# Alignement cognitif de symboles

Florence Dupin de Saint-Cyr Stéphane Loiseau

bannay@info.univ-angers.fr loiseau@info.univ-angers.fr

LERIA – Université d'Angers, 2, Boulevard Lavoisier - 49045 Angers cedex

**Résumé :**

L'idée de ce travail est de formaliser le problème suivant : à partir d'un ensemble de lignes contenant chacune un certain nombre de symboles, trouver une présentation la plus "factorisée" possible de ces lignes en permutant les symboles des lignes et les lignes entre elles. La factorisation consiste à faire disparaître un symbole déjà présent à la même colonne sur la ligne précédente en le remplaçant par un trait signifiant « cf. ci-dessus ». Cette idée présente trois avantages sur le plan cognitif : visualiser plus rapidement tous les symboles présents; visualiser rapidement les différences entre des lignes consécutives ; mettre en valeur les zones de ressemblance entre les lignes.

**Mots-clés** : Redondance, recherche documentaire.

**Abstract :**

The main idea of this work is to formalise the following problem: from a set of lines containing symbols, find the most "factorised" presentation of this document by permutating the symbols in a line and permutating the lines. The factorisation consists in replacing a symbol which is already present in the same column of the previous line by a dash which means "as above". Cognitively, this idea has three benefits: to quickly visualise the symbols which are present; to quickly see differences between consecutive lines; to underline similarity areas between lines.

**Keywords**: Redundancy, information retrieval.

## 1 Introduction

À partir d'un ensemble de lignes contenant chacune un certain nombre de symboles, nous proposons de fournir une présentation la plus "factorisée" possible de ces lignes, c'est-à-dire que nous désirons pouvoir retrouver toute l'information initiale en conservant le moins possible d'occurrence des différents symboles présents. L'idée est de ne pas réécrire un symbole lorsqu'il est déjà présent à la même colonne sur la ligne précédente, mais de le remplacer par un trait signifiant "cf. ci-dessus". Cette idée présente trois avantages sur le plan cognitif : elle permet de visualiser plus rapidement tous les symboles présents (puisque l'on élimine la gène créée par la redondance des symboles) ; elle permet de visualiser rapidement les différences entre des lignes consécutives ; elle met en valeur les zones de ressemblance entre les lignes.

Une des applications de ce problème est l'exploitation par des utilisateurs de données textuelles issues de systèmes informatiques. En effet, pour que la communication ait lieu entre l'homme et la machine, il faut que l'homme puisse comprendre les données qu'elle lui fournit. L'un des enjeux de l'interaction humain-machine est donc que la machine fournisse une présentation des données facilement accessible. De nombreux chercheurs ont proposé d'utiliser des représentations graphiques pour les données (voir, par exemple, [1,6,8]), mais, à notre connaissance, l'idée de minimiser le nombre de symboles utilisés n'a pas encore été exploitée de façon formelle et cognitive.

**Exemple** : Une application est la visualisation d'ouvrages suite à une recherche. Chaque ouvrage retenu est associé à une ligne de mots clefs. L'écran de la figure FIG.1 présente quatre ouvrages retenus avec leurs mots clefs.

```
O1:informatique      ,vision 3D,robot      .
O2:informatique      ,IA       ,validation ,basedeconnaissance.
O3:IA                ,vision 3D,informatique.
O4: basedeconnaissance,IA      ,informatique,génération.
```

FIG. 1 – Un example d'écran

Cet écran est présenté FIG.2 de sorte à factoriser sur une même colonne les mots clefs identiques. Cette factorisation a été obtenue par permutation de lignes (O3 apparaît en ligne 2), et de colonnes (l'ordre des mots caractérisant O3 a été inversé).

```
O1:informatique,vision 3D          ,robot.
O3:informatique,vision 3D          ,IA    .
O2:informatique,basedeconnaissance,IA  ,validation.
O4:informatique,basedeconnaissance,IA  ,génération.
```

FIG. 2 – écran aligné

L'écran précédent est présenté FIG3 en utilisant une notation sous forme de traits qui permet d'améliorer la vitesse de perception du résultat par le chercheur.

