



FRANÇOIS ROUSSELOT

Mes recherches en IA et Langue Naturelle : de l'équipe Pitrat à une méthode de conception inventive

Volume 3, n° 1-2 (2022), p. 81-93.

http://roia.centre-mersenne.org/item?id=ROIA_2022__3_1-2_81_0

© Association pour la diffusion de la recherche francophone en intelligence artificielle et les auteurs, 2022, certains droits réservés.



Cet article est diffusé sous la licence

CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.

<http://creativecommons.org/licenses/by/4.0/>



*La Revue Ouverte d'Intelligence Artificielle est membre du
Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org

Mes recherches en IA et Langue Naturelle : de l'équipe Pitrat à une méthode de conception inventive

François Rousselot^a

^a Unistra Université de Strasbourg, avant sa retraite membre du Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie (ICube) – UMR 7357

E-mail : francois.rousselot@laposte.net.

RÉSUMÉ. — Le thème de recherche IA et Langue Naturelle n'était pas central dans l'équipe de Jacques Pitrat au début des années 70. Cependant, le besoin de comprendre et de représenter le sens d'un texte est rapidement apparu dans l'équipe, il s'est concrétisé notamment par plusieurs thèses en compréhension et en génération de langue naturelle et par la publication d'un ouvrage de J. Pitrat sur le sujet en 1985. L'auteur montre ici comment il a opté pour ce thème et l'a développé pour sa thèse. Ensuite, il a quitté quelque peu cette voie pour s'orienter vers des recherches en acquisition, modélisation et ingénierie des connaissances. Finalement, il s'est orienté vers l'étude d'une théorie de l'inventivité créée pour la fabrication d'objets manufacturés, afin d'en fournir une première modélisation dans un cadre proche de la résolution de problèmes en IA.

MOTS-CLÉS. — Sémantique des textes, traitement de corpus, extraction de connaissances, ingénierie des connaissances, analogie, invention, extraction de connaissances des brevets.

1. INTRODUCTION

En 1971, l'expression « Intelligence Artificielle » était totalement inconnue de la majorité des chercheurs, le développement de l'informatique en était encore à ses débuts. La puissance des machines était ridicule comparée à celle dont nous disposons actuellement, les programmes et les données devaient être saisis sur cartes perforées. Il fallait programmer en assembleur ou mieux en Fortran, un des rares langages évolués répandus, car Lisp n'a été disponible sur les machines du centre de calcul de Strasbourg que tardivement en même temps que Prolog. Il n'y avait pas de réseau utilisable par les chercheurs, il n'est apparu que dans les années 80-90 avec des fonctionnalités réduites (transferts par FTP). Le caractère fastidieux et les applications standards de l'informatique étaient peu attirants. Aussi dès que j'ai eu connaissance de l'existence d'un DEA à l'université de Paris VI au programme duquel figurait « programmation des jeux », je m'y suis inscrit. C'est ainsi que j'ai pénétré dans le domaine de l'IA et fait la connaissance de J. Pitrat, premier à enseigner l'IA en France. Ensuite, je me suis dirigé vers le thème « IA et Langue Naturelle » sur lequel j'ai travaillé pendant

toute ma vie de chercheur en gardant toujours des contacts avec mon équipe initiale, notamment en assistant aux colloques annuels.

Dans les paragraphes qui suivent, je décris différents aspects de mes recherches sur l'IA et les langues naturelles, en les situant par rapport aux objectifs et approches adoptés par J. Pitrat dans ses travaux sur ce même thème.

2. MA DÉCOUVERTE DE L'IA : COMPRENDRE LA LANGUE NATURELLE

En 1971, au DEA intitulé « Intelligence Artificielle et Reconnaissance des formes » mon sujet de mémoire, après discussion avec J. Pitrat, fut la réalisation d'un programme jouant aux échecs en utilisant la procédure alpha-bêta d'élagage d'un arbre (ou *alpha beta pruning*).

Plus tard, suite à la présentation du robot SHRDLU de Winograd [36] par J. Pitrat et après discussion avec lui, j'ai choisi, pour ma thèse, d'essayer d'adapter ce programme de compréhension de l'anglais au français. Le but était, d'une part, de voir si le français n'introduisait pas de difficultés particulières et, d'autre part, de mieux comprendre les mécanismes utiles à la compréhension de textes écrits.

Ma thèse de 3^e cycle fut donc une adaptation au français du robot SHRDLU de Winograd [24]. La limitation apportée par le choix d'un monde de blocs était une bonne trouvaille de Winograd, car elle supprimait de fait pas mal de problèmes linguistiques généraux : pas de compréhension de récit, référence pronominale simplifiée, calcul du sens facile, pas de grosses masses de connaissances à gérer, etc. Par suite, le passage de l'anglais au français fut relativement aisé. Curieusement, la plus grande difficulté est venue du fait de l'obligation de programmer en Fortran, puisque nous n'avions pas LISP au Centre de Calcul du CNRS à Cronenbourg, et encore moins les interpréteurs Programmar et Planner à base de LISP utilisés par Winograd, il a fallu les écrire en Fortran.

