Proposition d'un système de recherche d'information dans un environnement numérique distribué et hétérogène : application à l'industrie manufacturière

Lise Kim

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR
[Laboratoire d'ingénierie des Systèmes Physique et Numériques (LISPEN EA 7515) & Laboratoire de Conception de Produits et Innovation (LCPI EA3927) – Campus d'Aix-en-Provence]

THÈSE

présentée par : **Lise KIM**

soutenue le : **28 octobre 2021**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Spécialité : **Conception et Industrialisation**

Proposition d'un système de recherche d'information dans un environnement numérique distribué et hétérogène : application à l'industrie manufacturière

THÈSE dirigée par : **Philippe VERON**
Co-dirigée par : **Frédéric SEGONDS**
Co-encadrée par : **Esma YAHIA**

Jury

M. Frédéric NOEL, Professeur des Universités, G-SCOP, Université Grenoble Alpes	Président
M. Sebti FOUFOU, Professeur des Universités, LIB, Université de Bourgogne	Rapporteur
M. Hervé PANETTO, Professeur des Universités, CRAN, Université de Lorraine	Rapporteur
Mme Nadège TROUSSIER, Professeur des Universités, ICD, UTT	Examinatrice
M. Philippe VERON, Professeur des Universités, LISPEN, Arts et Métiers HESAM Université	Examinateur
Mme Esma YAHIA, Maître de Conférences, LISPEN, Arts et Métiers HESAM Université	Examinatrice
M. Frédéric SEGONDS, Maître de Conférences HDR, LCPI, Arts et Métiers HESAM Université	Examinateur
M. Benjamin DEGUILHEM, Docteur, CAPGEMINI	Examinateur

T H È S E

“

Le monde apparait de plus en plus comme un vaste ensemble de systèmes relationnels où de nouvelles cartographies sont nécessaires pour habiter son intérieur

”

Jean-Max Noyer et Maryse Carmes (2014)

Remerciements

En premier lieu, je tiens à remercier l'ensemble de l'encadrement de cette thèse. Merci donc à Philippe VERON, Esma YAHIA et Frédéric SEGONDS pour leurs soutiens et accompagnements durant ces trois années. Esma, tu as su te rendre présente et disponible au quotidien pour échanger sur les travaux de thèse et je t'en remercie chaleureusement. Merci également aux différents acteurs de la société Capgemini pour nos échanges enrichissants, j'espère avoir le plaisir de construire de nouveaux projets avec vous d'ici peu. Je tiens également à remercier l'ensemble des doctorants qui ont croisés mon chemin, avec une intention particulière à Anthony GEROMIN, Nathanael DOUGIER, Jérémy MONTLAHUC, Kenza AMZIL et Thomas AMORETTI sans qui ces années n'auraient pas eu la même saveur. Je remercie également mon compagnon de vie, Christophe, qui a su me soutenir et me remonter à bloc lorsque nécessaire. Merci également à mes anciens collègues qui ont indirectement contribués à cette thèse : à Marc DATCHARY qui m'a donné le goût du PLM et m'a enseigné la recherche permanent de la vision d'ensemble et le coup d'avance en gestion de données ; à Vincent RIFFARD pour m'avoir conforté dans ma légitimité à poursuivre des travaux dans le domaine du PLM ; à Mickaël AVELINE pour m'avoir transmis la bonne offre de thèse ; à Audrey SAIGNOL, Raphaël BECK, Khalid KAJJOUA, Abdessamad ESSAYDI et l'ensemble des développeurs qui nous et vous ont rejoins depuis mes débuts en tant qu'AMOE, votre bonne humeur m'a manquée mais le souvenir de nos chansons et de vos blagues m'ont toujours fait sourire dans les moments plus moroses. Enfin, merci à mon père pour son soutien infaillible.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Contexte

Nous tentons par la modélisation des phénomènes régissant notre monde de comprendre les comportements. La compréhension de ces comportements nous aide alors à prendre des décisions éclairées et efficientes. Ce principe n'échappe pas à la prise de décision dans l'industrie manufacturière où l'accès à la bonne information, au bon moment, par la bonne personne et quelle que soit l'étape du cycle de vie du produit est un défi connu depuis le début de la numérisation de ses produits et de ses usines. La généralisation de l'acquisition des données renforcée par les principes clés de l'industrie 4.0 rend l'accès et l'exploitation du patrimoine informationnel de l'entreprise essentiels en plus d'élargir les perspectives d'usage des données (J. Li et al. 2015).

Problématique

Cependant, malgré les efforts déployés pour gérer et normaliser les informations, l'industrie manufacturière est confrontée à certaines difficultés comme le nombre de données à gérer et son organisation en silos. Le silotage de l'activité de l'entreprise a un double impact. Premièrement, il génère une importante hétérogénéité dans la modélisation de l'information rendant les solutions de valorisation de l'information adressant plusieurs silos ad hoc et donc coûteuses et périssables dans un environnement dynamique et modulaire. Deuxièmement, il engendre des ruptures dans la continuité numérique de l'information limitant alors sa pleine exploitation.

Objectif

C'est pour contribuer à l'enjeu majeur de la valorisation de l'information dans ce contexte que la thèse vise à proposer une vision holistique de l'information disponible en entreprise tout en limitant les ruptures de la chaîne numérique de l'information.

Démarche de recherche et organisation du mémoire

La thèse a suivi le cadre conceptuel de la démarche de recherche présentée dans (Hevner et al. 2004) et illustrée par la Figure 0.1.

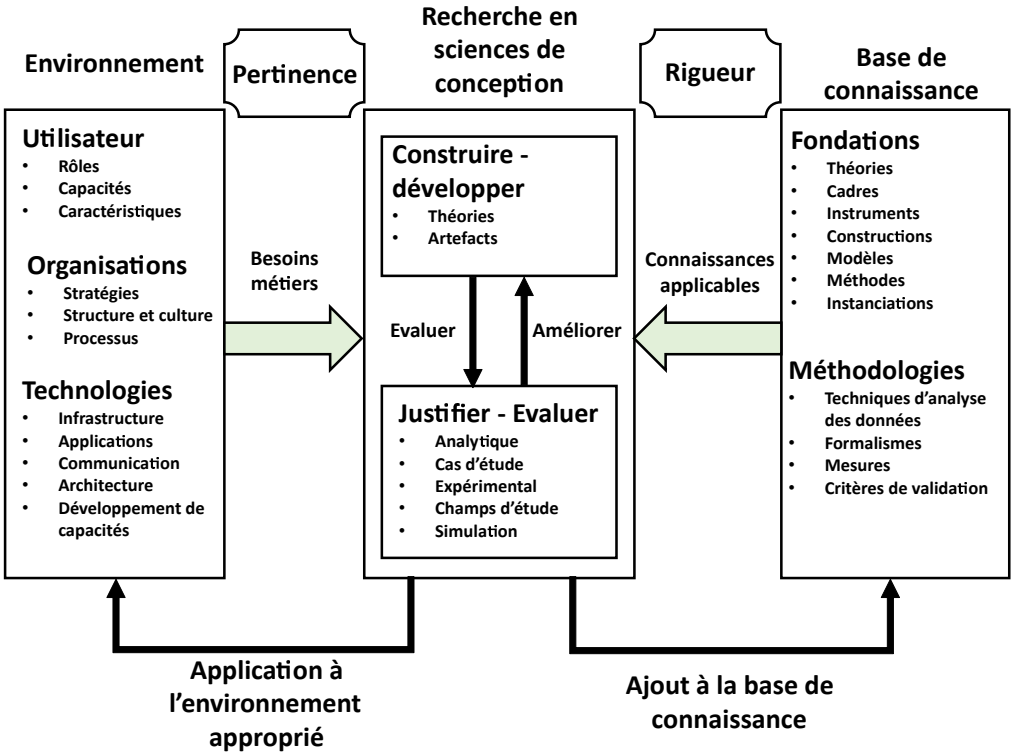


Fig. 0.1 : Cadre de recherche sur les systèmes d'information traduit en français (Hevner et al. 2004)

Selon le cadre de recherche sur les systèmes d'information de (Hevner et al. 2004), il est nécessaire de formuler un problème de recherche à partir de l'environnement dans lequel résident les phénomènes d'intérêts afin de garantir la pertinence de la recherche. C'est pourquoi l'objectif du premier chapitre "**Contexte et question de recherche**" est de présenter l'environnement de recherche de la thèse pour répondre à la question : "À quels enjeux contribue la thèse ?". On y présente en effet les différentes parties prenantes, certaines définitions utiles à la bonne compréhension du mémoire, les enjeux liés à l'exploitation de l'information dans l'industrie manufacturière puis ses freins. Les sections suivantes présentent la question de recherche et les exigences associées qui permettront d'évaluer la proposition générale de la thèse. Le chapitre conclut sur la présentation des contributions apportées par l'ensemble de l'étude.

Toujours selon le cadre de recherche que nous considérons, la base de connaissances fournit la matière première et la rigueur nécessaire dans la démarche de recherche. Ainsi, il est nécessaire de s'appuyer sur les fondements et méthodologies en relation au sujet de l'étude. C'est pourquoi l'objectif du second chapitre "**Etat de l'art**" est de répondre à la question : "Que nous apprend l'état de l'art sur les approches susceptibles de répondre à la question de recherche ?". Le chapitre est décomposé en trois parties. La première partie présente ce qu'est la Recherche d'Information et ses Systèmes, comment le do-

maine est actuellement présenté, implémenté et évalué. Cette partie inclut également une présentation du domaine d'Extraction d'Information et sa distinction avec celle de la Recherche d'Information. La seconde partie présente l'approche graphe comme modélisation de l'information en y incluant les notions essentielles sur la théorie des graphes, ce qu'elle permet dans la représentation et l'analyse des réseaux d'informations hétérogènes et dans la représentation des connaissances. La troisième partie présente les études sur la Recherche d'Information en entreprise en soulignant notamment celle utilisant la modélisation graphe.

Le pilier central de la démarche de recherche consiste à développer une proposition selon la problématique de recherche formulée grâce à l'environnement et selon les connaissances acquises constituant la base de connaissances. La construction de cette proposition rentre alors dans un cadre permettant de justifier, évaluer et réajuster chaque décision prise. Dans cette optique, l'objectif du troisième chapitre "**Cadre de construction de la proposition**" présente les éléments utilisés pour construire la réponse à la question de recherche. Ce chapitre présente dans un premier temps l'analyse fonctionnelle du système attendu puis le processus d'évaluation et d'enrichissement de la proposition utilisé dans les travaux. Le cas d'étude comprenant le jeu de données ainsi que les cas d'usages sélectionnés y sont également détaillés. L'objectif du quatrième chapitre "**Proposition i-Dataquest**" est quant à lui de présenter la première proposition pour répondre à la question de recherche, point de départ du pilier central de la démarche de recherche. On y présente donc l'architecture générale de la proposition puis le détail des théories et méthodes retenues pour réaliser chacune des fonctions du système. Enfin, la proposition est confrontée au cas d'étude permettant ainsi de dégager une liste d'enjeux clés à résoudre afin d'améliorer la proposition. L'objectif du cinquième chapitre "**Enrichissement de la proposition**" présente les travaux menés pour répondre à cette liste d'enjeux clés. Chaque nouvelle itération de proposition ainsi détaillée est ensuite vérifiée selon le cadre défini au troisième chapitre. Ces opérations sont représentées par les flèches internes au pilier central de la démarche de recherche présentée dans la Figure 0.1.

Enfin, l'objectif du sixième chapitre "**Validation et discussion**" est représenté par la flèche de retour sur l'environnement de la démarche de recherche. Cet objectif est d'établir dans quelle mesure la proposition répond à la question de recherche présentée dans le contexte. Pour cela, plusieurs exigences sont considérées et une validation empirique avec un second cas d'étude est réalisée. Une seconde partie du chapitre présente les implications théoriques et pratiques de l'étude ainsi que ses limites. Ce chapitre aboutit enfin sur une conclusion générale aux travaux de thèse et les perspectives envisageables dans un poursuite des travaux.

Ainsi, le mémoire se décompose en six chapitres dont les articulations entre chaque chapitre, les questions qui les animent ainsi que leurs livrables sont illustrés dans la Figure 0.2.

FIG. 0.2 : Organisation du mémoire

Chapitre 1

Contexte et question de recherche

1.1 Objectif du chapitre

Comme illustré dans la Figure 1.1, l'objectif de ce chapitre est de présenter le contexte de la thèse pour répondre à la question "A quels enjeux contribue-t-elle ?". Le chapitre est décomposé en trois parties : la première présentant les parties prenantes de la thèse et certaines définitions importantes pour sa compréhension, la seconde présente les enjeux, les freins et les approches d'exploitation de l'information dans l'industrie manufacturière, la troisième présente la question de recherche et les exigences associées.

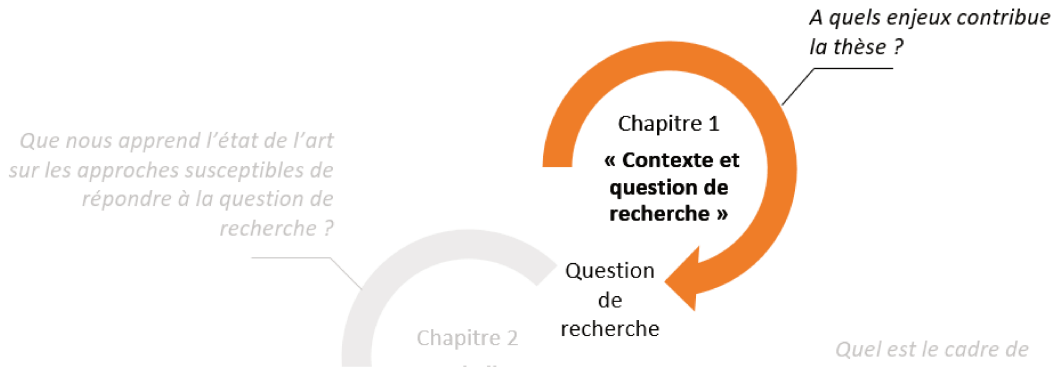


FIG. 1.1 : Organisation du mémoire : premier chapitre

1.2 Préambule

1.2.1 Les parties prenantes

La thèse présentée dans ce mémoire a été réalisée au sein des laboratoires LISPEN[1] et LCPI[2] de l'École Nationale Supérieure d'Arts et Métiers (ENSAM) ainsi que la société de conseils en numérique Capgemini[3] dans le cadre de la chaire "PLM du futur". La notion de PLM[4] *"est une approche métier intégrée pour la création, la gestion et la diffusion collaboratives des données d'ingénierie à travers l'entreprise étendue qui crée, fabrique et exploite des produits et des systèmes d'ingénierie"* (BOURAS et al. 2016). Les deux laboratoires font partie de l'institut Carnot ARTS visant à favoriser la recherche partenariale.

Arts et Métiers l'ENSAM est une école d'ingénieur orientée génie industriel et mécanique. Elle dispose de quinze laboratoires de recherche spécialisés.

LISPEN la recherche du laboratoire s'articule autour des systèmes dynamiques multi-physiques et virtuels pour l'Industrie du Futur. Le laboratoire intègre les quatre théma-

[1]LISPEN pour Laboratoire d'Ingénierie des Systèmes Physiques et Numériques - https ://lispen.ensam.eu/
[2]LCPI pour Laboratoire Conception de Produits et Innovation - http ://lcpi.ensam.eu/
[3]https ://www.capgemini.com
[4]PLM pour Product Lifecycle Management, la gestion du cycle de vie du produit

tiques principales suivantes : l'ingénierie système et les maquettes numériques, la simulation et le contrôle des systèmes, l'interaction Homme-Système et l'aide à la décision.

LCPI la recherche du laboratoire s'articule autour de l'optimisation du Processus de Conception et d'Innovation pour les produits industriels. Il fait appel principalement aux sciences pour l'ingénieur et aux sciences humaines et sociales.

Capgemini la société Capgemini est spécialisée dans le conseil, le service informatique et la transformation numérique dans de multiples secteurs comme l'industrie, le service financier ou encore la télécommunication. Créée en 1967 sous le nom de Sogeti, elle fait depuis partie des acteurs mondiaux du secteur et se place comme première Entreprise de Service en Numérique (ESN) de France en 2020[5]. Capgemini est engagée dans la recherche de nouvelles technologies pour ses clients avec son propre institut 'Capgemini Research Institute' mais également avec des partenariats comme celui de la chaire 'PLM du futur' avec les Arts et Métiers.

Chaire "PLM du futur" la chaire PLM du futur a été signée entre Capgemini et l'ENSAM afin de répondre aux enjeux de la transformation digitale industrielle s'appuyant notamment sur la convergence du monde physique et numérique du produit tout au long de son cycle de vie.

1.2.2 Définitions

Avant de poursuivre, il est important de clarifier les différents termes suivants :

Données, informations et connaissances : nous rejoignons l'explication de l'auteur dans (Serrano 2014) qui distingue les trois termes de la manière suivante : *"Une donnée est généralement définie comme un élément brut non traité et disponible hors de tout contexte. Une fois collectées et traitées, par le cerveau humain ou par une machine, ces données deviennent des informations. Une information est le résultat de la contextualisation d'un ensemble de données afin d'en saisir les liens et de leur donner un sens. L'information est statique et périssable, sa valeur diminue dans le temps (car dépendante de son contexte qui est amené à varier). À l'inverse, la connaissance est le résultat d'un processus dynamique visant à assimiler/comprendre les principes sous-jacents à l'ensemble des informations obtenues. Cette compréhension permet de prévoir l'évolution future d'une situation et d'entreprendre des actions en conséquence. Sous réserve de cette interprétation profonde, une plus grande quantité d'information mène à une meilleure connaissance d'un sujet donné."* Afin d'illustrer le propos, nous prenons l'exemple de la donnée, sous forme clé-valeur, suivante : "<code postal> : <13400>". La donnée n'est qu'un enchaînement de caractères pour l'ordinateur et ne porte donc pas de signification. Lorsqu'un cerveau humain l'interprète dans son contexte, il comprends alors qu'on traite d'une adresse

[5]Classée selon le chiffre d'affaires par le syndicat de l'industrie du numérique Syntec et le réseau de cabinet d'audit KPMG dans le classement des ESN ICT 3ème édition - novembre 2020 https ://syntec-numerique.fr/sites/default/files/Documents/Classement$_d es_E SN_I CT_2 020_v def_0 .pdf$

comportant un code postal, devenant ainsi une information. Enfin, combinées à d'autres informations, on peut engendrer une nouvelle connaissance comme "l'acheteur se trouve dans le département des Bouches-du-Rhône".

Données structurées et non-structurées : nous rejoignons la définition fournie par les auteurs de (Kassner et al. 2015) afin de distinguer ce qu'est une donnée structurée d'une donnée non-structurée : les données structurées sont celles organisées en table comme les bases de données relationnelles, les données non-structurées sont des ensembles non organisés comme des textes, 3Ds, images etc. Pour illustrer le propos, les données relatives aux ventes comme les références, les quantités et les prix sont des données structurées tandis que des fichiers de spécifications du produit ou des e-mails sont des données non structurées. Enfin, les données semi-structurées sont des hybridations de données structurées et non structurées comme dans les fichiers XMLs. Ces fichiers sont en effet des documents textuels principalement non structurés mais comportant des étiquetages (ou tabulations) organisant le texte.

Syntaxe VS sémantique : nous rejoignons l'explication de l'auteur dans (Fortineau 2013) sur la distinction entre la sémantique et la syntaxe dans le domaine de la programmation informatique. *"La syntaxe est [...] la forme du langage de programmation, quand la sémantique en est le fond. Michel Bréal, grand artisan de la sémantique, l'appelle dès son premier essai de sémantique 'la science des signifiés' (*Bréal *1987). Jusqu'alors, la syntaxe (ou l'art de la grammatique) était censée expliquer tout langage d'après une position logique. L'introduction de la sémantique permet de se pencher sur le sens des mots, et non plus uniquement sur leurs articulations réciproques."* Pour illustrer le propos, nous pouvons distinguer dans l'exemple de la donnée "<code postal> : <13400>" la syntaxe qui est la règle de structuration donnant pour chaque clé (ici code postal) une valeur (ici 13400). La sémantique est ici l'information apportée par "code postal" et "13400", à savoir un élément d'une adresse.

Relation explicite et implicite : nous utiliserons par la suite les termes de 'relation explicite' et 'relation implicite' que nous définissons ici. Une relation explicite est la retranscription informatique de l'information de lien entre deux données (ex. : lien de parenté entre éléments d'une arborescence numérique comme illustré par les liens bleus dans la Figure 1.2 ou lien d'assignation d'un élément à un autre dans une table). Une relation implicite est une information de lien entre deux éléments non retranscrite informatiquement (ex. : lien entre deux éléments se référant l'un à l'autre ou représentant la même entité du monde réel [6] comme illustré par le lien rouge dans la Figure 1.2).

[6] La notion d'entité du monde réel est empruntée au domaine de recherche de la mise en correspondance d'entités (entity resolution en anglais) (Talburt 2011)

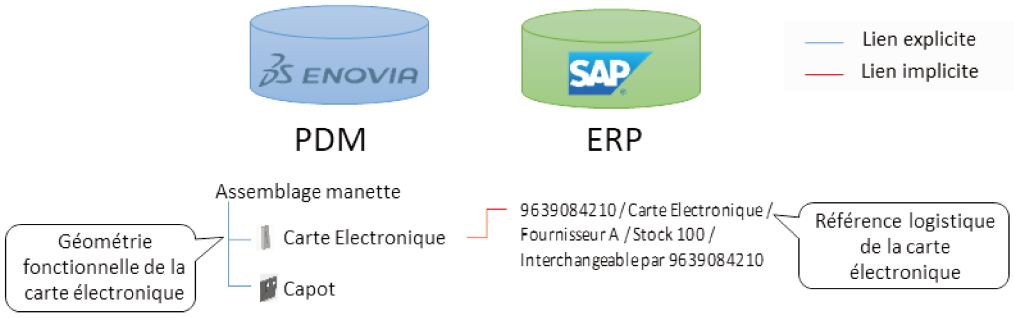


FIG. 1.2 : Illustration de liens explicites et implicites

Donnée, document et enregistrement : comme illustré dans la Figure 1.3, nous distinguerons deux types de données, les documents et les enregistrements. Les documents sont des fichiers créés au moyen d'un logiciel d'application comme un document word, pdf etc. On nommera 'métadonnées' pour parler de leurs propriétés (exemple : 'Nom', 'Modifie le' etc.). Les enregistrements sont les éléments listés dans les tables de base de données comprenant un certain nombre de champs descriptifs. Ces champs sont nommés 'attributs'.

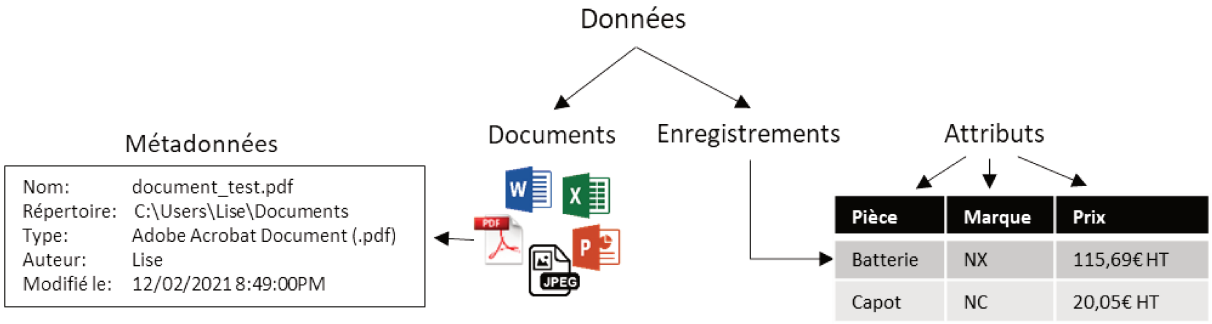


FIG. 1.3 : Illustration de la différence entre donnée, document et enregistrement

1.3 L'exploitation de l'information dans l'industrie manufacturière

1.3.1 Les enjeux

Les entreprises de l'industrie manufacturière ont un patrimoine informationnel important à leur disposition, et ce malgré une part de l'information non retranscrite numériquement. Ce patrimoine informationnel est un atout concurrentiel lorsqu'il est correctement exploité vis-à-vis des contraintes et de l'environnement de l'entreprise. Les enjeux sont détaillés ci-dessous et l'environnement de l'entreprise dans lequel ces enjeux s'articulent est illustré en Figure 1.4.

Valorisation du patrimoine informationnel

Comme l'illustre la partie ② de la Figure 1.4, le patrimoine informationnel de l'entreprise est composé de l'ensemble des données disponibles et acquises tout au long du cycle de vie du produit. Ces données décrivent le produit et son système de production, les clients et les fournisseurs, mais également des informations comme l'avancement du projet ou celles provenant des ressources humaines. C'est par l'acquisition de ces données ainsi que des démarches complémentaires de "gestion des connaissances" (en anglais "Knowledge Management") que la traçabilité et la capitalisation du retour d'expérience des collaborateurs de l'entreprise peuvent être réalisées. L'accès à cette capitalisation facilite alors la prise de décisions éclairées et de détecter de nouvelles opportunités à forte valeur ajoutée pouvant impliquer de multiples cas d'utilisation comme l'exploitation des exigences (Pinquié 2016). Selon l'enquête de Forrester[7] (Taylor 2019), les entreprises estiment qu'améliorer la gestion et l'utilisation des données disponibles permet une augmentation de 27% de la vitesse de prise de décision et une amélioration de 23% de la productivité. Selon la même enquête, plus de 50% des entreprises interrogées ont récemment investi dans des solutions de stockage et de gestion des données souhaitant améliorer leurs maturités sur le sujet.

Vision holistique des données

Un des enjeux dans l'exploitation de ce patrimoine informationnel est de considérer son ensemble malgré les silos de l'entreprise. Un exemple haut niveau d'organisation en silo est illustrée dans la partie ② de la Figure 1.4 composées pour l'exemple de différents systèmes d'information internes à l'entreprise comme un PDM[8] (pour la gestion du produit de conception), un ERP[9] (pour l'approvisionnement et la vente), une base de donnée relationnelle (pour la gestion des modifications du produit), un CRM[10] (pour la gestion de la relation client) et d'un autre PDM cette fois externe à l'entreprise (pour l'intégration des éléments de conception réalisés par le fournisseur). Cet ensemble de systèmes d'informations ainsi que leurs fonctionnalités sont des exemples. Cet enjeu n'est pas nouveau, on parlait déjà au début des années 90 d'Ingénierie Concourante (Prasad 1996) puis Intégrée (Andreasen et al. 2000) dont l'objectif était de prendre en compte tous les éléments du cycle de vie du produit afin d'éviter les effets négatifs du cloisonnement d'activités provenant notamment du processus séquentiel du développement de produits. Plus tard, les notions d'Ingénierie et de Conception Collaboratives (S.-Y. Lu et al. 2007) ont mis en lumière le besoin de prise de décision multi-expertises afin d'améliorer la qualité des produits. Plus récemment, bien qu'existante depuis plus d'une vingtaine d'années (MacLean et al. 1998), la notion de continuité numérique est utilisée par les acteurs du monde industriel. Cette notion évoque la chaîne numérique liant toutes les informations définissant un système, un produit et ses composants, et ce, tout au long de leurs cycles de vie (Meski et al. 2019). L'ensemble de ces notions, même si leurs noms diffèrent et étaient

[7] Entreprise d'étude de marché sur l'impact des technologies dans le monde des affaires - https://go.forrester.com/

[8] PDM pour 'Product Data Management'

[9] ERP pour 'Enterprise Ressource Planning'

[10] CRM pour Customer Relationship Management

à l'origine spécifiques à la conception produit, expriment le même besoin de faire face au silotage de l'information pour une prise de décision efficiente. La nécessaire interopérabilité des systèmes qui en découle est d'ailleurs encore un des enjeux clés de l'industrie 4.0 (Y. Lu 2017).

Diminution du temps de recherche de l'information

Parmi l'ensemble des étapes nécessaires à l'exploitation d'information, celle de la recherche à proprement parler (étape qui s'exécute entre le moment où le besoin d'information est émis et le moment où les résultats sont obtenus, illustrée en partie ③ de la Figure 1.4) n'a pas de valeur ajoutée en tant que telle pour l'entreprise. Le temps dédié à cette tâche devrait donc être le plus court possible ce qui n'est pas le cas. En effet, une enquête réalisée par l'IDC[11] estime la moyenne du temps d'activité d'un salarié dédié à rechercher de l'information à 9,5h par semaine (Feldman et al. 2005). Cette étude a été menée aux Etats-Unis sur un panel de 600 entreprises de catégories et tailles différentes.

Agilité à un environnement modulaire et étendu

L'un des challenges de l'industrie 4.0 est celui de s'adapter à une structure changeante en favorisant la modularité des systèmes qui la compose (Mittal et al. 2019). L'agilité d'organisation et d'infrastructure d'une entreprise est en effet un de ses facteurs clés de réussite (Harraf et al. 2015 ; Weill et al. 2002). Les 'méthodes agiles' pour le développement de logiciel sont d'ailleurs devenues une réalité terrain depuis plusieurs années (Larman 2004). Cette nécessité d'agilité est représentée par les pièces de puzzle dans la partie ② de la Figure 1.4. L'architecture des Systèmes d'Information de l'entreprise doit en effet s'adapter aux différentes solutions provenant de différent éditeurs qu'elle devra intégrer au fil des innovations et nouvelles propositions du marché. Elle devra également s'adapter aux choix de ses partenaires ayant leurs propres architectures dans un environnement d'entreprise étendue. Dans ce contexte, plus les solutions sont agnostiques et interopérables[12] entre-elles, plus le besoin d'interfaces spécifiques est limité. Elles seront donc plus maintenables et rentables au fil des années.

[11] IDC pour International Data Corporation est un groupe mondial d'étude de marché et de conseil en technologies de l'information - https ://www.idc.com/

[12] Interopérabilité : capacité de matériels, de logiciels ou de protocoles différents à fonctionner ensemble et à partager des informations - Dictionnaire Larousse, 2021

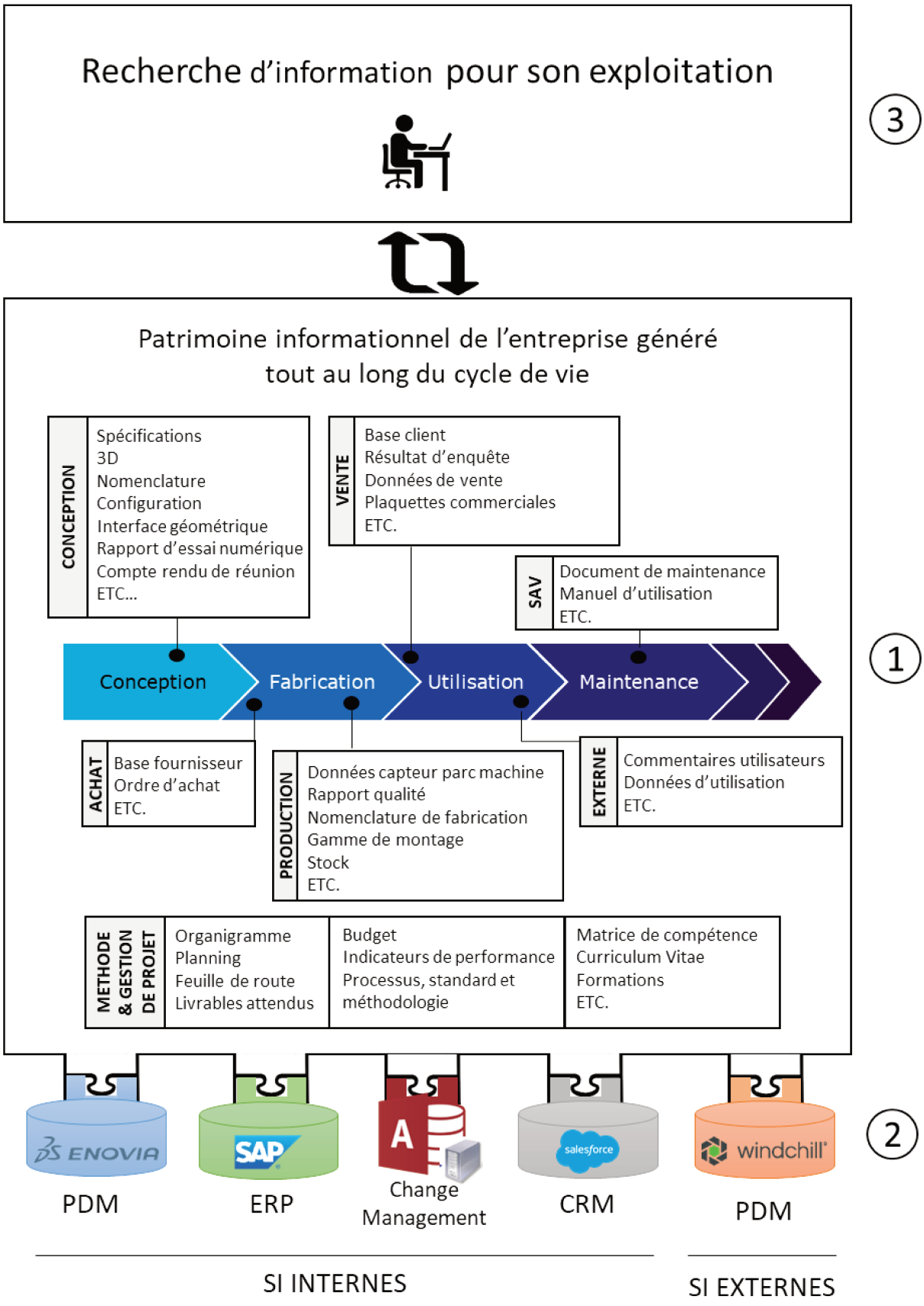


FIG. 1.4 : Illustration du patrimoine informationnel de l'entreprise et de son exploitation dans un environnement modulaire et étendu

1.3.2 Les freins à l'exploitation de l'information

De nombreuses données

L'entreprise accède à toujours plus de données. En dehors des capacités de stockage pouvant être facilitées par des baisses de prix au giga-octet ou par l'accès à de nouveaux services 'Cloud[13]', ce phénomène s'explique à la fois par les tendances d'ultra-personnalisation du produit et donc de la multiplication des configurations à gérer ainsi qu'à la modélisation numérique de toutes les informations avec lesquelles l'entreprise est en relation. Ce dernier point est d'ailleurs encouragé par des approches comme l'IoT (pour l'anglais Internet of Things) permettant d'acquérir numériquement et en temps réel des informations captées de l'environnement physique. Une étude (Rydning 2018) de l'IDC[14] montre que le nombre de données géré par l'industrie manufacturière mondiale est important (estimé à 450 millions de Go en 2018) et en perpétuelle croissance (estimée à 30% en 6 ans). Dans ce contexte, et sans filtrage préalable, il est alors difficile pour l'entreprise d'appliquer des solutions traitant de la totalité des données disponibles. Selon l'enquête de *Forrester* (Taylor 2019), il y a en moyenne 35 à 40% des données des entreprises qui sont inutilisées.

De plus, les données sont de plusieurs natures. On distingue notamment les données structurées des données semi- et non-structurées. Un grand nombre de données structurées de l'entreprise proviennent de bases de données gérées par les Systèmes d'Information. C'est le cas par exemple des tables de produits. Ces derniers ont pour mission de gérer numériquement et de manière centralisée les informations pour un ensemble d'activités. On parle alors des Systèmes de Gestion de Données qui comprend par exemple les ERP (Enterprise Resource Planning), les PDM (Product Data Management), les MES (Manufacturing Execution System) ou encore les CRM (Customer Relationship Management) dont une illustration est apportée par la Figure 1.5. Les données semi- et non-structurées de l'entreprise sont générées tous les jours par une multitude d'acteurs. Elles peuvent par exemple prendre la forme d'un compte rendu de réunion ou d'un livrable comme un manuel d'utilisation ou encore d'un journal d'activités (en anglais, activity log) captées en flux continu. Ces données sont créées par une variété de logiciels et stockées aussi bien localement que déposées sous des serveurs en partage.

Des données hétérogènes

On peut distinguer trois niveaux d'hétérogénéités (Bishr 1998) : (1) le niveau syntaxique marquant la diversité des formats et règles formelles utilisés indépendamment du sens porté, (2) le niveau sémantique marquant la diversité des descriptions utilisées pour une même signification et (3) le niveau schématique marquant la diversité dans l'organisation du modèle de données. Notons toutefois que la littérature inclut parfois le niveau schématique dans la description du niveau sémantique (Yahia 2011 ; Hevner et al. 2004).

[13]En français 'nuage' désigne la sous-traitance de services informatiques réalisée par un tiers et dont la gestion se fait par internet

[14]L'IDC pour International Data Corporation est le premier groupe mondial de conseil et d'études sur les marchés des technologies de l'information

1. L'hétérogénéité syntaxique désigne la divergence dans la grammaire utilisée en dehors de la signification réelle portée (Bishr 1998). L'hétérogénéité syntaxique est générée par la variété des représentations logiques des données utilisées en entreprise. C'est notamment le cas entre des données structurées et non structurées qui ont des règles de formalisation de l'information différentes. Un élément comme l'enregistrement d'un produit dans une base fournisseur peut être l'assemblage d'une décomposition d'attribut tandis qu'un autre élément comme une géométrie de pièce est un assemblage de points de coordonnées spatiales.

2. L'hétérogénéité sémantique désigne qu'*"un fait du monde réel peut avoir plus d'une description dans les bases de données sous-jacentes pour se conformer à diverses disciplines"* (Bishr 1998). L'hétérogénéité sémantique est générée à la fois par la diversité des acteurs et métiers composant l'entreprise mais également par la diversité des éditeurs et intégrateurs des Systèmes d'Information. En effet, il existe des divergences de langage entre chaque individu, chaque service et chaque partenaire. Chacun emploie son propre vocabulaire et langue selon ses expériences, connaissances et 'règles métier'. C'est ainsi qu'un composant comme une batterie pourrait être désigné par sa fonction "batterie du drone" pour le concepteur et par sa référence commerciale "4S5200" pour l'acheteur. Il existe également une divergence de description d'une même information entre les différentes solutions informatiques car chaque éditeur et intégrateur vont effectuer des choix de nommage différents les uns des autres. C'est ainsi qu'une propriété désignant la référence de la donnée pourrait être nommée "référence" par l'un et "titre" par l'autre. Cette diversité sémantique génère des conflits. Les auteurs dans (Goh et al. 1995) distinguent les types de conflits sémantiques suivants : conflit de confusion (signification différente d'un concept selon la temporalité contextuelle comme dans l'interprétation de la phrase "il y a 5 minutes"), conflit d'échelle (utilisation d'unité de mesure différente pour exprimer la même dimension comme dans l'utilisation du "centimètre" ou du "pouce") et conflit de nommage (problèmes taxonomiques et linguistiques comprenant les cas de synonymie comme dans l'utilisation de "distance" au lieu d'"éloigement" ou les cas d'homonymie comme dans l'utilisation de "paire" et "père"). Cette hétérogénéité a suscité de nombreux travaux comme l'étude de la sémantique des modèles de données pour évaluer l'interopérabilité des systèmes d'information d'entreprise (Yahia et al. 2012).

3. L'hétérogénéité schématique désigne quant à elle la diversité conceptuelle de la modélisation du méta-modèle où *"les classes d'objets peuvent avoir des hiérarchies d'agrégation ou de généralisation différentes"* (Bishr 1998). Cette hétérogénéité s'explique par la variété des comportements attendus pour un même modèle ainsi que par la diversité des acteurs et donc de leurs modélisations. Ainsi, les méta-modèles diffèrent les uns des autres impliquant des choix de structure différents. Par exemple, la notion d'appartenance d'un pièce à une classe 'électrique' ou 'mécanique' peut être modélisée par l'ajout d'un attribut (dont les valeurs peuvent être 'électrique' ou 'mécanique') ou modélisée par une relation (entre la pièce et deux nouveaux objets nommés 'électrique' ou 'mécanique').

Ainsi, les données à considérer n'ont ni la même syntaxe, ni la même sémantique et

n'utilisent pas un méta-modèle universel. Autrement dit, il est a priori impossible d'utiliser un langage et une logique unique pour accéder à l'ensemble du patrimoine informationnel de l'entreprise à moins de passer par des étapes de transformation intermédiaires.

Des données distribuées

L'environnement d'une entreprise étendue est composé de silos où les données sont gérées par des systèmes d'information propres à chaque service et partenaires. À moins d'interfaces dédiées, ces silos sont nativement hermétiques les uns aux autres comme les données y sont de syntaxe, sémantique et schéma variés. Cette fragmentation des données suscite de nombreuses études notamment dans l'industrie manufacturière (Shahrokni et al. 2015). Pourtant, ces silos traitent des mêmes produits mais sous des vues métiers différentes. On a alors une distribution des données dans les différents silos qui brise la chaîne numérique possible entre ces différentes vues d'un même produit ou autrement dit d'une même 'entité du monde réel'. Une illustration des données distribuées autour de l'élément 'carte électronique' se trouve à la Figure 1.5.

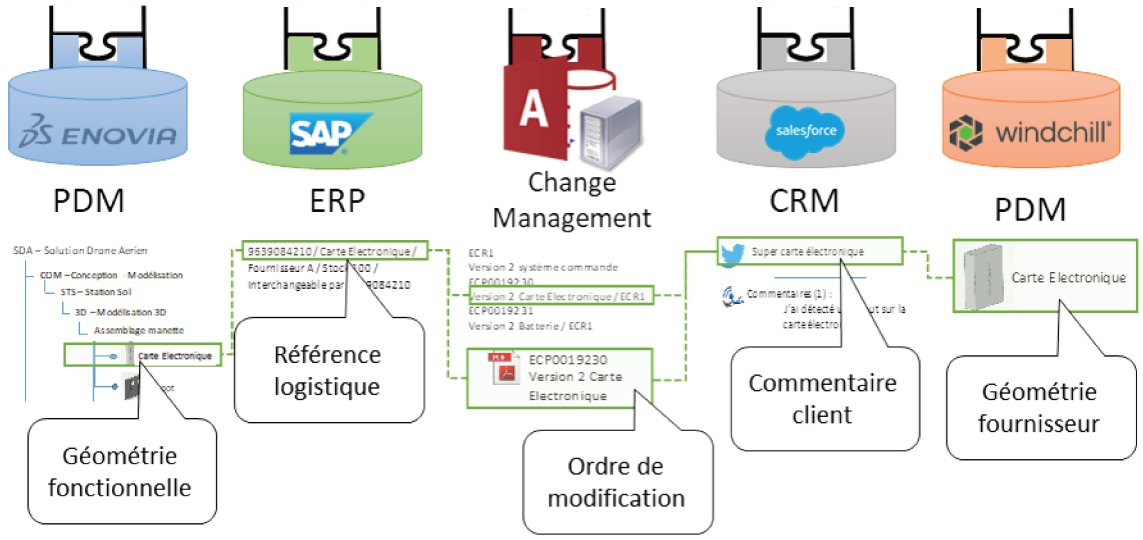


Fig. 1.5 : Illustration du caractère distribué des données représentant la même entité physique 'carte électronique'

1.3.3 Différentes approches pour faciliter l'accès et l'échange d'information

La standardisation pour l'échange d'information

Face au besoin d'échange d'informations pour l'ingénierie collaborative dans un environnement composé de solutions informatiques variées, l'approche par standardisation a été intégrée par les éditeurs. Ainsi, chaque éditeur doit permettre la génération et l'intégration des informations sous un format standard et connu de tous. Le standard STEP ou ISO 10303 (10303-1:2021 2021) est dédié à la représentation et l'échange des données liées au produit lors de son développement et de sa maintenance, notamment par le biais

de la Conception et Fabrication Assistée par Ordinateur. Issu de ce standard, le protocole d'application AP239 ou PLCS[15](10303-239:2012 2012) tente de fournir une modélisation commune et suffisamment riche pour l'échange entre les systèmes d'information tout au long du cycle de vie. Malheureusement, ces approches imposent soit une simplification de l'information pour être partagée à tous mais engendre alors une perte sémantique importante dans les échanges, soit une adaptation de chaque éditeur à l'ensemble des spécificités possibles ce qui engendrerait une complexification incommensurable. Il semble donc difficile d'envisager un accès uniformisé à l'ensemble du patrimoine informationnel uniquement grâce à la standardisation.

Les solutions fédératrices pour garantir la chaîne numérique

Une solution fédératrice aux différents systèmes d'information peut aider à la gestion transversale de l'information. La gestion des données de référence par l'approche Master Data Management (MDM) (Régnier-Pécastaing et al. 2008) en est un bon exemple. La démarche est alors d'interroger le système 'maître' à chaque opération réalisée dans les systèmes 'esclaves' afin de garantir la cohérence entre les données dites 'de référence' sur toute la chaîne numérique. Cette approche nécessite de définir des interfaces d'échanges spécifiques à cette tâche pour chaque système 'esclave' en utilisant par exemple des web services[16] (Milanovic et al. 2004). De plus, il est nécessaire de définir un espace limité d'élément de 'référence'.

Les entrepôts et puits de données pour stocker l'information

Les entrepôts de données (en anglais, Data Warehouse) et les 'puits de données' (en anglais, Data Lake) permettent de regrouper en un lieu un ensemble de données vastes et hétérogènes de l'entreprise. La distinction entre un 'entrepôt de donnée' et un 'puit de donnée' est que le premier stocke des informations transformées selon une stratégie établie tandis que le second stocke les données brutes pour une exploitation ultérieure non définie (Khine et al. 2018). Dans le premier cas, des solutions dites ETL (Extraction Transformation Load) permettent d'extraire les données de leurs différentes bases, de les transformer selon des règles prédéfinies pour enfin les charger dans l'entrepôt. Cette approche nécessite de définir les règles de transformation pour chaque type d'objet en entrée. La seconde approche demande quant à elle de venir transformer les données vis-à-vis du besoin à partir des données brutes et en vrac, mais cette fois-ci en l'absence du lien avec les systèmes sources. Dans les deux cas, une étape de transformation des données est nécessaire pour exploiter les données dans le cadre d'un besoin spécifique.

Les promesses d'une solution PLM unique

L'approche des solutions PLM est de permettre la création et la gestion des données tout au long du cycle de vie du produit. Ainsi, l'objectif est de garantir une conti-

[15]PLCS pour Product Life Cycle Support

[16]Protocole d'interface informatique pour communiquer et échanger entre les systèmes hétérogènes dans des environnements distribués.

nuité numérique, quelles que soient l'étape et l'activité de l'entreprise concernée, unifiant l'ensemble des données en un seul système commun. Elles pourraient alors promettre de remplacer l'ensemble des systèmes d'information comme les PDM, ERP ou CRM et autres logiciels de l'entreprise pour une solution de gestion unique. Si cette stratégie d'unification des silos est proposée par des éditeurs pour des périmètres comme le monde du développement, de la fabrication et de la simulation produit (ARDOUIN 2014), il semble encore difficile d'envisager une solution pour l'ensemble des données de l'entreprise. Tout d'abord, cette solution unique devrait prendre en compte chacune des spécificités de chaque métier restant à la pointe de la technologie et des besoins d'évolutions de processus interne à l'entreprise. Deuxièmement, l'entreprise a la nécessité d'impliquer de multiples partenaires qui ont leurs propres solutions. Enfin, la migration des données historiques de l'entreprise liées par exemple aux programmes dits 'legacy' de certaines grandes industries manufacturières est parfois si coûteuse que le choix de les maintenir dans leurs systèmes d'information d'origine reste plus rentable. L'ensemble de ces raisons réduit donc les chances d'entrevoir demain une solution tout-en-un permettant d'accéder à l'ensemble du patrimoine informationnel à partir d'un seul système.

1.3.4 Différentes solutions pour exploiter l'information

Nous pouvons citer plusieurs types de solutions actuellement utilisées dans l'industrie afin d'acquérir de l'information à partir des données distribuées au sein de l'entreprise.

Les outils de Business Intelligence (BI tools)

Les solutions de Business Intelligence ont suscité un vif intérêt pour l'industrie (Trieu 2017). Ces outils agrègent les différentes données structurées de différents Systèmes d'Informations pour fournir des indicateurs en réponse à un besoin d'analyse et de visualisation de la donnée. Néanmoins, la nature de ces outils oblige à chaque nouveau besoin d'indicateur une prise en charge par le département informatique ou un expert de la solution. C'est l'une des critiques de l'industrie face à ces solutions, elles dénoncent en effet un besoin de 'self-service' dans le processus d'analyse des données (Imhoff et al. 2011). Le souhait de l'industrie est d'avoir des solutions agiles, capable de s'adapter sans délai à de nouveaux besoins de croisement de données. De plus, les données non structurées et plus particulièrement leurs contenus textuels n'y sont pas exploités, imposant pour cela l'application de traitements supplémentaires sur la donnée comme l'extraction sémantique automatique (Baars et al. 2008).

