

Extraction de motifs dans des dialogues annotés par programmation dynamique

Émilie Chanoni*

Emilie.Chanoni@univ-rouen.fr

Thierry Lecroq†

Thierry.Lecroq@univ-rouen.fr

Alexandre Pauchet‡

Alexandre.Pauchet@insa-rouen.fr

* Université de Rouen, Psy.NCA EA 4306,
76821 Mont-Saint-Aignan Cedex, France

† Université de Rouen, LITIS EA 4108,
76821 Mont-Saint-Aignan Cedex, France

‡ INSA-Rouen, LITIS EA 4108,
BP 08 - 76801 Saint-Étienne-du-Rouvray, France

Résumé :

Dans le cadre de la modélisation du dialogue, nous nous intéressons à la recherche de régularités dans les échanges entre interlocuteurs, et plus particulièrement dans l'expression d'un processus mentaliste. Nous utilisons pour cela une grille mentaliste afin de coder l'expression des états mentaux et des émotions dans les énoncés.

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. A 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.

Mots-clés : Dialogue annoté, extraction de motifs, identification de régularités, programmation dynamique

1 Introduction

La conception de systèmes informatiques interagissant efficacement avec l'utilisateur nécessite l'utilisation de modèles robustes et efficaces. Ces modèles doivent être définis non pas arbitrairement par le concepteur du système mais en s'appuyant sur une étude des mécanismes humains d'interaction et de communication. Par exemple, pour qu'une interface dialogique en langue naturelle soit agréable, elle ne doit pas s'appuyer uniquement sur un modèle de la tâche mais intégrer en sus un comportement langagier le plus naturel possible.

Il existe de nombreux modèles de dialogue, parmi lesquels on peut distinguer, comme le fait Lehuen [20], plusieurs approches distinctes :

- L'approche par la planification est fondée sur la reconnaissance de buts et la construction de plans dans les conversations. On citera comme exemples les modèles d'Allen et Per-

rault [1], de Balkanski et Hurault-Plantet [6] ou le système Artimis de Sadek [27].

- L'approche par la structuration cherche à dégager et à expliquer la structure des dialogues en décomposant les interactions verbales en unités de différents niveaux. Le modèle genevois de Roulet et Moeschler [26, 23] ou la théorie discursive de Vanderveken [33], par exemple, définissent les règles conversationnelles régissant ces unités.
- L'approche par l'interaction s'appuie sur des règles locales pour l'interprétation et la production d'actions communicatives durant les dialogues. Les exemples les plus connus sont les modèles de Trognon et Brassac [32], de Dessalles [14] et de Brassac et Pesty [8].

Il existe bien évidemment d'autres approches, ainsi que des approches mixtes, comme le modèle Coala [20], fondé sur la structuration et sur l'interaction.

La modélisation du dialogue par les approches citées précédemment s'appuie la plupart du temps sur un corpus de dialogues annotés, dont l'étude vise à repérer un ensemble de comportements récurrents significatifs, afin d'être en mesure de les expliquer. Lorsque les annotations sont judicieuses, ces comportements apparaissent alors sous la forme de motifs répétés dans les annotations. La représentation de ces répétitions peut se faire sous la forme d'automates [35, 7], d'automates temporisés [25], de réseaux de Petri [22], de diagrammes de séquences [31], *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 mentaliste, chaque énoncé doit être annoté par une série de codes. 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 répond tout à fait à cette problématique :

1. retranscription et annotation des dialogues ;
2. identification d'alignements de motifs en 2D par programmation dynamique ;
3. évaluation de la pertinence sémantique des motifs repérés (expertise) ;

La programmation dynamique [11] est une technique utilisée dans de multiples applications, voir par exemple [21, 16, 30, 29]. 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 [28] pour de plus amples détails. Il y a un grand nombre de travaux considérant la recherche exacte ou approchée en deux dimensions : étant donné un motif X et un texte Y , tous les deux en deux dimensions, trouver les occurrences de X dans Y (voir [3, 17]). Il y a eu des efforts pour indexer des matrices en utilisant des arbres de suffixes ou des tables de suffixes [15, 18, 24] 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 données ou le traitement des images. [19] considère des motifs en deux dimensions de même taille alors que [5] considère seulement l'alignement global de deux motifs en deux dimensions. A 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 [4] 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 $O(M \times N)$.

