

Travail noté #4 : Construisez un modèle de langage

Un modèle de langage est un outil mathématique qui permet d'établir la distribution statistique des mots : en présence (ou dans le contexte) de certains mots, quel mot a tendance à suivre, et dans quelle proportion des cas (liés. avec quelle probabilité). Un modèle de langage n'est pas un objet abstrait qui décrit une réalité théorique : il s'agit d'un modèle statistique entraîné sur des données particulières. De la même manière qu'un modèle de prédiction de la température pour la ville de Montréal est différent d'un modèle pour la ville de Québec, un modèle de langage créé par exemple à partir des données de 100 livres écrits au 19ième siècle, et un autre à partir de 100 livres écrits au 20ième siècles, seront deux modèles distincts, et auront des propriétés statistiques très différentes.

Tout comme le modèle de classification du travail noté #2, il est encore ici question de probabilité conditionnelle. Étant donné que nous allons construire un modèle bigramme, il s'agit donc de la probabilité d'un mot, étant donné le mot qui le précède (la barre verticale dans l'équation signifie "étant donné", ou "en présence de", ou "dans le contexte de") :

$$\text{Prob}(\text{mot à prédire} \mid \text{mot qui précède})$$

La tâche de notre modèle de classification pour les courriels était de discriminer (répondre oui ou non à la question : est-ce un pourriel?) tandis que la tâche de notre modèle bigramme sera ici de générer du nouveau texte, une fois le modèle entraîné, en faisant de l'échantillonnage. La génération de nouveau texte est ce qui permet à ChatGPT de répondre à une question, ou de composer un poème.

Entraînement du modèle

Voyons comment il est possible de calculer ces probabilités en entraînant un modèle bigramme génératif sur un texte très simple.

Nous allons encore une fois utiliser le tableur Google Sheets au lieu de Excel, car Google Sheets est plus accessible, et le langage de ses formules est plus facile à gérer (celui d'Excel dépend de la langue et des paramètres régionaux de votre système d'exploitation). Pour éviter la confusion dans le contexte de ce travail, nous devons tout d'abord nous assurer que la langue des fonctions et des paramètres régionaux est l'anglais :

Paramètres de cette feuille de calcul

Général

Calcul

Paramètres régionaux :

Canada (anglais)

Cette option affecte les détails de formatage tels que les fonctions, les dates et la devise.

Fuseau horaire

L'historique de votre feuille de calcul sera enregistré selon ce fuseau horaire. Cela affectera toutes les fonctions relatives à l'heure.

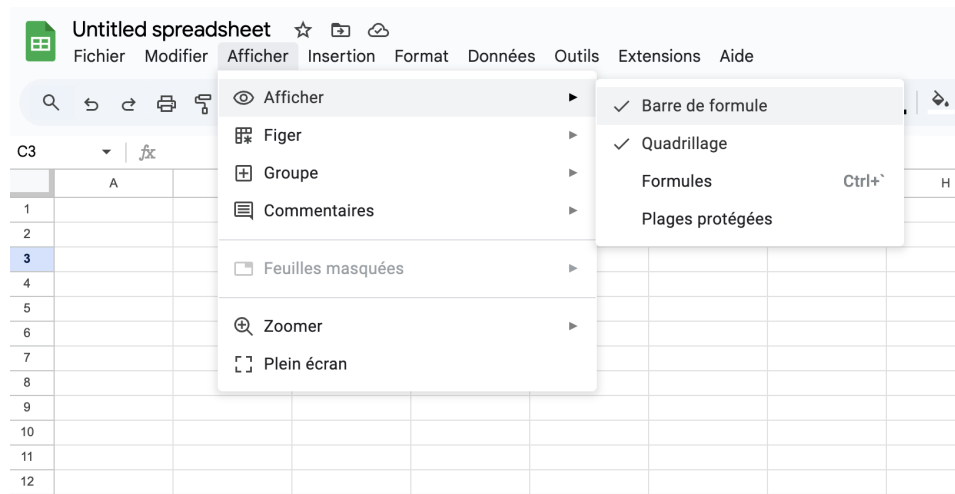
Langue d'affichage : Français (Canada)

☒ Toujours utiliser les noms de fonctions en anglais

Annuler

Enregistrer et recharger

Assurez-vous ensuite que la "barre de formules" soit visible :



Copiez tout d'abord les mots de ce texte dans la colonne A d'une nouvelle "feuille" Google Sheets, un mot par rangée :

le
chat
dort
le
chien
mange
le
chat
mange
une
souris
le
chien
dort
la
souris
court
la
souris
mange
le
fromage
le
chat
court
le
chien
voit
le
chat
le
chat
voit
la
souris
le
chien

court

Dans la première cellule de la colonne B (donc B1), entrez cette formule :

=A2

La colonne B devrait être étendue jusqu'à la cellule B37, en double-cliquant sur le petit "+" qui apparaît quand votre curseur est placé au-dessus du coin inférieur droit de la cellule B1 (il est possible que Google Sheets offre de le faire pour vous, automatiquement).

