Numéro de candidat : 17526

Modélisation probabiliste du langage :

Définition d'un modèle de langage :

- Langage
- Corpus/Vocabulaire
- Modèle de langage
- Ponctuation : <.s>

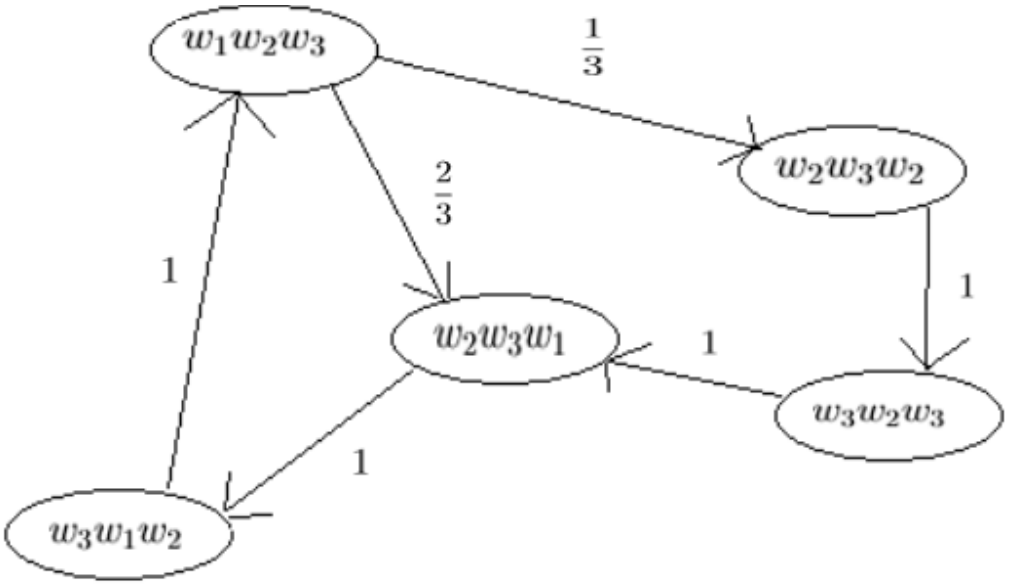
$$w_i w_{i+1} w_{i+2} w_{i+3} w_{i+4} w_{i+5}$$

$$C, V$$

$$c(w_i)\ c(w_i w_{i+1})\ c(w_i w_{i+1} w_{i+2})$$

Le modèle N-gram :

- Chaîne de Markov
- Etats considérés
- Transitions
- On utilise les n-1 mots précédents



Probabilité d'un Ngram :

- Fréquence d'apparition dans un corpus
- Définition de fréquences conditionnelles
- On utilise au plus le modèle trigramme

$$f(w_i) = \frac{c(w_i)}{\sum_{w_i \in V} c(w_i)}$$

$$f_{w_i}(w_j) = \frac{c(w_i w_j)}{\sum_{w_j \in V} c(w_i w_j)} = \frac{c(w_i w_j)}{c(w_i)}$$

$$f_{w_i w_j}(w_k) = \frac{c(w_i w_j w_k)}{c(w_i w_j)}$$

Lissage additif :

- Redistribution de la masse de probabilité
- Modifications de toutes les probabilités
- Raisonnement

$$P(w_k) = \frac{c(w_k) + \alpha}{\sum_{w \in V} (c(w) + \alpha)} = \frac{c(w_k) + \alpha}{N + \alpha * card(V)}$$

$$P_{w_j}(w_k) = \frac{c(w_j w_k) + \alpha}{c(w_j) + \alpha * card(V)}$$

$$P_{w_i w_j}(w_k) = \frac{c(w_i w_j w_k) + \alpha}{c(w_i w_j) + \alpha * card(V)}$$

Lissage de Jelinek-Mercer :

