# Identification de séquences “pertinentes” de mots

- Tentative de caractérisation des collocations
- Représentation d’un texte en table de suffixes
- Mesures de la pertinence de séquences monolingues de mots
- Application à la détermination de lexiques bilingues
- Affinités lexicales et application à la classification de textes
- Découvrir des traductions dans des corpus non parallèles
- Trouver des co-occurrences similaires dans un texte

# Introduction

## Que veut-dire qu'un n-gram est pertinent ?

Pour tenter d'y répondre, notez les séquences de mots suivantes de 1 à 5 avec la convention: **1** = très pertinente, **2** = pertinente, **3** = bof, **4** = plutôt pas pertinente, **5** = vraiment pas pertinente...

| | |
|---|---|
| intérêts du canada | intérêts des sociétés multinationales |
| fin du printemps | enfreint le règlement |
| les lignes directrices sur les conflits d' intérêts | n' a pas |
| le projet de loi | : monsieur le président , je |
| le gouvernement ne | et des |

# Collocations

Chaque auteur utilise sa propre définion du terme. On en trouve dans Smadja [93] une caractérisation assez complète.

**Propriété 1:** Les collocations sont arbitraires. Exemple (pris dans Smadja [93]) de la collocation *to break down/force the door*:

| langue | expression | équivalent anglais |
|---|---|---|
| français | enfoncer la porte | to push the door through |
| allemand | die Tür aufbrechen | to break the door |
| italien | sfondare la porta | to hit/demolish the door |
| espagnol | tumbar la puerta | to fall the door |
| turque | kapiyi kirmak | to break the door |

(en revanche dans ces 5 langues, l'expression **to see the door** se "traduit" de manière compositionnelle.)

# Collocations

**Propriété 2:** Les collocations sont dépendantes du domaine (terme)

↪ Ex. (connu du plongeur québécois, mais pas nécesssairement du plongeur marseillais): **dry suit**

**Propriété 3:** Les collocations sont récurrentes

↪ Par récurrente, on veut dire, que l'on retrouve normalement plusieurs fois dans un passage une collocation donnée.

**Propriété 4:** Les collocations sont des unités cohérentes

↪ La donnée d'un mot (ou de plusieurs) d'une collocation permet de reconstituer la collocation (c'est sur des études perceptives de ce genre que se basent les lexicographes pour décider de la nature collocationnelle ou non d'une séquence de mots).

# Détente: remplacez ?? par le bon mot

| |
|---|
| If a fire breaks out, the alarm will ?? |
| The boy doesn't know how to ?? his bicycle |
| The American congress can ?? a presidential veto |
| Before eating your bag of microwavable popcorn, you have to ?? it |

Exemple pris dans Smadja [93], lui même rapporté de Benson.

# Le bon mot

**Si vous n'avez pas réussi,** voici la réponse[1]

- If a fire breaks out, the alarm will ??
  ↪ **ring** / go off / **sound** / start
- The boy doesn't know how to ?? his bicycle
  ↪ drive / **ride** / conduct
- The American congress can ?? a presidential veto
  ↪ ban / cancel / delete / reject / turn down / abrogate / **overrule**
- Before eating your bag of microwavable popcorn, you have to ?? it
  ↪ **cook** / **nuke** / broil / fry / bake

[1] Un grand merci à Graham Russell

# Différentes formes de collocations

## Collocations prédicatives

| type | exemples |
|---|---|
| N-Adj | heavy/light [] trading/smoker/traffic |
| N-Adj | strong [] tea |
| S-V | stock [] rose, fell, jumped ... |
| V-Part | take [] from – raise [] by – mix [] with |

## Collocations rigides

- Souvent des groupes nominaux:
  *traffic jam*, *foreign exchange*, *New York Stock Exchange*, *Stock Market*, *White House Spokesman Marlin Fitwater*.
- L'idée est que ces groupes ne sont pas séparables sans perte de sens.

# Différentes formes de collocations

## Patrons idiomatiques

- Ex. de formes idiomatiques:
  *Je te le donne en mille, ni une ni deux, de proche en proche, broche à foin, à brûle-pourpoint*
- Ex. d'un patron idiomatique selon Smadja [93]:
  *The closely watched index had been down about NUMBER points in the first hour of trading*

**Résumons:** une collocation peut prendre de multiples formes, faire intervenir un nombre quelconque de mots. Il semble difficile de les énumérer toutes (dépendantes du domaine), mais leur connaissance est néanmoins cruciale pour la maîtrise d'une langue.

**Question:** est-il possible de les identifier automatiquement dans un texte?

# Tableaux de suffixes (*Suffix arrays*)

## Accès rapide à n'importe quelle séquence de mots

- Une représentation qui a été largement exploitée aussi bien dans le cadre monolingue que dans le cadre bilingue pour découvrir des **collocations**. Voir par exemple Nagao and Mori [1994], Ikehara et al. [1996], Haruno et al. [1996], Shimohata et al. [1997], Russell [1998].
- **Idée:** indexer **toutes** les séquences possibles rencontrées dans un texte.
- **But:** pouvoir calculer des statistiques sur ces séquences pour découvrir par exemple des séquences pertinentes (au sens de la statistique utilisée).
- **Comment:** par création d'un tableau $LPC$ des plus longs préfixes communs.
- **Note:** Les transparents qui suivent reprennent les algorithmes décrits dans Russell [1998]. Pour une implantation efficace des tableaux de suffixes lire les articles Gonnet et al. [1992], Manber and Myers [93].

# Some of the words of the sentence are the same

- Ce texte contient 10 mots (positions 0 à 9) et 7 types (indexes 0 à 6).
- Soit un ordre à priori sur les types d'un texte; ici l'ordre lexicographique ($Some$ = 0, $words$=6):

  $Some < are < of < same < sentence < the < words$.

- On peut coder un suffix array $SFX$ — où $SFX[i]$ désigne une position dans le texte à indexer — en regroupant les suffixes dans le texte qui commencent par un mot donné. Pour cela on passe en revue tous les types un par un en ordre:

| 0 | 7 | 4 | 1 | 9 | 6 | 8 | 5 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|---|

  $SFX$ représente les suffixes ordonnés d'un texte (d'où le nom de la méthode d'indexation).

# Exemple (suite)

## Calcul de la table des plus long prefixes $LPC$

- $LPC$ stocke la taille du plus grand préfixe commun que deux suffixes partagent (ex: *of the words* et *of the sentence* ont un préfixe commun de deux mots). Pour cela, on fait un passage sur $SFX$ en comparant les suffixes deux à deux.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | 0 | 0 | 0 | 1 | 1 | 0 |

- $LPC[i]$ indique le nombre de mots communs que partagent les deux séquences de mots qui commencent respectivement aux positions $SFX[i]$ et $SFX[i+1]$.

# Retrouver une séquence dans un texte

**Input:** une occurrence d'une séquence $key$ a été trouvée en position $found$ dans $SFX$.
**Output:** compte le nombre d'occurrences de cette séquence, $left$ indique la première position d'occurrence dans $SFX$.

**Algorithme:**

$left \leftarrow right \leftarrow found$
**while** $left > 0 \ \wedge \ lcps[left - 1] \geq |key|$ **do**
  $left \leftarrow left - 1$
**while** $right < |lcps| \ \wedge lcps[right] \geq |key|$ **do**
  $right \leftarrow right + 1$
return$(right - left + 1)$

**Note:** On n'a pas besoin de la table $SFX$.

# Extraire les séquences

**But:** On souhaite extraire toutes les séquences d'au moins $min\_length$ mots et de fréquence minimale $min\_freq$ depuis la table $LPC$.

**Structure:** une séquence $s$ est caractérisée par un triplet $\langle l, f, p \rangle$, où $l$ est la longueur d'une séquence, $f$ sa fréquence, et $p$ la première position dans $SFX$ qui pointe (dans le texte) sur une séquence dont le préfixe est $s$.

**Exemple:** la séquence *of the* dans notre exemple est caractérisée par le triplet $\langle 2, 2, 2 \rangle$. La séquence *the same* est caractérisée par le triplet $\langle 2, 1, 6 \rangle$

**Idée:** en maintenant deux ensembles de triplets $active$ et $result$. Le premier étant un ensemble temporaire, qui contient les séquences potentiellement intéressantes en cours d'analyse; le second contenant la réponse.

# Extraction des séquences

1: $results \leftarrow active \leftarrow \emptyset$
2: $new\_length \leftarrow prev\_length \leftarrow min\_length$
3: $this\_pos \leftarrow 0$
4: **while** $this\_pos < |lcps|$ **do**
5: $\quad this\_length \leftarrow lcps[this\_pos]$
6: $\quad$ **if** $this\_length < min\_length$ **then**
7: $\quad\quad$ **for all** triples $t = \langle len, freq, pos \rangle$ in $active$ **do**
8: $\quad\quad\quad active \leftarrow active - \{t\}$
9: $\quad\quad\quad$ **if** $freq \geq min\_freq$ **then**
10: $\quad\quad\quad\quad results \leftarrow results \cup \{t\}$
11: $\quad\quad new\_length \leftarrow min\_length$
12: $\quad$ **else if** $this\_length \geq prev\_length$ **then**
13: $\quad\quad$ **for all** triples $\langle len, freq, pos \rangle$ in $active$ **do**
14: $\quad\quad\quad freq \leftarrow freq + 1$
15: $\quad\quad$ **while** $new\_length \leq this\_length$ **do**
16: $\quad\quad\quad active \leftarrow active \cup \{\langle new\_length, 2, this\_pos \rangle\}$
17: $\quad\quad\quad new\_length \leftarrow new\_length + 1$

18: $new\_length \leftarrow this\_length + 1$
19: **else**
20: **for all** triples $t = \langle len, freq, pos \rangle$ in $active$ **do**
21: **if** $len \leq this\_length$ **then**
22: $freq \leftarrow freq + 1$
23: **else**
24: $active \leftarrow active - \{t\}$
25: **if** $freq \geq min\_freq$ **then**
26: $results \leftarrow results \cup \{t\}$
27: $new\_length \leftarrow this\_length + 1$
28: $prev\_length \leftarrow this\_length$
29: $this\_pos \leftarrow this\_pos + 1$
30: **for all** triples $t = \langle len, freq, pos \rangle$ in $active$ **do**
31: **if** $freq \geq min\_freq$ **then**
32: $results \leftarrow results \cup \{t\}$
33: return$(results)$

# Extraction des séquences

Trois cas de figure (exclusifs) dans cet algorithme:

$this\_length < min\_length$ (6–11)

On vide $active$ en gardant (dans $result$) les séquences qui vérifient le critère de fréquence.