La fouille de données (ou 'data mining' en anglais)

Le processus de fouille de données est appliqué afin de faire émerger d'un ensemble de données important un modèle non évident (Jiaying Liu et al. 2019). Ce processus peut être vu comme un projet à part entière. Il impose d'immobiliser des compétences spécifiques comme des 'data scientist/analyst' pour un besoin précis et prédéfini. Pour illustrer ce point, prenons l'exemple des modèles de processus de fouille de donnée "KDD (Knowledge Discovery Databases)", "CRISP-DM (Cross Industry Standard Process for Data Mining)"

ou encore "SEMMA (Sample, Explore, Modify, Model, Assess)" représentatifs du domaine. Pour révéler une information définie pour un objectif métier, les trois modèles inclus des étapes de préparation puis de manipulation de la donnée à partir d'un échantillon de donnée cible (Shafique et al. 2014). De ce fait, si l'objectif métier change c'est l'ensemble de ces étapes du processus qui change rendant ainsi le processus non agile à la fluctuation et l'évolution du besoin d'information initial. Enfin, les processus de fouille de données sont appliqués à des données structurées sauf dans le cas de l'extraction de connaissance appliqué à du texte alors appelé 'fouille de textes' ('text mining' en anglais).

La recherche d'information d'entreprise

La recherche d'information d'entreprise (ou 'enterprise search' en anglais) (Y. Li et al. 2014) fait appel au domaine de la Recherche d'Information (RI) et permettent à l'instar des moteurs de recherche web d'obtenir une liste de résultats en réponse à une requête utilisateur. Le Système de Recherche d'Information (SRI) peut rechercher dans le contenu textuel des données non-structurées (recherche dans les pages web par exemple) mais également dans les propriétés des données structurées (filtrage par métadonnées dans la recherche d'une base de donnée par exemple). Il peut être intégré à un système d'information par l'éditeur (mais se limite alors aux données que seul celui-ci gère) ou bien être ajouté par l'entreprise afin d'interroger l'ensemble des informations qui lui sont connectées. À l'opposé des outils de BI et des processus de fouille de donnée, une nouvelle requête utilisateur (ou autrement dit un nouveau besoin d'information énoncé à l'instant t) n'engendre pas un nouveau projet informatique. Néanmoins, la solution se limite la plupart du temps à la restitution des résultats sans aides supplémentaires à leurs exploitations et leurs analyses. De plus ces solutions exploitent des données stockées dans des entrepôts de données engendrant de potentielles pertes d'informations comme les relations entre les données.

Il existe donc plusieurs types de solutions permettant plus ou moins l'exploitation de l'information après son acquisition. Des solutions comme les moteurs de recherche d'entreprise sont particulièrement agiles à la variété des besoins de recherche d'information mais n'exploitent pas a priori l'ensemble des caractéristiques des données sources comme les relations entre-elles. Ils permettent par contre d'exploiter à la fois des données structurées et non structurées sans traitements supplémentaires.

1.4 Question de recherche

1.4.1 La question de recherche

En synthèse des précédentes sections, cette thèse cherche à contribuer à l'enjeu majeur de la valorisation du patrimoine informationnel de l'industrie manufacturière malgré les nombreuses données hétérogènes, distribuées et implicitement liées. Il n'existe pour l'instant pas d'approche évidente pour faciliter l'accès et l'échange d'information, néanmoins concernant le manque d'agilité aux besoins d'informations et l'exploitation des données structurées et non structurées, la solution de type 'moteur de recherche d'entreprise' se

démarque.

Ces principaux éléments nous ont aidés à formuler la question de recherche présentée dans la Figure 1.6.

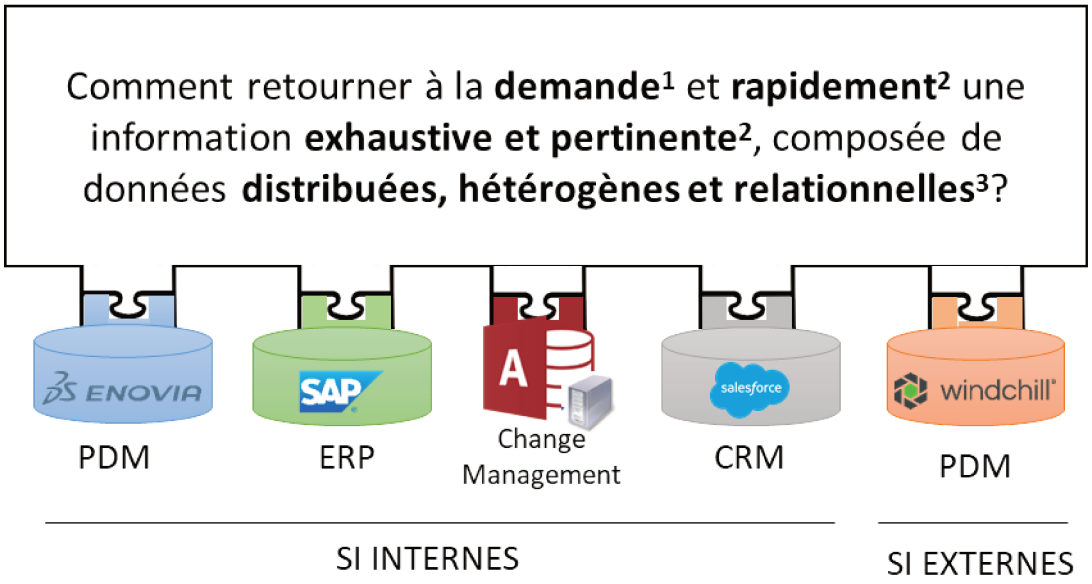


FIG. 1.6 : Illustration de la question de recherche dans son contexte

[1] **Retourner à la demande** évoque l'agilité d'acquisition variée d'informations provenant de l'entreprise.

[2] **Rapidement** signifie que l'acquisition de l'information doit être rapide, **de manière exhaustive** signifie que tous les résultats pertinents doivent être restitués et **de manière pertinente** signifie qu'aucun résultat non pertinent ne doit être restitué.

[3] **Adapté au caractère distribué, hétérogène et relationnel** des données suppose que ces éléments, définis dans l'étude du contexte, sont pris en compte afin de limiter la perte d'information.

1.4.2 Objectif et exigences

L'objectif de la thèse est de répondre à la question de recherche en proposant une approche à la fois théorique et pratique que nous évaluons selon plusieurs exigences. En effet, pour évaluer la réponse proposée par la thèse, nous devons au préalable établir les exigences de performance (je souhaite que ma solution fournisse des résultats performants sur la base de mesures quantitatives) et les exigences de structure (je souhaite que ma solution soit efficace face à mon enjeu sur la base de mesures qualitatives). Nous utiliserons dans cette optique le carré de validation présenté dans (SEEPERSAD et al. 2006).

Exigences de performance - mesures quantitatives

Répondre à la question de recherche, c'est le faire correctement pour les différents besoins d'informations transposés par les usages attendus précédents. La solution doit

considérer plusieurs exigences :

- Exigence 1 : la solution doit être capable de fournir toutes les informations attendues et éviter les silences[17]
- Exigence 2 : la solution ne doit fournir que des informations pertinentes et éviter les bruits[18]
- Exigence 3 : en réponse à l'enjeu présenté en Section 1.3.1, le temps de recherche entre l'émission du besoin d'information et son obtention doit être faible.

Exigences de structure - mesures qualitatives

La réponse à la question de recherche doit donner des résultats performants, mais également être efficace face aux enjeux. Nous listons deux critères supplémentaires :

- Exigence 4 : en réponse aux principales critiques présentées en Section 1.3.4, l'**agilité** de la solution à répondre aux différents besoins d'informations de l'entreprise est un premier critère à considérer. La solution ne doit donc pas imposer d'opérations coûteuses pour s'adapter à des besoins de recherche d'information non prévus.
- Exigence 5 : en réponse à l'environnement modulaire présenté en Section 1.3.1, l'**interopérabilité** de la solution est également un critère important. La solution doit être agnostique, afin de limiter les coûts (de temps et de moyens) d'adaptations structurelles.

1.5 Synthèse et contributions

La thèse présentée dans ce mémoire, réalisée au sein des laboratoires LISPEN et LCPI de l'Ecole Nationale Supérieure d'Arts et Métiers et la société en service du numérique Capgemini, contribue à la valorisation du patrimoine informationnel dans l'industrie manufacturière. Cette valorisation est réalisée dans un contexte où les données sont nombreuses, hétérogènes, explicitement et implicitement liées entre les différents systèmes d'information qui sont quant à eux variables dans le temps. Le souhait est d'apporter une solution transversale aux différentes activités de l'entreprise en répondant à la question de recherche suivante **"Comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?"**. Dans cette optique, la contribution de la thèse se décompose en :

- un état de l'art sur l'approche par système de recherche d'information et la modélisation graphe des données hétérogènes applicable à notre contexte (voir Chapitre 2),

[17] En science de l'information, le silence se réfère aux réponses pertinentes non restituées alors qu'elles existent

[18] En science de l'information, le bruit se réfère aux réponses non pertinentes restituées

- une proposition de réponse à la question de recherche (voir Chapitre 4) respectant un cadre prédéfini (voir Chapitre 3),
- la définition par expérimentation des défis clés à considérer pour enrichir la proposition (voir Chapitre 4),
- l'enrichissement de la proposition vis-à-vis de ces défis clés (voir Chapitre 5),
- le prototypage de la proposition ainsi que sa validation (voir Chapitre 6).

Afin de favoriser les échanges avec la communauté et contribuer au partage des connaissances, nous rendons également disponibles les différents jeux de données utilisés sous une plateforme de partage de jeu de données applicables aux exercices de science des données. Les liens permettant d'accéder à ces jeux de données sont fournis en Section 3.4.4 et en Section 6.3.3.

Chapitre 2

État de l'art

2.1 Objectif du chapitre

Comme illustré dans la Figure 2.1, l'objectif de ce chapitre est de présenter un état de l'art relatif à notre question de recherche "Comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?".

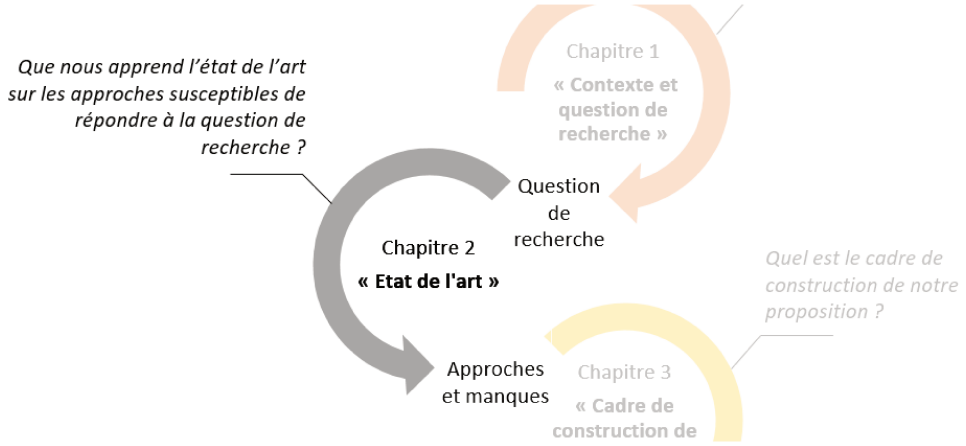


Fig. 2.1 : Organisation du mémoire : second chapitre

Le chapitre est décomposé en trois parties : tout d'abord, nous présentons le domaine de la recherche d'information dont les systèmes permettent de renvoyer de l'information à la demande, puis nous présentons celui de l'approche graphe permettant d'exploiter le caractère relationnel des réseaux d'informations. Enfin, nous présentons un état de l'art des travaux à l'intersection de ces deux domaines appliqués aux entreprises, c'est-à-dire la recherche d'information en entreprise puis celle utilisant les graphes. Nous soulignons également dans cette partie l'absence de discussion autour de ce sujet appliqué à l'industrie manufacturière, contexte nous intéressant particulièrement. La représentation des différents domaines d'intérêt et de leurs intersections est illustré dans la Figure 2.2.

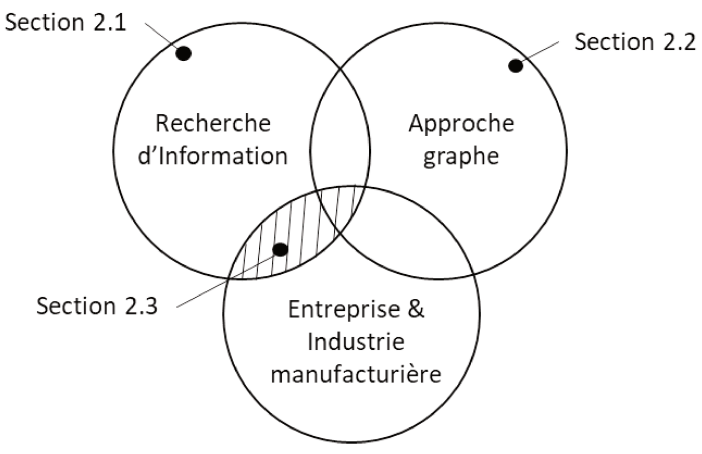


Fig. 2.2 : Domaines d'études et ses intersections

2.2 La recherche d'information

Le domaine de la Recherche d'Information (RI) a pour objectif répondre au besoin d'identification d'informations à partir d'une collection de documents. Pour y parvenir, les Systèmes de Recherche d'Information (SRI) évaluent la pertinence des informations disponibles vis-à-vis d'un besoin exprimé par une requête utilisateur afin de restituer à la fois tous les documents pertinents (et ainsi éviter les silences) tout en excluant tout résultat non pertinent (et ainsi éviter les bruits). Une des premières définitions du domaine est proposée en 1968 par Salton : *"Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information"* (Gerald Salton 1968). Quarante ans plus tard, des définitions proches sont encore données comme dans l'ouvrage (Amati 2003) *"Information retrieval is an empirical science that studies representation, storage, and access to information [...]"*.

2.2.1 Le processus de recherche d'information

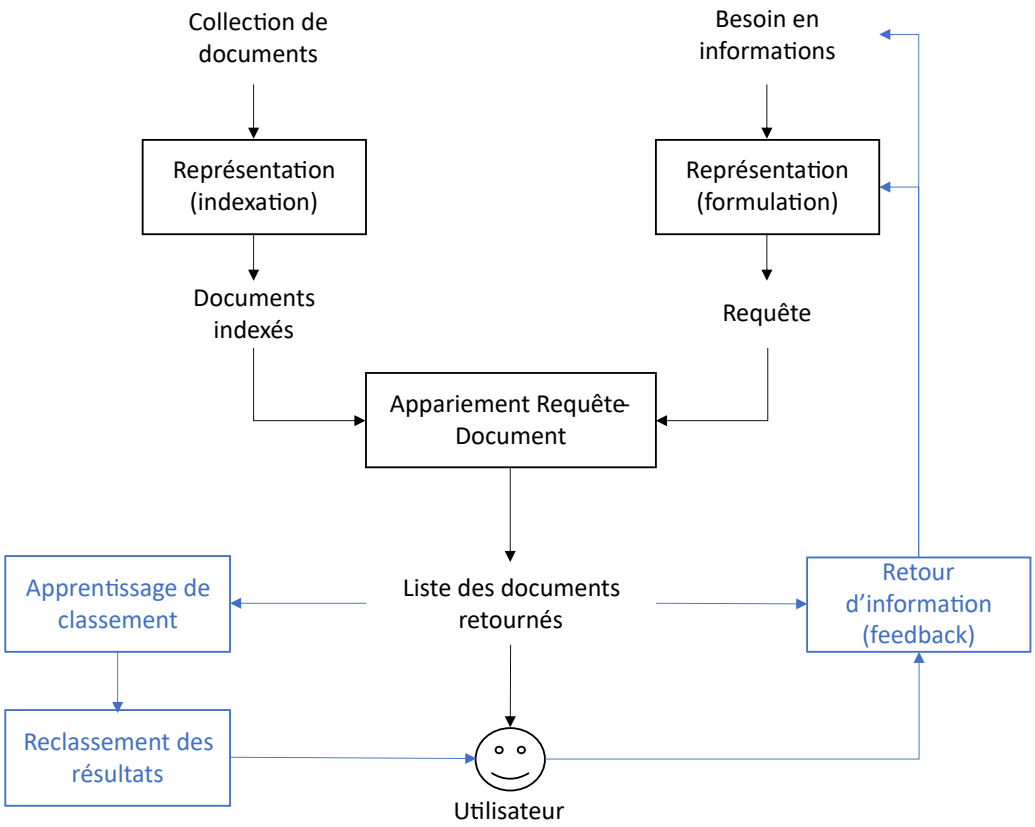


Fig. 2.3 : Illustration du processus de recherche d'information en U adapté de (Alkilinç et al. 2018)

Le processus de recherche d'information dans sa forme la plus simple est modélisé en U (Gerard Salton et al. 1983b ; Belkin et al. 1992) comme présenté en éléments noirs dans la Figure 2.3. Les premières solutions d'automatisation de ce processus remontent aux années 1950 (Luhn 1957). Le processus est composé des trois étapes suivantes :

- **La représentation des documents - indexation** : afin d'optimiser la recherche (temps d'acquisition et pertinence des résultats), une étape d'indexation des documents à disposition est nécessaire. Cette étape consiste à analyser les documents de la collection afin d'en établir les descripteurs importants (concepts importants portés par les documents). Ces descripteurs constituent alors une représentation des documents permettant d'y accéder lors de la recherche d'information survenant par la suite.
- **La représentation du besoin utilisateur - requête** : un besoin utilisateur exprimé à travers une requête peut se former de mots-clés avec ou sans opérateurs logiques entre eux ou encore par langage naturel. Selon (Kleinberg et al. 2011), ces requêtes peuvent être classées selon trois types. La requête spécifique (« quelle est l'exigence d'autonomie du drone »), large (« trouve les informations sur la batterie du drone ») ou par similarité (« trouve les documents similaires à cet autre document »).
- **L'appariement entre le besoin et les documents - sélection d'une liste ordonnée des documents** : cette étape permet de prédire la pertinence de chaque document d'une collection selon la requête. La pertinence est représentée par un score de similarité nommé également « Relevance Status Value » et que l'on note $RSV(d, q)$ où d représente le document et q la requête. C'est ce score qui permet de renvoyer une liste ordonnée de résultats, du plus au moins pertinent.

Cette formalisation du processus peut également être enrichie pour y faire apparaître des étapes complémentaires comme l'utilisation des résultats obtenus et de leurs analyses par l'utilisateur entraînant une reformulation du besoin, une reformulation de la requête ou encore un nouvel ordonnancement des résultats par exemple (Alkilinç et al. 2018). Ces éléments sont illustrés par les éléments en bleus de la Figure 2.3.

2.2.2 Indexation des documents

La représentation des documents est un enjeu majeur dans l'automatisation du processus de RI. En effet, une bonne représentation de la collection doit permettre d'accéder rapidement au bon document tout en évitant les bruits et les silences. Ainsi, une bonne représentation des documents devrait contenir les informations utiles et uniquement celles-ci pour les besoins d'informations ultérieurs tout en pensant sa structuration pour l'optimisation du temps de réponse.

Dans les méthodes conventionnelles de RI, nous parlons plus précisément d'indexation. L'indexation a pour objectif de lister dans un index l'ensemble des termes descriptifs dits « descripteurs » des documents de la collection considérée (Gerard Salton et al. 1983b). Chaque descripteur est alors associé à tous les documents s'y réfèrent ce qui permet alors d'accéder rapidement aux documents à partir du terme descriptif.

Comme distinguée par (Dinh 2012), l'indexation peut soit être établie de manière libre à partir de la collection soit de manière contrôlée avec un vocabulaire connu et sélectionné à priori. Bien que l'indexation libre soit plus facile et rapide à mettre en œuvre, l'indexation contrôlée permet d'exploiter les connaissances à priori et de les formaliser. Prenons par exemple le cas d'une recherche d'information dans un domaine particulier, l'indexation

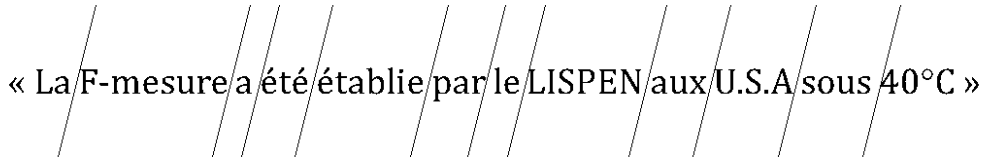
pourrait alors s'appuyer sur sa terminologie spécifique ainsi que sur les relations entre termes connues (hiérarchique, équivalence, association, etc.).

On distingue quatre étapes fondamentales dans l'indexation automatique. Illustrons ces étapes avec l'indexation de la phrase suivante :

« La F-mesure a été établie par le LISPEN aux U.S.A sous 40°C »

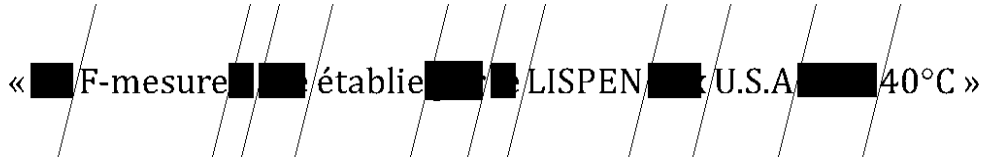
L'analyse lexicale, segmentation ou encore tokenization

Cette étape consiste à segmenter le texte en unité lexicale (ou token), tout en prenant en compte les ponctuations, les espaces ou les chiffres. En reprenant notre exemple, l'opération donne alors en résultat les unités lexicales suivantes :



La sélection ou nettoyage

Cette étape consiste à utiliser des stop-list (ou anti-lexique) supprimant les tokens peu porteurs de sens, dits mots vides, comme les pronoms personnels, les articles, les mots de liaison ou les prépositions. De notre exemple, nous ne retiendrons donc que les descripteurs suivants :



La normalisation ou racinisation ou lemmatisation

Cette étape consiste à homogénéiser les termes soit en se rapportant à leurs racines (ou stemme) soit en se rapportant à leurs lemmes. Dans notre exemple, le mot « établie » sera rapporté soit à sa racine « établi » soit au lemme « établir » tout comme les autres termes variants de cette racine. On obtient, avec le choix de la lemmatisation l'index suivant :

Descripteurs
F-mesure
Etablir
LISPEN
U.S.A
40°C

La pondération

La dernière étape est de pondérer les descripteurs par une valeur numérique. On évalue ainsi les descripteurs selon leur pouvoir de représentativité du document. Il existe plusieurs fonctions de pondération dont la fonction TF-IDF (term frequency-inverse document frequency) dont la définition et une étude de son utilisation sont données dans (Qaiser et al. 2018). Cette fonction prend en considération la loi de distribution des termes dans un texte formellement énoncé par la loi de Zipf (Zipf 1949). Cette loi énonce que pour retrouver les termes les plus représentatifs d'un texte, on doit à la fois éliminer les termes à très hautes fréquences (peu porteur de sens) et à très faibles fréquences (erreur de frappe, néologisme[1]). Soit t le terme dans le document d, $tf_{t,d}$ le nombre d'occurrences du terme t dans le document d, N le nombre total de document dans la collection et df_t le nombre de documents contenant le terme t . Le poids du terme t pour le document d est donné par :

$$w_{t,d} = tf_{t,d} * log\frac{N}{df_t} \tag{2.1}$$

Cette formule est la plus courante notamment grâce à sa simplicité de calcul mais il existe également d'autres modèles prenant en compte d'autres critères comme la longueur des documents (Singhal et al. 2017) ou les lois de probabilités (modèle BM25 (S. Robertson et al. 2009) ou modèle étendu du modèle TF-IDF (Aizawa 2003)).

On obtient pour l'exemple la pondération suivante :

Descripteurs	Pondération
40°C	1.4
LISPEN	1
U.S.A	1
F-mesure	0.7
Etablir	0.3

2.2.3 Formulation du besoin d'information

Expression du besoin informationnel

Dans (Pradel 2013), l'auteur distingue les besoins d'informations par leur type de réponse : celles impliquant une réponse factuelle, l'expression d'une opinion ou un résumé. La première catégorie, la plus commune, est ensuite divisée selon si les questions sont dichotomiques (réponse en oui ou non), en dénombrement (réponse numérique) ou en liste (réponse par une liste d'éléments). Ceux sont ces derniers qui concernent les SRI standard.

Selon (Latour 2014), le besoin informationnel dans les SRI peut se traduire par trois méthodes : l'exploration thématique, le langage de requêtes et le langage naturel.

[1]Néoligisme : tout mot de création récente ou emprunté depuis peu à une autre langue ou toute acception nouvelle donnée à un mot ou à une expression qui existaient déjà dans la langue. - Dictionnaire Larousse, 2021

L'exploration thématique permet d'accéder à l'information par l'exploration de catégorie et sous-catégorie proposée par le système. Ces catégories peuvent prendre la forme d'arborescence, de liens hypertextes ou de cartographie sémantique. L'avantage est ici que l'utilisateur est guidé dans son exploration et n'aura donc pas besoin de connaissances du vocabulaire employé. Il devra néanmoins explorer manuellement la collection pour trouver l'information recherchée. Le langage de requête permet quant à lui d'exprimer son besoin par des mots-clés via un formulaire de recherche. C'est une approche simple pour l'utilisateur, offrant également des possibilités de recherche avancée comme dans l'utilisation d'opérateur de recherche (ET, OU, PROCHE etc.) ou par l'ajout de filtres complémentaires (métadonnées des résultats comme l'auteur, la date etc.). Enfin, le langage naturel, grâce aux outils de Traitement Automatique du Langage Naturel (TALN) en français ou 'Natural Language Processing (NLP) en anglais, peut être exploité pour exprimer le besoin d'information. L'objectif ici est de ne pas contraindre l'utilisateur à transformer l'expression de son besoin en un nouveau langage compréhensible par le SRI. L'intégration du besoin est néanmoins plus complexe à traiter par le système.

Enfin, selon (Kefi-Khelif 2006), la syntaxe de la requête contient des unités d'interrogations (descripteurs du besoin sous forme de syntagme[2] dans la représentation courante en RI) dont on peut distinguer le *focus* correspondant à l'information nouvelle apportée par la réponse (Beyssade 2006).

Extension de requêtes

L'extension de requête est la méthode permettant d'améliorer les performances de RI par l'ajout de nouveaux termes à la requête (Efthimiadis 1996). Dans (Ooi et al. 2015) et (Raza et al. 2019), les auteurs distinguent l'approche par utilisation d'un modèle de connaissance selon le corpus ou non (utilisation de réseaux sémantiques tels des thésaurus ou des ontologies (Bhogal et al. 2007)), par l'utilisation d'un modèle de langue[3] donnant les probabilités de relation entre termes et par l'utilisation d'un retour de pertinence des résultats. La première et seconde approche permet d'avoir à disposition la fonctionnalité dès l'initialisation du système ce qui n'est pas le cas de la troisième, néanmoins elles ne sont pas adaptées à l'utilisateur contrairement à la dernière. Celle-ci peut être réalisée à "l'aveugle" (les termes employés dans les résultats sont associés aux termes employés dans la requête), "explicitement" (une évaluation des résultats est demandée pour associer les termes) et "implicitement" (observation du comportement de l'utilisateur pour évaluer les résultats à sélectionner). L'étude de (Azad et al. 2019) montre notamment qu'une des tendances du domaine est d'utiliser une approche hybride entre les plusieurs possibilités vues précédemment, notamment utiliser des modèles de connaissance indépendants et dépendants du corpus en complément de retours de pertinence afin d'augmenter la précision de l'extension.

[2]syntagme : en linguistique structurale, groupe d'éléments formant une unité dans une organisation hiérarchisée. - Dictionnaire Larousse, 2021. Par exemple, "Le petit chien blanc de mon voisin" est un syntagme.

[3]Construire un modèle de langue statistique consiste à estimer une distribution de probabilité sur des séquences de mots[1] dans une langue particulière. Il permet ainsi d'assigner une probabilité à une séquence de mots - https ://www.smalsresearch.be/nlp-modeles-de-langue/$_f tn1 consult le 27/07/2021$

Suggestion et raffinement de requêtes

Dans (Ooi et al. 2015), les auteurs distinguent deux autres domaines cherchant à résoudre l'ambiguïté et l'inexactitude possibles dans l'expression de la requête. La première est celle de la suggestion de requête proposée à l'utilisateur à la suite de l'analyse de son comportement (requêtes déjà formulées dans la même session utilisateur, dans une session passée ou celle d'un autre utilisateur). La seconde est celle du raffinement de la requête la transformant pour être plus proche du besoin d'information initial.

2.2.4 L'appariement

Le domaine distingue trois grandes classes d'approches mathématiques afin d'établir le score de pertinence à l'appariement entre un document et une requête : les modèles ensemblistes basés sur la théorie des ensembles, les modèles algébriques et les modèles probabilistes (Baeza-Yates et al. 1999 ; Singhal et al. 2001 ; Dong et al. 2008 ; Dinh 2012).

Les modèles ensemblistes

Le modèle booléen est le premier modèle qui s'est imposé dans le monde de la recherche d'information. Il utilise les opérateurs logiques ET , OU et NON pour lier chaque terme utilisé dans la requête. Le document évalué est donc jugé pertinent si l'opération booléenne est vraie, et non pertinent sinon. Pour une opération booléenne q définissant la requête et un document d_i , le score de similarité $RSV(d_{i,q})$ équivaut à :

$$RSV(d_{i,q}) = \begin{cases} 1 & \text{si } q(d_i) = 1 \\ 0 & \text{sinon.} \end{cases} \tag{2.2}$$

Ce modèle a l'avantage d'être simple à implémenter et transparent pour l'utilisateur. Ce dernier assure que tout résultat qui lui est renvoyé correspond exactement à l'expression logique de la requête. Son principal inconvénient est de ne pas renvoyer une liste de résultats ordonnés (du plus au moins pertinent) afin de guider l'utilisateur dans le nombre potentiellement important de résultats et nécessite une traduction du besoin en requête booléenne.

Le modèle booléen étendu intègre la notion de pondération des termes afin d'établir une liste de résultats ordonnés, de plus au moins pertinent mais demande néanmoins un coût informatique non négligeable (Dong et al. 2008). Le modèle a été introduit en 1983 par (Gerard Salton et al. 1983a).

Connaissant le poids des termes pour un document et dans une représentation de ce dernier dans un espace à m dimensions (m étant le nombre de termes dans la requête), les coordonnées du document sont $d_i = w_{i1}, w_{i2}, ..., w_{im}$ où w_{im} est le poids calculé de chaque terme T de la requête q.

Le modèle des ensembles flous fait appel à la logique floue introduite par Zadeh (Goguen 1973) et largement utilisée depuis, notamment dans le traitement des Big

Data (Wang et al. 2017). Elle tend à simuler le raisonnement humain puisque le cerveau apprécie des variables d'entrées et de sortie approximatives du type "le terme est peu/moyennement/très présent dans le document". La logique floue intègre ainsi la notion de "degré de vérité" noté $_{A(x)}$ de la proposition "x appartient à l'ensemble flou A". Ce degré de vérité va permettre d'ordonner les résultats dans une adaptation à la recherche d'information (Gerard Salton 1989 ; Miyamoto 1990).

L'avantage de ce modèle est de proposer un ordonnancement des documents pertinents selon le degré d'appartenance des documents aux termes de la requête, sachant que ces termes peuvent être pondérés. Néanmoins, la création des fonctions d'appartenance est manuelle. Le coût induit par l'expertise nécessaire est donc un critère à prendre en compte. De plus, comme le soulignent les auteurs dans (Dinh 2012 ; Bosc et al. 2009), certaines opérations booléennes sur les ensembles flous peuvent ordonner les documents de manière non pertinente.

Les modèles algébriques

Le modèle vectoriel défini par Salton en 1975 (Gerard Salton et al. 1975) représente le document sous forme de vecteur dont les coordonnées sont les poids des termes dans un espace vectoriel à t dimensions, t étant le nombre total de termes uniques de la collection de documents. La requête est également représentée par un vecteur dans le même espace vectoriel et suivant le même principe. La pertinence de d_i vis-à-vis de q est alors calculée selon la proximité de leurs représentations vectorielle. Il existe plusieurs méthodes afin de calculer la proximité des vecteurs. Dans les plus citées pour la RI, la mesure du cosinus est la plus populaire (Turney et al. 2010) et la mesure de Dice et de Jaccard sont les plus performantes (Curran et al. 2002).

Le modèle permet d'ordonner les résultats et de retourner des documents ne répondant ainsi que partiellement à la demande. Néanmoins, la principale critique de ce modèle est qu'il ne prend pas en compte la dépendance entre les termes, dans le cas d'utilisation d'un synonyme du terme de la requête par exemple.

Le modèle vectoriel généralisé défini dans les travaux de (Wong et al. 1985) propose de supprimer le critère d'orthogonalité deux à deux des vecteurs termes et d'introduire un facteur de corrélation entre les termes. Le principal avantage est donc de prendre en compte la corrélation entre les termes. Néanmoins, il nécessite un coût de calcul supplémentaire critiqué vis-à-vis des gains réels espérés (Tamine 2000 ; Sauvagnat 2005 ; Dong et al. 2008).

Le modèle d'indexation sémantique latente proposée par Furnas (Furnas et al. 2017) réduit l'index contenant des termes descripteurs de la collection à un index contenant des concepts qui expriment la sémantique cachée des termes (d'où la notion de sémantique latente). Ces concepts permettent notamment de regrouper les termes synonymes en un concept ou à l'inverse distinguer les termes polysémiques en deux termes distincts.

Le modèle connexionniste (ou neuronal) utilise les réseaux de neurones. L'adaptation du réseau de neurones pour l'appariement requête-document date de la fin des années 1980 (Kwok 1989). L'avantage principal du réseau de neurones est sa capacité

d'apprentissage lui permettant d'être adaptatif à l'exercice d'appariement. Néanmoins, il peut être jugé de "boîte noire" car l'utilisateur n'accède pas au processus de prise de décision et l'entraînement du réseau est coûteux en termes de calcul (Uddin et al. 2019).

Les modèles probabilistes

Le principe de classement probabiliste (S. E. Robertson 1977) traduit le fait de mettre en avant les documents lorsqu'ils ont à la fois une forte probabilité de pertinence P et une faible probabilité de non-pertinence $\overline{P}$ par rapport à la requête. Afin d'estimer ces probabilités, plusieurs modèles existent comme le modèle BIR (Binary Independence Retrieval) ou le modèle BM25. Ce dernier étant souvent associé au terme "Okapi" du nom du premier système de recherche au sein de l'université de Londres qui l'utilisait (S. E. Robertson et al. 1995). Considérant la distribution des mots dans les documents, la longueur du document et la longueur moyenne des documents de la collection, ce modèle est l'un des plus importants du domaine.

Le modèle de langue ne calcule pas la similarité entre une requête et un document, mais calcule la probabilité que le document puisse être associé la requête (Ponte et al. 1998). On estime alors un modèle de langue du document noté Θ_d qu'on utilise pour l'appariement requête-document et on formule alors que le score de similarité est la probabilité que la requête soit pertinente sachant Θ_d.

Le réseau d'inférence bayésien permet de calculer des probabilités conditionnelles d'un réseau représenté en graphe. Pour la recherche d'appariement requête-document (Turtle et al. 1989) le graphe est composé de nœuds étant les variables requête q, termes t_i et documents d_i et d'arcs étant les relations de cause à effet entre les variables. Le réseau permet de calculer la probabilité que le document soit pertinent sachant la requête $P(\frac{d}{q})$. L'avantage du modèle est de rendre compte des liens entre les termes. Les performances du modèle INQUIRY, proposant des opérateurs afin de créer le réseau de manière automatique en prenant en compte des opérateurs de proximités et de synonymie entre les termes, ont étés prouvés (Callan et al. 1995). Néanmoins, le coût de calcul des probabilités à travers le réseau bayésien est important.

2.2.5 L'évaluation en recherche d'information

Les mesures

Cleverdon s'est penché sur l'évaluation des SRI dès la fin des années 1960 (Cleverdon 1967). Afin d'évaluer les SRI, les mesures établissent la pertinence des documents retournés selon une requête et considèrent donc q la requête appartenant à l'ensemble Q des requêtes, Np le Nombre de documents Pertinents et Nr le Nombre de documents Restitués. Les ensembles Np et Nr sont illustrés dans la Figure 2.4.

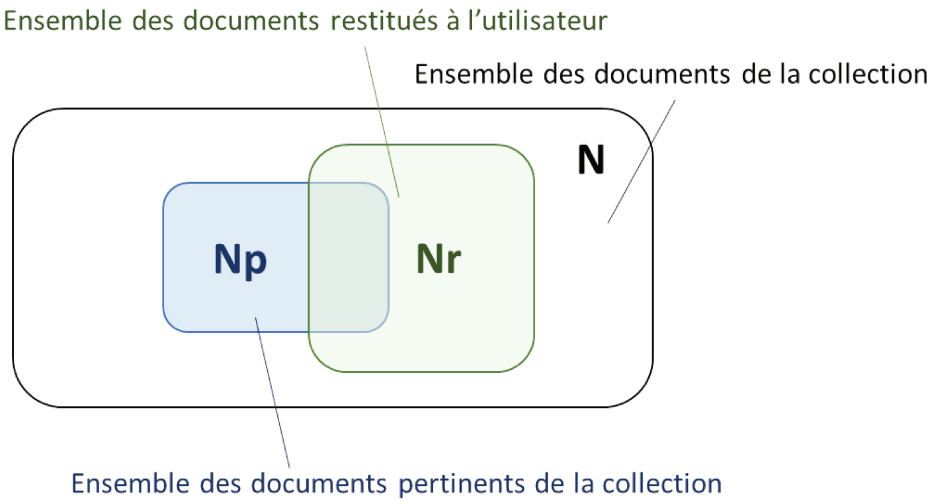


FIG. 2.4 : Répartition des documents pour l'évaluation en Recherche d'Information

Le rappel

Le rappel évalue la capacité du SRI à restituer l'ensemble des réponses pertinentes à l'utilisateur. Il va donc discriminer les systèmes avec un fort silence (absence des résultats pertinents en réponse). Le rappel est exprimé par le rapport du nombre des documents pertinents restitués par le nombre total des documents pertinents :

$$Ra = \frac{1}{Q} * \sum_{q \in Q} \frac{|Np \cap Nr|}{|Np|}$$

La précision

La précision évalue la capacité du SRI à restituer des résultats pertinents à l'utilisateur. Il va donc discriminer les systèmes fournissant du bruit (de mauvaises réponses en résultat). La précision est exprimée par le rapport du nombre des documents pertinents restitués par le nombre total des documents restitués :

$$Pr = \frac{1}{Q} * \sum_{q \in Q} \frac{|Np \cap Nr|}{|Nr|}$$

Il existe également la notion de mesure de hautes précisions ou P@X qui mesure la précision pour les X rangs considérés. Exemple P@10 : précision considérant uniquement les 10 premiers résultats retournés.

La moyenne harmonique

La moyenne harmonique, également appelée F-Mesure, permet de combiner précision et rappel dans une évaluation pondérée par β précisant si l'on accorde plus de poids à

l'un qu'à l'autre. Par exemple, si $\beta = 2$ la précision est deux fois plus importante que le rappel. La F-Mesure est exprimée ainsi :

$$F_\beta = \frac{1}{Q} * \sum_{q \in Q} (1 + \beta^2) * \frac{Pr * Ra}{\beta^2 Pr + Ra} \tag{2.3}$$

La courbe du rappel et de la précision

Afin d'établir une courbe du rappel et de la précision permettant visuellement d'évaluer le système de recherche d'information, on évalue le rappel et la précision à chaque document pertinent considérant que le nombre total de documents retournés à chaque rang R est la valeur du rang. On obtient alors, comme l'illustre la Figure 2.5[4], une courbe qui plus elle descend lentement, meilleure est l'évaluation du système.

Ra	Pr
0,07	1,00
0,13	0,50
0,20	0,75
0,27	0,67
0,33	0,71
0,40	0,67
0,47	0,64
0,53	0,67
0,60	0,64
0,67	0,67
0,90	0,01

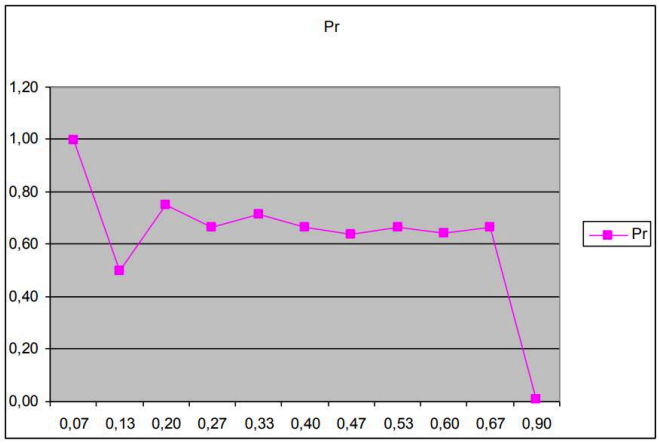


FIG. 2.5 : Exemple de courbe du rappel et de la précision

La précision moyenne non interpolée

Cette mesure appelée également MAP pour Mean Average Precision, considère les rangs des résultats pertinents trouvés afin d'évaluer la précision moyenne en traduisant la qualité du classement. La précision moyenne non interpolée est exprimée ainsi :

$$MAP = \frac{1}{Q} * \sum_{q \in Q} \frac{1}{Np} \sum_{i \in Np} \frac{i}{rang_i} \tag{2.4}$$

où i est le document pertinent retourné et $rang_i$ est le rang auquel i est apparu dans les résultats.

La mesure de hautes précisions, la courbe de rappel/précision et la précision moyenne non interpolée sont utilisées pour des méthodes d'appariement considérant l'ordonnance-

[4]Extrait du cours Recherche d'Information à l'Université Paul Sabatier - https ://www.irit.fr/ Mohand.Boughanem/slides/RI/chap11-Evaluation.pdf accès le 28/07/2021

ment des résultats restitués. En effet, ces mesures permettent de juger de la capacité du SRI à fournir un ordonnancement performant.

Les campagnes de tests

Les premières évaluations des SRI sont proposées par (Cleverdon 1967) définissant notamment le contenu d'une *collection test* : "une collection de documents, un ensemble de requêtes (25 au minimum) et un ensemble de documents jugés pertinents pour chaque requête" (Kefi-Khelif 2006). Plusieurs campagnes d'évaluations des SRI sont proposées (Schütze et al. 2008) afin de fournir un protocole et un cadre uniforme pour mesurer leurs performances. La campagne d'évaluation TREC[5] créée en 1992 est une référence dont les objectifs sont notamment d'encourager la recherche dans le domaine et rendre disponibles les techniques d'évaluation mais également de réaliser une passerelle entre les différents acteurs (recherche, gouvernement et industrie). Une variété d'exercice a été considérée comme la recherche d'information par question/réponse (E. M. Voorhees et al. 1999), la recherche dans une collection de vidéo (Darwish et al. 2001) ou dans les génomes (Hersh et al. 2003). Encore récemment, de nombreux travaux mettent en places des évaluations TREC comme la recherche de podcast (R. Jones et al. 2021) ou encore la recherche de documents spécifiques comme ceux liés au COVID-19 (E. Voorhees et al. 2021). La campagne Amaryllis[6] spécialisée pour les SRI utilisant le français (Chevallet et al. 2000) peut également être citée. À notre connaissance aujourd'hui, aucune campagne d'évaluation et donc *collection test* n'a été réalisée dans le cadre de l'information hétérogène de l'industrie manufacturière.

2.2.6 Distinction entre extraction d'information et recherche d'information

Définition

L'Extraction d'Information et la Recherche d'Information sont complémentaires mais ne sont pas à confondre. La recherche d'information identifie les documents pertinents à partir d'une collection, le résultat est donc une liste de documents. L'extraction d'information quant à elle vise à identifier dans un texte l'information pertinente recherchée, le résultat ici est alors un extrait du contenu des documents. L'extraction d'information est une composante de la fouille de textes (text mining en anglais) et utilise notamment les outils du traitement du langage naturel (TALN). L'une des définitions de l'extraction d'information est qu'il s'agit d'extraire automatiquement des types d'informations prédéfinis à partir de textes courts en langage naturel (Gaizauskas et al. 1998). L'objectif est donc de récupérer à partir d'un texte en langage naturel les éléments souhaités (faits sur des types d'évènements, d'entités ou de relation prédéterminés) que l'on intègre ensuite comme valeur de champs dans un formulaire. L'utilisation de formulaires, comme l'illustre la Figure 2.6, est une méthode employée dans le domaine afin d'évaluer les systèmes d'ex-

[5]https ://www.nist.gov/programs-projects/text-retrieval-conference-trec

[6]Corpus d'Amaryllis disponible sur http ://catalog.elra.info/en-us/repository/browse/ELRA-W0029/ accès le 28/07/2021

traction d'information, notamment lors des campagnes MUC (Message Understanding Conference) qui se sont déroulées de 1987 à 1998 (Grishman et al. 1996). Le projet ACE (The Automatic Content Extraction Program) lui succède (Doddington et al. 2004). Pour l'évaluation francophone, ESTER[7] (Campagne d'Evaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques) est une campagne permettant notamment d'extraire de l'information à partir de retranscription radio (Galliano et al. 2005).

TEXTE :

« San Slavador, 19 avril 1989 (ACAN-EFE) – [texte] Le président du San Salvador, Alfredo Cristani a condamné l'attentat d'origine terroriste du ministre de la justice Roberto Garcia Alavo et a accusé du meurtre le Front de Libération National Farabundo Marti, (...) »

FORMULAIRE :

Date de l'incident	19 avril 1989
Lieu de l'incident	El Salvador : San Salvador (City)
Auteur	Front de Libération National Farabundo Marti
Victime	Roberto Garcia Alavo

Fig. 2.6 : Exemple de texte à analyser et de formulaire à remplir extrait de MUC-4 (Poibeau 2003)

Tâches de l'extraction d'information

Un système d'extraction d'information se constitue d'un nombre important de composants (Piskorski et al. 2013). Nous n'en détaillons qu'une partie ici, car l'objectif est d'appréhender les techniques permettant d'identifier des extraits de texte porteurs d'une information recherchée comme l'identification d'une exigence ou d'une phrase de justification.

La segmentation (ou tokenisation) fait partie des prétraitements nécessaires à la tâche d'extraction d'information. L'étiquetage morphosyntaxique (Part-Of-Speech tagging) permet d'associer aux termes du texte les informations grammaticales correspondantes (le terme est un verbe, un adjectif, etc.). Les systèmes d'étiquetage s'appuient sur le contexte du mot dans la phrase et sur des connaissances lexicales supplémentaires.

La reconnaissance des entités nommées permet d'identifier des unités lexicales et de les étiqueter selon des catégories comme 'nom de personne', 'organisation', 'emplacement géographique', 'notions de temps', 'monnaie' et 'pourcentage' (Sang et al. 2003).

[7]http ://www.afcp-parole.org/doc/camp$_e$*val*$_s$*ystemes*$_t$*ranscription/index.htmlcontact*

La coréférence en linguistique représente le phénomène où des syntagmes nominaux différents se réfèrent à la même entité (Adger 2003). On distingue en résolution de coréférence plusieurs cas : l'utilisation du nom et de son pronom, la relation entre pronom personnel et réfléchi, l'utilisation de périphrase et la relation entre un nom propre et son sigle.

L'extraction de relation consiste à détecter et à classer les relations définies entre les entités du texte. Plusieurs cas de relation existent comme celle entre une personne et une organisation (exemple : "Mme X travaille pour le Lispen", entre une personne et un lieu (exemple : "Mme X travaille à Aix-en-Provence) et entre deux organisations (exemple : "le Lispen, laboratoire de l'ENSAM").

L'extraction d'évènement a pour objectif d'identifier qui a fait quoi, quand, où, comment et pourquoi. Cette activité nécessite d'avoir réalisé les tâches précédemment décrites. Afin d'identifier un ensemble d'évènements du texte, une approche linguistique tente de déterminer les termes à considérer comme potentiel déclencheur d'évènement (dits "noms d'évènement").

Complémentarité de la recherche d'information et de l'extraction d'information

La recherche d'information vue dans la précédente partie et l'extraction d'information sont complémentaires. En effet, un système peut tout aussi bien effectuer une identification des documents puis extraire l'information de ce document avec les méthodes d'extraction d'information (et ainsi permettre des recherches plus précises) ou encore utiliser l'extraction d'information afin de constituer une sous-collection adaptée à la recherche d'information ou encore fournir les résultats d'une recherche d'information filtrée ou pondérer par des critères d'extraction d'information (Even 2005). Le premier cas est notamment utilisé dans certains systèmes de question-réponse comme QALC (the Question-Answering system of LIMSI-CNRS) (Ferret et al. 2000) qui suite à l'analyse de la question en langage naturel (reconnaissance de la catégorie concernée) vise dans un premier temps à identifier un sous-ensemble de document puis applique l'extraction d'information pour obtenir la réponse.

