Une nouvelle heuristique pour l'alignement de motifs 2D par programmation dynamique

Émilie Chanoni[1], Thierry Lecroq[2], and Alexandre Pauchet[3]

[1] Université de Rouen, Psy.NCA EA 4306, 76821 Mont-Saint-Aignan Cedex, France,
Emilie.Chanoni@univ-rouen.fr

[2] Université de Rouen, LITIS EA 4108, 76821 Mont-Saint-Aignan Cedex, France,
Thierry.Lecroq@univ-rouen.fr,

[3] INSA-Rouen, LITIS EA 4108, BP 08 - 76801 Saint-Étienne-du-Rouvray, France,
Alexandre.Pauchet@insa-rouen.fr

Résumé : Nous présentons une nouvelle heuristique pour calculer des alignements globaux et locaux de motifs en deux dimensions par programmation dynamique. Cette méthode consiste en une généralisation des formules de récurrence utilisées pour les alignements de séquences. À notre connaissance, cette étude est la première qui considère le calcul d'alignements locaux de motifs 2D qui plus est avec des scores quelconques. Nous avons évalué cette méthode sur la recherche de motifs répétés sur des annotations de dialogues entre parents et enfants de quatre ans lors de la narration de deux histoires enfantines. Nous avons pu mettre en évidence un motif commun à plusieurs dialogues.

1 Introduction

La programmation dynamique est une technique utilisée dans de multiples applications, voir par exemple Lemaître *et al.* (2002); Hansen *et al.* (2004); Szer & Charpillet (2006); Sutton (1990). Les mots (séquences de symboles appartenant à un alphabet Σ) peuvent être facilement comparés à l'aide de cette technique. Des applications de ce problème incluent la bioinformatique, le traitement automatique de la langue naturelle ou bien le traitement de la parole. Les arbres peuvent aussi être comparés en utilisant des techniques similaires. Le lecteur peut se référer à Sankoff & Kruskal (1999) pour de plus amples détails. Il y a un grand nombre de travaux considérant la recherche exacte ou approchée en deux dimensions : étant donnés un motif X et un texte Y, tous les deux en deux dimensions, trouver les occurrences de X dans Y (voir Amir *et al.* (1994); Kärkkäinen & Ukkonen (1994); Amir & Farach (1997)). Il y a eu des efforts pour indexer des matrices en utilisant des arbres de suffixes ou des tables de suffixes (Giancarlo & Grossi, 1997; Kim *et al.*, 2003; Na *et al.*, 2007) mais ces index permettent de trouver des répétitions exactes et la généralisation pour trouver des motifs approchés est loin d'être immédiate. Très peu d'attention a été portée sur l'alignement de matrices à deux dimensions bien que les applications de ce problème incluent des domaines fondamentaux comme la fouille de donnés ou le traitement des images. Krithivasan & Sitalakshmi (1987) considère des motifs en deux dimensions de même taille alors que Baeza-Yates (1998) considère seulement l'alignement global de deux motifs en

deux dimensions. À notre connaissance, le présent article constitue la première tentative pour formaliser le calcul d'alignements locaux de deux motifs en deux dimensions par programmation dynamique. Récemment Arslan (2007) a considéré le problème de la recherche de plus long sous motifs communs en dimensions $d > 2$ mais en utilisant des techniques différentes.

Nous donnons des formules de récurrence pour calculer des alignements globaux et locaux de motifs en deux dimensions de taille respective M et N en temps et espace $O(M \times N)$. Pour cela nous devons précalculer les similarités entre tous les préfixes de toutes les lignes et tous les préfixes de toutes les colonnes des deux motifs. Cela peut également être effectué en temps et espace $O(M \times N)$.

Une première évaluation de cette méthode sur la recherche de motifs répétés sur des annotations de dialogues entre parents et enfants de quatre ans lors de la narration de deux histoires enfantines a été réalisée. Elle a déjà permis d'identifier un motif significatif dans ces dialogues.

Le reste de l'article est organisé comme suit : la section 2 rappelle les techniques de programmation dynamique pour l'alignement de séquences ; la section 3 présente la nouvelle méthode de programmation dynamique pour l'alignement de motifs en deux dimensions ; la section 4 montre l'évaluation réalisée, enfin la section 5 donne nos conclusions et des perspectives.

