APNEE3

Ieva Petrulionyte, Dorian Thivolle

Nous avons implémenté un modèle de langage (LM) qui attribue des probabilités à des séquences de deux mots (bigrammes). Nous avons assigné des probabilités aux bigrammes du texte donné et généré un autre texte en estimant la probabilité du mot suivant par rapport au mot précédent.

Texte généré par le LM Pipotron

[1] Afin de gagner en moyenne (nlog(n)) . Cette étude nous n'avons pas ou 1000 pour verifier que celle du motif . Calcul du texte : nous même , et N2 . Commentaires : Nous avons du texte passé en paramètre divisé par n2 . Cela m'a permis de Karp-Rabin . Il faut tout point qu'il est de l'algorithme implémenté l'algorithme , j'ai lu diffère par insertion et N2 le programme RechercheMotif prend respectivement à l'indice I dans une table de façon exponentielle , contenant la version actuelle comporte pas instantanée , fichier de Karp-Rabin nous avons traité intégralité du texte à m opérations . Nous avons ensuite travailler sur les performances de tests , nous avons commencé par insertion vaut O(n2) . Nous aurions pu aller jusqu'à la courbe de temps , ainsi le coût entre deux conclusions possibles : Nous avons pu évaluer l'efficacité des données . Le but de KR est de hachage . En effet , désolés pour déterminer lequel est construit à la nouvelle valeure du motif est (nm +1)*m , lorsque le motif M de motif de comparaisons effectué? augmente encore la première boucle while correspond à l'algorithme selon le temps d'exécution est constant . L'analyse des problèmes de la boucle , de complexité au pire est encore plus performante que la version utilisant une seconde utilisant une idée d'ensemble . Puis , nous avons pu comprendre chaque case du calculer son coût maximum . Nous avons ensuite calculer un b .

Validation du comportement

	Texte d'apprentissage	*Texte généré 100*	*Texte généré 1000*	*Texte généré 100k*	*Texte généré 1m*
mot le plus fréquent	de	de	de	de	de
2_{ieme} *mot le plus fréquent*	.	,	.	.	.
3_{ieme}	,	caractère	le	,	,
4_{ieme}	le	.	,	le	le
5_{ieme}	la	le	la	la	la
6_{ieme}	et	que	et	et	et
7_{ieme}	que	un	pour	que	que
8_{ieme}	est	d'abord	temps	est	est
9_{ieme}	des	dans	des	des	des
10_{ieme}	du	pour	est	du	du

[1] Nous avons eu des problèmes d'encodage du texte (les accents français dans le texte). Nous avons essayé de changer les paramètres de langue, d'utiliser les fonctions java avec les options d'encodage du texte mais rien n'a fonctionné. Nous avons donc changé les mauvais caractères avec des lettres françaises correctes.

Analyse du modèle

Notre modèle approxime la probabilité d'un mot par rapport à tous les mots précédents en utilisant uniquement la probabilité conditionnelle d'un mot précédent $P(w_n|w_{n-1})$. Nous faisons donc une hypothèse "Markovienne" et obtenons l'approximation suivante :

$P(w_n|w_1, w_2, w_3 \dots w_{n-1}) \approx P(w_n|w_{n-1})$.

Nous utilisons le principe du maximum de vraisemblance (MLE) pour déterminer la probabilité de mot suivant. Pour cela nous calculons d'abord le nombre d'occurrences des bigrammes à partir des données de formation en les normalisant pour qu'ils soient compris entre 0 et 1. Cependant, il y a une sélection aléatoire à chaque étape, et donc on ne choisira pas toujours l'élément le plus probable.

Pour calculer une probabilité d'une bigramme particulière des deux mots w_{n-1} et w_n, nous calculons le nombre de bigrammes $C(w_{n-1}w_n)$ et le normalisons par la somme de tous les bigrammes qui partagent le même premier mot w_{n-1} (on simplifie en $C(w_{n-1})$ nombre d'occurrences de mot w_{n-1}).

$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$

Notre modèle pourrait être amélioré en utilisant des N-grammes d'ordre supérieur ($N > 2$). D'autre part, plus les N-grammes d'ordre supérieur sont élevés, plus la mémoire et le temps d'initialisation des probabilités de transition sont importants (exponentiel avec l'ordre de N). Nous pourrions également introduire un élément aléatoire que le programme produise un mot qui ne soit pas basé sur l'un des bigrammes des données de formation. Nous remarquons que le priori est uniforme dans notre estimation du mot suivant et nous pouvons donc imaginer d'inclure le priori dans notre modèle et d'obtenir des résultats légèrement différents, probablement plus proches de l'ensemble des données de formation.

Nombre d'occurrences des 15 mots les plus fréquents dans le texte d'apprentissage