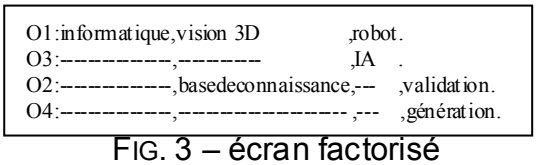


FIG. 3 – écran factorisé

En définitive, le problème consiste à trouver une permutation des symboles dans les lignes et des lignes entre-elles afin d'obtenir pour chaque ligne, un maximum de symboles alignés avec des symboles de la ligne précédente. Un symbole apparaissant à la même colonne sur deux lignes consécutives est effacé sur la deuxième ligne et représenté par un trait.

Le travail présenté dans cet article est un travail préliminaire : les aspects permettant de quantifier précisément son gain cognitif et d'étudier les algorithmes sont simplement évoqués ici. Notons que l'exemple d'application à la présentation de liste de mots clefs fait l'hypothèse forte que l'ordre des mots clefs n'a pas d'importance, et que les problèmes de synonymie ou d'homonymie ne sont pas pris en compte.

Dans la partie 2, nous formalisons le problème et montrons qu'il existe des solutions dans le cas où la surface d'affichage est infinie ou finie. Dans la partie 3, nous présentons rapidement les méthodes d'alignement avec lesquelles nous avons mené notre expérimentation qui est elle-même présentée en partie 4 où des premiers éléments d'analyse sont fournis.

## 2 Formalisation

Soit Symb un ensemble de symboles. On appelle *ligne* un ensemble de p *symboles* distincts $\{ s_1, .. s_i .. s_p \}$ où $s_i \in$Symb, on appelle *configuration* un ensemble de n *lignes* $\{ L_1, .. L_k .. L_n \}$. On cherche à trouver une méthode permettant d'afficher une *configuration* $C_0$ sur une surface S fixée de façon à écrire le **moins de symboles** possibles. Le *symbole* placé à la position i sur la ligne k est repéré par $s_{i,k}$ . Soit *pmax* le nombre maximum de symboles dans une ligne.

**Définition** : $\forall$ i $\in$[1..pmax], $\forall$ k $\in$[2..n], le symbole $s_{i,k}$ est dit *redondant* si le symbole $s_{i,k-1}$ est identique : $s_{i,k} = s_{i,k-1}$

Notre hypothèse de travail est que l'ordre des lignes et des symboles de ces lignes est indifférent. Ayant une configuration $C_0$, l'approche a pour but de trouver une configuration *équivalente* $C_j$ qui par permutation des lignes de $C_0$ et des symboles de ces lignes permette d'avoir le plus de symboles redondants. Dans l'exemple de l'introduction l'affichage de la deuxième configuration fait gagner sept symboles redondants, il est plus facile à lire.

Dans toute cette partie, pour simplifier, nous supposons que tous les symboles sont de longueur 1 et que la surface d'affichage est illimitée. Soit une configuration C composée de n lignes $L_k$ de symboles $s_{i,k}$. La surface d'affichage étant infinie, le nombre de *lignes* et le nombre de *symboles* affichables ne sont pas bornés. Montrons qu'il existe une *configuration* $C_{min}$ telle que le nombre de *symboles* affichés soit minimal. Construisons d'abord l'ensemble LCP des *configurations possibles* formé par la permutation des lignes de la *configuration initiale* $C_0$. A partir de cet ensemble, nous construisons l'ensemble LCF des configurations finales obtenu par permutation des symboles de chaque ligne d'une configuration possible de LCP (notons que LCF contient au plus n! × pmax! configurations, avec pmax le nombre maximal de symbole d'une ligne). Puis nous évaluons chacune des configurations en calculant le nombre de symboles qui y sont redondants. Les configurations ayant le maximum de symboles redondants sont des solutions. Le problème est donc décidable. Nous proposons en partie 3 un algorithme exact de type A* qui renvoie une meilleure configuration en minimisant au mieux le nombre de configurations à explorer et ébauchons quelques pistes pour des algorithmes approchés.