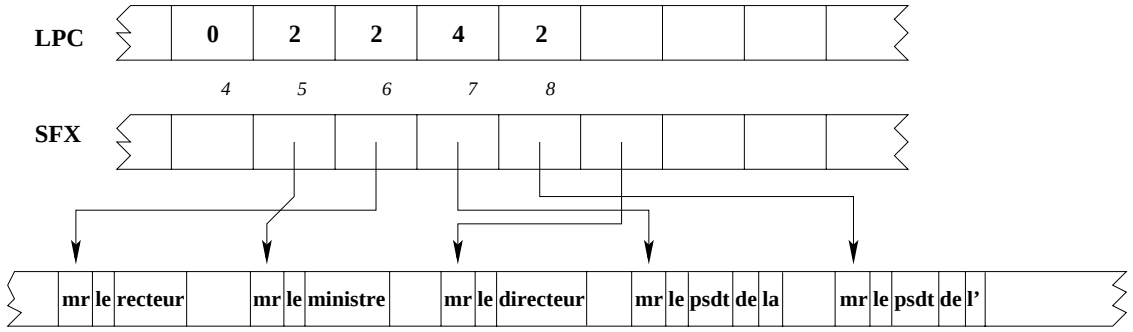
$this\_length \geq prev\_length$ (12–18)

- On augmente de 1 toutes les séquences déjà dans $active$.
- idée de la boucle sur $new\_length$: "lorsqu'on voir une séquence de taille $n$, on voit également une séquence de taille $n - 1$"

$this\_length < prev\_length$ **et** $this\_length \geq min\_length$ (20–27)

- On augmente de 1 toutes les séquences d'au plus $this\_length$ mots dans $active$.
- On retire de $active$ les autres séquences en gardant dans $result$ celles qui vérifient le critère de fréquence

# Exemple




Évolution de l'ensemble $active$ lorsque $min\_length = 2$:

| $this\_pos$ | $this\_length$ | $new\_length$ | active | |
|---|---|---|---|---|
| 4 | 0 | 2 | $\{\}$ | cas 1 |
| 5 | 2 | 3 | $\{< 2, 2, 5 >\}$ | cas 2 |
| 6 | 2 | 3 | $\{< 2, 3, 5 >\}$ | cas 2 |
| 7 | 4 | 5 | $\{< 2, 4, 5 >, < 3, 2, 7 >, < 4, 2, 7 >\}$ | cas 2 |
| 8 | 2 | 3 | $\{< 2, 5, 5 >\}$ | cas 3 |

# Signification d'un triplet $\langle l, f, p \rangle$

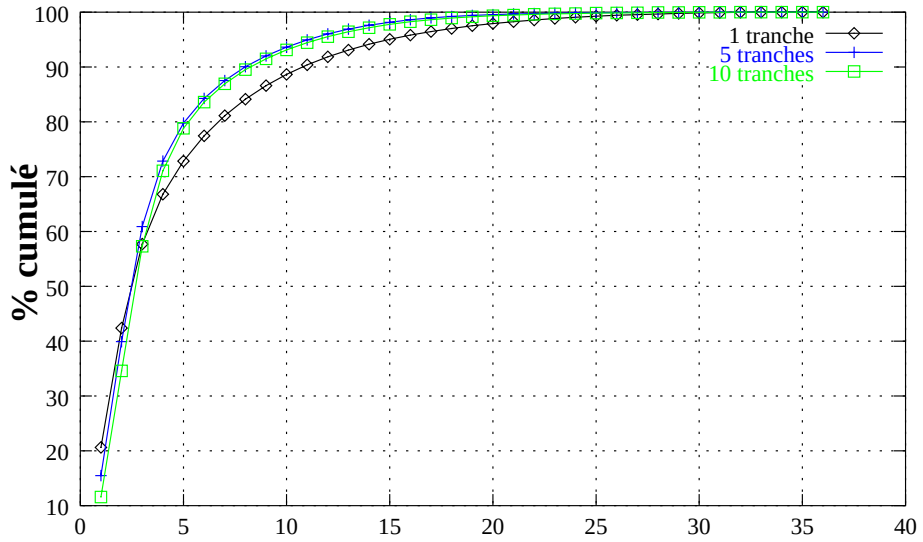
Représente un ensemble de séquences identiques. La cardinalité de cet ensemble est la fréquence de la séquence. Si $T$ désigne le texte d'origine, alors:

$$\{\langle T[SFX[j]], \ldots T[SFX[j]+l-1]\rangle : p \leq j < p+f\}$$

**En pratique**, on est intéressé aux séquences qui ne contiennent pas de marqueur de fin de phrase; cad qu'on recherche habituellement les seules séquences contenues à l'intérieur des phrases d'un texte.

# Distribution des séquences (longueur $\geq$ 1)

## vues au moins deux fois




| | | | | |
|---|---|---|---|---|
| 1 | tranche du Hansard | = | 4629 types, | 22457 séquences |
| 5 | tranches du hansard | = | 10172 types, | 65715 séquences |
| 10 | tranches du hansard | = | 14832 types, | 128253 séquences |

# Comment sélectionner les bonnes ?

**Une première idée (simpliste):** la fréquence

| long. | fréq. | séquence | long. | fréq. | séquence |
|---|---|---|---|---|---|
| 2 | 1760 | de la | 2 | 893 | ) : |
| 2 | 1594 | de l' | 2 | 867 | qu' il |
| 2 | 1391 | le président | 2 | 778 | , le |
| 2 | 1083 | monsieur le | 2 | 724 | : monsieur |
| 3 | 1076 | monsieur le président | 3 | 724 | : monsieur le |
| 2 | 1069 | le gouvernement | 4 | 723 | : monsieur le président |
| 2 | 1040 | à la | 5 | 720 | : monsieur le président , |
| 2 | 988 | c' est | 2 | 701 | à l' |
| 2 | 985 | président , | 2 | 643 | le député |
| 3 | 983 | le président , | 2 | 640 | , je |
| 4 | 971 | monsieur le président , | 3 | 608 | ) : monsieur |
| 2 | 913 | que le | 5 | 608 | ) : monsieur le président |

Visiblement pas la bonne solution, du moins sans un bon filtre

# Comment sélectionner les bonnes séquences ?

Une multitude de statistiques possibles (voir Manning and Schütze [1999], chapitre 5 pour un vaste panorama). Une métrique souvent utilisée Church and Hanks [1989]: **l'information mutuelle** (plus précisément, *pointwise mutual information*).

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} = \log_2 \frac{P(x|y)}{P(x)} = \log_2 \frac{P(y|x)}{P(y)}$$

Ex.[2]:

| | chambre | ¬ chambre |
|---|---|---|
| house | 31950 | 12004 |
| ¬ house | 4793 | 848330 |

| | communes | ¬ communes |
|---|---|---|
| house | 4974 | 38980 |
| ¬ house | 441 | 852682 |

$$\log \frac{P(house|chambre)}{p(house)} = \log \frac{\frac{31950}{31950+4793}}{P(house)} \approx \log \frac{0.87}{P(house)}$$

$$\log \frac{P(house|communes)}{p(house)} = \log \frac{\frac{4974}{4974+441}}{P(house)} \approx \log \frac{0.92}{P(house)}$$

[2] Manning and Schütze [1999], page 179.

# Comment sélectionner les bonnes séquences ?

Le test de vraisemblance (*likelihood ratio*) Dunning [1993]

$$\begin{aligned} -\log\lambda \quad = \quad & a\log a + b\log b + c\log c + d\log d + N\log N \\ & -(a+c)\log(a+c) - (a+b)\log(a+b) \\ & -(c+d)\log(c+d) - (d+b)\log(d+b) \end{aligned}$$

$a$: est le nombre de fois où $A$ et $B$ se suivent
$b$: est le nombre de fois où $A$ apparait non suivi de $B$
$c$: est le nombre de fois où $B$ est non précédé de $A$
$d$: est le nombre de fois où ni $A$ ni $B$ n'apparaissent, dans cet ordre, dans le corpus
$N$: a + b + c + d = taille du corpus

Plus la quantité $-\log\lambda$ est grande, plus les mots testés sont dépendants.

# Ratio de vraisemblance

## D'après Dunning [1993]

Le rapport de vraisemblance d'une hypothèse particulière ($H_0$) est donné par:

$$\lambda = \frac{max_{\omega \in \Omega_0} H(\omega; k)}{max_{\omega \in \Omega} H(\omega; k)}$$

où $\Omega$ est l'espace des paramètres, et $\Omega_0$ est l'espace des paramètres correspondant à l'hypothèse .

