Extraction de motifs dialogiques bidimensionnels

Zacharie Alès, Alexandre Pauchet, Arnaud Knippel, Laurent Vercouter, Christian Gout

Extraction de motifs dialogiques bidimensionnels

Zacharie Ales[1,2] Alexandre Pauchet[1] Arnaud Knippel[2] Laurent Vercouter[1] Christian Gout[2]

[1] LITIS, Normandie Université, INSA Rouen (EA 4108)
[2] LMI, Normandie Université, INSA Rouen (EA 3226)

zacharie.ales@insa-rouen.fr

Résumé

Cet article aborde le problème de l'extraction de régularités dans des dialogues sous la forme de motifs dialogiques. Nous présentons un algorithme de programmation dynamique en $O(n^3)$ permettant d'extraire de tableaux bidimensionnels d'annotations représentant des dialogues, des motifs récurrents. Cet algorithme, combiné à une méthode de clustering permet d'obtenir des régularités caractéristiques d'un corpus annoté. Les paramètres de la méthode sont évalués par tests statistiques.

Mots Clef

Extraction de régularités, motifs dialogiques, modélisation du dialogue.

Abstract

This article addresses the problem of regularity extraction in dialogues in the shape of dialogical patterns. We present a dynamic programming algorithm which runs in $O(n^3)$ which enables to extract patterns from two-dimensional dialogue annotations. We show how it can be combined to a clustering heuristique in order to extract relevant regularities from an annotated corpus. The parameters of the method are evaluated thanks to statistical tests.

Keywords

Regularity extraction ; Dialogical patterns ; Dialogue modeling.

1 Introduction

Plus de quarante ans après la création du premier chatterbot ELIZA, un système de dialogue permettant d'interagir naturellement avec des humains reste encore à développer. Actuellement, lors d'échanges avec des agents conversationnels, l'humain doit s'adapter à des schémas rigides et des gestions du dialogue linéaires. Les systèmes de réponses vocales interactives, par exemple, restreignent les interactions humaines à des mots clés énoncés sur des plages de temps bien spécifiées. En d'autres termes, le dialogue est dirigé par le système.

Le dialogue étant une activité typiquement humaine faite de régularités, nous postulons que des motifs récurrents peuvent être extraits d'un corpus et constituer une base sur laquelle pourra s'appuyer un modèle de dialogue performant. Dans cette optique, nous proposons une méthodologie, présentée en figure 1. Selon cette approche, des dialogues textuels – composés de retranscriptions de dialogues et potentiellement d'informations multimodales (attitude, expression faciale, prosodie, ...) – sont tout d'abord obtenus via une étape de transcodage. La représentation du dialogue est alors complétée par des informations additionnelles durant la phase d'annotation, effectuée manuellement ou automatiquement. Enfin, les régularités sont détectées et utilisées en tant que base pour créer un modèle de dialogue robuste. Ceci est effectué en extrayant tout d'abord des motifs dialogiques puis en les regroupant via une étape de partitionnement.

La représentation du dialogue et l'extraction de régularités sont deux points clés de cette méthodologie. Comme le souligne Bunt, la gestion du dialogue implique des aspects multidimensionnels [1]. Cette multidimensionnalité complique la tâche d'extraction de régularités dans des dialogues textuels. Pour pallier cela, nous proposons de représenter un dialogue par un tableau d'annotations.

La suite de l'article est organisé comme suit. La section suivante est dédiée à un état de l'art sur l'extraction de régularités dans divers domaines. La représentation bidimensionnelle des annotations de dialogues est décrite section 3.

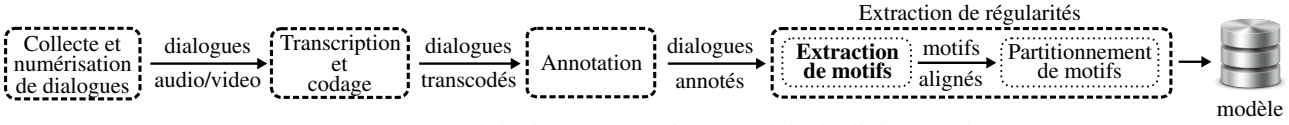

FIGURE 1 – Méthodologie pour obtenir un modèle de dialogue robuste.

En section 4, l'algorithme permettant d'en extraire des motifs récurrents est présenté et le choix de la méthode de clustering est explicité. Les résultats obtenus sont décrits en section 5. Enfin, la section 6 est dédiée à la conclusion.