Une extension de ce problème est de considérer que la surface d'affichage est finie. On peut choisir de fixer la largeur et la hauteur ou simplement la surface. Si on sait

résoudre le problème dans le cas où la largeur et la hauteur sont fixées alors on sait aussi résoudre le problème dans le cas d'une surface fixée. L'ensemble LCF des configurations obtenues par permutations des lignes et des symboles de ces lignes calculé précédemment dans le cas d'une surface illimitée est aussi l'ensemble des configurations possibles pour ce problème. Ce qui change c'est la méthode d'évaluation des configurations possibles : on cherche une configuration qui maximise les symboles redondants lors d'un affichage page par page. Cette évaluation nécessite de segmenter la configuration en pages. Sur chacune de ces pages, la ligne de début devra être affichée entièrement. Ensuite les redondances sont calculées comme pour une surface infinie.

Nous avons vu dans cette partie, d'une part en quoi consiste le problème de l'affichage de symboles redondants, d'autre part, nous avons montré comment résoudre ce problème de manière exacte, et ce, que la surface d'affichage soit infinie ou non. La résolution se décompose en deux phases : la création de l'ensemble des configurations obtenues par permutations des lignes et de leurs symboles, et l'évaluation d'une configuration. Nous allons maintenant présenter l'intérêt de ce problème par le biais de son application à la recherche d'articles classés par thèmes, préalablement nous évoquons les algorithmes moins coûteux qui peuvent être utilisés.

## 3 Algorithmes

La solution consistant à générer toutes les configurations puis à les évaluer est de complexité rédhibitoire : O( n! × pmax! ).

On a d'abord utilisé un algorithme exact basé sur A* et la propagation de contraintes d'intervalles. L'idée est de partir d'une configuration vide, et de passer d'un nœud père au nœud fils en lui ajoutant une ligne de la configuration initiale. Un nœud contient non seulement un ensemble ordonné de lignes mais aussi un ensemble de contraintes sur les places possibles des symboles de la dernière ligne. Pour estimer les nœuds à développer on utilise deux fonctions. La fonction g compte le nombre de symboles redondants dans la configuration courante compte tenu des contraintes d'intervalles de ces symboles. La fonction h estime le nombre de symboles que l'on pourra effacer lorsque l'on aura intégré toutes les lignes de la configuration initiale. Lorsque l'on développe un nœud (en ajoutant une ligne à une configuration existante), les contraintes existantes sur les symboles de la ligne précédente génèrent des contraintes sur les symboles de la ligne à ajouter. Lorsque l'on a inséré toutes les lignes de la configuration initiale, la feuille obtenue contient les lignes ordonnées ainsi que les contraintes sur la place des symboles de la dernière ligne. Il suffit ensuite de re-propager les contraintes vers les lignes précédentes pour obtenir les solutions (en terme de permutations de symboles dans ces lignes) qui correspondent à cette feuille.

Cet algorithme bien que réduisant le nombre de nœuds à explorer est toujours d'une complexité trop forte (il tourne sur des exemples avec 5 lignes et 5 symboles, mais pas en temps raisonnable sur les exemples présentés plus loin). C'est pourquoi nous proposons d'utiliser des algorithmes gloutons, qui ne fournissent pas la meilleure solution mais une solution acceptable. Une idée est de se baser sur le nombre d'occurrences de chaque mot, puis de classer les lignes selon la somme des scores de leurs mots. Ensuite afficher comme première ligne, une ligne de score maximum puis la ligne suivante à afficher sera une des lignes restantes la mieux corrélée à la ligne courante. Ensuite, il reste à permuter les mots dans les lignes afin d'obtenir des redondances maximales. C'est ce type d'algorithme qui a été utilisé pour fournir les résultats de la partie suivante.

Une des particularités de ce problème est que l'on peut facilement déterminer la borne optimale du nombre de redondances à éliminer : il suffit de soustraire au nombre total de mots le nombre de mots distincts initial. On peut affiner cette borne en considérant des contraintes de place.