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.

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 point 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

colonnes telle que $t[i, j] = d(x[0..i], y[0..j])$ pour $i = 0, \dots, m-1$ and $j = 0, \dots, n-1$. 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 $i = 0, 1, \dots, m-1$ et $j = 0, 1, \dots, n-1$:

$$t[-1, -1] = 0,$$

$$t[i, -1] = t[i-1, -1] + Del(x[i]),$$

$$t[-1, j] = t[-1, j-1] + Ins(y[j]),$$

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

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 [12]).

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 $s(x, y) = \text{similarité maximale entre un segment de } x \text{ et un segment de } y$.

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 $i = 0, \dots, m-1$ et $j = 0, \dots, n-1$. Donc $s(x, y) =$

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

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

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

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 deux dimensions D_R, D_C, I_R et I_C définies comme suit pour $0 \leq i \leq m_1-1, 0 \leq j \leq n_1-1, 0 \leq k \leq m_2-1$ et $0 \leq \ell \leq n_2-1$:

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

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

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

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

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 $m_1 \times n_1 \times m_2 \times n_2$ définies comme suit : $R[i, j, k, \ell] = d(X[i, 0..j], Y[k, 0..l])$, et $C[i, j, k, \ell] = d(X[0..i, j], Y[0..k, l])$. 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 pour $0 \leq i \leq m_1-1, 0 \leq j \leq n_1-1, 0 \leq k \leq m_2-1$ et $0 \leq \ell \leq n_2-1$ comme présenté FIG. 1 et FIG. 2.

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 $O(M \times N)$.

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

FIG. 1 – Calcul de T , avec $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$