2 Etat de l'art

2.1 Extraction de régularités dans du texte

L'extraction de régularités à partir de textes en langage naturel est un problème qui a été étudié de manière intensive. Dans le domaine de l'extraction d'informations, de plus en plus de systèmes sont basés sur un dictionnaire linguistique de motifs. Le contenu de tels dictionnaires est très dépendant du domaine considéré et leur création se révèle extrêmement chronophage. C'est pourquoi diverses méthodes ont été développées afin d'extraire automatiquement des motifs pertinents d'un corpus.

Une des premières, appelée AutoSlog [2], utilise diverses heuristiques pour spécialiser des motifs syntaxiques, tels que <*sujet*> <*verbe passif*>, afin d'extraire des phrases pertinentes d'un corpus. Crystal [3] est une méthode similaire permettant d'obtenir des motifs plus complexes grâce à une étape de généralisation des motifs. Yangarber et al [4] utilisent un mécanisme de généralisation similaire, mais leur approche nécessite moins de prérequis puisque l'utilisateur doit simplement fournir quelques motifs appelés "seeds". Sudo et al [5] utilisent des motifs syntaxiques plus souples puisqu'ils s'intéressent à des textes en japonais dans lesquels l'ordre des mots est plus flexible. Pour ce faire, ils utilisent une structure d'arbre pour représenter les motifs, dans laquelle un noeud représente une unité langagière (exemple : verbe, sujet, entreprise, ...) et une arête correspond à une dépendance entre deux noeuds.

Un désavantage de ces méthodes d'extraction d'information est que les motifs considérés ne peuvent couvrir plus d'une phrase. La portée des motifs est donc restreinte. Ainsi, des régularités du type "<question><réponse>" ne peuvent être identifiés à l'aide de ces techniques.

D'Mello et at [6] s'intéressent à des dialogues entre étudiants et tuteurs. Les interventions du tuteur sont annotées selon un schéma de codage contenant 27 annotations (*e.g.* : instruction, explication, indice, ...) et 16 annotations sont utilisées pour l'étudiant (*e.g.* : réponse correcte, question, ...). Un graphe de transition entre les différentes annotations est créé puis des motifs en sont extraits sous forme de chemins, cycles et circuits.

Bien que performantes, ces méthodes sont spécifiques à l'analyse de textes et ne se prêtent pas à l'extraction de données bidimensionnelles.

2.2 Extraction de régularités dans des données non textuelles

De nombreux efforts ont été fournis dans le domaine de la reconnaissance de motifs dans des dialogues audio. Barzilay et al [7] tentent d'identifier des phrases récurrentes permettant de déterminer le locuteur. Ils utilisent des méthodes d'apprentissage statistique sur l'ensemble des n-grams d'un corpus d'apprentissage pour $n \in [\![1, 5]\!]$.

Rao et al [8] considèrent des systèmes numériques à grande échelle, tels que des compilateurs, qu'ils souhaitent réduire à une collection de sous systèmes appelés *templates*. L'objectif double consiste à minimiser le nombre de templates différents ainsi que le nombre total de templates utilisés pour représenter le système. Ces objectifs relativement éloignés des nôtres puisque nous ne souhaitons pas représenter l'ensemble d'un dialogue par des motifs ni minimiser le nombre de motifs extraits.

Dans le domaine de la fouille de données, l'extraction de motifs récurrents est une tâche courante. Comme le soulignent Han et al [9], trois types de motifs peuvent y être considérés, à savoir : itemsets, séquences d'itemsets et motifs structurels. Les deux premiers types sont trop contraignants sur la forme que peuvent prendre les motifs. Bien que les motifs structurels soient en général recherchés de manière exacte, le système SubDue [10] permet d'en extraire des approchés. Cependant, cette méthode utilise le principe de MDL (*Minimum Description Length*) ce qui tend à extraire les motifs les plus grands et pas nécessairement les plus récurrents.

A notre connaissance, le problème de l'extraction de régularités dans des annotations en deux dimensions n'a à ce jour pas été étudié. Nous proposons, dans la section suivante, une nouvelle méthode adaptée à ce type de problèmes.