Il n'était pas question à l'époque de compréhension de la langue générale et tous les chercheurs se fixaient de fortes limitations sur le contexte et le vocabulaire, par exemple en se restreignant à des histoires courtes [31], au petit monde de Schank [32] et bien sûr, à des commentaires de jeux d'échecs [20].

Comme l'avait bien vu J. Pitrat, le problème posé par la compréhension de la langue générale est la nécessité d'accéder à une énorme masse de connaissances pour comprendre des phrases simples. Il le dit d'ailleurs clairement dans son ouvrage de 1985 [20] « Textes, ordinateurs et compréhension », tout texte peut aborder des domaines divers très différents du domaine principal, par le biais de métaphores, par exemple lorsqu'on parle de guerre dans un commentaire d'échecs.

Ces réflexions me conduisirent à renoncer à m'attaquer au problème de la compréhension de la langue générale et à essayer d'imaginer un contexte différent, limitatif qui permette d'affronter des difficultés raisonnables. C'est ainsi que j'ai mis un certain temps à déterminer le sujet de ma thèse d'état [25] qui a porté sur le langage de livres pour enfants.

Plus que le sujet d'application, la partie intéressante de ce travail fut l'utilisation des grammaires d'unification empruntées à M. Kay [13]. Ce langage déclaratif avec son interpréteur⁽¹⁾ fournissait un nouveau procédé de construction du sens par une unification spécifique, plus souple que l'unification de Prolog et adaptée à la construction de structures syntaxico-sémantiques. Ce formalisme, appelé *Descriptions Fonctionnelles* (DF dans la suite), permettait de définir aisément les connaissances déclaratives du système, qu'elles soient d'ordre syntaxique ou sémantique. Il permettait également d'effectuer de la génération de langue [14]. Ma thèse était une tentative de généraliser l'usage des DF à la compréhension de récits.

Cette unification d'arbres avec coréférences avait une complexité correcte [27]. Pour des travaux proches, le lecteur peut se reporter aux travaux des auteurs de LOGIN ([2] [3]), ils ont aussi étudié formellement la complexité d'une unification très semblable.

3. COMPRENDRE EN UTILISANT DES CONNAISSANCES DU DOMAINE

3.1. LINGUISTIQUE, LANGUE NATURELLE ET REPRÉSENTATION DES CONNAISSANCES

Ma nomination en 1984 dans une faculté de sciences humaines m'amena à discuter fréquemment avec des linguistes⁽²⁾ et à m'intéresser de près à la linguistique. J. Pitrat aborde cette question des rapports avec la linguistique dès l'introduction de son ouvrage [20] déjà cité : « les chercheurs en intelligence artificielle qui s'intéressent au traitement des langues ont besoin des linguistes. Notre but est de nous arranger pour que le linguiste puisse fournir le plus commodément possible aux programmes tout ce qu'ils doivent savoir. Nous ne voulons pas nous substituer aux linguistes dont l'expérience et le sens de la langue sont précieux. »

Cependant, sur le terrain, j'eus à essayer – et pas seulement à Strasbourg – les critiques toujours courtoises et malheureusement fondées qu'ils formulaient sur mes travaux et aussi généralement sur les approches de traitement des langues en informatique. Cette période fut pour moi une période de réflexion au sujet du Traitement Automatique des Langues (TAL). La croyance qui prédominait était la compositionnalité du sens, c'est-à-dire l'idée que le sens d'une phrase pouvait se construire à partir du sens de ses parties, ce qui est quelque peu simplificateur.

En plus des problèmes liés à la complexité de la langue, la compréhension de la langue nécessitait de bien représenter les connaissances, de les acquérir et de les structurer. Pensant que s'intéresser aux connaissances générales était prématuré, contrairement à D. Lenat dont le projet Cyc [16] n'a pas eu le développement espéré, tout ajout de connaissances se faisant manuellement, je me suis intéressé aux connaissances propres à un domaine. On peut noter ici que J. Pitrat [20] avait une position différente de celle de D. Lenat, il pensait qu'un système qui visait la constitution de la grosse masse

⁽¹⁾L'analyseur et l'interpréteur du langage furent réalisés en Pascal sur une des deux machines IBM du CNRS de Cronenbourg.

⁽²⁾Notamment Georges Kleiber, médaille d'argent du CNRS, qui devint et est toujours mon ami.

des connaissances pragmatiques nécessaires à la compréhension devait être capable d'acquérir automatiquement de nouvelles connaissances

3.2. DEUX APPLICATIONS DANS DES DOMAINES SPÉCIALISÉS

J'ai travaillé d'abord avec une approche générale incluant des aspects linguistiques de deux domaines spécifiques : celui de l'enseignement de la géométrie et celui de la génétique.

