

# Entropizer 1.1 : un outil informatique pour l'analyse séquentielle<sup>1</sup>

Aris Xanthos

UNIL - Linguistique - BFSH 2 - CH-1015 Lausanne - Suisse

## Abstract

We introduce a software (Xanthos, 1999) bringing together a couple of tools designed for the analysis of text viewed as a qualitative time serie or sequential analysis. First we give a brief summary of the theoretical foundations of this approach. Then we look over the main functions of ENTROPIZER 1.1, and show on the basis of many examples how the program enables us to build a model of a process of text generation and to predict future states of the variable.

## Résumé

Nous présentons ici un logiciel (Xanthos, 1999) adapté à l'analyse du texte envisagé comme série temporelle catégorielle ou analyse séquentielle. Nous donnerons un bref rappel des fondements théoriques de cette approche avant de passer en revue les principales fonctions d'ENTROPIZER 1.1, et de montrer sur la base de nombreux exemples comment le programme rend possible la modélisation d'un processus de génération textuelle et la prédiction des états futurs de la variable.

**Mots-clés:** logiciel, statistique textuelle, analyse séquentielle, séries temporelles catégorielles, théorie de l'information, entropie

## 1. Introduction

L'analyse séquentielle est une voie relativement peu explorée de la statistique textuelle. Elle est fondée sur l'idée qu'il est possible de modéliser les mécanismes qui régissent la génération d'une série temporelle catégorielle, et d'en prédire l'évolution future.

Par exemple, supposons un long texte dont nous aurions pris connaissance. Ce texte se trouve être interrompu, et l'on souhaiterait naturellement en prédire le prochain symbole. Si, pour ce faire, on se fie au seul dernier symbole du texte (un 's', par exemple), on est conduit à parier sur des issues passablement incertaines. A l'inverse, une longue séquence comme 'ier prit une chais' est bien plus qu'il n'en faut pour se décider à parier sur un 'e'. On comprend intuitivement qu'il existe un seuil au-delà duquel une meilleure connaissance du contexte ne modifie plus la qualité de notre prédiction. Par l'observation de l'entropie sur les distributions de symboles, l'analyse séquentielle permet d'inférer la valeur de ce seuil.

Les principes de l'analyse des séries temporelles catégorielles sont d'une complexité raisonnable<sup>2</sup>, et nous pensons que c'est dans une mise oeuvre relativement coûteuse qu'il faut chercher les causes de la méconnaissance de l'approche - et la motivation du logiciel que nous présentons ici.

---

<sup>1</sup> disponible gratuitement par E-mail auprès de l'auteur (Aris.Xanthos@ling.unil.ch) ou sur le site de la section de linguistique de l'Université de Lausanne (<http://www.unil.ch/ling>)

<sup>2</sup> quoique malheureusement dispersés dans la littérature

## 2. Les bases de l'analyse séquentielle

### 2.1 Les chaînes de Markov

Soit une variable catégorielle  $X_t$ , prenant à chaque instant  $t$  une valeur dans l'ensemble fini  $A = \{a_1, a_2, \dots, a_m\}$ . Sous l'hypothèse d'indépendance des états successifs, on peut modéliser ce processus en spécifiant uniquement les probabilités  $P(a_i) := P(X = a_i)$  pour  $i = 1, \dots, m$ . En revanche, si l'on conçoit le futur  $X_{t+1}$  comme dépendant du présent  $X_t$ , on considérera des probabilités de transition de la forme:

$$p(a_i \rightarrow a_j) := P(X_{t+1} = a_j | X_t = a_i) \quad (1)$$

où l'on fait usage de l'hypothèse de stationnarité, selon laquelle les probabilités ne varient pas en fonction de  $t$ . On parlera alors d'une chaîne de Markov d'ordre 1, entièrement définie par la matrice de transition  $P$  de composantes  $P_{ij} = p(a_i \rightarrow a_j)$ . En généralisant ce qui précède, on peut construire des modèles d'ordre  $k$  plus élevé, où les composantes de  $P$  sont données par  $P_{ij} = p(\omega_i \rightarrow a_j)$ , avec  $\omega_i \in A^k$  (et  $A^k$  l'ensemble des  $k$ -grammes ou séquences de  $k$  modalités).