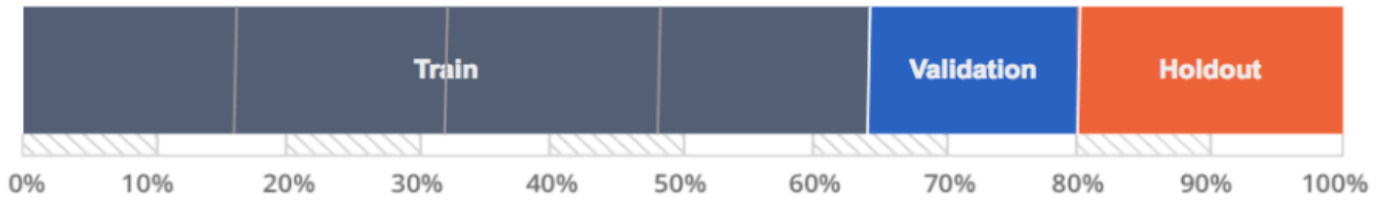
$$P_{w_i w_j}(w_k) = \lambda_3 f_{w_i w_j}(w_k) + \lambda_2 f_{w_j}(w_k) + \lambda_1 f(w_k)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

- Les coefficients sont les poids associés aux modèles
- Ils peuvent être considérés comme indépendants du contexte
- On interpole des fréquences lissées

Détermination des coefficients : mise en place :

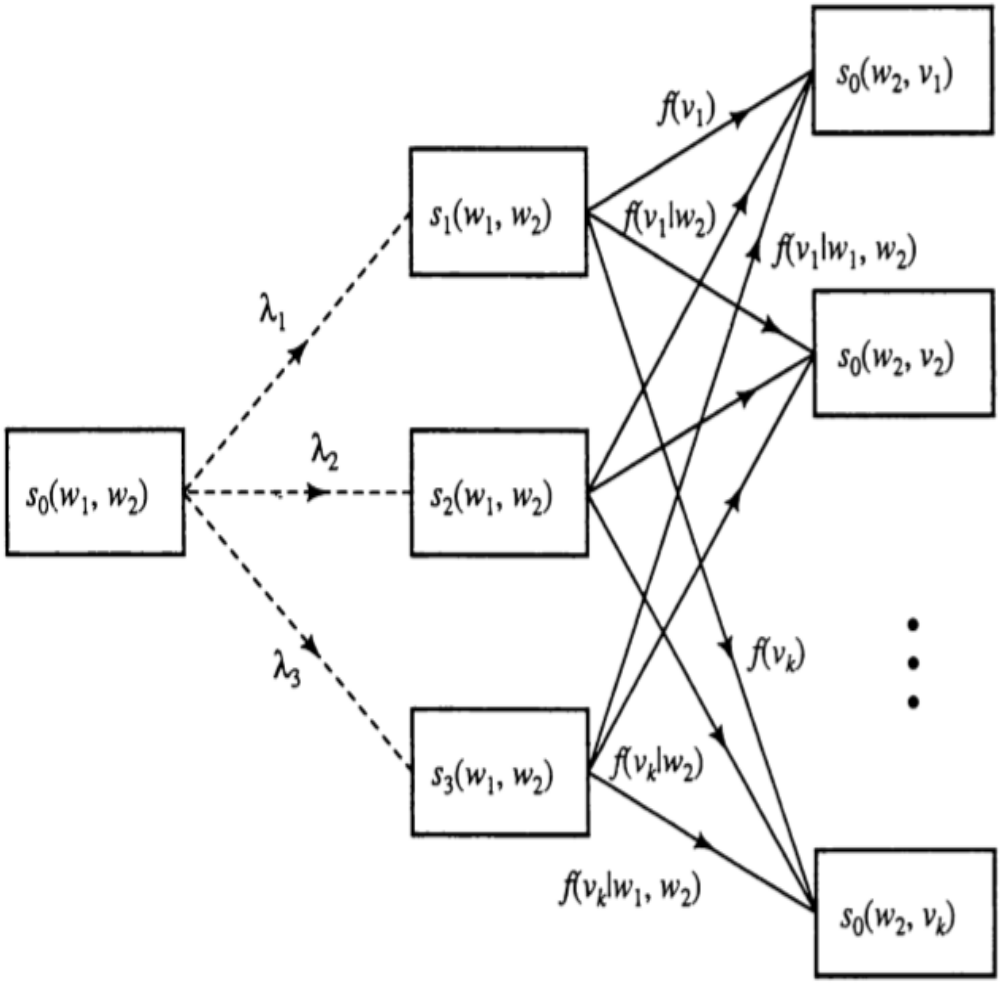
- Méthode de « deleted interpolation »
- Procédé de la validation croisée