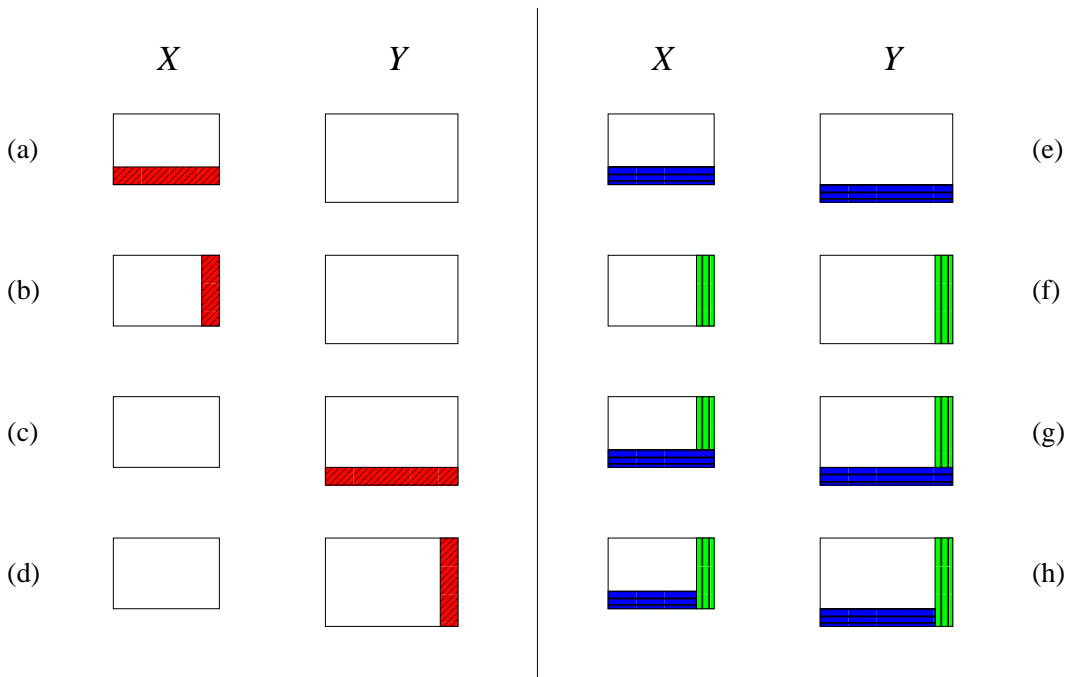


FIG. 2 – (a) : suppression de $X[i, 0..j]$; (b) : suppression de $X[0..i, j]$; (c) : insertion de $Y[k, 0.. \ell]$; (d) : insertion de $Y[0..k, \ell]$; (e) : substitution de $X[i, 0..j]$ par $Y[k, 0.. \ell]$; (f) : substitution de $X[0..i, j]$ par $Y[0..k, \ell]$; (g) : substitution de $X[i, 0..j]$ par $Y[k, 0.. \ell]$ 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 $Y[0..k, \ell]$.

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 $T[3, 2, 2, 1] = 6$, comme présenté FIG. 3.

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 :

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

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 présentée FIG. 4.

Nous avons les initialisations marginales :

$$\begin{aligned} T_S[-1, j, k, \ell] &= T_S[i, -1, k, \ell] \\ &= T_S[i, j, -1, \ell] \\ &= T_S[i, j, k, -1] \\ &= 0 \end{aligned}$$

pour $0 \leq i \leq m_1 - 1, 0 \leq j \leq n_1 - 1, 0 \leq k \leq$

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 $a \neq b$, FIG. 5 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.

4 Application

L'algorithme a été testé sur des dialogues entre parents et enfants de 4 ans lors de la narration d'une histoire enfantine dont le contexte est mentaliste. Chaque énoncé est tout d'abord retranscrit et codé selon la grille de Chanoni [10, 9]. Nous avons ensuite réalisé les alignements de toutes les paires de dialogues distincts, ainsi que les alignements intra-dialogues, afin d'en extraire des régularités dans les comportements verbaux des parents et des enfants.

4.1 Retranscription et codage des dialogues

La grille utilisée pour le codage des dialogues est celle de Chanoni [10, 9], conçue spécialement 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 [34], selon l'âge de l'enfant auquel ils s'adressent.

Cette grille est composée de cinq colonnes, chaque colonne comporte entre 2 et 7 codages (6, 3, 7, 2, 2) (voir FIG. 6) :

– Les indices relevés dans la première colonne

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

FIG. 3 – Exemples de scores pour les opérations d'édition

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

FIG. 4 – Calcul de T_S , avec $0 \leq i \leq m_1 - 1$, $0 \leq j \leq n_1 - 1$, $0 \leq k \leq m_2 - 1$ et $0 \leq \ell \leq n_2 - 1$

A b C J e f g T i j 5 l m n o	A b C J e f g T i
B b C d E f g h i j k l m n o	B b C d E f g h i
B b C J L f g h i j k l m n o	B b C J L f g h i
A b C d E f g h i j k l m n o	A b C d E f g h i j k l m n

A b C J e f g T i j 5 l m n	A b C J e f g T i
B b C d E f g h i , ; : ! ?	B b C d E f g h i
x y z t u v w a c e p q r s	C
B b C J L f a b a b a c a b	B b C J L f a b a
A b C d E f g h i j k l m n	A b C d E f g h i j k l m n
a j g j h a j h i j a j j j	

FIG. 5 – Alignement local de motifs 2D.