### 2.2 Estimation des paramètres du modèle

Pour aborder le problème du point de vue des données, considérons un texte de longueur  $n$  dont les symboles<sup>3</sup> (modalités) composent l'alphabet  $A = \{a_1, a_2, \dots, a_m\}$ . On peut estimer les probabilités  $P(a)$  d'un symbole  $a \in A$  par la fréquence empirique correspondante  $f(a) := n(a) / n$ , où  $n(a)$  dénote le nombre d'occurrences de  $a$  dans le texte; plus généralement, la probabilité d'un  $k$ -gramme  $\omega \in A^k$  peut être estimée par:

$$f(\omega) := \frac{n(\omega)}{(n-k)+1} \quad (2)$$

L'estimation de la probabilité de transition  $p(\omega \rightarrow a)$  que le symbole  $a \in A$  suive immédiatement le  $k$ -gramme  $\omega \in A^k$  est donnée par:

$$\hat{p}(\omega \rightarrow a) := \frac{n(\omega * a)}{\sum_{\tilde{a} \in A} n(\omega * \tilde{a})} \quad (3)$$

où le signe  $*$  représente l'opérateur de concaténation.

### 2.3 Le concept d'entropie

Soit la distribution des  $k$ -grammes ou séquences de  $k$  symboles  $\omega \in A^k$  pour notre variable  $X$ . L'entropie d'ordre  $k$  est une mesure de l'incertitude liée à cette distribution<sup>4</sup> (Shannon, 1948):

$$H_k = - \sum_{\omega \in A^k} P(\omega) \log P(\omega) \quad (4)$$

$H_k$  est minimale et nulle ssi l'un des  $k$ -grammes a une probabilité de 1. Elle est maximale (égale à  $\log |A^k|$  bits) ssi les  $k$ -grammes sont équiprobables. La quantité  $h_k := H_k - H_{k-1}$  pour  $k \geq 2$  (et  $h_1 = H_1$ ) est appelée entropie conditionnelle d'ordre  $k$  et s'interprète comme l'incertitude moyenne sur le  $k^{\text{e}}$  symbole d'une séquence étant donnés les  $k-1$  précédents. Enfin, la quantité

<sup>3</sup> Nous travaillerons ici au niveau du symbole, mais toutes nos considérations restent valables à celui du mot (Bavaud, 2000).

<sup>4</sup> Dans le présent document comme dans le logiciel que nous décrivons, nous utilisons par convention le logarithme en base 2, d'où une entropie exprimée en bits.

$d_k := h_k - h_{k+1}$ , définie pour  $k \geq 1$ , est appelée entropie résiduelle d'ordre  $k$  et s'interprète comme la réduction d'incertitude moyenne sur le  $k+1^{\text{è}}$  symbole d'une séquence étant donnés les  $k$  précédents plutôt que  $k-1$  seulement (Bavaud, 1998).

### 2.4 Le test de l'ordre du processus

Si les états successifs de  $X$  sont indépendants, on s'attend à ce que toutes les séquences de  $k$  symboles soient possibles ( $|A^k| = |A|^k = m^k$ ) et uniformément distribuées, d'où une entropie maximale à tous les ordres:  $H_k = \log m^k$  bits pour  $k = 1, 2, \dots$ . L'entropie conditionnelle sera constante, traduisant le fait que l'incertitude sur l'apparition d'un symbole est indépendante du contexte:  $h_k = \log m$  bits pour  $k = 1, 2, \dots$ . Il s'ensuit que l'entropie résiduelle sera constante et nulle, puisque aucune réduction d'incertitude n'est possible:  $d_k = 0$  pour  $k = 1, 2, \dots$ .