Provient de www.datarobot.com

Détermination des coefficients : mise en place :

- Certaines transitions ont la même probabilité
- On lie les transitions entre elles



Provient de Statistical methods for speech recognition de Frederick Jelinek

Détermination des coefficients :

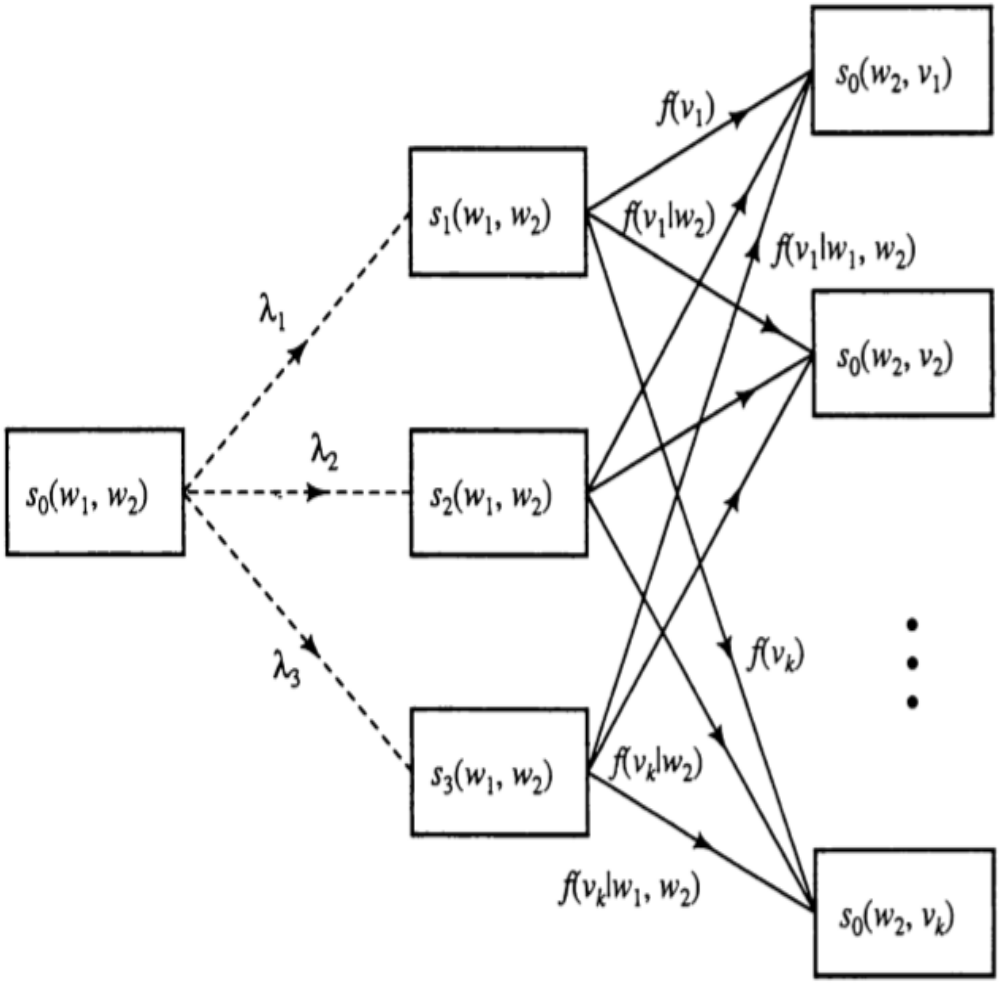
On se place à l'itération numéro l+1 et on parcourt la séquence : $w_1 w_2 w_3 \ldots w_{n-2} w_{n-1} w_n$