# 4 Application

Ce travail a d'abord été motivé par le développement d'une application qui permette de visualiser des ensembles de contraintes dont l'ajout permet de corriger des erreurs dans une base de connaissances [4,5]. Nous présentons ici une autre application de notre approche. Elle concerne la conférence RFIA qui a été organisée par notre laboratoire en 2002 [7]. Il s'agit d'une application qui permet de visualiser les différents articles soumis en fonction de leurs thèmes décrits sous forme de mots clefs par les auteurs. Un article correspond à une ligne et est caractérisé par au maximum trois mots clefs. L'utilisation de notre approche permet par exemple de se faire rapidement une idée des thèmes dominants de la conférence ainsi que des liens entre les mots clefs. Nous avons travaillé d'abord sur les 36 premiers articles puis sur l'ensemble des articles soumis à la conférence.

## 4.1 Tests

Le premier test a consisté à traiter la redondance de l'affichage d'une page contenant les numéros des trente six premiers articles associés à leur thèmes. L'écran qui présente les trente six articles dans l'ordre des données est fourni FIG4. L'écran qui présente les trente six premiers articles dans l'ordre optimisé par notre algorithme est fourni FIG5. D'un point de vue quantitatif, 30 mots différents sont présents dans la page, 79 mots apparaissent; si on appelle taux de redondance, le nombre moyen d'occurrences d'un mot, il y a un taux de redondance de 2,3 (79/30). L'ordre optimisé par notre algorithme permet d'enlever 34 redondances sur les 49 existantes, 15 redondances n'ont donc pas pu être éliminées. On constate facilement que la vitesse de perception des articles et de leurs mots clefs est plus rapide sur FIG5. Le même test a été mené sur les 36 articles suivants, les résultats sont proches : 26 mots différents sur les 82 mots, 44 redondances enlevées.

D'un point de vue plus qualitatif, une première synthèse des papiers apparaît lors de la lecture de FIG5. Ainsi "interaction homme-Machine", "Applications", "Vision dynamique", "Algorithmique", "indexation" apparaissent comme des thèmes forts, leur taux de redondance est d'au moins quatre. Seul le mot clé "représentation des connaissances", qui a un taux de redondance de 3 avec notre algorithme mais est répété 4 fois dans le texte, n'apparaît pas clairement. Notre approche permet donc en une seule visualisation et une lecture rapide, de fournir une première analyse du contenu de l'ensemble des papiers.

Notre méthode suppose que l'ordre n'a aucune importance ; ceci n'est pas vrai pour certains problèmes. Si l'ordre des lignes est impératif, seuls les changements de colonnes permettent d'enlever des redondances. Dans l'exemple RFIA, si on impose de voir apparaître les papiers dans l'ordre où ils sont édités, seuls trois redondances peuvent être enlevées. Cette notion d'ordre peut jouer aussi sur les symboles ; pour faciliter les recherches, on peut imposer un ordre alphabétique sur le premier mot de chaque colonne et/ou sur les symboles d'une même ligne. L'ordre alphabétique permet aussi d'éliminer des redondances mais les redondances qui apparaissent sont tributaires de cet ordre alphabétique, si les articles traités étaient très redondants sur un thème commençant par Z, ce classement aurait peu de chance de détecter cette redondance.

Un second test a été mené sur l'ensemble des 209 articles soumis à la conférence, la liste des mots contenait 40 mots distincts, le nombre total de mots était 472, le nombre de redondances éliminées est 317. La présentation calculée est dans http ://www.info. univ-angers.fr/pub/bannay/RECH/rfiamotscles.htm.

## 4.2 Analyse

L'intérêt de la présentation qui minimise le nombre de redondances apparaît clairement sur les tests effectués. Trois aspects essentiels doivent être soulignés.

- l'élimination des redondances permet de présenter des informations d'une manière plus simple et rapide à saisir par l'utilisateur.

- l'élimination des redondances met en valeur les redondances éliminées ; en effet pour un même mot, les redondances éliminées

apparaissent par des pavés de soulignés. La hauteur du pavé traduit directement le nombre de redondances. Par exemple, d'un seul coup d'œil, on découvre sur la figure FIG5 que certains mots clés sont redondants: Algorithmique, Applications (avec des redondances fortes :6) ; Interaction homme-machine (5) ; Indexation, Vision dynamique (4) ; Coopération personne-système, Représentation de connaissances, Traitement d'image, Vision3D (3) ; Classification, Contraintes, Formalisation des raisonnements, Langues naturelles (2).