Si la probabilité d'un symbole est entièrement conditionnée par les  $r$  précédents (i.e. si le processus est d'ordre  $r$ ),  $H_k$  sera croissante jusqu'à  $k = r$ , puis constante:  $H_{r+1} = H_{r+2} = \dots = H_r$ ;  $h_k$  et  $d_k$  seront nulles pour  $k \geq r+1$ , puisque aucune incertitude ne subsiste dès lors que le  $r$ -gramme précédent est connu. C'est cette propriété de  $d_k$  qui fonde le test (itératif) de l'ordre du processus; on cherche à déterminer l'ordre  $k$  le plus élevé induisant une réduction d'incertitude  $d_k$  significative. En pratique, on fixe un seuil  $\alpha$  et l'on oppose:

$H_0(k)$ : "le processus est d'ordre  $k$ "

$H_1(k)$ : "le processus est d'ordre  $k+1$ "

On rejette  $H_0(k)$  au niveau  $\alpha$  si

$$2 \ln 2(n-k)d_{k+1} \geq \chi_{1-\alpha}^2 [m^k(m-1)^2] \quad (5)$$

où  $m$  représente la taille de l'alphabet, et  $n$  la longueur du texte utilisé pour l'estimation des paramètres du modèle<sup>5</sup>. On commence par  $k = 0$ , et le test est réitéré tant que  $k+1$  est petit relativement à  $\log n / \log m$  (Bavaud, 1998), que nous utilisons comme estimation de l'ordre à partir duquel l'effet de bord<sup>6</sup> commence à se manifester.

### 2.5 Exemple

On cherche à déterminer l'ordre du processus ayant généré la série (6) de longueur  $n = 100$ :

BACBCBCBAACBABCABCABCABCCBAABACCBAACABCABACBACCBAACCBACBACBCABCABACBCCABCCAB  
CCBACBAACBBCABCACBCABAABC

Comme on a  $\log n / \log m = 4.2$ , on ne fait l'inventaire des  $k$ -grammes que pour  $k \leq 4$ . Pour  $k = 1$ , on obtient la distribution suivante  $P(A) = 0.32$ ,  $P(B) = 0.32$ ,  $P(C) = 0.36$ , dont l'entropie vaut  $H_1 = -(0.32 \log 0.32 + 0.32 \log 0.32 + 0.36 \log 0.36) = 1.58$ . En poursuivant, on trouve que  $H_2 = 2.97$ ,  $H_3 = 4$  et  $H_4 = 4.86$ . On trouve également que  $h_1 = H_1 = 1.58$ ,  $h_2 = H_2 - H_1 = 1.39$ ,  $h_3 = 1.03$ ,  $h_4 = 0.86$ , et  $d_1 = h_1 - h_2 = 0.19$ ,  $d_2 = 0.36$ ,  $d_3 = 0.17$ . Pour le test de l'ordre du processus, on trouve au niveau  $\alpha = 0.01$ :

$k$	$2 \ln 2(n-k)d_{k+1}$	$\chi_{1-\alpha}^2 [m^k(m-1)^2]$
0	26.34	13.28
1	49.41	26.22
2	23.1	58.62

<sup>5</sup> Il est à noter que la présence du facteur  $\ln 2$  est liée à l'utilisation du logarithme binaire, et qu'il disparaîtrait si l'on travaillait en base  $e$ .

<sup>6</sup> L'inférence est d'autant plus fiable que le texte est grand et l'alphabet petit.