Faisons l'hypothèse que les événements (mots) sont distribués selon une loi binomiale:

$$H(p; k, n) = \binom{n}{k} p^k (1-p)^{n-k}$$

# Ratio de vraisemblance

La comparaison de deux binomiales de paramètres respectifs $p_1$ et $p_2$ peut se faire en prenant comme hypothèse $H_0$ le fait que $p = p_1 = p_2$. Dans ce cas le rapport de vraisemblance s'écrit:

$$\lambda = \frac{max_p H(p, p; k_1, n_1, k_2, n_2)}{max_{p_1,p_2} H(p_1, p_2; k_1, n_1, k_2, n_2)}$$

où la vraisemblance est donnée par:
$H(p_1, p_2; k_1, n_1, k_2, n_2) = \binom{n_1}{k_1} p_1^{k_1}(1-p_1)^{n_1-k_1} \times \binom{n_2}{k_2} p_2^{k_2}(1-p_2)^{n_2-k_2}$

Le maximum du dénominateur est obtenu pour $p_1 = \frac{k_1}{n_1}$ et $p_2 = \frac{k_2}{n_2}$
Le maximum du numérateur est obtenu pour $p = \frac{k_1+k_2}{n_1+n2}$

# Ratio de vraisemblance

On a déjà vu que le maximum de vraisemblance d'une binomiale correspond au maximum de vraisemblance:

$$\frac{\delta H(p_1, p_2; k_1, n_1, k_2, n_2)}{\delta p_1} = 0$$

$$\begin{aligned}
\Rightarrow \quad & \overbrace{\binom{n_1}{k_1}\binom{n_2}{k_2}}^{cst} p_2^{k_2}(1-p_2)^{n_2-k_2} \frac{\delta p_1^{k_1}(1-p_1)^{n_1-k_1}}{\delta p_1} = 0 \\
\Rightarrow \quad & p_1^{k_1-1}\left[k_1(1-p_1)^{n_1-k_1} - p_1(n_1-k_1)(1-p_1)^{n_1-k_1-1}\right] = 0 \\
& (1-p_1)^{n_1-k_1-1}\left[k_1(1-p_1) - p(n_1-k_1)\right] = 0 \\
& k_1(1-p_1) - p_1(n_1-k_1) = 0 \\
& k_1 - k_1p_1 - p_1n_1 + k_1p_1 = 0 \\
& p_1 = \frac{k_1}{n_1}
\end{aligned}$$

Idem pour $p_2$, et même genre de calcul pour $p$.

# Ratio de vraisemblance

Le rapport de vraisemblance s'écrit alors:

$$\lambda = \frac{L(p, k_1, n_1) \times L(p, k_2, n_2)}{L(p_1, k_1, n_1) \times L(p2, k_2, n_2)}$$

où $L(p, k, n) = p^k(1-p)^{n-k}$

Soit, en prenant le logarithme:

$$\begin{aligned} -\log \lambda = \quad & \log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) \\ & - \log L(p, k_1, n_1) - \log L(p, k_2, n_2) \end{aligned}$$

avec $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$, et $p = \frac{k_1+k_2}{n_1+n2}$

# Ratio de vraisemblance appliqué à l'identification de séquences

Soit $A$ et $B$ deux mots dont nous voulons mesurer le degré de dépendance. Notre hypothèse $H_0$ peut être formulée en suivant le moule précédant, en statuant l'indépendance de $A$ et $B$ par $P(B|A) = P(B|\neg A) = P(B)$.

Avec la table de contingence suivante, où par exemple, $a$ (resp. $b$) indique le nombre de fois où $B$ suit (resp. ne suit pas) $A$ dans un corpus:

| | $B$ | $\neg B$ |
|---|---|---|
| $A$ | a | b |
| $\neg A$ | c | d |

alors $p_1 = \frac{a}{a+b}$ (la probabilité $p(B|A)$) et $p_2 = \frac{c}{c+d}$ (la probabilité $p(B|\neg A)$).

En développant, on retombe sur la formulation initiale du rapport de vraisemblance.

# Comment sélectionner les bonnes séquences ?

Les 16 premières (et les 4 dernières) séquences de deux mots selon le score $-\log\lambda$ sur les 10 tranches du Hansard:

| fréq. | $-\log\lambda$ | séquence | fréq. | $-\log\lambda$ | séquence |
|---|---|---|---|---|---|
| 1391 | 4690.07 | le président | 985 | 2502.4 | président , |
| 988 | 4400.53 | c' est | 307 | 2387.39 | p. 100 |
| 1083 | 3935.63 | monsieur le | 592 | 2117.01 | la chambre |
| 893 | 3504.88 | ) : | 1594 | 1943.32 | de l' |
| 724 | 2930.47 | : monsieur | 1760 | 1896.21 | de la |
| 540 | 2899.87 | j' ai | 643 | 1679.89 | le député |
| 1069 | 2734.26 | le gouvernement | 385 | 1670.77 | nous avons |
| 867 | 2650.22 | qu' il | 293 | 1509.37 | premier ministre |
| 2 | 3.7e-6 | maintenant un | 10 | 2.2e-6 | loi que |
| 2 | 2.3e-6 | là a | 4 | 7.9e-7 | jamais . |

**Question:** comment étendre cette métrique à des séquences de taille variable ?

# Étendre une mesure d'association binaire pour noter toute séquence

**Idée:** Une séquence est non pertinente si on la retrouve comme préfixe ou suffixe de séquences pertinentes Ries et al. [1995].

Le score $\rho$ d'une séquence $w_1^n$ peut être donné en utilisant une mesure d'association binaire $A$ par:

$$\rho(w_1^n) = \min_{i \in [1, n-1]} A(w_1^i, w_{i+1}^n)$$

Ce score mesure la "résistance d'une séquence à la division", et donc reflète d'une certaine manière son degré de cohésion.

# Comment sélectionner les bonnes séquences ?

Les 12 séquences les mieux notées selon $\rho$ sur 10 tranches du Hansard, et les 4 séquences les moins bien notées:

| fréq. | $\rho(s)$ | s=séquence | fréq. | $\rho(s)$ | s=séquence |
|---|---|---|---|---|---|
| 1076 | 6297.4 | monsieur le président | 608 | 3120.2 | ) : monsieur |
| 1391 | 4690.0 | le président | 971 | 3075.9 | monsieur le président , |
| 988 | 4400.5 | c' est | 723 | 2936.7 | : monsieur le président |
| 1083 | 3935.6 | monsieur le | 724 | 2930.4 | : monsieur |
| 893 | 3504.8 | ) : | 540 | 2899.8 | j' ai |
| 608 | 3122.0 | ) : monsieur le président | 459 | 2875.2 | projet de loi |
| 3 | 1.8e-6 | est sur le | 3 | 5.9e-7 | des mesures . |
| 4 | 7.9e-7 | jamais . | 2 | 3.9e-7 | la souveraineté . |

# Comment sélectionner les bonnes séquences ?

D'autres scores sont possibles. Par exemple Shimohata et al. [1997] ont reporté un bon comportement du score d'entropie suivant:

$$\begin{aligned} e(w_1^n) &= (e_{left}(w_1^n) + e_{right}(w_1^n))/2 \\ e_{left}(s) &= \sum_{w/ws \in T} h(\frac{|ws|}{|s|}) \\ e_{right}(s) &= \sum_{w/sw \in T} h(\frac{|sw|}{|s|}) \\ h(x) &= x \log(x) \end{aligned}$$

$e_{left}(s)$ (resp. $e_{right}(s)$) est nul quand seulement une forme suit (resp. précède) $s$ dans toutes les occurrences de $s$; il est maximal, lorsqu'il y a exactement $freq(s)$ types qui suivent (resp. précèdent) $s$.

↪**Intuitivement:** une séquence cohérente devrait apparaître dans un nombre varié de contextes ⇒ entropie élevée.

# Illustration du score d'entropie

$$\left.\begin{array}{l} - \\ , \\ : \\ \text{ce} \end{array}\right\} \text{ monsieur } \left\{\begin{array}{l} \text{a} \\ \text{le} \end{array}\right.$$

faible entropie

$$(364) \left\{\begin{array}{l} \text{abandon} \\ \text{accepté} \\ \vdots \\ \text{vraiment} \end{array}\right\} \text{ le } \left\{\begin{array}{l} 10 \\ \text{baril} \\ \vdots \\ \text{yukon} \end{array}\right\} (440)$$

forte entropie

# Score d'entropie $e$

| fréq. | $e(s)$ | s=séquence | fréq. | $e(s)$ | s=séquence |
|---|---|---|---|---|---|
| 1760 | 2.5362 | de la | 393 | 2.39531 | d' un |
| 333 | 2.53098 | a été | 517 | 2.34532 | que les |
| 370 | 2.46355 | , les | 371 | 2.34049 | d' une |
| 256 | 2.44128 | et les | 156 | 2.2844 | ont été |
| 385 | 2.4374 | nous avons | 379 | 2.28096 | il est |

**Note:** 40% des séquences de deux mots ou plus ont un score nul avec cette métrique.

# Séquences ($f \geq 2$) contenant *aviation safety* (Hansard)

| | $\rho(s)$ | $e(s)$ | freq. | $l$ | s |
|---|---|---|---|---|---|
| 1 | 113.32 | 0.23 | 16 | 2 | aviation safety |
| 2 | 109.19 | 1.14 | 15 | 3 | aviation safety board |
| 3 | 92.02 | 0.24 | 15 | 3 | canadian aviation safety |
| 4 | 85.69 | 1.13 | 14 | 4 | canadian aviation safety board |
| 5 | 35.33 | 1.23 | 14 | 4 | the canadian aviation safety |
| 6 | 32.62 | 1.99 | 13 | 5 | the canadian aviation safety board |
| 7 | 12.91 | 0.69 | 2 | 4 | aviation safety board recommendations |
| 8 | 6.29 | 1.39 | 4 | 6 | the canadian aviation safety board , |
| 9 | 5.97 | 0.69 | 4 | 5 | canadian aviation safety board , |
| 10 | 5.67 | 0.69 | 4 | 4 | aviation safety board , |
| 11 | 5.65 | 0.69 | 2 | 6 | by the canadian aviation safety board |
| 12 | 5.49 | 0.35 | 2 | 5 | by the canadian aviation safety |
| 13 | 5.28 | 0.35 | 2 | 7 | of the canadian aviation safety board . |
| 14 | 3.81 | 1.10 | 3 | 6 | , the canadian aviation safety board |
| 15 | 3.59 | 0.55 | 3 | 5 | , the canadian aviation safety |
| 16 | 2.71 | 0.32 | 3 | 6 | the canadian aviation safety board . |
| 17 | 2.51 | 0 | 3 | 5 | canadian aviation safety board . |
| 18 | 2.32 | 0 | 3 | 4 | aviation safety board . |
| 19 | 1.89 | 0.35 | 2 | 6 | of the canadian aviation safety board |
| 20 | 1.76 | 0.35 | 2 | 5 | of the canadian aviation safety |