- des liens entre redondances apparaissent; ainsi lorsque deux pavés de redondances sont voisins, les mots qu'ils représentent contiennent une corrélation. Sur l'exemple FIG5, on repère qu'il y a des liens forts (3 articles commun) entre Traitement d'image et Applications, entre Classification et Indexation, Vision dynamique et Vision 3D, Algorithmique et Contraintes, Formalisation des raisonnements et Coopération personne-systèmes (2 articles commun).

## 5 Perspectives du modèle

Nous avons formalisé la présentation de lignes de symboles de manière à optimiser l'interaction. Destiné au départ à améliorer l'interactivité de la présentation de documents, les perspectives de notre modèle de présentation sont multiples :

- aide à l'extirpation de thèmes et sous-thèmes : nous avons vu que notre modèle permettait de mettre en évidence des liens entre mots clés, ces liens peuvent permettre de déterminer qu'un mot-clé est une sous-catégorie d'un autre.

- outil de navigation : on peut envisager d'utiliser ce modèle de présentation en le raffinant. Par exemple, en permettant à l'utilisateur de sélectionner un mot clé à partir duquel il veut commencer une recherche et en affichant ensuite de manière optimisée les articles contenant ce mot, ainsi de suite jusqu'à découverte par l'utilisateur des articles qui l'intéressent. De plus, si l'on dispose d'une décomposition en thèmes et sous-thèmes, un aspect intéressant serait d'utiliser un effet de zoom sur un thème.

- interaction pour compréhension de textes : l'idée clé que l'on peut retirer de l'analyse de notre modèle est qu'il permet de repérer des mots ou concepts mais également des liens entre ceux-ci. Un tel outil peut aider à la structuration de la pensée.

Du côté algorithmique, il semble exister des liens avec la recherche sur des problèmes combinatoires comme l'alignement de séquences ADN [3] ou de voyageur de commerce [9]. Cependant il semble que notre problème se situe dans une classe de complexité supérieure puisque l'on doit examiner non pas toutes les possibilités dues à une permutation mais à deux permutations. Il reste des pistes à explorer du côté de la programmation dynamique et de la programmation à domaines finis sous contraintes. Une dernière piste algorithmique est l'utilisation des méthodes aveugles de compression de données (du type GZIP) [2].

## Références


[1] Barthet. Logiciels interactifs et ergonomie. Dunod, 1988.

[2] Benedetto, Caglioti, and Loreto, Language Trees and Zipping, Physical Review Letters 88 048702, January 2002.

[3] Bishop and Thompson. Maximum likelihood alignment of DNA sequences. Journal of Molecular Biology, 1986.

[4] Daligny. Validation de BC et interaction. Rapport DEA Orsay 2000.

[5] Dupin de Saint-Cyr, Loiseau. Révision à priori. RFIA2002, p.715-722.

[6] Furet. Une assistance cognitive pour les utilisateurs. Doctorat INA-PG, Paris 1995.

[7] Hao. éditeur. Actes du 12ème congrès RFIA, Angers 2002.

[8] Kuntz, Lehn, Briand. Visualisation d'un modèle graphique, RFIA00, vol.1, p.445-452.

[9] Lawler, Lenstra, Rinnooy Kan, Shmoys. The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, 1985.

