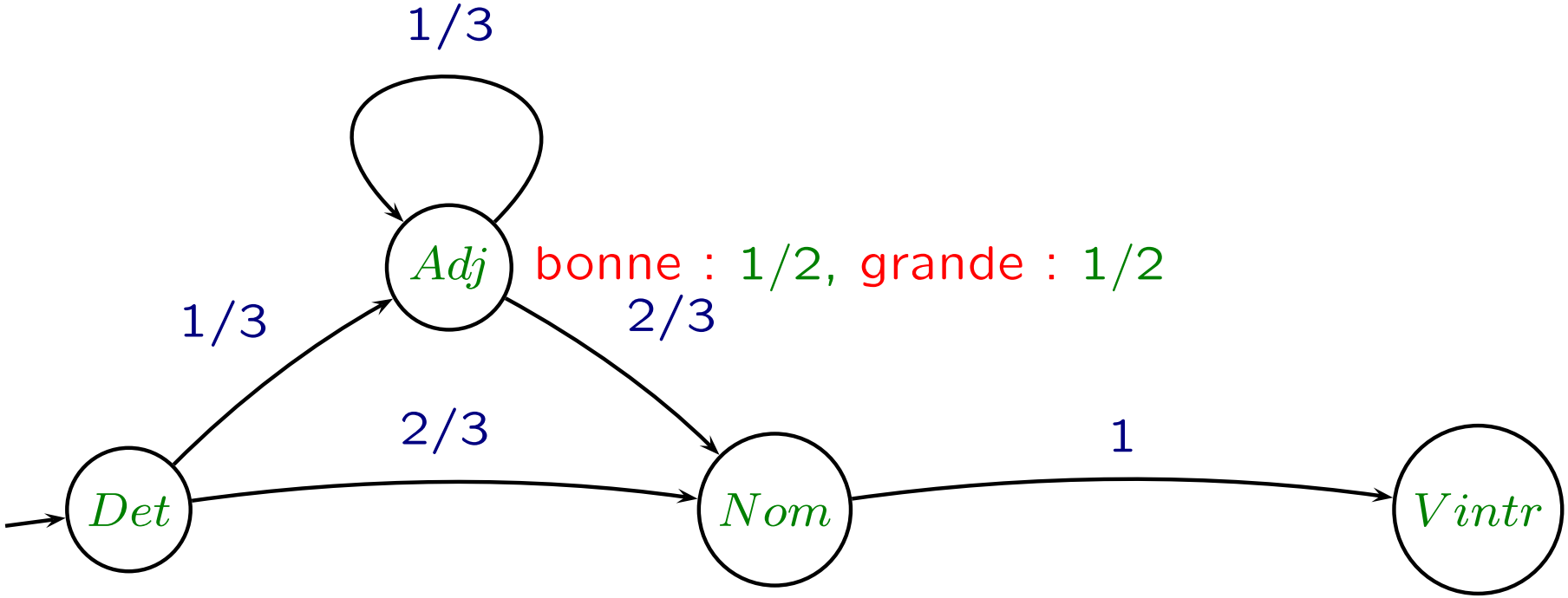


Les CRF "linéaires"




- le graphe sur $Y$ est une chaîne linéaire du 1er ordre
- implicite : chaque $Y_i$ relié à chaque $X_j$
- les cliques sont les variables $Y_i$ et les couples $(Y_{i-1}, Y_i)$ (en gris)
- exemples de features $f_k(y_c, x, c)$ sur des séquences à la position $i$ :
  * $f_k(y_{i-1}, y_i, x, i) = 1$ si $x_{i-1} \in \{la, une\}$ et $y_{i-1} = Det$ et $y_i = Nom$
    $= 0$ sinon
  * $f_{k'}(y_{i-1}, y_i, x, i) = 1$ si $\{M., Mme, Melle\} \cap \{x_{i-3}, ..., x_{i-1}\} \neq \emptyset$
    et $y_i = EN$
    $= 0$ sinon

Transformer un HMM en un CRF




- $f_1(y_i, x, 1) = 1$ si $y_i = Det$ et $x_i = la$ (et $= 0$ sinon), $\lambda_1 = log(2/3)$
- $f_2(y_{i-1}, y_i, x, 1) = 1$ si $y_{i-1} = Det$ et $y_i = Adj$ (et $= 0$ sinon), $\lambda_2 = log(1/3)$ (si la transition est nulle $\lambda = -\infty$)
- Toute distribution de proba d'un HMM s'obtient avec un tel CRF

Les features : d'où viennent-elles ?