2 L'alignement de séquences

Les alignements sont généralement utilisés pour comparer les séquences. Ils sont largement répandus en bioinformatique. Ils constituent un moyen pour visualiser la ressemblance entre les séquences. Ils sont basés sur des notions de distance ou de similarité. Ils sont usuellement calculés par programmation dynamique. On considère deux types d'alignements de deux séquences x et y : les alignements globaux (qui considèrent la totalité des deux séquences) et les alignements locaux (qui permettent de détecter le segment de x le plus similaire à un segment de y).

Exemple

A	**C**	**G**	–	–	**A**
A	**T**	**G**	**C**	**T**	**A**

est un alignement global de **ACGA** et **ATGCTA**.

Une solution peut aussi être donnée sous forme de script d'édition comme suit :

Opération	séquence résultante
substitution de **A** par **A**	**A**
substitution de **C** par **T**	**AT**
substitution de **G** par **G**	**ATG**
insertion de **C**	**ATGC**
insertion de **T**	**ATGCT**
substitution de **A** par **A**	**ATGCTA**

2.1 Alignement global

Un alignement global de deux séquences x et y peut être obtenu en calculant la distance entre x et y. La notion de distance entre deux séquences est largement utilisée pour comparer des fichiers. La commande **diff** d'UNIX implante un algorithme basé sur cette notion où chaque ligne des fichiers est considérée comme un symbole. La sortie d'une comparaison des deux fichiers effectuée par **diff** est le nombre minimum

d'opérations d'édition (substitution, insertion ou suppression) pour transformer le premier fichier en le deuxième. Cette commande réalise un alignement global entre les deux fichiers.

On définit la distance d'édition entre deux séquences x et y comme étant le nombre minimum d'opérations d'édition élémentaires pour transformer x en y. Les opérations d'édition élémentaires sont : substitution d'un symbole de x par un symbole de y, suppression d'un symbole de x et insertion d'un symbole de y. Il est également possible de calculer des alignements globaux en utilisant des scores de similarités plutôt qu'une distance. Un score est associé à chaque opération d'édition élémentaire. Pour $a, b \in \Sigma$: ***Sub***(a, b) est le score de la substitution du symbole a par le symbole b, ***Del***(a) est le score de la suppression du symbole a, ***Ins***(a) est le score de l'insertion du symbole a. Pour deux symboles a et b, une valeur positive de ***Sub***(a, b) signifie que les deux symboles sont proches (ou similaires), et une valeur négative de ***Sub***(a, b) signifie que les deux symboles sont éloignés (ou dissimilaires). Nous pouvons maintenant définir le score d'édition généralisé entre deux séquences x et y par $d(x, y) = \max\{\text{score de } \gamma \mid \gamma \in \Gamma_{x,y}\}$ où $\Gamma_{x,y}$ est l'ensemble de toutes les séquences d'opérations d'édition qui transforment x en y, et le score d'un élément $\gamma \in \Gamma_{x,y}$ est la somme des scores de ses opérations d'édition élémentaires.

Pour calculer $d(x, y)$ pour deux séquences x et y de longueur respective m et n, on utilise une table à deux dimensions t à $m + 1$ lignes et $n + 1$ colonnes telle que $t[i, j] = d(x[0\,..\,i], y[0\,..\,j])$ pour $i = 0, \ldots, m - 1$ and $j = 0, \ldots, n - 1$. Il s'ensuit que $d(x, y) = t[m - 1, n - 1]$.

Les valeurs de la table t peuvent être calculées en suivant les formules de récurrence pour $i = 0, 1, \ldots, m - 1$ et $j = 0, 1, \ldots, n - 1$:

$$t[-1,-1] = 0, t[i,-1] = t[i-1,-1] + \textit{Del}(x[i]), t[-1,j] = t[-1,j-1] + \textit{Ins}(y[j]),$$

$$t[i,j] = \max \begin{cases} t[i-1,j-1] + \textit{Sub}(x[i], y[j]) \,, \\ t[i-1,j] + \textit{Del}(x[i]) \,, \\ t[i,j-1] + \textit{Ins}(y[j]) \,. \end{cases}$$

La valeur à la position (i, j) de la table t ne dépend que des valeurs aux trois positions voisines $(i - 1, j - 1)$, $(i - 1, j)$ et $(i, j - 1)$ (voir Crochemore *et al.* (2007)).

Une application directe de la formule de récurrence ci-dessus donne un algorithme en temps exponentiel pour calculer $t[m - 1, n - 1]$. Cependant la totalité de la table t peut être calculée en temps quadratique, par une technique connue sous le nom de « programmation dynamique ». C'est une technique générale qui peut être utilisée pour résoudre différents types d'alignements.

