2.2 Matrice terme-terme

2.2.1 Transition : inversion des matrices

Si on revient à notre matrice terme-document déjà vue plus haut (rappelée ici, table 2.2), on peut faire une observation très simple : au lieu de représenter un document comme un vecteur dans un espace dont les dimensions sont des mots, on peut représenter les mots comme des vecteurs dans un espace dont les dimensions sont les documents.

	QuatreVT 119 Kw	Voyage Bal 82 kw	Bête Hum. 128 kw	Mme Bovary 117 kw
bataille	35	4	6	2
clair	105	26	96	52
facile	12	19	6	10
politique	11	0	9	5
voyage	17	196	94	44
idiot	2	1	2	6
amour	19	0	47	94

TABLE 2.2 – Matrice terme-documents pour quelques mots et 4 romans (Quatrevingt-treize (Hugo) ; Le voyage en ballon (Verne) ; La bête humaine (Zola) ; Mme Bovary (Flaubert)).

Ainsi, par exemple, le mot *amour* se caractérise par une coordonnée nulle sur la dimension "Voyage en ballon" (comme le mot *politique* d'ailleurs). Cette représentation des mots peut aussi faire l'objet d'une visualisation dans l'espace 2D, comme à la figure 2.5 [2].

bataille	(35,2)
politique	(11,5)
amour	(19,94)
voyage	(17,44)

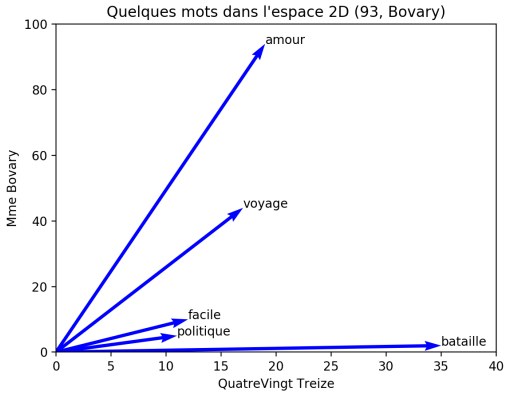


FIGURE 2.5 – représentation graphique des mots dans le plan (93, Bovary)

Mais cette représentation peut être vue comme une première version de représentation distributionnelle : chaque dimension (= document) est un "contexte" : les coordonnées d'un mot m représentent le nombre de fois que m se trouve dans ce contexte. Deux mots qui ont des coordonnées comparables (éventuellement à un facteur multiplicatif près : colinéarité) sont des mots qui ont le même comportement dans un contexte donné.

2. Ne pas oublier que cette figure représente les mots dans un espace à seulement deux dimensions (toutes les autres dimensions étant ignorées) et que par conséquent cela ne donne qu'une vision très partielle de la représentation réelle des mots induite par cette inversion de la matrice.

L'étape qui manque pour arriver au principe fondamental de l'hypothèse distributionnelle va consister à prendre des dimensions différentes : au lieu de choisir des documents, on va tout simplement choisir des voisinages de mots.

2.2.2 Mots (voisinages) comme dimensions

2.2.2.1 Principe

Les dimensions dans ce qui précède sont des documents pris comme *bags of words* ; on va maintenant, en s'inspirant des concordanciers qui fournissent des représentations KWIC, prendre comme dimension des suites de tokens (ou mots/mwe/morphèmes...) centrées autour d'un mot donné (en pratique, on ne découpe pas le textes en n-grammes, mais on recueille tous les mots qui apparaissent dans une fenêtre de dimension fixe autour d'un token donné). Voir ci-dessous l'exemple des contextes de 3 mots avant, 3 mots après, autour du mot *voyagiste*, tels qu'ils sont donnés par frWaC.

je connaissais le	voyagiste	qui les amène
voyages : Différent du	voyagiste	ou tour-opérateur . En
compagnie à un	voyagiste	. Animaux domestiques : Lorsqu'
que disent certains	voyagistes	, légalement , aucune n'
Transport / Hébergement Le	voyagiste	FRAM propose les
plus vite votre	voyagiste	. Le Comité scientifique
le monde ... 6ème	voyagiste	européen , Kuoni est
autres , les grands	voyagistes	essayent de prendre
fév 2008 Ces	voyagistes	, qui opèrent principalement
parmis les meilleurs	voyagistes	en ligne . Service
les 15 principaux	voyagistes	en ligne en
ses amies des	voyagistes	locaux qui fêtent
foyers ...) et des	voyagistes	. La commercialisation Le
lancé vers des	voyagistes	susceptibles d' offrir
clients chez un	voyagiste	- Développement d' un
des plus grands	voyagistes	européens dont ils
voyage à cheval .	Voyagiste	spécialiste du voyage
clients chez un	voyagiste	- Développement d' un
sur cette déstination	Voyagiste	Catalogue déstination du
hôtel , à votre	voyagiste	, à votre ambassade

frWaC, consultation 2021-08-11

Si on prend la liste complète de tous ces mots, on obtient quelque chose de très similaire à un document pris comme *bag of words*. Et par conséquent on peut appliquer la méthode précédente et prendre ces contextes comme dimension.