3.2.1. Recherche en EIAO

Au milieu des années 80, j'ouvris une parenthèse « géométrie » en travaillant avec un groupe de l'IREM (Institut de Recherche dans l'Enseignement des Mathématiques de Strasbourg) dans le domaine de l'EIAO (Enseignement Intelligemment Assisté par Ordinateur) sur l'enseignement de cette matière [11, 12]. Ce bref travail a consisté à étudier de près les énoncés des problèmes de géométrie et les procédés mis en œuvre pour leur résolution, afin de proposer une aide adaptée à l'élève. Nous avons dû étudier très précisément le sens des mots de ces énoncés, afin de comprendre les énoncés pour en établir une représentation formelle.

3.2.2. Recherche IA appliquée au génie génétique

Les dotations des équipes de recherche dans une Faculté de Sciences Humaines étant très faibles, je me mis donc à travailler avec des collègues de l'ENSAIS (Ecole Nationale Supérieure des Arts et Industries de Strasbourg), aujourd'hui INSA (Institut National des Sciences Appliquées). L'équipe de recherches ERIC (Equipe de Recherche en Ingénierie des Connaissances) de Strasbourg fut créée. Le premier projet fut un travail sur la génétique en collaboration avec un biologiste. C'était le début des groupes pluridisciplinaires bio-informatique, les recherches de ces groupes étaient massivement tournées vers la reconnaissance de séquences dans les chaînes d'ADN.

Avec un généticien spécialiste du clonage⁽³⁾, nous avons étudié la possibilité d'apporter une aide informatique au clonage. L'adaptation au vocabulaire du clonage ayant été difficile, nous avons réalisé un glossaire ou plutôt, sans le savoir, une ontologie des termes employés en fonction de leurs places dans les différentes actions possibles du biologiste. Après une étude de l'existant, nous avons créé un modèle du clonage qui a abouti finalement à un logiciel, une extension de Prolog qui possédait des fonctions spécifiques au clonage : gestion des séquences d'ADN et simulation de l'action d'un enzyme [17, 19].

Ces travaux furent l'occasion pour moi de rattraper mon retard dans les cultures lispennes et prologiennes et d'expérimenter la programmation de *bootstraps*. J'ai en effet reprogrammé en CommonLisp, puis en PROLOG, pour mon plaisir, l'interpréteur du métalangage réflexif AIRELLE [1] qui était écrit en LeLisp.

⁽³⁾Le clonage est comparable à une paire de ciseaux qui coupe une séquence d'ADN en plusieurs endroits chaque fois qu'il rencontre une séquence de bases déterminée.

4. EXTRACTION ET STRUCTURATION DE CONNAISSANCES À PARTIR DE TEXTES : VERS LES ONTOLOGIES

Les deux recherches précédentes m'ont amené à réfléchir à la problématique de l'acquisition des connaissances. Considérant qu'écrire un programme d'acquisition automatique des connaissances comme souhaité par J. Pitrat était prématuré et pas à ma portée, je me suis orienté vers des approches différentes : l'extraction des connaissances à partir de textes et l'Ingénierie des Connaissances. Je me suis focalisé plus précisément sur deux axes : la structuration des connaissances (structuration facilitant leur recherche en mémoire dans des systèmes) et l'acquisition à partir de textes.

4.1. STRUCTURATION DES CONNAISSANCES

Il fallait structurer les connaissances recueillies pour pouvoir s'en servir efficacement. C'est ainsi que cette étape généra plusieurs travaux de notre équipe de l'ENSAIS portant sur l'utilisation de Logiques de Description (DL) [8, 10] et plusieurs thèses. Elle se concrétisa par la mise au point d'une Logique de Description opérationnelle, CICLOP [7] qui fut utilisée dans plusieurs applications.

4.2. EXTRACTION DES CONNAISSANCES

Nos premiers travaux en extraction des connaissances débutèrent en 92 [27]. Ils furent suivis d'une recherche basée sur la détection de segments répétés [9] qui aboutit à la création de la station d'acquisition des connaissances Likes [29]; cette station servit à plusieurs projets de notre équipe de Strasbourg.

Parallèlement, dans le cadre des groupes AFCET (Association Française pour la Cybernétique Économique et Technique), j'ai travaillé avec Pierre-Yves Raccah, en 1992, à la mise en route d'un groupe « Informatique et Langue Naturelle ». Son existence brève a été, d'une certaine manière, à l'origine de la création du groupe Terminologie et Intelligence Artificielle en 1993 (TIA) centré sur l'extraction de connaissances spécifiques aux domaines et spécialement sur le statut et la gestion des terminologies [33]⁽⁴⁾.

Le groupe TIA s'intéressa tout naturellement aussi au statut et à la gestion des ontologies. La composition du groupe était intéressante, car y figuraient un certain nombre de linguistes avec lesquels les discussions furent très bénéfiques. Ce groupe eut une forte activité et organisa des journées⁽⁵⁾ jusqu'en 2020.

⁽⁴⁾Pour la majorité des linguistes, une terminologie n'est pas normative, car elle ne peut représenter qu'un usage des termes dans un domaine.

