



**HAL**  
open science

## Reconnaissance de motifs dialogiques approchés

Zacharie Alès, Alexandre Pauchet

► **To cite this version:**

Zacharie Alès, Alexandre Pauchet. Reconnaissance de motifs dialogiques approchés. *Models Formels de l'Interaction*, Jun 2011, Rouen, France. pp.9-19. hal-03503109

**HAL Id: hal-03503109**

**<https://ensta-paris.hal.science/hal-03503109v1>**

Submitted on 27 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reconnaissance de motifs dialogiques approchés

Zacharie Alès†

Zacharie.Ales@insa-rouen.fr

Alexandre Pauchet†

Alexandre.Pauchet@insa-rouen.fr

† INSA-Rouen, LITIS EA 4108,

BP 08 - 76801 Saint-Étienne-du-Rouvray, France

## Résumé :

Dans le cadre de la modélisation du dialogue, nous présentons une méthode générique permettant l'alignement de motifs approchés en deux dimensions appliquée à des matrices d'annotations codant des dialogues. L'approche utilisée, basée sur un algorithme de programmation dynamique et une mesure de similarité, permet l'identification de motifs similaires à un motif de référence.

Ce procédé a été mis en pratique par la recherche de motifs récurrents dans des annotations de dialogues entre parents et enfants, lors de la narration d'histoires enfantines. Les motifs de référence utilisés sont issus d'un précédent travail [12]. Nous présentons brièvement quelques exemples d'alignements obtenus à partir d'un motif.

**Mots-clés :** Modélisation du dialogue, reconnaissance de motifs, identification de régularités, programmation dynamique

## 1 Introduction

Les systèmes interactifs existants, et plus particulièrement ceux intégrant une dimension dialogique, sont relativement robustes mais peu conviviaux. L'utilisateur doit souvent s'adapter à la machine, car les interfaces ne tiennent pas compte de certaines caractéristiques spécifiques aux interactions humaines. La difficulté majeure dans la conception de systèmes dialogiques agréables vient de la modélisation inadaptée des comportements humains à partir de l'analyse de situations réelles.

La problématique est en réalité plus large que la simple conception d'interfaces dialogiques. En effet, la modélisation du dialogue sous toutes ses modalités est un domaine dont les champs d'applications sont nombreux et pluridisciplinaires : informatique (interfaces dialogiques en langue naturelle, systèmes multi-agents hétérogènes, jeux vidéo), psychologie (développement socio-cognitif de l'enfant, pragmatiques développementales, apprentissage dans l'interaction), sciences humaines et sociales (formation aux entretiens, à la négociation), biologie et santé (détection de l'autisme, évaluation de la douleur), *etc.* Cependant, bien que les outils utilisés diffèrent selon le domaine d'application, la modélisation du dialogue s'appuie en général sur un corpus de dialogues annotés, dont l'analyse vise à repérer un ensemble de comporte-

ments récurrents significatifs, afin d'être en mesure de les expliquer (*i.e.* fournir un modèle).

L'approche adoptée pour étudier ces dialogues est alors similaire dans la plupart des domaines :

1. *Collecte d'un corpus de dialogues* en situation, sous forme audio, ou vidéo si la source d'information multimodale doit être prise en compte. Habituellement, cette collecte se fait au cours d'une expérimentation conçue spécifiquement. Les dialogues sont alors finalisés, et le cadre d'application du modèle est restreint à une tâche.
2. *Retranscription des dialogues* sous forme numérique, de façon à ce qu'ils soient archivés et puissent être analysés ultérieurement. Cette retranscription peut être plus ou moins fine selon les besoins (tours de parole, énoncés, onomatopées, pauses, *etc.*).
3. *Annotation des dialogues* selon des critères spécifiques, définis par un schéma de codage. Chaque énoncé est décrit par une codification prédéfinie (ex : force illocutoire appliquée à une proposition en théorie des actes de langage, grille spécifique en psychologie, inflexions de voix, ton, prosodie, mimiques, attitudes, postures, gestes, regards, émotions, *etc.*). En pratique, ce processus est souvent effectué manuellement.
4. *Extraction d'informations* à partir des annotations, consistant souvent à une recherche de régularités ou de répétitions, pouvant se faire de manière manuelle, semi-automatique ou automatique. Cette phase intègre parfois des connaissances spécifiques au type de dialogues étudiés.
5. *Le modèle des dialogues* s'appuie sur les régularités repérées lors de la phase d'extraction d'informations. Il peut être vu comme une banque de motifs caractéristiques représentés sous une forme exploitable informatiquement (ex : automates, réseaux de Petri, ensemble de règles, *etc.*).
6. *L'exploitation du modèle*, dépend alors du domaine d'application.

Avec la multiplication des éléments pris en compte par les modèles de dialogues, les annotations sont maintenant le plus souvent multi-dimensionnelles ([1, 10, 11] présentent des exemples de schémas de codages multi-dimensionnels). Cet article fait suite à un travail précédent visant à extraire des régularités dans des annotations multi-dimensionnelles de dialogues par programmation dynamique [12], correspondant à l'extraction d'informations (étape 4). Nous nous intéressons ici à la conception d'un algorithme de reconnaissance de motifs dialogiques approchés (collection d'annotations multi-dimensionnelles codant chacune un énoncé). L'intérêt de ce travail est double : 1) permettre la recherche d'un motif dialogique dans un corpus de dialogues et faciliter ainsi la conception d'une banque de motifs caractéristiques des interactions dialogiques dans un contexte donné (étape 5) et 2) disposer d'un algorithme de reconnaissance de situation dialogique (détection de motif) pouvant être intégré à un agent logiciel (étape 6).

Dans la suite de cet article, nous allons tout d'abord donner, en section 2, une description du problème et du projet dans lequel s'insèrent ces travaux. La section 3 présente un aperçu non exhaustif des différents modèles de dialogues et nous citerons plusieurs méthodes de reconnaissance de motifs. La section 4 décrit ensuite la méthode que nous avons développée spécifiquement pour répondre à notre problème. Dans la section 5, nous présenterons les résultats obtenus. Enfin, nous concluons cet article en donnant quelques perspectives pour ces travaux.