On peut donc construire des matrices d'un type différent, que nous appelons terme-terme, où les objets définis sont des mots (les colonnes), et les dimensions (les lignes) sont aussi des mots — mais dans le sens précédent : il s'agit plutôt du regroupement de tous les contextes de taille fixe autour du mot.

Voici par exemple des comptages effectués dans le corpus frWaC (table 2.3). On a compté le nombre de fois que le mot *bataille* apparaissait dans un voisinage immédiat (c'est-à-dire dans les 5 mots avant ou les 5 mots après) du mot *arriver* (246 $\times$), etc.

On peut bien sûr multiplier les lignes (ie les dimensions) ou les colonnes (le vocabulaire),

	bataille	voyage	homme	femme
arriver	246	470	1 819	890
tomber	100	83	1 205	384
habiller	2	4	339	384
mourir	180	116	339	1 088
	55 331	208 520	668 289	346 093

TABLE 2.3 – Matrice Terme-Terme obtenue dans frWaC. Contextes en ligne.

au point de faire converger les listes : pour un vocabulaire V, on aurait un vecteur de dimension $|V|$ distinct pour chaque mot de V.

Un peu d'algorithmique Dans ce qui précède, les lignes et les colonnes du tableau sont présentées de façon distincte : les colonnes sont des vecteurs qui représentent les mots dans un certain espace ; les lignes sont des dimensions, *i.e.* des contextes (voire des pseudo-documents/*bag of words* constitués par tous les mots qui apparaissent dans le voisinage du mot correspondant à la dimension). Mais on peut observer que si l'on compte 231 occurrences du mot *arriver* dans une fenêtre de 5 mots autour de *bataille*, et bien on peut s'attendre à compter 231 occurrences du mot *bataille* dans une fenêtre de 5 mots autour de *arriver*[3]. Autrement dit, les matrices terme-terme sont inversibles, ou, dit autrement, les rôles des lignes et des colonnes sont interchangeables. On peut donc préférer présenter la matrice de la table 2.3 sous la forme inversée de la table 2.4.

	arriver	tomber	habiller	mourir	
bataille	246	100	2	180	55 331
voyage	470	83	4	116	208 520
homme	1 819	1 205	339	1 499	668 289
femme	890	660	384	1 088	346 093

TABLE 2.4 – Matrice Terme-Terme obtenue dans frWaC. Contextes en colonnes.

2.2.2.2 Les coordonnées

Dans ce qui précède, une fois qu'on a choisi les mots comme dimensions, on se trouve à donner comme coordonnées des fréquences (pour la dimension *bataille*, la coordonnée du mot *chien* est le nombre de fois que *chien* apparaît dans le contexte (fenêtre de k mots avant/après) du mot *bataille*).

Mais on peut essayer de réfléchir à d'autres approches. Une partie de la réflexion engagée plus haut sur les coordonnées s'applique (normalisations...) mais en revanche on a perdu ici la notion de "document" : les mesures de type tf-idf ne peuvent pas s'appliquer. En revanche, il existe une possibilité qui mérite qu'on y consacre un peu de temps : la possibilité d'utiliser comme coordonnées les PPMI.

3. Petite expérience sur frWaC : en faisant une requête sur le lemme *bataille* filtré par les occurrences du lemme *arriver*, on trouve 40 152 occurrences puis 231 ; en faisant une requête sur le lemme *arriver*, filtré par les occurrences du lemme *bataille*, on trouve 336 786 occurrence puis 231. Ça ne prouve rien, bien sûr.

Positive Pointwise Mutual Information La mesure d'information mutuelle (entre mots ou entre évènements) tente de rendre compte du fait qu'un évènement peut être particulièrement informatif à propos d'un autre évènement.

La question est : est-ce que deux évènements x et y apparaissent plus souvent que s'ils étaient indépendants ? On sait que si deux évènements sont indépendants, leur probabilité conjointe est égale au produit de leurs probabilités : $P(A \cap B) = P(A) \times P(B)$. Par conséquent, le ratio entre ces deux probabilités sera supérieur à 1 si les évènements ne sont pas indépendants, autrement dit si la survenue d'un des évènements augmente la chance de survenir de l'autre évènement.