2.2.7 Synthèse

La Recherche d'Information est le domaine conventionnel permettant de retrouver des informations à partir de requête. Le Tableau 2.1 synthétise les différentes méthodes vues dans cette partie de l'état de l'art avec leurs principaux avantages et inconvénients.

Tab. 2.1 : Comparaison des méthodes vues dans l'état de l'art : Recherche d'Information

Fonction	Méthode/Outil	Avantage	Inconvénient
Indexation	TF-IDF	Simplicité de calcul	Non prise en compte de la longueur, de la probabilité
	Autres (TF-IDF étendu, BM25)	Prise en compte de la longueur, probabilité	Complexité de calcul
Expression du besoin	Thématique	Guidé	Nécessité de fouille
	Requête	Simplicité d'expression	Transformation nécessaire
	Naturel	Pas de transformation	Complexité d'intégration
	Liste	Simplicité	-
	Valeur	Précision	Complexité d'intégration
Extension de requête	Modèle de connaissance	A disposition dès l'initialisation	Peut être inadapté au besoin
	Modèle de langue	A disposition dès l'initialisation	Peut être inadapté au besoin
	Retour de pertinence à l'aveugle	Adapté à l'utilisation	Besoin d'initialisation
	Hybride (Modèle de connaissance et retour de pertinence)	Augmente la précision	-
Appariement	Modèle booléen simple	Simplicité de calcul	Pas d'ordonnancement des résultats
	Autres modèles booléens (étendu et flou)	Ordonnancement des résultats	Coût de calcul
	Modèle algébrique	Ordonnancement des résultats et liens entre termes	Coût de calcul
	Modèle Probabiliste	Prise en compte de la structure des documents	Coût de calcul

Fonction	Méthode/Outil	Avantage	Inconvénient
Evaluation	Rappel	Usuel	Pas d'ordonnancement des résultats
	Précision	Usuel	Pas d'ordonnancement des résultats
	F-Mesure	Usuel et pondère le rappel sur la précision	Pas d'ordonnancement des résultats
	Autres (Courbe Rappel/Précision, précision moyenne)	Prise en compte de l'ordonnancement des résultats	-
Extraction d'information	RI puis EI	Précision de réponse	-
	EI puis RI	Filtrage de collection	-
	Résultats RI pondéré par EI	Pondération des résultats	-

2.3 La représentation graphe

2.3.1 Théorie des graphes

Définition, vocabulaires et propriétés

La théorie des graphes est un domaine de recherche mathématique avec une forte application informatique. D'après Tommy R. Jensen (Sammut et al. 2017), le succès de la théorie des graphes est dû à la facilité de compréhension des idées et preuves pouvant être communiquées de manière imagée contrairement à l'utilisation de symboliques complexes.

Le graphe nommé G est défini par le couple suivant :

$$G = (V, E)$$

où V (pour 'vertices') est l'ensemble des sommets ou noeuds et E (pour 'edges') est l'ensemble des arêtes ou liens connectant un sous-ensemble de deux sommets notés u et v de V. Ainsi on peut définir l'ensemble E comme :

$$E \subseteq \{\{u, v\} | (u, v) \in V^2 \wedge u \neq v\}$$

Le graphe tel que décrit par les formules précédentes se nomme plus précisément **graphe simple non orienté**. Représenté en (1) de la Figure 2.7, il se distingue des graphes listés ci-après et illustrés à la suite dans la même figure. Le **graphe simple orienté** (2) est composé d'arêtes orientées formant ainsi des couples (paires ordonnées) de sommets. Le **multigraphe orienté ou non** (3) accepte plusieurs arrêtes entre la même paire de sommets et possiblement des arêtes liant un sommet à lui-même. L'**hypergraphe** quant à lui comporte des arêtes liant un nombre quelconque de sommets (4).

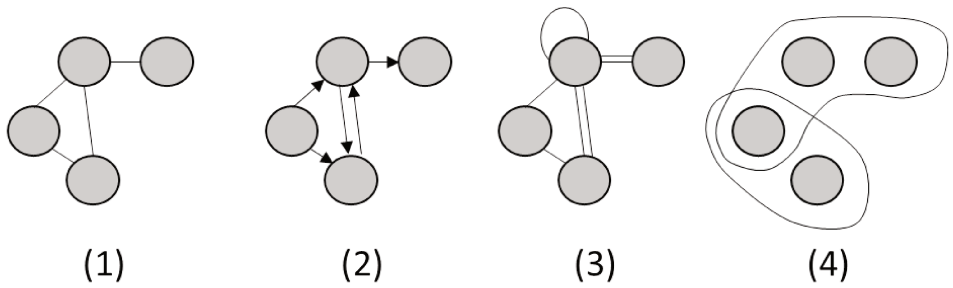


Fig. 2.7 : Illustration d'un graphe non orienté ① et orienté ②, d'un multigraphe ③ et d'un hypergraphe ④

Deux sommets d'un graphe sont dits adjacents s'il existe une arête qui les relie. On dit également qu'ils sont voisins. Le degré d'un sommet est le nombre d'arêtes auxquels il est relié (on parle d'arêtes incidentes). Deux arêtes sont adjacentes si elles ont un sommet en commun. Un chemin est une suite d'arêtes permettant de lier deux sommets non voisins. La longueur du chemin est le nombre d'arêtes permettant de lier les deux sommets en question. Un graphe est connexe si pour tout u et v, le graphe contient au moins un chemin entre u et v. Un cycle est un chemin reliant le même sommet en passant par plus d'une arête.

Étiquetage du graphe : par défaut, un graphe est étiqueté à ses sommets car c'est cette propriété qui permet de les distinguer les uns des autres. Le graphe peut également être étiqueté à ses arêtes.

Pondération du graphe : un graphe pondéré est un graphe dont les arrêtes portent une valeur représentative d'un poids.

Isomorphisme des graphes : considérant deux graphes $G = (V, E)$ et $G' = (V', E')$, l'isomorphisme est une bijection $f : V \to V'$ entre G et G' qui préserve l'adjacence, de telle sorte que $\{u, v\} \in E$ si et seulement si $\{f(u), f(v)\} \in E'$.

Cas du graphe biparti : un graphe biparti est un graphe simple pouvant être partitionné en deux sous-ensembles de sommets U et V et dont chaque arête est adjacente dans une de ses extrémités dans U et dans son autre extrémité dans V.

Cas du graphe complet : un graphe complet est un graphe simple dont tous les sommets sont voisins.

Représentation matricielle des graphes : l'utilisation de la représentation matricielle des graphes est utilisée afin de les stocker et y réaliser des calculs à moindre coût.

De nombreux algorithmes de fouille de graphes

Il existe de nombreux algorithmes s'appuyant sur la théorie des graphes (Arumugam et al. 2016). En reprenant la sous-décomposition de (Aggarwal et al. 2010), ouvrage introduisant aux différents algorithmes de fouille de graphe, nous distinguons ceux de :

- recherche de parcours : peut chercher un chemin commun à plusieurs graphes ou identifier un chemin unique dans un seul graphe,
- regroupement : peuvent soit chercher à regrouper un ensemble de sommets ou un ensemble de graphes avec des similarités,
- classification : cherchent par exemple à étiqueter des sommets (à partir d'une liste préexistante de sommets déjà étiquetés) ou à étiqueter des graphes (même logique qu'avec les sommets),
- analyse temporelle des graphes : peuvent notamment chercher à détecter la densification ou le rétrécissement d'un graphe dans le temps.

2.3.2 Base de données graphe

Les bases de données orientées graphe reprennent les notions de la théorie des graphes pour structurer et gérer les données. Ce type de base de données fait parti du spectre plus

large des bases de données NoSQL, dont la terminologie a émergé au début des années 2000 et été popularisée par les GAFA[8] (Leavitt 2010). Ces bases de données NoSQL ont pris de l'importance notamment dans le contexte big data pour répondre à la problématique de stockage et de gestion de grandes quantités de données (Moniruzzaman et al. 2013). Nous présentons dans un premier temps les principes clés des bases de données NoSQL, puis après avoir présenté succinctement chaque types de base de données nous détaillons celui de l'orienté graphe.

Base de données NoSQL

La notion de NoSQL provient de la notion de 'non relationnel'. Les bases de données NoSQL s'opposent donc aux bases de données relationnelles devenues trop restrictives et lentes pour le requêtage des éléments (Nayak et al. 2013). Contrairement aux bases de données relationnelles classiques, les bases de données NoSQL éliminent une grande partie des contraintes du modèle de données. Elles n'imposent par exemple pas à un enregistrement d'avoir l'ensemble des attributs de la même table. Détaillé dans (Meier et al. 2019), on distingue quatre types de bases de données :

- **la base de données orientée clé-valeur** dont l'avantage est sa simplicité de modélisation et sa capacité à être gérée dans des systèmes distribués mais dont la possibilité d'ajout de métadonnée est restreinte,
- **la base de données orientée document** qui permet, contrairement à la base de données clé-valeur, d'associer à l'enregistrement un document semi-structuré identifiable,
- **la base de données orientée colonne** qui est devenue particulièrement utilisée grâce à la puissance et la flexibilité d'analyse de large éventail de données (Abadi 2008),
- **la base de données orientée graphe** généralement utilisée pour modéliser des réseaux et explorer les relations entre les entités (Angles et al. 2008).

Avantages et limites de la base de données graphe

Pour souligner l'avantage de la base de données graphe, différentes recherches dont (Miller 2013) ont montré leurs puissances dans l'analyse des données de nature relationnelle. Comme le présente (Meier et al. 2019), ces bases de données graphe permettent de retranscrire l'information de lien comme dans les bases de données relationnelles mais surpassent ces derniers dans l'accomplissement de requête utilisant ces relations. Dans (Pokorný 2015), l'auteur rappelle le caractère de plus en plus commun des bases de données orientées graphes pour traiter les données connectées mais souligne également certaines de ses limites comme la mise à l'échelle (restant plus difficile que celles d'autres bases de données car la distribution sur plusieurs machines du graphe engendre des relations distribuées, problème nommé "point de coupure minimal") et certaines opérations d'extraction d'information encore coûteuses en termes de calcul.

[8]GAFA pour Google, Apple, Facebook, Amazone

Principales caractéristiques des solutions du marché

L'auteur de (Pokorný 2015) distingue les solutions générales de base de données orientée graphe comme la solution Neo4J[9], InfiniteGraph[10], Sparksee[11], Titan[12], GraphBase[13] et Trinity[14] aux solutions employant de multi modèles combinant le parcours de relations par graphe à une gestion des données orientées documents (cas d'OrientDB[15]) ou orientées clé-valeur (cas d'HyperGraphDB[16]).

Dans (Fernandes et al. 2018), les auteurs quant à eux incluent dans la comparaison des différentes solutions du marché un critère spécifique sur cette capacité à intégrer du multi modèles. L'étude prend en compte les bases de données orientées graphe les plus populaires, à savoir AllegroGraph[17], ArangoDB[18], InfiniteGraph, Neo4J et OrientDB, selon les 8 fonctionnalités suivantes :

- **la flexibilité du schéma** - ne pas avoir à modifier l'ensemble du schéma de donnée pour en intégrer de nouveaux types de données,
- **la puissance du langage de requête** - ce qu'il permet dans l'interrogation du graphe,
- **les capacités de distributivité[19] de la base** - permettant de diviser et distribuer la base sur un certain nombre de machines,
- **capacité de sauvegarde** - permettant la planification et la restauration de la base,
- **l'intégration du multi modèle** - permettant l'exploitation des données également sous d'autre types de modèles comme document et clé valeur,
- **la multi architecture** - capacité d'intégration de spécificité structurelle de la base,
- **l'évolutivité[20]** - capacité de changement d'échelle en nombre de transactions possibles à la seconde et l'accès cloud (gestion de l'évolutivité accès à une base externe à l'entreprise).

En synthèse, la comparaison des différentes solutions positionne ArrangoDB et Neo4J comme leader. Concernant la flexibilité du schéma et la puissance du langage de requête, Neo4J est jugé meilleur que les autres.

[9] https://neo4j.com/
[10] https://objectivity.com/infinitegraph/
[11] https://www.sparsity-technologies.com/
[12] https://titan.thinkaurelius.com/
[13] https://graphbase.ai/
[14] https://www.microsoft.com/en-us/research/project/trinity/
[15] https://www.orientdb.org/
[16] http://www.hypergraphdb.org/
[17] https://allegrograph.com/products/allegrograph/
[18] https://www.arangodb.com/
[19] en anglais : sharding
[20] en anglais : scalability

Neo4J

Le système de gestion de base de données Neo4J intègre : (1) le stockage en graphe des données, (2) un langage de requête nommé CYPHER permettant de formuler une recherche sur le graphe, (3) une méthode d'indexation native ou importée, (4) la capacité d'intégrer des plug-ins pour enrichir la complexité d'opération réalisée dans le graphe.

1. Le modèle de données utilisé dans Neo4J est illustré en figure 2.8. Il est constitué de noeuds et de relations entre noeuds . Chaque nœud et relation portent une ou plusieurs étiquettes et des propriétés de type clés-valeurs. Les propriétés sont non contraintes par un schéma, chaque nœud peut donc avoir des propriétés différentes bien que de même étiquette.

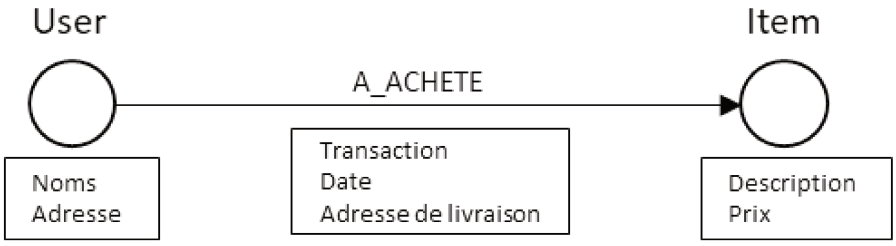


Fig. 2.8 : Illustration d'un modèle de données graphe dans Neo4J (Amine Lies Benhenni 2016)

2. Le langage de requête CYPHER est propre à Neo4J et permet d'interroger et de manipuler le graphe. Les principales fonctions listées dans le Tableau 2.2 permettent de créer, modifier, supprimer et rechercher des éléments. L'utilisation du langage de requête peut se réaliser en ligne de commande via un exécuteur ou grâce à une interface utilisateur embarquée. Suite à une requête, le résultat est fourni en liste, en tableau ou en graphe selon les éléments de résultat. Le nombre total d'éléments rapporté en réponse et le temps d'exécution de la requête sont également indiqués.

Tab. 2.2 : Liste des principales fonctions de requête CYPHER

Nom de la fonction	Opération appelée
START	Indique un nœud de départ
MATCH	Indique la recherche d'éléments du graphe
WHERE	Applique un filtre aux éléments recherchés
RETURN	Indique le type d'information attendu en résultat
CREATE, DELETE et SET	Créé, supprime ou met à jour un élément du graphe
FOREACH	Itère une opération de la requête
UNION ALL	Combine plusieurs requêtes

3. L'indexation dans Neo4J se réalise également avec le langage CYPHER. L'indexation ici permet de gagner du temps afin de ne pas parcourir l'ensemble du graphe pour recherche des nœuds mais également pour ajouter des contraintes d'unicité

ou de propriété sur les données appartenant à un index. La création d'un index se réalise avec la fonction CREATE INDEX ON et permet de construire la méthode d'indexation souhaitée.

4. Neo4J permet d'intégrer des plug-ins (ou procédures) afin de personnaliser les requêtes et intégrer une plus grande complexité d'opérations et de fonctions. Le lancement de la procédure peut se réaliser avec la fonction CALL…YIELD.

2.3.3 Représentation des réseaux d'informations hétérogènes

Définitions

Plusieurs termes peuvent être employés pour nommer des réseaux d'informations. On distingue notamment la notion de 'big networks' et de 'réseau d'informations hétérogènes'. Le premier terme est largement utilisé dans la littérature, les auteurs dans (Bedru et al. 2020) le présente comme désignant les réseaux à la fois large et complexe dans leurs structures internes. Les auteurs de (Shi et al. 2016) présentent quant à eux la notion de 'réseau d'information hétérogène' (en anglais : Heterogeneous Information Network (HIN)), soulignant la prise en compte du caractère multi typé des données du réseau. En effet, ce sont de plus en plus de recherches qui considèrent, à travers cette notion, la richesse de structure et de sémantique des données interconnectées et hétérogènes des réseaux d'informations, contrairement aux travaux les considérant encore comme des réseaux homogènes. Une définition précise de ce qu'est un réseau d'information y est rappelée et la distinction entre un réseau d'information homogène et hétérogène est également donné :

- **Réseau d'information par (Sun et al. 2013)** - Un réseau d'information est défini par un graphe dirigé $G = (V, E)$ avec une fonction de correspondance de type d'objet $\tau : V \to A$ et une fonction de correspondance de type de liens $\phi : E \to R$, où chaque objet $v \in V$ appartient à un type d'objet particulier $\tau(v) \in A$ et chaque lien $e \in E$ appartient à une relation particulière $\phi(e) \in R$. Si deux liens appartiennent au même type de relation, les deux liens partagent le même type d'objet de départ ainsi que le type d'objet d'arrivée.
- **Réseau d'information hétérogène/homogène par (Shi et al. 2016)** - Le réseau d'information est appelé 'réseau d'information hétérogène' si le nombre de type d'objet $|A| > 1$ ou nombre de type de relations $|R| > 1$, sinon, le réseau est appelé 'réseau d'information homogène'.

L'analyse des réseaux

Les auteurs dans (Bedru et al. 2020) présentent l'analyse des réseaux selon plusieurs niveaux structurels : mniveau micro (au niveau des sommets et des arêtes, par l'étude par exemple de l'interaction être deux individus), niveau méso (au niveau d'un regroupement de sommets et arêtes distinguant par exemple deux groupes aux motifs différents ou complémentaires) et niveau macro (au niveau du réseau global comme l'analyse de densité du réseau). Ils distinguent également l'analyse des réseaux selon la considération ou non de

l'évolution du réseau dans le temps. Tandis que (Shi et al. 2016) distinguent les exercices d'analyse des réseaux par l'analyse : de similarité, de regroupement, de classification, de prédiction de liens, de classement, de recommandations et de fusion d'information, les auteurs de (Bedru et al. 2020) les regroupent en trois grandes familles que sont la détection de communauté, la prédiction de liens et la recommandation. Les auteurs précisent au préalable les principales techniques employées : le classement (de sommets, de motif ou de communauté), le partitionnement de graphe ou encore la représentation vectorielle de graphe.

- **La détection de communauté** cherche à détecter des collections de sommets fortement liés entre eux et faiblement liés aux autres sommets du graphe. Plusieurs méthodes permettent de réaliser cette détection de communauté comme la méthode de propagation d'étiquette, de Louvain (De Meo et al. 2011) ou de marche aléatoire (Lai et al. 2010).

- **La prédiction de liens** estime la présence d'une arête entre deux sommets. C'est un exercice particulièrement utilisé dans l'analyse de réseaux dynamiques comme les réseaux sociaux où des liens peuvent se créer ou se défaire. Elle peut se réaliser par mesure de similarité, par factorisation de matrice ou modèle graphe probabiliste. Les techniques de prédictions de liens sont notamment effectuées par l'analyse structurelle du graphe.

- **La recommandation** prédit l'information particulièrement intéressante à restituer à l'utilisateur par l'étude des préférences antérieures pouvant être complété par des attributs spatiaux temporels (Christoforidis et al. 2018). L'objectif est de filtrer dans l'amas d'information pour tendre vers celle dont l'utilisateur a besoin. La recommandation peut alors prendre la forme de suggestion par rapport au contenu de l'information, par rapport à l'expérience des utilisateurs précédents ou une combinaison de ces deux méthodes.

Domaines d'applications

Comme le rappelle (Pokornỳ 2015), le choix d'utiliser la modélisation graphe est de plus en plus commun pour représenter et traiter des données interconnectées. Il y a autant d'application possible que de réseau d'information existant. Nous nous concentrons ici uniquement sur les travaux dont le graphe représente les données sources (et non à la représentation de connaissance vue dans la prochaine section). Nous pouvons citer, le domaine de la cybersécurité qui suscite de nombreux sujets de recherche (Dawood 2014), notamment avec la perspective de nouvelles méthodes d'apprentissage automatique dédié au graphe (Bowman et al. 2021) ou avec l'application des travaux de (Noel et al. 2016) qui propose CyGraph, un système permettant d'améliorer la sécurité des réseaux adoptant un modèle unifié en graphe des données hétérogènes dont les informations d'infrastructures et d'évènements d'attaques. Les données gouvernementales sont également modélisables en graphe pour y exploiter les informations en réseau. On permet ainsi la détection des fraudes (Erven et al. 2017) ou l'anticipation des besoins de maintenance d'infrastructure en mêlant dans un unique graphe de multiples données provenant de

sources différentes et ouvertes (Lina 2020). Des cas d'applications également dans l'analyse des groupes de collaboration politiques sont étudiés (Scott 2016). Dans le domaine du journalisme également, les travaux de (Balalau et al. 2020) propose une approche pour intégrer les sources de données ouvertes et hétérogènes (structurées, semi-structurées et non structurées) en un modèle graphe. Le domaine de la biologie utilise également la modélisation graphe afin de comprendre les relations complexes entre les données biologiques hétérogènes (Touré et al. 2016), en intégrant notamment des bases de données de protéines ou de génomes (Yoon et al. 2017).

Spécifique au contexte de l'industrie manufacturière et plus particulièrement aux données d'ingénierie, les auteurs de (Mordinyi et al. 2015) propose une approche graphe pour stocker et versionner les données, elles-mêmes liées à l'aide d'une ontologie pour détecter les concepts communs. Les auteurs de (Schabus et al. 2017) se concentrent quant à eux sur la représentation d'un environnement de fabrication de semi-conducteurs incluant la typologie particulière qu'est la représentation spatiale des différents éléments. Dans ces deux cas, les sources d'informations sont limitées et connues, dimensionnant l'intégration des données en graphe.

Enfin, de manière transversale aux domaines d'application, les auteurs de (Martínez-Bazan et al. 2007) proposent d'intégrer les données hétérogènes de manière générique à travers leur solution nommée 'DEX' appelée désormais commercialement 'Sparksee'. La solution permet ensuite d'y effectuer de la recherche mais également de l'analyse dans le réseau d'information ainsi modélisé. Néanmoins, l'ensemble du contenu textuel des données sources n'est pas exploité.

Enjeux

La liste suivante des enjeux autour de la gestion et l'analyse des grands réseaux d'informations hétérogènes peut être considérée :

- **considérer plus de complexité comme le bruit et la temporalité**. En effet, les auteurs dans (Shi et al. 2016) expliquent que même si elle provient de données structurées comme de base de données relationnelle, l'information est composée de bruit. Il y a des opérations de consolidation à réaliser comme la mise en relation d'éléments traitant de la même entité du monde réel ou la détection de relation incomplète dans les systèmes sources. Dans le cas des données non structurées, le défi est de considérer la sémantique des contenus, notamment par l'exploitation du langage naturel pour le contenu textuel. De plus, dans (Bedru et al. 2020), les auteurs énoncent également l'enjeu d'une prise en compte de la nature dynamique des réseaux plus importante.
- **augmenter les performances de fouille**. Les auteurs de (Shi et al. 2016) soulignent l'importance d'adapter l'organisation de la structure du graphe et d'exploiter la captation sémantique obtenue par les liens ('meta-path') du graphe afin d'améliorer la performance d'analyse. Les auteurs de (Bedru et al. 2020) suggèrent notamment de privilégier les sous-graphes aux éléments élémentaires.
- **augmenter les performances de calculs**. Les auteurs de (Shi et al. 2016) et

(Bedru et al. 2020) se rejoignent afin de souligner la nécessité de considérer de plus grands réseaux dans l'application et la validation des méthodes de représentation et d'analyse des réseaux d'informations. En effet, la plupart des taches sont évaluées sur de petits jeux de données et ne réussissent pas à atteindre la rapidité de calcul nécessaire sur des cas réels (Shi et al. 2016).

2.3.4 Représentation des connaissances

Graphe de connaissances

Le graphe permet également de représenter un réseau de connaissances. C'est le cas notamment du graphe de connaissances introduit par Google en 2012 (Singhal 2012) 'Google Knowledge Graph'. Depuis, de nombreuses grandes sociétés ont créé leurs propres graphes de connaissances normalisant ainsi leurs utilisations (Mika et al. 2014). Les auteurs de (Noy et al. 2019) présentent notamment le graphe de connaissance de Microsoft, celui de Facebook, d'eBay ou encore d'IBM. Comme le souligne (Monnin 2020), plusieurs définitions de ce qu'est un graphe de connaissances sont données dans la littérature. Certaines définitions sont larges comme celle de (Paulheim 2017) signalant que n'importe quel graphe représentant des connaissances peut être considéré comme un graphe de connaissances tandis que d'autres sont plus restrictives suggérant que les graphes de connaissances ont le but de produire de nouvelles connaissances (Ehrlinger et al. 2016). C'est notamment cette notion qui distinguera une base de connaissances à celui d'un graphe de connaissances. Les graphes de connaissances sont utilisés dans de nombreuses applications (Zou 2020), notamment pour enrichir les capacités de recommandations (Jiangzhou Liu et al. 2021) et étendre les requêtes (Raza et al. 2019). Dans ce second cas, l'utilisation du graphe de connaissances multilingue ConceptNet[21] (Speer et al. 2017), dont l'interface de navigation est présenté dans la Figure 2.9, a été évaluée plus performante par (Balaneshinkordan et al. 2016) qu'avec les autres sources externes DBpedia[22] et Freebase (transféré depuis dans WikiData[23]) ou encore par l'utilisation de ressource par analyse du corpus. La comparaison a été réalisée sur trois jeux de données de de la campagne TREC.

[21] https://conceptnet.io/
[22] https://www.dbpedia.org/
[23] https://www.wikidata.org/

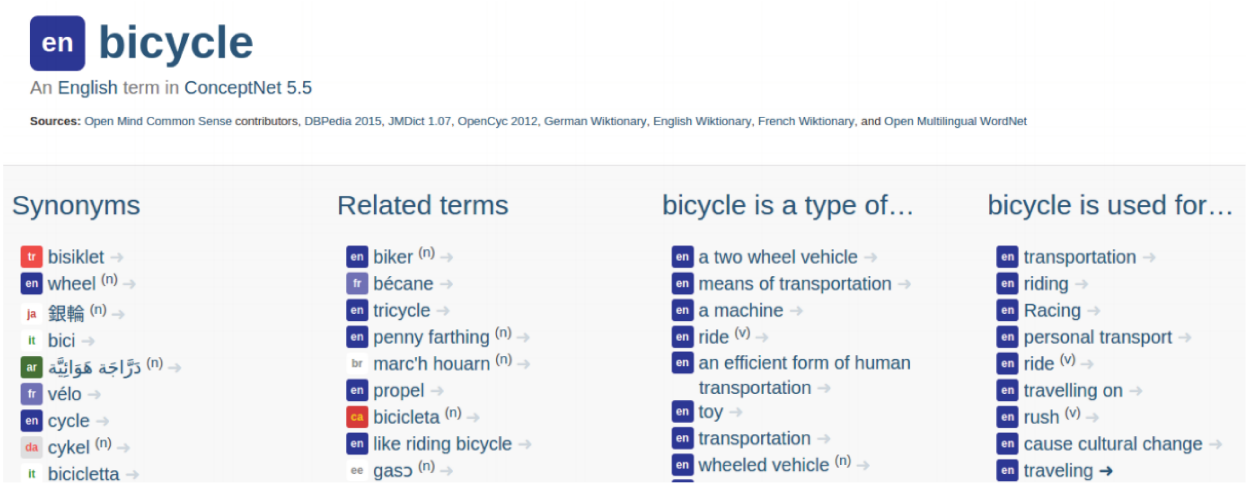


FIG. 2.9 : Interface de navigation de ConceptNet sur l'exemple du mot 'bicyle' (SPEER et al. 2017)

Ontologie

Une des premières introductions à l'ontologie la défini comme 'la spécification explicite d'une conceptualisation' (GRUBER 1993), mais comme le présente l'auteur de (GAYE 2017), elle a désormais de nombreuses définitions s'adaptant aux contextes d'utilisation. La définition "représentation spécifique et formelle d'une conceptualisation partagée d'un domaine" semble néanmoins la plus utilisée dans la littérature (STUDER et al. 1998). De manière générale, l'ontologie est une représentation formelle d'un domaine spécifique par ses concepts et les relations entre ces concepts. Il existe plusieurs autres niveaux de conceptualisation rappelés dans (GAYE 2017) comme l'ontologie de haut niveau reprenant des concepts plus généraux permettant de lier les ontologies de domaine ainsi que le niveau plus bas d'applications offrant des spécificités dédiées aux champs d'application et donc non réutilisables par ailleurs. De nombreux domaines variés ont publié des ontologies comme en génétique (CONSORTIUM 2004), en géopolitique (S. KIM et al. 2013), en finance (BENNETT 2013) ou transversaux aux domaines comme BabelNet (NAVIGLI et al. 2010). Les applications d'ontologies étudiées par la littérature scientifique sont tout autant diversifiées comme les systèmes de recommandations (TARUS et al. 2018), l'extension de requête (BHOGAL et al. 2007), l'intéropérabilité dans la gestion du cycle de vie du produit (FRAGA et al. 2020) ou plus généralement au soutien à l'autonomie des robots (OLIVARES-ALARCOS et al. 2019).

Standard du web sémantique

Le web sémantique s'est fait connaître du grand public en 2001 (BERNERS-LEE et al. 2001). Son objectif est de représenter de manière standardisée les informations du web afin de les rendre exploitables par les machines. Ainsi, il fournit entre autres un cadre de représentation des connaissances du web à travers un ensemble de standards préconisé par le w3c[24] (ABITEBOUL et al. 2011). L'URIs (Uniform Resource Identifiers) permet

[24] https://www.w3.org/

d'identifier les ressources par un identifiant unique, l'XML permet de décrire le contenu des ressources, le RDF (Resource Description Framework) permet de décrire les relations entre ressources selon un triplet 'sujet, prédicat, objet' (le sujet étant lié à un objet selon une relation identifiée par le prédicat). Utilisant la syntaxe RDF, le RDFS (Resource Description Framework Schema) est le langage de description des vocabulaires associé à RDF et OWL celui des ontologies. Enfin, SPARQL est le protocole de requête pour les données RDF.

Quelques enjeux

Tandis que les revues des enjeux liés aux ontologies sont concentrées sur leurs constructions (Al-Arfaj et al. 2015) ou leurs mises en correspondance (Ramar et al. 2016), concernant les graphes de connaissances, les auteurs dans (Noy et al. 2019) listent les différents enjeux rencontrés par les industries dans leurs créations et leurs utilisations : la désambiguïsation des entités, la gestion du multi typage de l'entité, la gestion de l'évolution des connaissances, l'extraction des connaissances à partir de sources structurées et non structurées ainsi que la gestion de la performance lors de la mise à l'échelle.

2.3.5 Synthèse

Basée sur la théorie des graphes, domaine de recherche permettant notamment d'utiliser de nombreux algorithmes de fouille de graphe, la base de données orientée graphe est appliquée pour le stockage et la gestion de données fortement relationnelles. En effet, elle permet de retranscrire les liens entre les données sans les contraintes de schéma des bases de données relationnelles. Bien que certains enjeux restent entiers, comme la consolidation de l'information et l'amélioration des performances de fouille et de calculs, de multiples domaines ont d'ores et déjà adoptés la modélisation graphe dans la représentation de leurs réseaux d'informations hétérogènes permettant notamment leurs analyses. Malgré les nombreuses études, nous notons néanmoins qu'aucune proposition de modélisation graphe de l'ensemble des données de l'industrie manufacturière n'est faite. Nous entendons par l'ensemble des données : les données structurées et non structurées provenant de tous les services d'entreprise générées pendant tous le cycle de vie du produit. Le Tableau 2.3 synthétise les différentes méthodes vues dans cette partie de l'état de l'art avec leurs principaux avantages et inconvénients.

Tab. 2.3 : Comparaison des méthodes vues dans l'état de l'art : représentation graphe

Fonction	Méthode/Outil	Avantage	Inconvénient
Type de base de données	Graphe	Exploitation des relations	Coût de calcul et distributivité
	Clé-valeur	Simple et distribuable	Ajout de métadonnée impossible , Recherche à travers les relations difficile
	Document	Simple avec documents	Recherche à travers les relations difficile
	Colonne	Rapidité et flexibilité	Recherche à travers les relations difficile
Base de données graphe	Neo4J	Meilleurs flexibilité de schéma et puissance de requête	Moins bonnes distributivité et capacité de multi-modèle
	ArrangoDB	Meilleurs distributivité et capacité de multi-modèle	Moins bonnes flexibilité de schéma et puissance de requête
	Autres (AllegroGraph, InfiniteGraph, OrientDB)	-	Moins bonnes flexibilité de schéma, puissance de requête, distributivité et capacité de multi-modèle
Représentation des connaissances	Graphe de connaissance	Représentation d'un réseau de connaissance au sens large	-
	Ontologie	Description des concepts et relations entre concepts d'un domaine	-
Graphe de connaissance pour extension de requête	ConceptNet	Meilleures précision et rappel	-
	Autres (DBpedia, Freebase ou par corpus)	-	Moins bonne précision et rappel

2.4 Recherche d'information en entreprise de l'industrie manufacturière

2.4.1 Recherche d'information en entreprise

Les systèmes de recherche d'information en entreprise ou 'enterprise search'

Les systèmes de recherche d'information d'entreprise, désigné en anglais par 'enterprise search' *"permettent à un public limité d'effectuer des recherches dans le contenu de plusieurs sources d'entreprise, telles que des intranets, des dépôts de documents et des bases de données, avec précision et rapidité*[25] *"* (Deolekar et al. 2018). Les auteurs de (Deolekar et al. 2018) rappellent notamment que le domaine n'a reçu jusqu'à présent que peu d'attention par la communauté de recherche du web, notamment en raison de problèmes difficiles à résoudre que nous listons dans la section 2.4.1.

Utilité et distinction avec la recherche web

La recherche d'information en entreprise sert aux scénarios de découvertes et à la résolution de problèmes (Russell-Rose et al. 2011). Le comportement de recherche est alors très différent du comportement général de recherche sur le web, utilisant des requêtes plus précises avec des termes spécialisés démontrant un besoin de fonctionnalités additionnelles (Freund et al. 2006). Les auteurs de (Deolekar et al. 2018) distinguent notamment cinq points majeurs de divergences :

- Le type de réponse : en entreprise, on recherche la réponse exacte à notre recherche et non pas la meilleure ou la plus populaire. Ce point engendre notamment un besoin de précision supérieure à la recherche sur le web,
- La création de contenu : le création de contenu est limitée aux besoins d'informations nécessaires à l'organisation de l'entreprise. Elle est ainsi contrôlée contrairement au web où la création de contenu est ouverte et réalisée par quiconque ayant le souhait de s'exprimer,
- Les techniques de recherche : sur le web, on utilise des techniques propres à son architecture comme dans l'évaluation de l'autorité d'un site par le nombre de fois où il est référencé dans d'autres sites. Ce sont des techniques non applicables à la structure des documents de l'entreprise (Hawking 2004),
- Un environnement complexe : une solution de recherche d'entreprise doit considérer de multiples organisations disposant de matériels, plateformes, sécurité, contenu et formats hétérogènes contrairement au web,
- L'adaptation et la flexibilité : une conséquence directe du précédent point entraîne une adaptation et une flexibilité plus difficiles à maintenir dans un environnement d'entreprise hétérogène que dans le web.

[25] https://www.aiim.org/What-is-Enterprise-Search#

Par contre, les besoins semblent similaires entre différents domaines d'activités comme la recherche juridique, le recrutement, la gestion de brevet et de santé (Russell-Rose et al. 2018) avec néanmoins quelques distinctions comme l'importance donnée à l'évaluation du rappel (capacité du système à fournir tous les résultats pertinents) ou de la précision (capacité du système à ne fournir que des résultats pertinents) selon le domaine.

Des défis et propositions hétérogènes

Les défis autour de la recherche d'information en entreprise sont nombreux et variés résultant d'études et d'enquêtes diverses (Mukherjee et al. 2004 ; Hawking 2004 ; Mirza 2008 ; Y. Li et al. 2014 ; Stocker et al. 2015).

Les auteurs dans (Mukherjee et al. 2004) énoncent comme défi la gestion de multiples référentiels hétérogènes et de multiples formats de données ainsi que les problèmes de sécurité, de conformité et de déploiement. Ils projettent dans le futur de la recherche d'entreprise certaines évolutions comme une plus grande participation des employés dans l'évaluation des documents, de meilleurs algorithmes évaluant la pertinence des résultats, l'utilisation des bases données des réseaux sociaux, une plus grande automatisation dans l'annotation des documents mais également des règles plus strictes dans la création du contenu. Les auteurs de (Gunadi et al. 2015) observent quant à eux les défis liés au caractère distribué des données de l'entreprise considérant ainsi les trois points que sont la sécurité, l'accessibilité et le regroupement des sources de données. Les auteurs de (Mukherjee et al. 2004) soulignent le défi de recherche sur les sites intranets d'entreprise sans les textes d'ancrage usuellement utilisés pour l'indexation des pages web. Les auteurs de (Dmitriev et al. 2006) et (Cortez et al. 2015) les rejoignent et proposent notamment l'utilisation d'annotation pour y remédier. Les auteurs dans (Freund et al. 2006) précisent quant à eux qu'il est nécessaire de fournir aux utilisateurs des moyens de requêtes complexes et d'utilisation de terminologies spécialisées encourageant la recherche dans du vocabulaire contrôlé ou des outils de traitement des requêtes. Ce point est notamment confirmé par l'étude menée dans (D. E. Jones et al. 2015). Ils précisent également qu'extraire des informations particulières comme les métadonnées (auteurs, versions ou dates par exemple) pourrait être un avantage à la recherche. L'ensemble de ces défis sont également repris par (Stocker et al. 2015) ajoutant toutefois la perception globale de l'utilité de la recherche d'information par les utilisateurs en entreprise. Enfin, les auteurs dans (Hawking 2004) ajoutent le défi de définir une collection de tests appropriée à la recherche d'entreprise souhaitée, considérant ainsi la complexité réelle de l'environnement de recherche. On note dans tous ces travaux l'absence de discussion autour des enjeux spécifique à la recherche d'information dans l'industrie manufacturière. Seul les travaux de (D. E. Jones et al. 2015) discute de l'environnement d'ingénierie mais se concentre sur la formulation des requêtes.

Concernant le type de solutions rencontrées pour contribuer à la recherche d'information en entreprise, les auteurs dans (Mirza 2008) distinguent six familles proposées dans la littérature : (i) la récupération des données et leurs indexations, (ii) le nettoyage préalable de la collection de documents pour une meilleure pertinence, (iii) l'utilisation d'une taxonomie et d'une classification des documents pour aider à la navigation, (iv) l'extraction et la fouille de texte dans les documents, (v) la recherche par un système

fédérateur gérant les contraintes de sécurité et enfin (vi) l'utilisation d'information de pertinence apportée par l'utilisateur.

2.4.2 L'approche graphe dans la recherche d'information d'entreprise

L'état de l'art a été réalisé en recherchant l'ensemble des travaux utilisant une approche graphe à des fins de recherche d'information en entreprise.

Représentation sémantique et graphe de connaissances au service de la RI

L'utilisation des graphes pour la recherche de connaissance utilisant notamment une interprétation sémantique des requêtes et des données n'est pas un nouveau concept comme le démontre la proposition dans (Kamel et al. 1990). L'utilisation des graphes permet notamment de sous-décomposer le texte des documents et ainsi améliorer la recherche d'information (Jin et al. 2007). Dans le cadre de la recherche d'information d'entreprise les auteurs dans (Modoni et al. 2014) proposent une approche par modèle de données commun structuré par l'extraction des connaissances. Cette extraction est supportée par un référentiel sémantique de type ontologie. Un des avantages mis en avant de la démarche est notamment d'exposer les interactions implicites et explicites entre les entités et pouvoir par la suite les exploiter dans les requêtes. L'utilisation d'ontologie au service de la recherche d'information est également présentée dans (Koopman et al. 2012) l'utilisant pour pondérer les concepts médicaux des documents en texte libre ou encore dans (Brauer et al. 2010) permettant d'identifier les concepts clés de la requête et des documents afin de faire disparaître l'ambiguïté possible des termes de la requête.

L'utilisation des graphes de connaissances semble également une évidence pour la recherche d'information. Les auteurs de (Reinanda et al. 2020) listent les utilisations possibles suivantes : (i) identification des documents afin d'améliorer le classement des résultats, (ii) identification d'expression linguistique de référence (entité nommée) dans les documents, (iii) recommandation d'entité nommée selon l'expression de la recherche et du contexte et (iv) explication de la relation entre les différentes entités nommées. Il existe donc de nombreuses applications de graphe de connaissances pour le soutien à la RI, notamment celle soulignée en Section 2.3.4 où il est utilisé pour l'extension des requêtes.

Représentation graphe des données au service de la RI

Tandis que les solutions commerciales de recherche d'information d'entreprise sont multiples comme l'atteste la récente présentation des éditeurs de technologie de recherche cognitive réalisée par Forrester (Gualtieri et al. 2021), d'après les connaissances académiques accessibles, seule la solution 'Sparksee', anciennement appelé DEX (cf. Section 2.3.3) propose une modélisation graphe des données hétérogènes de multiples sources d'informations permettant d'y faire de la recherche d'information et de l'analyse. La prise en compte du contenu textuel des documents non structurés n'est néanmoins pas considéré dans l'agrégation de données et aucune formulation générique de requête n'est également

proposé. Ces travaux ont toutefois démontré la capacité d'exploiter un tel graphe. Les auteurs dans (Rudolf et al. 2013) proposent quant à eux d'enrichir des solutions du marché comme SAP HANA[26] en ajoutant des opérations graphes directement dans le moteur de base de données.

2.4.3 Synthèse et application à nos travaux

Notre question de recherche s'intitule "comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?". La Recherche d'Information est le domaine conventionnel permettant de retrouver des informations à partir de requête. Elle correspond ainsi au fait de "retourner à la demande une information" dans notre question de recherche. Le domaine permet notamment de structurer, par le processus en U, les grandes étapes dimensionnant des solutions de Système de Recherche d'Information. Ce sont ces grandes étapes que nous reprenons dans l'analyse fonctionnelle de notre proposition de système. De plus, nous souhaitons utiliser les méthodes d'évaluation **usuelles** du domaine. Vis-à-vis des caractéristiques de la recherche d'information en entreprise vue dans cette dernière partie, nous souhaitons également permettre la recherche d'information **précise** tout en privilégiant quand c'est possible des méthodes **simples** de mise en oeuvre et de calcul. De plus, afin de s'adapter à la nature relationnelle et hétérogène des données de l'industrie manufacturière, nous souhaitons privilégier les méthodes et outils favorisant l'**exploitation des relations** entre données et capable d'une grande **flexibilité de schéma**. Enfin, concernant l'extension sémantique des requêtes, nous privilégions l'utilisation d'un graphe de connaissance pour l'utilisation d'un réseau de termes liés les uns aux autres sans contraintes inutiles à l'exercice (description des relations entre concepts pour décrire un domaine par exemple). L'ensemble de ces critères indiqués en gras a permis de choisir les méthodes et outils du Tableau 2.1 et du Tableau 2.3 aux plus pertinents pour notre étude. La sélection des différentes méthodes et outils ainsi que les critères de sélection sont indiqués en couleur bleue dans le Tableau récapitulatif 2.4.

Enfin, nous notons plusieurs absences dans l'état de l'art. En effet, pour définir les enjeux de la recherche d'information en entreprise, nous signalons qu'aucune étude ne s'est appuyée sur l'analyse méthodique d'un cas d'étude représentatif d'un contexte réel. Pour ce faire, nous notons également qu'aucune collection de tests usuels à l'évaluation de la Recherche d'Information n'est représentative des données de l'industrie manufacturière (données distribuées, hétérogènes et relationnelles). Enfin, concernant les opportunités d'utilisation du graphe au service de la Recherche d'Information, nous notons également l'absence d'une approche qui agrègent l'ensemble des données structurées et non structurées dans une modélisation graphe et adapté au contexte de l'industrie manufacturière. Nous proposons ainsi dans nos travaux de compléter ces manques.

[26]SAP HANA est une solution de stockage et de récupération de données pour des applications externes utilisant une combinaison de technologies dont l'accès en mémoire (in-memory) pour obtenir des temps de réponse performants

Tab. 2.4 : Sélection des méthodes vues dans l'état de l'art

Fonction	Méthode/Outil	Avantage	Inconvénient
Indexation	TF-IDF	Simplicité de calcul	Non prise en compte de la longueur, de la probabilité
	Autres (TF-IDF étendu, BM25)	Prise en compte de la longueur, probabilité	Complexité de calcul
Expression du besoin	Thématique	Guidé	Nécessité de fouille
	Requête	Simplicité d'expression	Transformation nécessaire
	Naturel	Pas de transformation	Complexité d'intégration
	Liste	Simplicité	-
	Valeur	Précision	Complexité d'intégration
Extension de requête	Modèle de connaissance	A disposition dès l'initialisation	Peut être inadapté au besoin
	Modèle de langue	A disposition dès l'initialisation	Peut être inadapté au besoin
	Retour de pertinence à l'aveugle	Adapté à l'utilisation	Besoin d'initialisation
	Hybride (Modèle de connaissance et retour de pertinence)	Augmente la précision	-
Appariement	Modèle booléen simple	Simplicité de calcul	Pas d'ordonnancement des résultats
	Autres modèles booléens (étendu et flou)	Ordonnancement des résultats	Coût de calcul
	Modèle algébrique	Ordonnancement des résultats et liens entre termes	Coût de calcul
	Modèle Probabiliste	Prise en compte de la structure des documents	Coût de calcul

Fonction	Méthode/Outil	Avantage	Inconvénient
Evaluation	Rappel	Usuel	Pas d'ordonnancement des résultats
	Précision	Usuel	Pas d'ordonnancement des résultats
	F-Mesure	Usuel et pondère rappel/précision	Pas d'ordonnancement des résultats
	Autres (Courbe Rappel/Précision, précision moyenne)	Prise en compte de l'ordonnancement des résultats	-
Extraction d'information	RI puis EI	Précision de réponse	-
	EI puis RI	Filtrage de collection	-
	Résultats RI pondéré par EI	Pondération des résultats	-
Type de base de données	Graphe	Exploitation des relations	Coût de calcul et distributivité
	Clé-valeur	Simple et distribuable	Ajout de métadonnée impossible , Recherche à travers les relations difficile
	Document	Simple avec documents	Recherche à travers les relations difficile
	Colonne	Rapidité et flexibilité	Recherche à travers les relations difficile

Fonction	Méthode/Outil	Avantage	Inconvénient
Base de données graphe	Neo4J	Meilleurs flexibilité de schéma et puissance de requête	Moins bonnes distributivité et capacité de multi-modèle
	ArrangoDB	Meilleurs distributivité et capacité de multi-modèle	Moins bonnes flexibilité de schéma et puissance de requête
	Autres (AllegroGraph, InfiniteGraph, OrientDB)	-	Moins bonnes flexibilité de schéma, puissance de requête, distributivité et capacité de multi-modèle
Représentation des connaissances	Graphe de connaissance	Représentation d'un réseau de connaissance au sens large	-
	Ontologie	Description des concepts et relations entre concepts d'un domaine	-
Graphe de connaissance pour extension de requête	ConceptNet	Meilleures précision et rappel	-
	Autres (DBpedia, Freebase ou par corpus)	-	Moins bonne précision et rappel

2.5 Synthèse de l'état de l'art

La première partie du chapitre présente le domaine de la Recherche d'Information et ses Systèmes permettant d'obtenir des réponses à des besoins d'informations en exploitant une collection de documents. Nous retenons vis-à-vis de notre contexte l'utilisation de la fonction TF-IDF pour pondérer les termes selon leurs représentativités dans les documents, du modèle booléen pour l'appariement entre les documents et les requêtes, du *focus* dans l'expression du besoin d'information, des mesures du rappel, de la précision et de la F-Mesure pour évaluer les Systèmes de Recherche d'Information et enfin l'utilisation de l'Extraction d'Information en complément à la Recherche d'Information pour venir identifier des extraits de document.

La seconde partie du chapitre de l'état de l'art présente la modélisation graphe de l'information hétérogène et ses avantages pour la recherche et l'exploitation des données dans un contexte où elles sont notamment relationnelles. Nous retenons vis-à-vis de notre contexte l'utilisation de Neo4J comme gestionnaire de base de donnée graphe et l'utilisation du graphe de connaissance multi-lingue ConceptNet comme ressource externe pour l'extension sémantique des requêtes.