Locuteur	**Énoncé**	**Annotations**			
P	C'est la couronne,	-	-	-	-
P	mais Babar ne l'a pas vu.	P	OC	O	ES
P	Elle est cachée derrière la porte,	-	OC	-	-
P	mais tu la vois, c'est bien.	I	OC	O	PC
E	Oui je la vois.	L	-	-	-

TABLE 1 – Extrait d'un dialogue annoté issu du corpus de dialogues considéré

3 Représentation bidimensionnelles des annotations de dialogues

Un dialogue annoté est composé d'une série d'énoncés ordonnés chronologiquement. Chaque énoncé est caractérisé par un vecteur d'annotations, dont les composantes correspondent aux différentes dimensions de codage. Chaque dimension comporte son propre alphabet.

Dans cet article, nous considérons un corpus de dialogues annotés entre un parent et son enfant lors de la narration d'une histoire. Dans l'extrait présenté en table 1, chaque énoncé est caractérisé par son locuteur, une transcription ainsi que son encodage suivant quatre colonnes :

– La première colonne est dédiée à la réferenciation de l'énoncé (P : au personnage, I : à l'interlocuteur, L : au

locuteur) ;
- La seconde colonne encode les états mentaux (E : émotion, V : volition, S : surprise, OC et NOC : respectivement cognition observables et non observables) ;
- Les deux dernière colonnes sont consacrées aux justifications (O : par opposition, C : par cause / CH : pour expliquer l'histoire, PC : pour expliquer une situation par l'évocation d'un contexte personnel).

Le caractère '-' est utilisé pour représenter l'absence d'annotation dans un énoncé pour un axe de codage donné.
Une fois le schéma de codage déterminé et l'annotation d'un corpus de dialogues effectuée, la recherche de motifs dialogiques récurrents peut alors être initiée. Dans cet article, nous définissons un motif récurrent comme un ensemble d'annotations figurant, de manière exacte ou approché, dans plusieurs dialogues du corpus considéré. L'approximation entre deux motifs peut intervenir au niveau des annotations en elle même (*e.g.* : faire correspondre l'annotation V avec l'annotation S) ainsi qu'au niveau du positionnement des annotations (*i.e.* : sauts de lignes ou décalages d'un groupe d'annotations) .

4 Extraction de motifs dialogiques

Soit $\mathcal{C} = \{d_i\}_{i=1}^{n}$ un corpus composé des annotations de n dialogues. L'extraction de motifs récurrents de $\mathcal{C}$ est effectuée via l'extraction d'alignements locaux entre toutes les paires (d_i, d_j), $i, j \in [\![1, n]\!]$.
Soient s_i une sous-partie de d_i et s_j une sous-partie de d_j. Le couple (s_i, s_j) définit un alignement si sa similarité - selon une métrique donnée - est significative (*i.e.* : au dessus d'un seuil choisi). Un alignement est *global* si $s_i = d_i$ et $s_j = d_j$. Dans le cas contraire, l'alignement est dit *local*.
De nombreuses méthodes ont été développées afin d'aligner des séquences ADN ; des algorithmes approchés tels que BLAST [11] et FAST [12] permettent d'aligner rapidement une séquence choisie sur une base de données de séquences. Parmi les méthodes exactes, une fonction de similarité dérivée de la *distance de Levenshtein* est souvent utilisée.
Dans la section suivante, nous décrivons comment des alignements locaux optimaux peuvent être extraits de séquences unidimensionnelles grâce à la distance de Levenshtein. Par la suite, une généralisation de ces mécanismes est utilisée pour extraire heuristiquement des alignement locaux de séquences bidimensionnelles. Enfin, une adaptation originale - spécifique à l'extraction de motifs d'annotations - permettant d'améliorer les temps de calculs est détaillée.

4.1 Alignement local de séquences

Soient deux séquences de caractères e_1 et e_2, de taille respectives m_1 et m_2. La distance de Levenshtein $ed(e_1, e_2)$ entre ces deux séquences est égale au nombre minimum d'opérations d'édition (insertion, suppression et substitution) permettant de transformer e_1 en e_2. Afin de calculer cette distance, un coût de 1 est associé à chaque opération d'édition, exceptée la substitution d'un caractère par lui même dont le coût est fixé à 0. Plus généralement, lorsque chaque opération d'édition possède son propre coût, le terme de *distance d'édition* est utilisé. D'après la définition précédente, un alignement peut être vu comme une superposition (contenant potentiellement des espaces) de e_1 et e_2 (voir exemples en figure 2 (B)). L'algorithme de Needleman-Wunsch utilisant la programmation dynamique peut être utilisé de manière efficace pour calculer des distances d'édition. Une table T de taille $(m_1 + 1) \times (m_2 + 1)$ est tout d'abord calculée de telle sorte que $T[i][j]$ soit égale à la distance d'édition entre $e_1[1..i]$ et $e_2[1..j]$, $\forall (i, j) \in [\![1, m_1]\!] \times [\![1, m_2]\!]$. La distance d'édition entre e_1 et e_2 est ainsi obtenue en $T[m_1][m_2]$. Cet algorithme est basé sur l'idée que la dernière opération d'édition d'un alignement est soit une substitution, soit une insertion, soit une suppression. La distance d'édition entre "ATCA" et "ATGC" peut, par exemple, être écrite