Pour manipuler plus facilement cette mesure, on passe au logarithme :

$$pmi(x, y) = \log_2 \frac{P(x \cap y)}{P(x)P(y)}$$

Le passage au logarithme permet de distribuer les valeurs sur $]-\infty, +\infty[$, et d'avoir des valeurs positives si le ratio est supérieur à 1 (puisque $\log_2 1 = 0$), et négatives sinon.

En ce qui concerne les mots, l'évènement "conjoint" est la co-occurrence de deux mots dans un contexte donné (typiquement une fenêtre de 10 mots). Quand on dispose d'un corpus, on peut estimer les probabilités en faisant un simple comptage des fréquences, divisé par le nombre total de tokens.

On pourrait donc décider de créer une matrice terme-terme où les valeurs dans chaque cellule seraient la *pmi*. Il y a cependant un problème avec les mesures négatives de la *pmi* : comment interpréter une valeur négative ? Avoir une valeur négative signifie que la co-occurrence des deux mots donnés est moins fréquente que le hasard. Il est très difficile d'interpréter une telle information, qui de plus est particulièrement peu fiable à moins de disposer d'un corpus vraiment immense [4]. Il faut noter aussi qu'alors que les valeurs positives suggèrent une proximité, une relation entre les deux mots, qui peut correspondre à l'intuition des locuteurs, les valeurs négatives correspondraient à une notion de non-relation, qui ne correspond pas à une intuition linguistique.

C'est la raison pour laquelle on adopte la mesure *ppmi* (*positive pmi*), qui consiste à mettre à zéro les valeurs négatives :

$$ppmi(x, y) = \max(\log_2 \frac{P(x \cap y)}{P(x)P(y)}, 0)$$

Exemple détaillé (repris de Jurafsky et Martin (2019)) On calcule une matrice avec m lignes (mots — à représenter), et c colonnes (contextes). La matrice peut-être carrée avec exactement les mêmes éléments en ligne et en colonne, mais ce n'est pas nécessaire, comme on le verra dans les exemples suivants.

Soit f_{ij} la fréquence d'apparition du mot w_i dans le contexte c_j. Voici la matrice f que nous allons prendre comme exemple (dans la suite la colonne aardvark va disparaître car elle ne contient que des zéros) :

4. Si deux mots ont une probabilité de 10^{-6}, il est difficile de décider si la probabilité de leur coocurrence est significativement différente de 10^{-12}.

f	aardvark	computer	data	pinch	result	sugar
apricot	0	0	0	1	0	1
pineapple	0	0	0	1	0	1
digital	0	2	1	0	1	0
information	0	1	6	0	4	0

Alors on peut définir la probabilité conjointe p_{ab} comme le ratio entre la fréquence des éléments conjoints et le nombre total d'occurrences de tous les mots dans tous les contextes :

$$p_{ab} = \frac{f_{ab}}{\Sigma_i^W \Sigma_j^C f_{ij}}$$

La probabilité p_{a*} est la probabilité d'occurrence du mot w_a, on peut la calculer en déterminant le nombre total d'occurrences du mot divisé par le nombre total d'occurrences de tous les mots, mais on peut la calculer aussi en utilisant la fréquence f_{ij} :

$$p_{a*} = \frac{\Sigma_j^C f_{aj}}{\Sigma_i^W \Sigma_j^C f_{ij}}$$

De façon analogue, la probabilité d'avoir le contexte w_b peut être obtenue ainsi :

$$p_{*b} = \frac{\Sigma_i^W f_{ib}}{\Sigma_i^W \Sigma_j^C f_{ij}}$$

Si on reprend notre exemple précédent, on peut ajouter les sommes d'occurrences, ce qui donne la table :

f	computer	data	pinch	result	sugar	Σ
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
Σ	3	7	2	5	2	19