Dialogue b7

:
 24 Parent on va la retrouver ta couronne
 25 Parent t'inquiète pas
 26 Parent donc là ils se cachent
 27 Parent ils cherchent
 28 Parent qui pourrait avoir pris la couronne
 29 Enfant elle dedans, elle est dedans la couronne
 30 Parent donc ils suspectent plein de monde,
 Cornélius, Céleste, la vieille dame
 31 Parent qui a bien pu prendre la couronne ?
 32 Enfant la couronne elle est dedans
 33 Parent tu crois ? !
 34 Enfant oui
 35 Parent mais Babar il ne sait pas qu'elle est dedans
 36 Parent donc il se dit que c'est une bombe, la couronne
 37 Parent ou je ne sais quoi ?
 :

Grille mentaliste

24	a	[{)]
25	A	P	E)]
26	A	P	B)]
27	a	[{)]
28	q	[{)]
29	a	[{)]
30	A	P	Y	C	J
31	q	[{)]
32	a	[{)]
33	Q	H	K)]
34	a	[{)]
35	A	P	N	O	J
36	A	P	N	C	J
37	A	R	N)]

FIG. 6 – Codage des dialogues

s'agir d'une affirmation (a ou A), d'une question (q ou Q). L'état de majuscule (A, Q) correspond à la présence d'un état mental. Est aussi relevé dans cette colonne s'il s'agit d'une demande d'attention sur l'histoire (AH) ou 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 (H) ou au locuteur (R). La référenciation n'est codée qu'en association avec la présence d'un état mental.
- La troisième colonne relève les états mentaux. Les partenaires du dialogue peuvent décrire l'émotion (E), la volition (V), la cognition observable (B) ou non observable (N), l'épistémie (K), l'hypothèse (Y) 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 (J), soit pour expliquer une situation par l'évocation d'un contexte personnel (CP).

Les parenthèses, accolades et autres crochets ([, ,),] sont utilisés pour coder l'absence d'indice dans une colonne donnée.

Par exemple, nous présentons de façon plus précise le codage de l'énoncé figurant ligne 35 du dialogue b7 (voir FIG. 6) : “ *mais Babar, il ne sait pas qu'elle est dedans* ”. Cet énoncé est une

assertion contenant un état mental, c'est pourquoi nous codons A dans la première colonne. Cet état mental, référencé au personnage (“ *il ne sait pas* ”), est codé P dans la deuxième colonne. L'état mental (“ *sait* ”), relatif à la cognition non observable, est codé N dans la troisième colonne. Nous relevons par ailleurs une justification par opposition repérée grâce à la conjonction de coordination “ *mais* ”, que nous codons O dans la quatrième colonne. Le contexte de cette justification est codé dans la cinquième et dernière colonne. Pour cet énoncé, le contexte fait référence à l'histoire en tant que telle (“ *qu'elle [la couronne de Babar] est dedans* ”) que nous codons J.

4.2 Extraction de motifs par programmation dynamique

Nous avons ensuite réalisé les alignements de toutes les paires de dialogues et les alignements intra-dialogues du corpus de dialogues présenté précédemment, afin de repérer des motifs approchés (ou flous) se répétant.

Une matrice de substitution a été spécifiquement construite pour cette application, afin d'exprimer les opérations d'édition élémentaires. Cette matrice peut être vue comme une représentation des liens existants entre les différents indices du codage. Le lien entre deux indices est représenté par une valeur allant de 0 à 10. Lorsque deux indices sont liés par une valeur faible, cela indique qu'ils ne peuvent pas être substitués l'un

par l'autre. A l'inverse, deux indices liés par une valeur élevée peuvent être rapprochés et donc substitués l'un par l'autre.

Pour cette étude nous avons souhaité centrer notre analyse sur les états mentaux et leurs justifications mettant de côté les éléments indiquant la nature grammaticale de l'énoncé. Ainsi, nous avons choisi de lier plus fortement entre elles les catégories référenciation (2ème colonne), états mentaux (3ème colonne), justifications (4ème colonne) et contexte (5ème colonne). Nous avons volontairement distingué la catégorie nature de l'énoncé (1ère colonne) des autres catégories.

Il faut noter que la grille choisie pour tester l'algorithme n'est pas dédiée à la modélisation du dialogue mais a été conçue particulièrement pour une étude psychologique donnée. Elles contiennent notamment des redondances dont le biais est levé par les valeurs choisies pour la matrice de substitution.