$$p(t_1) = \lambda_1^{(l)}$$
$$p(t_2) = \lambda_2^{(l)}$$
$$p(t_3) = \lambda_3^{(l)}$$

$$P_{w_i w_{i+1}}(w) = \lambda_1^{(l)} f(w) + \lambda_2^{(l)} f_{w_{i+1}}(w) + \lambda_3^{(l)} f_{w_i w_{i+1}}(w)$$

$$\lambda_1^{(l)} \frac{f(w)}{P_{w_i w_{i+1}}(w)} + \lambda_2^{(l)} \frac{f_{w_{i+1}}(w)}{P_{w_i w_{i+1}}(w)} + \lambda_3^{(l)} \frac{f_{w_i w_{i+1}}(w)}{P_{w_i w_{i+1}}(w)} = 1$$

$$P(t^{(i)} = t_1) = \lambda_1^{(l)} \frac{f(w_{i+2})}{P_{w_i w_{i+1}}(w_{i+2})}$$



Provient de Statistical methods for speech recognition de Frederick Jelinek

Détermination des coefficients :

$$c(t_1) = \sum_{k=0}^{n-2} P(t^{(k)} = t_1)$$

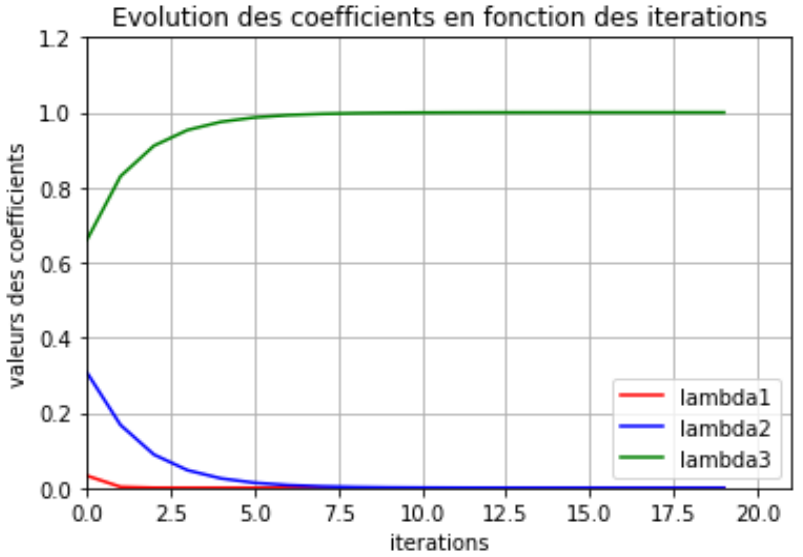
$$p_{reestime}(t_1) = \frac{c(t_1)}{c(t_1) + c(t_2) + c(t_3)} = \lambda_1^{(l+1)}$$