Un alignement optimal (de score maximal) peut alors être produit. Cela consiste à effectuer un tracé arrière du calcul des valeurs de la table t à partir de la position $(m - 1, n - 1)$ jusqu'à la position $(-1, -1)$.

2.2 Alignement local

Calculer un alignement local de deux séquences x et y consiste à trouver le segment de x qui est le plus similaire à un segment de y.

Le score local d'édition de deux séquences x et y est défini par :
$s(x, y) =$ similarité maximale entre un segment de x et un segment de y.

Pour calculer $s(x, y)$ pour deux séquences x et y de longueur respective m et n, nous utilisons une table à deux dimensions t_S à $m + 1$ lignes et $n + 1$ colonnes telle

que $t_S[i,j] = \max\{s(x[\ell\,..\,i], y[k\,..\,j]) \mid 0 \leq \ell \leq i \text{ et } 0 \leq k \leq j\} \cup \{0\}$, pour $i = 0, \ldots, m-1$ et $j = 0, \ldots, n-1$. Donc $s(x,y) =$ valeur maximale dans t_S.

Les valeurs de la table t_S peuvent être calculées avec les formules de récurrence suivantes : pour $i = 0, 1, \ldots, m-1$ et $j = 0, 1, \ldots, n-1$:

$$t_S[-1,-1] = t_S[i,-1] = t_S[-1,j] = 0$$

$$t_S[i,j] = \max \begin{cases} t_S[i-1,j-1] + \mathit{Sub}(x[i], y[j])\,, \\ t_S[i-1,j] + \mathit{Del}(x[i])\,, \\ t_S[i,j-1] + \mathit{Ins}(y[j])\,, \\ 0\,. \end{cases}$$

Calculer les valeurs de t_S pour un alignement local de x et y peut être fait en temps et espace $O(mn)$. Retrouver un alignement local peut être fait d'une manière similaire au cas des alignements globaux à ceci près que le tracé arrière doit commencer à une position d'une valeur maximale dans t_S plutôt qu'à la position $(m-1, n-1)$.

3 Alignement de motifs 2D

Considérons maintenant deux motifs rectangulaires $X = X[0\,..\,m_1-1, 0\,..\,n_1-1]$ et $Y = Y[0\,..\,m_2-1, 0\,..\,n_2-1]$, de taille respective $M = m_1 \times n_1$ et $N = m_2 \times n_2$. Chaque élément $X[i,j]$ avec $0 \leq i \leq m_1 - 1$ et $0 \leq j \leq n_1 - 1$ et $Y[k,\ell]$ avec $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$ appartient à l'alphabet Σ.

Nous voulons maintenant calculer le score maximum des opérations d'insertion, de suppression ou de substitution de symboles individuels pour transformer X en Y. Les symboles peuvent être insérés, supprimés ou substitués séparément ou par portion de lignes ou par portion de colonnes.

3.1 Alignement global

Aligner X et Y en utilisant la programmation dynamique consiste à généraliser les formules de récurrence utilisées pour l'alignement de séquences.

Pour cela nous avons besoin de quatre tables à quatre dimensions D_R, D_C, I_R et I_C définies comme suit pour $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq i \leq m_2 - 1$ et $0 \leq j \leq n_2 - 1$:

$$D_R[i,j] = \sum_{p=0}^{j} \mathit{Del}(X[i,p]) \qquad \text{et} \qquad D_C[i,j] = \sum_{p=0}^{i} \mathit{Del}(X[p,j])$$

$$I_R[i,j] = \sum_{p=0}^{j} \mathit{Ins}(Y[i,p]) \qquad \text{et} \qquad I_C[i,j] = \sum_{p=0}^{i} \mathit{Ins}(Y[p,j]).$$

En d'autres termes, $D_R[i,j]$ est le score de la suppression du préfixe de longueur $j+1$ de la ligne i de X, $D_C[i,j]$ est le score de la suppression du préfixe de longueur $i+1$ de la colonne j de X, $I_R[i,j]$ est le score de l'insertion du préfixe de longueur $j+1$ de la ligne i de Y et $I_C[i,j]$ est le score de l'insertion du préfixe de longueur $i+1$ de la ligne j de Y. Les tables D_R et D_C peuvent être calculées en temps et espace $O(m_1 \times n_1)$. Les tables I_R et I_C peuvent être calculées en temps et espace $O(m_2 \times n_2)$.

