CYK Probabiliste

Lapraye & Lévêque & Viegas

Paris VII

30 juin 2016

Les PCFG

- Les CFG : un quadruplet (Σ, V, S, P)
- Les CFG pondérées : ajout d'une fonction de poids $f : p \mapsto \alpha, w \in W, \alpha \in \mathbb{R}$
- Les CFG probabilistes : les poids correspondent à des probabilités pour une réécriture donnée.

$$f : p \mapsto \alpha, p \in P, \alpha \in [0, 1]$$

$$\forall X \in V, \sum_{X \to \alpha \in [0,1]} p(X \to \alpha) = 1$$

- Les CFG probabilistes représentent un modèle de prédiction déduit à partir du corpus dont elles sont extraites.

Les PCFG

- Les CFG : un quadruplet (Σ, V, S, P)
- Les CFG pondérées : ajout d'une fonction de poids $f : p \mapsto \alpha, w \in W, \alpha \in \mathbb{R}$
- Les CFG probabilistes : les poids correspondent à des probabilités pour une réécriture donnée.

$$f : p \mapsto \alpha, p \in P, \alpha \in [0, 1]$$

$$\forall X \in V, \sum_{X \to \alpha \in [0,1]} p(X \to \alpha) = 1$$

- Les CFG probabilistes représentent un modèle de prédiction déduit à partir du corpus dont elles sont extraites.

Les PCFG

- Les CFG : un quadruplet (Σ, V, S, P)
- Les CFG pondérées : ajout d'une fonction de poids $f : p \mapsto \alpha, w \in W, \alpha \in \mathbb{R}$
- Les CFG probabilistes : les poids correspondent à des probabilités pour une réécriture donnée.

$$f : p \mapsto \alpha, p \in P, \alpha \in [0, 1]$$

$$\forall X \in V, \sum_{X \to \alpha \in [0,1]} p(X \to \alpha) = 1$$

- Les CFG probabilistes représentent un modèle de prédiction déduit à partir du corpus dont elles sont extraites.

Les PCFG

- Les CFG : un quadruplet (Σ, V, S, P)
- Les CFG pondérées : ajout d'une fonction de poids $f : p \mapsto \alpha, w \in W, \alpha \in \mathbb{R}$
- Les CFG probabilistes : les poids correspondent à des probabilités pour une réécriture donnée.

$$f : p \mapsto \alpha, p \in P, \alpha \in [0, 1]$$

$$\forall X \in V, \sum_{X \to \alpha \in [0,1]} p(X \to \alpha) = 1$$

- Les CFG probabilistes représentent un modèle de prédiction déduit à partir du corpus dont elles sont extraites.

L'Algorithme CYK

- Un algorithme de parsing ascendant

L'Algorithme CYK

- Un algorithme de parsing ascendant
- Complexité $\mathcal{O}(|G|n^3)$

L'Algorithme CYK

- Un algorithme de parsing ascendant
- Complexité $\mathcal{O}(|G|n^3)$
- Parsing tabulaire

L'Algorithme CYK

- Un algorithme de parsing ascendant
- Complexité $\mathcal{O}(|G|n^3)$
- Parsing tabulaire
- Extention aux grammaire hors-contexte probabilistes (PCFG)

L'Algorithme CYK

La forme normale de Chomsky (CNF)

- l'axiome S est inaccessible
- Les règles de production adoptent une des formes suivantes :

$$A \rightarrow BC$$

$$D \rightarrow e$$

$$S \rightarrow \varepsilon$$

, avec

$$A, B, C, D \in V, e \in \Sigma$$

et ε la production vide.

Transformer la grammaire en CNF

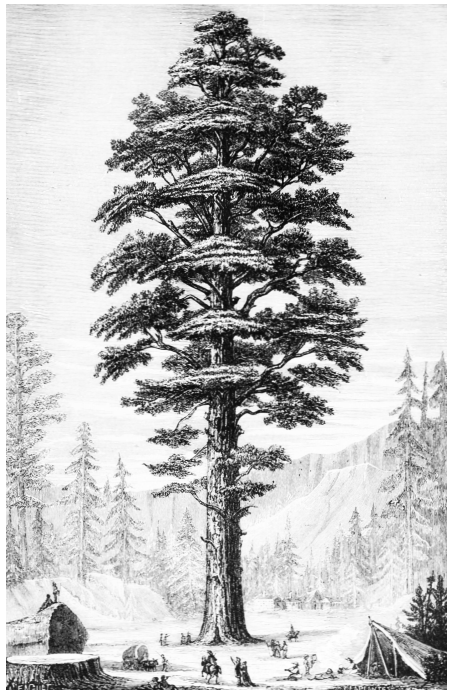
1. Faire en sorte que l'axiome n'apparaisse plus dans les parties droites de règles
2. Supprimer les règles d'effacement (c'est à dire de la forme $A \rightarrow^* \varepsilon$) pour les non-terminaux autres que l'axiome.
3. Faire en sorte que tout les terminaux apparaissent uniquement dans la partie droite de règles unaires
4. Remplacer les règles de production n-aire par des règles binaires équivalentes.
5. Supprimer les productions singulières de non-terminaux, c'est à dire les règles de la forme $A \rightarrow B$ avec $A, B \in V$

Transformer la grammaire en CNF

1. Faire en sorte que l'axiome n'apparaisse plus dans les parties droites de règles
2. Supprimer les règles d'effacement (c'est à dire de la forme $A \rightarrow^* \varepsilon$) pour les non-terminaux autres que l'axiome.
3. Faire en sorte que tout les terminaux apparaissent uniquement dans la partie droite de règles unaires
4. Remplacer les règles de production n-aire par des règles binaires équivalentes.
5. Supprimer les productions singulières de non-terminaux, c'est à dire les règles de la forme $A \rightarrow B$ avec $A, B \in V$

Le corpus Sequoia

- Un corpus diversifié
- Des phrases de longueur variable
-



Notre implémentation du CYK

Evaluation

References

Brian Roark, Richard Sproat.

Computational Approaches to Morphology and Syntax.

Oxford University Press, 2007.

Mariana Romanyshyn, Vsevolod Dyomkin.

The Dirty Little Secret of Constituency Parser Evaluation, 2014.

http ://tech.grammarly.com/blog/posts/The-Dirty-Little-Secret-of-Constituency-Parser-Evaluation.html

Martin Lange, Hans Leiss

« To CNF or not to CNF : An Efficient Yet Presentable Version of the CYK Algorithm », 2009

Informatica Didactica N° 8

E. Black, S.Abney et al.

« Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars »

1991, DARPA Speech and Natural Language Workshop