$$ed(ATGC, ATCA) = min \begin{cases} ed(ATG, ATC) & +sub(C, A) \\ ed(ATG, ATCA) & +sup(C) \\ ed(ATGC, ATC) & +ins(A) \end{cases}$$

Cette observation mène naturellement à une formule de récurrence contenant trois termes à minimiser. La table T obtenue lors du calcul de la distance de Levenshtein sur l'exemple précédent est présenté en figure 2 (A).

Lors d'une seconde phase, un tracé arrière de $T[m_1][m_2]$ à $T[0][0]$, inférant les caractères de l'alignement, est réalisé. Si plusieurs chemins mènent à $T[m_1][m_2]$, chacun d'entre eux donne un alignement différent (voir exemple en figure 2 (B)). Ces alignements sont globaux, puisque l'ensemble des caractères de e_1 et e_2 y figurent.

Afin d'extraire des alignements locaux de séquences, l'algorithme de Smith-Waterman, une variante de l'algorithme précédent, peut être utilisé. Les coûts des opérations d'édition y sont remplacés par des scores dans $\mathbb{Z}$. Au cours de l'algorithme, les valeurs de T négatives sont fixées à 0. Ainsi, T ne contient plus des distances globales mais des similarités locales. L'alignement local optimal est obtenu en réalisant un tracé arrière depuis la position de T contenant la plus grande valeur (et non plus depuis $T[m_1][m_2]$) jusqu'à ce qu'une position de score nul soit atteinte.

	A	T	G	C
A	0	1	2	3
T	1	0	1	2
C	2	1	1	1
A	3	2	2	2

(A)

$$\begin{pmatrix} A & T & G & C \\ A & T & C & A \end{pmatrix}$$

$$\begin{pmatrix} A & T & G & C & - \\ A & T & - & C & A \end{pmatrix}$$

(B)

FIGURE 2 – Résultat de l'algorithme de Needleman-Wunsch, lorsque les coûts de la distance de Levenshtein sont utilisés sur les séquences "ATCA" et "ATGC". **(A)** Table T (sans sa première ligne et sa première colonne). La case grise contient la distance de Levenshtein. **(B)** Alignements optimaux correspondants.

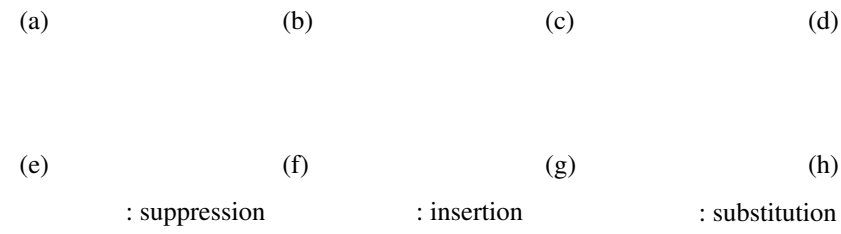

FIGURE 3 – Représentation des huit opérations d'édition bidimensionnelles considérées, appliquées à deux tableaux de tailles respectives 2×3 et 3×2. La position courrante de l'algorithme est représentée par un point et les caractères impactés par l'opération d'édition sont grisés. **(a)** : Suppression de ligne. **(b)** : Suppression de colonne. **(c)** : Insertion de ligne. **(d)** : Insertion de colonne. **(e)** : Substitution de ligne. **(f)** : Substitution de colonne. **(g)** et **(h)** : Substitution de ligne et de colonne (dans un ordre différent).

4.2 Extraction d'alignements locaux bidimensionnels

Une adaptation de l'algorithme de Smith-Waterman à des données bidimensionnelles a récemment été développée [13]. Bien que contraignant légèrement la forme des motifs, cet algorithme possède une complexité relativement faible ($O(n^4)$), faisant de lui le meilleur candidat - à notre connaissance - pour l'extraction de motifs récurrents dans des tableaux bidimensionnels.