⁽⁵⁾https://tia.limsi.fr/2019/Slides/Le_groupe_Terminologie_et_Intelligence_ArtificielleV4.pdf

5. INTELLIGENCE ARTIFICIELLE ET INVENTIVITÉ : LA MÉTHODE TRIZ

5.1. DÉCOUVERTE DE LA MÉTHODE INVENTIVE TRIZ

En assistant au séminaire de recherche de l'ENSAIS dans les années 97-98, je découvris une méthode qui « permettait d'inventer ». Celle-ci, présentée comme une théorie de l'invention, éveilla ma curiosité : c'était la TRIZ (acronyme russe de la Théorie de Résolution des Problèmes Inventifs) élaborée par G. Altshuller à partir de 1946 [4].

La capacité à inventer est un défi très important pour l'IA ; de plus, je savais que J. Pitrat s'était intéressé à cette problématique très tôt [19] et que dans ses réflexions sur les métaconnaissances [21], il avait imaginé des mécanismes qui permettaient de créer de nouveaux concepts. J'avais donc hâte de voir si cette « théorie » avait un rapport avec des préoccupations présentes dans la communauté IA.

J'ai compris qu'en réalité, la TRIZ est une méthode (et non une théorie) composée d'un ensemble de règles pragmatiques s'appuyant sur des connaissances obtenues par une analyse manuelle approfondie d'une très grande quantité de brevets (des dizaines de milliers). Les équipes qui ont travaillé pour G. Altshuller lui ont permis de réaliser une synthèse permettant une mise en œuvre efficace de la méthode.

Un groupe d'informaticiens entreprit d'analyser cette méthode avec moi. Nous avons la preuve de son efficacité grâce à des comptes rendus de projets réalisés avec elle pour des applications industrielles réelles⁽⁶⁾. C'est également cet aspect qui aiguillonna notre curiosité avec l'espoir de participer à la conception d'un système qui ait des résultats pratiques dans le monde réel.

La TRIZ est utilisée en R&D pour débloquer des situations où le développement et l'amélioration d'un produit semblent marquer le pas alors que toutes les voies semblent avoir été explorées. Ses concepts ont été mis au point pragmatiquement par des ingénieurs travaillant sur des problèmes industriels et techniques dans les années 50. Ils sont donc très éloignés à la fois de certains types de problèmes plus récents et aussi de notre culture informatique.

Le principe général de la TRIZ est celui de la résolution d'un problème de contradiction. Le terme « contradiction » en TRIZ ne désigne pas une contradiction logique, il se situe plutôt dans le domaine du qualitatif : on veut que la valeur d'un certain paramètre augmente, mais alors la valeur d'un autre paramètre qu'on voudrait élevée diminue. On a deux objectifs contradictoires. A notre sens, le terme de « dilemme » qu'on va utiliser dans la suite, prête moins à confusion. Le terme de « résolution » du dilemme signifie qu'il va falloir proposer une solution qui le supprime ou qui va l'atténuer fortement, en minimisant les effets négatifs qu'il peut induire.

Après avoir présenté les concepts de base utilisés dans TRIZ, nous allons illustrer ce principe sur un exemple qui ne représente évidemment en rien la complexité des situations réelles d'application.

⁽⁶⁾Notamment chez Alstom, Arcelor Mittal, Chanel, Airbus.

5.2. TRIZ : LES CONCEPTS DE BASE

En préalable à l'exemple, nous devons présenter certains outils et connaissances utilisés par TRIZ, résultant de la synthèse des brevets évoquée plus haut : les paramètres, la matrice et les principes.

Les paramètres

L'analyse des brevets effectuée par les équipes de Altshuller a permis d'établir une liste de 39 paramètres abstraits, c'est à dire non liés à un domaine technique particulier, par exemple : longueur, masse, forme, flexibilité. Cette liste permet, après une phase d'abstraction, d'établir un modèle d'une situation sous la forme de deux paramètres abstraits qui engendrent un dilemme.

La matrice

Cette analyse a également permis d'établir une matrice 39x39 basée sur ces paramètres, qui consigne dans chaque élément de la matrice, correspondant à un dilemme, les principes à utiliser fonctionnant généralement pour résoudre ce dilemme.

Les principes

Il existe 40 principes⁽⁷⁾, par exemple *Segmentation*, Division d'un objet en parties indépendantes, *Extraction*, Séparation de l'objet d'une partie (ou propriété) « perturbatrice » ou, au contraire, extraction d'une partie (ou propriété) nécessaire.

5.3. TRIZ : LA MÉTHODE SUR UN EXEMPLE

Exemple : Un menuisier construit des tables avec un plateau en bois massif très épais. Il veut construire une table très longue sans rajouter de pieds. Il se trouve devant le dilemme suivant : si je désire avoir un plateau de grande taille, de même aspect, je dois le faire de même épaisseur et massif, mais je ne peux pas, car les pieds ne supporteront plus le plateau.