Considérant notre question de recherche : "comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?", nous avons alors listé en troisième et dernière partie du chapitre de l'état de l'art les travaux utilisant l'approche de recherche d'information et de graphe pour répondre à des besoins de recherche d'informations adaptés à l'entreprise. Malgré une recherche active sur la modélisation de la connaissance en graphe dans le domaine, aucune proposition de Système de Recherche d'Information supportée par la modélisation graphe de l'ensemble des données structurées et non structurées n'est proposée et détaillée ce que présente le coeur de notre proposition dans le Chapitre 4. De plus, les listes d'enjeux à considérer dans les solutions de recherche d'informations adaptées à l'entreprise ne s'appuient pas sur l'analyse empirique de cas d'étude ce que nous proposons également de faire dans le Chapitre 4. Afin de construire l'architecture d'une première proposition, nous restons attentifs aux trois points suivants :

- la définition pertinente d'une collection de tests adaptée, sujet que nous traitons aussi bien pour la construction de la proposition mais également pour sa validation,
- la recherche adaptée à celle de l'entreprise influençant ainsi la formulation de la requête au delà d'un simple champ par mots-clés,
- la prise en compte d'un besoin de précision plus importante que celle du rappel dans le contexte de l'entreprise, et donc une pondération de la F-Mesure à 0.5 (le besoin de précision est deux fois plus important que le rappel).

Ce point nous permettra d'enrichir la proposition dans le Chapitre 5. Au préalable et en utilisant les enseignements de l'état de l'art, nous détaillons le cadre utilisé dans la construction de la proposition.

Chapitre 3

Cadre de construction de la proposition

3.1 Objectif du chapitre

Comme illustré dans la Figure 3.1, l'objectif de ce chapitre est de présenter l'ensemble du cadre nous ayant permis de construire et vérifier notre proposition de réponse à la question de recherche. Ce cadre se compose notamment de l'analyse fonctionnelle du système conforme aux étapes du processus de Recherche d'Information vue dans l'état de l'art. Les fonctions ainsi déterminées guideront la manière de présenter la proposition par la suite. Nous verrons ensuite le processus de vérification de la proposition et d'analyse des résultats obtenus utilisés dans les deux prochains chapitres. Enfin nous présentons le cas d'étude "PAINT'R" qui a été utilisé ainsi que les requêtes associées choisies et la liste des résultats pertinents pour chacune de ces requêtes.

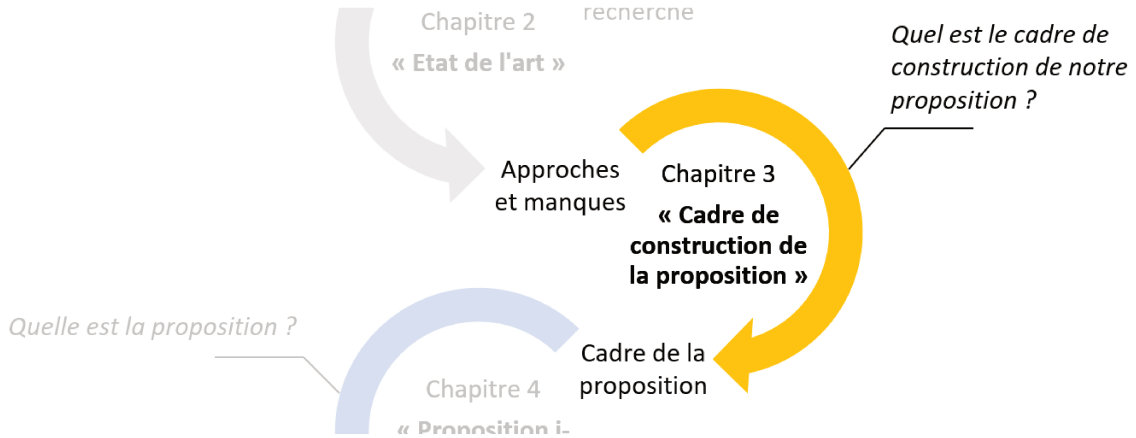


FIG. 3.1 : Organisation du mémoire : troisième chapitre

3.2 Analyse fonctionnelle

3.2.1 Système de recherche d'information

L'analyse fonctionnelle se concentre sur la description des fonctions offertes par un produit pour un besoin utilisateur défini (TASSINARI 2006). On distingue l'analyse fonctionnelle externe où l'on représente le besoin tel que l'utilisateur le voit et ne détaillant ainsi pas la combinaison des fonctions internes au produit. Ces détails sont fournis dans l'analyse fonctionnelle interne. Dans notre cas, l'analyse externe considère le besoin de rechercher une information à la demande, avec en entrée le besoin exprimé par l'utilisateur, la collection de données de l'entreprise et en sortie la ou les réponses au besoin d'information d'après la collection de données. Comme vu dans l'état de l'art, les systèmes de recherche d'information sont tout à fait appropriés afin de réaliser cette fonction, composante principale de notre question de recherche. Le système étudié a donc comme fonction principale la recherche d'information. Concernant les contraintes de cette fonction, elles sont au nombre de deux : le langage utilisé par l'utilisateur lors de l'expression du besoin ainsi que la nature des données à traiter. Comme illustrée dans la Figure 3.2 représentée avec la méthode IDEF0 (PRESLEY et al. 1995), la fonction principale de recherche d'information doit se décomposer dans une analyse fonctionnelle interne en plusieurs fonctions

internes afin de traiter séparément ces contraintes. Les trois fonctions internes identifiées sont :

- **Fonction 1 - Pré-traiter les données** : l'objectif de cette deuxième fonction est de transformer les données hétérogènes sources afin qu'elles soient adaptées au besoin et au langage de la Fonction 3,
- **Fonction 2 - Pré-traiter la requête** : l'objectif de cette fonction est de transformer la requête exprimée dans le langage de l'utilisateur en une requête adaptée aux besoins et au langage de la Fonction 3,
- **Fonction 3 - Identifier les réponses** : l'objectif de cette troisième fonction est de fournir les réponses aux requêtes pré-traitées par la Fonction 1 à partir des données pré-traitées par la Fonction 2.

Finalement, nous retrouvons les différentes étapes du processus de Recherche d'Information vue dans l'état de l'art à la Section 2.2.1, à savoir : la représentation des documents (ici étendue aux données de tout type), la représentation du besoin utilisateur par une requête et l'appariement entre la requête et les documents.

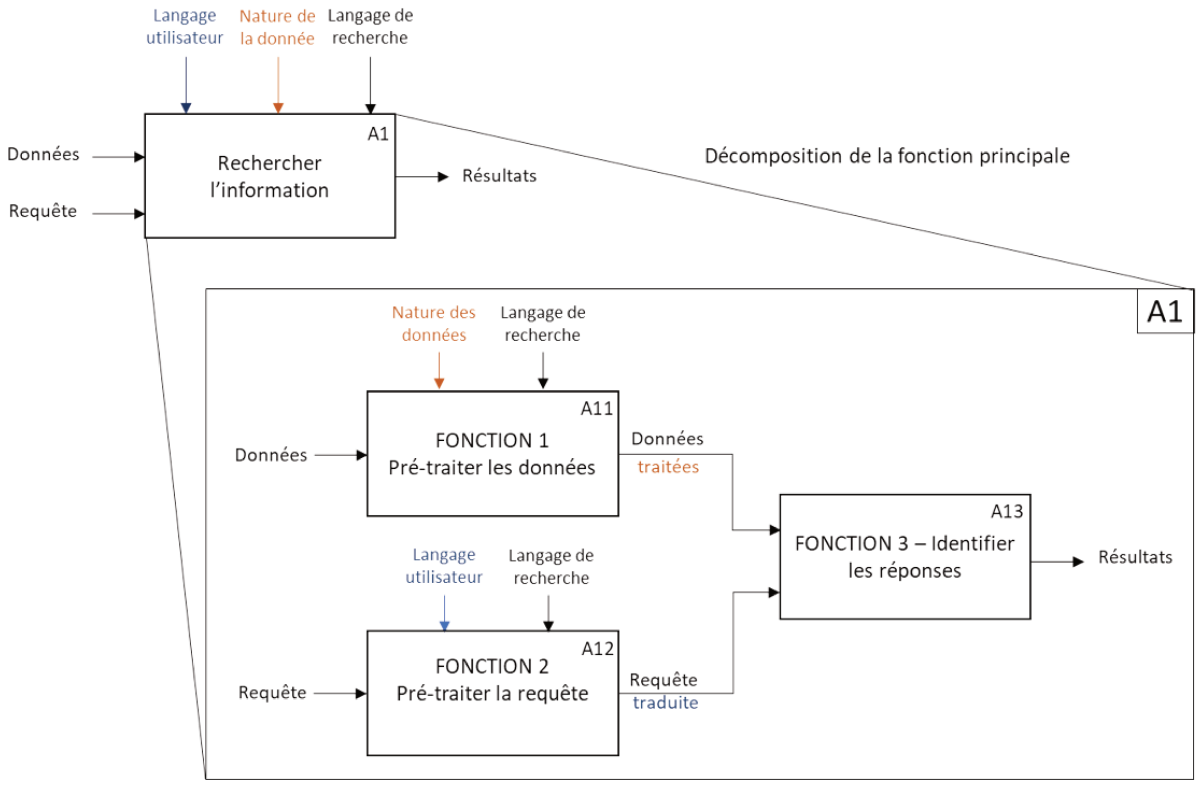


FIG. 3.2 : Fonctions principales de Recherche d'Information et sa sous-décomposition selon le formalisme IDEF0 (PRESLEY et al. 1995)

3.2.2 Modélisation induite par l'approche graphe

L'étude de l'approche graphe vue dans l'état de l'art nous oriente vers la représentation graphe des données hétérogènes sources afin d'exploiter au mieux leurs natures relation-

nelles, à la fois dans leurs recherches, mais également pour de potentielles analyses du réseau d'information. La modélisation graphe des données et de ses relations devient donc un outil dimensionnant de la "Fonction 3 - Identifier les réponses", contraint la "Fonction 2 - Pré-traiter la requête" car impose un langage de recherche adapté à la modélisation graphe et contraint la "Fonction 1 - Pré-traiter les données" car impose la transformation des modèles de données sources en un modèle de données graphe.

Les fonctions soumises à ce nouveau outils et ces nouvelles contraintes sont illustrées dans la Figure 3.3.

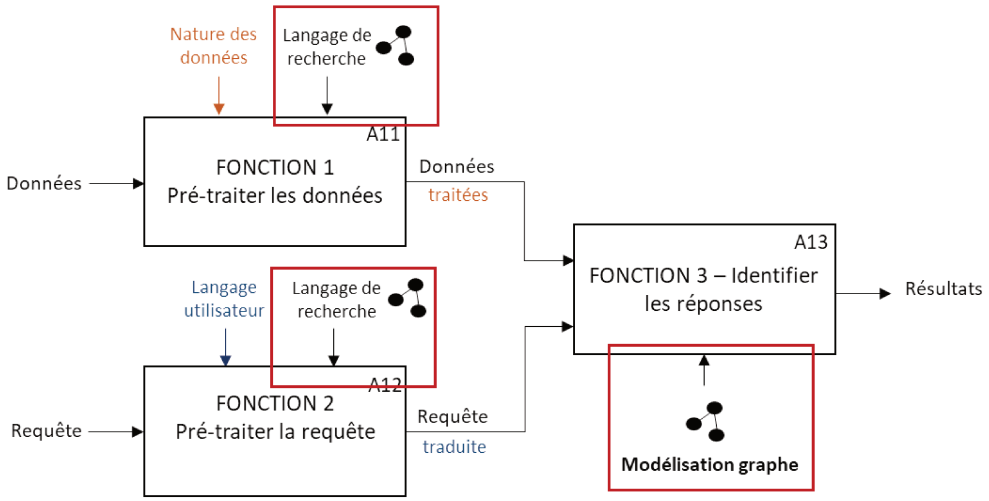


FIG. 3.3 : Représentation des fonctions d'un système de recherche d'information utilisant la modélisation graphe

3.3 Processus d'évaluation et d'enrichissement de la proposition

3.3.1 Mesures utilisées

Afin d'évaluer la proposition, notamment au cours de sa construction, nous évaluons l'Exigence 1 'la proposition doit fournir tous les résultats attendus' et l'Exigence 2 'la proposition ne doit fournir que des résultats corrects' (présentées dans le Chapitre 1) en utilisant respectivement la mesure du rappel et celle de la précision, toutes deux vues dans l'état de l'art (présenté dans le Chapitre 2 à la Section 2.2.5). Nous présenterons également la F-Mesure, également présentée dans l'état de l'art, afin de donner un score global à la performance du système. Nous proposons, conformément aux conclusions de l'état de l'art, de pondérer à 0.5 cette moyenne signifiant ainsi que le besoin de précision est deux fois plus important que le rappel dans le contexte de l'entreprise. Nous rappelons ci-après la formulation des trois mesures :

- Le rappel Ra s'exprime par :

$$\frac{1}{Q} * \sum_{q \in Q} \frac{|Np \cap Nr|}{|Np|}$$

- La précision Pr s'exprime par :

$$\frac{1}{Q} * \sum_{q \in Q} \frac{|Np \cap Nr|}{|Nr|}$$

- La F-Mesure F_β s'exprime par :

$$\frac{1}{Q} * \sum_{q \in Q} (1 + \beta^2) * \frac{Pr * Ra}{\beta^2 Pr + Ra}$$

Avec q la requête appartenant à l'ensemble Q des requêtes, Np le Nombre de documents pertinents, Nr le Nombre de documents restitués et β la pondération du rappel sur la précision de la F-Mesure.

3.3.2 Le processus

Afin d'évaluer les performances de la proposition mais également de cibler les modifications nécessaires à son amélioration, nous appliquons un processus avec une étape d'analyse des causes comme le suggèrent d'autres études de cas (Ershadi et al. 2018). Ce processus permet d'itérer sur la définition des fonctions principales jusqu'à atteindre les objectifs de performance. Le processus est illustré dans la Figure 3.4 avec la méthode de représentation IDEF0 (Presley et al. 1995).

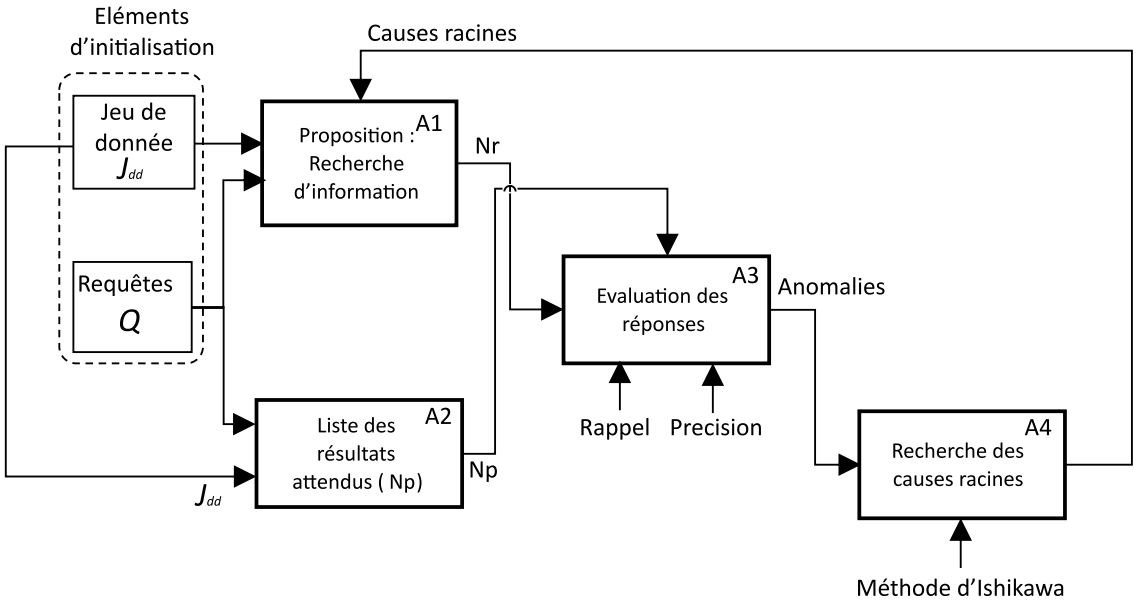


Fig. 3.4 : Processus d'évaluation et de modification de la proposition

Éléments d'initialisation : nous définissons ces éléments comme étant composées du jeu de données d'entrée J_{dd} et de l'ensemble des requêtes Q. Le jeu de données et les requêtes doivent être représentatifs du contexte de l'étude décrit dans le Chapitre 1.

Etape A1 : cette étape applique les requêtes Q au jeu de données J_{dd} en utilisant la proposition. On obtient la liste des résultats restitués Nr, ensemble du J_{dd} filtré par Q .

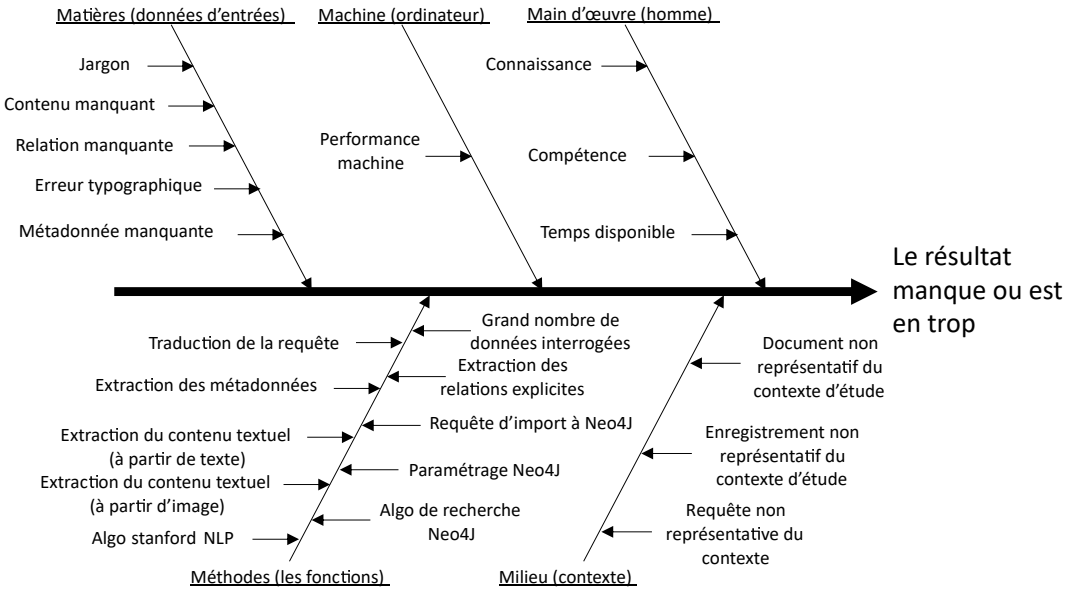


FIG. 3.5 : Diagramme d'Ishikawa adapté au cas d'étude

Etape A2 : cette étape permet de définir pour chaque requête la liste des résultats pertinents (Np) qui servira à évaluer les résultats restitués (Nr) par la proposition. Dans l'idéal, cette étape est réalisée par un groupement de personnes connaissant le jeu de données afin d'éviter des silences et même des bruits biaisant l'analyse des résultats.

Etape A3 : cette étape compare les ensembles Np et Nr afin d'obtenir les anomalies composées des bruits (soit les éléments de Nr n'appartenant pas à Np) et des silences (soit les éléments de Np exclu de Nr). Nous évaluons l'ensemble par les mesures de rappel et de précision vues dans l'état de l'art en Section 2.2.5.

Etape A4 : cette étape répartit ensuite chaque anomalie précédemment identifiée selon une liste de causes. Cette analyse des causes est basée sur la Méthode du diagramme d'Ishikawa (BARSALOU 2014) déjà éprouvé dans de l'analyse de systèmes de recherche d'information (WIECZERNIAK et al. 2017). Nous sélectionnons pour notre contexte cinq des grandes familles de causes données usuellement par la méthode : les données d'entrées (soit la Matière), les ressources utilisées (soit le Matériel), la main-d'oeuvre utilisée, les méthodes choisies (soit la Méthode) ou encore le contexte (soit le Milieu). Le diagramme ainsi construit est illustré en Figure 3.5. Une fois les anomalies réparties selon les cinq familles, nous obtenons une liste de causes réduisant la performance de la proposition.

Itérations : l'application du processus se réalise autant de fois que nécessaire pour atteindre les performances souhaitées. Une itération de processus revient alors à une modification de la proposition utilisée à l'étape A2, souhaitant contribuer au moins partiellement à la résolution d'une des causes détectées.

3.4 Cas d'étude PAINT'R - Sélection du jeu de données

Cette section présente le jeu de donnée J_{dd} sélectionné, élément d'initialisation du processus d'évaluation et de modification de la proposition comme le présente la Figure 3.6.

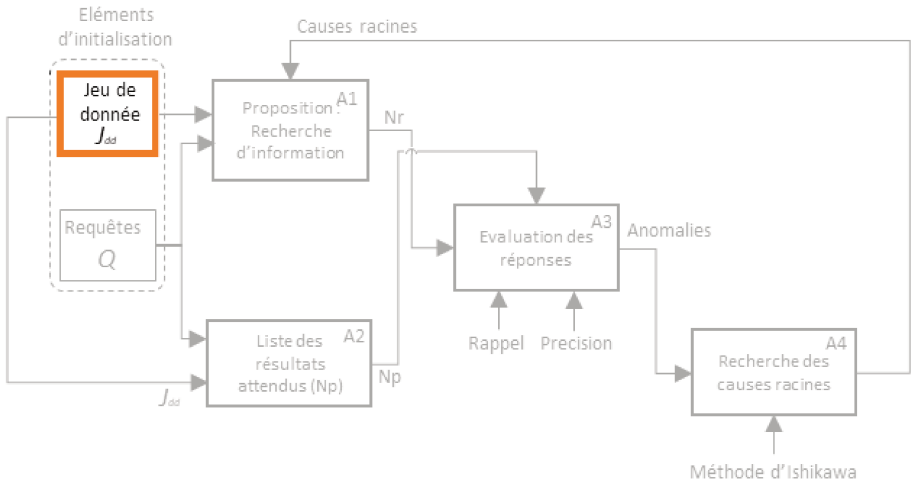


FIG. 3.6 : Processus d'évaluation et de modification de la proposition : définition du jeu de données

3.4.1 Présentation

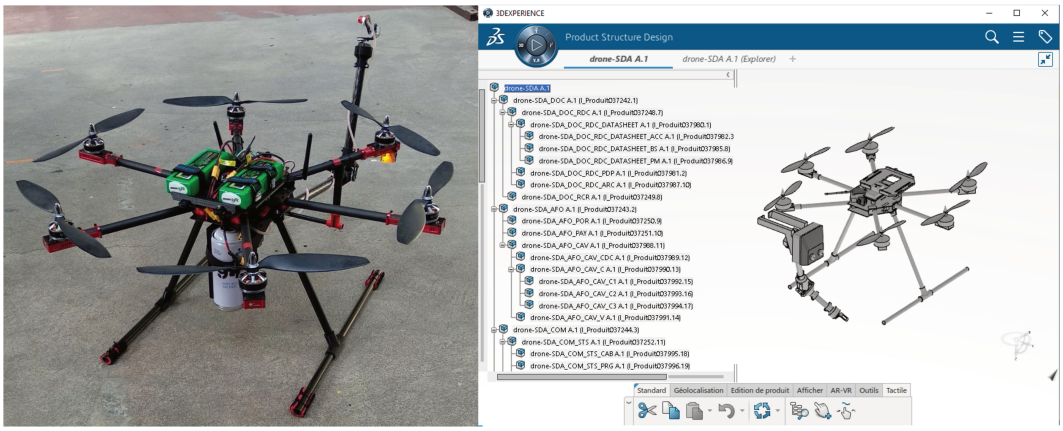


FIG. 3.7 : Drone Paint'Air et sa maquette numérique

Similaire aux données d'une entreprise manufacturière, le cas d'étude PAINT'R a été sélectionné pour sa couverture fonctionnelle multi-activités, sa forte hétérogénéité syntaxique incluant des données structurées et non structurées ainsi que son hétérogénéité sémantique due à de multiples créateurs et à deux langues utilisées. Le drone PAINT'R, dont la photo et la maquette numérique sont présentées dans la Figure 3.7, est un projet d'étudiants de master spécialisé "Créateur de solutions drones : technologies et usages

innovants" de l'ENSAM. L'objectif du projet a été de développer jusqu'au prototypage une solution de drone permettant de faire des retouches de peinture ou traitements de corrosion sur des ouvrages en hauteur. Le projet a été réalisé en distribuant les étudiants en plusieurs groupes de travail. Ces groupes d'étudiants avaient pour objectif de consigner l'ensemble des éléments liés au projet sous une maquette numérique initialement créée sous CATIA puis migrée sous la 3DExperience.

3.4.2 Enrichissement

Afin de considérer des bases de données relationnelles dans l'hétérogénéité des données, et ainsi être plus représentatif d'un cas d'entreprise de l'industrie manufacturière, le jeu de données a été enrichi pour l'étude. L'enrichissement a été réalisé avec des données soit déjà présentes mais sous une autre nature, soit par de nouvelles informations qu'une entreprise aurait réellement à sa disposition. Par exemple, les informations obtenues grâce aux demandes d'achats initialement sous format word ont permis de créer des bases de données relationnelles avec MySQL[1] liant les regroupements d'ordres d'achat et la liste des articles commandés à l'instar d'un ERP. Le jeu de données a également été enrichi avec des informations obtenues sur internet comme la description de fournisseurs impliqués dans le prototypage du drone et des commentaires clients sur des produits similaires. Au total, l'enrichissement représente 19% du volume total du jeu de données final.

3.4.3 Caractéristiques du jeu de données enrichi

Couverture fonctionnelle de l'ensemble de données : le jeu de données représente celles d'une entreprise de fabrication de drones composée de plusieurs services opérationnels. Ces différents services genèrent des données tout au long du cycle de vie du produit. Les activités de conception ont généré des documents présentant la gestion des exigences, la définition du produit et de ses composants, la simulation numérique et la nomenclature d'ingénierie. Les activités de production, de logistique et d'achat ont généré des documents relatifs aux chaînes de montage, aux fournisseurs et aux bons de commande. Les activités de maintenance et d'après-vente ont généré des manuels de maintenance et des retours clients. Les activités liées aux méthodes et à la qualité ont généré des documents relatifs aux processus, aux normes et aux méthodologies d'entreprise. Les activités de gestion de projet ont généré des documents de planification, des feuilles de route et des revues de projet. Enfin, l'activité des ressources humaines a généré des CVs et des rapports de la médecine du travail.

Hétérogénéité syntaxique des données : les données structurées proviennent du modèle numérique du drone géré sous la solution 3DExperience de Dassault Systèmes (y compris le stockage des éléments 3D) et de deux bases de données relationnelles gérées en MySQL[2]. Les données portent également de l'information non structurée telles que des

[1] https ://www.mysql.com/fr/

[2] MySQL est un système de gestion de bases de données relationnelles classique - https://www.mysql.com/

fichiers textes (.doc .xml .txt .log .ppt .pdf), des images (.jpg .png), des vidéos (.mp4), des 3D (.catpart, .catproduct, .stl) et des feuilles de calcul (.xls).

Hétérogénéité sémantique des données : une multitude d'acteurs ayant néanmoins le même parcours universitaire a participé à la création du jeu de données. De ce fait, le vocabulaire utilisé est tout de même varié, utilisant l'anglais et le français.

Quantité : le jeu de données est composé de 472 éléments répartis ainsi : 47% des données sont de type document tel que des fichiers textes, des images, des vidéos et des feuilles de calcul, 21% sont des éléments dont la seule fonction est la relation entre un objet et un autre, 17% sont des éléments de bases de données relationnelles comme la base achat et 15% sont des modèles CAO.

3.4.4 Accessibilité

Nous avons rendu disponible le jeu de données sous la plateforme Kaggle, site dédié à des exercices de science des données, sous licence Creative Commons CC BY-NC-SA 4.0 :

www.kaggle.com/dataset/a4ba6c3dbe1bc5a1cc8f05bb7ad825bcce106bff68ab58

2877a82107c000f9b1

3.5 Définitions des usages attendus et des requêtes

Les exigences de performance et de structure de la proposition doivent être évaluées sur des usages représentatifs au contexte de l'étude, ici à des usages de recherche d'information dans l'industrie manufacturière. Cette section présente donc la sélection des requêtes Q à partir de ces usages, élément d'initialisation du processus d'évaluation et de modification de la proposition comme le présente la Figure 3.8.

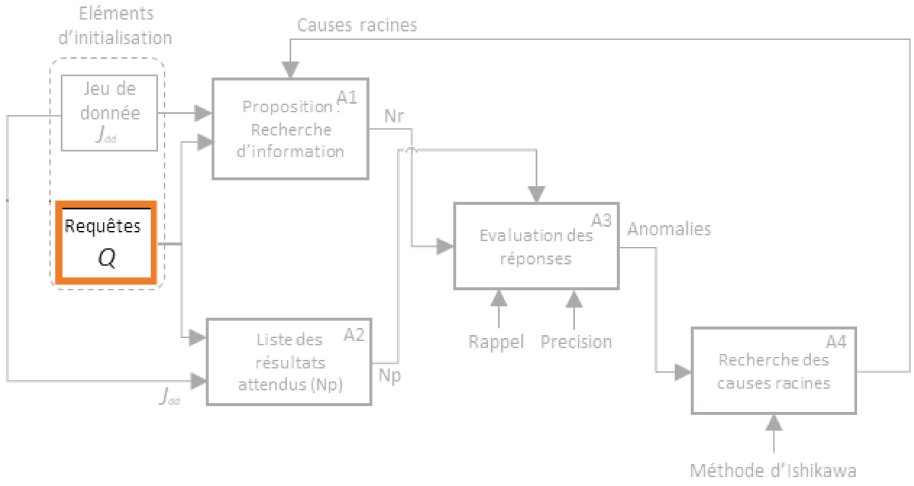


Fig. 3.8 : Processus d'évaluation et de modification de la proposition : définition des requêtes

3.5.1 Protocole d'identification des usages attendus

Pour déterminer quels sont les besoins représentatifs en recherche d'information pour l'industrie manufacturière, nous avons tout d'abord exploré des articles comme (Zhang et al. 2017) ou (J. Li et al. 2015) traitant entre autres des nouveaux usages émergeant avec l'avènement du big data dans un contexte PLM et/ou industriel. Ces articles nous ont permis d'élargir les réflexions et perspectives sur les usages possibles des données. Nous avons ensuite réalisé un brainstorming au sein du laboratoire afin d'enrichir la liste d'usages préalablement obtenue. Pour cela, nous avons utilisé les illustrations présentées en Figure 3.9 et Figure 3.10. Ces illustrations représentent les différents départements d'une entreprise avec les données qu'ils génèrent ainsi que des exemples d'acteurs et de leurs missions. Enfin une convergence avec la société Capgemini a été réalisée afin de classer cette liste selon trois critères : 1-"pertinent/innovant", 2-"à explorer si possible" et 3-"peu innovant/peu pertinent". Les usages ainsi gardés à l'étude répondant aux critères 1 et 2 sont listés dans le Tableau 3.1.

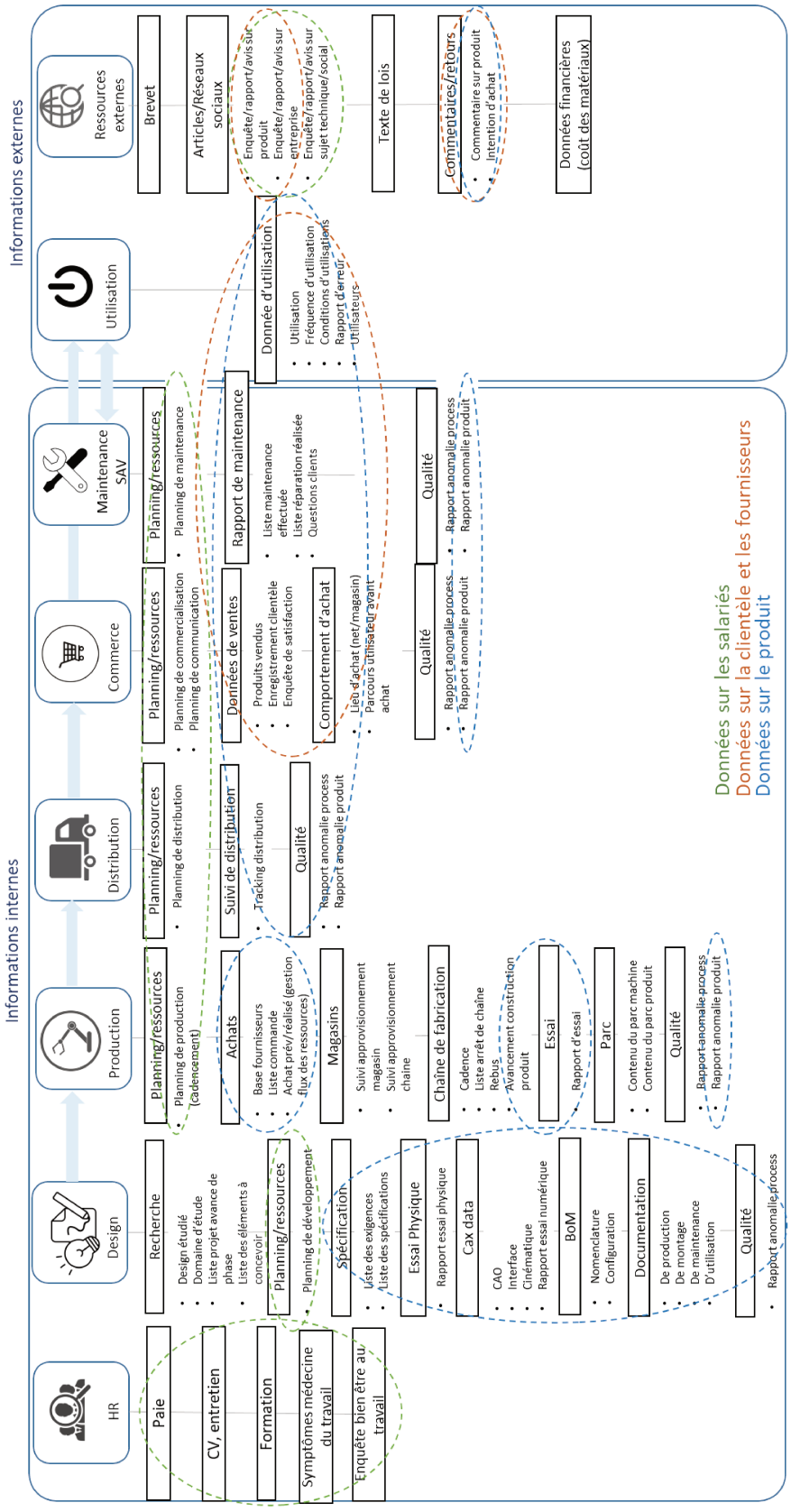


FIG. 3.9 : Illustration des départements et de leurs données au sein d'une entreprise de l'industrie manufacturière

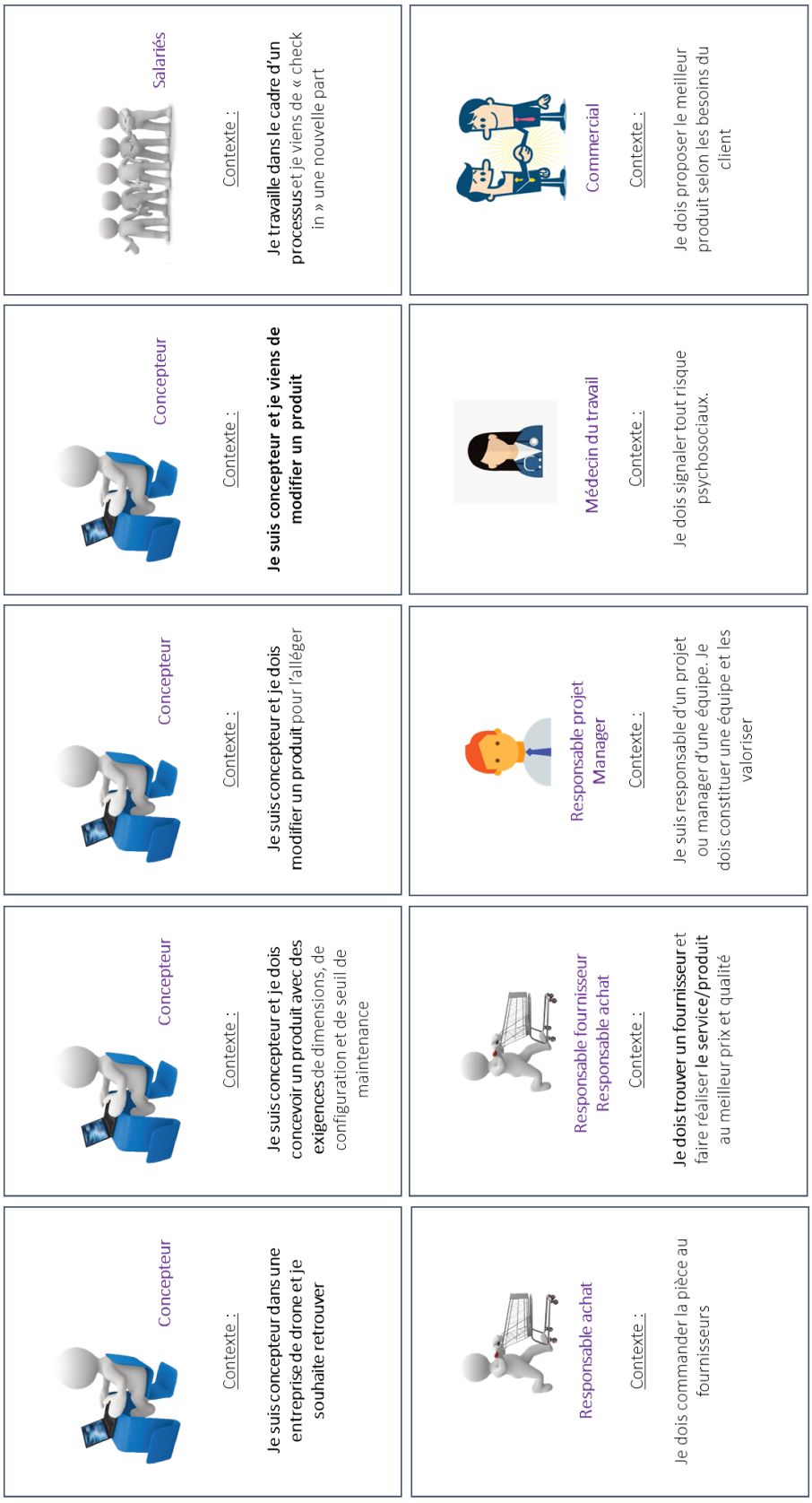


FIG. 3.10 : Exemples d'acteurs recherchant de l'information au cours du cycle de vie du produit

Tab. 3.1 : Liste des usages de recherche d'information dans l'industrie manufacturière

Numéro de l'usage	Description de l'usage
UC1	Identifier les produits existants pour favoriser la reconduction
UC2	Détecter des innovations pouvant répondre à mes exigences
UC3	Accéder aux processus, standards et méthodologies à appliquer pour une activité
UC4	Identifier des collaborateurs disponibles ayant des compétences souhaitées
UC5	Comparer deux produits selon des critères de choix
UC6	Identifier des partenaires commerciaux aux compétences souhaitées
UC7	Identifier les risques physiques et psychosociaux
UC8	Identifier des configurations adaptées à chaque client
UC9	Accéder aux justifications de choix de conception
UC10	Identifier les exigences liées à un système ou un composant

3.5.2 Définition des requêtes

Les requêtes Q des utilisateurs peuvent être formulées de différentes manières : langage naturel, utilisation de mots-clés ou recherches avancées avec des opérateurs tels que "OR", "AND" etc. Pour considérer les demandes d'information provenant de l'industrie manufacturière, nous avons traduit les usages définis dans la Section 3.5.1 en requêtes que nous présentons dans le Tableau 3.2. On considère que les utilisateurs emploient plusieurs langues que nous limiterons au français et à l'anglais pour notre étude.

D'autre part, suite à l'observation des requêtes obtenues, nous les regroupons selon quatre typologies présentées ci-dessous. Ces typologies permettront des analyses plus approfondies dans les prochains chapitres.

Les requêtes de type (A)

La première catégorie de requête concerne la recherche de tous les éléments mentionnant des informations spécifiques comme la recherche de tous les éléments relatifs à un produit.

Les requêtes de type (B)

La seconde catégorie de requête concerne la recherche d'un certain type d'éléments (mentionnant des informations spécifiques) comme la recherche des brevets relatifs à une technologie.

Les requêtes de type (C)

La troisième catégorie de requête concerne la recherche d'une valeur précise comme la recherche du prix d'un produit.

Les requêtes de type (D)

Enfin, la quatrième catégorie de requête concerne la recherche de phrases exprimant des notions spécifiques comme la recherche des phrases d'exigences liées à un produit".

Selon la classification des requêtes vue dans l'état de l'art, les requêtes de type (A) sont des requêtes larges tandis que les requêtes de type (B), (C) et (D) sont des requêtes spécifiques. Nous ne traiterons donc pas des requêtes par similarité comme la recherche d'un document ressemblant à un autre.

Ainsi, nous avons défini la liste de requêtes adaptée au cas d'étude et présentée dans le Tableau 3.2.

Tab. 3.2 : Liste des requêtes appliquées au jeu de données PAINT'R

N° d'usage	N° de requête	Requête	Type
UC1	q_1	Trouver tous les éléments mentionnant le terme de batterie	(A)
UC1	q_2	Find all items mentioning the term brake	(A)
UC1	q_3	Trouver tous les éléments mentionnant l'articulation du bras	(A)
UC1	q_4	Trouver tous les éléments mentionnant le terme de télécommande	(A)
UC2	q_5	Find all patents on blade 'additive manufacturing'	(B)
UC5	q_6	Trouver toutes les simulations sur le capot	(B)
UC3	q_7	Trouver toutes les revues de projet	(B)
UC3	q_8	Trouver tous les processus sur recrutement	(B)
UC3	q_9	Trouver tous les standards sur nettoyage filtre	(B)
UC3	q_{10}	Trouver toutes les méthodologies sur le nettoyage du filtre	(B)
UC3	q_{11}	Trouver toutes les procédures de test	(B)
UC4	q_{12}	Trouver le planning de Frederic Segonds	(B)
UC1	q_{13}	Trouver toutes les références de batterie	(C)
UC1	q_{14}	Trouver toutes les références de batterie de 14.8V	(C)
UC5	q_{15}	Find all reference of propeller	(C)
UC5	q_{16}	Trouver les prix de la référence 4S5200	(C)
UC5	q_{17}	Trouver toutes les propriétés du moteur	(C)
UC5	q_{18}	Trouver tous les commentaires sur le poids	(C)
UC5	q_{19}	Trouver tous les commentaires sur la batterie	(C)
UC6	q_{20}	Find all suppliers with skills on drone	(C)
UC4	q_{21}	Find all employees with skills on 'additive manufacturing'	(C)
UC7	q_{22}	Trouver la liste des symptômes médicaux touchants les salariés	(C)
UC8	q_{23}	Trouver les vitesses utilisées par le client Serge Bernard	(C)
UC10	q_{24}	Find all requirements containing the term battery	(D)
UC9	q_{25}	Find all the justifications for the choice of the engine	(D)

3.6 Définition des ensembles de résultats pertinents

Cette section présente la sélection des résultats pertinents par requête N_p représenté par le bloc A2 du processus d'évaluation et de modification de la proposition comme le présente la Figure 3.11.

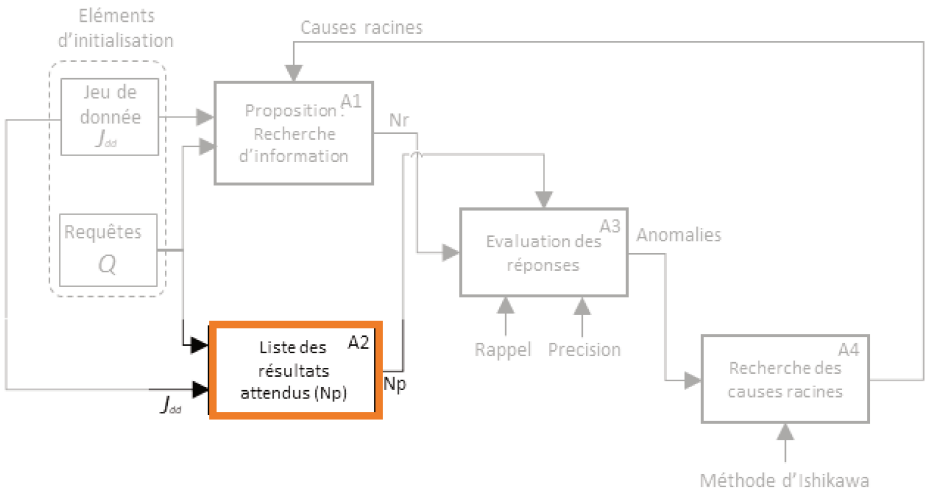


FIG. 3.11 : Processus d'évaluation et de modification de la proposition : définition des résultats attendus

3.6.1 Démarche

La recherche des résultats pertinents Np pour chacune des requêtes q a été réalisée manuellement suite à une prise de connaissance approfondie du jeu de données par la doctorante. Cette prise de connaissance s'est faite par exploration de chacune des branches de la maquette numérique du drone en question ainsi que par la lecture de l'ensemble des documents. De plus, la doctorante est également celle qui a enrichi le jeu de données. L'implication de la doctorante pour définir l'ensemble des résultats attendus Np par requête apparaît donc opportun.

3.6.2 Vérification de la démarche

Afin d'estimer néanmoins un pourcentage d'erreur à considérer lors de la définition des ensembles Np, une expérimentation faisant intervenir des tiers a été menée.

Le protocole présenté en Figure 3.12 a été appliqué pour 5 requêtes des 17 requêtes utilisées dans la définition des enjeux clés du Chapitre 4. La sélection des requêtes a été réalisée afin de couvrir plusieurs types de requête. Le protocole a été réalisé par des participants ayant des connaissances hétérogènes du jeu de données :

- Le participant 0 est celui qui a défini l'ensemble des résultats pertinents initial Np_0. Il a une connaissance importante du jeu de données l'ayant exploité pour ses travaux de recherches. La définition de ces ensembles a pris plus d'une journée.

- Les participants 1 et 2 ayant respectivement définis les ensembles Np_1 et Np_2 avaient des connaissances nulles sur le jeu de données avant le démarrage de l'expérience.
- Le participant 3 ayant défini l'ensemble Np_3 avait des connaissances sur le jeu de données avant le démarrage de l'expérience. En effet, il avait encadré des étudiants l'utilisant à plusieurs reprises.

Les résultats obtenus sont affichés dans le Tableau 3.3. Np' consolidé étant l'ensemble des résultats revus suite à l'analyse des Np_x où x désigne le participant entre 1 et 3. En effet, 2 résultats sur la requête Q13 apportés par le participant 1 ont été jugés pertinents. On obtient donc que le participant 0 avait défini 92% des résultats pertinents et qu'en moyenne, les autres participants obtiennent 32% de l'ensemble des résultats pertinents. De plus, certains résultats fournis par les participants ont étés jugés après analyse non pertinents.

Tab. 3.3 : Analyse des résultats obtenus lors de l'application du protocole de définition des résultats attendus

N° de requête	Np_0	Np' consolidé	Np_1 ∩Np'/Np'	Np_2 ∩Np'/Np'	Np_3 ∩Np'/Np'
Q1	10	10	2/10	2/10	0/10
Q4	5	5	2/5	1/5	1/2
Q6	2	2	2/2	1/2	1/2
Q9	2	2	2/2	1/2	2/2
Q13	4	6	5/6	0/6	1/6
Total	23	25	13	5	6
Répartition	92%	100%	52%	20%	24%
Temps (min)	$\sim >$ 300	-	76	41	25

En synthèse, nous estimons que la définition des ensembles Np peut souffrir de 8% de silence, mais reste 68% plus complète que la moyenne identifiable par des tiers.