Nous allons aussi utiliser deux tables à quatre dimensions R et C de taille $m_1 \times n_1 \times m_2 \times n_2$ definies comme suit : $R[i,j,k,\ell] = d(X[i,0\,..\,j], Y[k,0\,..\,\ell])$, et

$C[i,j,k,\ell] = d(X[0\,..\,i,j], Y[0\,..\,k,\ell])$. En fait $R[i,j,k,\ell]$ contient la distance entre le préfixe de longueur $j+1$ de la ligne i de X et le préfixe de longueur $\ell+1$ de la ligne k de Y. De manière similaire $C[i,j,k,\ell]$ contient la distance entre le préfixe de longueur $i+1$ de la colonne j de X et le préfixe de longueur $k+1$ de la colonne ℓ de Y. Les deux tables R et C peuvent être calculées en temps et espace $O(m_1 \times n_1 \times m_2 \times n_2)$.

Nous utilisons alors une table à quatre dimensions T de taille $(m_1+1) \times (n_1+1) \times (m_2+1) \times (n_2+1)$ définie comme suit : $T[i,j,k,\ell] = \max\{\text{score de } \gamma \mid \gamma \in \Gamma_{X,Y}\}$ où $\Gamma_{X,Y}$ est l'ensemble de toutes les séquences d'opérations d'édition qui transforment X en Y, et le score d'un élément $\gamma \in \Gamma_{x,y}$ est la somme des scores de ses opérations d'édition élémentaires.

Les valeurs de la table T peuvent être calculées comme suit pour $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$ (voir Fig. 1) :

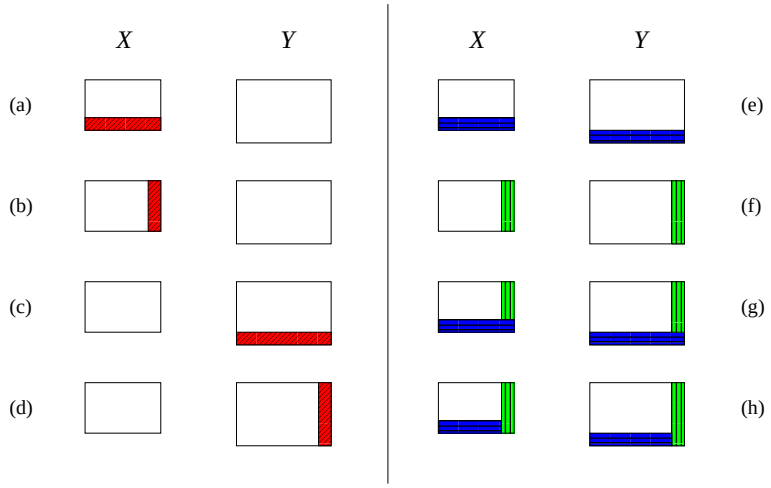

FIG. 1 – (a) : suppression de $X[i,0\,..\,j]$; (b) : suppression de $X[0\,..\,i,j]$; (c) : insertion de $Y[k,0\,..\,\ell]$; (d) : insertion de $Y[0\,..\,k,\ell]$; (e) : substitution de $X[i,0\,..\,j]$ par $Y[k,0\,..\,\ell]$; (f) : substitution de $X[0\,..\,i,j]$ par $Y[0\,..\,k,\ell]$; (g) : substitution de $X[i,0\,..\,j]$ par $Y[k,0\,..\,\ell]$ et substitution de $X[0\,..\,i-1,j]$ par $Y[0\,..\,k-1,\ell]$; (h) : substitution de $X[i,0\,..\,j-1]$ par $Y[k,0\,..\,\ell-1]$ et substitution de $X[0\,..\,i,j]$ par $Y[0\,..\,k,\ell]$.

$$T[i,j,k,\ell] = \max \begin{cases} T[i-1,j,k,\ell] + D_R[X[i,0\,..\,j]] \\ T[i,j-1,k,\ell] + D_C[X[0\,..\,i,j]] \\ T[i,j,k-1,\ell] + I_R[Y[k,0\,..\,\ell]] \\ T[i,j,k,\ell-1] + I_C[Y[0\,..\,k,\ell]] \\ T[i-1,j,k-1,\ell] + R[i,j,k,\ell] \\ T[i,j-1,k,\ell-1] + C[i,j,k,\ell] \\ T[i-1,j-1,k-1,\ell-1] + C[i-1,j,k-1,\ell] + R[i,j,k,\ell] \\ T[i-1,j-1,k-1,\ell-1] + C[i,j,k,\ell] + R[i,j-1,k,\ell-1] \end{cases} \quad (1)$$

Nous avons les initialisations marginales suivantes : $T[-1,j,k,\ell] = T[i,-1,k,\ell] = (k+1)\times(\ell+1)$ et $T[i,j,-1,\ell] = T[i,j,k,-1] = (i+1)\times(j+1)$ pour $0 \leq i \leq m_1-1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$.