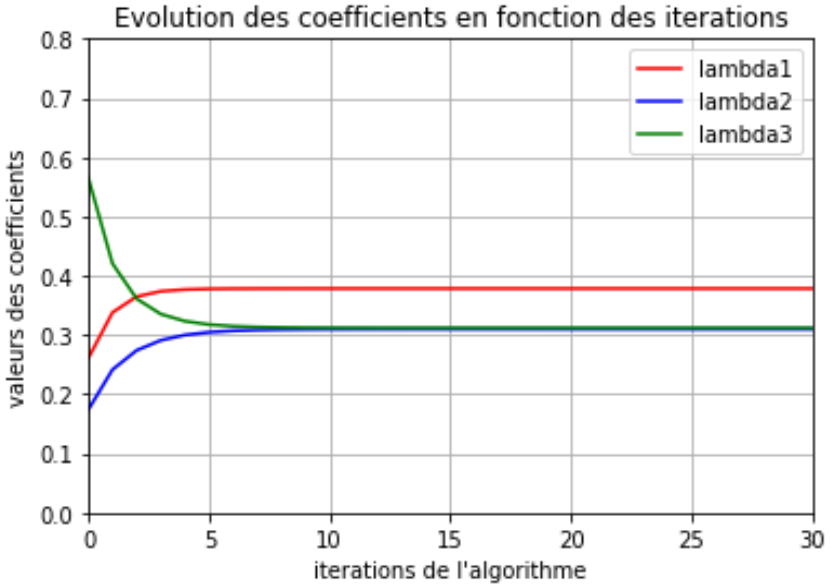
$$\lambda_1^{(l+1)} = \lambda_1^{(l)} \sum_{k=0}^{n-2} \frac{f(w_{k+2})}{\lambda_1^{(l)} f(w_{k+2}) + \lambda_2^{(l)} f_{w_{k+1}}(w_{k+2}) + \lambda_3^{(l)} f_{w_k w_{k+1}}(w_{k+2})}$$

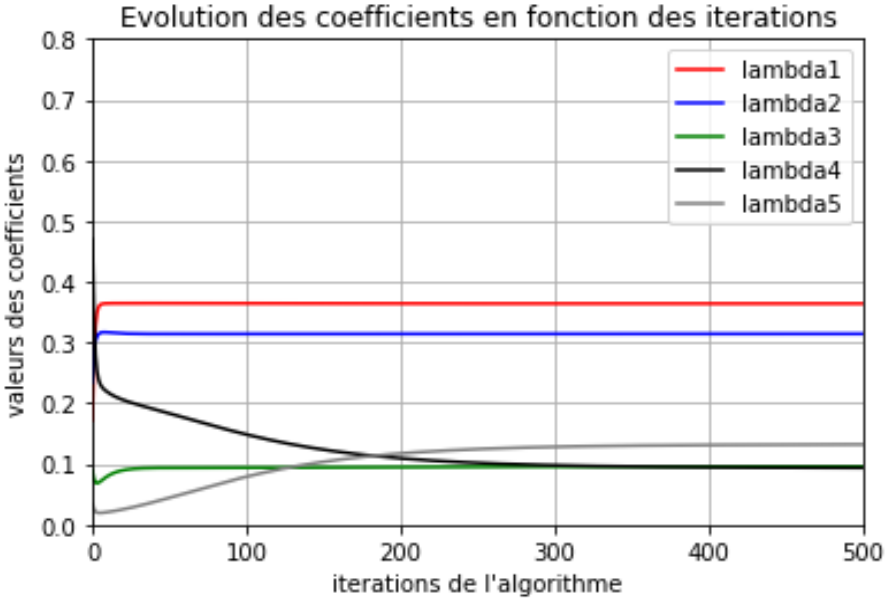
$$\lambda_2^{(l+1)} = \lambda_2^{(l)} \sum_{k=0}^{n-2} \frac{f_{w_{k+1}}(w_{k+2})}{\lambda_1^{(l)} f(w_{k+2}) + \lambda_2^{(l)} f_{w_{k+1}}(w_{k+2}) + \lambda_3^{(l)} f_{w_k w_{k+1}}(w_{k+2})}$$

$$\lambda_3^{(l+1)} = \lambda_3^{(l)} \sum_{k=0}^{n-2} \frac{f_{w_k w_{k+1}}(w_{k+2})}{\lambda_1^{(l)} f(w_{k+2}) + \lambda_2^{(l)} f_{w_{k+1}}(w_{k+2}) + \lambda_3^{(l)} f_{w_k w_{k+1}}(w_{k+2})}$$