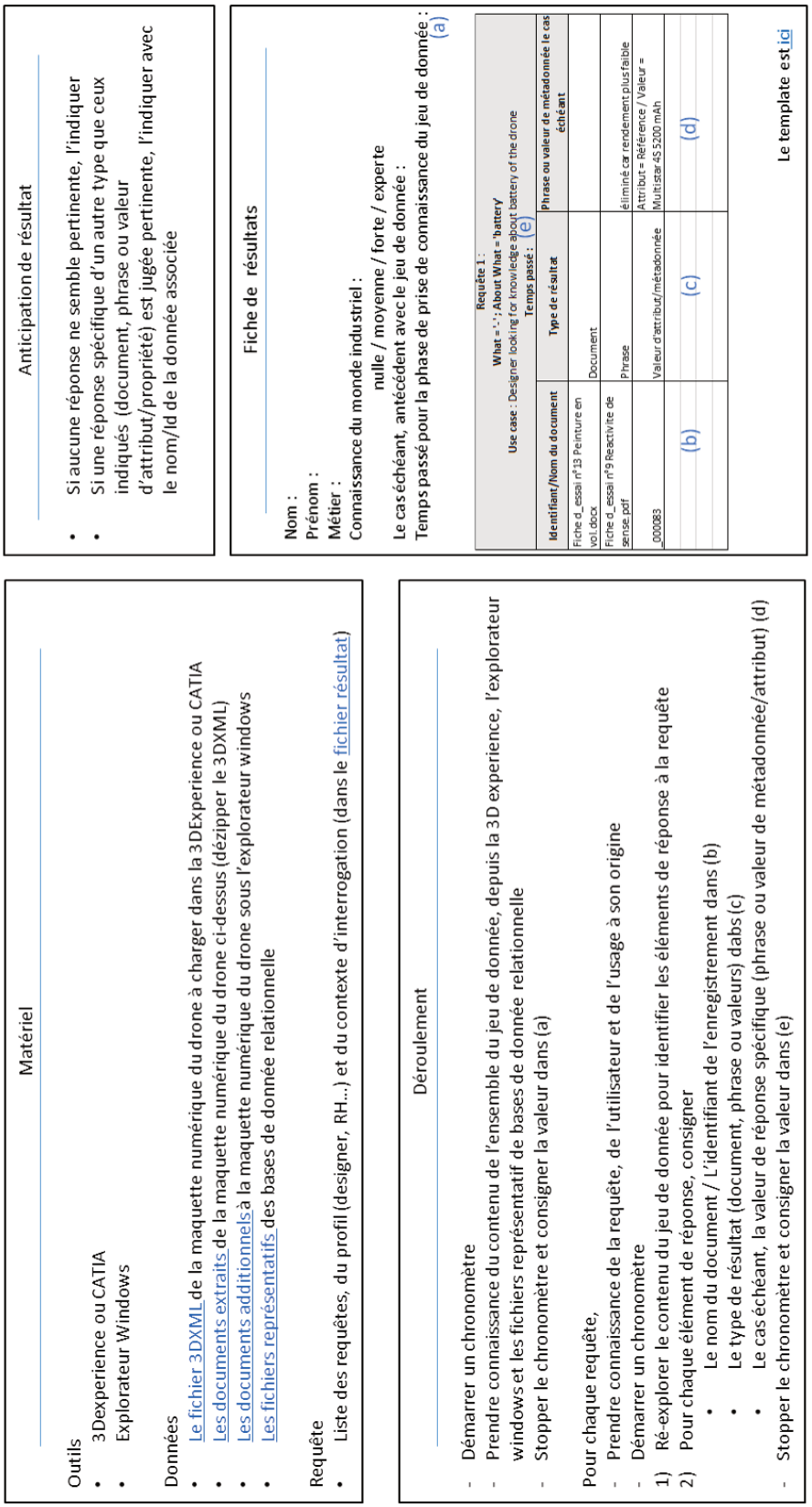


FIG. 3.12 : Protocole de définition des résultats attendus par requête – jeu de donnée PAINT'R

3.7 Synthèse

Nous avons présenté dans ce chapitre l'ensemble du cadre nous permettant de construire et d'évaluer pour optimiser notre proposition de réponse à la question de recherche. En effet, après avoir présenté les principales fonctions de la proposition, nous avons présenté le processus de validation et de modification de celle-ci en détaillant les éléments d'initialisation utilisées à savoir : le jeu de donnée du drone PAINT'R et la liste de requêtes répondant à des usages attendus de recherche d'information dans l'industrie manufacturière. Enfin, nous avons présenté la démarche utilisée pour établir les listes de résultats pertinents par requête, éléments déterminant dans l'évaluation de la proposition. C'est avec l'ensemble de ce cadre que le chapitre suivant présente et évalue notre proposition de réponse à la question de recherche.

Chapitre 4

Proposition i-Dataquest

4.1 Objectifs du chapitre

Comme illustré dans la Figure 4.1 et pour faire suite aux conclusions de l'état de l'art présentées dans le Chapitre 2 ainsi qu'à la présentation du cadre de construction et d'amélioration de la proposition de réponse à la question de recherche faite dans le Chapitre 3, l'objectif de ce chapitre est double :

- présentation de la proposition i-Dataquest, un système de recherche d'information exploitant la modélisation graphe des données hétérogènes sources,
- application de cette proposition au cas d'étude PAINT'R afin de définir les enjeux clés à considérer pour l'amélioration de la proposition.

La proposition i-Dataquest (L. Kim et al. 2020) ainsi que les enjeux clés (L. Kim et al. 2021) ont notamment été présentés lors de conférences internationales.

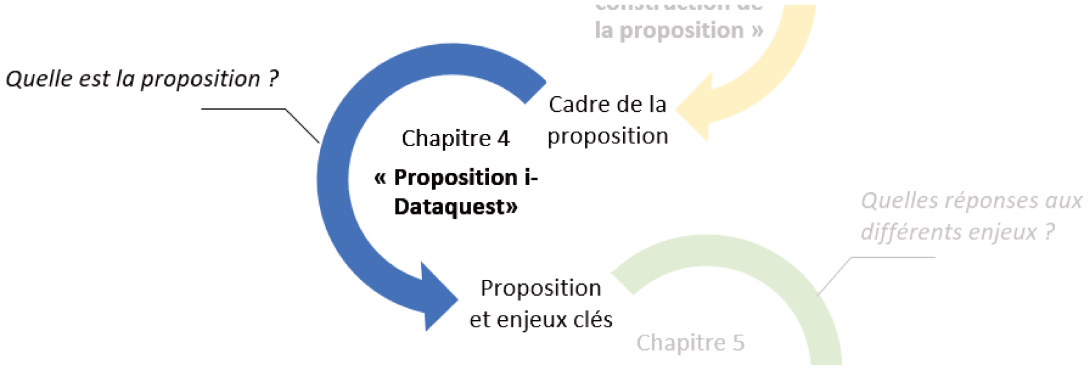


Fig. 4.1 : Organisation du mémoire : quatrième chapitre

4.2 Préambule

4.2.1 Notations

Les notations utilisées dans ce chapitre sont présentées dans le Tableau 4.1. Les notations comprennent J_{dd}, G_d et n déjà utilisées lors du précédent chapitre.

Tab. 4.1 : Liste des principales notations du Chapitre 4

Notations	Descriptions
J_{dd}	ensemble des données sources
G_d	graphe des données
n et m	sommets du graphe des données
p_n	propriété du sommet n
v_{p_n}	valeur de la propriété p du sommet n
l_n	étiquette du sommet n
r	arête de G_d
l_r	étiquette de l'arête r
$CREATE$	commande de création d'un objet dans G_d
$MATCH...WHERE...RETURN$	commande de recherche dans G_d
$UNIONALL$	concaténation de deux recherches
$requets.get(parameter)$	appel d'informations depuis une source externe

4.2.2 Echantillon de requête pour illustrer la proposition

Pour illustrer les propos de ce chapitre, nous utiliserons quatre des vingt-cinq requêtes présentées dans le Chapitre 3 à la Section 3.5.2. En effet, nous sélectionnons quatre requêtes afin de considérer un nombre de requêtes raisonnable tout en permettant d'illustrer les potentielles différences entre les quatre type de requêtes possibles (voir également en Section 3.5.2). Les quatre requêtes sont :

Requête q_1 :

Besoin d'information : quels sont les documents mentionnant le terme 'batterie' ?

Type de requête : (A)

Requête q_5 :

Besoin d'information : y a-t-il des brevets sur la fabrication additive des pales ?

Type de requête : (B)

Requête q_{16} :

Besoin d'information : quel est le prix de la batterie 4S5200 ?

Type de requête : (C)

Requête q_{23} :

Besoin d'information : quelles sont les exigences liées à la batterie ?

Type de requête : (D)

4.3 Architecture générale

4.3.1 Intégration dans l'entreprise

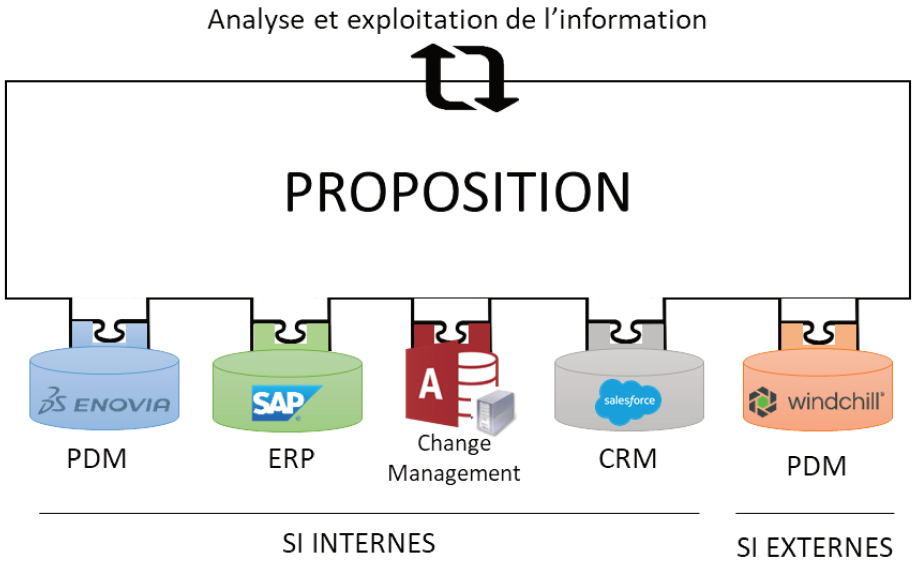


Fig. 4.2 : Positionnement de la proposition dans l'environnement d'entreprise

Comme l'illustre la Figure 4.2, la proposition i-Dataquest souhaite donner un accès unique à l'ensemble des informations hétérogènes de l'entreprise et ceci malgré ses silos et la modularité de ses systèmes d'information. Elle doit donc se connecter aux multiples sources de données sans besoin d'adaptation coûteuse. Les éléments de connexions à une base de données devraient se limiter dans l'idéal au renseignement de son 'nom', son 'type', à l' 'identifiant' et au 'mot de passe' de connexion ainsi que son 'adresse', le 'serveur' et le 'port de connexion' dédié. La reconnaissance du type de donnée, et donc la manière de la traiter, doit être automatisée. On suppose pour la suite de l'étude que l'ensemble de ces connexions informatiques entre la proposition et les sources de données sont réalisées.

4.3.2 Présentation de l'architecture

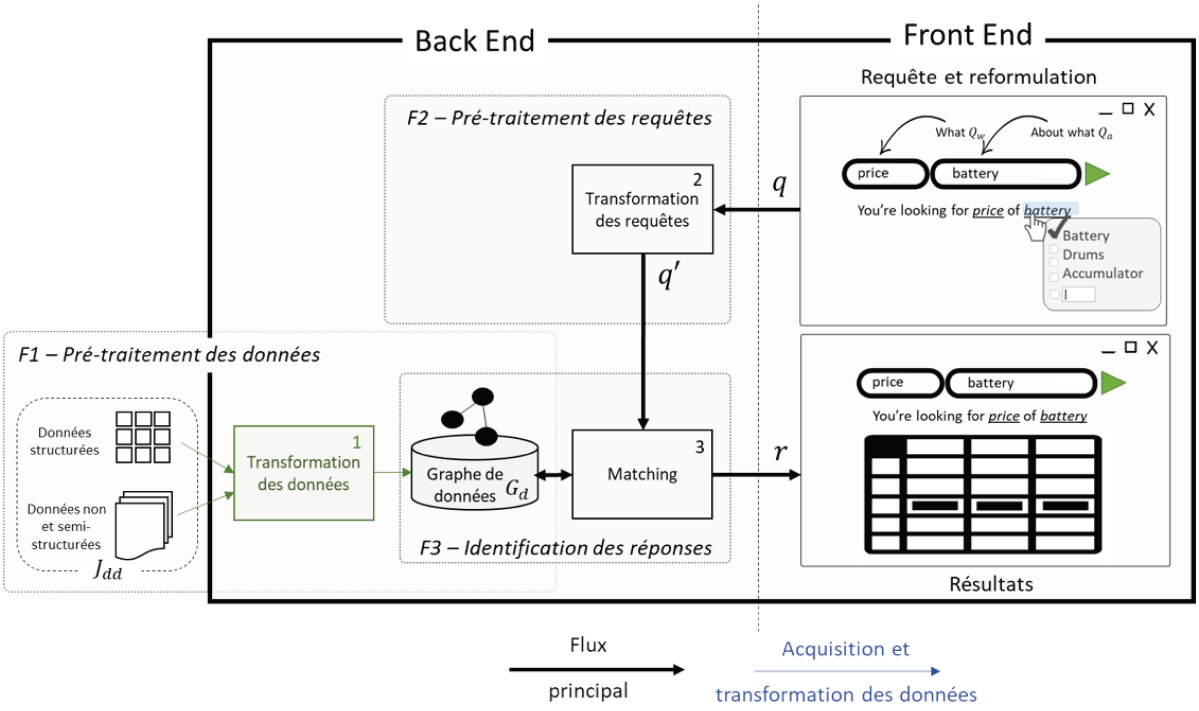


FIG. 4.3 : Présentation de l'architecture générale

L'architecture de la proposition i-Dataquest est présentée dans la Figure 4.3. Elle est décomposée en une partie 'Back End' et une partie 'Front end'. La première représente l'ensemble de l'architecture masquée à l'utilisateur. La partie 'Front End' est quant à elle composée des interfaces graphiques permettant le dialogue entre l'utilisateur et le 'Back End'. Cette dernière est décomposée selon les trois fonctions principales du système :

F1 - Pré-traitement des données : la première fonction est décrite dans la Section 4.4) et génère le graphe de donnée G_d dans lequel le système identifiera les réponses aux requêtes utilisateurs.

F2 - Pré-traitement des requêtes : la seconde fonction est quant à décrite en Section 4.5) et permet de transformer la requête utilisateur q en une requête graphe q'.

F3 - Identification des réponses : enfin, la troisième fonction permet d'identifier dans le graphe des données G_d les éléments de réponses r à la requête pré-traitée q'. Ces résultats sont ensuite affichés à l'utilisateur. Ces éléments sont décrits dans la Section 4.6.

La partie 'Front End' est quant à elle décomposée en deux sous-parties, celle de l'expression du besoin d'information qui permet d'alimenter la fonction 2 et celle de la présentation des réponses au besoin d'information, sortie de la fonction 3. La conception de l'interface utilisateur et donc de son expérience a été inspiré des moteurs de recherche web connus.

4.4 Génération du graphe des données G_d (FONCTION 1)

4.4.1 Définition du graphe des données G_d

Reprenant le cadre de la théorie des graphes vue dans l'état de l'art, nous nommons le graphe des données G_d et le définissons ainsi :

$$G_d = (V, E) = (N, R) \tag{4.1}$$

G_d est un graphe simple orienté non pondéré étiqueté aux sommets et aux arêtes. Chaque sommet contient également au moins une propriété nommée 'Id', identifiant unique du sommet dans le graphe. V pour 'Vertice' est la notation usuelle en théorie des graphes pour désigner l'ensemble des sommets tandis que N pour 'Node' est celle utilisée dans la pratique. E pour 'Edge' est la notation usuelle en théorie des graphes pour désigner l'ensemble des relations connectant un sous-ensemble de deux sommets de V tandis que R pour 'relationship' est celle utilisée dans la pratique. Comme illustrée dans la Figure 4.4, nous utiliserons par la suite la notation n et r pour désigner chaque sommets et arêtes entre sommets et désignons l'étiquette de l'arête r par l_r, l'étiquette du sommet n par l_n et ses propriétés par p_i.

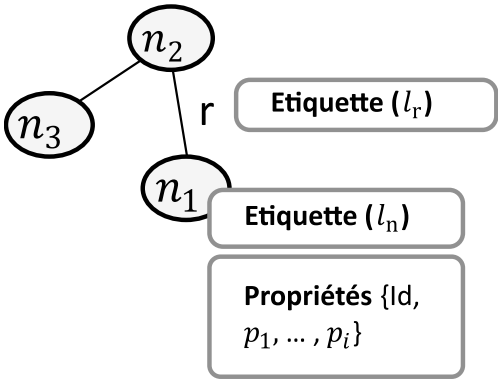


FIG. 4.4 : Modélisation du graphe des données G_d

4.4.2 Pré-traitement des données : transformation en graphe de données

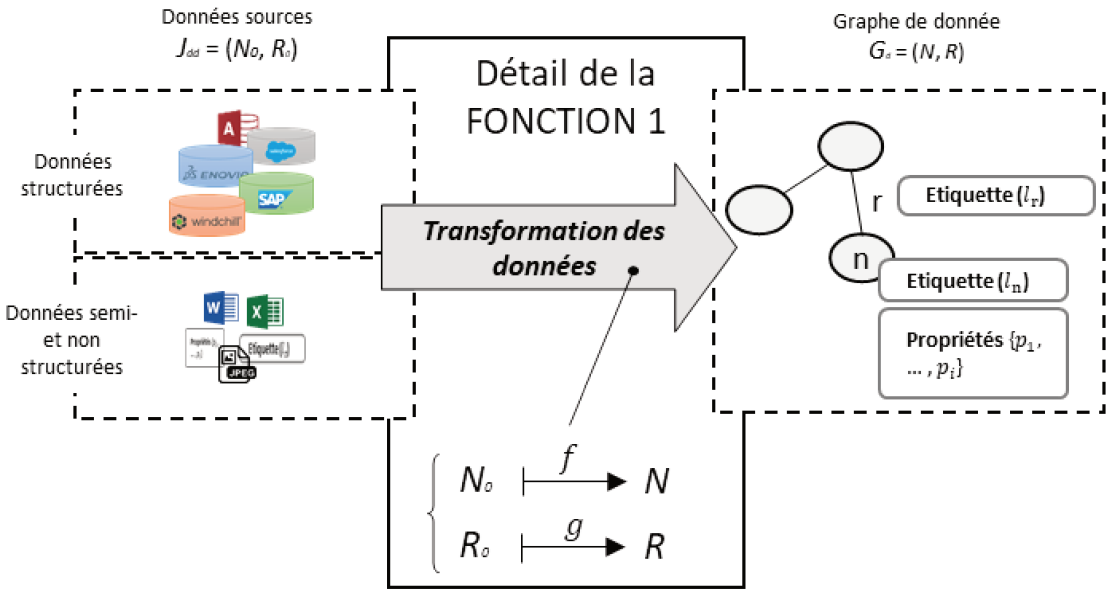


FIG. 4.5 : Des données hétérogènes et distribuées de l'entreprise à une modélisation graphe unique

Le graphe de données G_d est construit à partir des données structurées, semi- et non-structurées de l'entreprise. Les fichiers semi-structurés sont considérés par la suite comme des fichiers non-structurés, à savoir que dans le cas d'un contenu textuel, les éléments structurant comme les étiquettes internes ne sont pas distinguées du reste du texte. Ce choix est fait pour simplifier l'exercice de transformation dans cette proposition qui sera amélioré selon l'analyse de sa performance. Les données proviennent de multiples sources d'informations de l'entreprise regroupées sous la notation J_{dd} pour 'Jeu De Données' où $J_{dd} = (N_0, R_0)$, N_0 étant l'ensemble des données structurées ou non et R_0 étant l'ensemble des relations explicites entre données structurées ou non. Comme l'illustre la Figure 4.5, l'ensemble des sommets N est obtenue par une bijection f de l'ensemble des données sources N_0. L'ensemble des relations R est également obtenu par une bijection g de l'ensemble des relations explicites sources R_0. Ces bijections seront regroupées dans la désignation 'transformation des données' et leurs définitions, détaillées dans la section suivante, influencent la capacité du système à trouver l'information et donc la performance du système. En effet, l'absence d'informations recherchées dans le graphe peut entraîner des silences tandis qu'une génération d'éléments non recherchés dans le graphe peut entraîner du bruit. L'exercice est alors de tendre vers le bon niveau d'intégration d'information afin d'assurer les bonnes réponses aux recherches d'informations. Pour effectuer la transformation des données, on considère l'ensemble des données en entrée, que cela soit des enregistrements en bases de données relationnelles ou bien des documents comprenant (non exclusivement) du contenu textuel, image et tableur. Le souhait est de considérer tout type de données rencontré fréquemment en entreprise, qu'elles soient structurées, semi ou non structurées. Tous ces éléments sont ensuite transformés en modèle de données orientées graphe grâce à un ensemble d'opérations.

Le graphe G_d en sortie de la fonction 1 s'écrit alors par l'expression du couple 'sommet/arête' (n, r) où le sommet est détaillé par son étiquette l_n puis ses propriétés $p_i : v_{p_i}$ et l'arête est détaillée par son étiquette l_r et les sommets n et m qu'il permet de lier. En langage CYPHER (Section 2.3.2), l'expression donne :

$$G_d = (n, r) \leftrightarrow (: l_n\{p_1 : v_{p_1}, ..., p_i : v_{p_i}\}, (n) - [: l_r] \rightarrow (m)) \tag{4.2}$$

4.4.3 Les fonctions de transformation

Les fonctions f et g sont illustrées dans le Tableau 4.2 et le pseudo-code associé est présenté dans l'Algorithme 1. A chaque sommet créé, on indique une propriété 'id' dont la valeur est un identifiant incrémental et unique. Nous ne plaçons pas cette information dans l'Algorithme 1 afin de gagner en lisibilité.

Création des sommets n_1 à partir des bases de données relationnelles : les lignes 1 à 3 de l'Algorithme 1, représentées par les colonnes Ⓐ de la ligne ① du Tableau 4.2, transforment chaque enregistrement en un sommet indépendant du graphe. L'étiquette du sommet porte le nom de la table et les propriétés du sommet sont les attributs et les valeurs d'attributs dans la table.

Création des arêtes r à partir des bases de données relationnelles : les lignes 4 à 7 de l'Algorithme 1, représentées par les colonnes Ⓑ de la ligne ① du Tableau 4.2, identifient pour tous les enregistrements des tables comprenant des clés-étrangères[1] les enregistrements associés dans les autres tables. Elles créent ensuite les relations entre les sommets du graphe associés. L'étiquette de la relation se nomme 'in_relation_with'

Création des sommets n_2 à partir des documents non structurés : les lignes 8 à 10 de l'Algorithme 1, représentées par la ligne ② du Tableau 4.2, génèrent pour chaque document un sommet dont l'étiquette est le type du fichier (word, excel, etc.) et dont les propriétés sont renseignées avec les métadonnées du fichier. On ajoute la propriété 'contents' dont la valeur est le résultat de l'extraction du contenu textuel du document, provenant d'une image ou d'un texte. Le contenu non textuel n'est pas récupéré.

[1]Une clé étrangère identifie une colonne d'une table et la référence à une colonne d'une autre table créant ainsi une relation entre les deux tables

		Ⓐ sommet $N_0 \xmapsto{f} N$					Ⓑ arête $R_0 \xmapsto{g} R$	
	Données sources	n	l_n	Name (p_i) *	Value (p_i) *	Value($p_{content}$)	r	l_r
①	structurées	enregistrement	nom de table	nom d'attribut	valeur d'attribut	-	clé étrangère	:in_relation_with
②	Semi et non-structurées	fichier	propriété type du fichier	nom de propriété	valeur de propriété	contenu textuel du fichier	-	-

* Identifiant unique exprimé par Name (p_0) = 'Id' et Value (p_0) = '_00000x', x incrémentation à chaque sommet créé.

TAB. 4.2 : Règles de transformation des données structurées et non structurées en modèle de donnée orienté graphe

Algorithm 1 : Génération du graphe de données

Input : *table*, *enregistrement*, *fichier*

foreach *table* **do**
  foreach *enregistrement* **do**
    CREATE $n = (:table\{\ attribut : valeur_{attribut}\})$
foreach *table* **do**
  foreach *cleEtrangere* **do**
    foreach *enregistrement* **do**
      CREATE $(n :table)$-[:in_relation_with]->$(m :cleEtrangere)$
foreach *fichier* **do**
  $texte \leftarrow$ extraction du texte du fichier
  CREATE $n = (:type\{metadonnee : valeur_{metadonnee}, content : texte\})$

4.5 Transformation des requêtes (FONCTION 2)

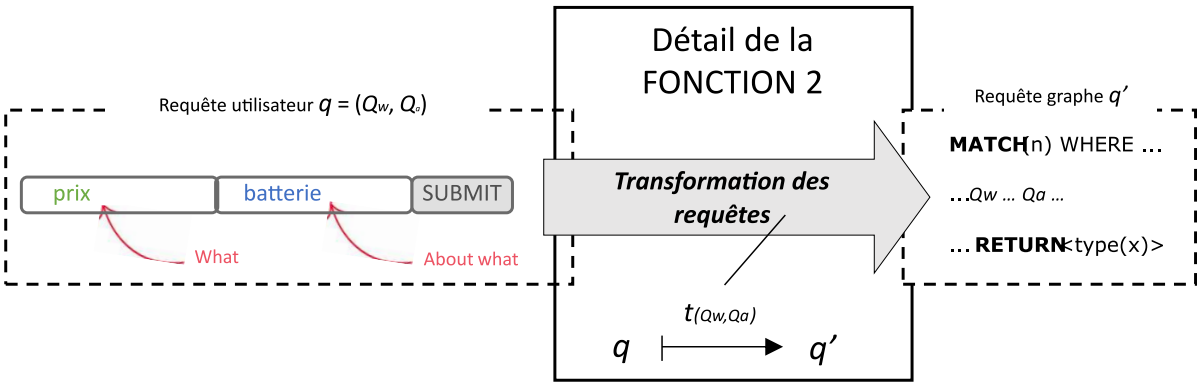


FIG. 4.6 : Illustration de la transformation des requêtes utilisateurs q en requêtes graphe q'

Comme l'illustre la Figure 4.6, les requêtes q sont transformées en requêtes graphe q' selon une fonction t nommée 'transformation des requêtes'. Cette transformation prend en compte les variables Q_w et Q_a expliquées dans la Section 4.5.1 et exprimant le besoin d'information. On considère plusieurs types de réponses possibles présentées en Section 4.5.2 afin d'aboutir à une requête en langage graphe expliquée dans la Section 4.5.3.

4.5.1 Expression des requêtes utilisateurs

L'analyse des requêtes exprimées pour le cas d'étude du drone nous permet de distinguer deux types de termes importants dans l'expression du besoin d'information. Il y a en effet les termes représentant le type d'élément que recherche l'utilisateur (soit le focus défini dans le Chapitre 2 à la Section 2.2.3) et les termes exprimant le sujet des éléments recherchés. Nous considérons donc par la suite la recherche d'information comme une expression des variables Q_w et Q_a soit respectivement le 'quoi' (w pour What) et le 'à propos de quoi' (a pour About what). La requête utilisateur q est alors définie par :

$$q = (Q_w, Q_a) \forall q \in Q \tag{4.3}$$

Dans les exemples de requêtes, les termes en **gras** représentent les Qw (quoi) et les termes en *italiques* représentent les Qa (à propos de quoi) :

Requête $\mathbf{q_1}$:

Besoin d'information : quels sont les **<documents>** mentionnant le terme '*<batterie>*' ?

Valorisation des variables : Qw = '-' et Qa = 'batterie'

Type de requête : (A)

Requête $\mathbf{q_5}$:

Besoin d'information : y a-t-il des **<brevets>** sur la *<fabrication additive>* des *pales* ?

Type de requête : (B)

Valorisation des variables : Qw = 'brevet' et Qa = ' "fabrication additive" '

Requête $\mathbf{q_{16}}$:

Besoin d'information : quel est le **<prix>** de la *<batterie 4S5200>* ?

Type de requête : (C)

Valorisation des variables : Qw = 'prix' et Qa = 'batterie 4S5200'

Requête $\mathbf{q_{23}}$:

Besoin d'information : quelles sont les **<exigences>** liées à la *<batterie>* ?

Type de requête : (D)

Valorisation des variables : Qw = 'exigences' et Qa = 'batterie'

La règle suivante est considérée : si plus d'un terme est entouré de guillemet, c'est l'expression exacte composée de ces termes qui est recherchée et non les termes séparément. Par exemple, la recherche de "fabrication additive" cherche à identifier les chaînes de caractère "fabrication additive" et non les éléments mentionnant "fabrication" et "additive" éloignés l'un de l'autre.

Une seconde règle considérée est celle d'étendre les termes dans Q_w et Q_a par leurs traductions anglaises et françaises.

4.5.2 Type de réponses attendu

L'analyse des différents ensembles de réponses pertinentes à chacune des requêtes du cas d'étude du drone nous permet de distinguer les trois types de réponses attendues suivants :

Réponse de type (I)

La liste de données (documents ou enregistrements en base de données) pour répondre à des requêtes telles que "Trouve les documents ..., les brevets, les simulations, etc.". Transposée à la modélisation graphe de la proposition, nous définissons ici les réponses de type (I) comme étant une **liste de sommets** pouvant être désignés par leurs identifiants uniques.

Réponse de type (II)

La liste de valeurs de propriété (valeur d'attribut ou valeur de métadonnée) pour répondre à des requêtes telles que "Trouve la quantité, le prix, le poids de ... ?". Les réponses peuvent alors ressembler à : "qté = 2", "prix = 46€" etc. Transposée à la modélisation graphe de la proposition, nous définissons ici les réponses de type (II) comme étant une **liste de valeurs de propriétés de sommets**.

Réponse de type (III)

La liste de phrases (contenue dans un attribut ou des métadonnées) pour répondre à des requêtes telles que : "quel est le prix de ... ? ", "quelle est l'exigence de ... ?". Les réponses peuvent alors ressembler à : "le prix de la batterie est de 26€." ou encore "la capacité de la batterie doit être proche de 10,9 Ah.". Transposée à la modélisation graphe de la proposition, nous définissons ici les réponses de type (III) comme étant une **liste de phrases contenues dans des valeurs de propriétés de sommets**.

4.5.3 Pré-traitement des requêtes : transformation en requêtes graphe

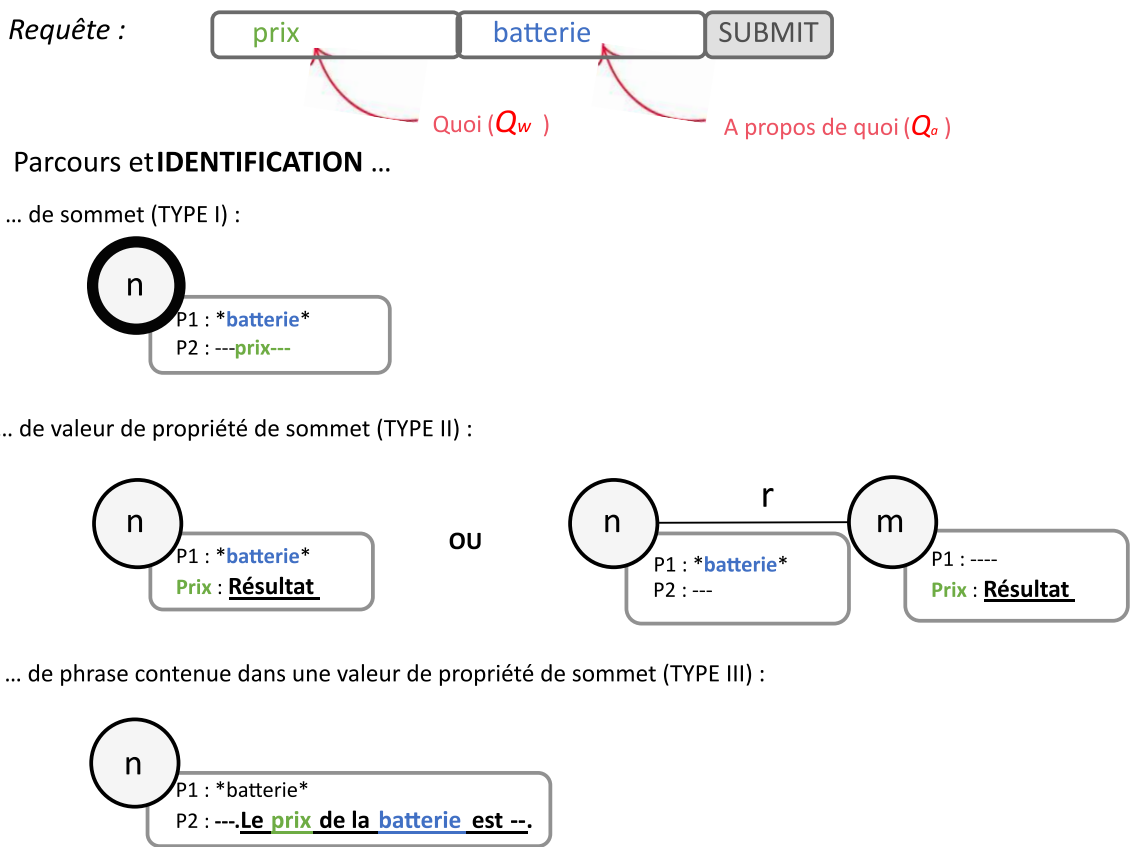


FIG. 4.7 : Illustration de la recherche graphe avec la requête $q = (Q_w =' prix', Q_a =' batterie')$

Les différents types de réponses sont présentés en gras dans la Figure 4.7 pour une requête q exprimée par $(Q_w =' prix', Q_a =' batterie')$.

Pour aboutir à ces types de réponses vis-à-vis des variables Q_w et Q_a, on applique les règles suivantes :

- Les réponses de type (I) (liste de sommets du graphe) sont obtenues en recherchant tous les sommets mentionnant Q_w et Q_a (et leurs traductions) dans au moins une valeur de leurs propriétés. Par exemple, si la requête est $Q_w =' -'$ et $Q_a =' batterie'$, la requête graphe cherche à identifier tous les sommets dont au moins une propriété mentionne le terme 'batterie' ou 'battery'.
- Les réponses de type (II) (liste de valeurs propriétés de sommets) sont obtenues en recherchant toutes les valeurs des propriétés nommées comme Q_w (ou sa traduction) si au moins une valeur des propriétés du même sommet ou d'un sommet voisin mentionne Q_a (ou sa traduction). Par exemple, si la requête est $Q_w =' prix'$ et $Q_a =' batterie'$, la requête graphe cherche à identifier tous les sommets dont une des propriétés mentionne 'batterie' ou 'battery' et une de ses autres propriétés ou propriétés d'un noeud voisin se nomme 'prix' ou 'price'. La requête renvoie alors la valeur de l'étiquette 'prix' ou 'price'.
- Les réponses de type (III) (phrases contenues dans une liste de valeurs propriétés de sommets) sont obtenues par l'identification des phrases contenant Q_w et Q_a (et leurs traductions) dans des valeurs propriétés. L'identification des phrases est faite grâce aux algorithmes de NLP (Hirschberg et al. 2015) permettant l'extraction d'information de contenu textuel. Par exemple, si la requête est $Q_w =' prix'$ et $Q_a =' batterie'$, la requête graphe cherche à identifier tous les sommets contenant dans une valeur de ses propriétés une phrase contenant 'prix' ou 'price' et 'batterie' ou 'battery'. La requête renvoie alors la phrase en réponse. Deux cas spécifiques sont ajoutés pour rechercher des phrases mentionnant les exigences et celles mentionnant des justifications de choix. Si Q_w = 'exigence' ou 'requirement', alors la requête recherche non pas le terme 'exigence' mais un ensemble de termes contenu dans un lexique propre à l'expression d'exigence (doit, devoir, requière, etc.) comme le suggère les travaux de (Pinquié et al. 2016). Dans la même logique, si Q_w = 'choix' ou 'choice', alors la requête recherche également des expressions équivalentes comme 'rejeté car', 'choisi parce que' etc.

Ainsi, on obtient pour les exemples q_1, q_5, q_{16} et q_{23} :

Requête q_1 :

Besoin d'information : quels sont les documents mentionnant le terme 'batterie' ?

Type de requête : (A)

Valorisation des variables : Qw = '-' et Qa = 'batterie'

Recherche d'éléments dans le graphe : tous les sommets mentionnant le terme 'batterie' (réponse de type (I)).

Requête q_5 :

Besoin d'information : y a-t-il des brevets sur la fabrication additive des pales ?

Type de requête : (B)

Valorisation des variables : Qw = 'brevet' et Qa = '"fabrication additive"'

Recherche d'éléments dans le graphe : tous les sommets mentionnant 'brevet' dans une de ses propriétés et "fabrication additive" dans une autre (réponse de type (I)).

Requête q_{16} :

Besoin d'information : quel est le prix de la batterie 4S5200 ?

Type de requête : (C)

Valorisation des variables : Qw = 'prix' et Qa = 'batterie 4S5200'

Recherche d'éléments dans le graphe : toutes les valeurs de propriété nommée 'prix' si le sommet ou un des sommets voisins mentionne également dans une de ses propriétés 'batterie' (réponse de type (II)) et toutes les phrases contenant le terme 'prix' et le terme 'batterie' - (réponse de type (III)).

Requête q_{23} :

Besoin d'information : quelles sont les exigences liées à la batterie ?

Type de requête : (D)

Valorisation des variables : Qw = 'exigences' et Qa = 'batterie'

Recherche d'éléments dans le graphe : toutes les phrases exprimant l'exigence grâce au lexique de verbes et modaux associés et mentionnant le terme 'batterie' (réponse de type (III)).

On note notamment les règles listées dans le Tableau 4.3 entre le type de requête exprimée par la valorisation de Q_w et Q_a et le type de réponses attendues. Ces règles nous permettent de filtrer le type de réponse (II) et (III) lorsque seul Q_a est non nulle afin d'éviter des réponses peu pertinentes pour le besoin d'information.

Tab. 4.3 : Règles de transformations entre la valorisation de Q_w et Q_a et le type de réponse attendue

Type de requête	Valorisation de Q_w et Q_a	Type de réponse
(A)	($Q_w = \emptyset$ et $Q_a \neq \emptyset$) ou ($Q_w \neq \emptyset$ et $Q_a \neq \emptyset$)	Type de réponse (I)
(B)	($Q_w \neq \emptyset$ et $Q_a \neq \emptyset$)	Type de réponse (I)
(C)	($Q_w \neq \emptyset$ et $Q_a \neq \emptyset$)	Type de réponse (II)
(D)	($Q_w \neq \emptyset$ et $Q_a \neq \emptyset$)	Type de réponse (III)

4.5.4 Les fonctions de transformation

La Figure 4.8 fourni un exemple de requête graphe attendue pour la requête q exprimée par $(Q_w =' prix', Q_a =' batterie')$. L'Algorithme 2 décrit en pseudo-code les différentes fonctions enclenchées par l'utilisateur lors de la soumission de la requête.

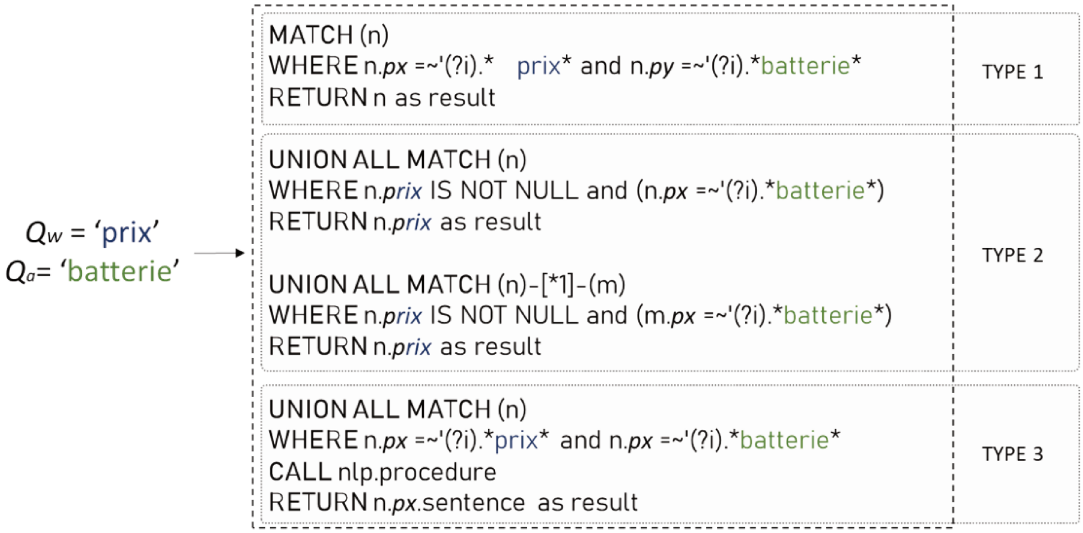


FIG. 4.8 : Illustration d'une requête graphe lorsque que Q_w = 'prix' et $Q_a =' batterie'$

Algorithm 2 : Génération des requêtes graphe

Input : Q_w, Q_a, G_d
Output : query

```
prop ← extraction des propriétés dans G_d
propAsQw ← prop nommé comme Q_w
if Q_a = '' then
    query1 ← génération (type1,Q_a,Q_w)
else
    query1 ← génération (type1,Q_a,Q_w)
    query2 ← génération (type2, Q_a,Q_w)
    query3 ← génération (type3, Q_a,Q_w)
query = query1 + query2 + query3
return query
```

Récupération de Q_w et Q_a : le besoin d'information de l'utilisateur étant exprimé par la valorisation de Q_w (le quoi) et de Q_a (le à propos de quoi), l'Algorithme 2 récupère ces valeurs au moment de la soumission de requête. Dans le cadre de l'étude, Q_w ne peut contenir qu'un mot-clé tandis que Q_a en accepte plus d'un comme dans la requête 2 et 3. L'information de guillemet dans Q_a est également indiquée comme expliqué dans la Section 4.5.1 : les termes doivent être recherchés ensembles.

Récupération des propriétés du graphe : la ligne 1 de l'Algorithme 2 récupère la liste des propriétés du graphe afin de pouvoir les interroger et en identifier des particulièrement intéressantes par la suite. Cette liste est récupérée avec la fonction "call db.propertyKeys".

Récupération des propriétés nommées comme Q_w : la ligne 2 de l'Algorithme 2 permet d'identifier dans la liste des propriétés du graphe celles nommées comme Q_w afin de venir les sélectionner dans les réponses de type (II).

Récupération des réponses de type (I) : les lignes 3 à 6 de l'Algorithme 2 permettent d'obtenir la requête graphe identifiant tous les documents mentionnant le terme Q_a. L'expression simplifiée de la recherche est :

> MATCH (n) WHERE $n.p_{(1\rightarrow i)}$ = *Q_w* AND $n.p_{(1\rightarrow i)}$ = *Q_a* RETURN n
>
> UNION ALL MATCH (n) -[:Contains*1]->(m) WHERE $n.p_{(1\rightarrow i)}$ = *Q_w* AND $m.p_{(1\rightarrow i)}$ = *Q_a* RETURN n

où i est le nombre des propriétés du sommet n.

Récupération des réponses de type (II) : la ligne 7 de l'Algorithme 2 permet d'obtenir la requête graphe identifiant toutes les valeurs de propriétés nommées comme Q_w notées $propAsQw$ et dont le sommet ou un sommet voisin mentionne le terme Q_a. L'expression simplifiée de la recherche est :

> MATCH (n) WHERE $n.propAsQw$ IS NOT NULL and $n.p_{(1\rightarrow i)}$ = *Q_a* RETURN $n.propAsQw$
>
> UNION ALL MATCH (n)-[*1]-(m) WHERE $m.propAsQw$ IS NOT NULL and $n.p_{(1\rightarrow j)}$ = *Q_a* RETURN $m.propAsQw$

où i et j sont respectivement les nombres de propriétés des sommets n et m.

Récupération des réponses de type (III) : la ligne 8 de l'Algorithme 2 permet d'obtenir la requête graphe identifiant toutes les phrases mentionnant Q_w et Q_a. Une première étape consiste à identifier toutes les valeurs des propriétés contenant Q_w et Q_a, de décomposer le contenu textuel concerné en sous-sommets adjacents étiquetés par chacune des phrases et de retourner uniquement les sous-sommets contenant Q_w et Q_a. De multiples autres méthodes d'analyse du langage naturel transposées en informations graphe pourraient être exploitées, mais nous nous limitons à la sous-décomposition des phrases pour la proposition actuelle. L'expression simplifiée de la recherche est :

> MATCH (n) WHERE $n.p_{(1\rightarrow i)}$ = *Q_w* and $n.p_{(1\rightarrow i)}$ = *Q_a* RETURN n.sentence

où i est le nombre de propriété du sommet n.

Concaténation de la requête : enfin, la ligne 9 de l'Algorithme 2 permet d'exprimer la recherche englobant l'ensemble des types de réponses recherchés. Afin de limiter les ressources nécessaires lors de la recherche, la proposition permet à l'utilisateur de préciser quel type de réponse il souhaite. Ce choix est opéré grâce à la valorisation binaire (O/N) des trois variables suivantes : *Document* pour les réponses de type (I), $Valeur$ pour les réponses de type (II) et *Phrase* pour les réponses de type (III). Cette valorisation est contrôlée par les règles suivantes : Si Q_w = '-' et $Q_a \neq$ '-', possibilité de ne sélectionner que *Document*, Si $Q_w \neq$ '-' et Q_a = '-', possibilité de ne sélectionner que $Valeur$.

4.6 Identifier les réponses (FONCTION 3)

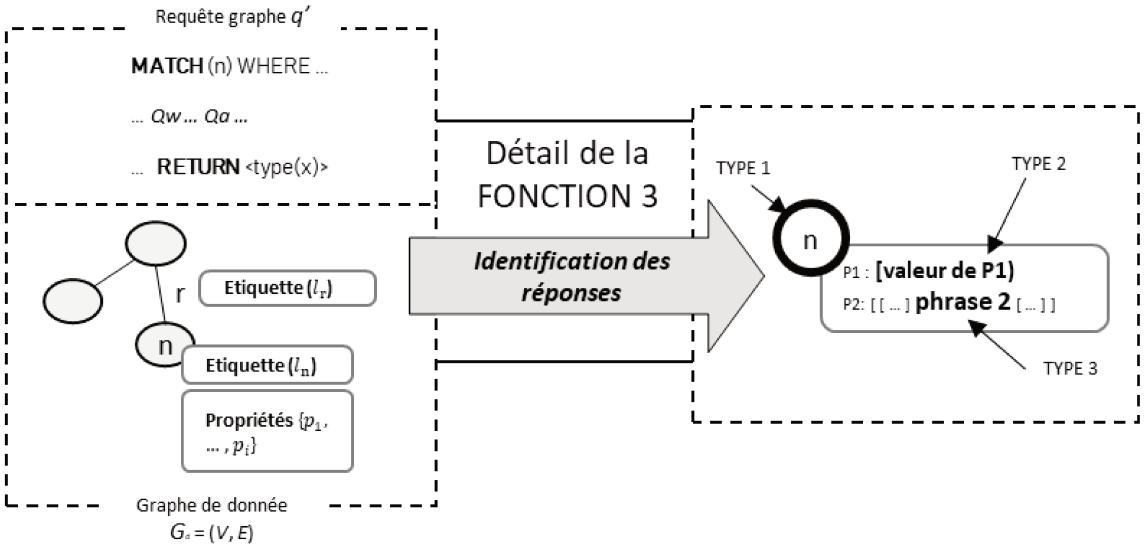


FIG. 4.9 : Illustration de l'identification des réponses

Comme le présente la Figure 4.9, la requête graphe est soumise au graphe des données G_d afin de retourner le sous-graphe qui y répond. Ce sous-graphe est alors transposé en liste de résultats affichée sous forme de tableau dans l'interface graphique. Le tableau est directement placé sous la requête émise par l'utilisateur lui permettant ainsi de ne pas changer de fenêtre.

- Dans le cas des réponses de type I, c'est-à-dire une liste de sommets, on affiche la liste des identifiants uniques des sommets concernés comme ci-dessous :

Identifiant
ID1
ID2
...

Dans l'exemple de la requête q_1 "Quels sont les documents mentionnant le terme 'batterie' ?", on obtiendrait par exemple les résultats comme affichés dans la Figure 4.10.

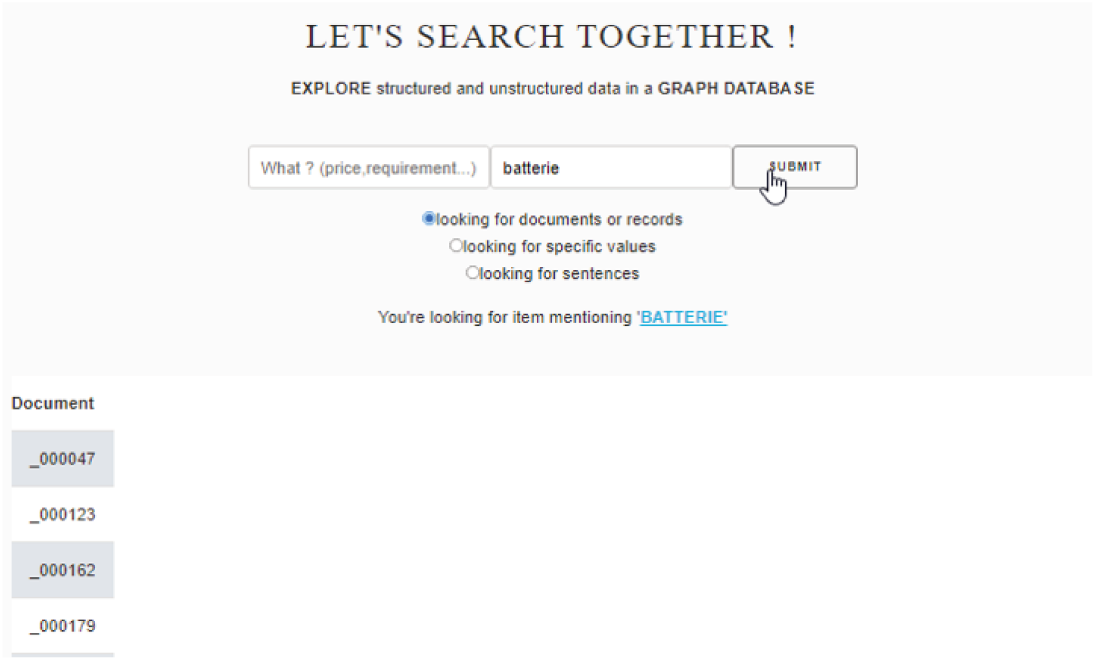


FIG. 4.10 : Exemple d'affichage de réponses de type I

- Dans le cas des réponses de type II, c'est-à-dire une liste des valeurs des propriétés p des sommets, on affiche à la fois l'identifiant du sommet portant p ainsi que la valeur de p en indiquant en nom de colonne son nom.