## 2 Description du problème

Cet article s'inscrit dans le cadre d'un projet visant à mettre en place un système permettant d'automatiser ou de semi-automatiser les différentes étapes d'analyse d'un corpus de dialogues : retranscription automatique des énoncés, annotations semi-automatique suivant différents schémas de codage, extraction de régularités dans les annotations et création d'une banque de motifs dialogiques caractéristiques d'un corpus de dialogues annotés.

Notre approche se veut générique afin de définir un ensemble d'algorithmes indépendants du corpus de dialogues étudié et du type d'annotations utilisées. Chaque énoncé de chaque dialogue analysé est codé par un vecteur d'annotations, dont les composantes correspondent aux

différents espaces de codage à appliquer aux dialogues. Ainsi, un dialogue sera représenté par une matrice : une ligne par énoncé (temporalité verticale), une colonne par espace de codage (dimensionnalité horizontale).

Afin d'illustrer cette représentation matricielle, considérons un dialogue fictif entre un guichetier et un client dans une gare (voir figure 1).

G	Bonjour, que puis-je pour vous ?	E	I	V
C	Bonjour, un Paris-Rouen.	D	I	M
C	Vite, s'il vous plaît !	D	P	
G	Je vérifie...	E		A
G	Il y en a un dans 10mn.	A		A
G	Je vous sors un billet ?	E		V

FIGURE 1 – Exemple d'annotations sur trois colonnes d'un dialogue fictif guichetier/client

Chaque énoncé est caractérisé par son locuteur (G : guichetier, C : client), une retranscription (ici en énoncés et non en tours de parole) et son codage suivant 3 colonnes :

- La première colonne code l'énoncé par un type d'acte de langage (E : engageant, D : directif et A : assertif) [31].
- La seconde colonne code la présence d'un élément de nature sociale (I : bonjour/au revoir, P : politesse).
- La troisième colonne code le regard (M : contact visuel mutuel, V : regard unilatéral vers le visage de l'interlocuteur, A : regard vers un autre élément identifié).

Contrairement à la colonne 1, les colonnes 2 et 3 peuvent, pour un énoncé donné, ne contenir aucun élément codant. Cette matrice de codage pourrait aisément être complétée afin d'obtenir une analyse plus fine du dialogue, en intégrant par exemple une dimension émotionnelle ou des informations de référencement.

Une telle représentation matricielle permet de faire apparaître les liens existants entre les différentes dimensions codées, sous la forme de motifs dialogiques. Un *motif* est défini comme un ensemble d'annotations dont la disposition se retrouve - de manière exacte ou approchée - dans plusieurs dialogues. Aucune contrainte structurelle n'est imposée : un motif peut être composé d'annotations non adjacentes en ligne ou en colonne (*i.e.* il peut contenir des "trous"). L'exemple de la figure 2 illustre ce propos.

A partir d'un corpus de dialogues annotés sous une forme matricielle similaire à celle précédemment décrite, l'objectif est de construire une banque de motifs caractéristiques des interactions dialogiques. L'intérêt d'une telle banque

	C <sub>1</sub>	C <sub>2</sub>	...	C <sub>n-1</sub>	C <sub>n</sub>
<b>A</b>	A	d		h	X
	B	e		k	Y
	C	d		h	X
	A	d		m	Z
	A	d		m	Z
	C	e	...	j	Y
	B	e		k	Y
	B	d		m	Y
	A	d		h	W
	A	d		j	Z

	C <sub>1</sub>	C <sub>2</sub>	...	C <sub>n-1</sub>	C <sub>n</sub>
<b>B</b>	C	f		j	W
	C	f		j	Z
	C	d		m	Z
	A	d		j	Y
	A	d		m	Z
	C	f		j	Y
	C	e		k	Y
	A	e		k	Z
	C	f		m	Z
	B	f		j	X

FIGURE 2 – Exemple de motifs similaires présents dans deux dialogues, **A** et **B**, différents. Le codage matricielle comporte  $n$  colonnes. Chaque colonne possède un alphabet qui lui est propre. L’alphabet d’une colonne correspond à l’ensemble des annotations susceptibles de se trouver dans cette colonne. En bleu sont représentés les deux versions du motif.

de motifs est de pouvoir être intégrée à un système de dialogues ou à un agent conversationnel afin d’en améliorer l’efficacité. La création d’une banque de motifs passe par une phase de recherche de motifs similaires dans les matrices codant les dialogues étudiés. Dans cette optique, un algorithme d’alignement de matrices par programmation dynamique a été développé durant un travail précédent [12]. Ce dernier retourne une liste de paires de motifs similaires, ordonnée suivant une mesure de similarité (distance d’édition). L’algorithme a donné lieu à un dépôt logiciel sur sourceforge d’APSAAD, dont l’interface est présentée figure 3<sup>1</sup>. Chaque motif de chaque paire retournée doit alors faire l’objet d’une recherche dans l’intégralité du corpus afin de constituer la banque de motifs. L’algorithme de reconnaissance de motifs approchés, objet du présent article, est utilisé à cette occasion.

### 3 Dialogues et motifs dialogiques

Dans cette section, nous présentons tout d’abord un ensemble de modèle de dialogues afin de mettre en avant la pertinence d’une approche

fondée sur la reconnaissance de motifs pour les systèmes de dialogues et agents conversationnels. Puis nous introduisons différents travaux sur la reconnaissance de motifs en deux dimensions (ou alignement de matrices), avec pour perspective de pouvoir les utiliser sur les annotations codant les dialogues d’un corpus.

#### 3.1 Modélisation du dialogue

La théorie des actes de langage est la théorie dominante en informatique pour identifier les énoncés selon leur caractère performatif [31]. L’intérêt de la communauté informatique pour la théorie des actes de langage est né des avantages que la logique illocutoire apportait [30]. En prolongeant la théorie des actes de langage aux conversations, Vanderveken dégage une logique du discours [35] et introduit comme unités de base du raisonnement les états mentaux.