Le dilemme mis au jour ici concerne les deux paramètres abstraits : longueur et masse, longueur qui augmente et masse qui ne peut plus augmenter. La matrice contient, à l'intersection de la ligne et de la colonne correspondant aux deux paramètres, les principes à utiliser pour « résoudre » ce dilemme. Dans notre exemple, on utilisera le principe d'extraction et on enlèvera suffisamment de bois en dessous du plateau pour l'alléger sans affecter sa solidité et sans affecter son esthétique. Il y a d'autres solutions possibles, utiliser du placage ou renforcer par une pièce métallique cachée, etc.

D'autres sources de connaissances plus spécifiques sont accessibles, par exemple des brevets qui résolvent ce type de dilemmes dans toutes sortes de domaines qui ne sont pas forcément la menuiserie, l'idée étant qu'on pourra peut-être y trouver une solution nouvelle en adaptant une solution d'un autre domaine.

⁽⁷⁾Liste établie manuellement et consacrée par son efficacité pragmatique.

5.4. TRIZ : LE FONCTIONNEMENT

La méthode TRIZ fonctionne en quatre phases.

La phase initiale consiste à modéliser le problème. Cette phase est capitale, elle permet, d'une part, de déterminer les dilemmes, c.-à-d. les « contradictions », et, d'autre part, de construire un modèle abstrait sous la forme d'une « contradiction » ou d'un « réseau de contradictions ». En effet, les dilemmes ne peuvent généralement pas être exprimés par des contraintes numériques, mais plusieurs peuvent apparaître reliés alors par des relations de dépendances. Cette phase qui exige de s'abstraire du domaine permet notamment de mieux comprendre quel est le verrou qui bloque l'évolution d'un produit (l'amélioration souhaitée)⁽⁸⁾.

La deuxième phase consiste à sélectionner la contradiction qu'il faut résoudre, celle qui apportera un maximum de bénéfice⁽⁹⁾.

La troisième phase consiste à chercher à résoudre la contradiction choisie. On utilise la matrice comme un pointeur sur les principes, et les principes permettent de sélectionner des brevets intéressants dans d'autres domaines, éventuellement utiles pour être adaptés et résoudre le dilemme dans le domaine initial.

Ces brevets vont fournir des exemples où le dilemme a été réglé, ils vont générer des solutions abstraites (appelées « concepts de solution » en TRIZ), car il faut encore voir si la solution trouvée dans un autre domaine et formulée indépendamment de celui-ci est adaptable.

La quatrième phase est du ressort de l'ingénierie : il s'agit d'évaluer l'adaptabilité du concept de solution au domaine initial. Cela consiste généralement à établir un modèle mathématique de la solution, puis éventuellement à créer un prototype afin d'évaluer au mieux le coût de l'adaptation et le bénéfice apporté au système par celle-ci.

Je me suis alors demandé quelle assistance l'informatique pouvait apporter à la mise à jour des composants (principes, paramètres et matrice) et au déroulement des phases.

5.5. MODÉLISER ET INSTRUMENTER TRIZ

Pour mieux comprendre et clarifier les notions utilisées dans TRIZ, nous avons créé un modèle KADS (*Knowledge Acquisition and Documentation Structuring*) [30], puis élaboré une ontologie de ses concepts [26, 37] qui a permis finalement de générer un modèle de son fonctionnement.

Notre relecture de la théorie avec un chercheur, praticien et enseignant de celle-ci, a abouti à la redéfinir [6] en une méthode formalisée appelée « Méthode de Conception

⁽⁸⁾ Et de contourner les éventuelles rivalités entre experts de spécialités différentes dans les systèmes complexes.

⁽⁹⁾ Là intervient le contexte de l'application, on peut opter pour un bénéfice écologique, un moindre coût, une innovation maximale indépendamment du coût, etc.

Inventive » (MCI) qui met en œuvre notre modélisation. La méthode considérée du point de vue de l'IA peut se résumer ainsi :

- Création du modèle abstrait de la situation⁽¹⁰⁾,
- Recherche d'une analogie avec d'autres domaines,
- Évaluation de l'intérêt de cette analogie.

Le modèle développé a permis de concevoir des outils informatiques adaptés et utiles : d'une part, le logiciel STEPS d'aide à la construction du modèle de problème initial et à l'application de la méthode⁽¹¹⁾, d'autre part, un outil d'accès aux sources de connaissances.

Nous ne les détaillons pas ici, mais plusieurs autres « outils » ou bases de connaissances TRIZ sont à la disposition du praticien de la méthode, outils qu'il doit successivement utiliser pour résoudre un dilemme. La plupart de ces connaissances correspondent dans nos formalismes IA à des métarègles. Par exemple, chaque élément de la matrice pointe sur un certain nombre de procédures à essayer dans le cas identifié par les deux entrées de la matrice. La matrice représente donc plus de 1 500⁽¹²⁾ métarègles.

Une thèse résumée dans [5] a d'ailleurs été effectuée sur l'un des modèles de TRIZ (modèle substance-champ faisant appel à des lois physiques), on y utilise des règles de type SNARK [15]. J'avais pour l'occasion programmé un interpréteur dans une architecture hybride à base de règles et de logique de description pour gérer une base de connaissances.