Les données présentées ici sont issues de la comparaison de deux dialogues entre deux mères et leur enfant de 4 ans (voir dialogues b7 et b9, FIG. 7 et FIG. 8).

L'algorithme fait apparaître des motifs en forme de L, se répétant au sein des deux dialogues (b7 et b9). Ces motifs flous peuvent être décomposés en deux parties significatives :

- Le premier ensemble est composé d'une série de lignes (partie verticale du L) qui comportent des indices relatant la présence d'un état mental associé à une référenciation.
- La deuxième partie forme la partie horizontale du L (lignes 35 et 36 pour le dialogue b7 par exemple) pour lesquelles nous notons des énoncés comportant des états mentaux associés à des justifications.

L'intérêt de ce motif réside dans le fait qu'il semble caractériser à la fois une organisation dialogique et un contenu. En effet, le premier ensemble du motif (partie verticale du L) fait apparaître au moins trois énoncés mentalistes au milieu d'une petite dizaine d'énoncés. Il est intéressant de noter qu'il ne s'agit pas d'une reformulation d'un seul état mental, mais de l'utilisation diversifiée d'états mentaux. Dans le dialogue b7, nous relevons la cognition observable (B), l'hypothèse (Y), l'épistémie (K) et la cognition non observable (N). La deuxième partie du motif est composée de deux énoncés complémentaires

qui regroupent chacun une justification associée à une interprétation mentaliste.

Ainsi, notre hypothèse est que les parents prennent soin de proposer plusieurs interprétations mentalistes de la situation, en décrivant et précisant les états mentaux en jeu, tout en prenant soin de les aérer par des énoncés simples (assertions ou questions). Les parents peuvent ensuite énoncer une double justification complexe, précisant les liens de cause, de conséquence ou d'opposition qui déterminent les états mentaux du personnage et ses comportements. Cette préparation de l'enfant à comprendre une justification mentaliste que nous venons de décrire est un fait délicat, voire impossible à observer "à l'oeil nu". En effet, pour cet exemple, ce motif a été mis en lumière à l'intérieur d'une série de plus de 37 et 86 lignes (respectivement pour les dialogues b7 et b9).

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 méthode permet de calculer des alignements globaux et locaux. Nous avons appliqué cette méthode à l'extraction de motifs dans des dialogues entre parents et enfants de quatre ans, annotés par une grille mentaliste.

Une première difficulté consiste à visualiser les alignements ainsi calculés. En effet, le nombre de motifs qui peuvent être mis en évidence est assez conséquent mais nécessitent à chaque fois une expertise humaine afin d'en évaluer la pertinence. De plus, il n'existe pas à ce jour de méthode d'affichage pratique permettant de visualiser et comparer aisément deux motifs rapprochés. Enfin, quand un motif se répète dans plusieurs dialogues, il est aussi difficile de choisir un motif "consensuel", permettant de représenter toutes les occurrences.

Du point de vue technique, beaucoup de questions peuvent être posées. Est-il possible d'ali-

Dialogue b7						Dialogue b9					
24	a		{)		8	q		{)	
25	A	P	E)		9	A	P	B)	
26	A	P	B)		10	q		})	
27	a		})		11	q		})	
28	q		})		12	q		})	
29	a		})		13	A	P	B)	
30	A	P	Y	C	J	14	a		})	
31	q		})		15	A	P	N)	
32	a		})		16	q		})	
33	Q	H	K)		17	a		})	
34	a		})		18	a		})	
35	A	P	N	O	J	19	A	P	V)	
36	A	P	N	C	J	20	A	P	B	O	J
37	A	R	N)		21	A	P	S	O	J
						22	a		})	
						23	a		})	

FIG. 7 – Alignement de motifs mentalistes

Dialogue b7

24 Parent on va la retrouver ta couronne
 25 Parent t'inquiète pas
 26 Parent donc là ils se cachent
 27 Parent ils cherchent
 28 Parent qui pourrait avoir pris la couronne
 29 Enfant elle dedans, elle est dedans la couronne
 30 Parent donc ils suspectent plein de monde, Cornélius, Céleste, la vieille dame
 31 Parent qui a bien pu prendre la couronne ?
 32 Enfant la couronne elle est dedans
 33 Parent tu crois ?!
 34 Enfant oui
 35 Parent mais Babar il ne sait pas qu'elle est dedans
 36 Parent donc il se dit que c'est une bombe, la couronne
 37 Parent ou je ne sais quoi ?