Donc, comme l'ordre  $k$  le plus élevé pour lequel nous pouvons rejeter  $H_0(k)$ : "le processus est d'ordre  $k$ " est  $k = 1$ , nous acceptons  $H_1(1)$ : "le processus est d'ordre 2" au niveau  $\alpha = 0.01$ ; en fait, la série a effectivement été générée par un processus d'ordre 2, sur la base de la matrice de transition  $P$  de composantes  $P_{ij} = p(\omega_i \rightarrow a_j)$ , avec  $a_j \in A = \{A, B, C\}$  et  $\omega_i \in A^2 = \{AA, AB, AC, BA, BB, BC, CA, CB, CC\}$  (dans l'ordre indiqué):

$$P = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/6 & 1/6 & 2/3 \\ 1/6 & 2/3 & 1/6 \\ 1/6 & 1/6 & 2/3 \\ 1/2 & 0 & 1/2 \\ 2/3 & 1/6 & 1/6 \\ 1/6 & 2/3 & 1/6 \\ 2/3 & 1/6 & 1/6 \\ 1/2 & 1/2 & 0 \end{pmatrix} \quad (7)$$

### 3. Présentation du logiciel

#### 3.1 Généralités

ENTROPIZER 1.1 est un programme d'analyse séquentielle pour Macintosh PowerPC. Pour un texte donné, il calcule automatiquement la fréquence empirique des  $k$ -grammes jusqu'à une taille maximale de 15 symboles et permet de visualiser les distributions correspondantes.

Le logiciel calcule les entropies de Shannon, conditionnelle et résiduelle de divers ordres ainsi que l'entropie de Rényi (Bavaud, 2000) de paramètre  $\alpha \in [0,1]$ , définie par:

$$H_k^\alpha := \frac{1}{1-\alpha} \log \sum_{\omega \in A^k} P(\omega)^\alpha \quad (8)$$

et la variété  $v_k$ , égale au nombre  $|A^k|$  de  $k$ -grammes différents observés à chaque ordre. Au travers des entropogrammes (Bavaud, 1998) ou graphes des entropies rapportées à l'ordre, il facilite la lecture des quantités mesurées et la mise en oeuvre du test de l'ordre du processus.

Enfin, le logiciel incorpore plusieurs fonctions d'exportation: texte simulé, distributions de  $k$ -grammes, matrices de transitions, etc.

#### 3.2 Distributions de $k$ -grammes et probabilités de transition

Comme on l'a vu plus haut, ces objets sont au centre de l'analyse séquentielle. ENTROPIZER 1.1 simplifie leur manipulation au moyen de deux listes associées à un paramètre de taille  $k$  et  $k'$  (voir figure 1 page suivante). La première liste affiche la distribution des  $k$ -grammes  $\omega \in A^k$  observés dans le texte<sup>7</sup>; la seconde affiche celle des  $k'$ -grammes pouvant suivre immédiatement la séquence sélectionnée sur la première. Pour chaque modalité de chaque distribution sont indiqués le nombre d'occurrences  $n(\omega)$  et la fréquence empirique  $f(\omega)$ .

Sur la figure 1,  $k$  et  $k'$  valent 1, définissant ainsi l'affichage de la distribution des séquences de  $k = 1$  symbole, donc des symboles  $a \in A$ , dans la première liste, et des symboles pouvant succéder à 'a' (sélectionné) dans la seconde. En dessous de chaque distribution figure une série de mesures s'y appliquant: la taille  $|A^k|$  de l'alphabet d'ordre  $k$  (types); le nombre de  $k$ -grammes observés (tokens), égal à  $(n - k) + 1$ , où  $n$  est la taille du texte; le rapport de ces deux quantités; enfin, l'entropie exprimée en bits. Dans cet exemple, on peut voir notamment que l'entropie sur

<sup>7</sup> Ici, un extrait de 5304 caractères de *Ulysses* de James Joyce en version originale (et recodé sur les 26 lettres de l'alphabet plus l'espace, le tiret et l'apostrophe).

les symboles pouvant succéder à 'd' est relativement faible, ce qu'explique la proportion élevée de l'espace '\_'  $p(d \rightarrow \_) = 0.69$ , et 'd' apparaît en cela comme ayant une tendance remarquable à terminer les mots (du moins dans *Ulysses*).