Cependant, TRIZ semble être arrivée à un stade de développement où toute évolution est difficile : il est impossible pour l'instant de modifier les 3 entités essentielles, liste des paramètres, matrice, liste des principes. Ces entités sont le fruit d'études menées par les équipes d'Altshuller, dont il ne reste malheureusement aucune trace écrite.

En attendant, on assiste déjà à des tentatives d'évolution, par exemple avec l'adjonction de sources de connaissances du domaine animal avec le biomimétisme [35]. Mais la TRIZ devrait aussi pouvoir élargir nettement son champ d'action au-delà des problèmes techniques, avec la résolution de problèmes non physiques (problèmes de production d'artefacts numérisés) ou non concrets (problèmes informatiques, problèmes humains ou autres). Dans le domaine de l'informatique (logiciel ou matériel), des tentatives d'adaptation effectuées par des praticiens de TRIZ [23] existent, mais elles montrent bien que les 39 paramètres abstraits ne sont pas adaptés à traiter des problèmes qui se posent en informatique et qu'une évolution est nécessaire.

⁽¹⁰⁾L'analyse de la situation initiale permet souvent de mettre en évidence le verrou technologique qui bloque l'évolution d'un objet.

⁽¹¹⁾La société TimeToInnovate, créée par un de nos techniciens, a conçu puis commercialisé le logiciel STEPS. Aujourd'hui, la société EXELOP, qui a racheté TimeToInnovate, en commercialise une version améliorée sous le nom de PICC.

⁽¹²⁾ $39^2=1\ 521$.

5.6. EXTRACTION DE CONNAISSANCES DES BREVETS

Dans un contexte R&D, l'étude des brevets intervient en amont, lors de la phase d'analyse et en aval pour la résolution, il est donc important de soulager la tâche des ingénieurs qui opèrent manuellement. La phase d'étude des brevets, qui précède toute étude inventive, nécessite souvent l'analyse et la synthèse de deux ou trois cent brevets du domaine. Elle permet de mettre en évidence les paramètres importants du domaine potentiellement porteurs de contradictions ainsi que des contradictions.

La source de connaissances la plus naturellement évolutive utilisée par la TRIZ étant les bases de données de brevets, nous nous sommes intéressés à l'extraction des connaissances à partir de ceux-ci, particulièrement la mise en valeur des « contradictions » quand ceux-ci en résolvent. Nous avons élaboré un outil d'extraction des connaissances des brevets [34]. Ici l'extraction de connaissances est effectuée dans une optique TRIZ. Contrairement à ce qu'on pourrait penser, le langage utilisé dans les brevets n'est pas normalisé et recèle de nombreux pièges pour des logiciels d'extraction : notamment, on y trouve rarement des contradictions complètement exprimées.

Nous avons commencé ces travaux sur les brevets dans les années 2015, en utilisant une approche classique par schémas morphosyntaxiques, approche développée qui a fait ses preuves et a montré ses limites. D'autres techniques restent à explorer : par exemple des méthodes issues des recherches en linguistique de corpus, la modélisation par champs aléatoires conditionnels (CRF, pour *Conditional Random Fields*) ou d'autres méthodes statistiques.

Il y a encore beaucoup de choses à faire tant du point de vue de la collecte des solutions qui existent dans d'autres sources, notamment des textes moins normés que les brevets. Pour ce faire, les techniques actuelles du *deep learning* devraient pouvoir être adaptées et inclure de la sémantique, afin de trouver des informations pertinentes pour la conception inventive, par exemple pour y trouver des solutions biomimétiques.

5.7. PERSPECTIVES TRIZ ET IA

Comme on l'a vu plus haut, la méthode manipule des notions familières en IA : analogie, abstraction, instanciation (passage d'un niveau abstrait au niveau domaine), inventivité. L'utilisation de la matrice des contradictions synthétise des milliers de métarègles. Mais le fait que la liste des paramètres et les principes aient été établis manuellement dans les années 50 amène à se poser des questions sur leur statut et leur évolutivité. D'une part, rien ne prouve que le modèle ainsi établi soit encore optimal pour le domaine de la production industrielle, d'autre part, on pourrait chercher un niveau d'abstraction supérieur qui permette de dépasser cette limite et de modéliser d'autres domaines.

Revenons vers la quatrième phase de la méthode qui consiste à évaluer et adapter une solution existante. Cette phase est actuellement effectuée manuellement, il est donc apparemment impossible d'appliquer le système sur lui-même et de concevoir un amorçage comme dans [21] et aussi évidemment dans CAIA [22]. Cette impossibilité n'est peut-être qu'apparente, l'amorçage pourra peut-être se faire un jour grâce à des

connaissances que nous maîtriserons mieux, par exemple informatiques. Il faudra auparavant effectuer une reprise complète de la méthode dans une perspective IA afin d'en faire un système capable de s'utiliser sur lui-même et de résoudre ses contradictions pour optimiser son fonctionnement. Il est clair qu'elle devrait pouvoir fonctionner alors sur d'autres domaines avec des principes nouveaux, donc des métarègles nouvelles.