Identifiant	p
ID1	$v1_p$
ID2	$v2_p$
...	

Dans l'exemple de la requête q_{16} "Quel est le prix de la batterie 4S5200 ?", on obtiendrait par exemple le résultat comme affiché dans la Figure 4.11.

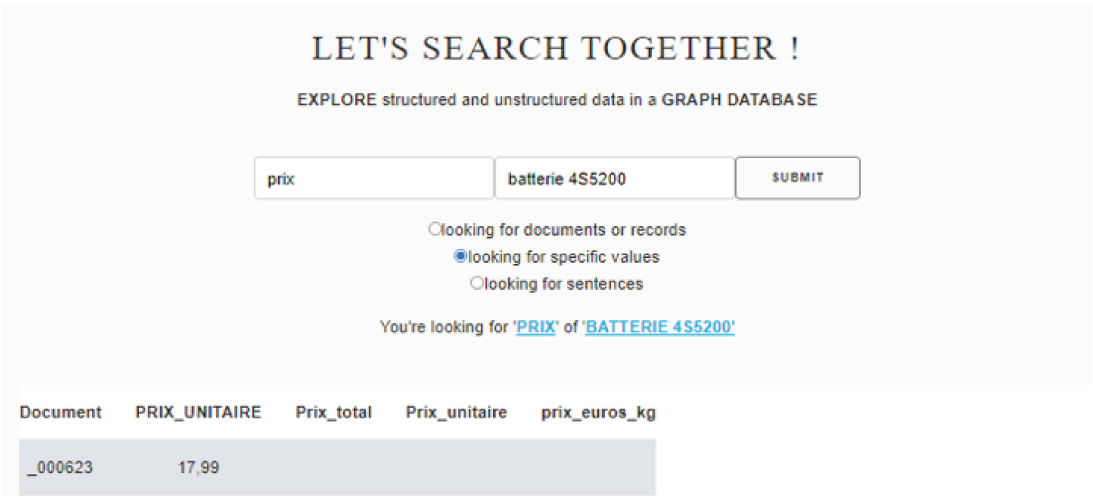


FIG. 4.11 : Exemple d'affichage de réponses de type II

- Dans le cas des réponses de type III, c'est-à-dire une phrase contenue dans une valeur des propriétés des sommets, on affiche l'identifiant du sommet portant la phrase ainsi que la phrase en tant que telle.

Identifiant	phrase
ID1	<phrase1>
ID2	<phrase2>
...	

Dans l'exemple de la requête q_{23} "Quelles sont les exigences liées à la batterie ?", obtiendrait par exemple le résultat comme affiché dans la Figure 4.12.

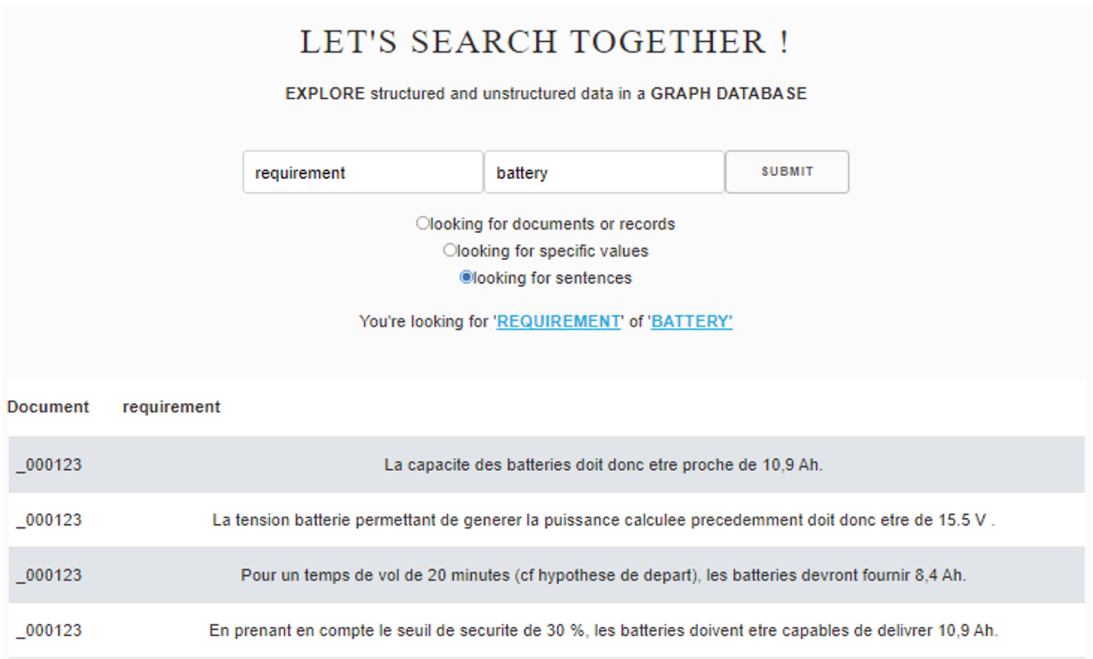


FIG. 4.12 : Exemple d'affichage de réponses de type III

4.7 Développement et application de la proposition

4.7.1 Développement de la proposition

Le développement de la proposition a fait l'objet d'une sélection de plusieurs outils. Le Tableau 4.4 récapitule les fonctions attendues et les outils qui ont été sélectionnés. Leurs descriptions détaillées suit le tableau.

TAB. 4.4 : Liste des outils sélectionnés pour le développement de la proposition

Fonction	Outils
Langage de développement	Python, HTML et CSS
Stockage et gestion graphe	Neo4J
Extraction de texte	Apache Tika et POI
Extraction de texte à partir d'image	Tesseract
Traitement du langage naturel	Standford CoreNLP

Python, HTML et CSS

Nous avons choisi le langage de programmation python[2] car intuitif et massivement utilisé, donc bien documenté. De plus, de nombreuses bibliothèques open source sont à disposition. Nous utilisons notamment la bibliothèque Py2no[3] permettant de communiquer avec Neo4J à partir d'un programme en langage python. Nous utilisons également l'api Translation[4] de google pour la recherche des traductions français/anglais des mots-clés recherchés. Enfin, nous avons utilisé le langage HTML et CSS afin de réaliser l'interface utilisateur de l'application dont l'aperçu se trouve par exemple à la Figure 4.10.

Neo4J

Pour rappel, Neo4J est choisi comme système de gestion des données en graphe car il est jugé, avec ArangoDB, comme celui qui offre les meilleures fonctionnalités, une grande simplicité de prise en main et une bonne puissance d'interrogation (FERNANDES et al. 2018). De plus, Neo4J dispose d'une grande communauté dont le partenaire industriel fait parti. La documentation sur le système Neo4J est riche et intègre le composant additionnel 'Neo4J ETL

[2] https://www.python.org/
[3] https://py2neo.org/2021.1/
[4] https ://cloud.google.com/translate/ ?hl=fr

Tools'[5] qui permet la transformation des données provenant des bases de données relationnelles. Neo4J a d'ailleurs été utilisé dans plusieurs travaux cités dans l'état de l'art (Schabus et al. 2017 ; Noel et al. 2016).

Apache Tika

La fondation Apache est une organisation à but non lucratif qui propose des solutions open source comme Tika[6] et POI[7]. Tika est choisi comme analyseur syntaxique du contenu textuel et des métadonnées des documents. Il traite des fichiers comme les .pdf, .ptt(x), .doc(x). Afin de traiter les fichiers de type tableurs comme les .xls(x), nous choisissons POI. Ce sont deux outils standards utilisés notamment dans des travaux cités en état de l'art (Alhabashneh et al. 2011).

Nota bene : une partie de la récupération d'informations provenant des tableaux depuis les documents textuels a été réalisée manuellement. En effet, si la détection des tableaux depuis des fichiers tableurs est réalisable grâce aux outils précédemment cités, la détection des tableaux (ainsi que leurs éléments d'en-tête) au sein de documents textuels ou images ne l'est pas. Ce point est une limite discutée dans le Chapitre 6 à la Section 6.5.2.

Tesseract

Dans le cas particulier du texte contenu dans les images, nous utilisons l'outil Tesseract[8], également sous licence apache, pour permettre l'extraction des caractères par reconnaissance optique.

Standford CoreNLP

Plusieurs regroupements d'algorithmes effectuent du traitement du langage naturel (StanfordCoreNLP, Natural Language Toolkit (NLTK), OpenNLPetc.). Nous avons sélectionné StanfordCoreNLP[9] pour sa capacité de détection de phrases (Boyer et al. 2018), fonctionnalité utilisée dans notre proposition, et pour l'utilisation du plug-in Neo4J[10]. Ce plug-in permet de lancer les procédures vues dans l'état de l'art en appliquant la sous-décomposition des textes en sous-sommets et arêtes étiquetables.

[5] https://neo4j.com/labs/etl-tool/
[6] https://tika.apache.org/
[7] https://poi.apache.org/
[8] https://opensource.google/projects/tesseract
[9] https://standfordnlp.github.io/CoreNLP/
[10] https ://github.com/graphaware/neo4j-nlp

4.7.2 Application de la proposition

Le jeu de données PAINT'R et les 25 requêtes listés en Section 3.5.2 ont étés appliqués à la proposition développée. Les listes de résultats obtenus à chaque requêtes ont été comparées à l'ensemble des résultats attendus. Pour rappel, cette comparaison permet d'évaluer la performance de la proposition quand à sa capacité de fournir tous les résultats (mesure de rappel) et uniquement de bons résultats (mesure de la précision). Une moyenne harmonique pondérée à 0.5 (la précision est deux fois plus importante que le rappel) est alors donnée pour estimer une performance globale de la proposition. Plus ces mesures se rapprochent de 1, plus elles indiquent que la proposition est performante.

Les résultats obtenus sont donnés dans le Tableau 4.5. Ces résultats sont également présentés dans la Figure 4.13. Dans ce graphique, l'objectif est d'atteindre les coordonnées (1:1) représentant un cas où l'ensemble des résultats pertinents et uniquement eux sont présentés à l'utilisateur. Le score avec la proposition actuelle est indiquée par la croix 'situation initiale' qui est le point à partir duquel nous souhaitons améliorer les résultats.

Les résultats montrent un pourcentage de réponses pertinentes affichées faible. En effet, le rappel exprime que seul 37% des résultats attendus apparaissent. La précision quant à elle est également faible indiquant que 57% des résultats affichés ne font pas partie des résultats pertinents. La moyenne $F_{\beta=0.5}$ vaut quant à elle 0,41.

Tab. 4.5 : Résultats d'évaluation de la proposition

	Score
Précision [0,1]	0.43
Rappel [0,1]	0.37
$F_{\beta=0.5}$ [0,1]	0.41

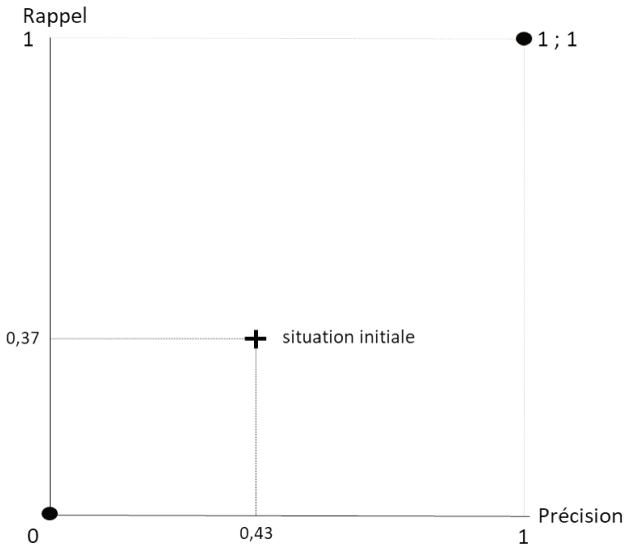


Fig. 4.13 : Score de rappel et de précision

4.7.3 Analyse des causes racines et pistes d'amélioration

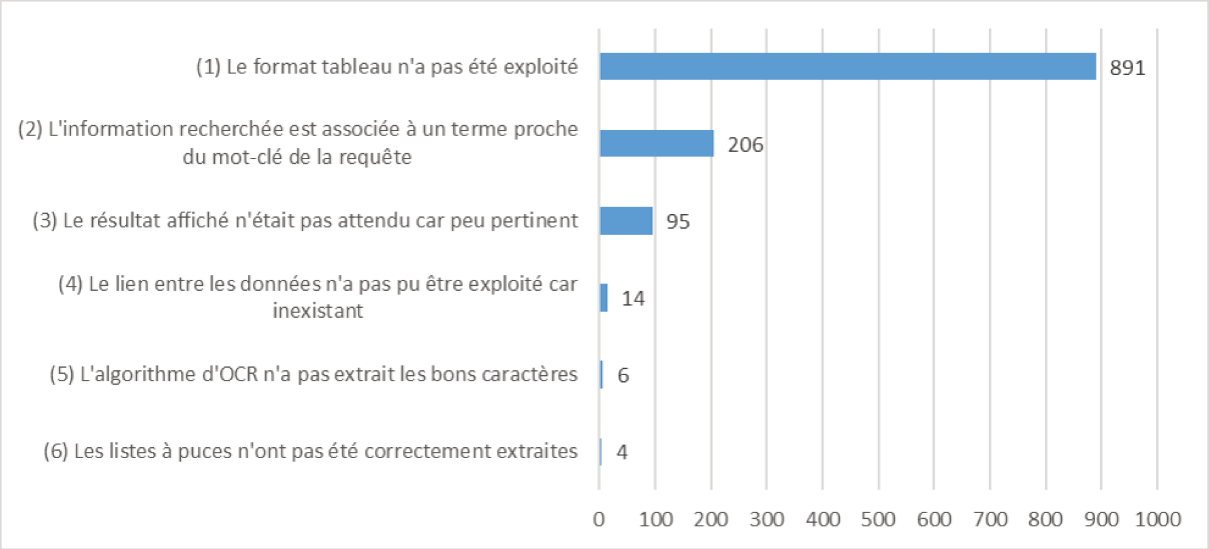


FIG. 4.14 : Distribution des causes identifiées après l'analyse des anomalies

Comme présenté dans le Chapitre 3, la démarche d'amélioration de la proposition nécessite une analyse des causes racines. La classification de chaque anomalie en une liste de causes est répertoriée dans la Figure 4.14.

Les résultats obtenus permettent d'identifier 6 causes impactant les performances de notre système. La cause '*(5) L'algorithme OCR*[11] *n'a pas extrait les bons caractères*' est due aux performances insuffisantes de l'outil de reconnaissance de caractères dans les images. Nous décidons de retirer cette cause de la liste des enjeux à résoudre, car elle ne concerne pas notre champ d'études. Nous proposons alors de définir pour les 5 causes restantes 4 grandes familles d'enjeux. Par ordre décroissant de nombre d'anomalies impactées, les 4 grandes familles sont :

Considérer les spécificités syntaxiques des données, notamment des tableaux (SYNT) : la cause (1) indique que certaines informations recherchées sont contenues dans des cellules spécifiques de tableurs que l'extraction textuelle en vrac ne suffit pas à identifier. Par exemple, la référence portée par une cellule d'une nomenclature sous un fichier excel n'est pas identifiable, car les informations de colonne et de ligne ne sont pas retranscrites dans le graphe. La cause (6) indique que l'extraction du contenu textuel n'identifie et ne traite pas les listes à puces ce qui génère de multiples concaténations de phrases et crée alors du bruit. Il semble donc nécessaire d'effectuer des traitements supplémentaires dans la transformation des données, notamment sur les tableaux et les listes à puces du contenu textuel.

Étendre sémantiquement les termes de la recherche (SEM) : la cause (2) indique que la recherche d'un terme strictement exact à celui de la recherche (et de sa traduction) n'est pas suffisante et qu'un rapprochement entre différents termes sémantiquement

[11] OCR pour Optical Character Recognition

proches est nécessaire. Par exemple, si le terme *reference* est utilisé dans la requête, le terme *Part Number* devrait également l'être ou encore, si l'on recherche les phrases d'exigences liées à la *batterie*, on souhaite également rechercher les phrases d'exigences mentionnant l'*autonomie*.

L'identification des résultats particulièrement pertinents (PERT) : le cadre de la proposition ne prévoit pas d'ordre d'affichage des résultats par pertinence et la cause (3) indique que des résultats inattendus (mais potentiellement pertinents) sont affichés de la même manière que les résultats attendus. Par exemple, la recherche de la *référence de la batterie* fournit de nombreux documents contenant ces termes bien qu'ils soient éloignés les uns des autres et n'apportent donc pas d'informations intéressantes à l'utilisateur.

La détection des liens implicites entre les données distribuées (LIEN) : les liens implicites entre les données ne sont pas exploitables. La cause (4) met en effet en évidence le cas où l'information recherchée aurait pu être identifiée si deux éléments liés implicitement l'avaient été explicitement. C'est le cas par exemple pour la recherche du **prix** d'une **référence** de batterie (q_{16}). Si l'utilisateur énonce une désignation fonctionnelle de la batterie, mais que le prix n'est associé qu'à une référence fournisseur différente de la désignation fonctionnelle, l'accès à l'information de prix est alors impossible.

4.7.4 Répartition des anomalies selon le type de requêtes

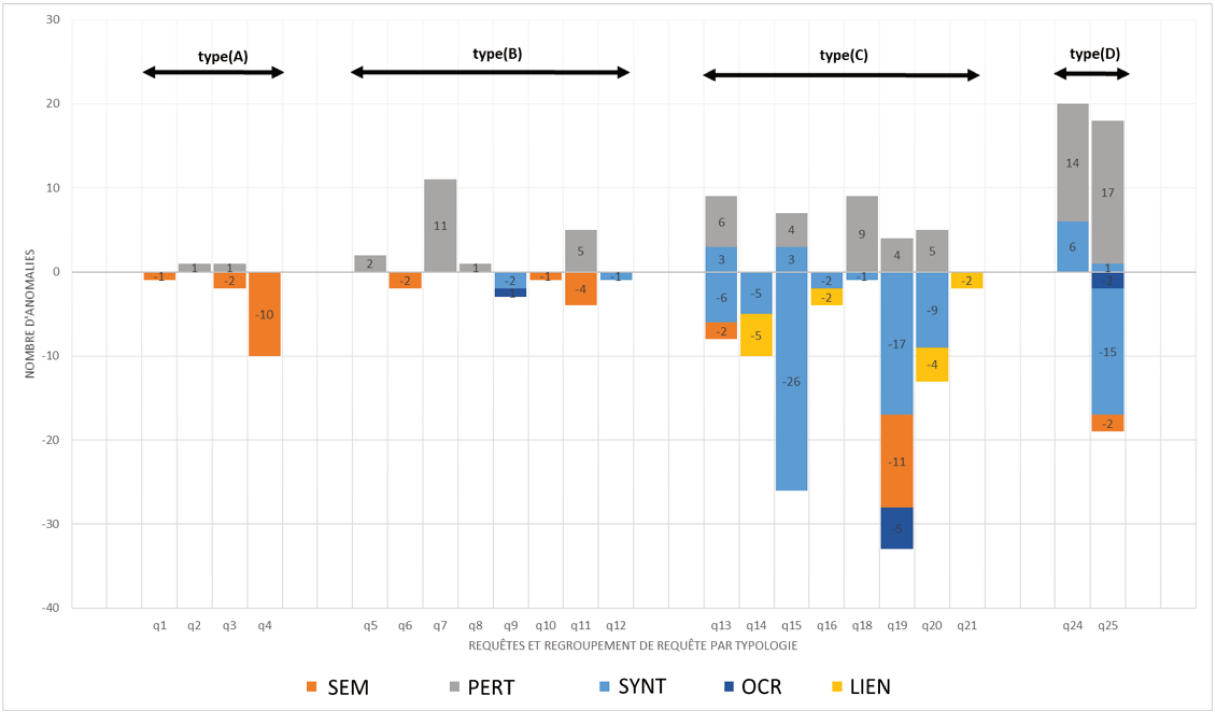


FIG. 4.15 : Répartition des anomalies selon le type des requêtes pour le jeu de données drone

La Figure 4.15 présente la distribution des anomalies par type de requête. A noter que les requêtes q_{17}, q_{22} et q_{23} ont étées exclues de la figure afin d'améliorer la visibilité du graphique. En effet, la requête q_{17} où Q_w = 'propriété' et Q_a= 'moteur' a un grand nombre de réponses pertinentes, mais représente plus de 150 anomalies liées au même problème de recherche sémantique : la recherche du terme "propriété" n'a pas trouvé les attributs nommés "prop", diminutif de propriété. La requête 23 où Q_w = 'vitesse' et Q_a = 'Serge Bernard' représente également un grand nombre de réponses pertinentes (somme des vitesses enregistrées par un capteur) mais représente plus de 700 anomalies liées au format des données d'entrées. La requête 22, où Q_w = 'symptômes', ne présente quant à elle aucune anomalie.

Afin de renforcer l'analyse sur la répartition des anomalies selon le type de requête, nous avons défini un score s. Ce score exprime la représentativité d'une anomalie (SYNT, SEM, PERT ou LIEN) pour un type de requête (A, B, C ou D). Nous lissons le score selon la répartition des requêtes par type de requête (le nombre de requêtes n'étant pas équitablement répartis selon leurs types). On exprime ainsi le score :

$$S_{i,j} = \frac{N_{i,j}}{1 - \frac{Q_i}{Q}} * \frac{1}{\sum_{i=1}^{4} \left(\frac{N_{i,j}}{1 - \frac{Q_i}{Q}} \right)}$$

où i est le type de requête considéré, j le type d'anomalie considéré, Q le nombre de requêtes total, Q_i le nombre de requêtes du type considéré et $N_{i,j}$ le nombre d'anomalies de type j pour le type de requête i.

Considérant l'ensemble des requêtes, cette fois-ci donc sans l'exclusion des autres requêtes, on obtient le tableau suivant :

Tab. 4.6 : Répartition des anomalies selon le type de requête

Type de requête \ Type d'anomalie	SYNT	SEM	PERT	LIEN
(A)	-	0.09	0.03	-
(B)	0.00	0.04	0.20	-
(C)	0.96	0.86	0.30	1.00
(D)	0.04	0.02	0.47	-

Grâce aux visuels apportés par la Figure 4.15 ainsi qu'aux valeurs numériques obtenues dans le Tableau 4.6 nous notons que :

- les anomalies de sémantique et de pertinence impactent l'ensemble des types de requêtes. La résolution des enjeux associés devrait donc améliorer tous les types de requêtes soumises à la proposition.
- les anomalies concernant la syntaxe sont majoritaires et impactent principalement les requêtes de type (C) 'recherche de valeur' mais non exclusivement. Ce constat

est renforcé par la requête 23 exclue du graphique. La résolution de l'enjeu associé doit donc prendre en compte ce type de requête pour résoudre un grand nombre d'anomalies.

- les anomalies concernant la sémantique sont également nombreuses, principalement à cause de la requête 22 de type (C) 'recherche de valeur'. Sans ce cas, la répartition est néanmoins homogène entre les différents types de requêtes.
- enfin, seules les requêtes de type (C) 'recherche de valeur' sont concernées par les anomalies de liens implicites. C'est en effet un type de requête qui suppose le parcours de liens.

4.8 Synthèse

Afin de répondre à la question de recherche "Comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?", nous proposons i-Dataquest, un Système de Recherche d'Information utilisant un modèle de donnée graphe pour unifier les données sources. Nous avons notamment présenté l'architecture générale de la proposition, le modèle de donnée graphe considéré et les différentes règles de transformations permettant d'unifier les données hétérogènes sous ce seul modèle de donnée (FONCTION 1). L'expression du besoin d'information utilisé dans la proposition est également décrite ainsi que le type de réponses attendu et les règles de transformation du besoin exprimé par l'utilisateur en requête graphe (FONCTION 2). L'expression du besoin est notamment valorisée par le 'quoi' et le 'à propos de quoi' recherchés ce qui permet de distinguer trois types de réponses possibles : le 'document', la 'valeur' d'une propriété ou la 'phrase'. On décrit également les résultats affichés à l'utilisateur suite à la soumission de la requête aux graphe de donnée (FONCTION 3). Enfin, nous avons appliqué la proposition au cas d'étude du PAINT'R ce qui nous a permis d'établir que la proposition permet de restituer 37% des résultats pertinents et permet de présenter sur l'ensemble des résultats restitués 43% de résultats pertinents. La moyenne de ces deux critères évaluée avec la F-Mesure est alors de 0,41. Les résultats obtenus étant éloignés de la situation idéale (tous les résultats pertinents et aucun résultats non pertinents restituées), il est nécessaire de traiter les enjeux clés identifiés grâce à l'analyse des anomalies. Ainsi, il est nécessaire de traiter les quatre enjeux clés listés ci-dessous :

Les quatre enjeux clés à considérer sont :

- le traitement des spécificités syntaxiques des données (ENJEU 1),
- l'extension sémantique des termes de la recherche (ENJEU 2),
- le traitement des résultats peu et particulièrement pertinents (ENJEU 3),
- la détection de liens implicites entre des données distribuées (ENJEU 4).

Le traitement de ces quatre enjeux dans la proposition de système de recherche d'information utilisant un graphe de donnée pourrait alors nous conduire à une proposition performante pour répondre à la question de recherche. La description de la proposition sera détaillée au Chapitre 5.

Chapitre 5

Enrichissement de la proposition

5.1 Objectifs du chapitre

La proposition i-Dataquest a été présentée dans le précédent chapitre. Nous y avons également identifié quatre enjeux clés à traiter afin que cette proposition puisse répondre de manière performante à la question de recherche, à savoir :

- le traitement des spécificités syntaxiques des données (ENJEU 1),
- l'extension sémantique des termes de la recherche (ENJEU 2),
- le traitement des résultats peu et particulièrement pertinents (ENJEU 3),
- la détection de liens implicites entre les données distribuées (ENJEU 4).

Comme illustré dans la Figure 5.1, l'objectif de ce chapitre est donc de présenter les différentes propositions d'optimisations du système i-Dataquest afin de répondre aux différents enjeux. Chacune de ces propositions est évaluée suivant le processus présenté dans le Chapitre 3 et utilisé dans l'application de la proposition dans le Chapitre 4.

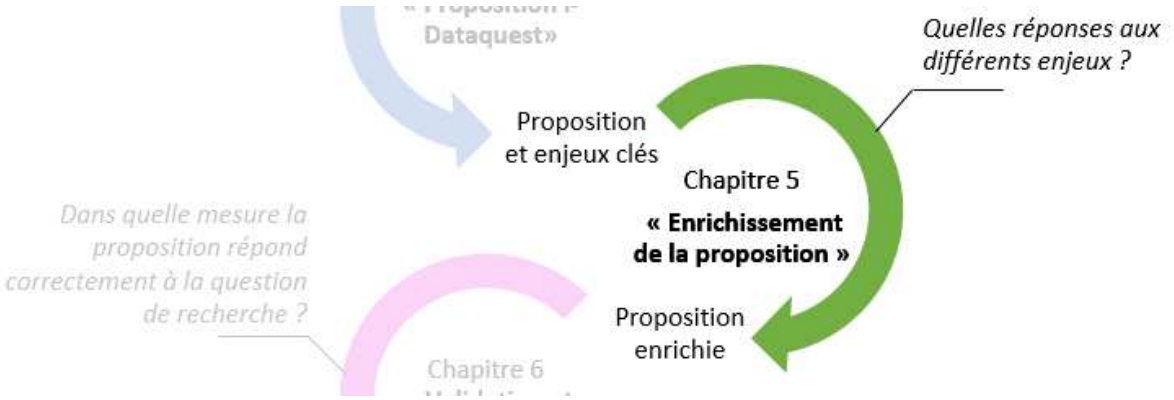


Fig. 5.1 : Organisation du mémoire : cinquième chapitre

5.2 Traitement des spécificités syntaxiques des données (ENJEU 1)

Selon l'analyse des anomalies rencontrées lors de l'application de PAINT'R à la proposition vue en Chapitre 4, deux types d'éléments sont en causes. Le premier est celui des tableaux dont les valeurs contenues dans des cellules ne sont pas identifiables par la proposition. Le second est celui des listes à puces contenues dans des documents textuels qui ne permet pas d'identifier certaines phrases recherchées. Ces deux types d'éléments font l'office de propositions disjointes que nous présentons ci-après.

Considérer les spécificités syntaxiques des données, notamment des tableaux (SYNT) : la cause (1) indique que certaines informations recherchées sont contenues dans des cellules spécifiques de tableurs que l'extraction textuelle en vrac ne suffit pas à

identifier. Par exemple, la référence portée par une cellule d'une nomenclature sous un fichier excel n'est pas identifiable, car les informations de colonne et de ligne ne sont pas retranscrites dans le graphe. La cause (6) indique que l'extraction du contenu textuel n'identifie et ne traite pas les listes à puces ce qui génère de multiples concaténations de phrases et crée alors du bruit. Il semble donc nécessaire d'effectuer des traitements supplémentaires dans la transformation des données, notamment sur les tableaux et les listes à puces du contenu textuel.

5.2.1 Traitement des tableaux

Afin d'identifier les valeurs recherchées (requête de type (C)) contenues dans des cellules spécifiques de tableaux présents dans les documents tableurs (.xls, .csv etc.) ou des documents textuels (.pdf, .ppt etc.), nous proposons d'enrichir les règles de transformation des données non structurées pour modéliser les tableaux en sous-graphes. Ainsi, chaque tableau de ces documents n'intégrera donc pas la propriété 'contents' comme précédemment, mais sera décomposé pour chacune de ses lignes en sommet adjacent au sommet représentant le document. Chaque propriété du sommet ainsi créé représente les colonnes du tableau et son étiquette reprend le nom du tableau (nom de classeur dans le cas d'un document tableur ou la légende dans le cas d'un document textuel). Finalement, les règles de transformations des données sources s'enrichissent comme présenté par les éléments en gras dans le Tableau 5.1. L'algorithme de transformation des données devient l'Algorithme 3 avec l'ajout des lignes en bleues.

		sommet $N_\circ \xrightarrow{f} N$					arête $R_\circ \xrightarrow{g} R$	
	Données sources	n	l_n	Nom (p_i)	Valeur (p_i)	Valeur ($p_{content}$)	r	l_r
	Base relationnelle	enregistrement	nom de table	nom d'attribut	valeur d'attribut	-	clé étrangère	:in_relation_with
①	Fichier (hors tableaux)	fichier	propriété 'type' du fichier	nom de propriété	valeur de propriété	**contenu textuel hors tableau**	-	-
②	**Tableaux dans fichier**	**ligne**	**nom de l'onglet ou légende**	**nom de colonne**	**valeur de cellule**	-	**Avec $n_{fichier}$**	**:contains**

Tab. 5.1 : Règles de transformation enrichies des données structurées et non structurées en modèle de donnée orienté graphe

Algorithm 3 : Génération du graphe de données

Input : *table*, *enregistrement*, *fichier*

```
1  foreach table do
2      foreach enregistrement do
3          CREATE n = (:table{ attribut : valeur_attribut})
4  foreach table do
5      foreach cleEtrangere do
6          foreach enregistrement do
7              CREATE (n :table)-[:in_relation_with]->(m :cleEtrangere)
8  foreach fichier do
9      texte ← extaction du texte du fichier
10     tableau ← extaction des tableaux du fichier
11     CREATE n = (:type{metadonnee : valeur_metadonnee,content : texte})
12     foreach tableau do
13         foreach ligne do
14             CREATE m = (:nom_tableau { colonne : valeur_colonne})
15         CREATE (n)-[:contains]->(m)
```

Extraction du contenu tableau des sommets n_1 représentant les documents non structurés : la ligne 10 de l'Algorithme 3, intégrée dans la ligne ① de la Figure 4.2, précise que le contenu textuel des tableaux n'est pas intégré à la propriété 'contents' du sommet représentant le document.

Création des sommets n_2 représentant les lignes des tableaux des fichiers non structurés : les lignes 12 à 15 de l'Algorithme 3, représentées par la ligne ② de la Figure 4.2, génèrent pour chaque ligne de tableau présent dans le contenu textuel un nouveau sommet $n2$. L'étiquette des nouveaux sommets $n2$ est valorisée par le nom du tableau. Ce nom est récupéré soit de la légende du tableau soit du nom de l'onglet dans le cas d'un fichier tableur. Les propriétés de $n2$ sont renseignées avec le nom et les valeurs des colonnes de la ligne du tableau. L'identification des tableaux et de sa structure s'appuie sur des règles d'identification. Par exemple, le tableau doit présenter une ligne d'en-tête identifiable par sa forme (bordure et/ou surlignage etc.).

Création des arêtes r entre les sommets n_1 (documents) et les sommets n_2 (lignes des tableaux) : les lignes 14 et 15 de l'Algorithme 3, représentées par la seconde partie de la ligne ② de la Figure 4.2, permettent la création de la relation entre les sommets n_2 et le sommet parent n_1 sous l'étiquette nommée 'contains'.

5.2.2 Traitement des listes à puces

Afin d'identifier les phrases contenues dans des listes à puces qui sont scindées en plusieurs parties, une opération de reconstruction des phrases est réalisée. En effet, la reconnaissance des phrases par les outils de NLP s'effectuant notamment avec l'analyse de la syntaxe et la présence du caractère « . », c'est l'ensemble de la liste à puces qui est extraite ligne par ligne.

Par exemple, la liste à puces suivante (avant) :

La batterie :

- doit permettre une autonomie du drone de 5h,
- est acheté à un prestataire.

devient alors (après) :

La batterie doit permettre une autonomie du drone de 5h. La batterie est acheté à un prestataire.

Dans cet exemple et dans la situation avant proposition d'enrichissement, la recherche des exigences liées à la batterie fournissait en réponse : "La batterie : doit permettre une autonomie du drone de 5h, est acheté à un prestataire." ce qui ne corresponds pas au résultat attendu. Après le traitement des listes à puces, nous obtenons en réponse : "La batterie doit permettre une autonomie du drone de 5h", ce qui est la réponse attendue.

5.2.3 Application au jeu de données

Les opérations de traitement des tableaux et des listes à puces ont étés réalisées manuellement sur le jeu de données du cas d'étude PAINT'R. Les résultats obtenus avant et après l'intégration de ces nouvelles règles sont présentées dans le Tableau 5.2 ainsi que la Figure 5.2. L'évolution du rappel et de la précision se visualise à l'aide du vecteur annoté "avec traitement syntaxique".

Tab. 5.2 : Résultats d'évaluation de la proposition avant et après le traitement de l'ENJEU 1

	Score précédent	Nouveau score
Précision [0,1]	0.43	0.58
Rappel [0,1]	0.37	0.66
$F_{\beta=0.5}$ [0,1]	0.41	0.61

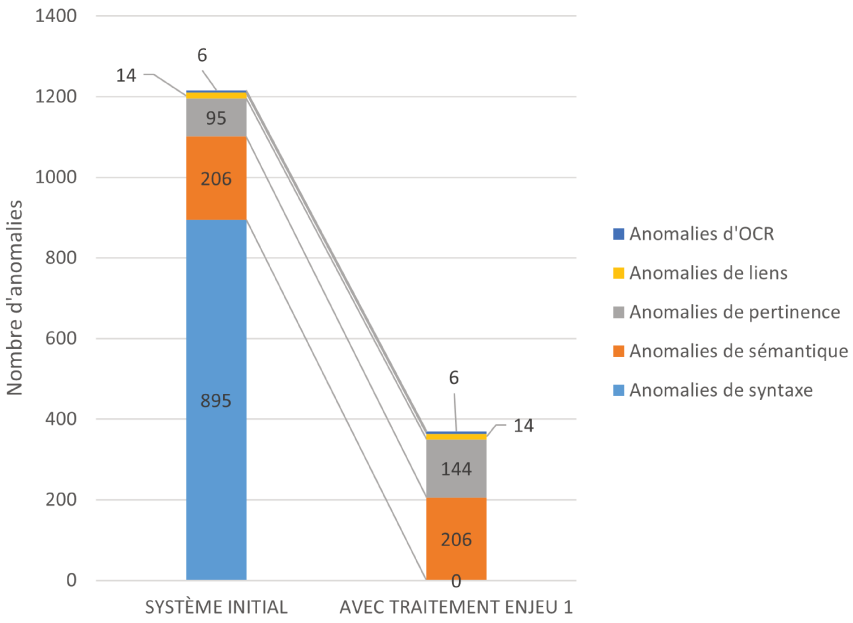


FIG. 5.3 : Répartition des anomalies avant et après traitement de l'ENJEU 1

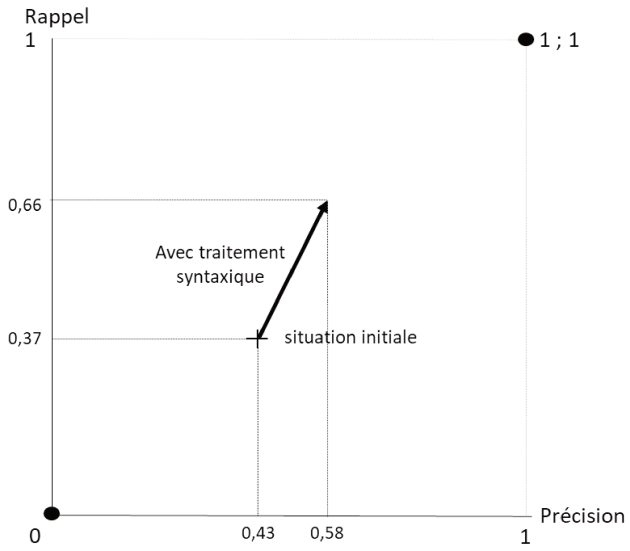


FIG. 5.2 : Score de rappel et de précision après le traitement de l'ENJEU 1

Nous obtenons une nette amélioration de la performance de la proposition visible à la fois sur la précision et le rappel. Désormais, le système permet de restituer 66% des résultats pertinents (gagnant ainsi 29 points de pourcentage[1] par rapport à la proposition initiale) et présente sur l'ensemble des résultats restitués 58% de résultats pertinents (gagnant ainsi 15 points par rapport à la proposition initiale). Sur la moyenne des deux mesures, le nouveau score de F-Mesure est de 0.61 contre 0.41 avec la proposition initiale, restant encore éloigné de 39 points de la cible idéale à savoir F-Mesure = 1.

Concernant la répartition des anomalies restantes, la Figure 5.3 montre la résolu-

[1]Point de pourcentage désigne la différence arithmétique entre deux pourcentages

tion totale des anomalies de type syntaxe avec néanmoins une augmentation du nombre d'anomalies liée à la pertinence des résultats affichés. En effet, le score de rappel (soit le nombre de résultats restitués) ayant augmenté, le nombre de résultats potentiellement moins pertinents dans ces nouveaux résultats augmente également.

5.3 Extension sémantique des termes de la recherche (ENJEU 2)

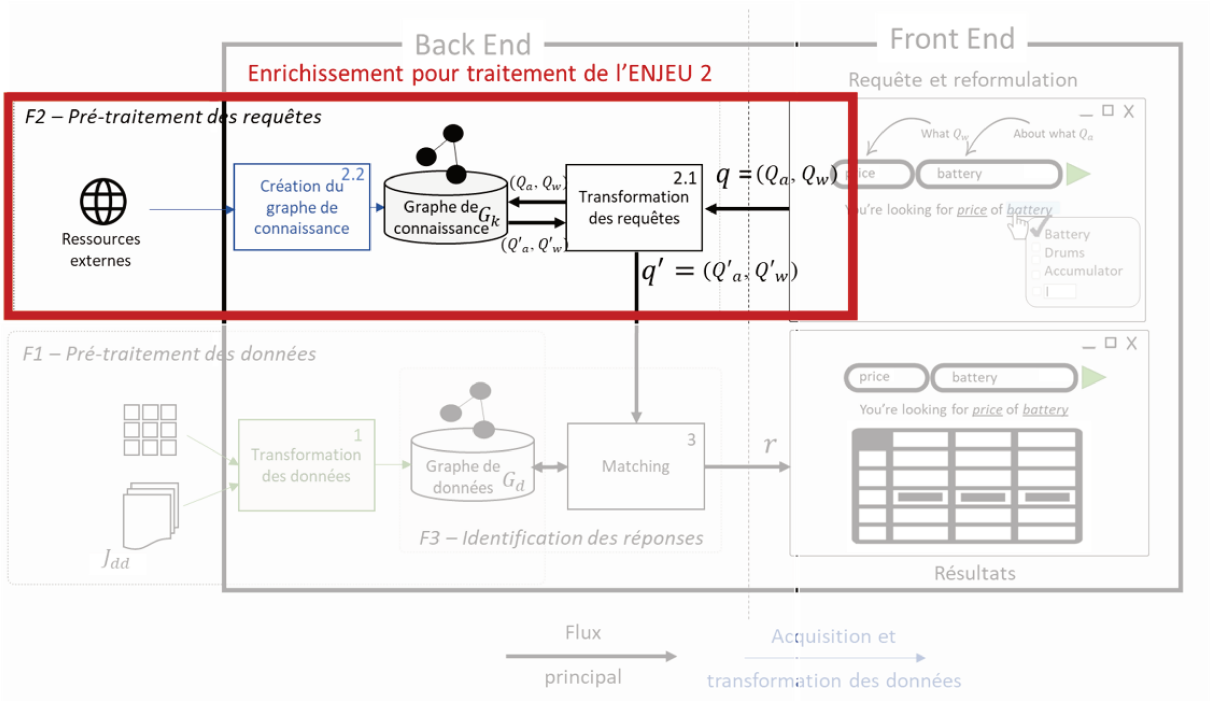
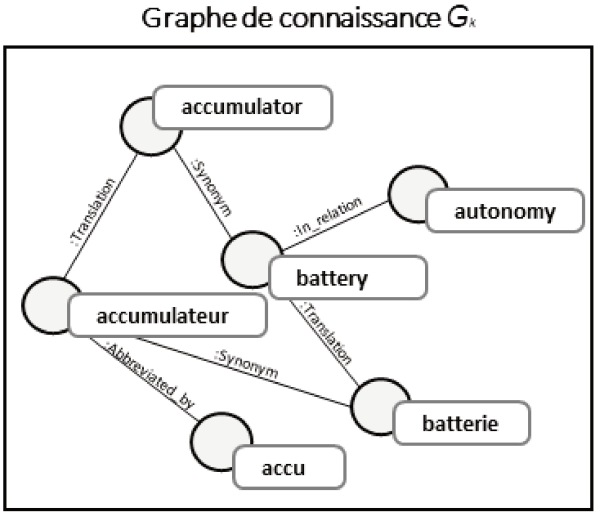


FIG. 5.4 : Transformation de requête par le graphe de connaissance G_k

Comme l'illustre la Figure 5.4, l'extension sémantique des termes de la requête est proposée par l'interrogation d'un graphe de connaissance générant ainsi à partir de la requête $q = (Q_w,Q_a)$ la requête étendue $q' = (Q'_w,Q'_a)$. Dans la suite de la section, nous présentons le graphe de connaissance, l'opération d'extension des termes de la requête, la génération du graphe de connaissance et enfin son enrichissement au fil du temps.

5.3.1 Le graphe de connaissance G_k



Propriétés

name	*language*	*weight*
battery	en	1
batterie	fr	1
accumulator	en	1
accumulateur	fr	1
accu	fr	1
autonomy	en	1

FIG. 5.5 : Illustration du graphe de connaissance

Le graphe de connaissance est représenté par la Figure 5.5. C'est un multigraphe orienté et pondéré aux sommets. Les sommets du graphe portent trois propriétés : la propriété 'name' contenant le mot, la propriété 'language' contenant la langue du mot comprise dans la liste *langue* (pour l'étude, on considère le français et l'anglais) et la propriété 'weight' contenant la pondération w du terme en fonction de sa fréquence d'utilisation. La valeur de 'weight' est un entier relatif compris entre [-100;100]. Les arêtes du graphe représentent les relations entre les termes, ces arêtes sont étiquetées en conséquence pouvant prendre quatre valeurs de *type* : 'Synonym' (les deux termes sont synonymes), 'Translation' (le terme se traduit par l'autre et inversement), 'Abbreviated_by' (le terme est une abréviation d'un autre) et 'In_relation' (les deux termes sont autrement liés par ex : contexte, type, fonctions d'utilisation, etc.). L'orientation des arêtes n'est à considérer que dans le cas des 'Abbreviated_by'. Selon ces principes, le graphe de connaissance s'écrit alors par l'expression du couple 'sommet/arête' (n, r) où le sommet est détaillé par son étiquette KG pour 'Knowledge Graph' puis ses propriétés 'label', 'langue' et 'weight' et l'arête est détaillée par son étiquette *type* et les sommets n et m qu'il permet de lier. En langage CYPHER (Section 2.3.2), l'expression donne :

$$G_k = (n, r) \leftrightarrow (: KG\{label : mot, language : langue, weight : w\}, (n) - [: type] \rightarrow (m)) \tag{5.1}$$

5.3.2 Exploitation du graphe de connaissance

Algorithm 4 : Extension des mots-clés

Input : Q, G_k
Output : Q'

1 $type$ = ['Synonym', 'Translation', 'In_relation', 'Abbreviated_by']
2 $labelDirect \leftarrow$ voisins de Q dans G_k
3 $labelIndirect1 \leftarrow$ 'Synonym', 'Abbreviated_by' et 'In_relation' de 'Translation' dans G_k
4 $labelIndirect2 \leftarrow$ 'Translation' et 'Abbreviated_by' de 'Synonym' dans G_k
5 $Q' = labelDirect + labelIndirect1 + labelIndirect2$
6 **return** Q'

Les variables Q_w et Q_a sont étendus vers Q'_w et Q'_a grâce aux opérations présentées dans l'Algorithme 4.

Récupération des termes de distance 1 : la ligne 2 de l'Algorithme 4 déclenche la récupération de tous les termes voisins, c'est-à-dire tous les sommets voisins au sommet dont la propriété 'name' équivaut au terme de la requête. On ne récupère les sommets 'Synonym', 'Translation' et 'Abbreviated_by' que si leurs poids sont supérieurs à 0. Pour les sommets 'In_relation', l'extension n'est pas automatique mais seulement à la demande de l'utilisateur (voir Section 5.3.5). Avec $mot = Q_w$ ou un des mots de Q_a à étendre, l'exploration du graphe de connaissance s'exprime par :

MATCH (n)-[*1]-(m) WHERE $n.label = mot$ and $m.weight > 0$ RETURN $m.label$

Récupération des termes de distance 2 liés aux traductions : la ligne 3 de l'Algorithme 4 déclenche la récupération de tous les termes voisins aux traductions du *mot* initial à l'exception des nouvelles traductions. Ici aussi, les poids des termes doivent être supérieurs à 0 sauf dans le cas des termes 'In_relation' que l'on récupère uniquement à la demande de l'utilisateur. La requête graphe s'exprime ainsi :

MATCH (n)-[:Translation*1]-(o)-[:Synonym*1]-(m) WHERE $n.label = mot$ and $m.weight > 0$ RETURN $m.label$

UNION ALL MATCH (n)-[:Translation*1]-(o)-[:Abbreviated_by*1]-(m) WHERE $n.label = mot$ and $m.weight > 0$ RETURN $m.label$

UNION ALL MATCH (n)-[:Translation*1]-(o)-[:In_relation*1]-(m) WHERE $n.label = mot$ and $m.weight > 0$ RETURN $m.label$

Récupération des termes de distance 2 liés aux synonymes : la ligne 4 de l'Algorithme 4 déclenche la récupération des traductions et abréviations de tous les synonymes du *mot* initial. La requête graphe s'exprime ainsi :

MATCH (n)-[:Synonym*1]-(o)-[:Translation*1]-(m) WHERE $n.label = mot$ and $m.weight > 0$ RETURN $m.label$

UNION ALL MATCH (n)-[:Synonym*1]-(o)-[:Abbreviated_by*1]-(m) WHERE $n.label = mot$ and $m.weight > 0$ RETURN $m.label$

Application à la requête q_3 : en reprenant la requête 3, nous détaillons ci-dessous les termes étendus de Qw et Qa. La représentation de la requête graphe q' est donnée dans la Figure 5.6.