Ensuite un tracé arrière peut être effectué à partir de $T[m_1, n_1, m_2, n_2]$ comme dans le cas de l'alignement de séquences.

L'alignement global de deux motifs rectangulaires de taille respective $M = m_1 \times n_1$ et $N = m_2 \times n_2$ peut être effectué en temps et espace $O(M \times N)$.

Exemple

Avec les motifs suivants

X	0	1	2
0	**A**	**B**	**C**
1	**D**	**E**	**F**
2	**G**	**H**	**I**
3	**J**	**K**	**L**

et

Y	0	1
0	**E**	**C**
1	**H**	**I**
2	**K**	**L**

et en considérant les scores suivants pour les opérations d'édition (*Ins*$(a) =$ *Del*$(a) = -1$, *Sub*$(a, a) = 2$ et *Sub*$(a, b) = -1$ pour $a, b \in \Sigma$ tels que $a \neq b$), un script possible en remontant depuis $T[3, 2, 2, 1] = 6$ est :

Opération	Score
Substitution de $X[0\,..\,3, 2]$ par $Y[0\,..\,2, 1]$	5
Substitution de $X[0\,..\,3, 1]$ par $Y[0\,..\,2, 0]$	5
Suppression de $X[0\,..\,3, 0]$	-4

3.2 Alignement local

Pour calculer un alignement local entre deux motifs rectangulaires X et Y on utilise deux tables à quatre dimensions R_S et C_S de taille $m_1 \times n_1 \times m_2 \times n_2$ définies comme suit :

$R_S[i, j, k, \ell] = s(X[i, 0\,..\,j], Y[k, 0\,..\,\ell])$ et $C_S[i, j, k, \ell] = s(X[0\,..\,i, j], Y[0\,..\,k, \ell])$.

En fait, $R_S[i, j, k, \ell]$ contient la similarité maximale entre un suffixe de $X[i, 0\,..\,j]$ et un suffixe de $Y[k, 0\,..\,\ell]$. De manière similaire $C_S[i, j, k, \ell]$ contient la similarité maximale entre un suffixe de $X[0\,..\,i, j]$ et un suffixe de $Y[0\,..\,k, \ell]$. Ces deux tables peuvent être calculées en utilisant la formule de récurrence habituelle pour les séquences en temps et espace $O(m_1 \times n_1 \times m_2 \times n_2)$.

Ensuite on utilise une table à quatre dimensions T_S de taille $(m_1 + 1) \times (n_1 + 1) \times (m_2 + 1) \times (n_2 + 1)$ définie comme suit :

$$T_S[i, j, k, \ell] = \max \quad \{s(X[i'\,..\,i, j'\,..\,j], Y[k'\,..\,k, \ell'\,..\,\ell]) \mid 0 \leq i' \leq i, 0 \leq j' \leq j, 0 \leq k' \leq k \text{ et } 0 \leq \ell' \leq \ell\} \cup \{0\}.$$

On note : $r = R_S[i, j, k, \ell]$, $c = C_S[i, j, k, \ell]$, $r' = R_S[i, j-1, k, \ell-1]$, $c' = C_S[i-1, j, k-1, \ell]$ et $q =$ *Del*$(X[i, j])$ + *Ins*$(Y[k, \ell])$.

Les valeurs de la table T_S peuvent être calculées à l'aide de l'équation ci-dessous.

$$T_S[i, j, k, \ell] = \max \begin{cases} T_S[i-1, j, k, \ell] + \textit{Del}(X[i, j]) \\ T_S[i, j-1, k, \ell] + \textit{Del}(X[i, j]) \\ T_S[i, j, k-1, \ell] + \textit{Ins}(Y[k, \ell]) \\ T_S[i, j, k, \ell-1] + \textit{Ins}(Y[k, \ell]) \\ T_S[i-1, j, k-1, \ell] + (r \text{ si } r \neq 0 \text{ sinon } q) \\ T_S[i, j-1, k, \ell-1] + (c \text{ si } c \neq 0 \text{ sinon } q) \\ T_S[i-1, j-1, k-1, \ell-1] + (c' + r \text{ si } c', r \neq 0 \text{ sinon } q) \\ T_S[i-1, j-1, k-1, \ell-1] + (c + r' \text{ si } c, r' \neq 0 \text{ sinon } q) \\ 0 \end{cases} \quad (2)$$

pour $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$.