Les travaux de Grosz et Sidner font glisser le modèle du dialogue s’appuyant sur des relations entre énoncés vers un modèle reposant sur un ensemble de relations entre les intentions [20]. Ils distinguent la structure linguistique découpée en segments dialogiques. Selon ces auteurs [19], le dialogue est une collaboration entre les partenaires, la modélisation du dialogue à partir de structures de plans partagés sera donc privilégiée. Clark et Schaeffer ont décrypté ce processus de partage de connaissances communes [13]. Tout d’abord, A fait une proposition à B, puis B accepte l’énoncé en témoignant de sa compréhension, l’énoncé prenant alors le statut de terrain commun. Il s’agit là, même si cela n’a pas été formalisé comme tel par les auteurs, d’un motif dialogique linéaire basique.

Le modèle de Roulet [15] tente de rassembler dans un modèle fédérateur plusieurs théories (actes de langage, théorie de l’argumentation et éléments d’analyse conversationnelle). Le modèle a comme avantage de rendre compte de la structure hiérarchique du dialogue en construisant une représentation des liens entre les différents énoncés. Souhaitant préciser les relations rhétoriques entre les diverses parties du dialogue, la RST (Rhetoric Structure Theory) [24] propose une vingtaine de relations permettant ainsi de compléter le modèle de Roulet. Les relations ainsi caractérisées deviennent un moyen d’étudier l’intentionnalité. La SDRT (Segmented Discourse Representation Theory) [6], terminera de démontrer que le dialogue n’est plus composé d’une suite d’actes de langage mais de segments reliés entre eux par des relations

1. <http://sourceforge.net/projects/apsaad/>

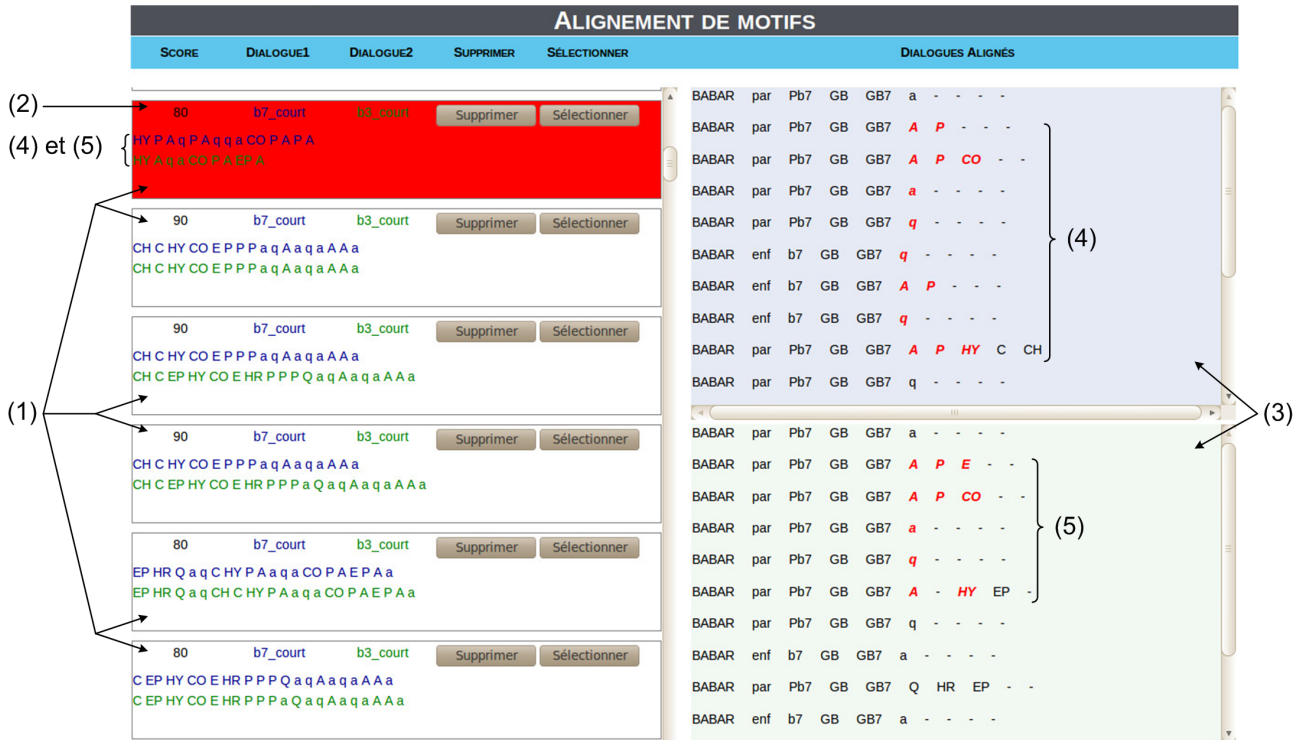


FIGURE 3 – Logiciel APSAD permettant l’extraction de motifs dans des annotations de dialogues. (1) Liste des alignements trouvés. (2) Alignement sélectionné par l’utilisateur. (3) Visualisation de l’alignement sélectionné. (4) Annotations du premier dialogue faisant partie de l’alignement sélectionné. (5) Annotations du second dialogue faisant partie de l’alignement sélectionné.

de discours. Nous considérons que ces relations permettent de former des motifs dialogiques, cette fois en deux dimensions, représentatifs de la structure hiérarchique du dialogue.

Le système TRAINS est l’un des premiers à dialoguer avec un humain, dans le cadre de la résolution collaborative de problèmes [17]. L’analyse de corpus de dialogues oraux orientés tâche a conduit à la conception de différents systèmes pour interroger en langue naturelle une base de données contenant des horaires de trains. TRIPS, successeur de TRAINS, permet en plus de communiquer par des graphiques afin d’améliorer la compréhension mutuelle [16]. Ces travaux ont mené jusqu’à une architecture d’agent pour la planification collaborative. Il génère des actes de résolution collaborative de problèmes que les agents reconnaissent et utilisent pour coordonner leurs actions individuelles. Le module de communication est similaire à celui de Traum [34]. Il est fondé sur la planification et a pour but de reconnaître le plan de l’interlocuteur pour émettre une réponse appropriée. Le projet COLLAGEN a pour but de développer un agent en charge des communications, indépendant de l’application, pour faciliter les interac-

tions collaboratives entre agents logiciels et utilisateurs [29]. COLLAGEN utilise la notion de plans partagés de [19], pour modéliser les dialogues humains. Les énoncés et actions à effectuer sont sélectionnés selon leur contribution au dialogue. Les projets TRAINS et COLLAGEN ont une approche du dialogue identique, basée sur la planification. La structure de plan regroupant les différents actes communicatifs sont, une fois encore, similaires à des motifs dialogiques.