Besoin d'information : Quel est le prix de la batterie 4S5200 ?

Valorisation des variables : Qw = 'prix' et Qa = 'batterie'

Termes étendus $Q'w$ = 'prix', 'price', 'coût' ...

Termes étendus $Q'a$ = '4S5200', 'batterie', 'battery', 'accumulateur', 'accu' ...

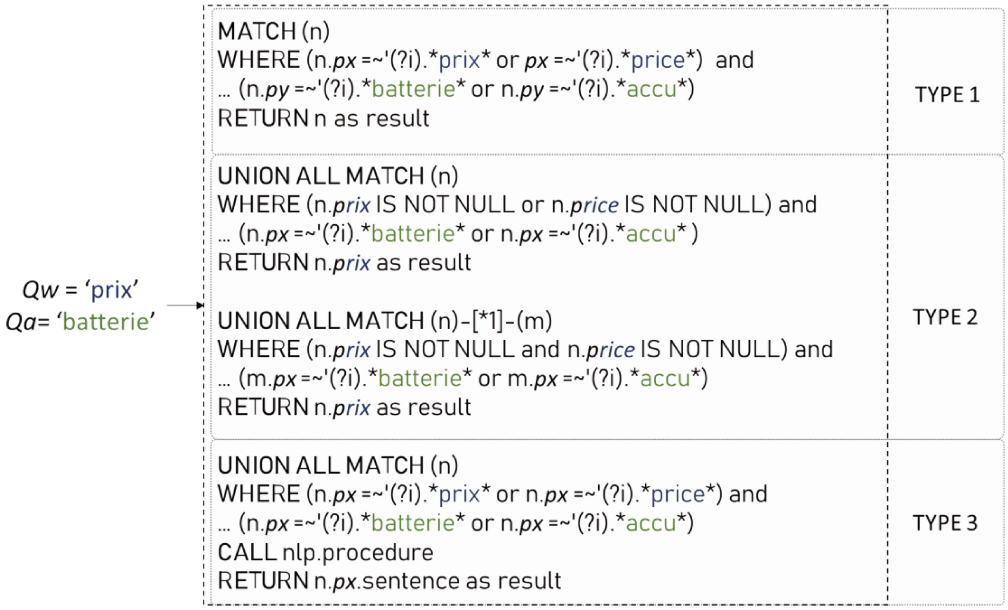


FIG. 5.6 : Illustration simplifiée d'une requête graphe lorsque que Q_w = 'prix' et Q_a = 'batterie'

5.3.3 Extension des requêtes graphe

Les termes de la requête ainsi étendus doivent intervenir dans la création de la requête graphe. Ainsi l'Algorithme 5 de génération des requêtes graphe intègre les deux nouvelles lignes 2 et 3 en bleues.

Algorithm 5 : Génération des requêtes graphe

Input : Q_w, Q_a, G_d
Output : query

1 $prop \leftarrow$ extraction des propriétés dans G_d
2 $Q'_w \leftarrow$ algorithme4(Q_w))
3 $Q'_a \leftarrow$ algorithme4(Q_a))
4 $propAsQw \leftarrow prop$ nommé comme Q'_w
5 **if** Q_a = " **then**
6 query1 $\leftarrow$ génération (type1,Q'_a,Q'_w)
7 **else**
8 query1 $\leftarrow$ génération (type1,Q'_a,Q'_w)
9 query2 $\leftarrow$ génération (type2, Q'_a,Q'_w)
10 query3 $\leftarrow$ génération (type3, Q'_a,Q'_w)
11 query = query1 + query2 + query3
12 **return** query

5.3.4 Génération du graphe de connaissance

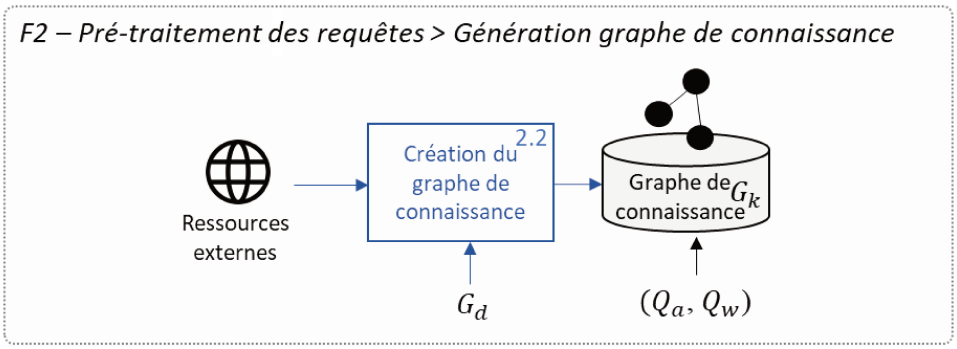


FIG. 5.7 : Illustration de la génération du graphe des données

Algorithm 6 : Génération du graphe de connaissance

```
   Input : G_d
 1 mot ← tf-idf(extraction des mots dans G_d)
 2 langue = ['français','anglais']
 3 french ← termes du dictionnaire de français
 4 english ← termes du dictionnaire d'anglais
 5 type = ['Synonym', 'Translation', 'In_relation']
 6 foreach mot do
 7     foreach langue do
 8         if mot in langue then
 9             CREATE n = (:Gk{ label :mot, language :langue, weight :0})
10         else
11             pass
12         foreach type do
13             mot' ← requests.get(langue,mots,type)
14             foreach mot' do
15                 if type = Synonym et mot commence avec mot' then
16                     type = 'Abbreviated_by'
17                 else
18                     pass
19                 CREATE m = (:Gk{ label :mot', language :langue,weight :0})
20                 CREATE (n)-[:type]->(m)
```

Le graphe G_k est généré suivant une suite d'opération présentée dans l'Algorithme 6 et illustrée dans la Figure 5.7.

Élimination des termes inutiles : afin de limiter la présence de termes inutiles pour l'entreprise, la ligne 1 de l'Algorithme 6 récupère l'ensemble des termes dans le contenu textuel des données soumis à une vectorisation TF-IDF et une élimination des mots vides (stop words), méthodes vues dans l'état de l'art à la Section 2.2.2. Ainsi on obtient la liste des termes initiaux *mot*.

Génération des sommets initiaux de G_k la ligne 9 de l'Algorithme 6 génère pour chaque terme initial le sommet associé. Un contrôle sur la langue est réalisé grâce à un dictionnaire afin de créer les sommets avec la propriété 'language' correcte. Le poids initial du sommet est de 0.

Génération des sommets de niveau 1 : grâce à une ressource lexicale externe permettant d'obtenir les synonymes, les traductions et les mots en relation, les lignes 12 à 20 de l'Algorithme 6 permettent de créer les sommets et arêtes associées. Le cas des arêtes de type 'Abbreviated_by' est obtenu dans le cas où le type est 'Synonym' mais que le terme en relation est le début du terme initial (exemple : prop et propriété).

5.3.5 Enrichissement et reformulation

Comme illustré dans la Figure 5.8, l'objectif de l'enrichissement et de la reformulation est de proposer à l'utilisateur des termes pour étendre la requête. Une reformulation automatique est ensuite réalisée lorsque les termes ont déjà été sélectionnés dans une précédente requête.

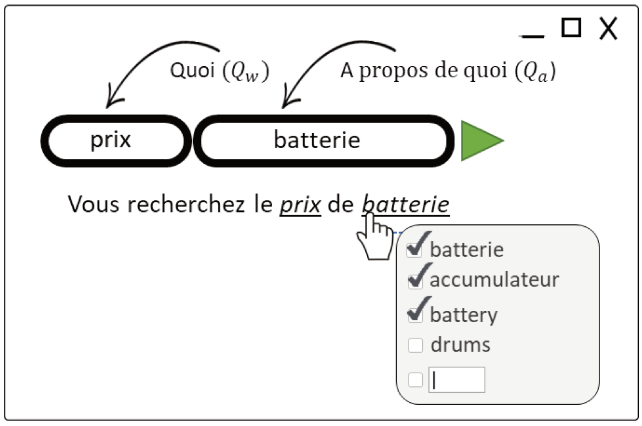


FIG. 5.8 : Interface graphique pour l'expression et la reformulation du besoin d'information

Démarrage du système : la pondération initiale de tous les sommets de G_k est de 0. Dans le cas où aucune traduction n'est trouvée, car aucune n'a de poids supérieur à 0, alors nous les étendons vers leurs traductions de poids inférieurs ou égaux à 0.

Incrémentation des poids selon l'utilisation en requête : pour chaque Qw et Qa soumis par l'utilisateur, on incrémente de 1 le poids des sommets de G_k associés.

Incrémentation des poids selon la reformulation utilisateur : comme l'illustre la Figure 5.8, à chaque soumission de requête, une reformulation de la recherche est présentée à l'utilisateur lui permettant notamment (i) de sélectionner manuellement les termes du graphe G_k non retenus dans l'extension sémantique car de pondération inférieure ou égale à 0, (ii) de dé-sélectionner manuellement les termes sélectionnés par l'extension sémantique, (iii) d'ajouter manuellement un nouveau terme en relation avec le mot-clé de la requête initiale. Dans ce cas, la 'langue' du terme à ajouter ainsi que le 'type' de relation à créer avec le terme initial est également demandé à l'utilisateur. Ces trois actions viennent mettre à jour le graphe G_k et permettent de capitaliser la reformulation pour les futures requêtes. L'action (i) incrémente de 1 et l'action (ii) décrémente de 1 le poids des sommets concernés. L'action (iii) créé le sommet et l'arête associés.

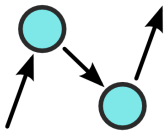
5.3.6 Application au jeu de données

Les dictionnaires français et anglais ont été choisis pour provenir d'une même ressource en ligne[2] avec 194 000 mots en anglais et 208 000 mots en français. La génération du graphe

[2] http://www.gwicks.net/justwords.htm

de connaissance a été réalisée à partir du graphe de connaissance ConceptNet[3] présenté ci-dessous.

ConceptNet



ConceptNet a été choisi comme ressource lexicale externe car celle-ci est multilingue, comprenant les relations entre termes comme des synonymes, mais également des relations plus complexes comme le contexte ou le type d'utilisation du terme. Concept Net permet également une interrogation externe grâce au langage Json[4]. L'utilisation de ConceptNet pour l'extension de requête a notamment été étudié dans (Kotov et al. 2012) et (Speer et al. 2017) montrant une plus grande performance qu'avec d'autres principales sources externes.

Les résultats obtenus avant et après l'intégration de l'extension sémantique des termes sont présentés dans le Tableau 5.3 ainsi que la Figure 5.9. L'évolution du rappel et de la précision se visualise à l'aide du vecteur annoté "avec extension sémantique".

Tab. 5.3 : Résultat d'évaluation de la proposition avant et après le traitement de l'ENJEU 2

	Score précédent	Nouveau score
Précision [0,1]	0.58	0.63
Rappel [0,1]	0.66	0.84
$F_{\beta=0.5}$ [0,1]	0.61	0.69

[3] https://conceptnet.io/
[4] https://docs.python.org/fr/3/library/json.html

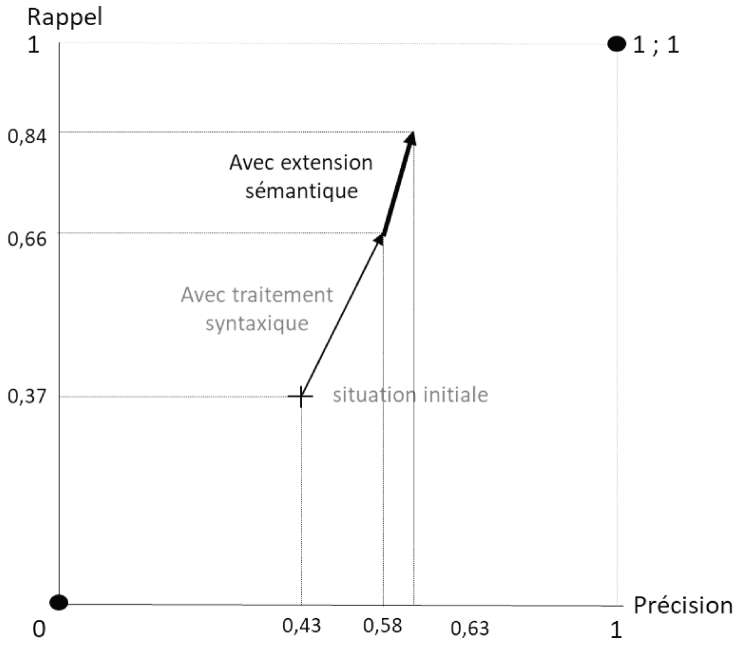


FIG. 5.9 : Score de rappel et de précision après le traitement de l'ENJEU 2

Nous obtenons une nette amélioration du rappel. Désormais, le système permet de restituer 84% des résultats pertinents (gagnant ainsi 18 points par rapport à la proposition initiale). La précision est également améliorée bien qu'avec une moindre importance. Désormais, sur l'ensemble des résultats restitués il y a 63% de résultats pertinents (gagnant ainsi 5 points par rapport à la proposition initiale). Sur la moyenne des deux mesures, le nouveau score de F-Mesure est de 0.69 contre 0.61 avec la proposition initiale. En appliquant uniquement l'enrichissement lié à l'ENJEU 2 sans l'enrichissement lié à l'ENJEU 1, nous augmentons le rappel avec un nombre de résultats restitués plus grand car faisant intervenir un plus large vocabulaire. Néanmoins, cette augmentation est moins importante qu'en ayant au préalable traité l'ENJEU 1 car elle est limitée aux résultats exploitables sans le traitement des tableaux.

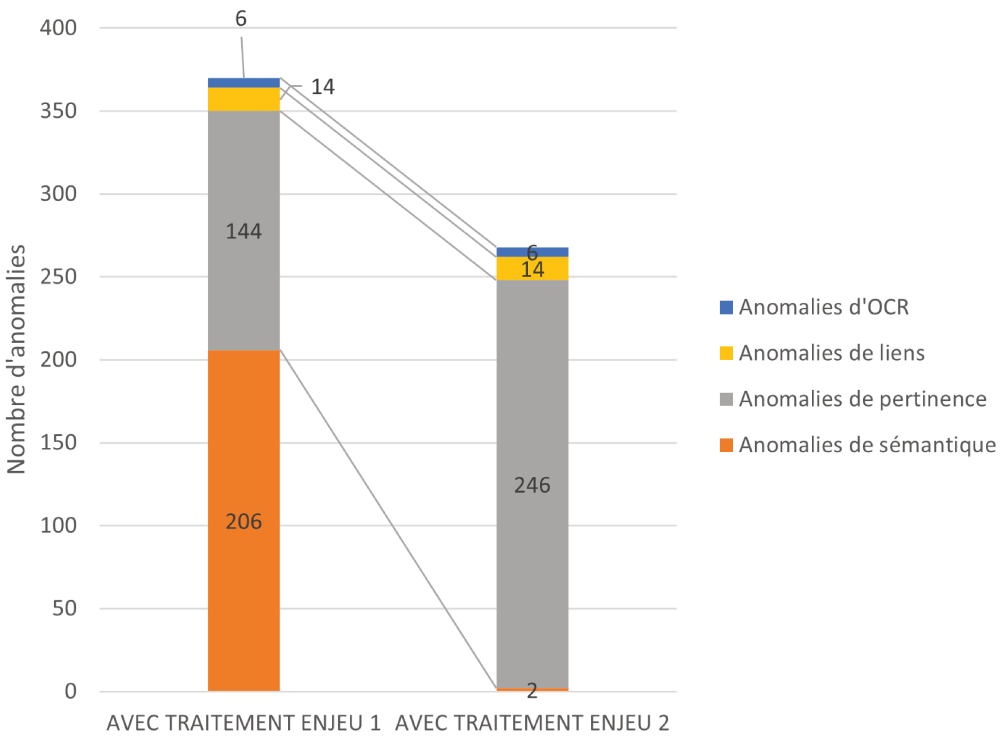


FIG. 5.10 : Répartition des anomalies avant et après traitement de l'ENJEU 2

Concernant la répartition des anomalies restantes, la Figure 5.10 montre une résolution pratiquement totale des anomalies de type sémantique. Seules deux anomalies persistent concernant l'utilisation d'une abréviation non évidente ('batt' pour batterie). Le nombre d'anomalies de type 'moins pertinent' augmente néanmoins de manière importante. En effet, l'extension sémantique des termes a étendu la liste des résultats restitués vers des documents correspondants aux termes mais peu pertinents pour le besoin de l'utilisateur. Le traitement de l'ENJEU 3 est donc d'autant plus important à cette étape d'amélioration de la proposition.

5.4 Traitement des résultats peu pertinents (ENJEU 3)

5.4.1 Paramétrages supplémentaires

Afin de supprimer les résultats restitués peu pertinents, plusieurs opérations sont proposées :

- traitement supplémentaire afin de retirer toutes les réponses obtenues avec Qw et Qa trouvées dans la même valeur de propriété mais séparées de plus de 8 mots. Le chiffre 8 a été choisi suite à des tests successifs sur l'ensemble des requêtes appliquées au jeu de données du drone, c'est donc un choix indépendant de la littérature,
- ajout de règles retirant des résultats restitués ceux obtenus lorsque Qa est dans

un terme mais que Qa est désigné comme une abréviation (par exemple, lorsqu'on recherche 'prop', abréviation de propriété, on retire les éléments contenant 'propriétaire'). En effet, l'utilisation de l'astérisque dans les requêtes graphe notifie que nous recherchons le terme même s'il est contenu dans un autre, ce qui n'est pas toujours le cas. L'identification des abréviations est permise grâce au graphe G_k présenté à la Section 5.3.1.

- Enfin, nous ajoutons une option dans l'interface utilisateur afin que ce dernier puisse sélectionner le type de réponse attendu. Ainsi, s'il attend exclusivement des phrases exprimant de l'exigence, aucun résultat de type document contenant uniquement le terme 'exigence' ne lui sera fourni.

5.4.2 Application au jeu de données

Les résultats obtenus avant et après l'intégration des paramétrages supplémentaires pour traiter l'ENJEU 3 sont présentés dans le Tableau 5.4 ainsi que la Figure 5.11. L'évolution du rappel et de la précision se visualise à l'aide du vecteur annoté "avec filtrage supplémentaire".

Tab. 5.4 : Résultat d'évaluation de la proposition avant et après le traitement de l'ENJEU 3

	Score précédent	Nouveau score
Précision [0,1]	0.63	0.80
Rappel [0,1]	0.84	0.75
$F_{\beta=0.5}$ [0,1]	0.69	0.79

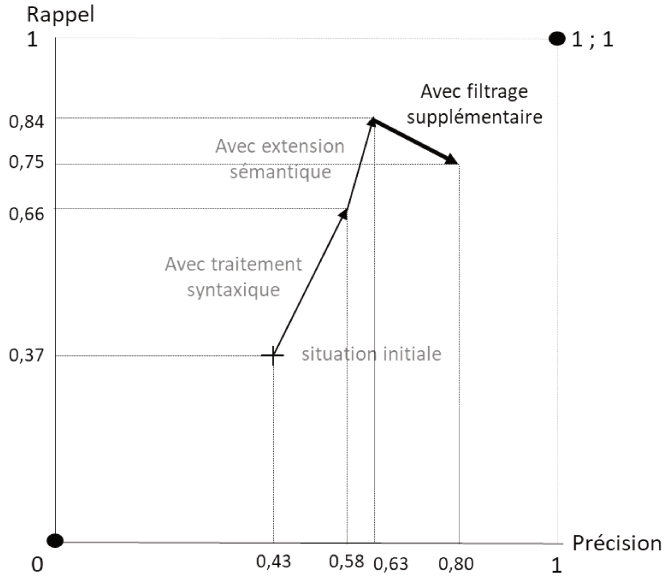


Fig. 5.11 : Score de rappel et de précision après le traitement de l'ENJEU 3

Nous obtenons une nette amélioration de la précision, comportement attendu suite à l'application des filtres sur les résultats restitués peu pertinents. Désormais, le système permet de restituer 80% de résultats pertinents sur l'ensemble des résultats restitués (gagnant ainsi 17 points par rapport à la proposition initiale). Le rappel est quant à lui diminué de 9 points car la proposition filtre également les résultats pertinents. Au global, la moyenne s'améliore avec une F-Mesure de 0.79 contre 0.69 avec la proposition initiale.

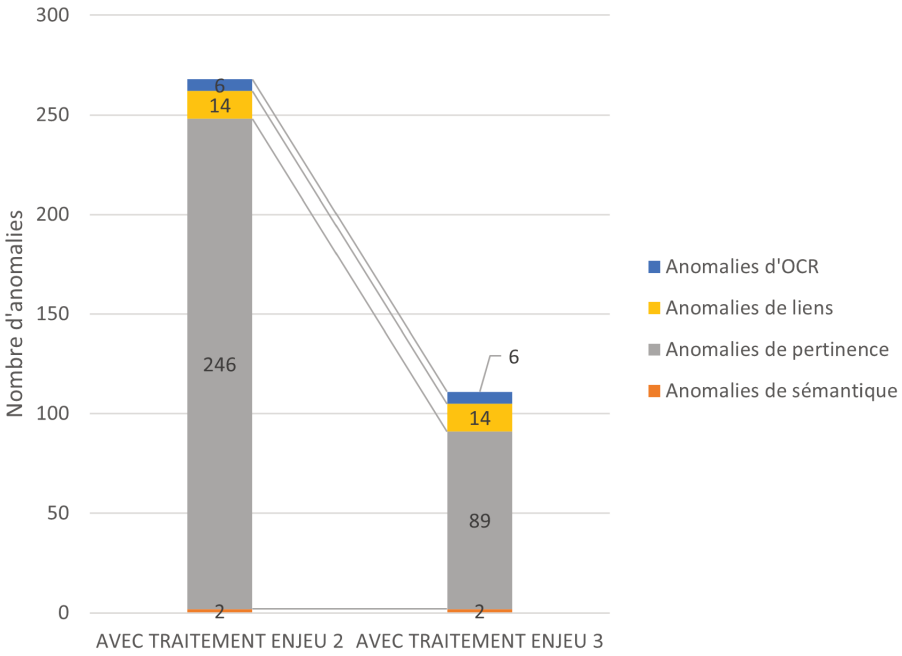


FIG. 5.12 : Répartition des anomalies avant et après traitement de l'enjeu 3

Concernant la répartition des anomalies restantes, la Figure 5.12 montre la résolution partielle des anomalies liées à l'affichage de résultats peu pertinent. C'est tout de même plus de la moitié des anomalies qui ont étés supprimées grâce à la liste des filtres ajoutés. Cet enjeu pourrait être amélioré afin de réduire le nombre de résultats peu pertinent sans masquer les résultats pertinents. Nous pourrions également ajouter un ordonnancement des résultats et donc une évaluation selon cet ordre comme dans les moteurs de recherches classiques du web. Nous listons des pistes à ces sujets dans les perspectives de la thèse, l'étude pour cet enjeu ayant été arrêté à l'obtention des résultats ci-dessus.

5.5 Détection des liens implicites (ENJEU 4)

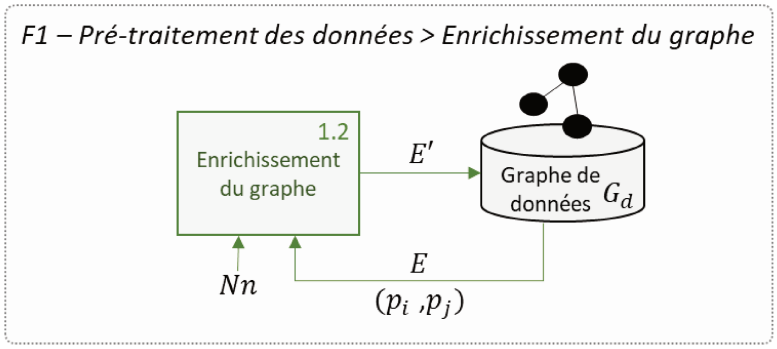


FIG. 5.13 : Illustration de l'enrichissement du graphe des données

Le graphe des données G_d est enrichi par de nouvelles arêtes. Ces arêtes composent l'ensemble E' représentant les relations implicites entre sommets se référant l'un à l'autre. Pour identifier les nouvelles arêtes à créer, on exécute un réseau de neurones entraîné à associer des paires de propriétés de deux sommets de G_d. Cette opération est réalisée dans la fonction 1.2 illustrée dans la Figure 5.13.

5.5.1 Appariement des liens

Prédiction des liens : l'exercice d'appariement de deux sommets d'un graphe peut être vu comme un exercice de prédiction de lien. Néanmoins, comme nous l'avons vu dans la Section 2.3.3 de l'état de l'art, ces approches utilisent l'analyse structurelle existante du graphe. Or au démarrage, les nouveaux sommets n'ont pas de relations sur lesquelles s'appuyer. La détection des relations implicites ne peut donc pas uniquement s'appuyer sur l'analyse structurelle, tout du moins à l'initialisation. Elle peut néanmoins intervenir dans une seconde étape d'enrichissement ultérieure du graphe.

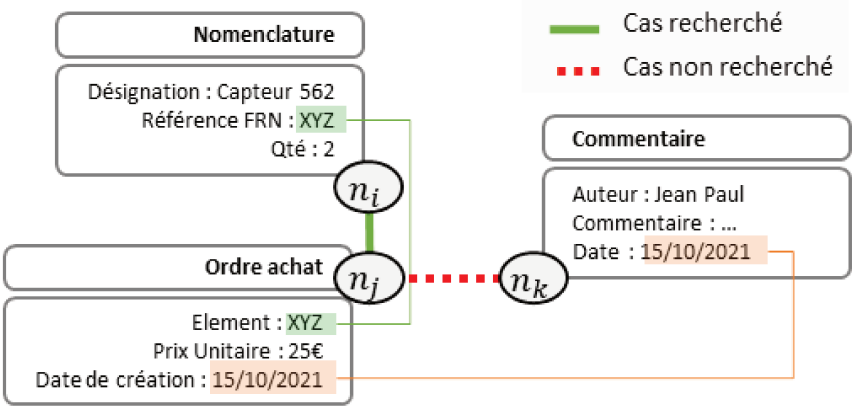


FIG. 5.14 : Exemples de sommets à lier et à ne pas lier

Pour la suite des explications, nous fournissons dans la Figure 5.14 une illustration d'un cas recherché et d'un cas non recherché de détection de relation implicite.

Chaînage d'enregistrement - complément à l'état de l'art : dans l'analyse des bases de données, l'exercice de chaînage d'enregistrement (en anglais appelé également 'record linkage', 'entity resolution' ou encore 'duplicate detection' (Christen 2012)) permet de lier deux éléments se référant à la même entité du monde réel (Talburt 2011). Pour cela, les méthodes considèrent le nombre de champs similaires entre deux enregistrements. La similarité, nommée également distance, est calculée par une comparaison approximative des chaînes de caractères. Cette similarité est réalisée par des méthodes comme celle de Jaro-Winkler (exploitant la longueur de chaîne de caractère, le nombre de caractères communs, leurs distances dans la chaîne ainsi que l'identification d'un préfixe commun (Winkler 1990)) ou de Levenshtein (évaluant le nombre d'opérations nécessaires pour passer d'un champ à l'autre (Levenshtein 1966)). La présence du poids des champs est introduite par des méthodes probabilistes comme celle de Fellegi et Sunter (Fellegi et al. 1969). En effet, la comparaison de deux colonnes 'identifiant' aura plus d'impact que celles de deux colonnes nommées 'prénom'. Néanmoins, dans notre cas, les deux sommets que nous souhaitons lier n'ont pas forcément de multiples champs communs. Les deux sommets peuvent se référer l'un à l'autre, ou comme dans l'exemple de la Figure 5.14, être des vues différentes impliquant un seul champ commun. De plus, les méthodes de chaînage d'enregistrement considèrent l'exercice avec des attributs à comparer présélectionnés. Dans notre cas, cette opération prendrait des ressources importantes face au nombre et à la diversité des propriétés. C'est pour cette raison que des tentatives d'application de solution telle que JeDai (Papadakis et al. 2018), outils open source de résolution d'entité, ne nous ont pas permis de résoudre l'exercice de bout en bout. L'étape de chaînage d'enregistrement peut alors être combinée à une étape de correspondance des schémas nommée 'schema matching' (le 'schéma' décrit la structure de la base de données). Dans (Rahm et al. 2001), vue d'ensemble des méthodes du domaine, on distingue les approches au niveau du schéma et de l'instance, au niveau de l'élément et de la structure, et celles au niveau du langage et des contraintes. Dans notre cas, nous souhaitons être indépendants des informations contenues par le modèle de donnée propre à chacune des sources de données (recherche d'agilité dans un environnement modulaire). C'est pourquoi nous retenons l'utilisation au niveau de l'instance et du schéma en éliminant les informations de structure et de contraintes. Nous retenons également l'approche linguistique comme la recherche des synonymes dans le nom des attributs et celles sur la distance de chaîne de caractères au niveau de l'instance. Ces deux approches seront nommées par la suite et respectivement 'proximité sémantique du nom de la propriété' et 'proximité des chaînes de caractères des valeurs des propriétés'.

Transposition à notre contexte et opérations cognitives : en listant les opérations cognitives réalisées pour l'exercice de détection, on identifie des étapes liées soit au niveau du nom des attributs (correspondance des schémas) soit au niveau de la valeur des attributs (correspondance de l'instance) :

- Analyse de la sémantique du nom de la propriété (fait-elle référence à la désignation d'objet ou plutôt à une description quelconque comme le poids, le prix... ?) - correspondance du schéma,
- Analyse de la proximité sémantique et la proximité des chaînes de caractères des

noms de propriétés (est-ce que les propriétés comparées se réfèrent à la même notion comme peut l'être 'référence' et 'identifiant') - correspondance du schéma,

- Analyse de la chaîne de caractère de la valeur de la propriété (ressemble-t-elle à une référence ou à une chaîne de caractère quelconque) - correspondance de l'instance,
- Analyse de la proximité des chaînes de caractères des valeurs des propriétés comparées - correspondance de l'instance.

Classification automatique binaire et complément à l'état de l'art : traduit en exercice d'apprentissage automatique, c'est finalement une classification binaire des paires de valeurs de propriétés dont la classe 1 serait : 'les deux valeurs nomment la même entité du monde réel' et la classe 0 serait : 'les deux valeurs ne nomment pas la même entité du monde réel', ceci en considérant les quatre grandes familles d'analyses citées précédemment en données d'entrées. Dans l'ensemble des solutions de classification automatique et en s'appuyant sur la comparaison des méthodes d'apprentissage automatique supervisée utilisée par les auteurs dans (Uddin et al. 2019)[5], celle des réseaux de neurones permet à la fois de détecter des relations complexes entre les variables d'entrées qu'importe leurs dépendances, sans distinction de leurs ordres d'apparitions et ceci sans fort coût de calcul une fois défini. Les autres méthodes, plus performantes sur certains aspects, n'arrivent pas à réunir l'ensemble de ces points. L'utilisation des réseaux de neurones pour la mise en correspondance d'attributs hétérogènes a d'ailleurs été introduite dans l'outil SEMINT (W.-S. Li et al. 2000). Depuis, plusieurs autres travaux ont étés menés dans cette direction dont une synthèse des contributions est donnée par (Barlaug et al. 2021). Nous y retenons notamment la piste de traitement de la correspondance des schémas et du chaînage d'enregistrement sous un même exercice de classification automatique. Nous souhaitons adapter cette piste en incluant une analyse linguistique propre aux termes de désignation ou de caractérisation des objets de l'industrie : 'référence', 'part number', 'prix' etc.

5.5.2 Le réseau de neurones sélectionné

Les variables : on fournit au réseau de neurones les couples de propriétés vectorisés selon 13 variables dont les valeurs se situent entre 0 et 1. Les variables peuvent être regroupées selon les quatre familles d'analyse :

- descriptif du nom des propriétés p comprenant leurs distances sémantiques avec les termes d'un lexique lié à la désignation d'élément (ex : 'référence', 'number' etc.). La distance sémantique est utilisée par le calcul de distance des sommets du graphe de connaissance G_k défini dans la Section 5.3.1,
- descriptif des valeurs des propriétés v_p comprenant le nombre de mots et la répartition entre caractères numériques, alphanumériques et spéciaux,
- descriptif de comparaison des noms des propriétés *pi-pj* selon leurs distances de chaîne de caractère et sémantiques (calculées également par la distance des sommets de G_k),

[5]Comparaison récapitulée dans le tableau 4 de l'article (Uddin et al. 2019)

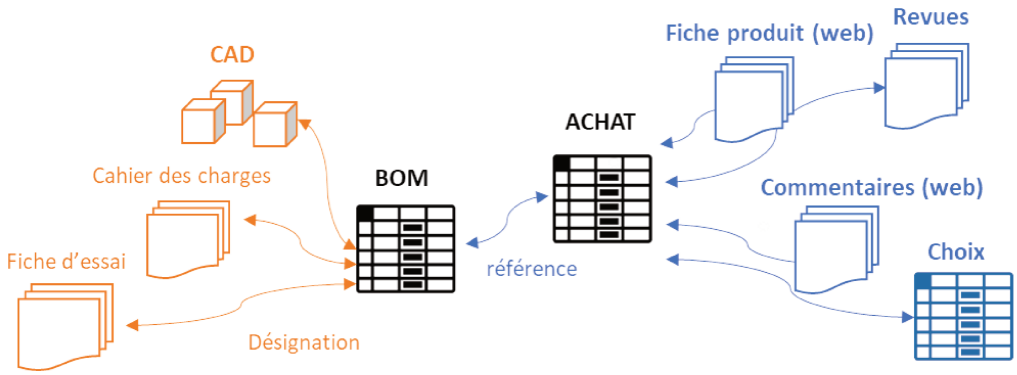


FIG. 5.15 : Sélection d'un périmètre restreint du jeu de données d'entraînement

- descriptif de comparaison des valeurs des propriétés v_{pi}-v_{pj} selon leurs distances de chaîne de caractère (proximité et inclusion de l'une dans l'autre).

Restriction du jeu de données d'entraînement : le jeu de données utilisé pour l'entraînement du réseau de neurones est celui du drone PAINT'R. Pour limiter le temps de traitement nécessaire à l'étiquetage du jeu de données d'entraînement et limiter le temps nécessaire à la vectorisation des individus impactés par la combinatoire importante de paires de propriétés, nous avons restreint le périmètre considéré. Premièrement, et comme le présente la Figure 5.15, nous avons sélectionné les sommets portant une référence ou une désignation référencée dans les documents de type nomenclature et base de données achat. Deuxièmement, nous avons exclu toutes les propriétés p_i dont le nom comporte un terme proche des termes d'un lexique lié à la description d'élément (ex : 'quantité', 'prix' etc.). Cette restriction limite les capacité du réseau à apprendre de cas diversifiés et peut donc impacter négativement les résultats.

Ré-équilibrage du jeu de données d'entraînement : le jeu de données d'entraînement est dit 'déséquilibré', c'est à dire qu'une des classes est majoritairement représentée. Dans notre cas et selon le jeu de données étiqueté, le nombre de propriétés à lier représente moins de 1% des cas. Entraîner un réseau de neurones sur un jeu de données aussi déséquilibré risque de générer de nombreux faux négatifs, autrement dit ne prévoir que des cas où les propriétés ne sont pas à lier (PATTERSON et al. 2018). Afin de ré-équilibrer le jeu de données, on peut pratiquer le sur-échantillonnage de la classe minoritaire, le sous-échantillonnage de la classe majoritaire ou une combinaison de ces deux pratiques. C'est ce troisième choix que nous avons sélectionné suite à une comparaison des résultats obtenus en suivant chacune des pistes. Nous avons donc appliqué au jeu de données un sous-échantillonnage aléatoire des individus de classe 0 puis un sur-échantillonnage des individus de classe 1 avec la méthode SMOTE (CHAWLA et al. 2002).

Définition des hyperparamètres : afin de déterminer le nombre de couches, le nombre de noeuds par couche et les fonctions d'activation du réseau, nous avons utilisé la proposition de (AMZIL et al. 2021) utilisant la neuroevolution pour sélectionner automatiquement

le modèle le plus performant. Cette proposition permet de voir la configuration des hyperparamètres d'un réseau de neurone comme un exercice d'optimisation successif. Nous choisissons cette méthode car elle permet un choix méthodique des hyperparamètres mais permet surtout de restreindre le temps nécessaire pour l'obtention d'un résultat, la méthode classique étant l'expérimentation de chaque modèle possible. Le modèle de réseau de neurone retenu est composé d'une couche cachée composée de quatre noeuds dont les fonctions d'activation sont des tangentes hyperboliques et la fonction d'activation de la couche de sortie est une sigmoid, fonction particulièrement utilisée dans la classification binaire. Le réseau de neurone entraîné obtient alors un score de précision de 0.96 (96% des propriétés liées trouvées sont correctes) et un rappel de 0.99 (99% des propriétés à lier sont trouvées) sur le jeu de données de test.

5.5.3 Application du réseau de neurones

Algorithm 7 : Génération des liens implicites

```
   Input : G_d
   Output : E'
1  E ← extraction des sommets n de G_d
2  propDescription ← propriétés de E dans le lexique de description
3  foreach (n_i, n_j) in E do
4      (p_i, p_j) ← extraction des paires de propriétés si p_i et p_j non dans
         propDescription
5      foreach (p_i, p_j) do
6          (p_i, p_j)' ← vectorisation de (p_i, p_j)
7          score ← application du réseau de neurones Nn sur (p_i, p_j)'
8          if score>0.9 then
9              score(n_i, n_j) += 1
10         else
11             pass
12     if score(n_i, n_j)>0.9 then
13         E' ← n
14     else
15         pass
16 return E'
```

Sélection des paires de propriétés : la ligne 4 de l'Algorithme 7 sélectionne les paires de propriétés à considérer. Ces paires de propriétés sont obtenues par une suite de deux produits cartésiens : entre chaque sommet du graphe donc entre l'ensemble E récupéré en ligne 1 et lui-même (E^2), on obtient alors les paires de sommets ; puis entre l'ensemble des propriétés du premier sommet et celles du second sommet (p_i, p_j). De plus, on exclut de ces paires toutes celles dont une des propriétés se réfère à un lexique de description comme lors du filtrage réalisé dans la restriction du jeu de données d'entraînement.

Vectorisation des paires de propriétés : la ligne 6 de l'Algorithme 7 permet de vectoriser les 13 variables selon les informations de la Section 5.5.2.

Évaluation de la relation implicite : enfin, en appliquant le réseau de neurones, les lignes 7 à 15 de l'Algorithme 7 permettent d'évaluer le nombre de paires de propriétés vectorisées à lier pour une paire de sommets et d'intégrer la valeur dans la variable $score(n_i, n_j)$.

5.5.4 Application au jeu de données

Les résultats obtenus avant et après l'intégration de la détection des liens implicites sont présentés dans le Tableau 5.5 ainsi que la Figure 5.16. L'évolution du rappel et de la précision se visualise à l'aide du vecteur annoté "avec liens implicites".

Tab. 5.5 : Résultat d'évaluation de la proposition avant et après le traitement de l'ENJEU 4

	Score précédent	Nouveau score
Précision [0,1]	0.80	0.72
Rappel [0,1]	0.75	0.91
$F_{\beta=0.5}$ [0,1]	0.79	0.77

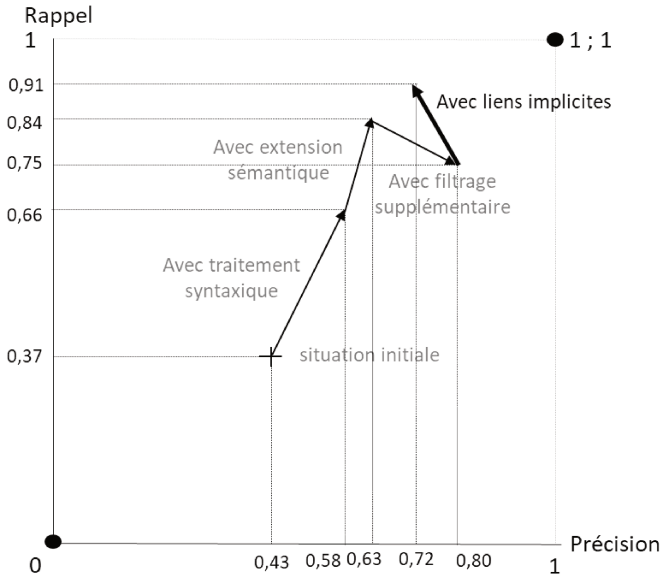


Fig. 5.16 : Score de rappel et de précision après le traitement de l'ENJEU 4

Nous obtenons une amélioration du rappel, comportement attendu suite à la détection des liens implicites manquants pour restituer certains résultats pertinents. Désormais, le

système permet de restituer 91% des résultats pertinents. Néanmoins, la précision chute avec une proposition dont seuls 72% des résultats restitués sont pertinents. En effet, la détection des liens implicites a généré des liens non pertinents entre sommets. La Figure 5.17, présentant la distribution des anomalies, démontre la problématique de liens détectés non pertinents restituant ainsi 494 nouvelles anomalies sous forme de résultats faux. Ces anomalies sont classés dans la figure sous la classe "Anomalies de liens". La proposition en l'état malgré une amélioration du score de rappel ne permet donc pas de répondre correctement à l'ENJEU 4. Ce point est souligné par la baisse de la F-Mesure de 2 points. Il est nécessaire d'entraîner le réseau de neurones sur de plus grandes variétés de cas d'apprentissage et si besoin compléter la démarche avec des solutions complémentaires comme l'ajout d'algorithmes intermédiaires filtrant les combinaisons peu pertinentes détectées.

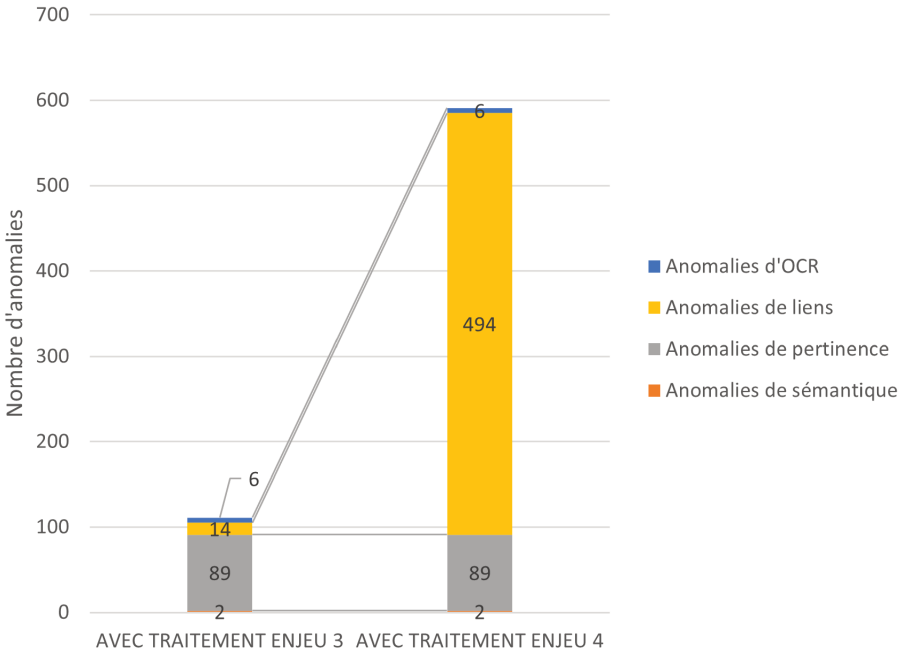


Fig. 5.17 : Répartition des anomalies avant et après traitement de l'ENJEU 4

5.6 Vue générale

Finalement, l'architecture de la proposition enrichie est présentée dans la Figure 5.18.

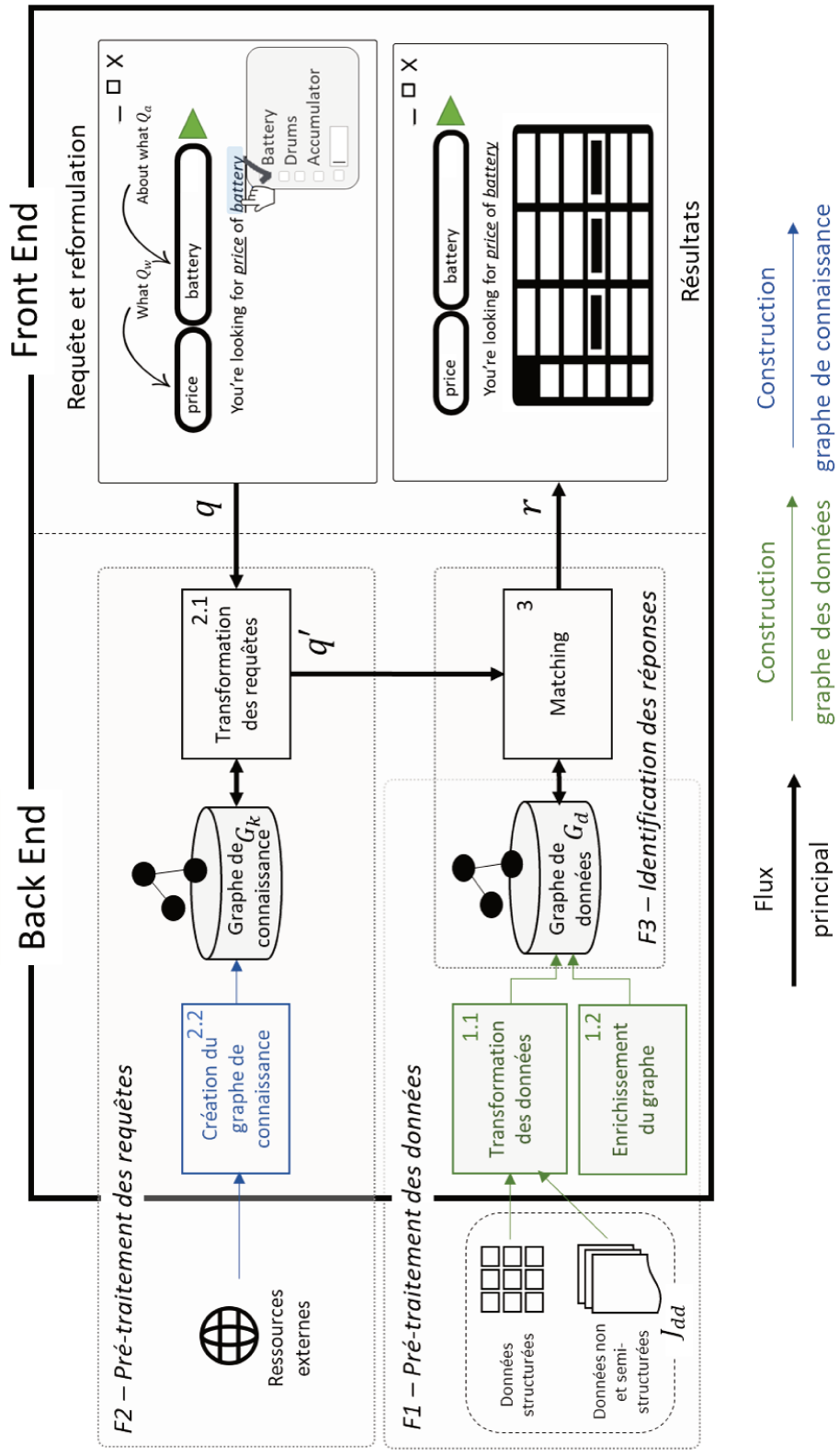


FIG. 5.18 : Architecture de la proposition et des flux d'informations

F1 - Pré-traitement des données : la première fonction est composée d'un premier bloc 1.1 qui génère le graphe de donnée G_d dans lequel le système identifiera les réponses aux requêtes utilisateurs. On y intègre notamment le traitement des tableaux, traitant ainsi l'enjeu 1. Le second bloc 1.2 propose d'enrichir quant à lui le graphe de donnée G_d avec les liens implicites pour traiter le quatrième enjeu. Néanmoins, pour y parvenir, la proposition qui utilise un réseau de neurones a engendré de nombreux liens implicites faux réduisant la précision du système et diminuant ainsi sa performance globale. L'étude de l'enjeu 4 lié à ce point doit donc être poursuivie sachant que la prise en compte d'un jeu de données d'entraînement plus large ainsi que l'ajout de filtres supplémentaires sont des pistes sérieuses.

F2 - Pré-traitement des requêtes : la seconde fonction est quant à elle composée d'un bloc 2.1 permettant de transformer la requête utilisateur q en une requête graphe q'. Plusieurs filtrages ont notamment été intégrés dans ce bloc afin de limiter la restitution de résultats peu pertinents, contribuant ainsi à l'enjeu 3. Un second bloc 2.2 permet d'étendre les mots-clés de la requête vers les termes qui leur sont sémantiquement proche grâce à un graphe de connaissance noté G_k, traitant ainsi le premier enjeu.

F3 - Identification des réponses : enfin, la troisième fonction permet d'identifier dans le graphe des données G_d les éléments de réponses r à la requête pré-traitée q'. Ces résultats sont ensuite affichés à l'utilisateur.