# Exemple de filtre possible

Choisissons d'éliminer une séquence si c'est le préfixe d'une séquence mieux notée par l'entropie.

```
Filtering of tv 113.323 h= 0.233792 f 16 lg 2 seq= aviation safety
Because of tv 109.194 h= 1.13558 f 15 lg 3 seq= aviation  safety board

Filtering of tv 92.0237 h= 0.24493 f 15 lg 3 seq= canadian aviation safety
Because of tv 85.6912 h= 1.13244 f 14 lg 4 seq= canadian  aviation safety board

Filtering of tv 86.9009 h= 0.233792 f 16 lg 2 seq= canadian aviation
Because of tv 85.6912 h= 1.13244 f 14 lg 4 seq= canadian  aviation safety board

Filtering of tv 38.0429 h= 1.27421 f 15 lg 3 seq= the canadian aviation
Because of tv 32.6178 h= 1.99151 f 13 lg 5 seq= the canadian  aviation safety board

Filtering of tv 35.3279 h= 1.23146 f 14 lg 4 seq= the canadian aviation safety
Because of tv 32.6178 h= 1.99151 f 13 lg 5 seq= the canadian aviation safety board
```

# Revenons au cas bilingue

- Supposons que nous soyons capables de déterminer pour chaque langue une liste d'unités pertinentes (soit $\mathcal{L}_s$ la liste source et $\mathcal{L}_c$ la liste cible).
- Une approche simple pour obtenir des associations bilingues de ces séquences consiste à "retokeniser" (découper à nouveau en "mots") le corpus d'entraînement au regard des listes $\mathcal{L}_s$ et $\mathcal{L}_c$.
- On peut alors entraîner un modèle de traduction dont les "mots" sont soit de véritables mots, soit des séquences de mots.
- **Note:** Cette approche est sensible à la qualité des regroupements effectués au moment de la transformation du corpus d'entraînement.

# Exemples d'associations (bruitées) ainsi obtenues:

| source unit ($s$) | $f(s)$ | target units ($[\alpha, p]$) |
|---|---|---|
| we have | 1748 | [nous,0.49] [avons,0.41] [, nous avons,0.07] |
| we must | 720 | [nous devons,0.61] [il faut,0.19] [nous,0.14] |
| this bill | 640 | [ce projet de loi,0.35] [projet de loi .,0.21] [projet de loi,0.18] |
| people of canada | 282 | [les canadiens,0.26] [des canadiens,0.21] [la population,0.07] |
| mr. speaker : | 269 | [m. le président :,0.80] [a,0.07] [à la,0.06] |
| what is happening | 190 | [ce qui se passe,0.21] [ce qui se,0.16] [et,0.15] |
| of course , | 178 | [évidemment ,,0.26] [naturellement,0.08] [bien sûr,0.08] |
| is it the pleasure of the house to adopt the | 14 | [plaît-il à la chambre d' adopter,0.49] [la motion ?,0.42] [motion ?,0.04] |

# Explication possible du bruit

- Ex: Soit le corpus d'entraînement $\mathcal{T}$: (il est parti en train de banlieue / he left with a commuter train)

- Et soit $\mathcal{L}_s = \{\text{en train de}, \text{train de banlieu}\}$ et $\mathcal{L}_t = \{\text{commuter train}\}$

```
[he] [left] [with] [a] [commuter train]
---------------------------------------
[il] [est] [parti] [en] [train de banlieu]
[il] [est] [parti] [en train de] [banlieu]
```

- Rien ne garantit que les unités sources ont un correspondant dans la liste des unités cibles (c'est l'inconvénient de déterminer isolément les unités dans chaque langue).

# Éliminer du bruit

De nombreux auteurs ont proposé de restreindre leur étude à des groupes nominaux (*noun phrase*) Gaussier [1995], Kupiec [1993], hua Chen and Chen [1994], Fung [1995], Evans and Zhai [1996].

Un filtre à expressions régulières peut faire l'affaire:

```
(NomC|Ordi|NomP|AdjQ|Quan)
( (Quan|Dete|NomC|Ordi|NomP|AdjQ|Prep|VPPR))*
(Quan|NomC|Ordi|NomP|AdjQ|VPPS)
```

# Extraits d'un modèle appris sur des mots et des NPs[3]

↪ *boom* → prospérité,0.32 essor,0.27 explosion démographique,0.2 explosion,0.11 vague de prospérité,0.11
↪ *fbdb* → banque fédérale de développement,1
↪ *rights of women* → droits des femmes,1
↪ *canadian aviation safety board* → bureau canadien de la sécurité aérienne,1
↪ *office of the superintendent of financial institutions* → bureau du surintendant des institutions financières,1
↪ *newfoundland unemployment* → taux de chômage à terre-neuve,1
↪ *small craft harbours* → ports pour petits bateaux,0.53 ports pour petites embarcations,0.47
↪ *airline industry* → industrie du transport aérien,0.73 secteur du transport aérien,0.13 industrie aérienne,0.13
↪ *food processing industry* → secteur de la transformation des aliments,1
↪ *ordinary canadians* → canadiens ordinaires,0.72 canadiens moyens,0.19 simples canadiens,0.082

**Note:** *fbdb* is an acronym for *Federal Business Development Bank*

[3] Obtenu à partir d'un extrait du Hansard après filtrage

# Cherchons des séquences non contigües

Maarek et al. [1991] décrivent une approche permettant de sélectionner des affinités lexicales (collocations non contigües)

Martin and van Sterkenburg [1983] ont montré que pour la langue anglaise, 98% des relations lexicales mettent en jeu des mots qui sont distants d'au plus 5 mots dans une phrase (sans compter les mots outils).

$\Rightarrow$ *pour un texte $\mathcal{D}$ donné, on fait glisser une fenêtre de taille fixe et on accumule les fréquences de tous les couples de mots apparus dans ces fenêtres. Ceci constitue une grande quantité de bigrammes (contigüs ou non), qu'il faut filtrer afin de ne garder que ceux qui sont pertinents.*

# Cherchons des séquences non contigües

Maarek et al. [1991] proposent de calculer pour une affinité lexicale $(m_1, m_2)$ un score (appelé pouvoir de résolution) comme suit:

$$\rho(m_1, m_2) = -freq_{\mathcal{D}}(m_1, m_2) \times log_2(p(m_1)p(m_2))$$

où $freq_{\mathcal{D}}(m_1, m_2)$ est la fréquence de l'affinité lexicale dans le document $\mathcal{D}$, et $p(x)$ désigne la probabilité de l'événement $x$ dans la langue (calculée à partir d'un grand corpus représentatif de la langue utilisée).

Maarek et al. [1991] proposent de conserver pour un document donné les seules affinités lexicales dont le pouvoir de résolution est supérieur à la moyenne de la distribution plus son écart type (why not !).

# Application possible des affinités lexicales

Soit un ensemble de documents. On peut calculer pour chacun d'eux un *profil* constitué des affinités lexicales du document les mieux notées.

Exemple de *profils* calculés sur des *group-news*:

| group | les premières affinités lexicales |
|---|---|
| hardware | [drive, hard] [clock speed] [clock oscillator] |
| baseball | [game, save] [baseball, player] [game team] [bolick, frank] |
| hockey | [gargle, howl] [game, play] [player team] [good team] |
| politics | [homosexual, male] [care health] [jews, mormons] |
| medical | [antibiotic, chapter] [bloom candida] [chronic, hepatitis] |

Lire Alvarez et al. [2004] pour une application des affinités lexicales en recherche d'information.

# Trouver des traductions dans des corpus bilingues parallèles mais bruités

Dans le cas bilingue, le calcul d'un score de similarité à partir d'une table de contingence impose que les segments sur lesquels on compte les événements soient identifiés. Ces segments nous sont donnés par une étape d'alignement de phrase. Que se passe-t-il dans le cas d'un corpus bruité ?

Pleins de réponses possibles, en voici une classique Fung and Church [1994]: les K-vecs.

**Idée:** Diviser chaque texte du corpus bilingue en K régions (uniformes). Associer à chaque mot $w$ (source et cible), un vecteur de positions $p$ où $p[i]$ vaut 1 si $w$ est dans la région $i \in [1, K]$ du texte. Deux mots sont en relation de traduction s'ils partagent des K-vec "proches".

# Trouver des traductions dans des corpus bilingues parallèles mais bruités

Pour contourner le problème de linéarité imposé par cette division uniforme des deux textes en K régions, les auteurs proposent de représenter des mots par des vecteurs de positions relatives $v$:

$$v[i] = p[i+1] - p[i] \;\; \forall i \in [1, K-1]$$

où $p$ est ici un vecteur positionnel dont la dimension correspond à la taille du texte (comptée en mots). Un algorithme de programmation dynamique est proposé pour aligner deux vecteurs (car ils n'ont pas forcément la même dimension).