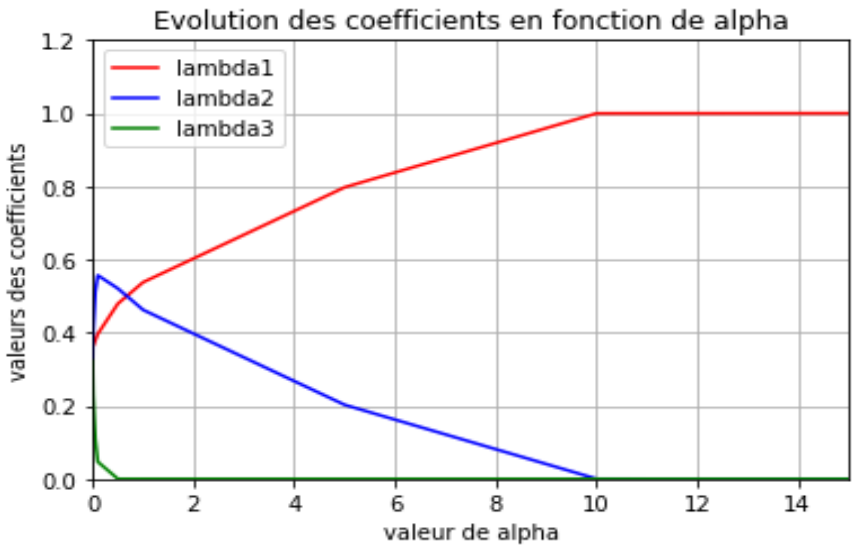
Détermination des coefficients :

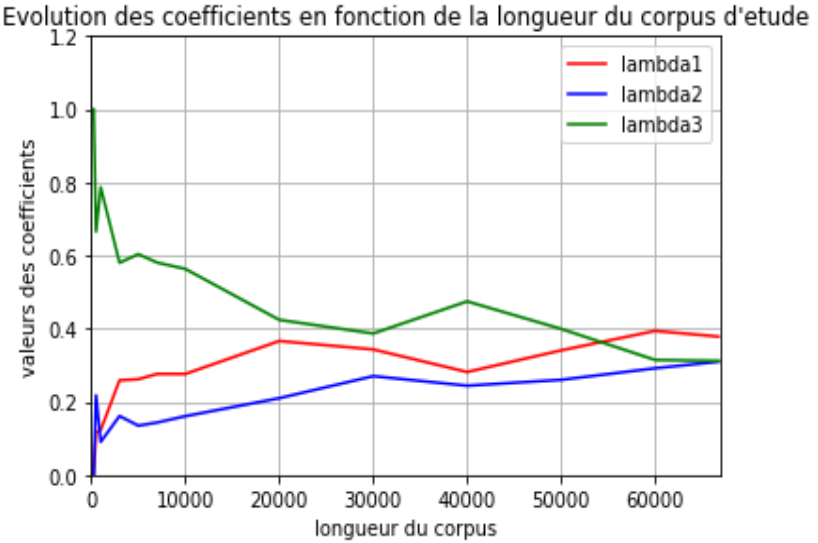






Variations des coefficients :





Création d'un modèle de langage :

- Ponctuation
- Génération aléatoire du début de la phrase
- Prend en compte le mot le plus probable d’après les précédents

- <.s> amanda drove the package to the store <.s>
- <.s> she was the book <.s>
- <.s> some of the book <.s>

Comparaison des modèles :