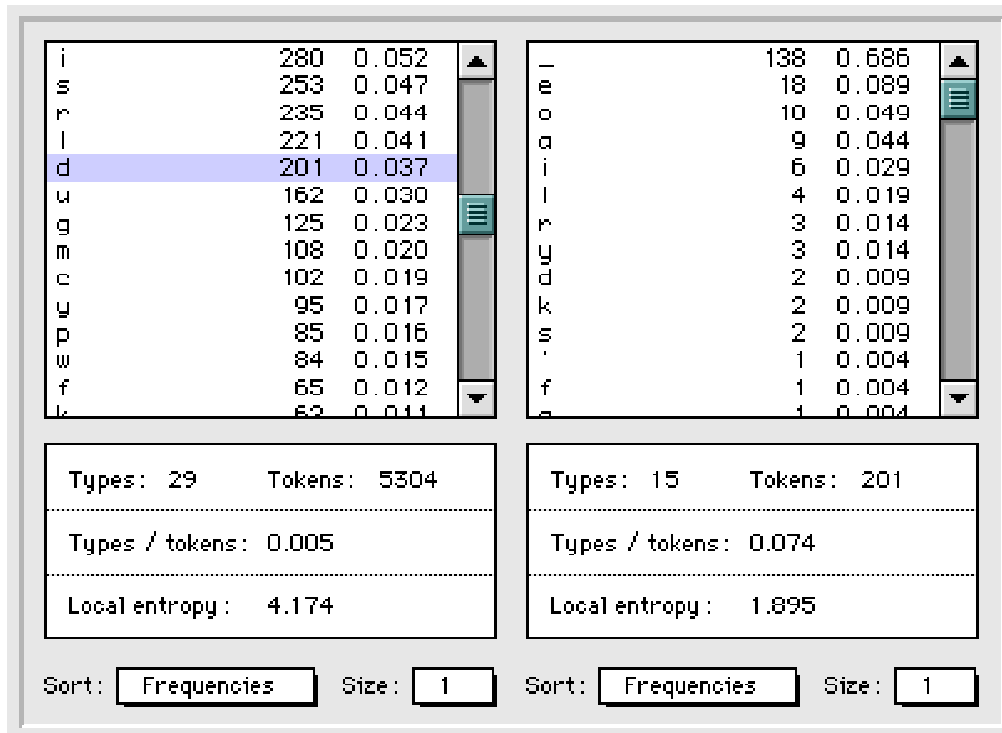


Figure 1: affichage des distributions de k-grammes

### 3.3 Entropies et test de l'ordre du processus

Le logiciel permet de visualiser graphiquement l'évolution (en fonction de l'ordre  $k$ ) des cinq indicateurs décrits précédemment: entropie (de Shannon), entropie conditionnelle et résiduelle, entropie de Rényi (de paramètre  $\alpha$ ) et variété. Par exemple, la figure 2 ci-dessous représente l'affichage de l'entropie résiduelle pour un texte de 5000 symboles généré avec la matrice de transition (7). La courbe discontinue représente le seuil de significativité de  $d_k$  au niveau  $\alpha = 0.01$ , dérivé de l'expression (5). Comme le texte est passablement plus grand que la série (6) utilisée plus haut, l'effet de bord ne se manifeste qu'à partir de  $k = 7$ , ce dont témoigne la barre grisée au bas du graphique; la significativité de  $d_2$  est également plus marquée que ne l'indiquaient les résultats précédents. Dans l'ensemble, ce second test confirme l'inférence déjà faite que le processus ayant généré les données est d'ordre 2.