Enfin, la théorie QUD est une théorie conventionnelle du dialogue fondé sur une sémantique formelle des questions et d’un tableau conversationnel structuré. Son but est de déterminer les propriétés des couples Question-Réponse (Q-R) et plus particulièrement les effets qu’ont les questions et les assertions sur le tableau conversationnel. En s’appuyant sur une approche ISU (Information State Update) [23], Ginzburg conçoit le dialogue en termes de coups dans un jeu. Il est possible de définir des relations intentionnelles entre jeux de dialogues [25, 26] ainsi que des relations sémantiques. Cette théorie a conduit à une implémentation d’un système, GoDIS [22], qui englobe certains principes de la planification du dialogue [28, 34]

tels que les plans partagés [19], ainsi que la prise en compte du principe d’accommodation [14]. De plus, le dialogue peut s’écarter de la tâche pour introduire une digression ou une explication, notamment dans le cadre d’une clarification ou de l’établissement du terrain commun.

Chacun des modèles précédemment cités est, en réalité, conçu à partir d’une collection de motifs reliant différents niveaux (pragmatique locale du dialogue, plans, *etc.*), et peut se structurer suivant différents axes (actes de dialogue, gestion des émotions, *etc.*). Un dialogue peut être modélisé par une structure matricielle dont chaque colonne représente un des axes de codage du dialogue. Fournir un modèle dialogique pour un corpus de dialogues donné revient donc à établir un ensemble de régularités (les motifs) dans les matrices codant le corpus.

### 3.2 Reconnaissance de motifs

Pour de nombreuses problématiques, la reconnaissance de motifs se limite à une seule dimension et peut être assimilée à un problème d’alignements de séquences de caractères : recherche de texte, traitements audios, reconnaissance de formes, alignement de séquences ADN, ... Dans ce cadre, la distance d’édition  $ed$  (ou distance de Levenshtein) entre deux chaînes de caractères  $s_1$  et  $s_2$  est parfaitement adaptée. Elle est définie comme le coût minimal des opérations d’édition permettant de passer de  $s_1$  à  $s_2$ . Les trois opérations possibles sont la suppression et l’insertion d’un caractère (*Del*, *Ins*) ainsi que la substitution d’un caractère par un autre (*Sub*). Dans un alphabet  $\Sigma$  de taille  $\sigma$ , à chaque caractère sera associé  $\sigma + 1$  coûts (1 insertion, 1 suppression et  $\sigma - 1$  substitutions). Un exemple d’alignement est présenté figure 4.

A	-	C	T	A	T	T	-
-	T	C	A	A	T	T	A

FIGURE 4 – Alignement obtenu lors du calcul de la distance de Levenshtein entre les chaînes de caractères *ACTATT* et *TCAATTA*. Ici le coût de substitution d’un caractère par lui-même est fixé à 0 tandis que toute autre opération d’édition coûte 1. Le coût total de cet alignement est égal à 4 (deux insertions, une suppression et une substitution d’un caractère par un autre).

Considérons un motif  $P[1..m]$  et un texte  $T[1..n]$ . L’alignement de chaînes de caractères approchées peut se définir comme la recherche

de segments de  $T$  dont la distance d’édition à  $P$  est au maximum égale à un entier  $k$  donné. Plus formellement, un alignement sera reporté en position  $j$  de  $T$  si

$$led(T[1..j], P[1..m]) \leq k$$

$$\text{avec } led(T[i..j], P[1..m]) = \min_{s \in i..j} ed(T[s..j], P[1..m]).$$

La reconnaissance de motifs dialogiques dans des annotations de dialogues s’apparente plutôt à la recherche de motifs approchés en deux dimensions (ou alignement de matrices). Elle nécessite la définition d’une métrique permettant d’évaluer la similarité entre deux tableaux de caractères. Krithivasan et Sitalakshmi (KS) ont proposé pour cela une extension de la distance d’édition [33]. Ainsi, deux tableaux de taille identique seront comparés en utilisant la somme de la distance d’édition de chacune des lignes correspondantes. Cette définition est pertinente dans des contextes où les suppressions et insertions de lignes sont impossibles. L’insertion d’un caractère ne permettra, par exemple, pas de décaler tous les caractères d’une colonne d’une position vers le bas. De même, la suppression de tous les éléments d’une ligne n’entraînera pas de décalage dans les lignes suivantes.

De nombreuses méthodes utilisant cette métrique ont été développées. Certaines imposent de fortes contraintes à la forme des motifs. En effet, il est parfois nécessaire que ceux-ci soient rectangulaires [21] ou mi-rectangulaires [4]. D’autres approches se restreignent à des sous-problèmes afin d’obtenir de meilleures performances. Bird [9] et Baeza-Yates [8], entre autres, se focalisent sur le cas particulier  $k = 0$ . Similairement, Amir et Landau [5] présentent une méthode où la substitution est la seule opération d’édition autorisée. Baeza-Yates et Navarro [7] ont développé un algorithme général, acceptant les trois types d’opérations d’édition et n’imposant pas de conditions sur la forme des motifs. Cette méthode a par la suite été améliorée pour qu’il soit possible de l’appliquer dans des contextes plus spécifiques. Amir et Eran l’ont, par exemple, utilisée pour l’alignement de motifs pouvant subir des mises à l’échelle [3]. De même, elle a été adaptée au cas de motifs comprenant plus de deux dimensions [27] ou pouvant subir des rotations [18].