- Sur le corpus de test
- Entropie
- Taux d'entropie
- Théorème de Shannon-McMillan-Breiman
- Perplexité

$$C_{test} = w_1 w_2 w_3 ... w_{n-2} w_{n-1} w_n$$

$$H(W_1...W_m) = - \sum_{w_1^m \in C_{test}} P(w_1...w_m) log_2(P(w_1...w_m))$$

$$h(W_1...W_m) = \frac{H(W_1...W_m)}{m}$$

$$h_{C_{test}} = \lim_{m \to +\infty} \frac{H(W_1...W_m)}{m} = - \lim_{m \to +\infty} \sum_{w_1^m \in C_{test}} \frac{P(w_1...w_m)}{m} log_2(P(w_1...w_m))$$

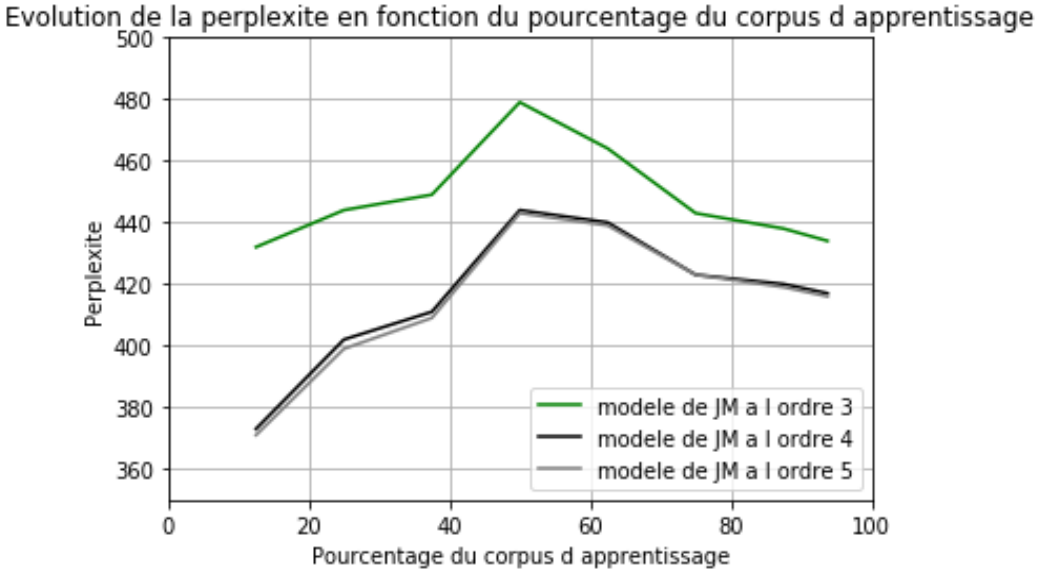
$$h_{C_{test}} = - \lim_{m \to +\infty} \frac{log_2(P(w_1...w_m))}{m}$$