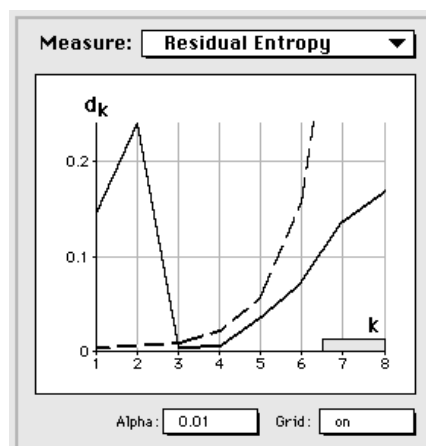


Figure 2: affichage de l'entropie résiduelle et test de l'ordre du processus

### 3.4 Simulation du processus

Nous avons vu comment construire un modèle séquentiel à partir de données textuelles. L'opération inverse, i.e. la génération de données sur la base d'un modèle est également prise en charge par Entropizer 1.1. Nous appelons simulation<sup>8</sup> (par une chaîne de Markov d'ordre  $k$ ) d'un processus de génération textuelle l'algorithme suivant:

1° Initialisation: sélection aléatoire d'un  $k$ -gramme  $\omega \in A^k$ , à partir de la distribution des  $k$ -grammes observés dans le texte;

2° Sélection aléatoire d'un symbole  $a \in A$ , à partir des probabilités de transition  $p(\omega \rightarrow a)$ , et concaténation avec le texte déjà généré, les  $k$  derniers symboles du texte résultant servant de contexte  $\omega$  pour la suite.

En répétant  $n$  fois 2°, on obtient un texte de taille  $n + k$ , que nous appelons texte simulé d'ordre  $k$ . A titre d'exemple, nous donnons ci-dessous quelques exemples de simulations de l'Anglais écrit pour  $k = 1, 2$  et  $3$ . Les modèles utilisés ont été estimés à partir d'un nouvel extrait d'*Ulysses*, de taille  $n = 38889$  (voir note 6 pour la norme de codage).

Pour  $k = 1$ , i.e. dans le cas où chaque symbole dépend du précédent, on a (9):

```
ene o y m doninocan's han lickndas s icken ofrd s fed teved f ind let cllly
g s owighe y celutowacke whegint thearathe ed aisoridimucond myod
leensetthe iere
```

Pour  $k = 2$ , i.e. dans le cas où chaque symbole dépend des deux précédents, on a (10):

```
id somin broming ton we an liganch the coure raying bo yours a cardeard did
cou by an oned iffew i do sing swither car a ving sin sphent ing eten bried
buts sho
```

Enfin<sup>9</sup>, pour  $k = 3$ , on a (11):

```
he towards head boreadful acrose rounger from it up he dang voice ans -- a
spoke blackerchs abouth remembermalach he door he what the lover a secread
had to his
```

### 3.5 Exemple d'application

Pour illustrer le fonctionnement du programme et en hommage à certain scientifique du passé, nous chercherons ici à modéliser et prédire la succession des voyelles et consonnes en Français. Etant donné un corpus de référence, la première étape est de le recoder suivant une norme prédéfinie<sup>10</sup>. Ainsi, à partir d'un essai de sémiotique narrative de longueur  $n = 13438$  caractères et dont est extraite la séquence (12):

```
[...]rtes de phrases ? L'affirmation, l'interrogation, le commandement peut-
être ? - Il en est d'in[...]
```

nous obtenons un texte de 12789 symboles qui se présente comme suit (13):

```
[...]ccvc cv ccvcvc cvccvcvcvc cvccvcvcvcvc cv cvccvcvcvcvcvc vccv vc
vc vcc cvc[...]
```

<sup>8</sup> par analogie avec les méthodes applicables aux variables quantitatives; on trouve aussi le terme d'approximation (Shannon, 1948).

<sup>9</sup> En principe, la taille du texte utilisé ne nous permet pas d'aller aussi loin:  $\log n / \log m = 3.14$ .

<sup>10</sup> En l'occurrence, le  $y$  est adjoint à l'ensemble des voyelles et le  $w$  à celui des consonnes. L'apostrophe est systématiquement supprimé et tous les autres symboles sont assimilés au séparateur (avec suppression des séparateurs consécutifs).