Une TRIZ renouvelée, largement fondée sur des approches TAL (Traitement Automatique des Langues) pour l'acquisition et la mise à jour des connaissances et des approches IA pour la représentation et le traitement des connaissances, semble possible. Nous retrouvons ici deux thèmes dont J. Pitrat a souligné l'importance à plusieurs reprises et pour lesquels il a fait des propositions.

6. CONCLUSION

La popularité et l'utilisation de l'Intelligence Artificielle semblent avoir explosé durant cette dernière décennie. Les méthodes et les moyens sont en pleine évolution. Les progrès effectués ouvrent de nouvelles perspectives fournies par des puissances de machines extrêmement augmentées permettant l'utilisation de techniques jusqu'ici inutilisables car trop chronophages.

Mais ceci ne devrait pas nous aveugler, les progrès actuels sont plus techniques que fondamentaux, ils mettent l'accent sur des techniques de l'IA capables d'acquérir et de traiter des masses extraordinairement importantes de données. Ces résultats ne doivent pas nous faire oublier qu'une recherche fondamentale est nécessaire, axée sur la création de systèmes capables d'évoluer par eux-mêmes, d'apprendre non seulement en ingurgitant des masses de données, mais en observant leur fonctionnement et en générant des règles et métarègles adaptées, systèmes dont CAIA est un précurseur.

BIBLIOGRAPHIE

- [1] A. ADAM-NICOLLE, « Le métalangage Airelle », *Revue d'intelligence artificielle* **4** (1990), n° 1.
- [2] H. AÏT-KACI, R. BOYER, P. LINCOLN & R. NASR, « Efficient Implementation of Lattice Operations », *ACM Trans. Program. Lang. Syst.* **11** (1989), n° 1, p. 115-146.
- [3] H. AÏT-KACI & R. NASR, « Login: a logic programming language with built-in inheritance », *The Journal of Logic Programming* **3** (1986), n° 3, p. 185-215.
- [4] G. ALTSHULLER & A. SEREDINSKI, *40 principes d'innovation TRIZ pour toutes applications*, Ed. Seredinski, 2004.
- [5] A. BULTEY, C. ZANNI-MERK, F. ROUSSELOT & F. DE BEUVRON, « A Hybrid System Combining Description Logics and Rules for Inventive Design », in *Proceedings of the 13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems: Part I*, Springer-Verlag, Berlin, Heidelberg, 2009, p. 87-94.
- [6] D. CAVALLUCCI, « Use of formal ontologies as a foundation for inventive design studies », in *Proceedings of Computers in Industry*, 2011.
- [7] F. DE BEUVRON, F. ROUSSELOT, M. GRATHWOHL, D. RUDLOFF & M. SCHLICK, « CICLOP », in *Proceedings of Description Logics*, 1999.
- [8] F. DE BEUVRON, F. ROUSSELOT & D. RUDLOFF, « Interpretation of DL for Natural Language and for Databases », in *Proceedings of Description Logics*, 1997.