Nous avons les initialisations marginales suivantes :

$$T_S[-1, j, k, \ell] = T_S[i, -1, k, \ell] = T_S[i, j, -1, \ell] = T_S[i, j, k, -1] = 0$$

pour $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$.

Les valeurs de la table T_S de taille $(m_1 + 1) \times (n_1 + 1) \times (m_2 + 1) \times (n_2 + 1)$ permettant de détecter des motifs similaires peuvent être calculées en temps et espace $O(m_1 \times n_1 \times m_2 \times n_2)$.

La procédure de tracé arrière doit commencer à la position contenant une valeur maximale plutôt qu'à la position $(m_1 - 1, n_1 - 1, m_2 - 1, n_2 - 1)$.

Exemple

Avec le système de score suivant : *Sub*$(a, a) = 1$, *Sub*$(a, b) = -1$ et *Ins*$(a) =$ *Del*$(a) = -1$, pour $a, b \in \Sigma$ tels que $a \neq b$, la Fig. 2 montre sur la gauche les deux motifs à aligner et sur la droite le meilleur alignement local où le symbole **c** sur la troisième ligne du second motif est inséré, tandis que les symboles **ghi** de la troisième ligne du premier motif sont substitués par les symboles **aba** de la quatrième ligne du second motif.

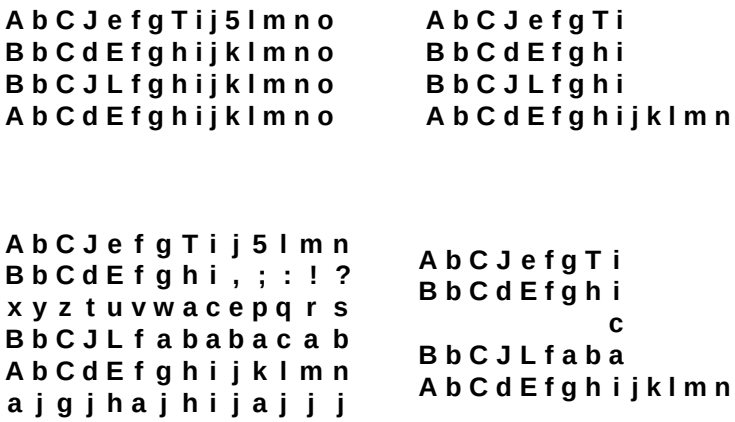

FIG. 2 – Alignement local de motifs 2D.

4 Applications

L'étude de fichiers log et de corpus de dialogues annotés vise la plupart du temps à repérer des comportements récurrents significatifs, afin de concevoir des outils de contrôle du système ou d'aide à la décision. Ces comportements apparaissent alors sous la forme de motifs répétés, représentés par des automates (Winograd & Flores, 1986; Bouzouba & Moulin, 2001), d'automates temporisés (Pauchet, 2006), de réseaux de Petri (Mazouzi, 2001), de diagrammes de séquences (FIPA, 2002), *etc.* Le point commun de toutes ces représentations étant qu'elles s'appuient sur des motifs linéaires.

Dans le cas particulier de corpus de dialogues dont les annotations visent à caractériser un processus émotionnel, chaque énoncé doit être annoté par une série de termes. Il en résulte une recherche de répétitions non plus linéaires, mais en deux dimensions, rendant impossible les approches classiques des études de corpus. L'approche, présentée ici, de recherche d'alignements de motifs 2D répond tout à fait à cette problématique. Il s'agit d'une démarche en trois étapes :

1. identification d'alignements de motifs en 2D grâce à l'algorithme présenté précédemment ;

2. évaluation de la pertinence sémantique des motifs repérés (expertise) ;
3. évaluation statistique de ces motifs.