- elles traduisent des connaissances externes
  - transformation d'une ressource en feature : si un dictionnaire donne pour chaque item $x$ ses $k$ étiquettes $y^k$ possibles, soit $f_k(y_i, x, i) = 1$ si $x_i = x$ et $y_i = y^k$ (et $= 0$ sinon)
  - chaque propriété des items de $x$ peut être traduite en feature : commencer par une majuscule ou contenir des chiffres (pour les EN), avoir ses $n$ dernières lettres dans une liste de flexions....
- elles proviennent de patrons instanciés par les exemples $(x, y)$ :

$$f(y_i, y_{i-1}, x, i) = 1 \text{ si } y_{i-1} = \ldots \text{ et } y_i = \ldots \text{ et } x_{i-2} = \ldots \text{ et } x_{i-1} = \ldots \text{ et } x_i = \ldots$$
$$= 0 \text{ sinon}$$

Ce que signifie programmer un CRF

- problème de l'apprentissage :
  - données : un ensemble $S$ de couples $(x, y)$
  - problème : trouver les paramètres $\lambda_k$ qui rendent le mieux compte des données
  - méthode : chercher à maximiser la "log-vraisemblance" :

$$log(\prod_{(x,y)\in S} p(y|x)) = \sum_{(x,y)\in S} log\ p(y|x)$$

    en cherchant $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k)$ où la dérivée s'annule...
- problème de l'annotation :
  - données : un CRF, une nouvelle donnée $x$
  - problème : trouver le $y$ qui maximise $p(y|x)$
- bibliothèques gratuites disponibles : Mallet, CRFSuite, CRF++...

Synthèse pour utiliser un CRF linéaire

- il faut fournir des exemples $(x, y)$
- il faut redéfinir sous forme de features ses connaissances
- il faut lancer l'apprentissage des $K$ poids $\lambda_k$ :
  - en maximisant la log-vraisemblance
  - méthode : descente de gradient
  - calcul en $K * N * |Y|^2$, peut être long ($|Y| = 100$, $K > 1\ 000\ 000$)
- une fois le modèle fixé, il permet de trouver l'annotation $y$ qui maximise $p(y|x)$ pour tout nouveau $x$
- grande souplesse pour enchaîner/combiner des apprentissages
- les CRF donnnent les meilleurs résultats actuels pour les tâches d'ingénierie linguistiques sur les chaînes

# Plan

1. Annoter pour quoi faire
2. Apprendre à annoter : pourquoi, comment ?
3. Apprendre avec un modèle graphique
4. Annnoter des chaînes avec les HMM
5. Les CRF et leur application aux chaînes
6. CRF sur les arbres
7. Conclusion

## Motivations (rappel)

- Nombreux documents structurés maintenant disponibles :
  - pages HTML/XML
  - corpus arborés : ensemble de phrases analysées syntaxiquement
- Tâches applicables sur les documents structurés :
  - extraction de nœuds
  - annotation de documents structurés (schémas médiateurs)
  - transformation de documents structurés (flux RSS...)
- Tâches applicables sur les corpus arborés :
  - annotation fonctionnelle : sujet, objet, etc.
  - annotation thématique : agent, patient, etc.
  - transformation d'arbres : transformation actif/passif, ou voie affirmative/voie interrogative, traduction automatique...

# 5. CRF sur les arbres

## Le graphe pour les arbres

- les variables $X_1 ... X_n$ et $Y_1 ... Y_n$ s'identifient aux nœuds d'un arbre
- quel graphe choisir sur $Y$ ? trois possibilités :
  - 1-CRF : chaque $Y_i$ est indépendant de chaque autre
  - 2-CRF : chaque $Y_i$ est relié à son père et à son ou ses fils
  - 3-CRF : chaque $Y_i$ est relié à son père et à ses "frères immédiats"
- pour chaque cas : une forme de cliques différente
- complexité des calculs croissante...