En l'ouvrant à l'aide d'ENTROPIZER 1.1, nous pouvons exporter les matrices de transition d'ordre  $k$  avec  $1 \leq k \leq 14$ . Par exemple, les transitions d'ordre 1 sont données par:

$$P = \begin{pmatrix} 0 & 0.78 & 0.22 \\ 0.19 & 0.22 & 0.59 \\ 0.21 & 0.61 & 0.18 \end{pmatrix} \quad (14)$$

avec  $P_{ij} = p(a_i \rightarrow a_j)$  et  $a_i, a_j \in A = \{_, c, v\}$  (dans cet ordre). Ce modèle ou un autre d'ordre supérieur peuvent être représentés graphiquement comme sur la figure 3 ci-dessous:

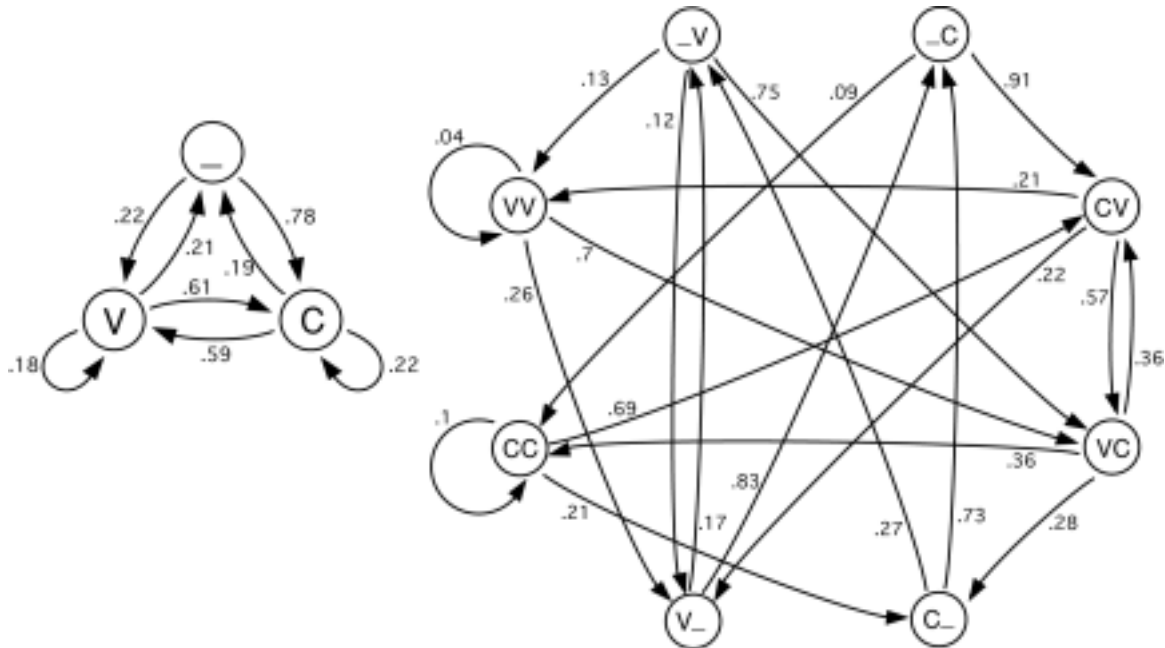


Figure 3: chaînes de Markov d'ordre 1 et 2 pour les voyelles et consonnes du Français

Pour prédire le prochain symbole de la séquence (13) on peut observer les probabilités de transition  $p(\omega_k \rightarrow a)$  où  $\omega_k \in A^k$  dénote le contexte (de taille  $k$ ) et  $a \in A$ , pour  $1 \leq k \leq 7$  (effet de bord:  $\log n / \log m = 8.63$ ). On obtient ainsi:

