

Modèle de langage

- Dans le modèle de bayes naïf multinomial, on peut distinguer deux parties

$$P([W_1, \dots, W_d], C) = P(C) \prod_i P(W_i | C)$$

modèle des catégories

modèle de langage

- Un **modèle de langage est une distribution sur du texte**, c.-à-d. sur des séquences de mots
 - ◆ étant donné un texte $[w_1, \dots, w_d]$, lui assigne une probabilité $P([w_1, \dots, w_d])$
- Dans le modèle de bayes naïf multinomial, le modèle de langage est très simple
 - ◆ les mots sont générés indépendamment les uns des autres (étant donnée la catégorie C)

Modèle n -gramme

- Un meilleur modèle générerait le i^{e} mot d'une phrase au moins à partir des quelques mots précédents dans la phrase

$$P([W_1, \dots, W_d]) = \prod_i P(W_i \mid \underbrace{W_{i-n+1}, \dots, W_{i-1}})$$

$n-1$ mots précédents



- On appelle de tels modèles de langage des **modèles n -gramme**
 - ◆ un **n -gramme est une sous-séquence de n mots**, extraite d'un corpus
 - ◆ on les appelle modèles n -gramme parce qu'ils **sont estimés à partir des fréquences de tous les n -grammes** d'un corpus
- Ces modèles sont en fait des **modèles (chaînes) de Markov d'ordre $n-1$**

Modèle n -gramme

- Exemple: dans le document

« Perceptron , un OOV OOV apprentissage OOV »

il y a:

- ◆ 7 **unigrammes** ($n=1$) dont 5 différents

« Perceptron »

« , »

« un »

« OOV »

« OOV »

« apprentissage »

« OOV »

Modèle n -gramme

- Exemple: dans le document

« Perceptron , un OOV OOV apprentissage OOV »

il y a:

- ◆ 6 **bigrammes** ($n=2$), tous différents

(« Perceptron », « , »)

(« , », « un »)

(« un », « OOV »)

(« OOV », « OOV »)

(« OOV », « apprentissage »)

(« apprentissage », « OOV »)

Modèle n -gramme

- Exemple: dans le document

« Perceptron , un OOV OOV apprentissage OOV »

il y a:

- ◆ 5 **trigrammes** ($n=3$), tous différents

(« Perceptron », « , », « un »)

(« , », « un », « OOV »)

(« un », « OOV », « OOV »)

(« OOV », « OOV », « apprentissage »)

(« OOV », « apprentissage », « OOV »)

- ◆ etc.

- Tendence historique des n -grammes:
<http://books.google.com/ngrams>

Apprentissage de modèle n -gramme

- On peut apprendre un modèle n -gramme à partir des fréquences de n -grammes dans un corpus de documents D_t

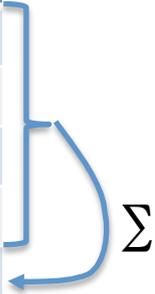
$$P(W_i = w \mid w_{i-n+1}, \dots, w_{i-1}) = \frac{\text{nb. de fois que } w \text{ suit les mots } w_{i-n+1}, \dots, w_{i-1}}{\text{nb. de fois que } w_{i-n+1}, \dots, w_{i-1} \text{ est suivi d'un mot}}$$
$$= \frac{\sum_t \text{freq}((w_{i-n+1}, \dots, w_{i-1}, w), D_t)}{\sum_t \text{freq}((w_{i-n+1}, \dots, w_{i-1}, *), D_t)}$$

mot quelconque 

Apprentissage de modèle n -gramme

- Exemple: soit les fréquences totales suivantes

n -gramme	freq(n -gramme, D)
(« modèle », « de », « Bayes »)	5
(« modèle », « de », « Markov »)	25
(« modèle », « de », « langage »)	10
...	...
(« modèle », « de », *)	200



- Alors le modèle trigramme assignerait les probabilités:

$$\begin{aligned} P(W_i = \text{« Bayes »} \mid W_{i-2} = \text{« modèle »}, W_{i-1} = \text{« de »}) &= 5 / 200 \\ P(W_i = \text{« Markov »} \mid W_{i-2} = \text{« modèle »}, W_{i-1} = \text{« de »}) &= 25 / 200 \\ P(W_i = \text{« langage »} \mid W_{i-2} = \text{« modèle »}, W_{i-1} = \text{« de »}) &= 10 / 200 \end{aligned}$$

...

...

Lissage de modèle n -gramme

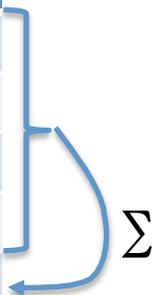
- On peut également lisser les modèles n -gramme en général
 - ◆ encore plus important, puisque plus un n -gramme est long, moins il sera fréquent
 - ◆ la plupart des n -grammes imaginables auront une fréquence de zéro, pour n grand
- Première approche: **lissage δ**

$$P(W_i = w \mid w_{i-n+1}, \dots, w_{i-1}) = \frac{\delta + \sum_t \text{freq}(w_{i-n+1}, \dots, w_{i-1}, w), D_t}{\delta (|V| + 1) + \sum_t \text{freq}(w_{i-n+1}, \dots, w_{i-1}, *), D_t}$$

Lissage δ

- Exemple: soit les fréquences totales suivantes

n -gramme	freq(n -gramme, D)
(« modèle », « de », « Bayes »)	5
(« modèle », « de », « langage »)	10
(« modèle », « de », « langue »)	0
...	...
(« modèle », « de », *)	200



- Trigramme avec lissage $\delta = 0.1$ et un vocabulaire de taille $|V|=999$

$$P(W_i = \text{« Bayes »} \mid W_{i-2} = \text{« modèle »}, W_{i-1} = \text{« de »}) = (0.1+5) / (100+200) = 5.1 / 300$$

$$P(W_i = \text{« langage »} \mid W_{i-2} = \text{« modèle »}, W_{i-1} = \text{« de »}) = (0.1+10) / (100+200) = 10.1 / 300$$

$$P(W_i = \text{« langue »} \mid W_{i-2} = \text{« modèle »}, W_{i-1} = \text{« de »}) = (0.1+0) / (100+200) = 0.1 / 300$$

...

...

Lissage par interpolation linéaire

- Deuxième approche: **lissage par interpolation linéaire**

- ◆ faire la moyenne (pondérée) de modèles unigrammes, bigrammes, trigrammes, ... jusqu'à n -gramme

$$P_{\lambda}(W_i = w \mid w_{i-n+1}, \dots, w_{i-1}) = \lambda_1 P(W_i = w) + \lambda_2 P(W_i = w \mid w_{i-1}) + \lambda_3 P(W_i = w \mid w_{i-2}, w_{i-1}) + \dots + \lambda_n P(W_i = w \mid w_{i-n+1}, \dots, w_{i-1})$$

où $\sum_i \lambda_i = 1$

- Exemple:

- ◆ le trigramme (« modèle », « de », « langue ») a une fréquence de 0
- ◆ le bigramme (« de », « langue ») est présent dans le corpus
- ◆ alors $P_{\lambda}(W_i = w \mid w_{i-n+1}, \dots, w_{i-1}) > 0$, en autant que λ_2 ou $\lambda_1 > 0$