Dans cet article, nous présentons une heuristique de reconnaissance de motifs en adaptant l’approche de Baeza-Yates et Navarro [7].

## 4 Reconnaissance de motifs dialogiques approchés par programmation dynamique

Considérons un motif  $P = [1..m, 1..l]$  et un texte annoté  $T = T[1..n, 1..l]$ , de taille respective  $M = m \times l$  et  $N = n \times l$ . Notons  $\Sigma$  l'alphabet étudié.

Nous souhaitons déterminer les zones de  $T$  pour lesquelles la distance d'édition à  $P$  est inférieure à un entier  $k$  donné. En d'autres termes, nous effectuons un alignement global du motif et un alignement local du texte.

Nous nous plaçons dans le contexte de KS en colonne, pour lequel les insertions ou suppressions de colonnes ne sont pas permises. Ceci se justifie par la structure même des matrices d'annotations. En effet, chaque colonne représente un axe de codage, tandis que les lignes représentent les énoncés successifs du dialogue. L'insertion ou la suppression d'énoncés est nécessaire pour permettre une plus grande souplesse dans la représentativité des dialogues. En revanche, chaque annotation étant relative à une colonne donnée, l'insertion ou la suppression d'un élément dans une colonne aurait pour conséquence le décalage des annotations suivante sur la ligne dans une colonne ne correspondant pas à leur dimension de codage.

$T$  et  $P$  ayant le même nombre de colonnes, leur correspondance est immédiate. Pour chaque couple de colonnes  $P[1..m, j], T[1..n, j], j \in \llbracket 1, l \rrbracket$  ainsi formés, nous déterminons les alignements compatibles ainsi que leurs coûts. Pour ce faire, un algorithme d'alignement de séquences (alignements en une dimension) est appliqué à chacune des  $l$  colonnes, suivant un ordre spécifique.

### 4.1 Alignement de séquences

De nombreux algorithmes en une dimension permettent de déterminer les alignements possibles entre deux chaînes de caractères. Le choix d'une méthode, dans un contexte donné, dépend principalement du coût maximal  $k$  et de la taille du motif  $m$ . Le lecteur peut se référer à [7] pour de plus amples informations. Afin de permettre une grande variabilité à  $m$  et  $k$ , nous avons choisi d'utiliser un algorithme de programmation dynamique classique, similaire à [32].

Considérons deux chaînes de caractères  $s_1$  et  $s_2$ ,

de taille respective  $t_1$  et  $t_2$ . L'algorithme permettant de calculer un alignement global de  $s_1$  et  $s_2$ , consiste à calculer une matrice  $C[0..t_1, 0..t_2]$  telle que  $C[i, j] = ed(s_1(1..i), s_2(1..j))$ . Le coût de l'alignement est alors égal à  $C[t_1, t_2]$  (voir FIG. 5 pour le détail de l'algorithme).

```

Entree :
  s1[1..t1], s2[1..t2] : chaines de caracteres

Sortie :
  C[t1, t2]

Debut
  Pour j = 0 a t2
    C[0, j] = j

  Pour i = 1 a t1
    C[i, 0] = i

  Pour i = 1 a t1
    Pour j = 1 a t2
      C[i, j] = min(C[i-1, j] + Del(s1[i]), _
                    C[i, j-1] + Ins(s2[j]), _
                    C[i-1, j-1] + Sub(s1[i], s2[j]))

```

FIGURE 5 – Alignement global en une dimension. *Del* et *Ins* sont les matrices des coûts de suppression et d'insertion d'un caractère et *Sub* est la matrice des coûts de substitution d'un caractère par un autre.

Afin que l'alignement soit local uniquement par rapport au texte, seule la première colonne de  $C$  est initialisée à 0 (un alignement local par rapport au motif simultanément nécessiterait également une initialisation de la première ligne de  $C$  à 0). La chaîne  $s_2$  peut donc débiter en n'importe quelle position avec un coût nul. Un alignement sera détecté pour tout  $i$  tel que  $C[i, t_2] \leq k$ . Un algorithme de tracé arrière à partir de la position correspondante permet d'identifier l'ensemble des alignements.

Les motifs annotés que nous cherchons, ne sont pas nécessairement de forme pleine ou rectangulaire. Il nous faut donc définir un caractère  $\varepsilon$  représentant une annotation non renseignée. Le coût de substitution de ce caractère par un autre sera nul. Son coût d'insertion sera égal à  $\max_{\sigma \in \Sigma} Ins(\sigma)$ , puisqu'insérer un  $\varepsilon$  revient à insérer n'importe quelle annotation. Il n'est pas nécessaire de définir un coût de suppression car le texte ne contiendra aucun  $\varepsilon$ .

### 4.2 Alignement de motifs

L'ordre dans lequel sont effectuées les alignements de séquences et plus particulièrement le choix de la première colonne testée, influe sur le temps d'exécution. Soit  $c_j \in \llbracket 1, l \rrbracket, j \in$

$\llbracket 1, l \rrbracket$ , la  $j^{\text{ème}}$  colonne testée par l'algorithme. Tout d'abord, des alignements sont cherchés sur l'ensemble de  $c_1$ , ce qui nous fournit une liste de solutions candidates. Ces dernières seront, par la suite, développées ou rejetées lors du test des autres colonnes. Idéalement, la première colonne sera celle fournissant un nombre de candidats minimal. Dans cette optique nous choisissons la colonne dont le motif comporte le plus grand nombre de caractères.

Un ou plusieurs alignements seront reportés pour  $c_1$ , en position  $i$  si

$$\text{led}(T[i-m-\lfloor \frac{k}{i_m} \rfloor + 1 \dots i, c_1], P[1 \dots m, c_1]) \leq k$$

avec  $i_m = \min_{\sigma \in \Sigma} \text{Ins}(\sigma)$ .

Il est nécessaire que la zone de recherche ait une longueur supérieure à celle du motif, afin de permettre d'éventuelles insertions de caractères du texte. Pour déterminer le nombre maximal de caractères pouvant être insérés, nous considérons ceux dont le coût d'insertion est minimal (*i.e.* égal à  $i_m$ ). En prenant la partie entière de la division entre  $k$  et  $i_m$ , nous obtenons le nombre maximal d'insertions possibles.