Etant donnés deux tableaux d_1 et d_2 de tailles respectives $m_1 \times n_1$ et $m_2 \times n_2$, cet algorithme calcule une table quadridimensionnelle T de taille $(m_1+1) \times (n_1+1) \times (m_2+1) \times (n_2+1)$ telle que $T[i][j][k][l]$ soit égal à la similarité locale entre $d_1[1..i][1..j]$ et $d_2[1..k][1..l]$ pour tout $(i,j,k,l) \in [\![1,m_1]\!] \times [\![1,n_1]\!] \times [\![1,m_2]\!] \times [\![1,n_2]\!]$. Afin de calculer T, deux tables R et C, de mêmes dimensions que T sont calculées en utilisant l'algorithme de Smith-Waterman de telle sorte que :

- $R[i][j][k][l]$ soit égal à la similarité locale entre $d_1[i][0..j]$ et $d_2[k][0..l]$;
- $C[i][j][k][l]$ soit égal à la similarité locale entre $d_1[0..i][j]$ et $d_2[0..k][l]$.

La table T est ensuite calculée en considérant huit opérations d'édition bidimensionnelles (représentées en figure 3). Par soucis de simplification, notons :

- $R[i][j][k][l]$ et $R[i\text{-}1][j][k\text{-}1][l]$ respectivement r et r' ;
- $C[i][j][k][l]$ et $C[i][j\text{-}1][k][l\text{-}1]$ respectivement c et c' ;
- $\forall x \in \mathbb{R}, q(x) = \begin{cases} x & \text{si } x \neq 0 \\ sup(d_1[i][j]) + ins(d_2[k][l]) & \text{sinon} \end{cases}$.

La formule de récurrence permettant de calculer $T[i][j][k][l]$ peut alors s'écrire :

$$\min \begin{cases} T[i-1,j,k,l] + ins(d_1[i][j]) & (a) \\ T[i,j-1,k,l] + ins(d_1[i][j]) & (b) \\ T[i,j,k-1,l] + sup(d_2[k][l]) & (c) \\ T[i,j,k,l-1] + sup(d_2[k][l]) & (d) \\ T[i-1,j,k-1,l] + q(r) & (e) \\ T[i,j-1,k,l-1] + q(c) & (f) \\ T[i-1,j-1,k-1,l-1] + q(r'+c) & (g) \\ T[i-1,j-1,k-1,l-1] + q(r+c') & (h) \\ 0 \end{cases}.$$

Chaque ligne de l'expression précédente correspond à une direction (représentée en figure 3) de l'algorithme de programmation dynamique. La dernière ligne du système permet de commencer un nouvel alignement local en position (i,j,k,l) lorsque le score des huit directions est négatif. Pour une description plus complète de cet algorithme, le lecteur peut se referer à [13].

Bien que la complexité de cet algorithme soit satisfaisante, le calcul de T sur l'ensemble des paires d'annotations de dialogues augmente rapidement avec la taille du corpus et des dialogues. Afin d'améliorer le temps d'exécution, nous présentons une adaptation de cette méthode spécifique au type de données considérées.

4.3 Alignement d'annotations de dialogues

La représentation des annotations de dialogues suit des règles spécifiques dont il est possible de tirer profit afin de réduire le temps d'exécution. Comme nous l'avons présenté (table 1), chaque colonne d'annotation correspond à une dimension de codage indépendante possédant son propre alphabet. L'alignement de deux annotations figurant dans des colonnes différentes n'est donc pas pertinent. En conséquence, l'insertion et la suppression de colonnes (opérations d'édition (c) et (d) de la figure 3) ne sont pas à considérer. Le nombre de termes dans la formule de récurrence est ainsi diminué de deux. Ainsi, deux annotations $a_1 \in d_1$ et $a_2 \in d_2$ alignées ensembles seront nécessairement dans des colonnes de même numéro. Les dimensions deux et quatre de T, indiquant respectivement le numéro de colonne dans d_1 et d_2, sont donc redondantes. En conséquence, la dimension de T peut être réduite de un et $T[i][j][k]$ correspond ainsi à la similarité locale entre $d_1[1..i][1..j]$ et $d_2[1..k][1..j]$.