# Exemple de représentation des mots

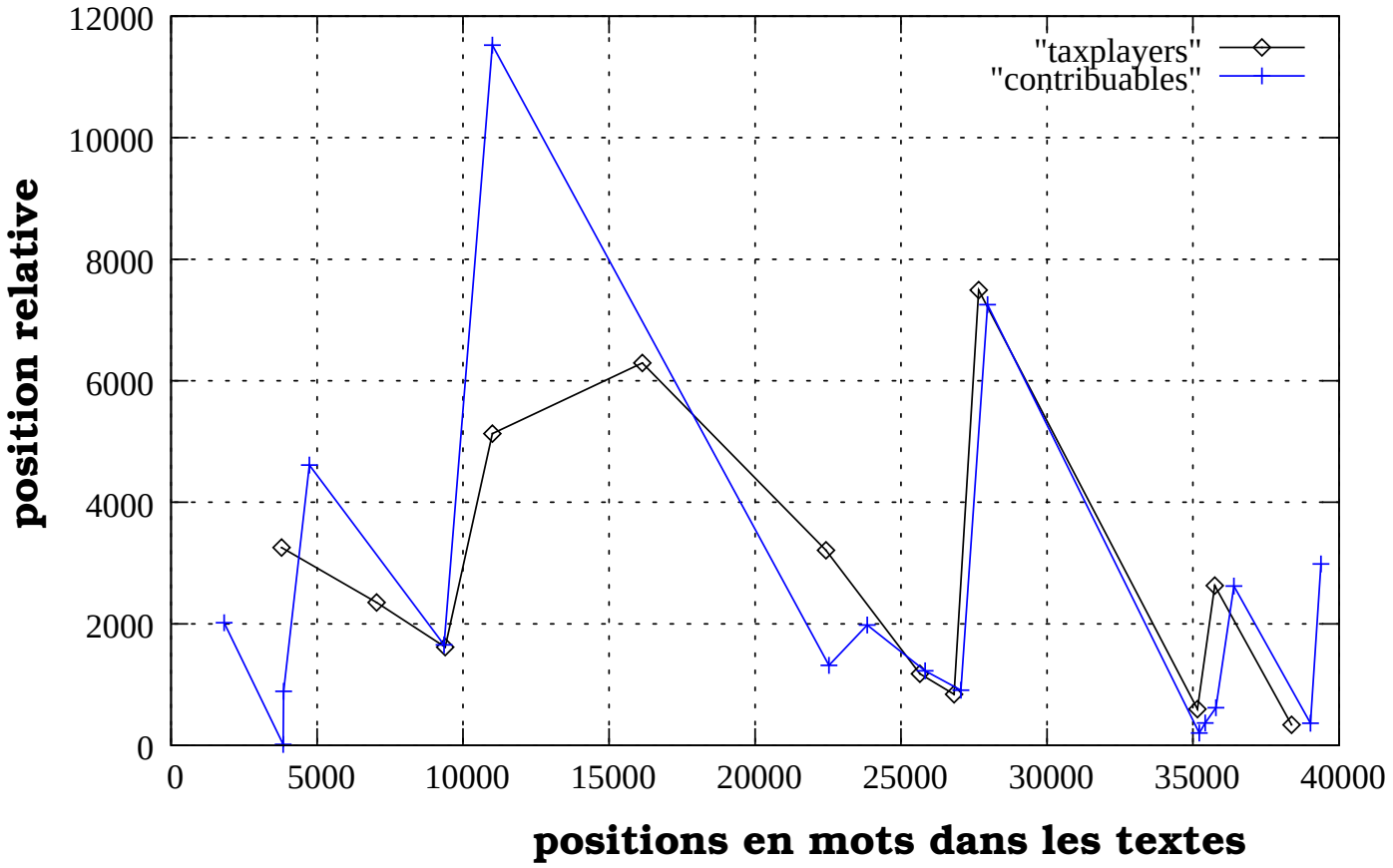
Dans un passage du Hansard 42 799 mots anglais et sa traduction française de 43 607 mots (1 867 phrases), voici les vecteurs de position de deux mots:

contribuables: 1821 3839 3853 4740 9353 11006 22529 23846 25826 27054 27961 35213 35416 35782 36400 39023 39385 42369
taxpayers: 3785 7041 9393 11008 16139 22433 25643 26819 27660 35153 35749 38376 38712

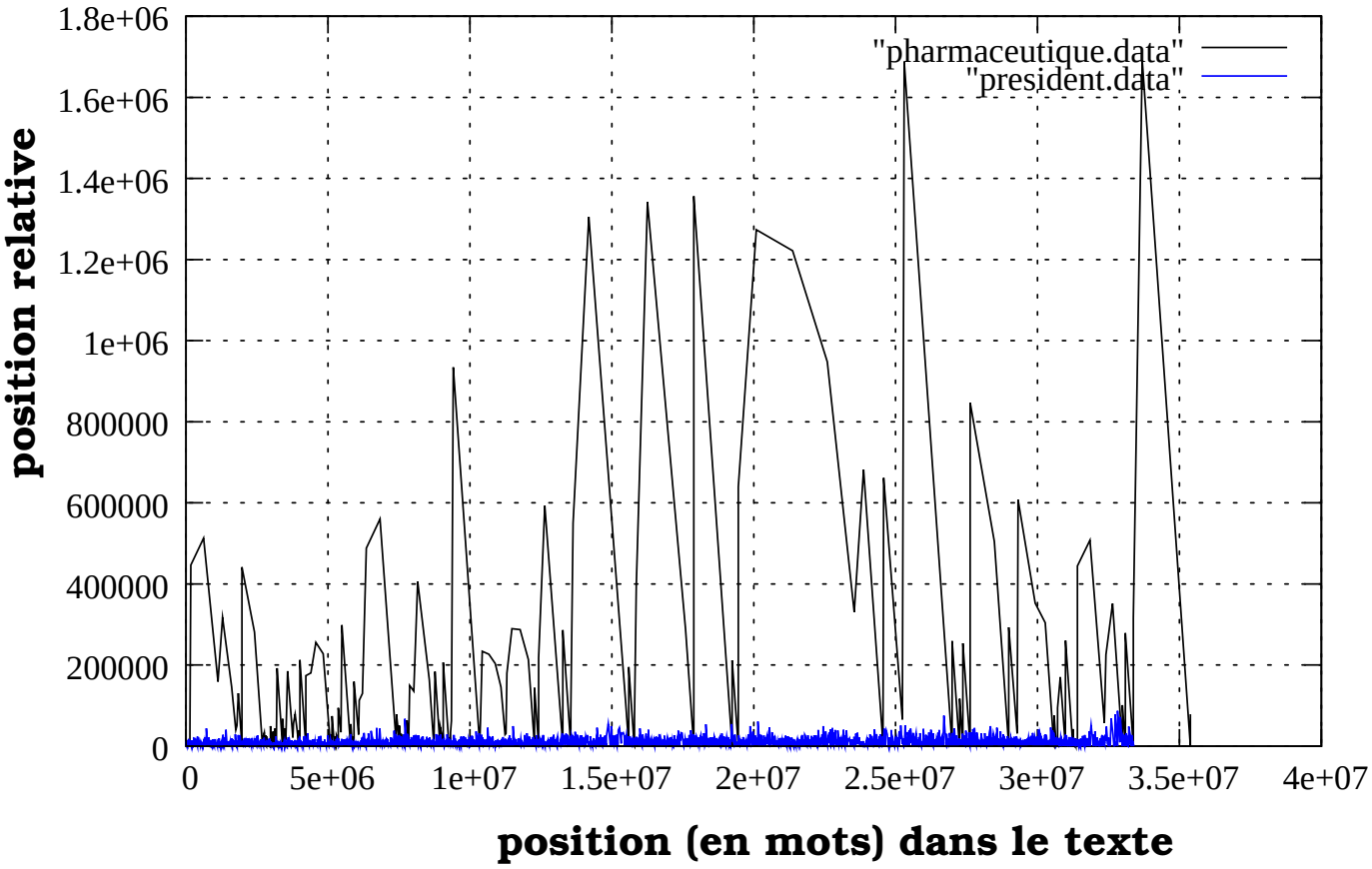
et leur encodage relatif:

contribuables: 2018 14 887 4613 1653 11523 1317 1980 1228 907 7252 203 366 618 2623 362 2984
taxpayers: 3256 2352 1615 5131 6294 3210 1176 841 7493 596 2627 336

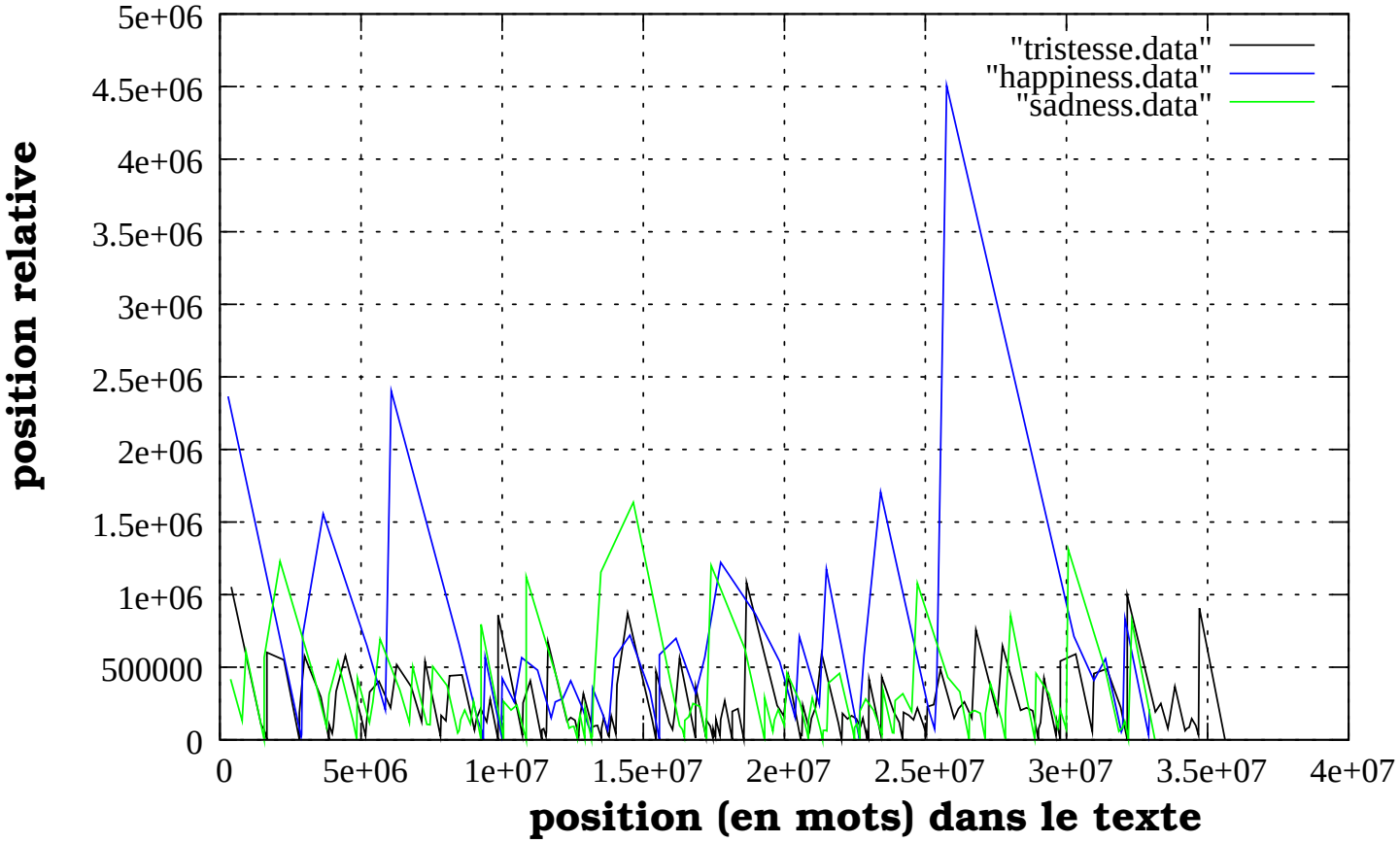
# Exemple de représentation des mots



# Exemple de représentation des mots



# Exemple de représentation des mots



# Trouver des traductions dans des corpus bilingues non parallèles

- **Challenge:** étant donnés deux textes de deux langues différentes, non corrélés ($\grave{a}\ priori$). Peut-on découvrir automatiquement des paires de mots source/cible qui sont traduction l'un de l'autre ?