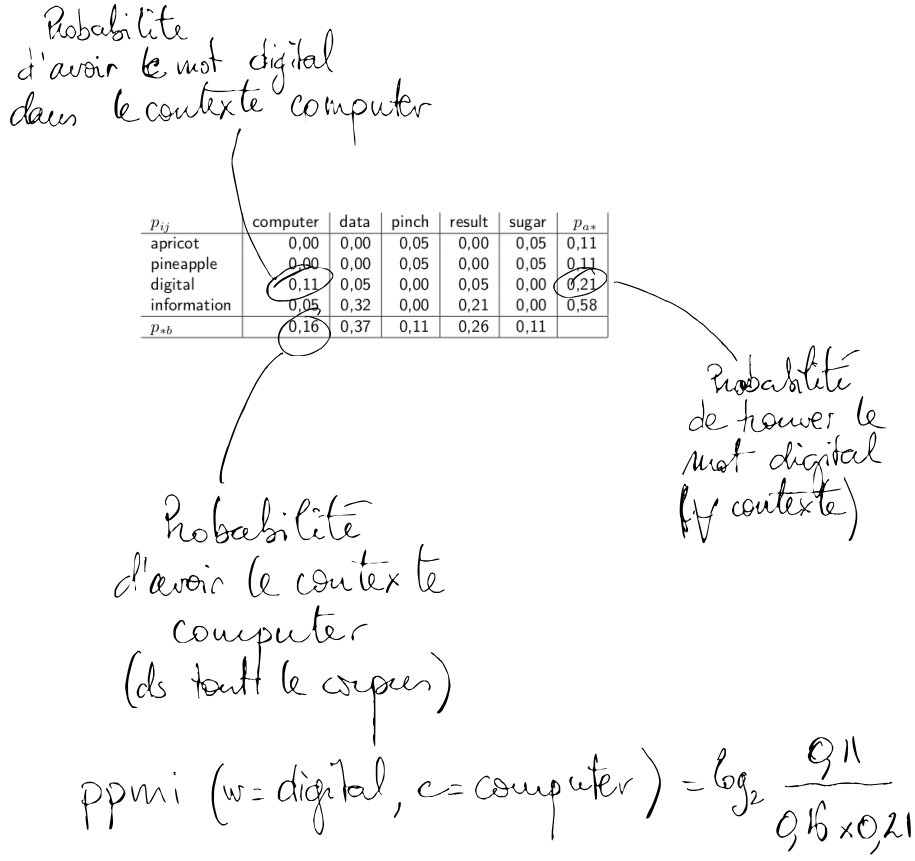
... à partir de laquelle on peut calculer les différentes probabilités :

p_{ij}	computer	data	pinch	result	sugar	p_{a*}
apricot	0/19	0/19	1/19	0/19	1/19	2/19
pineapple	0/19	0/19	1/19	0/19	1/19	2/19
digital	2/19	1/19	0/19	1/19	0/19	4/19
information	1/19	6/19	0/19	4/19	0/19	11/19
p_{*b}	3/19	7/19	2/19	5/19	2/19	19/19

... ce qui donne en valeurs décimales :

p_{ij}	computer	data	pinch	result	sugar	p_{a*}
apricot	0,00	0,00	0,05	0,00	0,05	0,11
pineapple	0,00	0,00	0,05	0,00	0,05	0,11
digital	0,11	0,05	0,00	0,05	0,00	0,21
information	0,05	0,32	0,00	0,21	0,00	0,58
p_{*b}	0,16	0,37	0,11	0,26	0,11	

Le même tableau accompagné de quelques commentaires pour mieux comprendre le calcul :



p_{ij}	computer	data	pinch	result	sugar	p_{a*}
apricot	0,00	0,00	0,05	0,00	0,05	0,11
pineapple	0,00	0,00	0,05	0,00	0,05	0,11
digital	0,11	0,05	0,00	0,05	0,00	0,21
information	0,05	0,32	0,00	0,21	0,00	0,58
p_{*b}	0,16	0,37	0,11	0,26	0,11	

On peut finalement aboutir au tableau des *ppmi* (les zéros correspondent à des cas où le ratio des probabilités était inférieur à 1, les tirets aux cas où le nombre d'occurrences étant nul, la ppmi n'a pas de pertinence) :

ppmi	computer	data	pinch	result	sugar
apricot	-	-	2,25	-	2,25
pineapple	-	-	2,25	-	2,25
digital	1,66	0,00	-	0,00	-
information	0,00	0,57	-	0,47	-

Quelques points à relever suite à cet exemple détaillé :

— La technique ne repose que sur la collecte de la matrice de co-occurrences, et même si conceptuellement on donne un statut différent aux colonnes (contextes) et aux lignes (mots à représenter), on a une symétrie complète au niveau du calcul, et on peut donc inverser la matrice (en pratique on n'a pas forcément besoin de le faire car les vocabulaires sont souvent exactement les mêmes).

— [Pour mémoire] Notre calcul s'est fait avec 19 occurrences en tout. Ne pas s'étonner si des valeurs semblent étranges.