Après avoir calculé T, similairement à l'algorithme unidimensionnel, un tracé arrière, permettant d'obtenir le meilleur alignement local, est réalisé à partir de la position contenant le score le plus élevé. Cependant, il est envisageable que deux dialogues annotés contiennent plusieurs sous-parties similaires, chacune capable de fournir un alignement pertinent. Nous avons donc adapté la phase de tracé arrière afin que l'algorithme retourne un ensemble or-

donné des sous-parties similaires. L'ensemble des positions de T dont le score est au-dessus d'un seuil τ, fixé par l'utilisateur, sont étiquetées comme positions candidates pour un tracé arrière. Il serait possible d'obtenir un alignement par position candidate. Cependant, ceci retournerait un nombre extrêmement élevé d'alignements redondants (et entraînerait une phase de tracés arrière coûteuse en temps de calcul). Ce phénomène est illustré figure 4. Les deux premiers éléments (A) et (B) de la figure représentent des sous-parties similaires de deux dialogues. Dans cet exemple, chaque opération d'édition possède un score d'édition de -1 excepté $sub(A, A)$ et $sub(B, B)$ dont le score est fixé à 1. Ainsi, les substitutions de A et des trois B donnent lieu à un score localement optimal de 4. Le chemin dans T suivi par l'algorithme pour atteindre ce score correspond à quatre substitutions allant de (i, j, k) à $(i+3, j+1, k+3)$ et peuvent être représentées de la manière suivante :

$$(i,j,k) \overset{sub(A,A)}{\longrightarrow} (i{+}1,j,k{+}1) \overset{sub(B,B)}{\longrightarrow} (i{+}1,j{+}1,k{+}1)$$

$$\overset{sub(B,B)}{\longrightarrow} (i+2,j+1,k+2) \overset{sub(B,B)}{\longrightarrow} (i+3,j+1,k+3)$$

Pour chacune de ces positions de $T[r_1][r_2][r_3]$, r_1 et r_3 vérifient la relation $r_3 = r_1 + k - i$. Le chemin suivi peut donc se representer en deux dimensions comme le montre la figure 4 (C). Dans ce contexte, si le score minimum τ est fixé à 3, la position de score maximum $(i+3, j+1, k+3)$ est bien candidate. Cependant certaines positions non désirées figurant à l'intérieur de l'alignement (*e.g.* $(i{+}2, j{+}1, k{+}3)$), ou autour (*e.g.* $(i{+}3, j{+}2, k{+}3)$) sont elles aussi candidates. Ces positions (représentées par une étoile dans la figure 4) correspondent à de légères variations de l'alignement localement optimal qu'il est nécessaire de filtrer. Pour cela, nous utilisons une heuristique simple basée sur une structure de données de type union-find.

Enfin, un alignement local est obtenu par tracé arrière à partir de chaque position candidate. Ces alignements sont composés de deux ensembles d'annotations (un pour chaque dialogue) qui correspondent à deux motifs différents. En appliquant cette méthode à chaque paire de dialogues un ensemble de motifs est obtenu.

4.4 Clustering de motifs

L'algorithme d'extraction présenté précédemment permet d'obtenir des motifs apparaissant dans au moins deux dialogues. Afin d'identifier les motifs les plus récurrents et caractéristiques du corpus, nous réalisons une seconde étape (comme présenté figure 1) durant laquelle les motifs extraits sont partitionnés en clusters de motifs.

Le choix de la méthode de clustering est un point délicat que nous avons traité lors d'un précédent travail [14]. Cinq méthodes de clustering utilisant des mécanismes variés (projection, voisinage, connectivité, ...) ont été étudiées. L'heuristique dénommée Rock [15], fournissant les résultats les plus prometteurs, a été retenue.

	Couple de valeurs $(36, 38)$	$(36, 40)$	$(38, 40)$
Le test est-il satisfait ?	oui	non	oui
En faveur de quelle valeur ?	38	-	38

TABLE 2 – Résultats des tests WSR sur chaque paire de valeurs de τ.