$k$	$\omega_k$	$p(\omega_k \rightarrow \_)$	$p(\omega_k \rightarrow c)$	$p(\omega_k \rightarrow v)$
1	c	0.19	0.22	<u>0.59</u>
2	vc	0.28	<u>0.36</u>	<u>0.36</u>
3	cvc	0.19	0.39	<u>0.42</u>
4	_cvc	0.2	<u>0.56</u>	0.24
5	c_cvc	0.23	<u>0.53</u>	0.24
6	cc_cvc	0.28	<u>0.51</u>	0.21
7	vcc_cvc	0.29	<u>0.49</u>	0.22

où les probabilités soulignées sont les plus élevées de leur distribution. Pour connaître le nombre moyen  $k$  de symboles dont devrait dépendre notre prédiction, on peut tester l'ordre du processus; comme on le voit sur la figure 4 (page suivante), on trouve ainsi que le processus est vraisemblablement d'ordre 4 (l'ordre 5 manque le coche, mais de peu), ce qui justifie la stabilisation de la valeur prédite pour  $k \geq 4$  dans le tableau ci-dessus.

Il est à noter que, dans le cas d'une variable à  $m = 3$  modalités, un modèle d'ordre  $k = 4$  correspond au plus à une matrice de  $m^k = 243$  probabilités de transition<sup>11</sup> et constitue à ce titre une représentation raisonnablement concise des phénomènes à l'étude.

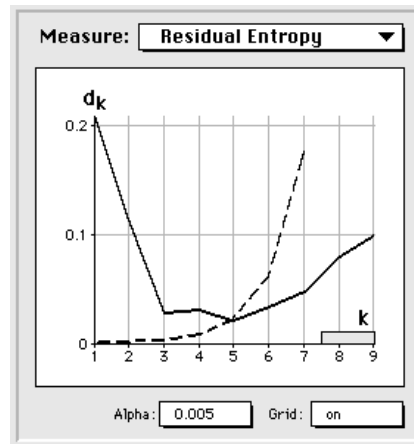


Figure 4: entropie résiduelle pour les voyelles et consonnes du Français

## 4. Conclusion

De la phonologie (Harris, 1951) à l'observation des séquences interactionnelles (Gottman et Roy, 1990), en passant par la cryptographie (Welsh, 1998), les disciplines susceptibles de tirer profit des méthodes de l'analyse séquentielle sont nombreuses. De fait, l'hétérogénéité des recherches entreprises et des savoirs convoqués a induit une forme d'éclatement théorique, que traduit le nombre restreint des ouvrages de synthèses. Parallèlement, la quasi absence d'outils informatiques *ad hoc* a passablement ralenti une recherche qui, si elle ne met pas en jeu des capacités d'abstraction surhumaines, implique un travail de comptage coûteux et minutieux.

Le développement d'ENTROPIZER 1.1 a été guidé par la volonté de combler une lacune dans l'arsenal des programmes d'analyse statistique, généralement mal équipés pour l'observation des séries temporelles catégorielles. Ainsi, nous souhaitons mettre à disposition des chercheurs un logiciel convivial intégrant les techniques de base de l'analyse séquentielle, et promouvoir à notre façon une approche qui nous semble moins connue et pratiquée qu'elle pourrait l'être.

## Références

- Bavaud F. (1998). *Modèles et Données*, l'Harmattan, Paris.
- Bavaud F. (2000). An Information Theoretical approach to Factor Analysis, *Proceedings of the 5th International Conference on the Statistical Analysis of Textual Data (JADT 2000)*.
- Damerau F. J. (1971). *Markov Models and Linguistic Theory*, Mouton, The Hague.
- Gottmann J. M. and Roy A. K. (1990). *Sequential Analysis*, Cambridge University Press, Cambridge.
- Harris Z. S. (1951). *Structural Linguistics*, University of Chicago Press, Chicago.
- Shannon C. E. (1948). A Mathematical Theory of Communication, *Bell Syst. tech. Journal* 27.
- Welsh D. (1988). *Codes and Cryptography*, Oxford Science Publications. □
- Xanthos A. (1999). *Entropizer 1.1*, Université de Lausanne.

<sup>11</sup> En fait, si l'on écarte les fréquences nulles, on obtient dans notre cas un modèle à 145 paramètres.