- **Quiz:** avez-vous une idée du type d'indice que l'on pourrait tenter d'exploiter ?

Exemple de corpus non parallèle:

| | |
|---|---|
| SRC | Le ciel est bleu, pas rouge. |
| TGT | All of a sudden, with a reassessment out of the clear blue sky this older couple is now faced with a $72,000 tax bill. |

# Trouver des traductions dans des corpus bilingues non parallèles

- Selon Rapp [1995], il est difficile de rassembler des corpus bilingues parallèles pour des domaines spécifiques, même pour des couples de langues répandus.

- En l'absence de bitexte, la plupart des algorithmes décrits sont inefficaces pour découvrir des associations bilingues (on part souvent de l'alignement en phrase).

- **Note:** Il existe plusieurs études qui montrent cependant qu'il est possible d'aller chercher automatiquement sur le web, à l'aide d'heuristiques sur le nom des pages (par exemple), des corpus parallèles de qualité suffisante Chen [2000][4].

[4] Version HTML disponible à `http://www.iro.umontreal.ca/~chen/thesis/`

# Trouver des traductions dans des corpus bilingues non parallèles

- On fait l'hypothèse que les co-occurrences fortes dans une langue sont également des co-occurrences fortes dans une autre langue.

- Cette hypothèse est partiellement vérifiée pour le couple anglais/allemand dans une simulation décrite dans Rapp [1995].

**Note:** plusieurs auteurs s'intéressent à la découverte d'associations bilingues (au niveau des mots ou des groupes de mots) dans des corpus non parallèles ou fortement bruités Fung [1995], Tanaka and Iwasaki [1996], Tanaka and Matsuo [1999] ou encore dans des corpus parallèles mais sans alignement préalable des phrases Ohmori and Higashida [1999].

# Illustration de l'idée développée dans Rapp [1995]

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $blue_1$ | | x | | | x | |
| $green_2$ | x | | x | | | |
| $plant_3$ | | x | | | | |
| $school_4$ | | | | | | x |
| $sky_5$ | x | | | | | |
| $teacher_6$ | | | | x | | |

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $blau_1$ | | x | x | | | |
| $grün_2$ | x | | | | x | |
| $Himmel_3$ | x | | | | | |
| $Lehrer_4$ | | | | | | x |
| $Pflanze_5$ | | x | | | | |
| $Schule_6$ | | | | x | | |

| | 1 | 2 | 5 | 6 | 3 | 4 |
|---|---|---|---|---|---|---|
| $blue_1$ | | x | x | | | |
| $green_2$ | x | | | | x | |
| $sky_5$ | x | | | | | |
| $teacher_6$ | | | | | | x |
| $plant_3$ | | x | | | | |
| $school_4$ | | | | x | | |

# Ressources utilisées dans Rapp [1999]

1. un corpus allemand (135 millions de mots du *Frankfurter Allgemeine Zeitung* couvrant une période de 1993 à 1996)
2. un corpus anglais (163 millions de mots du *Guardian* couvrant une période de 1990 à 1994)
3. un lexique bilingue de base allemand $\rightarrow$ anglais (16380 entrées extraites du *Collins Gem German Dictionnary*)
4. une liste de mots allemands (test) dont on cherche la traduction (100 mots); une traduction privillégiée a été associée à la main pour des fins d'évaluation

**Note:** Les corpus allemand et anglais ont été lemmatisés pour limiter le vocabulaire. Les mots outils sont également retirés.

# Étude décrite dans Rapp [1999]

**Étape 1:** calcul d'une matrice de co-occurrence pour la langue anglaise.

**lignes:** les différents types du corpus anglais de fréquence 100 ou plus;

**colonnes:** les mots anglais qui apparaissent dans le lexique de base.

**score:** rapport de vraisemblance, les lignes sont ensuite normalisées

# Étude décrite dans Rapp [1999]

**Étape 2:** calcul pour chaque mot test allemand d'un vecteur de co-occurrence. Même méthode que précédemment. $\Rightarrow X_i$ un vecteur pour le i-ème mot test allemand.

**Étape 3:** Comparer $X_i$ avec les vecteurs $Y_j$ de la matrice de co-occurrence et sélectionner le mot anglais ayant le meilleur taux de similarité. La similarité entre deux vecteurs A et B est ici:

$$s = \sum_i |A_i - B_i|$$

pour d'autres métriques couramment employées, voir Manning and Schütze [1999] page 299.

# Étude décrite dans Rapp [1999]

| allemand | attendu | rang | les 5 meilleurs traductions trouvées | |
|---|---|---|---|---|
| Baby | baby | 1 | baby | child mother daughter father |
| Brot | bread | 1 | bread | cheese meat food butter |
| Frau | woman | 2 | man | woman boy friend wife |
| gelb | yellow | 1 | yellow | blue red pink green |
| Haüschen | cottage | 2 | bungalow | cottage house hut village |
| Kind | child | 1 | child | daughter son father mother |
| Kohl | cabbage | 17074 | Major | Kohl Thatcher Gorbachev Bush |
| Musik | music | 1 | music | theatre musical dance song |
| Tabak | tobacco | 1 | tobacco | cigarette consumption nicotine drink |
| weiss | white | 46 | know | say thought see think |
| Whisky | whiskey | 11 | whisky | beer Scotch bottle wine |

65 des 100 mots avaient leur traduction privillégiée (choisie à la main) en tête, 72 des traductions proposées en tête étaient cependant correctes. 89 des mots du test avaient une traduction correcte dans les 10 premières positions.

# Trouver la probabilité d'une co-occurrence jamais vue

Étude décrite dans Dagan et al. [1993]

- **Note:** C'est une technique de lissage ainsi qu'une approche possible aux regroupements des mots en classes.

- **Idée:** $eat\ bread$ n'a jamais été vue, mais $eat\ toast$ a été vue. C'est probablement un bon bigramme pour notre estimée.

- Les techniques de lissage se rabattant sur l'unigramme perdent le lien qu'entretiennent ces mots:

  $\hookrightarrow$ eat bread et eat car peuvent avoir la même estimée si $bread$ et $car$ sont aussi fréquents dans un corpus d'entraînement.

# Probabilité d'une co-occurrence jamais vue

- **Hypothèse:** des co-occurrences similaires ont des valeurs d'information mutuelle similaires.

- **Définition** d'une co-occurrence dans Dagan et al. [1993]:

  Toute paire de mots qui co-occurent dans une fenêtre de $d$ mots (3 dans les expériences), abstraction faite des mots outils. Les paires sont directionnelles $(x, y) \neq (y, x)$.

- Pour une occurrence donnée, on peut mesurer l'association de ses mots. Les auteurs choisissent l'information mutuelle ($N$ est la taille du corpus):

$$I(x, y) = \log_2 \left( \frac{N}{d} \frac{f(x, y)}{f(x) f(y)} \right)$$

# Similarité de deux co-occurrences

**Définition:** deux co-occurrences $(w_1, w_2)$ et $(w'_1, w'_2)$ sont similaires ssi $w_1$ est similaire à $w'_1$ et $w_2$ est similaire à $w'_2$.

**Exemple:** (chapter,describes) n'est pas observée dans un corpus de 9 million de mots postés à USENET. Les co-occurrences suivantes ont cependant été vues dans ce même corpus et sont similaires à cette paire (au sens d'une métrique à préciser): *(introduction,describes)*, *(book,describes)*, *(section, describes)*.

**Idée:** l'information mutuelle d'une co-occurrence inconnue $(w_1, w_2)$ est estimée par la moyenne de l'information mutuelle des paires similaires: $\hat{I}(w_1, w_2)$. On peut alors obtenir notre estimée de la paire $(w_1, w_2)$ par:

$$\hat{f}(w_1, w_2) = \frac{d}{N} f(w_1) f(w_2) 2^{\hat{I}(w_1, w_2)}$$

# Similarité de deux co-occurrences

$I(introduction, describes) = 6.85$
$I(book, describes) = 6.27$
$I(section, describes) = 6.12.$

$\Rightarrow \hat{I}(chapter, book) = 6.41 \Rightarrow \hat{f}(chapter, book) = 0.124$

$\hookrightarrow$ C'est plus élevé que ce qu'on obtiendrait à partir des fréquences des mots considérés comme indépendants.

**Détail d'implémentation:** pour calculer $\hat{I}(w_1, w_2)$, les auteurs prennent les 6 mots les plus similaires à $w_1$ et les 6 mots les plus similaires à $w_2$, et la moyenne est calculée sur toutes les co-occurrences d'une combinaison de ces (au plus) 6x6 mots.

# Similarité entre deux mots

- **Idée:** Deux mots $w_1$ et $w_2$ sont similaires s'ils co-occurrent de manière semblable avec toutes sortes de mots (vive la récursivité...);

  $\hookrightarrow$ cad, s'ils ont des valeurs semblables d'information mutuelle avec d'autres mots du vocabulaire.