5 Evaluation

Le corpus de dialogue considéré dans cet article comporte 73 dialogues annotés entre un parent et son enfant. Le nombre d'énoncés par dialogue varie de 28 à 249 pour une moyenne de 82. Nous souhaitons évaluer l'influence des deux principaux paramètres de notre méthode. Le premier paramètre est le score minimal τ à partir duquel un score d'alignement dénote un motif. Ce paramètre impacte directement le nombre de motifs extraits. Trois valeurs de τ (36, 38 et 40) – retournant entre 120 et 394 motifs – ont été sélectionnées heuristiquement. La seconde variable évaluée est le nombre de clusters (noté k) obtenus par ROCK. Ce paramètre influence la granularité des résultats et donc la qualité des régularités obtenues. Trois valeurs de k de magnitude différente sont étudiées (5, 20 et 120). Pour chaque combinaison de valeurs de τ et k, un psychologue familier du corpus et expert de la psychologie de l'enfant, a donné une note entre 0 et 4 aux deux critères suivants : pertinence des clusters de motifs et pertinence du nombre de clusters. Le premier critère permettra d'évaluer τ tandis que le second sera utilisé pour k.

L'évaluation est effectuée grâce à des tests des rangs signés de Wilcoxon (WSR). Ce test permet d'affirmer si la moyenne de deux populations associées diffère. Le cas échéant, il existe un écart significatif entre les deux populations considérées et celle dont la moyenne est la plus haute est meilleure.

Les résultats obtenus pour le paramètre τ figure en table 2. La valeur moyenne de τ (38) est la meilleure. Contrairement à ce qu'on pourrait intuitivement penser, le valeur du paramètre retournant le plus de motifs (36) n'est pas celle qui donne le meilleur score. Selon l'expert, un nombre trop élevé de motifs complique l'extraction de sens des cluters de motifs.

La table 3 présente les résultats obtenus en fonction du nombre de clusters. On constate ici que les meilleurs résultats sont obtenus avec vingt clusters. En effet, une faible valeur de k tend à regrouper trop de motifs ensemble tandis que si k est trop élevé, de nombreux motifs similaires sont séparés.

Ces résultats montrent que les paramètres k et τ ont une influence forte sur la qualité des régularités extraites et que leur choix doit être réalisé avec précaution.

De plus, lors de cette expérience, des régularités interprétables par l'expert et que des extractions manuelles précédentes n'avaient pas permis d'identifier ont été extraites. Par exemple, les motifs représentés figure 5 ont été ex-

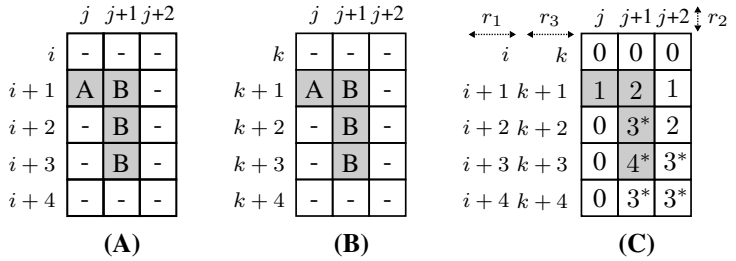

FIGURE 4 – **(A)** et **(B)** Deux annotations de dialogues. **(C)** Extrait des valeurs de la table $T[r_1][r_2][r_3]$ correspondante, sur l'intervalle $r_1 \in [\![i, i+4]\!], r_2 \in [\![j, j+2]\!], r_3 \in [\![k, k+4]\!]$ et tels que r_1 et r_3 sont liés par la relation $r_3 = r_1 + k - i$. Des étoiles sont utilisées pour représenter les positions candidates de T pour un score minimal τ de trois. Les cases grisées représentent les alignements locaux optimaux et leur chemin dans T.

	Couple de valeurs		
	$(5, 20)$	$(5, 120)$	$(20, 120)$
Le test est-il satisfait ?	oui	oui	oui
En faveur de quelle valeur ?	20	5	20

TABLE 3 – Résultats des tests WSR sur chaque paire de valeurs considérées pour k.

trait via l'algorithme de programmation dynamique puis regroupés ensembles par la méthode Rock. Selon l'expert, ce cluster correspond à une stratégie mise en oeuvre par les parents pour fournir une explication mentaliste. Afin d'exprimer les sentiments d'un personnage, l'état mental est tout d'abord exprimé (annotations E ou V). Ce dernier est ensuite expliqué par deux justifications dont la première contient un état mental proche voire identique (*e.g.* : "il *pleure*", "parce qu'il est *en colère*").

6 Conclusion et perspectives

Nous avons présenté une méthodologie en deux étapes permettant d'extraire des régularités à partir d'annotations bidimensionnelles de dialogues. Des alignements de motifs représentatifs du corpus sont, tout d'abord, extraits par un algorithme de programmation dynamique, possédant une faible complexité. Puis, des régularités sont obtenues en aggrégeant les motifs par une méthode de clustering. L'approche décrite ici est heuristique de par la complexité inhérente au problème et la taille des données étudiées.