Les autres colonnes seront testées localement en fonction de la position des solutions candidates restantes. Pour chaque position  $i$ , solution du test de  $c_{j-1}$ ,  $j \in \llbracket 2, l \rrbracket$ , un ou plusieurs alignements seront reportés pour  $c_j$  si

$$\text{led}(T[i-m-d_j+1 \dots i+d_j, c_j], P[1 \dots m, c_j]) \leq k$$

avec  $d_j = \lfloor \frac{k-\hat{k}_j}{i_m} \rfloor$ ,  $\hat{k}_j = \sum_{t=1}^{j-1} k_t$  et  $k_t$  le coût des erreurs en  $c_t$  pour la solution considérée.

L'augmentation de  $\hat{k}_j$  au fil des colonnes, permet une meilleure restriction des zones testées.

Ainsi, les solutions fournies par le test de  $c_l$  sont des alignements de  $P$  sur une sous-partie de  $T$  dont le coût est inférieur à  $k$ .

## 5 Application

### 5.1 Contexte applicatif

Comme déjà mentionné, cet article s'inscrit dans la continuité d'un travail antérieur permettant, à partir d'un corpus de dialogues annotés, d'extraire un ensemble de couples de motifs dialogiques similaires (alignements locaux

intra-dialogue et inter-dialogues, valués par une mesure de similarité) [12].

Afin de la valider, cette méthode a été appliquée à un corpus de dialogues annotés, issus d'interactions entre parents et enfants d'âge variable. La grille utilisée pour coder les énoncés comporte 5 colonnes, et a pour but de permettre l'étude des caractéristiques dialogiques, sémantiques et pragmatiques du discours que proposent les adultes pour expliquer le comportement d'un personnage soumis à une fausse croyance [11]. Chaque colonne comporte entre deux et sept codages :

- Les annotations de la première colonne caractérisent la nature de l'énoncé. Ce dernier peut être une question (q, Q) ou une affirmation (a, A). L'état de majuscule (A, Q) dénote la présence d'un état mental. L'énoncé peut aussi correspondre à une demande d'attention sur l'histoire (AH) ou une demande d'attention générale (AG).
- La colonne suivante permet de coder la référence de l'énoncé. Ce dernier peut se référer à un personnage (P), à l'interlocuteur (H) ou au locuteur (R). Un codage n'est effectué dans cette colonne qu'en cas de présence d'un état mental.
- La troisième colonne permet de préciser les états mentaux. Les différentes annotations possibles sont l'émotion (E), la volition (V), la cognition observable (B) ou non observable (N), l'épistémie (K), l'hypothèse (Y) et la surprise (S). La surprise est ici distinguée des autres émotions en raison de 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 annotations de type parenthèses accolades et crochets ([, {, }, ]) sont utilisées pour signifier l'absence d'indice dans une colonne donnée.

La figure 6 présente un des motifs issus de la liste d'alignements effectués à partir d'APSA, sur le corpus de dialogues précédemment décrit. Dans cet exemple, l'avant dernier énoncé ("*pour lui une surprise en réalité.*") a été codé de la manière suivante. Cet énoncé est une affirmation concernant un personnage et sera par conséquent codée A et P dans les deux premières colonnes. L'état mental relatif au personnage de cet énoncé est la surprise, une annotation S apparaît donc en troisième colonne. Enfin l'expres-



Babar remarque Cornélius donner un paquer à l'intrus.	A	P	B	)	]
Qu'est-ce qui peut y avoir dedans ?	q	[	{	)	]
Mais qui est donc cet individu masqué	q	[	{	)	]
qui a volé la couronne ?	q	[	{	)	]
Babar la démasque !	A	P	B	)	]
C'est la reine Céleste.	a	[	{	)	]
Il se pose bien des questions.	A	P	N	)	]
Pourquoi donc la reine Céleste s'est déguisée ?	q	[	{	)	]
Babar va donc chez la vieille dame lui demander.	a		}	)	]
Oui.	a		}	)	]
La vieille dame ne veut pas qu'il rentre !	A	P	V	)	]
Mais derrière se cachait	A	P	B	O	J
pour lui une surprise en réalité.	A	P	S	O	J
Puis Babar rentra chez lui.	a	[	{	)	]

FIGURE 6 – Motif de référence issu du dialogue 1-b9, obtenu à l'aide du logiciel APSAD

sion "*en réalité*" dénote une justification par opposition permettant l'explication de l'histoire. Les deux dernières colonnes d'annotations de cet énoncé comportent donc les codages O et J.

L'extraction de régularités sur les dialogues annotés selon cette grille a permis de mettre en évidence un ensemble de motifs similaires apparaissant dans des paires de dialogues. Ce sont les motifs issus de ces alignements que nous avons utilisés comme références, à rechercher dans l'intégralité des dialogues, pour l'évaluation de notre méthode de reconnaissance de motifs approchés. Le but de cette démarche est de déterminer s'ils n'apparaissent que dans deux dialogues ou s'ils sont bien récurrents, et peuvent donc être intégrés à une banque de motifs.

## 5.2 Résultats

Les résultats figurant dans la suite de cet article correspondent à la recherche du motif de référence présenté figure 6. Les annotations ainsi que la matrice de substitution utilisés (voir figure 7) ont été conçus spécifiquement pour cette application, et contient des coûts dont la valeur varie entre 1 et 10. Le coût maximal  $k$  a été fixé empiriquement à 50. Cette valeur permet de limiter le nombre de substitutions les plus coûteuses au sein d'un alignement, tout en permettant une certaine diversité. Nous avons appliqué notre méthode sur les 70 dialogues annotés du corpus. Au total, 27 instances différentes du motifs ont été reportées (voir exemples figure 8 et figure 9). Le coût moyen obtenu pour ces instances est égal à 42.