Dialogue b9

7 Parent même son ami Zéphir la cherche partout avec sa loupe
 8 Parent mais où est elle donc passée
 9 Parent Babar remarque Cornélius donner un paquet à l'intrus
 10 Parent Qu'est ce qui peut y avoir dedans ?
 11 Parent mais qui est donc cet individu masqué
 12 Parent qui a volé la couronne
 13 Parent Babar la démasque !
 14 Parent c'est la reine céleste !
 15 Parent il se pose bien des questions
 16 Parent pourquoi donc la reine Céleste s'est déguisée.
 17 Parent Babar va donc chez la vieille dame lui demander
 18 Enfant oui
 19 Parent la vieille dame ne veut pas qu'il rentre !
 20 Parent mais derrière se cachait
 21 Parent pour lui une surprise en réalité
 22 Parent puis Babar rentra chez lui

FIG. 8 – Motifs mentalistes similaires repérés dans deux dialogues différents

gner des motifs 2D en temps sous-quadratique comme c'est le cas pour les séquences ? (voir [13]). Est-il possible de développer une heuristique similaire à BLAST [2] 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. Enfin, il faut noter que cette méthode n'a pour l'instant été appliquée que sur des dialogues de taille modeste, avec une grille d'annotations relativement restreinte. Dans le cas d'un corpus plus conséquent (principalement concernant la longueur des dialogues et le volume des annotations) des méthodes d'approximation devront être envisagées.

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. 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 capturer 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 motifs dialogiques efficaces. Ainsi, la formation aux techniques d'entretiens pourrait s'en trouver améliorée. Nous pourrions penser qu'un jour cette méthode pourrait assister en temps réel le locuteur dans son entretien.

Références

- [1] J. Allen and R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15 :143–178, 1980.
- [2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *J. Mol. Biol.*, 215 :403–410, 1990.
- [3] A. Amir, G. Benson, and M. Farach. An alphabet independent approach to 2-dimensional pattern matching. *SIAM J. Comput.*, 23(2) :312–323, 1994.
- [4] A. N. Arslan. A largest common d-dimensional subsequence of two d-dimensional strings. In *FCT*, volume 4639 of *LNCS*, pages 40–51, 2007.
- [5] R. A. Baeza-Yates. Similarity in two-dimensional strings. In *COCOON*, volume 1449 of *LNCS*, pages 319–328, 1998.
- [6] C. Balkanski and M. Hurault-Plantet. Cooperative requests and replies in a collaborative dialogue model. *International Journal of Human-Computer Studies*, 53(6) :915–968, 2000.
- [7] K. Bouzouba and B. Moulin. Les interactions distribuées : Approche basée sur le point de vue personnel des agents. In *MFI*, pages 15–30, 2001.
- [8] C. Brassac and S. Pesty. Simuler la conversation : un défi pour les systèmes multiagents. In *Analyse et simulation de conversations. De la théorie des actes de discours aux systèmes multiagents*, chapter 9, pages 317–345. *L'interdisciplinaire*, 1999.
- [9] É. Chanoni. Comment les mères racontent une histoire de fausses croyances à leur enfant de 3 à 5 ans ? *Enfance*, 2004.
- [10] É. Chanoni. *Rôle du langage dans le développement de la théorie de l'esprit chez les enfants de 3 à 5 ans : contexte verbal et contexte narratif*. PhD thesis, Université de Lille, 2004.
- [11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.
- [12] M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on strings*. Cambridge University Press, 2007.
- [13] M. Crochemore, G. M. Landau, and M. Ziv-Ukelson. A subquadratic sequence alignment algorithm for unrestricted scoring matrices. *SIAM J. Comput.*, 32(6) :1654–1673, 2003.
- [14] J.-L. Dessalles. *Modèle cognitif de la communication spontanée appliqué à l'apprentissage*. PhD thesis, ENST Paris, 1993.
- [15] R. Giancarlo and R. Grossi. Suffix tree data structures for matrices. In A. Apostolico and Z. Galil, editors, *Pattern matching algorithms*, chapter 10, pages 293–340. Oxford University Press, 1997.