| | | | |
|---|---|---|---|
| 1 | Traitementd'image | Vision3D | Visiondynamique |
| 2 | Diagnostic | | |
| 3 | Applications | Diagnostic | InteractionHomme-Machine |
| 4 | Modélisationsdesimages | Traitementd'image | Applications |
| 5 | Visiondynamique | | |
| 6 | Traitementd'image | Algorithmique | Analysedescènes |
| 7 | Tuteursintelligents | Coopérationpersonne-système | |
| 8 | Systèmesmulti-agents | InteractionHomme-Machine | |
| 9 | Indexation | Visiondynamique | Classification |
| 10 | Reconnaissancedelaparole | Reconnaissancedesformes | InteractionHomme-Machine |
| 11 | Algorithmique | | |
| 12 | Résolutiondeproblèmes | | |
| 13 | Applications | Traitementd'image | |
| 14 | Classification | Algorithmique | |
| 15 | Traitementd'image | Applications | Reconnaissancedesformes |
| 16 | Algorithmique | | |
| 17 | Reconnaissancedesformes | Contraintes | Algorithmique |
| 18 | Contraintes | Résolutiondeproblèmes | Algorithmique |
| 19 | Représentationdeconnaissances | Modèlesconceptuels | Raisonnementprobabiliste |
| 20 | Indexation | Languesnaturelles | Classification |
| 21 | Coopérationpersonne-système | Formalisationdesraisonnements | |
| 22 | Visiondynamique | | |
| 23 | Modélisationsdesimages | | |
| 24 | Vision3D | | |
| 25 | Evaluationdeperformances | Fouillesdedonnées | |
| 26 | Raisonnementspatial | Indexation | |
| 27 | Reconnaissancedel'écrit | | |
| 28 | Vision3D | Applications | |
| 29 | Représentationdeconnaissances | Formalisationdesraisonnements | Coopérationpersonne-système |
| 30 | Classification | Représentationdeconnaissances | Reconnaissancedel'écrit |
| 31 | Applications | Evaluationdeperformances | Réseauxneuronaux |
| 32 | Ingénieriedesconnaissances | Formalisationdesraisonnements | |
| 33 | Vision3D | | |
| 34 | Vision3D | Visiondynamique | Raisonnementprobabiliste |
| 35 | Languesnaturelles | InteractionHomme-Machine | Représentationdeconnaissances |
| 36 | Traitementd'image | Indexation | Analysededocuments |

FIG. 4 – Les 36 premiers articles de RFIA 2002 dans l'ordre des données

| | | | |
|---|---|---|---|
| 26 | Raisonnement spatial | Indexation | |
| 36 | Analyse de documents | ---------- | Traitement d'image |
| 9 | Classification | ---------- | Vision dynamique |
| 20 | ---------- | ---------- | Langues naturelles |
| 35 | Interaction Homme-Machine | Représentation de connaissances | ---------- |
| 10 | ---------- | Reconnaissance de la parole | Reconnaissance des formes |
| 8 | ---------- | Systèmes multi-agents | |
| 3 | ---------- | Applications | Diagnostic |
| 31 | Evaluation de performances | ---------- | Réseaux neuronaux |
| 4 | Traitement d'image | ---------- | Modélisations des images |
| 15 | ---------- | ---------- | Reconnaissance des formes |
| 13 | ---------- | ---------- | |
| 28 | Vision 3D | ---------- | |
| 24 | --------- | | |
| 33 | --------- | | |
| 5 | Vision dynamique | | |
| 22 | ---------- | | |
| 34 | ---------- | Vision 3D | Raisonnement probabiliste |
| 1 | ---------- | --------- | Traitement d'image |
| 6 | Algorithmique | Analyse de scènes | ---------- |
| 16 | ---------- | | |
| 11 | ---------- | | |
| 17 | ---------- | Contraintes | Reconnaissance des formes |
| 18 | ---------- | ---------- | Résolution de problèmes |
| 14 | ---------- | Classification | |
| 30 | Reconnaissance de l'écrit | ---------- | Représentation de connaissances |
| 19 | Modèles conceptuels | Raisonnement probabiliste | ---------- |
| 29 | Formalisation des raisonnements | Coopération personne-système | ---------- |
| 21 | ---------- | ---------- | |
| 7 | Tuteurs intelligents | ---------- | |
| 2 | Diagnostic | | |
| 25 | Evaluation de performances | Fouilles de données | |
| 32 | Formalisation des raisonnements | Ingénierie des connaissances | |
| 23 | Modélisations des images | | |
| 12 | Résolution de problèmes | | |
| 27 | Reconnaissance de l'écrit | | |

FIG. 5 – Les 36 premiers articles de RFIA 2002 dans un ordre optimisé