Ces résultats sont encourageants, puisqu'ils permettent de confirmer le caractère récurrent de certains motifs dialogiques. Dans l'exemple donné, le motif correspond à une stratégie dialogique utilisée par les parents pour expliquer une fausse croyance à leur enfant.

## 6 Conclusion et perspectives

Nous avons présenté une méthode générique permettant l'alignement global d'un motif en deux dimensions sur une partie locale d'un texte. Cette méthode est utilisée dans notre cadre à la reconnaissance de motifs dialogiques dans un corpus de dialogues. Pour cela, chaque dialogue est codé à l'aide d'une matrice pour laquelle une ligne correspond à un énoncé et une colonne à un espace de codage.

Du point de vue algorithmique, plusieurs améliorations sont envisageables. Il serait tout d'abord judicieux d'adapter la méthode à l'alignement local de motifs. Ceci pourrait faciliter la détermination de motifs "consensuels" représentatifs d'un ensemble de motifs. En effet, il serait alors possible de ne conserver que les parties les plus représentatives d'un motif (*i.e.* celles qui apparaissent régulièrement dans les alignements locaux), ou bien seulement les parties communes à un ensemble de motifs. Pour réaliser des alignements locaux, il serait nécessaire de modifier l'algorithme de programmation dynamique, comme décrit dans [12], en passant de la minimisation d'un coût à la maximisation d'un gain. Cette modification ne sera pas immédiate puisqu'elle aura un impact sur la définition des zones de recherches d'une co-

	a	q	A	Q	G	D	P	R	H	C	O	M	J	F	E	B	N	V	K	Y	S
a	10																				
q	9	10																			
A	10	9	10																		
Q	9	10	9	10																	
G	2	8	2	8	10																
D	2	8	2	8	10	10															
P	0	0	0	0	0	0	10														
R	0	0	0	0	0	0	9	10													
H	0	0	0	0	0	0	7	8	10												
C	0	0	0	0	0	0	2	2	2	10											
O	0	0	0	0	0	0	2	2	2	9	10										
M	0	0	0	0	0	0	2	8	2	8	8	10									
J	0	0	0	0	0	0	7	4	3	1	1	2	10								
F	0	0	0	0	0	0	4	7	3	1	1	7	9	10							
E	0	0	0	0	0	0	5	7	5	3	3	8	2	4	10						
B	0	0	0	0	0	0	5	5	5	3	3	3	4	4	8	10					
N	0	0	0	0	0	0	5	5	5	3	3	3	4	4	8	10	10				
V	0	0	0	0	0	0	5	5	5	3	3	3	2	4	9	7	7	10			
K	0	0	0	0	0	0	7	5	5	3	6	3	4	4	8	7	8	7	10		
Y	0	0	0	0	0	0	5	5	5	4	3	3	4	4	8	7	8	7	9	10	
S	0	0	0	0	0	0	5	5	5	3	3	6	3	4	9	7	7	7	9	8	10

FIGURE 7 – Matrice de substitution dans le cas d’une maximisation des coûts

Donc là ils se cachent.	A	P	B	)	]
Ils cherchent	a	[	{	)	]
qui pourrait avoir pris la couronne ?	q	[	{	)	]
elle est dedans, elle est dedans la couronne.	a	[	{	)	]
Donc ils suspectent plein de monde, Cornélius, Céleste, la vieille dame.	A	P	Y	C	J
Qui a bien pu prendre la couronne ?	q	[	{	)	]
La couronne, elle est dedans.	a	[	{	)	J
Tu crois ?	Q	H	K	)	]
Oui.	a	[	{	)	]
Mais Babar il ne sait pas qu’elle est dedans.	A	P	N	O	J
Donc il se dit que c’est une bombe, la couronne	A	P	N	C	J
ou je ne sais quoi.	A	R	N	)	]
La couronne	a	[	{	)	]

FIGURE 8 – Alignement issu du dialogue 1-b7 (coût 25)

Regardez	A	P	B	)	]
C’est un bateau !	a	[	{	)	]
Il le prend	a	[	{	)	]
et il le pose sur l’eau.	a	[	{	)	]
Alors les cannetons tout contents montent dans le bateau,	A	P	E	C	J
Le bébé cygne voudrait bien monter dans le pot.	A	P	V	)	]
Et le pot coule.	a	[	{	)	]
Parce qu’il est trop gros,	A	P	{	C	J
alors la maman cygne arrive,	A	P	{	C	J
elle veut aller chercher le pot au fond de l’eau	A	P	V	)	]
elle veut le manger	A	P	V	)	]
non elle veut juste le rattraper mais avec son bec,	A	P	V	)	]
les cygnes n’ont pas de mains	a	[	{	)	]

FIGURE 9 – Alignement issu du dialogue 2-a2 (coût 25)

lonne à l'autre. Le temps d'exécution pourrait lui aussi être amélioré, en mettant en place une mémorisation des zones testées par l'algorithme en une dimension.

En revanche, certaines problématiques n'ont, à ce jour, pas trouvées de solutions pleinement satisfaisantes. L'ordre dans lequel les colonnes sont testées, par exemple, est présentement déterminé par le nombre d'annotations non vides que contient le motif dans chacune d'entre elles. Ceci ne permet pas une minimisation systématique du nombre de candidats. De même, le paramètre  $k$ , qui représente le nombre maximal d'erreurs, est actuellement fixé manuellement, de manière empirique. Il sera important, à terme, qu'il soit défini automatiquement en fonction de paramètres tels que la taille ou le contenu du motif considéré. Enfin, une gestion de la cohérence intra-colonnes des alignements doit être mise en place. Nous définissons, actuellement, pour chaque colonne une zone dans laquelle le motif sera recherché. Cependant la position d'un alignement dans cette zone n'influence pas le coût des solutions. Une méthode permettant de pénaliser les alignements, dont la forme s'éloigne de celle du motif de référence, reste à déterminer. Nous pourrions, par exemple, ajouter à la somme des distances d'édits des colonnes, un coût représentatif de la déformation du motif.