L'algorithme a été testé sur des dialogues entre parents et enfants de quatre ans lors de la narration de deux histoires enfantines (A et B). Chaque énoncé retranscrit et codé selon la grille de Chanoni (2004) correspond à une ligne. Cette grille a été utilisée dans le cadre d'une étude psychologique portant sur les interprétations mentalistes des comportements d'un personnage. L'objectif étant de repérer les caractéristiques dialogiques, sémantiques et pragmatiques du discours que proposent les adultes pour expliquer le comportement d'un personnage soumis à une fausse croyance (Wimmer & Perner, 1983). Cette grille est composée de cinq colonnes, chaque colonne comporte entre 2 et 7 codages (6, 3, 7, 2, 2) : les indices relevés dans la première colonne concernent la nature de l'énoncé. Il peut s'agir d'une affirmation (a ou A), d'une question (q ou Q), d'une demande d'attention sur l'histoire (AH), d'une demande d'attention générale (AG). La deuxième colonne concerne la référenciation de l'énoncé. L'énoncé peut se référer au personnage (P), à l'interlocuteur (HR) ou au locuteur (AR). La troisième colonne relève les états mentaux. Les partenaires du dialogue peuvent décrire l'émotion (E), la volition (V), la cognition observable ou non observable (CO ou CNO), l'épistémie (EP), l'hypothèse (HY) et la surprise (S). Ici la surprise est distinguée des émotions pour ses liens avec la croyance dont elle est issue. Les deux dernières colonnes sont consacrées aux justifications par cause / conséquence (C) ou par opposition (O) qui peuvent être appliquées soit pour expliquer l'histoire (CH), soit pour expliquer une situation par l'évocation d'un contexte personnel (CP). Une matrice de substitution a été spécifiquement construite pour cette application.

Après avoir réalisé les alignements de toutes les paires de dialogues et les alignements intra-dialogues, nous nous sommes focalisés sur trois dialogues. Deux des dialogues concerne l'histoire A et un autre l'histoire B, les partenaires de dialogues sont tous différents. Cette technique nous a permis de repérer des sous-motifs approchés (ou flous) se répétant à l'intérieur d'un même dialogue ainsi qu'au sein des deux autres. Sur des séries de 12, 13 et 16 lignes (selon les localisations) nous observons la répétition d'un motif qui se décompose en deux parties : le premier ensemble est composé d'une série de lignes (de 9 à 14) qui comportent presque des indices relatant la présence d'un état mental et d'une référenciation. Ce premier ensemble est suivi d'une série de deux lignes qui comporte chacune une justification associée à un état mental.

L'intérêt de ce sous-motif réside dans le fait qu'il semble caractériser à la fois une organisation dialogique et un contenu : fait délicat voire impossible à observer « à l'œil nu ». Pour cet exemple, il apparaît que les parents prennent soin de préparer la justification « mentaliste » de la situation. En effet, dans un premier temps, ils semblent décrire et préciser les états mentaux en jeu, puis dans un deuxième temps, ils précisent les liens de cause, de conséquence ou d'opposition qui expliquent les états mentaux du personnage et ses comportements.

À terme, tous les motifs détectés seront répertoriés et recherchés dans tous les dialogues. Ainsi pour chaque symbole de tous les dialogues, il sera possible de savoir dans quel motif il apparaît et quels sont les dialogues (avec les positions) dans lesquels ces motifs apparaissent. Le but étant d'aider un expert à détecter des motifs significatifs.

5 Conclusion et perspectives

Nous avons présenté une méthode très générique pour aligner des motifs en deux dimensions par programmation dynamique en temps et espace quadratique. Cette métho-

de permet de calculer des alignements globaux et locaux. Une première difficulté consiste à visualiser les alignements ainsi calculés. Beaucoup de questions peuvent être posées. Est-il possible d'aligner des motifs 2D en temps sous-quadratique comme c'est le cas pour les séquences ? (voir Crochemore *et al.* (2003)). Est-il possible de développer une heuristique similaire à BLAST (Altschul *et al.*, 1990) pour rapidement aligner un motif 2D contre une banque de motifs 2D ? D'autre part, il est nécessaire d'accompagner la méthode proposée d'une validation statistique pour estimer la pertinence des résultats trouvés.

Au-delà de l'intérêt scientifique et théorique, cette méthode peut à terme, non seulement fournir un outil d'aide au diagnostic dans le cadre de situations dialogiques asymétriques mais pourrait plus généralement fournir un outil à la formation à l'entretien.

En effet, lors d'une interaction des émotions non perceptibles par des termes explicites atteignent tout de même la subjectivité de l'interlocuteur, ces émotions sont indispensables à prendre en compte pour une bonne qualité de dialogue, elles le sont d'autant plus lors d'interactions visant à un diagnostic.

Imaginons un jeune enfant qui vient consulter pour une douleur à l'abdomen. L'enfant et le médecin n'ont ni le même niveau de langage, ni le même niveau de représentation du corps et de la douleur. Pourtant le diagnostic ne se fera que sur la base des propos que l'enfant aura eu à propos de sa douleur. Cette méthode en complément d'un entretien semi dirigé pourrait permettre de saisir des informations non saisissables d'emblée et pourtant nécessaires au diagnostic.