$$h_{C_{test}} = - \frac{log_2(P(w_1...w_n))}{n}$$

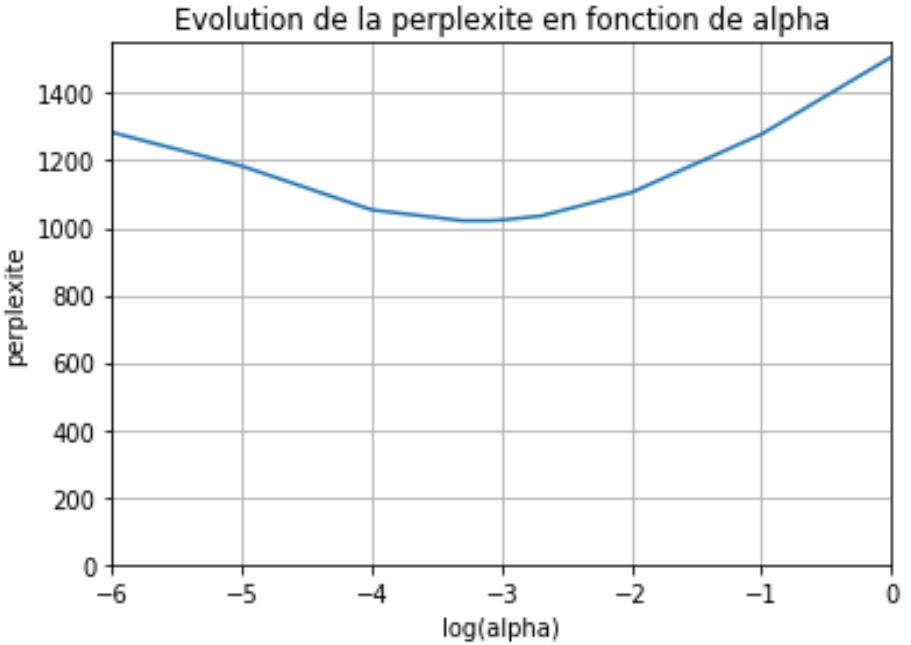
$$PP(C_{test}) = 2^{h_{C_{test}}} = \frac{1}{\sqrt[n]{P(w_1) P_{w_1}(w_2) \prod_{k=3}^{n} P_{w_{k-2} w_{k-1}}(w_k)}}$$

Performances du modèle :

- Corpus étudié



- Corpus de Brown



modèle	coefficients dépendants	coefficients indépendants	trigramme lissé
perplexité	244	1020	34000

Catégorisation de documents : entropie :

$$C_{doc} = w_1 w_2 ... w_n$$

$$H(P,Q) = H(W_1...W_n) + KL(P \,||\, Q)$$

$$KL(P \,||\, Q) = \sum_{w_1^m \in C_{doc}} P(w_1...w_m) log_2(\frac{P(w_1...w_m)}{Q(w_1...w_m)})$$

$$H(P,Q) = -\sum_{w_1^m \in C_{doc}} P(w_1...w_m) log_2(Q(w_1...w_m))$$

$$h(P,Q) = -\lim_{m \to +\infty} \sum_{w_1^m \in C_{doc}} \frac{P(w_1...w_m)}{m} log_2(Q(w_1...w_m))$$

$$h(P,Q) = -\frac{1}{n} log_2(Q(w_1...w_n))$$

Catégorisation de documents :

Test pour cinq articles sur l'économie

	Économie	Politique	Science	Sport
Article 1	568	954	959	1184
Article 2	550	830	924	944
Article 3	506	640	900	967
Article 4	421	569	758	792
Article 5	556	703	693	875

Test pour cinq articles sur la politique

	Économie	Politique	Science	Sport
Article 1	676	444	889	882
Article 2	550	507	726	699
Article 3	726	643	879	833
Article 4	624	439	810	747
Article 5	602	633	754	709

Test pour cinq articles sur la science

	Économie	Politique	Science	Sport
Article 1	704	765	638	758
Article 2	696	810	558	819
Article 3	742	819	708	880
Article 4	702	792	534	812
Article 5	692	753	697	766

Test pour cinq articles sur le sport

	Économie	Politique	Science	Sport
Article 1	649	665	761	549
Article 2	535	770	749	753
Article 3	880	846	977	484
Article 4	821	730	1012	527
Article 5	768	801	829	480

FIN

Lissage de Good-Turing :

- Basé sur le nombre d'apparition du Ngram
- Le Ngram le plus fréquent a une probabilité nulle
- Seuil

$$P(w) = \frac{(r+1)N_{r+1}}{N_r N}, avec\ w \in V\ et\ r = c(w)$$

$$P_{w_i}(w) = \frac{(r+1)N_{r+1}}{N_r N}, avec\ w \in V\ et\ r = c(w_i w)$$

$$P_{w_i w_j}(w) = \frac{(r+1)N_{r+1}}{N_r N}, avec\ w \in V\ et\ r = c(w_i w_j w)$$