Dans la cellule C1, entrez maintenant cette formule :

=A1 & " " & B1

Encore une fois, la colonne C doit s'étendre jusqu'à la cellule C37). À ce stade, votre feuille devrait ressembler à ceci :

	A	B	C	D	E	F	G
1	le	chat	le chat				
2	chat	dort	chat dort				
3	dort	le	dort le				
4	le	chien	le chien				
5	chien	mange	chien mange				
6	mange	le	mange le				
7	le	chat	le chat				
8	chat	mange	chat mange				
9	mange	une	mange une				
10	une	souris	une souris				
11	souris	le	souris le				
12	le	chien	le chien				
13	chien	dort	chien dort				
14	dort	la	dort la				
15	la	souris	la souris				
16	souris	court	souris court				
17	court	la	court la				
18	la	souris	la souris				
19	souris	mange	souris mange				
20	mange	le	mange le				
21	le	fromage	le fromage				
22	fromage	le	fromage le				
23	le	chat	le chat				
24	chat	court	chat court				
25	court	le	court le				
26	le	chien	le chien				
27	chien	voit	chien voit				
28	voit	le	voit le				
29	le	chat	le chat				
30	chat	le	chat le				
31	le	chat	le chat				
32	chat	voit	chat voit				
33	voit	la	voit la				
34	la	souris	la souris				
35	souris	le	souris le				
36	le	chien	le chien				
37	chien	court	chien court				
38	court						
39							

À ce stade, il devrait être clair pour vous que la colonne C contient tous les bigrammes extraits du texte de la colonne A (la colonne B n'est qu'un mécanisme pour les obtenir facilement).

Nous allons maintenant compter, dans la colonne D, le nombre de fois où un bigramme particulier apparaît dans le texte de la colonne A :

`=COUNTIF(C:C, C1)`

On constate par exemple que le bigramme "le chien" apparaît 4 fois, tandis que le bigramme "fromage le", apparaît seulement une fois.

Dans la colonne E, nous allons maintenant calculer les paramètres de notre modèle, soit la probabilité d'un mot, étant donné le mot qui le précède :

$$\text{Prob}(\text{mot de la col B} \mid \text{mot de la col A}) = \frac{\#(\text{mots A et B})}{\#(\text{mot A})}$$

Pour ce faire, entrez dans la cellule E1 :

`=D1 / COUNTIF(A$1:INDEX(A:A, COUNTA(A:A)-1), A1)`

La probabilité d'un mot étant donné le mot qui le précède est donc simplement le nombre de fois où ce bigramme particulier apparaît dans le texte, divisée par le nombre de fois où le premier mot du bigramme apparaît. Étant donné que cette valeur est une probabilité, elle doit nécessairement être contenue entre 0 et 1.

Dans la colonne F nous allons filtrer la colonne C (tous les bigrammes) pour ne retenir que les bigrammes uniques :

`=SORT(UNIQUE(FILTER(C:C, C:C<>"")))`

Et dans la colonne G nous allons ajouter les probabilités correspondantes :

`=INDEX(E:E, MATCH(F1, C:C, 0))`

On peut maintenant constater que le mot "souris" suit nécessairement (avec une probabilité de 1) le mot "la", tandis que le mot "le" peut être suivi des mots "chat", "chien" et "fromage" avec des probabilités de 0.5, 0.4 et 0.1, respectivement.

À ce stade, votre feuille devrait ressembler à ceci :

	A	B	C	D	E	F	G	H	I	J
1	le	chat	le chat	5	0.5	chat court	0.2			
2	chat	dort	chat dort	1	0.2	chat dort	0.2			
3	dort	le	dort le	1	0.5	chat le	0.2			
4	le	chien	le chien	4	0.4	chat mange	0.2			
5	chien	mange	chien mange	1	0.25	chat voit	0.2			
6	mange	le	mange le	2	0.666666667	chien court	0.25			
7	le	chat	le chat	5	0.5	chien dort	0.25			
8	chat	mange	chat mange	1	0.2	chien mange	0.25			
9	mange	une	mange une	1	0.333333333	chien voit	0.25			
10	une	souris	une souris	1	1	court la	0.5			
11	souris	le	souris le	2	0.5	court le	0.5			
12	le	chien	le chien	4	0.4	dort la	0.5			
13	chien	dort	chien dort	1	0.25	dort le	0.5			
14	dort	la	dort la	1	0.5	fromage le	1			
15	la	souris	la souris	3	1	la souris	1			
16	souris	court	souris court	1	0.25	le chat	0.5			
17	court	la	court la	1	0.5	le chien	0.4			
18	la	souris	la souris	3	1	le fromage	0.1			
19	souris	mange	souris mange	1	0.25	mange le	0.666666667			
20	mange	le	mange le	2	0.666666667	mange une	0.333333333			
21	le	fromage	le fromage	1	0.1	souris court	0.25			
22	fromage	le	fromage le	1	1	souris le	0.5			
23	le	chat	le chat	5	0.5	souris mange	0.25			
24	chat	court	chat court	1	0.2	une souris	1			
25	court	le	court le	1	0.5	voit la	0.5			
26	le	chien	le chien	4	0.4	voit le	0.5			
27	chien	voit	chien voit	1	0.25					
28	voit	le	voit le	1	0.5					
29	le	chat	le chat	5	0.5					
30	chat	le	chat le	1	0.2					
31	le	chat	le chat	5	0.5					
32	chat	voit	chat voit	1	0.2					
33	voit	la	voit la	1	0.5					
34	la	souris	la souris	3	1					
35	souris	le	souris le	2	0.5					
36	le	chien	le chien	4	0.4					
37	chien	court	chien court	1	0.25					
38	court									
39										

Utilisation du modèle (inférence)

Colonne H :

"le"

Colonne I :

```
=LET(
    next_word_mask, ARRAYFORMULA($G:$G = J1),
    next_words, FILTER($H:$H, next_word_mask),
    probs, FILTER($I:$I, next_word_mask),
    probs_cumul, SCAN(0, probs, LAMBDA(a, b, a + b)),
    sampled_word_idx, MATCH(RAND(), {0; probs_cumul}, 1),
    INDEX(next_words, sampled_word_idx)
)
```