- [16] E. Hansen, D. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on AI*, pages 709–715, 2004.
- [17] J. Kärkkäinen and E. Ukkonen. Two and higher dimensional pattern matching in optimal expected time. In *SODA*, pages 715–723, 1994.
- [18] D. K. Kim, Y. A. Kim, and K. Park. Generalizations of suffix arrays to multi-dimensional matrices. *Theoret. Comput. Sci.*, 302(1–3) :223–238, 2003.
- [19] K. Krithivasan and R. Sitalakshmi. Efficient two-dimensional pattern matching in the presence of errors. *Inf. Sci.*, 43 :169–184, 1987.
- [20] J. Lehuen. *Un modèle de dialogue dynamique et générique intégrant l’acquisition de sa compétence linguistique - Le système COALA*. PhD thesis, Université de Caen, 1997.
- [21] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5) :367–381, 2002.
- [22] H. Mazouzi. *Ingénierie des protocoles d’interaction : des systèmes distribués aux systèmes multi-agents*. PhD thesis, Université Paris IX, 2001.
- [23] J. Moeschler. *Modélisation du dialogue*. Hermès, 1989.
- [24] J. C. Na, R. Giancarlo, and K. Park. Online construction of two-dimensional suffix trees in $O(n^2 \log n)$ time. *Algorithmica*, 48(2) :173–186, 2007.
- [25] A. Pauchet. *Modélisation cognitive d’interactions humaines dans une cadre de planification multi-agents*. PhD thesis, Université Paris Nord, 2006.
- [26] E. Roulet, A. Auchlin, J. Moeschler, C. Rubattel, and M. Schelling. *L’articulation du discours en français*. Peter Lang, 1985.
- [27] D. Sadek, P. Brétier, and F. Panaget. ARTIMIS : Natural dialogue meets rational agency. In *Actes d’International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1030–1035, Yokohama, Japan, 1997.
- [28] D. Sankoff and J. B. Kruskal. *Time warps, string edits, and macromolecules : the theory and practice of sequence comparison*. Cambridge University Press, 1999. 2nd edition.
- [29] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the 7th International Conference on Machine Learning*, pages 216–224, 1990.
- [30] D. Szer and F. Charpillat. Point-based dynamic programming for DEC-POMDPs. In *Proceedings of the 21st National Conference on AI*, 2006.
- [31] FIPA. FIPA request interaction protocol specification. Technical report, Foundation for Intelligent Physical Agents, 2002. Avail. at <http://www.fipa.org>.
- [32] A. Trognon and C. Brassac. L’enchaînement conversationnel. In *Cahier de linguistique française*, volume 13, pages 76–116. Cazaubon, 1992.
- [33] D. Vanderveken. La structure logique des dialogues intelligents. In *Analyse et simulation de conversations. De la théorie des actes de discours aux systèmes multi-agents*, chapter 2, pages 61–100. L’interdisciplinaire, 1999.
- [34] H. Wimmer and J. Perner. Beliefs about beliefs : representation and constraining function of the wrong beliefs in young children’s understanding of deception. *Cognition*, (13) :103–128, 1983.
- [35] T. Winograd and F. Flores. *Understanding Computers and Cognition : A New Foundation for Design*. Ablex, 1986.