À moyen terme, ce projet pourrait permettre, d'une manière similaire à BLAST [2], la création d'une banque de motifs sur laquelle il serait rapidement possible d'effectuer des alignements. Cela offrirait la possibilité à un agent conversationnel intégrant notre heuristique de reconnaissance de motifs, d'accomplir des analyses de dialogues en temps réel. Durant un échange, l'agent pourrait réaliser dans un premier temps une phase d'identification des caractéristiques dialogiques de son interlocuteur, en fonction des motifs utilisés. Cette opération serait effectuée en alignant les annotations du dialogue courant avec un sous-ensemble de motifs de la banque. Au fil des échanges, le nombre de motifs alignés augmentant, l'analyse s'en trouvera affinée. Ainsi, l'agent pourrait adapter son comportement en vue de répondre le plus efficacement possible à sa fonction.

## Références

[1] J. Allen and M. Core. DAMSL : Dialog act markup in several layers. *Draft of manual*, 1997.

- [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 and E. Chencinski. Faster two dimensional scaled matching. *Algorithmica*, 56(2) :214–234, 2010.
- [4] A. Amir and M. Farach. Efficient 2-dimensional approximate matching of non-rectangular figures. In *Proceedings of the second annual ACM-SIAM symposium on Discrete algorithms*, pages 212–223. Society for Industrial and Applied Mathematics, 1991.
- [5] M. Amir Gad et al. Fast parallel and serial multidimensional approximate array matching. *Theoretical Computer Science*, 81(1) :97–115, 1991.
- [6] N. Asher. *Reference to abstract objects in discourse*. Kluwer Academic Publishers, Dordrecht, 1993.
- [7] G. Baeza-Yates, R. & Navarro. Fast two-dimensional approximate pattern matching. In Cláudio Lucchesi and Arnaldo Moura, editors, *LATIN'98 : Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 341–351. Springer Berlin / Heidelberg, 1998.
- [8] R.A. Baeza-Yates and M. Régnier. Fast two-dimensional pattern matching. *Information Processing Letters*, 45(1) :51–57, 1993.
- [9] R. S. Bird. Two-dimensional pattern matching. *Information Processing Letters*, 6(5) :168–170, October 1977.
- [10] H. Bunt. The DIT++ taxonomy for functional dialogue markup. In *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 13–24, 2009.
- [11] É. Chanoni. Comment les mères racontent une histoire de fausses croyances à leur enfant de 3 à 5 ans ? *Enfance*, (2), 2009.
- [12] É. Chanoni, T. Lecroq, and A. Pauchet. Extraction de motifs dans des dialogues annotés par programmation dynamique. In PY. & Guyomard M. Maudet, N. & Schobben, editor, *Modèles Formels de l'Interaction (MFI'09)*, pages 101 – 112, 2009.

- [13] E.F. Clark, H.H. & Schaefer. Contributing to discourse\* 1. *Cognitive science*, 13(2) :259–294, 1989.
- [14] H. Clark. *Using Language*. Cambridge University Press, Cambridge, 1996.
- [15] J. Moeschler C. Rubattel & M. Schelling E. Roulet, A. Auchlin. L'articulation du discours en français. *Berne, Peter Lang*, page 272, 1985.
- [16] J.F. Ferguson, G. & Allen. TRIPS : An integrated intelligent problem-solving assistant. In *Proceedings of the National Conference on Artificial Intelligence*, pages 567–573. JOHN WILEY & SONS LTD, 1998.
- [17] J. F. Allen & Bradford W. Miller Ferguson G. TRAINS-95 : Towards a mixed-initiative planning assistant. In *AIPS*, pages 70–77, 1996.
- [18] K. Fredriksson, G. Navarro, and E. Ukkonen. Optimal exact and fast approximate two dimensional pattern matching allowing rotations. In *Combinatorial Pattern Matching*, pages 235–248. Springer, 2002.
- [19] C. L. Grosz, B. J. & Sidner. Plans for discourse. In J. L. Morgan & M. E. Pollack P. R. Cohen, editor, *Intentions in Communication*. Bradford Books at MIT Press, 1990.
- [20] C.L. Grosz, B.J. & Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3) :175–204, 1986.
- [21] R. Krithivasan et al. Efficient two-dimensional pattern matching in the presence of errors\* 1. *Information Sciences*, 43(3) :169–184, 1987.
- [22] S. Larsson. *Issue-Based Dialogue Management*. Ph.d. dissertation, Department of Linguistics, Göteborg University, Göteborg, 2002.
- [23] S. Larsson and D.R. Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering*, 6(3&4) :323–340, 2000.
- [24] S.A. Mann, W.C. & Thompson. Rhetorical structure theory : Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3) :243–281, 1988.
- [25] N. Maudet. *Modéliser les conventions des interactions langagières : la contribution des jeux de dialogue*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, mai 2001.
- [26] J. A. Levin & J. A. Moore. Dialogue-games : Meta-communication structures for natural language interaction. *Cognitive Science*, 1(4) :395–420, 1977.
- [27] G. Navarro and R. Baeza-Yates. Fast multi-dimensional approximate pattern matching. In *Combinatorial Pattern Matching*, pages 243–257. Springer, 1999.
- [28] J. F. Allen & C. Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15(3) :143–178, 1980.
- [29] C.L. Rich, C. & Sidner. COLLAGEN : A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3) :315–350, 1998.
- [30] D. Searle, J.R. & Vanderveken. *Foundations of illocutionary logic*. Cambridge Univ Pr, 1985.
- [31] J.R. Searle. *Speech acts : An essay in the philosophy of language*. Cambridge university press, 1970.
- [32] P. H. Sellers. The theory and computation of evolutionary distances : Pattern recognition. *Journal of Algorithms*, 1(4) :359 – 373, 1980.
- [33] Krithivasan & Sitalakshmi. Efficient two-dimensional pattern matching in the presence of errors. *INFSCIENCES : Information Sciences*, 43, 1987.
- [34] D.R. Traum. *A computational theory of grounding in natural language conversation*. PhD thesis, University of Rochester, 1994.
- [35] D. Vanderveken. Illocutionary logic and discourse typology. *Revue internationale de philosophie*, (2) :243–255, 2001.