# 5. CRF sur les arbres

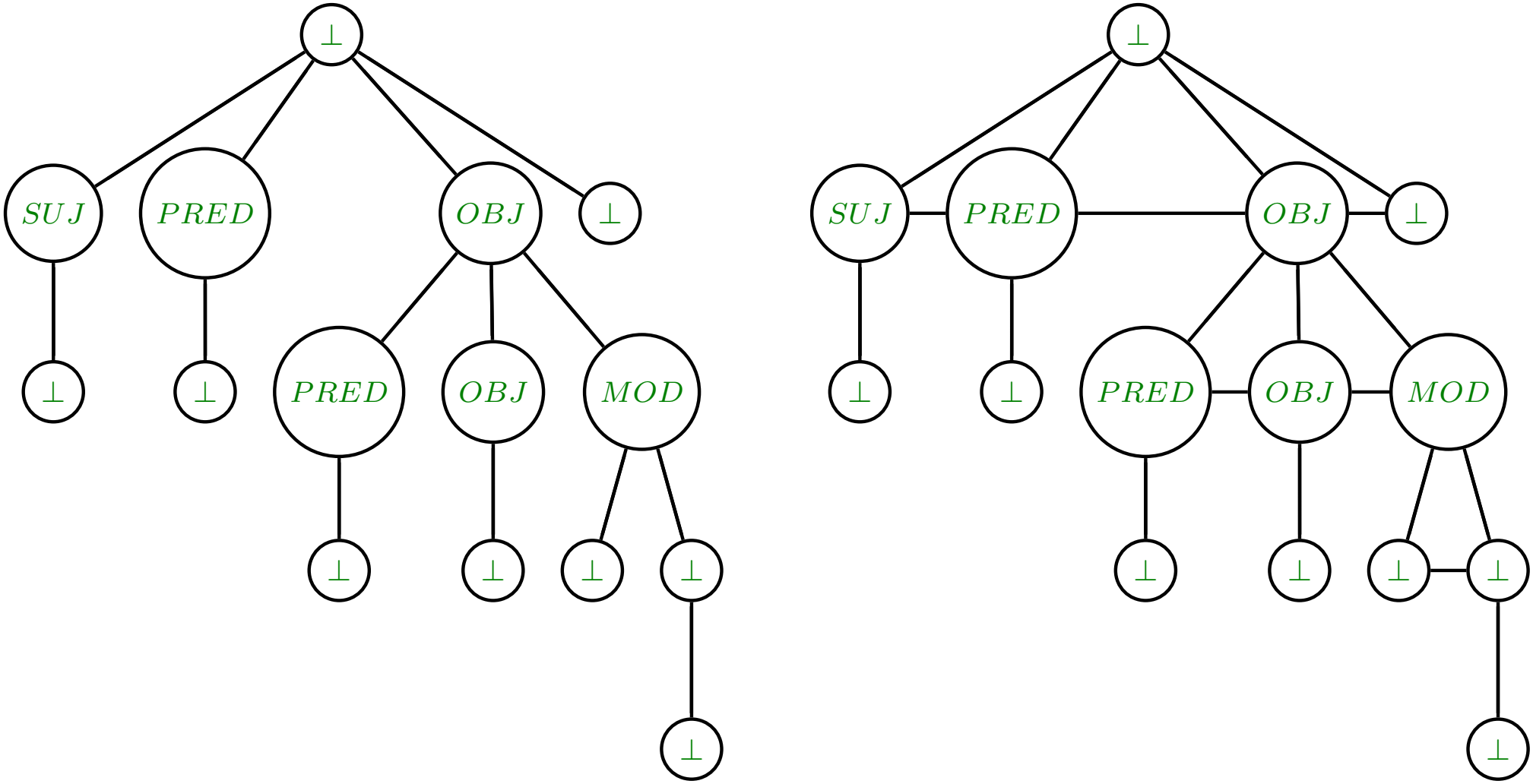



Figure 1: graphe pour un 2-CRF (à gauche) et un 3-CRF (à droite)

Variante 3-CRF

- les algorithmes d'apprentissage doivent être adaptés :
  - apprentissage : toujours par descente de gradients mais calculs en $N * K * |Y|^3$
  - annotation : programmation dynamique avec récursivité sur les 2 dimensions (horizontale/verticale)
- bibliothèque XCRF développée à Lille (Gilleron et alii 08)
- théorème : toutes les distributions de probas exprimables par les runs d'un automate d'arbre probabiliste (sur des arbres binaires) sont exprimables par un 3-CRF

# Plan

1. Annoter pour quoi faire
2. Apprendre à annoter : pourquoi, comment ?
3. Apprendre avec un modèle graphique
4. Annnoter des chaînes avec les HMM
5. Les CRF et leur application aux chaînes
6. CRF sur les arbres
7. Conclusion

# 5. Conclusion

Pour apprendre à annoter avec un modèle graphique, il faut

1. redéfinir la tâche comme une tâche d'annotation d'items :
   - identifier les items (mots, séparateurs, nœuds,...)
   - définir le vocabulaire d'annotation (POS, typage...)
   - ces modèles n'ont de sens que s'il y a des dépendances entre annotations
2. choisir le modèle et lui fournir de quoi fixer ses paramètres
   - avoir d'un échantillon d'apprentissage $S = \{(x, y)\}$
   - pour les HMM : choisir une structure, pour les CRF : définir des features
   - choisir une bibliothèque pour l'apprentissage/ l'annotation
3. pour évaluer la qualité du résultat : intégrer l'expérience dans un protocole d'apprentissage (validation croisée)