- Les paires étant directionnelles, il y a un ratio à droite et un ratio à gauche:

$$sim_L(w_1, w_2, w) = \frac{min(I(w,w_1),I(w,w_2))}{max(I(w,w_1),I(w,w_2))}$$
$$sim_R(w_1, w_2, w) = \frac{min(I(w_1,w),I(w_2,w))}{max(I(w_1,w),I(w_2,w))}$$

  (à valeur dans $[0, 1]$, $1 =$ très similaire, $0 =$ pas similaire).

- Ces ratios sont à calculer pour chaque mot $w$ du vocabulaire (gasp!)

# Similarité entre deux mots

Le tout mis bout à bout, voici la mesure de similarité entre deux mots:

$$sim(w_1, w_2) = \frac{\sum_{w \in V} min(I(w, w_1), I(w, w_2)) + min(I(w_1, w), I(w_2, w))}{\sum_{w \in V} max(I(w, w_1), I(w, w_2)) + max(I(w_1, w), I(w_2, w))}$$

Exemple des mots les plus similaires au mot *aspects*:

| recherche exhaustive | | approximation | |
|---|---|---|---|
| mots similaires | $sim$ | mots similaires | $sim$ |
| aspects | 1.0 | aspects | 1.0 |
| topics | 0.1 | topics | 0.1 |
| areas | 0.09 | areas | 0.09 |
| expert | 0.079 | expert | 0.079 |
| issues | 0.076 | issues | 0.076 |
| approaches | 0.072 | concerning | 0.069 |

# Similarité entre deux mots: détails

- La recherche approximative des mots similaires se fait par indirection et seuillage: les plus similaires des plus similaires du mot $w$ dont on cherche les mots similaires (réduction du temps de 17 minutes à 7 secondes pour trouver les mots similaires d'un mot).

- **Question à 10 piasses:** est-ce que ça marche ?

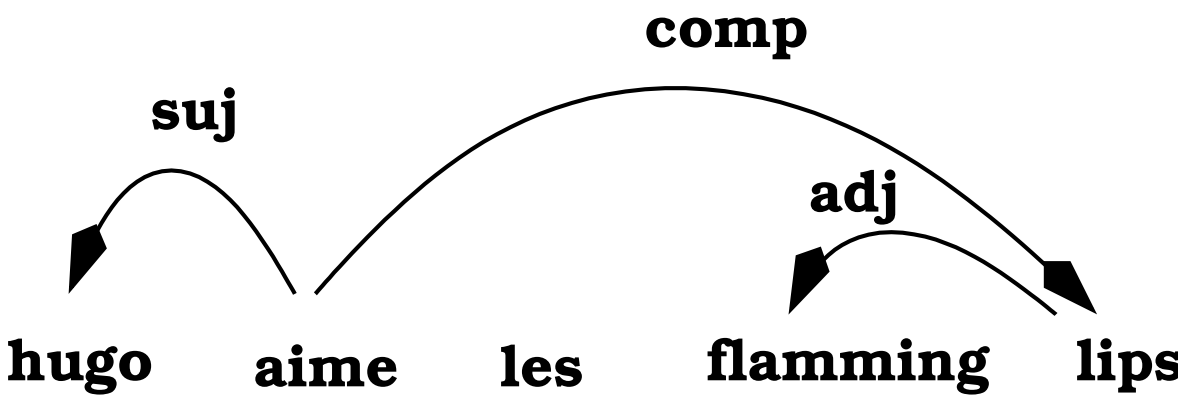
  $\hookrightarrow$ Réponse à 1 cent: lisez Dagan et al. [1993].

- **Note finale:** d'autres études sur le même thème: Lee and Pereira [1999], Lee [1999]

# Identification d'expressions non compositionnelles

## Travaux de Lin [1999]

Soit un analyseur grammatical en dépendances (ici Minipar[5]):




produisant des triplets (tête, relation, modifieur): (aime,sujet,hugo), (aime,comp,lips), (lips,adj,flammings)

Lire aussi Melamed [1997] pour une identification bilingue.

[5] http://www.cs.ualberta.ca/~lindek/minipar.htm

# Hypothèse de Lin [1999]

Une collocation $(H, R, M)$ est non compositionnelle si son information mutuelle diffère de manière significative de l'information mutuelle de collocations proches.

$\Rightarrow$ deux définitions:

- $(H', R, M')$ est proche de $(H, R, M)$ si $H' \in sim(H), M' \in sim(M)$ et $\neg\left(H' = H \wedge M' = M\right)$
- $I(H, R, M) = \log \frac{P(H,R,M)}{P(H|R)P(M|R)P(R)}$ où chaque distribution est apprise par fréquence relative sur un gros corpus (125 millions de mots, 80 millions de dépendances, filtrées par seuillage)

# Similarité entre deux mots Lin [1998]

Deux mots $w_1$ et $w_2$ sont similaires s'ils partagent des relations proches avec les autres mots (même idée que Dagan et al. [1993], mais avec les relations en plus):

$$sim(w_1, w_2) = \frac{\sum_{(r,w) \in T(w_1) \bigcap T(w_2)} I(w_1, r, w) + I(w_2, r, w)}{\sum_{(r,w) \in T(w_1)} I(w_1, r, w) + \sum_{(r,w) \in T(w_2)} I(w_2, r, w)}$$

où $T(w) = \{(r, w')/I(w, r, w') > 0\}$

s'approche d'autant plus de 0 que $w_1$ et $w_2$ partagent peu de relations

# Similarité entre deux mots Lin [1998]

En gardant les $N$ meilleurs associations ($N = 200$ ici) de chaque mot, l'auteur obtient un thésaurus dont voici quelques exemples[6]:

**brief(noun):** affidavit, 0.13, petition 0.05, memorandum 0.05, motion 0.05, document 0.04, paper 0.04,. . .

**brief(verb):** tell 0.09, urge 0.07, ask 0.07, meet 0.06, appoint 0.06, elect 0.05,. . .

**brief(adj):** lengthy 0.13, short 0.12, recent 0.09, prolonged 0.09, long 0.09, stormy 0.07,. . .

[6] Extrait de l'article.

# Une slide sur les thésaurus

Répertoire de termes normalisés classés alphabétiquement et répartis en structures correspondant aux divers champs de la connaissance[7].

http://thesaurus.reference.com/search?q=flower

```
Entry:  flower
Function:  noun
Definition:  bloom
Synonyms:  annual, blossom, bud, cluster, efflorescence, floret,
                floweret, head, herb, inflorescence, perennial, pompon,
                posy, shoot, spike, spray, vine

Source:  Roget's New Millennium Thesaurus, First Edition (v 1.0.5)
Copyright 2004 by Lexico Publishing Group, LLC. All rights reserved.
```

[7] Pris du dictionnaire encyclopédique de la langue française, 1996

# Non compositionnalité et IM Lin [1999]

**spill** one's **guts**[8]

**spill(verb):** leak 0.153, pour 0.127, spew 0.125, dump 0.098, seep 0.096, . . .

**gut(noun):** intestine 0.091, instinct 0.089, foresight 0.085, creativity 0.082, heart 0.079, . . .

$(spill, comp, gut)$ est apparu 13 fois dans le corpus (I=6.24), mais aucune n'est apparue en remplaçant l'un des mots par un mot similaire (ex: leak gut).

$\Rightarrow$ c'est une collocation non compositionnelle

[8] Pourrait se traduire par "se plaindre, geindre"

# Non compositionnalité et IM Lin [1999]

## red tape[9]

**red:** yellow 0.164, purple 0.149, pink 0.146, green 0.136, . . .

**tape:** videotape 0.196, cassette 0.177, videocassette 0.168, . . .

| verb | object | freq | I |
|---|---|---|---|
| red | tape | 259 | 5.87 |
| yellow | tape | 12 | 3.75 |
| orange | tape | 2 | 2.64 |
| black | tape | 9 | 1.07 |

d'autres collocations existent mais avec des informations mutuelles différentes ⇒ non compositionnelle

[9] Tracasseries administratives ?

# Non compositionnalité et IM Lin [1999]

## economic impact

**economic:** financial 0.305, political 0.243, social 0.219, fiscal 0.209, cultural 0.202, . . .

**impact:** effect 0.227, implication 0.163, consequence 0.156, significance 0.146, . . .

| verb | object | freq | I |
|---|---|---|---|
| economic | impact | 171 | 1.85 |
| economic | consequence | 59 | 1.88 |
| economic | repercussion | 7 | 1.84 |

Beaucoup de collocations ont le même genre d'information mutuelle $\Rightarrow$ compositionnelle

# Détour rapide: Intervalle de confiance

**Soit:** $X_i \sim D(\mu, \sigma^2), \quad i \in [1, N]$

- et un estimateur de la moyenne $\mu$ à partir des $N$ observations $X_{i=1}^n$: $\overline{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$
- On veut trouver l'intervalle de confiance à $(1-\alpha) \times 100\%$, cad l'intervalle à l'intérieur duquel on est certain à $(1-\alpha) \times 100\%$ que la vrai valeur est.

On sait que $\overline{X} \sim N(\mu, \sigma^2/n)$, donc $Z = \frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$.