Nous avons, de plus, montré que les paramètres τ et k de la méthode doivent être choisis avec soin puisqu'ils ont une influence directe sur la qualité des régularités extraites.

Les alignements de motifs obtenus avec notre méthode d'extraction sont influencés par l'ordre des colonnes d'annotations. La suppression de cette dépendance permettrait d'obtenir des motifs plus pertinents et pourrait, notamment, être effectuée en augmentant le nombre de directions considérées par l'algorithme de programmation dynamique.

Enfin, la qualité des régularités obtenues pourrait être optimisée, en résolvant, par exemple, le problème de clustering par une approche de programmation linéaire en nombres entiers fournissant un optimum global.

Enoncé	**Annotations**
Il pleure.	P E - -
Parce qu'il est en colère.	P E C CH
Parce qu'il est pas content.	P E C CH

Enoncé	**Annotations**
Leo veut la pelle de Thimothée.	P V - -
Mais Thimothée veut pas lui donner.	P V O CH
C'est parce qu'il a pas de pelle.	P - C CH

Enoncé	**Annotations**
Oh il est en colère.	P E - -
Parce que son ami veut pas lui donner la pelle.	P V C CH
Donc il donne un gros coup de pied dans le château du copain !	P - C CH

FIGURE 5 – Trois motifs extraits qui ont été regroupés ensemble par l'heuristique de partitionnement Rock.

Remerciements

Nous exprimons notre gratitude au projet PRIMO de l'Institut National des Sciences Appliquées de Rouen (INSA de Rouen) ainsi qu'à l'ANR-13-CORD-0015 dans le cadre desquels ces travaux ont été réalisés.

Références

[1] H. Bunt, "Multifunctionality in dialogue," *Comput. Speech and Lang.*, vol. 25, no. 2, pp. 222–245, 2011.

[2] E. Riloff *et al.*, "Automatically constructing a dictionary for information extraction tasks," in *NCAI*, pp. 811–811, JOHN WILEY & SONS LTD, 1993.

[3] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert, "Crystal : Inducing a conceptual dictionary," in *14th IJCAI*, pp. 1314–1319, 1995.

[4] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Unsupervised discovery of scenario-level patterns for information extraction," in *6th*

conference on ANLP, pp. 282–289, Association for Computational Linguistics, 2000.

[5] K. Sudo, S. Sekine, and R. Grishman, "Automatic pattern acquisition for japanese information extraction," in *1st conference on HLT*, pp. 1–7, Association for Computational Linguistics, 2001.

[6] S. D'Mello, A. Olney, and N. Person, "Mining collaborative patterns in tutorial dialogues," *JEDM*, vol. 2, no. 1, pp. 1–37, 2010.

[7] R. Barzilay, M. Collins, J. Hirschberg, and S. Whittaker, "The rules behind roles : Identifying speaker role in radio broadcasts," in *NCAI*, pp. 679–684, Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999, 2000.

[8] D. Rao and F. Kurdahi, "On clustering for maximal regularity extraction," *IEEE TCAD*, vol. 12, no. 8, pp. 1198–1208, 1993.

[9] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining : current status and future directions," *Data Min. Knowl. Disc.*, vol. 15, no. 1, pp. 55–86, 2007.

[10] L. Holder, D. Cook, S. Djoko, *et al.*, "Substructure discovery in the subdue system," in *AAAI Workshop on Knowledge Discovery in Databases*, pp. 169–180, 1994.

[11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, *et al.*, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, 1990.

[12] W. R. Pearson *et al.*, "Rapid and sensitive sequence comparison with fastp and fasta.," *Methods in Enzymology*, vol. 183, p. 63, 1990.

[13] T. Lecroq, A. Pauchet, É. Chanoni, and G. Solano, "Pattern discovery in annotated dialogues using dynamic programming," *IJIDS*, vol. 6, no. 6, pp. 603–618, 2012.

[14] Z. Ales, A. Pauchet, A. Knippel, and E. Chanoni, "Extraction and clustering of two-dimensional dialogical patterns," *IJAIT*, en soumission.

[15] S. Guha, R. Rastogi, and K. Shim, "Rock : A robust clustering algorithm for categorical attributes* 1," *Information Systems*, vol. 25, no. 5, pp. 345–366, 2000.