Discussion : lissage La PMI est une mesure intéressante, et qui donne en général des valeurs très pertinentes, à ceci près qu'elle a quand-même un biais vers les évènements rares : un mot très peu fréquent va donner (du moins pour les mots dans le contexte desquels il apparaît) des valeur de pmi très élevées. Il y a deux solutions fréquemment mises en œuvre pour tenter de corriger ce biais (parmi d'autres), et ces solutions sont intéressantes, car

elles font partie de la grande famille des méthodes de lissage (en. *smoothing*) [5] :

— Donner une probablité plus élevée aux évènements les plus rares ;
— Utiliser un lissage laplacien (add-one), ce qui a un effet similaire.

Augmentation des probabilités La méthode consiste à modifier la probablité des contextes, en utilisant une puissance α, avec $\alpha = 0,75$.

$$ppmi_\alpha(x, y) = \max(\log_2 \frac{P(x \cap y)}{P(x)P_\alpha(y)}, 0)$$

La probabilité $P_\alpha(y)$ est obtenue en élevant la fréquence de tous les évènements à la même puissance α. Cette opération est utile parce que si un évènement c est rare, on aura $P_\alpha(c) > P(c)$. Si au contraire un évènement est fréquent, on aura une probabilité inférieure.

$$P_\alpha(x) = \frac{f(x)^\alpha}{\Sigma f(i)^\alpha}$$

Par exemple, soient les évènements a t.q. $P(a) = 0,99$ et b t.q. $P(b) = 0,01$. Alors on obtient :

$$P_\alpha(a) = \frac{0,99^{0,75}}{0,99^{0,75} + 0,01^{0,75}} = 0,97 \quad P_\alpha(b) = \frac{0,01^{0,75}}{0,99^{0,75} + 0,01^{0,75}} = 0,03$$

Lissage laplacien Le lissage de Laplace consiste à ajouter 1 au nombre d'occurrences de tous les mots (cet ensemble est fini), avant d'estimer les probabilités comme nous l'avons fait auparavant. On fait donc l'hypothèse, avant même la consultation du corpus, que tous les mots ont au moins une occurrence (on peut utiliser la valeur 2, aussi). Les circonstances où ce genre de lissage devient inefficace sont lorsqu'il y a trop d'évènements jamais rencontrés : dans ce cas, presque tous les mots se retrouvent avec la même probabilité ($1/n$, avec n taille des données). Il faut alors utiliser d'autres techniques comme le fameux lissage de Good-Turing.

Jurafsky et Martin (2019) ont repris l'exemple déroulé précédemment et ont calculé l'impact d'un lissage laplacien (de 2). Cela donne tout d'abord des nouveaux décomptes :

f	computer	data	pinch	result	sugar
apricot	2	2	3	2	3
pineapple	2	2	3	2	3
digital	4	3	2	3	2
information	3	8	2	6	2

... ce qui aboutit aux nouvelles probabilités :

p_{ij}	computer	data	pinch	result	sugar	p_{a*}
apricot	0,03	0,03	0,05	0,03	0,05	0,20
pineapple	0,03	0,03	0,05	0,03	0,05	0,20
digital	0,07	0,05	0,03	0,05	0,03	0,24
information	0,05	0,14	0,03	0,10	0,03	0,36
p_{*b}	0,19	0,25	0,17	0,22	0,17	

5. Le lissage consiste à redistribuer une partie de la masse des probabilités sur des évènements dont la probabilité est sous-estimée (voire nulle). C'est une technique utile en apprentissage machine, car il arrive fréquemment qu'on rencontre au test (ou dans l'utilisation) d'un système des évènements jamais rencontrés à l'entraînement.

... et on peut alors comparer le tableau des ppmi calculé initialement avec celui qu'on obtient (toutes les étapes du calcul ne sont pas détaillées) avec un lissage laplacien +2, voir figure 2.6. On peut noter que les valeurs de ppmi sont « lissées », avec des taux d'information mutuelle qui semblent, au premier regard, bien plus comparables. Il faut garder en mémoire cependant que l'exemple que nous avons déroulé est un exemple « jouet ».

ppmi	computer	data	pinch	result	sugar
apricot	-	-	2,25	-	2,25
pineapple	-	-	2,25	-	2,25
digital	1,66	0,00	-	0,00	-
information	0,00	0,57	-	0,47	-
ppmi [add2]	computer	data	pinch	result	sugar
apricot	0,00	0,00	0,56	0,00	0,56
pineapple	0,00	0,00	0,56	0,00	0,56
digital	0,62	0,00	0,00	0,00	0,00
information	0,00	0,58	0,00	0,37	0,00

FIGURE 2.6 – Comparaison de mesures de *ppmi* sur l'exemple de (Jurafsky et Martin, 2019), avec ou sans lissage laplacien +2