- [9] P. FRATH, R. QUESLATI & F. ROUSSELOT, « Identification de relations sémantiques par repérage et analyse de cooccurrences de signes linguistiques », in *Actes de Ingénierie des Connaissances. Évolutions récentes et nouveaux défis, IC 2000*, Eyrolles, 2000.
- [10] M. GRATHWOHL, F. DE BEUVRON & F. ROUSSELOT, « A New Application for Description Logics: Disaster Management », in *Proceedings of Description Logics*, 1999.
- [11] D. GUIN, F. ROUSSELOT & H. EL HASSANI, « Un système expert résolvant des problèmes de géométrie (4e). Recherche en vue de la réalisation d'un programme d'EAO », in *Colloque Intelligence Artificielle, Strasbourg*, Cahier du LAFORIA, n° 60, 1986.
- [12] D. GUIN, F. ROUSSELOT & F. TODIRASCU et al. Groupe Intelligence Artificielle de l'IREM de Strasbourg, « Recherche en vue de la réalisation d'un programme d'EAO d'aide à la démonstration en géométrie », in *Actes de COGNITIVA 87*, 1987.
- [13] M. KAY, *Unification Grammars*, Xerox publication, 1981.
- [14] J.-M. LANCEL, F. ROUSSELOT & N. SIMONIN, « Grammar Used for Parsing and Generation », in *Proceedings of Coling 1986, Volume 1: The 11th International Conference on Computational Linguistics*, 1986, p. 536-539.
- [15] J.-L. LAURIÈRE & M. VIALATTE, « SNARK: A Language to Represent Declarative Knowledge and an Inference Engine which Uses Heuristics », in *Proceedings of the IFIP 10th World Computer Congress, Dublin, Ireland*, 1986, p. 811-816.
- [16] D. B. LENAT, « CYC: A Large-Scale Investment in Knowledge Infrastructure », *Commun. ACM* **38** (1995), n° 11, p. 33-38.
- [17] B. MIGAULT, F. ROUSSELOT & L. CERVONI, « Prolog as a Host Language », in *Proceedings of ICLP 91*, 1991, p. 890.
- [18] B. MIGAULT, F. ROUSSELOT, B. KEITH & J.-M. JELTSCH, « Un système d'aide à l'expérimentation en génie génétique », in *Actes du 9^e Séminaire de Programmation Logique SPLT 1988*, 1988, p. 247.
- [19] J. PITRAT, « Programmes capables d'invention », in *Symposium sur l'informatique médicale et les intelligences artificielles, Toulouse*, 1967.
- [20] ———, *Textes, ordinateurs et compréhension*, Eyrolles, 1985.
- [21] ———, *Métaconnaissance, futur de l'Intelligence Artificielle*, Hermès, 1990.
- [22] ———, « A Step toward an Artificial Intelligence Scientist », Tech. report, Paris VI University, France, 2008, Available at: <http://pagesperso-orange.fr/jacques.pitrat/>.
- [23] K. C. REA, « Using TRIZ in Computer Science – Concurrency », in *Triz Journal Aug 1999*, 1999.
- [24] F. ROUSSELOT, « Un robot qui comprend et exécute des ordres donnés en français », Thèse de 3^e cycle, Université de Paris VI, 1975.
- [25] ———, « Réalisation d'un programme comprenant des textes en utilisant un formalisme unique », Thèse d'état, Université de Paris VI, 1984.
- [26] F. ROUSSELOT, A. BULTEY & C. ZANNI, « Formalisation des connaissances capitalisées en Conception Inventive », in *Actes des 19^{es} Journées Francophones d'Ingénierie des Connaissances IC 2008*, 2008.
- [27] F. ROUSSELOT & H. GROSCOT, « Un langage déclaratif uniforme et un analyseur syntaxico-sémantique », in *Actes de Cognitiva 85, Paris*, 1985.
- [28] F. ROUSSELOT & B. MIGAULT, « Élaboration de techniques d'analyse adaptées à la construction d'une base de connaissances », in *Proceedings of COLING 1992*, 1992, p. 483-489.
- [29] F. ROUSSELOT & N. MONTESSUIT, « LIKES : un environnement d'ingénierie linguistique et d'ingénierie des connaissances », in *Formaliser les langues avec l'ordinateur : de INTEX à Nooj* (D. Maurel & M. Silberstein, éd.), Presses Universitaires de Franche Comté, 2007, <https://books.openedition.org/pufc/26889>, p. 389-409.
- [30] F. ROUSSELOT & C. ZANNI, « La Conception Innovante : Synthèse de systèmes ou résolution de problèmes? », in *Actes d'IC 2006*, 2006, p. 121-130.
- [31] G. SABAH, « A conversational system which understands short stories », in *Proceedings of the 1978 AISB/GI Conference on Artificial Intelligence*, 1978.
- [32] R. SCHANK, « A conceptual dependency parser for natural language », in *Proceedings of the 1969 conference on Computational linguistics*, 1969.
- [33] M. SŁODZIAN, « Comment revisiter la doctrine terminologique aujourd'hui? », *La Banque des mots* (1995), n° 7, numéro spécial : Terminologie et Intelligence Artificielle.

- [34] A. SOUILI, D. CAVALLUCCI & F. ROUSSELOT, « Identifying and Reformulating Knowledge Items to Fit with the Inventive Design Method (IDM) Model for a Semantically-based Patent Mining », *Procedia Engineering* **131** (2015), p. 1130-1139.
- [35] J. F. VINCENT, « TRIZ as a Primary Tool for Biomimetics », in *Research and Practice on the Theory of Inventive Problem Solving (TRIZ)*, Springer, Cham, 2016.
- [36] T. WINOGRAD, « Procedures as a Representation for Data in a Computer Program for Understanding Natural Language », Doctoral thesis, Massachusetts Inst. of Tech. Cambridge, 1971.
- [37] C. ZANNI-MERK, F. DE BEUVRON, F. ROUSSELOT & W. YAN, « A formal ontology for a generalized inventive design methodology », *Appl. Ontology* **8** (2013), n° 4, p. 231-273.

ABSTRACT. — The AI and Natural Language research theme was not central to Jacques Pitrat's team in the early 1970s. However, the need to understand and to represent the meaning of a text quickly appeared in the team, it was concretized in particular by several theses in comprehension and generation of natural language and by the publication of a book by J. Pitrat on the subject in 1985. The author shows here how he chose this theme and developed it for his thesis. Then, he left this path somewhat to focus on research in knowledge acquisition, modeling and engineering. Finally, he turned to the study of a theory of inventiveness created for the manufacture of manufactured objects, in order to provide a first modeling in a framework close to the problem solving in AI.

KEYWORDS. — Text semantics, corpus processing, knowledge extraction, knowledge engineering, analogy, invention, knowledge extraction from patents.

Manuscrit reçu le 30 mars 2021, révisé le 14 novembre 2021, accepté le 15 décembre 2021.