# Détour rapide: Intervalle de confiance

Donc:

$$P(-Z_{\alpha/2} < \frac{\overline{X}-\mu}{\sigma/\sqrt{n}} < Z_{\alpha/2}) = 1-\alpha$$
$$P(\overline{X} - Z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \overline{X} + Z_{\alpha/2}\frac{\sigma}{\sqrt{n}}) = 1-\alpha$$

$\hookrightarrow$ Si $\overline{X}$ est notre estimée, alors on est certain à $(1-\alpha)\times 100\%$ que la moyenne $\mu$ de la distribution est dans $(\overline{X} - Z_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \overline{X} + Z_{\alpha/2}\frac{\sigma}{\sqrt{n}})$

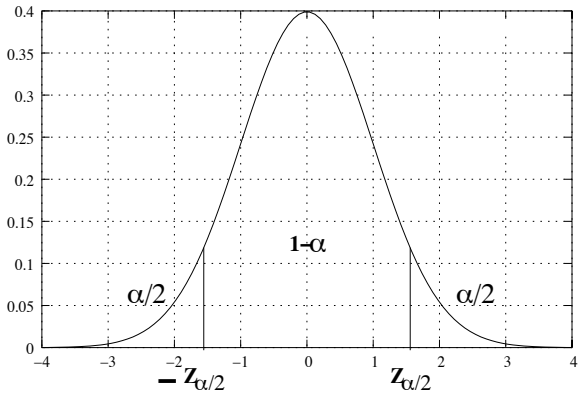
- Intervalle de confiance à 95%

$$(\overline{X} - 1.96\frac{\sigma}{\sqrt{n}}, \overline{X} + 1.96\frac{\sigma}{\sqrt{n}})$$

- Intervalle de confiance à 99%

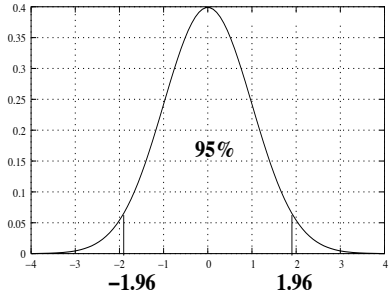
$$(\overline{X} - 2.58\frac{\sigma}{\sqrt{n}}, \overline{X} + 2.58\frac{\sigma}{\sqrt{n}})$$

# Intervalle de confiance

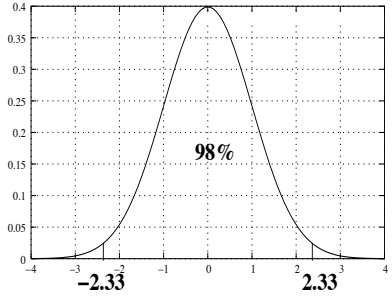



$$P(Z_{\alpha/2} < Z < Z_{\alpha/2}) = 1 - \alpha$$
$$P(Z > Z_{\alpha/2}) = \alpha/2 = 1 - \phi(Z_{\alpha/2})$$

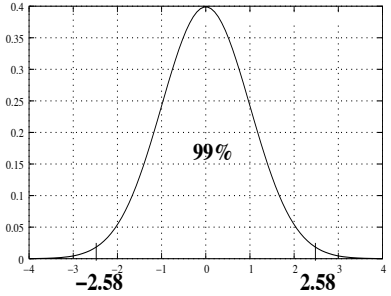
| $\alpha$ | $\alpha/2$ | percent. | $Z_{\alpha/2}$ |
|---|---|---|---|
| 0.01 | 0.005 | 99 | 2.58 |
| 0.02 | 0.01 | 98 | 2.33 |
| 0.05 | 0.025 | 95 | 1.96 |




$95\% \in [-1.96, 1.96]$




$98\% \in [-2.33, 2.33]$




$99\% \in [-2.58, 2.58]$

# Intervalle de confiance

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}y^2} dy$$

| x | .00 | .01 | .02 | .03 | .04 | .05 | . . . |
|---|---|---|---|---|---|---|---|
| 0.0 | .5000 | .5040 | .5080 | .5120 | .5160 | .5199 | |
| 0.1 | .5398 | .5438 | .5478 | .5517 | .5557 | .5596 | |
| 0.2 | .5793 | .5832 | .5871 | .5910 | .5948 | .5987 | |
| . . . | | | | | | | |
| 1.0 | .8413 | .8438 | .8461 | .8485 | .8508 | .8531 | |
| . . . | | | | | | | |
| 1.5 | .9332 | .9345 | .9357 | .9370 | .9382 | .9394 | |
| 1.6 | .9452 | .9463 | .9474 | .9484 | .9495 | .9505 | |
| . . . | | | | | | | |
| 1.9 | .9713 | .9719 | .9726 | .9732 | .9738 | .9744 | |
| 2.0 | .9772 | .9778 | .9783 | .9788 | .9793 | .9798 | |
| . . . | | | | | | | |
| 2.3 | .9893 | .9896 | .9898 | .9901 | .9904 | .9906 | |
| . . . | | | | | | | |
| 2.5 | .9938 | .9940 | .9941 | .9943 | .9945 | .9946 | |
| . . . | | | | | | | |
| 3.4 | .9997 | .9997 | .9997 | .9997 | .9997 | .9997 | |

# Que veut dire même genre d'information mutuelle ?

Réponse de Lin [1999]:

La fréquence $k$ d'un triplet est une variable aléatoire $B(n, \underline{p})$, $n$ le nombre total de triplets observés en corpus. Pour des comptes élevés (le cas ici), on peut approcher la binomiale par une gaussienne, pour laquelle on peut calculer un intervalle de confiance:

$$\frac{k}{n} \pm z_N \frac{\sqrt{k(1-\frac{k}{n})}}{n}$$

On a donc deux bornes pour le calcul de l'information mutuelle (on suppose que l'estimation de $p(M|R)$ et de $p(H|R)$ est juste et que seule l'estimation de la distribution jointe $p(H, R, M)$ est bruitée).

# Que veut dire même genre d'information mutuelle ?

| 95% verb-object | freq. | IM | lower bound | upper bound |
|---|---|---|---|---|
| make difference | 1489 | 2.928 | 2.876 | 2.978 |
| make change | 1779 | 2.194 | 2.146 | 2.239 |

Pas de recouvrement $\Rightarrow$ pas la même information mutuelle

**Def:** Une collocation est non compositionnelle s'il n'existe pas d'autre collocation proche dont les intervalles de confiance (à 95%) se chevauchent.

Ex: take a bit, take advantage, take a look, take part

# Références

ACL-31. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, Ohio, June 1993.

ACL-33. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Cambridge, Massachusetts, June 1995.

ACL-37. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, College Park, Maryland, June 1999.

Carmen Alvarez, Philippe Langlais, and Jian-Yun Nie. Word pairs in language modeling for information retrieval. In *7th Conference on Computer Assisted Information Retrieval (RIAO)*, pages 686–705, Avignon, France, 2004.

Jiang Chen. Parallel text mining for cross-language information retrieval using a statistical translation model. Master's thesis, Université de Montréal, apr 2000.

K.W. Church and P. Hanks. Word association norms, mutual information, and lexicography.

In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 76–83, Vancouver, British Columbia, 1989.

COLING-94. *Proceedings of the International Conference on Computational Linguistics (COLING) 1994*, Kyoto, Japan, August 1994.

COLING-96. *Proceedings of the International Conference on Computational Linguistics (COLING) 1996*, Copenhagen, Denmark, August 1996.

Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity ans estimation from sparse data. In ACL-31, pages 164–171.

Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 1993.

David A. Evans and Chengxiang Zhai. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 17–24, Santa Cruz, California, September 1996.

P. Fung and K.W. Church. K-vec: A new approach for aligning sentences in bilingual

corpora. In COLING-94, pages 1096–1102.

Pascale Fung. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In ACL-33, pages 236–243.

Éric Gaussier. *Modèles statistiques et patrons morphosyntaxiques pour l'extraction de lexiques bilingues*. PhD thesis, Université de Paris 7, janvier 1995.

G. Gonnet, R. Baeza-Yates, and T. Snider. *New Indices for Text: PAT trees and PAT arrays*, chapter Information Retrieval: Data Structures and Algorithms. Information Retrieval: Data Structures and Algorithms. B. Frakes and R. Baeza-Yates (eds.), Englewood Cliffs: Prentice Hall, 1992.

Masahiko Haruno, Satoru Ikehara, and Takefumi Yamazaki. Learning bilingual collocations by word-level sorting. In COLING-96, pages 525–530.

Kuang hua Chen and Hsin-Hsi Chen. Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 234–241, Las Cruces, New Mexico, June 1994.

Satoru Ikehara, Satoshi Shirai, and Hajine Uchino. A statistical method for extracting uninterrupted and interrupted collocations from very large corpora. In COLING-96, pages 574–579.

Julian Kupiec. An algorithm for finding noun phrase correspondences in bilingual corpora. In ACL-31, pages 17–22.

Lillian Lee. Measures of distributional similarity. In ACL-37, pages 25–32.

Lillian Lee and Fernando Pereira. Distributional similarity models: Clustering vs. nearest neighbors. In ACL-37, pages 33–40.

Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING/ACL*, Montreal, 1998.

Dekang Lin. Automatic identification of non-compositional phrases. In *ACL*, pages 317–324, 1999.

Yoëlle S. Maarek, Daniel M. Berry, and Gail E. Kaiser. An information retrieval approach for automatically constructing software librairies. In *IEEE Transactions on Software*

*Engineering*, volume 17(8), pages 800–813, aug 1991.

U. Manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 93.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

Martin and van Sterkenburg. On the processing of a text corpus: from textual data to lexicographic information. In Hartmann Ed., editor, *Lexicography: Principles and Practices*. London Academic, 1983.

I. Dan Melamed. Automatic discovery of non-compositional compounds in parallel data. In *EMNLP*, Providence, RI, 1997.

Makoto Nagao and Shinsuke Mori. A new method of n-gram statistics for large number of n and automatic extraction of words and phrases from large text data of japanese. In COLING-94, pages 611–615.

Kumiko Ohmori and Masanobu Higashida. Extracting bilingual collocations from

non-aligned parallel corpora. In TMI-8, pages 88–97.

Reinhard Rapp. Identifying word translation in non-parallel texts. In ACL-33, pages 320–322.

Reinhard Rapp. Automatic identification of word translations from unrelated english and german corpora. In ACL-37, pages 519–526.

Klaus Ries, Finn Dag Buo, and Ye-Yi Wang. Improved language modeling by unsupervised acquisition of structure. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 1995*, pages 193–196, Detroit, Michigan, 1995. IEEE.

Graham Russell. Identification of salient token sequences. Internal Report, RALI, 1998.

Sayori Shimohata, Toshiyuki Sugio, and Junji Nagata. Retrieving collocations by co-occurrences and word order constraints. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 476–481, Madrid, Spain, July 1997.

Frank Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):144–177, 93.

Kumiko Tanaka and Hideya Iwasaki. Extraction of lexical translations from non-aligned corpora. In COLING-96.

Takaaki Tanaka and Yoshihiro Matsuo. Extraction of translation equivalents from non-parallel corpora. In TMI-8, pages 109–119.

TMI-8. *Proceedings of the 8th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Chester, England, 1999.