Par ailleurs, cette méthode accompagnée d'une expertise psychologique permet de relever les techniques ou les patterns (dialogiques ou non) efficaces. Ainsi, la formation aux techniques d'entretien pourrait s'en trouver améliorée. Nous pourrions penser qu'un jour cette méthode pourrait assister en temps réel l'apprenant dans sa formation à l'entretien.

Références

ALTSCHUL S. F., GISH W., MILLER W., MYERS E. W. & LIPMAN D. J. (1990). A basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

AMIR A., BENSON G. & FARACH M. (1994). An alphabet independent approach to 2-dimensional pattern matching. *SIAM J. Comput.*, **23**(2), 313–323.

AMIR A. & FARACH M. (1997). Two dimensional matching. In A. APOSTOLICO & Z. GALIL, Eds., *Pattern Matching Algorithms*, chapter 9, p. 267–292. Oxford University Press.

ARSLAN A. N. (2007). A largest common d-dimensional subsequence of two d-dimensional strings. In *FCT*, volume 4639 of *LNCS*, p. 40–51.

BAEZA-YATES R. A. (1998). Similarity in two-dimensional strings. In *COCOON*, volume 1449 of *LNCS*, p. 319–328.

BOUZOUBA K. & MOULIN B. (2001). Les interactions distribuées : Approche basée sur le point de vue personnel des agents. In *MFI*, p. 15–30.

CHANONI E. (2004). *Rôle du langage dans le développement de la théorie de lesprit chez les enfants de 3 à 5 ans : contexte verbal et contexte narratif*. PhD thesis, Presse Universitaire de Lille.

CROCHEMORE M., HANCART C. & LECROQ T. (2007). *Algorithms on strings*. Cambridge University Press.

CROCHEMORE M., LANDAU G. M. & ZIV-UKELSON M. (2003). A subquadratic sequence alignment algorithm for unrestricted scoring matrices. *SIAM J. Comput.*, **32**(6), 1654–1673.

GIANCARLO R. & GROSSI R. (1997). Suffix tree data structures for matrices. In A. APOSTOLICO & Z. GALIL, Eds., *Pattern matching algorithms*, chapter 10, p. 293–340. Oxford University Press.

HANSEN E., BERNSTEIN D. & ZILBERSTEIN S. (2004). Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on AI*, p. 709–715.

KÄRKKÄINEN J. & UKKONEN E. (1994). Two and higher dimensional pattern matching in optimal expected time. In *SODA*, p. 715–723.

KIM D. K., KIM Y. A. & PARK K. (2003). Generalizations of suffix arrays to multidimensional matrices. *Theoret. Comput. Sci.*, **302**(1–3), 223–238.

KRITHIVASAN K. & SITALAKSHMI R. (1987). Efficient two-dimensional pattern matching in the presence of errors. *Inf. Sci.*, **43**, 169–184.

LEMAÎTRE M., VERFAILLIE G., JOUHAUD F., LACHIVER J.-M. & BATAILLE N. (2002). Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, **6**(5), 367–381.

MAZOUZI H. (2001). *Ingénierie des protocoles d'interaction : des systèmes distribués aux systèmes multi-agents*. PhD thesis, Université Paris IX.

NA J. C., GIANCARLO R. & PARK K. (2007). On-line construction of two-dimensional suffix trees in $O(n^2 \log n)$ time. *Algorithmica*, **48**(2), 173–186.

PAUCHET A. (2006). *Modélisation cognitive d'interactions humaines dans une cadre de planification multi-agents*. PhD thesis, Université Paris Nord.

SANKOFF D. & KRUSKAL J. B. (1999). *Time warps, string edits, and macromolecules : the theory and practice of sequence comparison*. Cambridge University Press. 2nd edition.

SUTTON R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the 7th International Conference on Machine Learning*, p. 216–224.

SZER D. & CHARPILLET F. (2006). Point-based dynamic programming for DEC-POMDPs. In *Proceedings of the 21st National Conference on AI*.

FIPA (2002). FIPA *Request Interaction Protocol Specification*. Rapport interne, Foundation for Intelligent Physical Agents. Avail. at `http://www.fipa.org`.

WIMMER H. & PERNER J. (1983). Beliefs about beliefs : representation and constraining function of the wrong beliefs in young childrens understandig of deception. *Cognition*, **13**, 103–128.

WINOGRAD T. & FLORES F. (1986). *Understanding Computers and Cognition : A New Foundation for Design*. Ablex.