5.7 Synthèse

Plusieurs propositions d'optimisation de la proposition i-Dataquest ont été présentées dans ce chapitre afin de contribuer aux enjeux énoncés en introduction. L'objectif général de la proposition ainsi enrichie est de répondre à la question de recherche "Comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?". Plusieurs propositions donnent de bons résultats comme la transformation des tableaux en graphe, l'extension sémantique des termes de la requête et l'application de filtrage supplémentaire pour mieux cibler les résultats pertinents. On obtient notamment avec l'ensemble de ces éléments une précision de 0.80, un rappel de 0.75 et une F-Mesure de 0.78. Notons que le changement d'ordre d'exécution des différents enrichissement aurait changé les résultats intermédiaires mais pas le résultat final. La quatrième partie du chapitre propose une solution avec réseau de neurones afin de détecter des liens implicites entre des données. Les résultats obtenus sont mitigés, notamment à cause de l'espace limité des cas d'apprentissage fourni au réseau. Néanmoins, le nombre d'anomalies liées à cet enjeu reste limité (initialement 14/1216).

Chapitre 6

Validation et discussion

6.1 Objectif du chapitre

Il est essentiel d'évaluer la pertinence de la proposition vis-à-vis de notre objectif initial afin de renforcer la confiance dans les nouvelles connaissances apportées. C'est à partir de cette évaluation que l'on pourra quantifier et qualifier l'apport scientifique des travaux. Dans notre cas et comme illustré dans la Figure 6.1, l'objectif de ce chapitre est d'évaluer la proposition i-Dataquest vis-à-vis de la question de recherche, à savoir "Comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?".

Fig. 6.1 : Organisation du mémoire : sixième chapitre

6.2 Le carré de validation

Plusieurs méthodes peuvent être employées pour valider une proposition (Barth et al. 2011). Nous retiendrons celle du carré de validation (Seepersad et al. 2006) qui s'applique au domaine de l'ingénierie. Elle permet notamment d'évaluer la proposition sur les deux critères suivants : 'est-ce que la structure proposée est efficace[1] ?' et 'est-ce que la proposition est performante[2] ?'. Pour cela, la méthodologie décompose le carré de validation en quatre étapes comme l'illustre la Figure 6.2.

[1]En anglais, effectiveness
[2]En anglais, efficiency

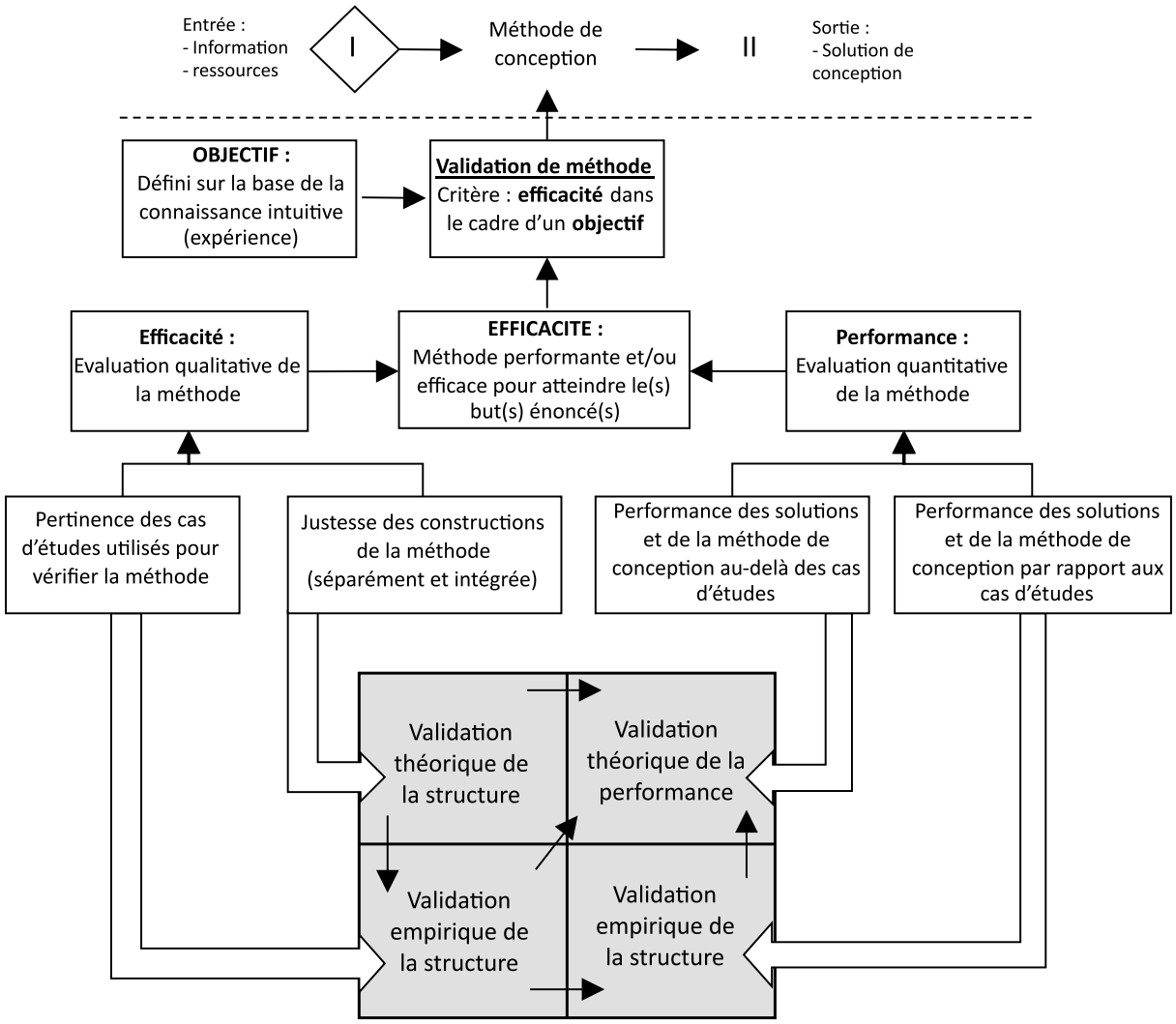


FIG. 6.2 : Le carré de validation traduit de (SEEPERSAD et al. 2006)

6.2.1 Validation de performance

Les validations de la performance (colonne de droite du carré de validation) suggèrent d'évaluer théoriquement et empiriquement la performance de la proposition sur la base de mesures quantitatives. Définies dans le Chapitre 1, les trois exigences suivantes sont à valider :

- Exigence 1 : La proposition doit fournir tous les résultats attendus,
- Exigence 2 : La proposition ne doit fournir que des résultats corrects,
- Exigence 3 : La proposition doit fournir des résultats en un temps limité.

De plus, les enrichissements successifs de la proposition pour répondre aux différents enjeux présentés dans le Chapitre 4 ont générés différentes versions du code d'i-Dataquest. Ainsi, en utilisant tour à tour chacune des versions du code, il est plus simple d'identifier la réelle contribution de chaque itération de la proposition. C'est pourquoi, l'évaluation

empirique de la performance portera sur chacune des étapes d'enrichissement de la proposition. Les étapes successives sont les suivantes :

- **Etape 0 - situation initiale** : le système considéré est celui de la proposition décrite dans le Chapitre 4 et ayant permis de détecter les différents enjeux à traiter. L'architecture n'intègre donc encore aucune proposition à ces enjeux.

- **Etape 1 - avec le traitement des tableaux** : le système considéré à cette étape intègre la proposition à l'enjeu 1 en plus du système initial. On traite donc les tableaux dans les documents comme des sous-sommets des sommets 'document' associés. On intègre donc l'ensemble de la génération du graphe des données décrite dans la Section 5.2.

- **Etape 2 - avec l'extension sémantique** : le système considéré à cette étape intègre, en plus des fonctionnalités liées à l'étape 1, la proposition liée à l'enjeu 2 : l'extension sémantique des mots-clés de la requête grâce au graphe de connaissance multilingue. Cette extension est décrite dans la Section 5.3.

- **Etape 3 - avec le filtrage de résultats** : le système considéré à cette étape intègre, en plus des fonctionnalités liées à l'étape 2, les paramétrages supplémentaires indiqués dans la Section 5.4 contribuant à l'enjeu 3 'traitement des résultats peu pertinents'.

6.2.2 Validation de structure

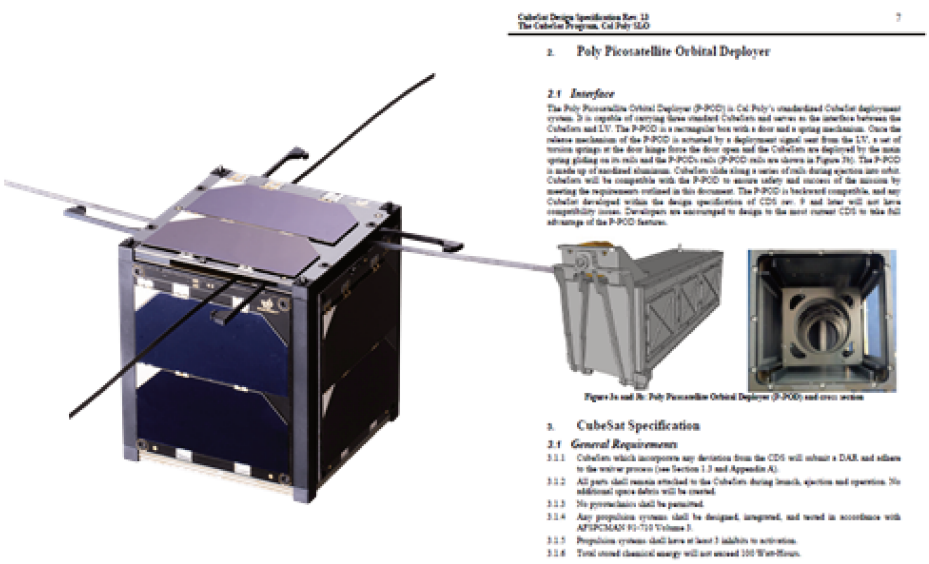
Les validations de structure (colonne de gauche du carré de validation) suggèrent d'évaluer théoriquement et empiriquement l'efficacité de la proposition face aux enjeux initiaux, et ceci sur la base de mesures qualitatives. Définies dans le Chapitre 1, les deux exigences suivantes sont à valider :

- Exigence 4 : Agilité de la proposition à répondre à la variété des besoins d'informations possibles,

- Exigence 5 : Interopérabilité de la proposition vis-à-vis des données sources.

6.2.3 Validation théorique

La validation théorique (première ligne du carré de validation), qui doit démontré l'utilité de la proposition au delà des cas d'études, sera supportée par l'étude de l'architecture de la proposition et de ses fonctions. Il est à noter qu'une partie de la validation des fonctions a été réalisée avec l'application du jeu de données d'étude **PAINT'R** lors du Chapitre 5. Ces validations interviennent alors en soutien à la validation théorique de la performance (exigences 1,2 et 3) en Section 6.4.1 puis celle de la structure (exigences 4 et 5) en Section 6.4.3.



Cubesat Design Specification Rev. 13 7
The CubeSat Program, Cal Poly SLO

2. Poly Picosatellite Orbital Deployer

2.1 *Interface*

The Poly Picosatellite Orbital Deployer (P-POD) is Cal Poly's standardized CubeSat deployment system. It is capable of carrying three standard CubeSats and serves as the interface between the CubeSats and LV. The P-POD is a rectangular box with a door and a spring mechanism. Once the release mechanism of the P-POD is actuated by a deployment signal sent from the LV, a set of torsion springs at the door hinge force the door open and the CubeSats are deployed by the main spring pushing on its rails and the P-PODs rails (P-POD rails are shown in Figure 3b). The P-POD is made up of anodized aluminum. CubeSats slide along a series of rails during ejection into orbit. CubeSats will be compatible with the P-POD to ensure safety and success of the mission by meeting the requirements outlined in this document. The P-POD is backward compatible, and any CubeSat developed within the design specification of CDS rev. 9 and later will not have compatibility issues. Developers are encouraged to design to the most current CDS to take full advantage of the P-POD features.

Figure 3a and 3b: Poly Picosatellite Orbital Deployer (P-POD) and cross section

3. CubeSat Specification

3.1 *General Requirements*

3.1.1 CubeSats which incorporate any deviation from the CDS will submit a DAR, and adhere to the waiver process (see Section 1.3 and Appendix A).

3.1.2 All parts shall remain attached to the CubeSats during launch, ejection and operation. No additional space debris will be created.

3.1.3 No pyrotechnics shall be permitted.

3.1.4 Any propulsion systems shall be designed, integrated, and tested in accordance with AFSPCMAN 91-710 Volume 3.

3.1.5 Propulsion systems shall have at least 3 inhibits to activation.

3.1.6 Total stored chemical energy will not exceed 100 Watt-Hours.

FIG. 6.3 : Représentation numérique d'un CubeSat et un extrait de spécification

6.2.4 Validation empirique

La validation empirique (deuxième ligne du carré de validation) sera quant à elle portée par l'application d'un nouveau jeu de donnée **CubeSat** et des requêtes associées.

Ainsi, après avoir présenté dans la section Section 6.3 le nouveau jeu de données CubeSat et les requêtes associées ainsi que leurs pertinences pour vérifier la proposition, nous validerons empiriquement, avec ce jeu de données, la performance (exigences 1,2 et 3) en Section 6.4.2 puis la structure (exigences 4 et 5) en Section 6.4.4.

6.3 Le cas d'étude : CubeSat

CubeSat est un nano satellite pouvant intégrer (si les spécifications sont respectées) des missions spatiales pour leurs mises en orbite. Une grande diversité d'acteurs génèrent et utilisent les informations autour de CubeSat (du projet universitaire au projet commercial). Cette diversité d'acteurs et d'activités induite autour de la conception de ce produit et de son lancement génère ainsi une grande hétérogénéité syntaxique et sémantique des données. Le jeu de données CubeSat a été sélectionné pour toutes ces raisons et parce qu'il traite d'un produit de conception partagé sur tout le cycle de vie du produit. Beaucoup de spécifications, de données de prototypage et de production sont disponibles librement sur internet.

6.3.1 Récupération des données

Les informations liées au produit CubeSat ont été récupérées à partir du web en explorant plusieurs sites de référence. Voici une liste non exhaustive des sites exploités :

- cubeSat.org[3] : le site officiel dans lequel nous retrouvons les documents de spécifications et de licences nécessaires à la conception d'un CubeSat, les missions spatiales incluant des CubeSat mais également les fournisseurs de références utilisés pour la production d'un CubeSat.
- nasa.gov[4] : le site de la NASA dont plusieurs parties traitent d'information sur les CubeSat comme l'espace 'CubeSats_initiative' ou 'smallsat-institute'. On y retrouve des billets journalistiques et des documents de spécifications. Le site nasa3d[5] a également permis d'obtenir des éléments 3D.
- wikipedia[6] : wikipedia a été utilisé pour retrouver de l'information générique sur le produit, mais également d'autres sites de référence comme la liste de projets et documentations complémentaires.
- libreCube[7] : ce site permet la diffusion des connaissances liées aux produits d'exploration de l'espace et de la Terre. Il référence notamment des répertoires GitHub avec notamment des données de prototypage associé à CubeSat (photos, plans, code, schéma électrique, etc.).
- cubsatdw[8] : ce site référence des documents de workshop réalisés autour de diverses thématiques de recherche liées à CubeSat (solution de communication, de micropropulsions, etc.).
- 3dmdb[9] : ce site a été utilisé pour ajouter des éléments 3D de CubeSat.

6.3.2 Traitement

Provenant exclusivement d'internet et composé majoritairement de page .html, le jeu de données a dû être traité. Ces pages .html ont été sauvegardées en .pdf afin de les traiter comme des documents textuels. Les relations internes à un site exprimées par des liens URL ont été traduites comme des relations explicites. D'autre part, certains fichiers archivés (.zip) ont été désarchivés afin d'exploiter les documents et les relations entre les dossiers et documents internes au fichier. Enfin, certaines vidéos obtenues à partir de page .html ont été récupérées dans un format vidéo (.mp4).

6.3.3 Caractéristiques du jeu de données

Couverture fonctionnelle de l'ensemble de données : elle est souhaitée la plus large possible afin d'être représentatif de l'ensemble des métiers générant des données tout au long du cycle de vie du produit. Le jeu de données est composé d'éléments provenant de multiples compétences : l'activité de conception dont on retrouve les spécifications, la

[3] https://www.cubesat.org/
[4] https://www.nasa.gov/
[5] https://nasa3d.arc.nasa.gov/
[6] https://fr.wikipedia.org/wiki/CubeSat
[7] https://librecube.org/
[8] https://www.cubesatdw.org/
[9] https://3dmdb.com/

définition du produit à la fois mécanique et électronique, l'activité assimilable à la logistique avec l'ordonnancement des différentes configurations de CubeSat selon les missions, l'activité de production avec les nomenclatures, les activités juridiques avec les procédures pour obtenir les licences nécessaires à l'envoi d'un CubeSat en orbite et des activités de revues de projets avec des présentations de travaux de recherches, leurs acteurs et leurs planifications lors de congrès.

Hétérogénéité syntaxique des données : elle est souhaitée grande, comprenant une variété de type de données, structurées et non structurées. Le jeu de données souffre d'un manque de représentativité de données structurées, car les informations obtenues proviennent exclusivement du web. Le contenu non structuré est quant à lui diversifié comprenant des fichiers textes (.pdf, .txt), des images (.jpg, .png), des vidéos (.mp4), des tableurs (.ods), des plans et 3D (.catdrawing, .catpart, .catproduct, .igs, .stl) mais également de code (.py, .dockerfile, .js, .cpp).

Hétérogénéité sémantique des données : elle est souhaitée grande avec une multitude d'acteurs participant à la création du jeu de données ce qui est le cas de CubeSat. De ce fait, le vocabulaire utilisé est varié utilisant principalement la langue anglaise.

Quantité de données : elle est souhaitée à la fois suffisamment grande pour valider un cas d'étude non trivial mais à la fois limité afin de permettre la définition des ensembles de résultats pertinents par requête et l'analyse des documents restitués. Nous pouvons, après analyse du jeu de données PAINT'R comprenant 472 éléments et celui-ci en comprenant 1241, qu'un nombre compris entre 472 et 1241 est adéquat. Le jeu de données de CubeSat est donc composé de 1241 éléments répartis comme suit : 20% sont des documents contenant du texte, des images, des vidéos et des feuilles de calcul, 4% sont des modèles CAO, 19% sont des éléments dont la seule fonction est la relation entre un objet et un autre, et 57% des données au contenu non structuré d'autres types.

Accessibilité : nous avons rendu disponible le jeu de données sous la plateforme Kaggle sous licence Creative Commons CC BY-NC-SA 4.0, site dédié à des exercices de science des données :

www.kaggle.com/dataset/b16a8961020c9f0b537aa66bbcf9384a2bf6f423d3b0c8bae3efe8f0ad21eb32

Pertinence du jeu de données et complémentarité avec PAINT'R : en comparaison avec le jeu de données PAINT'R, le jeu de donnée CubeSat provient d'une plus grande diversité d'acteurs bien que majoritairement anglophone et d'un projet cette fois déployé à grande échelle (plus de 855 CubeSat lancés en orbites au 31 mai 2018 (Villela et al. 2019)). De plus, le nombre de données est plus de deux fois supérieurs à celui de PAINT'R tout en restant exploitable et analysable pour la validation. On notera néanmoins un périmètre fonctionnel des données plus limité ainsi qu'une représentation des

données provenant des bases de données relationnelles nulle, deux éléments que le jeu de données PAINT'R vient renforcer.

6.3.4 Les requêtes

Pour garantir la cohérence des requêtes utilisées lors de la validation vis-à-vis des usages de l'industrie manufacturière, les requêtes répondent à la liste d'usages définie en Section 3.5.1 de notre contexte. Notons néanmoins l'exception du UC7 nécessitant des données médicales non incluses dans le jeu de données. Ainsi, nous avons choisis les requêtes répondant aux différents usages attendus et en utilisant le vocabulaire de CubeSat comme nous l'avions fait pour PAINT'R. Les requêtes sont majoritairement en anglais afin de correspondre au mieux à la langue utilisée dans le jeu de données.

Tab. 6.1 : Liste des requêtes appliquées au jeu de données CubeSat

N° d'usage	N° de requête	Requête	Type
UC1	q_1	Find all the items mentioning the term battery	(A)
UC2	q_2	Find all patents on electric power	(B)
UC2	q_3	Find all simulations on the scrambler	(B)
UC3	q_4	Find all methodologies on engineering	(B)
UC4	q_5	Find thematic of Charles Acton	(C)
UC5	q_6	Find all price of bq24002[10]	(C)
UC6	q_7	Find all suppliers of Schottky diode	(C)
UC8	q_8	Find all satellite with configuration 3U[11]	(C)
UC9	q_9	Find all the justifications for the choice of the C8051F310[12]	(D)
UC10	q_{10}	Find all requirements containing the term battery	(D)

[10] circuit intégré pour la gestion de charge linéaire des batteries Li-Ion [11] assemblage de 3 CubeSat [12] référence de microcontrolleur (MCU)

6.4 Évaluation

6.4.1 Évaluation théorique de la performance

Pour répondre aux premières exigences de performance 'La proposition doit fournir tous les résultats attendus (exigence 1) et ne doit fournir que des résultats corrects (exigence 2)', nos travaux ont inclus une première évaluation quantitative de ces exigences dans le cadre de la proposition non enrichie présentée dans le Chapitre 4. Cette évaluation nous a permis d'évaluer l'ensemble des enjeux à traiter afin d'obtenir 'tous les résultats attendus' (rappel à 1) et 'uniquement des résultats corrects' (précision à 1). C'est donc en traitant chacun de ces enjeux dans une proposition enrichie que nous traitons ces deux

exigences. Le Tableau 6.2 présente pour chaque enjeu les composants de la proposition enrichie y répondant. Il présente également le résultat de la vérification faite sur chacune de ces propositions réalisée avec le jeu de données PAINT'R. En conclusion, la proposition répond correctement aux trois premiers enjeux et donc aux exigences 1 et 2 dans la limite des anomalies liées au quatrième enjeu engendrant une baisse de la performance globale (voir la conclusion du Chapitre 5). Nous ne retenons donc pas l'intégration du quatrième enjeu dans la suite de la validation.

Tab. 6.2 : Intégration des enjeux par la proposition

Enjeu	Proposition	Section	Validation
Enjeu 1 - Traitement des spécificités syntaxiques des données	Proposition d'un modèle de données unique graphe sous-décomposant notamment les tableaux en sommets adjacents	cf. 4.4	✓
Enjeu 2 - Extension sémantique des termes de la recherche	Extension sémantique par le parcours d'un graphe de connaissance pondéré multilingue	cf. 5.3	✓
Enjeu 3 - Traitement des résultats peu et particulièrement pertinents	Application de paramétrages permettant de distinguer des documents selon la proximité des termes recherchés et l'emplacement des termes dans les chaînes de caractères	Cf. 5.4	✓
Enjeu 4 - Détection des liens implicites entre données	Application d'un réseau de neurones de détection de liens implicites	Cf. 5.5	*X*

L'exigence concernant le temps limité d'accès à l'information (exigence 3) est essentielle pour une solution optimale et applicable sur le terrain. Nous y distinguons deux sous-exigences.

Sous-exigence 1 : la proposition doit permettre l'accès à l'information plus rapidement que sans (manuellement ou avec d'autres solutions). Ici, notre proposition intègre un ensemble de fonctionnalités permettant l'accès à des résultats autrement inatteignables. En effet, aucune solution ne permet à la fois :

- de rechercher des documents et des enregistrements en base de données ainsi que des valeurs de propriétés et des phrases,
- d'exploiter les relations dans la recherche des valeurs de propriétés,
- d'étendre sémantiquement les mots-clés de l'utilisateur,
- d'exploiter le contenu textuel des documents textes et images,
- d'identifier des phrases exprimant notamment des concepts comme l'exigence ou la justification de choix.

Sous-exigence 2 : l'utilisateur doit attendre moins d'une seconde (Arapakis et al. 2014) entre le temps où il émet sa recherche et le moment où il obtient les résultats. Ce point n'a pas justifié de traitement particulier dans la construction de la proposition, car jugé non prioritaire vis-à-vis des autres connaissances recherchées. Des perspectives pour le traiter sont néanmoins listées dans la Section 6.5.2.

6.4.2 Évaluation empirique de la performance

Évolution du rappel et de la précision

Tab. 6.3 : Valeurs de la F-Mesure à l'application de CubeSat

Etape	F-Mesure
Etape 0 - situation initiale	0.52
Etape 1 - avec le traitement des tableaux	0.59
Etape 2 - avec l'extension sémantique	0.79
Etape 3 - avec le filtrage de résultats	0.84

Le Tableau 6.3 présente l'évolution de la F-Mesure selon les différentes étapes. La pondération de β est égal à 0.5, c'est-à-dire que nous estimons que la précision a deux fois plus de poids que le rappel dans notre contexte. On observe la même tendance qu'avec le jeu de donnée du drone, une amélioration à chaque étape.

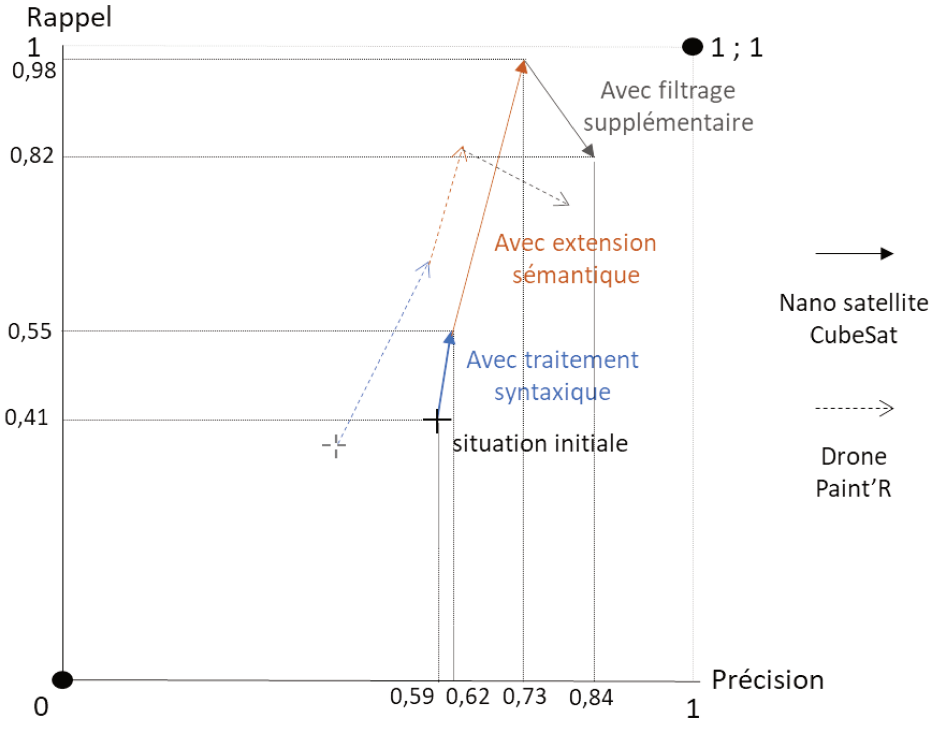


Fig. 6.4 : Représentation vectorielle de l'amélioration du rappel et de la précision

La Figure 6.4 présente les mesures de rappel et de précision sur un graphique cartésien orthogonal normalisé. On obtient visuellement des vecteurs illustrant l'amélioration des performances du système selon les deux critères. Les vecteurs en traits pleins sont les résultats obtenus avec le jeu de données CubeSat. A titre indicatif, les vecteurs en pointillés représentent ceux obtenus avec le jeu de données PAINT'R lors du Chapitre 4 et 5. La situation idéale se situant aux coordonnées 1:1.

L'application de la proposition au cas d'étude CubeSat obtient un rappel de 0.82 et une précision de 0.84. La contribution des différentes étapes a donc permis d'améliorer le rappel de 41 points et la précision de 25 points par rapport au système initial.

Les étapes successives ont améliorées les deux mesures hormis l'étape 3 qui a diminué la valeur du rappel. La diminution du rappel à l'étape 3 s'explique par l'application des filtres sur les résultats moins pertinents masquant également des résultats pertinents. C'est un effet qui a été renforcé par la diminution de l'efficacité des filtres au jeu de données comme le montre l'augmentation de 37 à 41 anomalies sur le graphique 6.5. Néanmoins, l'amélioration de la précision étant avantagée dans le calcul de la performance globale du système, le masquage des quelques résultats non pertinents suffit pour augmenter la précision et donc la F-Mesure. On notera également une augmentation des anomalies de type sémantique à l'étape 1 due à l'augmentation du nombre de résultats restitués. Ces conclusions sont identiques à celles obtenues avec le cas d'étude du drone. Néanmoins des anomalies liées à la syntaxe persistent malgré l'application de l'étape 1. En effet, le traitement des listes à puces n'a pas été opéré et de nouveaux types de fichiers contenant du texte n'a pas été pris en compte dans l'extraction du contenu textuel (fichiers d'extensions .h et .c). L'anomalie liée à la technologie d'OCR est engendrée par l'extraction en vrac de plusieurs termes contenus dans un schéma (ex : 'requirement' et 'battery'). Enfin, le jeu de donnée de CubeSat étant composé d'un grand nombre de document .pdf de taille conséquente, le nombre de phrase répondant à la requête q_{10} identifiant les exigences a généré de nombreuses réponses non pertinentes. Des pistes d'amélioration de la proposition concernant ces points sont fournis dans la conclusion générale.

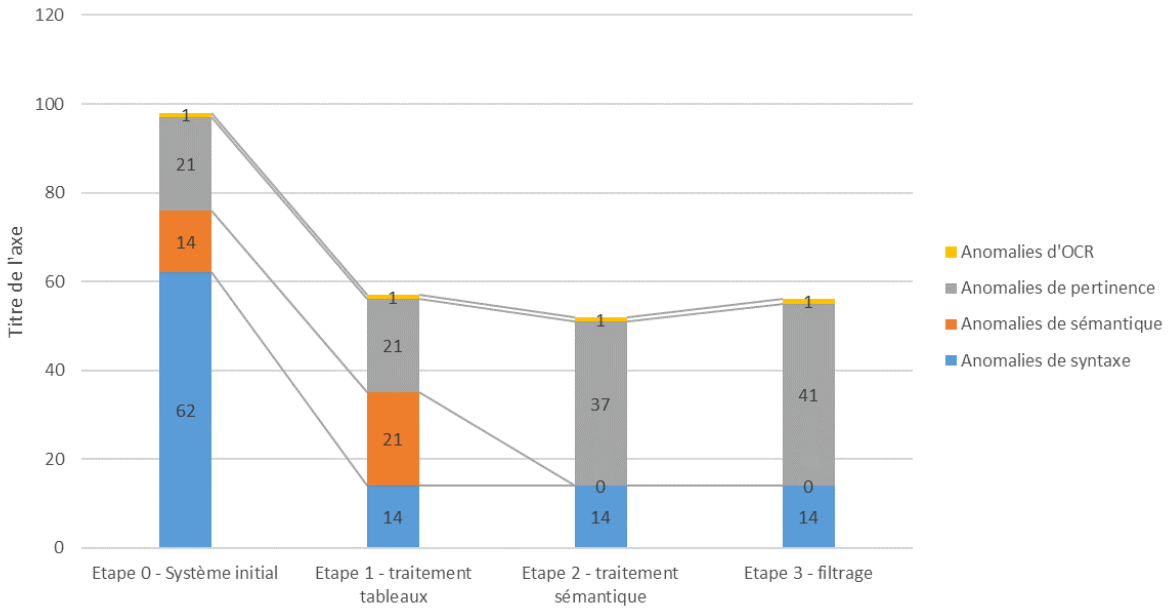


FIG. 6.5 : Distribution des anomalies appliquée au jeu de donnée CubeSat

Le temps de réponse moyen sur les dix requêtes est de 18 minutes. Des optimisations de ce temps de réponse sont donc nécessaires, certaines pistes sont données dans la Section 6.5.2.

6.4.3 Évaluation théorique de la structure

Les systèmes de recherche d'information répondent nativement au caractère 'à la demande' exprimé dans l'**exigence 4 'Agilité de la proposition à répondre à la variété des besoins d'informations'**. En effet, en raison de leurs capacités à intégrer n'importe quel terme dans les requêtes, ils ne supposent pas d'opérations coûteuses pour s'adapter à de nouveaux besoins. De plus, la proposition permet des recherches non incluses dans les systèmes de recherche d'information standard comme 'trouver la valeur de [...] ? ' ou 'trouver les exigences de [...] ? '.

L'étape de transformation des données en un modèle graphe doit considérer les données d'entrées quelques soient leurs syntaxes et leurs provenances. C'est en effet une condition nécessaire pour garantir l'interopérabilité de la proposition vis-à-vis des données sources et ainsi répondre à l'**exigence 5 'Interopérabilité de la proposition vis-à-vis des données sources'**. C'est pour considérer ce point que les règles de transformation définies dans la Section 4.5 considèrent tous les documents et tous les enregistrements en base de donnée. Les seules informations nécessaires pour intégrer la proposition dans un nouvel environnement se limitent alors aux éléments de connexions aux bases de données et aux serveurs. L'intégration d'une base de données No-Sql n'a néanmoins pas été étudiée. Des perspectives pour traiter la question sont fournies dans la Section 6.5.2.

6.4.4 Évaluation empirique de la structure

Le cas d'étude CubeSat a été utilisé dans l'évaluation de la performance et a permis d'expérimenter dix requêtes de différents types comme le montre le Tableau 6.4. A titre inficatif, nous indiquons également dans le tableau les caractéristiques des requêtes utilisées lors de l'application de PAINT'R. Les résultats obtenus montrent la capacité de la proposition à admettre une variété de besoins d'informations exprimés par des requêtes variées, répondant ainsi à l'**exigence 4 'Agilité de la proposition à répondre à la variété des besoins d'informations'**.

Tab. 6.4 : Caractéristique des requêtes utilisées dans les deux cas d'études

Caractéristiques des requêtes	CubeSat	PAINT'R
Requêtes utilisées	10	25
Termes employés[13]	15	36
Langues utilisées	1	2
Profil d'utilisateur	6	7
Type de requête	4	4

[13] hors nom propre et valeurs numériques.

Interopérabilité de la proposition

Une limite de nos travaux est l'intégration automatisée de la proposition. En effet, une partie des opérations de transformation de données se fait par des opérations manuelles. Ainsi, et en l'état, l'évaluation de l'**exigence 5 'Interopérabilité de la proposition vis-à-vis des données sources'** est une limite de notre validation de la proposition.

6.4.5 Synthèse

La proposition i-Dataquest a été construite afin de répondre aux enjeux définis dans le Chapitre 4 permettant, s'ils sont résolus, d'obtenir une solution performante à une demande d'information dans l'industrie manufacturière. Nous avons évalué empiriquement la proposition pour valider ces points (exigences 1 et 2) et par la même occasion éprouvé sa capacité d'adaptation à tout type de demande d'information (exigence 4).

A l'application du jeu de données CubeSat, la proposition obtient un score de rappel de 0.82, de précision 0.84 ce qui donne une F-Mesure de 0.84. Chacune des fonctionnalités principales de la proposition a permis soit une amélioration de la précision, soit du rappel ou les deux. L'apport des fonctionnalités sur le traitement des syntaxes particulières comme les tableaux ainsi que les fonctionnalités sur l'extension sémantique a permis d'améliorer significativement les résultats. Les fonctionnalités ajoutées pour l'affichage des résultats particulièrement pertinents a également permis d'améliorer la précision mais au détriment de nouveaux silences sans toutefois éliminer tous les bruits. Concernant le temps d'affichage des résultats (exigence 3), nous obtenons une moyenne supérieure à un quart

d'heure. Il serait nécessaire de comparer ce temps à d'autres solutions du marché. Néanmoins, cette partie reste à réaliser tout comme l'automatisation de la transformation des données afin d'évaluer empiriquement l'interopérabilité de la solution (exigence 5).

6.5 Discussion

6.5.1 Implications

Les travaux menés ont plusieurs implications pratiques. Tout d'abord, par l'expression d'une requête en deux types de mots-clés, le "quoi" et le "à propos de quoi", la proposition i-Dataquest permet de retrouver des listes de résultats, des valeurs précises d'information ainsi que des phrases exprimant des concepts spécifiques comme l'exigence ou la justification de choix. De plus, la proposition permet de traiter une grande hétérogénéité de données, qu'elles soient structurées dans des bases de données relationnelles ou non structurées prenant la forme de documents textuels, images ou tableurs. Enfin, la proposition permet également d'identifier de l'information par proximité sémantique des termes employés par l'utilisateur.

Concernant les implications théoriques, l'étude montre que l'utilisation d'une modélisation en graphe de l'ensemble des données hétérogènes, distribuées et relationnelles, incluant les enregistrements des bases de données ainsi que la variété des documents composant l'entreprise est une piste intéressante pour y effectuer de la recherche d'information. D'autre part, le Chapitre 4 démontre que l'étude d'un tel système doit inclure des solutions pour l'extraction des spécificités syntaxiques des données d'entrée, l'extension sémantique des termes employés, la sélection des résultats particulièrement pertinents et la détection des liens implicites. Enfin, les résultats obtenus dans ce chapitre montrent que le traitement des tableaux et l'extension sémantique proposés dans i-Dataquest sont des solutions particulièrement efficaces pour la performance d'un tel système.

6.5.2 Limites de l'étude

Les limites actuelles de la proposition sont listées ci-dessous. Ce sont des points à lever pour une potentielle industrialisation.

Automatisation des transformations

L'accès et la transformation des données d'entrée ne sont pas complètement automatisés. La détection des tableaux et de leurs structures dans les documents (détection des en-têtes de colonne et leurs valeurs associées par enregistrement ainsi que leurs titres) pourrait notamment s'inspirer des travaux de (Shafait et al. 2010). Néanmoins, il n'est pas à exclure qu'un standard de création de ces tableaux soit nécessaire aux bonnes performances de la proposition. L'automatisation du reste des transformations présentées dans i-Dataquest peut être réalisée via des outils d'ETL[14] (Kherdekar et al. 2016) mais

[14]ETL pour Extraction Transformation and Load

également avec la méthode d'ATL[15] (Jouault et al. 2005). Le cas particulier des bases de données NoSQL n'a également pas été discuté. Il peut être envisagé qu'une base de données orientée colonnes ou documents puisse être traitée comme les bases de données relationnelles à l'exception des clés étrangères. Les bases de données orientée graphes pourrait être traduite directement dans le graphe de données d'i-Dataquest en respectant les sommets et les arêtes d'origines.

Réconciliation des données au fil de l'eau

L'intégration des données sources n'a été étudiée qu'en mode batch[16] et pas en continu comme leurs générations dans l'entreprise. Il serait donc nécessaire d'enrichir le graphe au fil de l'eau. Notons que le plug-in ETL de Neo4J propose déjà cette option. Une fois cette opération réalisée, il sera nécessaire d'y détecter les évènements liés aux éléments déjà présents dans le graphe. La modification d'un élément (suscitant une nouvelle itération, une nouvelle version ou son abandon) devra alors être traduite et réconciliée au sommet d'origine et très certainement priorisée dans l'affichage de l'information. La réconciliation des différentes versions des données est donc importante et l'utilisation des graphes pour le gérer pourrait être pertinente (Mordinyi et al. 2015).

Gestion des droits

La gestion des droits est un principe clé et fondamental pour une entreprise, et encore plus dans le cadre d'une entreprise étendue. Chaque utilisateur dispose de droits spécifiques lui permettant de voir et modifier les données. Ces droits peuvent être hérités ou non d'un ou plusieurs profils pré-configurés. Dans le cadre de la proposition, seuls les droits de visibilité de l'information sont à considérer. Trois axes semblent possibles : (i) créer de nouveaux profils récupérant et fusionnant les droits des systèmes sources, (ii) créer des profils avec une nouvelle gestion de droit ou (iii) limiter la visibilité des informations sensibles dans la proposition. La première solution (i) semble difficilement réalisable, la gestion de droit de chaque système d'information étant propre à chaque éditeur. La seconde solution est relativement aisée à intégrer, mais son utilisation terrain semble impossible car elle nécessite un expert connaissant l'ensemble du champ des données du graphe. La troisième solution semble prometteuse, la question étant alors de connaître le bon degré d'intégration de l'information accessible à la requête. L'accès à l'information détaillée pouvant être ensuite renvoyée dans le système source où les droits de l'utilisateur sont considérés.

Mise à l'échelle

Le jeu de données, bien que sélectionné pour être au plus proche des caractéristiques des données de l'industrie manufacturière, reste à une échelle exploitable pour les travaux de thèse et donc largement inférieur à un cas industriel réel. La capacité en ressource

[15] ATL pour ATLAS Transformation Language
[16] par lot

permettant la modélisation puis la recherche des éléments en graphe doit donc être considérée. En ce qui concerne les possibilités logicielles, l'optimisation des requêtes graphes par une indexation intermédiaire ou la considération de la typologie du graphe comme suggéré par l'étude (Paradies et al. 2015) ainsi que l'utilisation des graphes distribués (à l'instar de la distribution des fichiers HDFS[17] et permise par OrientDB (Fernandes et al. 2018)) sont des pistes intéressantes. En ce qui concerne les possibilités matérielles, l'intégration d'une indexation de la base de donnée 'in-memory', utilisant alors la mémoire vive d'un ou plusieurs ordinateurs, pourrait également être envisagé (Lake et al. 2013). Le traitement 'in-memory' sur de larges graphes distribués a notamment été étudié dans (Ahn et al. 2015).

[17]HDFS pour Hadoop Distributed File System

Conclusion et perspectives

Conclusion générale

L'exploitation du patrimoine informationnel de l'entreprise est un enjeu majeur pour l'industrie manufacturière. Cette valorisation doit être réalisée dans un contexte où les données sont nombreuses, hétérogènes, explicitement et implicitement liées entre les différents systèmes d'information qui sont eux variables dans le temps. Une solution pour accéder à cet ensemble d'informations doit donc être agile à cet environnement, mais également s'adapter à la variété des besoins d'informations survenant tout au long du cycle de vie du produit et émis par une grande diversité d'acteurs composant l'entreprise étendue.

Pour contribuer à cet enjeu, les travaux de cette thèse cherchent à répondre à la question de recherche "Comment retourner rapidement et à la demande une information exhaustive et pertinente, composée de données distribuées, hétérogènes et relationnelles ?". Pour cela, nous avons effectué un état de l'art sur le domaine de la Recherche d'Information dont les Systèmes (SRI) permettent de renvoyer de l'information à la demande. Nous avons ensuite présenté l'approche graphe qui permet d'exploiter le caractère relationnel des données et des réseaux d'informations. Enfin, nous avons présenté les enjeux liés à la recherche d'information en entreprise, notamment ceux utilisant l'approche graphe. Il apparaît qu'aucune approche de système de recherche d'information utilisant un modèle de donnée graphe pour unifier l'ensemble des données structurées et non structurées de l'entreprise n'est proposée et détaillée. De plus, les listes d'enjeux à considérer dans le contexte de SRI en entreprise ne s'appuient pas sur l'analyse empirique de cas d'étude propre à notre contexte.

C'est pourquoi nous avons proposé un SRI nommé i-Dataquest supporté par un graphe de donnée modélisant l'ensemble des données structurées et non structurées de l'entreprise. L'ensemble des règles de transformation des données ainsi que celle de l'expression du besoin d'information en recherche graphe ont étés détaillés. La transformation des données récupère notamment les enregistrements des tables des bases de données, mais également les documents avec leurs contenus textuels. L'expression du besoin est quant à lui valorisée par le 'quoi' et le 'à propos de quoi' recherchés générant trois types de réponses possibles : la liste de 'documents/enregistrements', la liste de 'valeurs' ou la liste de 'phrases'.

Nous avons confronté la proposition à un cas d'étude dont les caractéristiques sont similaire à celles à l'industrie manufacturière et représentant les données d'un constructeur de drones. L'analyse des anomalies rencontrées nous a permis de lister des enjeux clés à traiter pour améliorer les performances de la proposition. Ces enjeux sont : (i) le

traitement de spécificités syntaxiques des données, (ii) l'extension sémantique des termes de la recherche, (iii) le traitement des résultats peu et particulièrement pertinents et (iv) la détection de liens implicites entre des données distribuées.

Pour traiter ces différents enjeux, nous avons proposé plusieurs enrichissements de la proposition. Dans un premier temps, nous avons ajouté des règles de transformation des données sources en intégrant notamment la traduction des tableaux dans les documents textes et tableurs sous forme de sous-graphe. Dans un second temps, nous avons intégré une extension sémantique des termes de la recherche réalisée grâce à l'interrogation d'un graphe de connaissance construit à partir de ressources lexicales externes confronté à l'ensemble des données de l'entreprise et enrichi lors de l'utilisation du système. Dans un troisième temps, nous avons proposé un mix de paramétrages supplémentaires pour restreindre les résultats affichés au plus pertinents. Enfin, nous avons proposé un enrichissement des liens du graphe de données par l'utilisation d'un réseau de neurones entraîné à détecter des noeuds se référant l'un à l'autre.

L'enrichissement successif de la proposition a été évalué avec le cas d'étude du drone PAINT'R et a conforté les choix réalisés pour le traitement des trois premiers enjeux avec une capacité finale d'affichage de 75% des résultats pertinents attendus avec une précision dans les résultats restitués de 80%. Néanmoins, les résultats obtenus après la proposition d'enrichissement des liens n'ont pas été suffisamment satisfaisants suggérant l'exclusion de ce point à la solution finale i-Dataquest.

La validation de la proposition a été réalisée dans le cadre du carré de validation. L'évaluation de la performance a notamment été réalisée à l'aide d'un second cas d'étude porté sur le nano satellite CubeSat et a fourni les mêmes conclusions qu'avec le cas d'étude du drone. Nous notons néanmoins une limite de validation concernant l'amélioration des temps de réponse vis-à-vis des autres solutions du marché. L'évaluation de la structure suggère que les différents choix rendent la proposition agile vis-à-vis de la variété de besoins d'informations et à l'environnement hétérogène et modulaire de l'architecture. Il est néanmoins nécessaire de confronter empiriquement la proposition à de nombreux autres cas d'études afin de confirmer ce point.

Pour conclure, la proposition i-Dataquest permet une recherche d'information dans le patrimoine informationnel de l'entreprise tout en restant agile à son environnement et au besoin d'information. Son originalité se situe dans l'alliance du domaine de la recherche d'information et de l'approche graphe en considérant l'ensemble des données structurées et non structurées de l'entreprise. C'est une proposition qui s'est appuyée sur l'analyse empirique d'un cas d'étude similaire au contexte de la thèse pour établir les enjeux à traiter. Cette proposition a ensuite été validée grâce à un second cas d'étude. Certaines limites sont toutefois listées dans la Chapitre 6. Plusieurs perspectives sont également détaillées dont l'enrichissement du graphe, l'adaptation de la proposition à l'utilisateur et l'amélioration des possibilités d'interrogation et d'exploration du graphe des données.

Perspectives

La perspective principale de nos travaux est celle d'offrir un point d'accès unique à l'ensemble de l'information hétérogène de l'entreprise, réunifié et 'expliqué' en un modèle

de donnée graphe. Une perspective court terme est celle d'améliorer la proposition d'enrichissement des liens implicites afin d'obtenir une augmentation de la performance du système. Élargir le jeu de données d'entraînement ainsi qu'ajouter des filtrages additionnels pour limiter la détection de liens non pertinents sont deux pistes importantes. D'autre part, le choix de modélisation en graphe permet d'exploiter les capacités d'analyses des réseaux d'informations complexes (Sun et al. 2013) en utilisant les exercices sur les graphes comme la recherche de communauté, de centralité, de similarité et la prédiction de liens. Il pourrait également être envisagé, une fois le graphe des données généré, d'en déduire des réseaux sémantiques ou les méta-modèles transversaux appliqués dans l'entreprise. Afin de contribuer à ces objectifs principaux, plusieurs pistes complémentaires, à plus ou moins long terme, peuvent être suivies.

Enrichir le graphe

Trois types d'enrichissement intéressant du graphe des données permettraient d'étendre les possibilités de recherche et d'analyse :

- Ajouter de nouvelles données qui n'ont pas été vues dans les cas d'étude ni dans les cas d'usages comme les informations provenant de la logistique (avec la représentation des machines utilisées et de l'infrastructure environnante) ou des informations obtenues directement par l'interrogation du web.
- Détailler les spécificités syntaxiques de données déjà transformées en sommets du graphe. Par exemple, et sans exhaustivité, il y a la transformation : (i) des documents semi-structurés de type XML dont l'étiquetage pourrait alors être exploité, (ii) des géométries 3D, images et vidéos qui, une fois traduites, permettrait d'effectuer de la recherche par forme et distinguer par exemple la distance spatiale entre deux éléments recherchés, (iii) le style utilisé dans les documents textuels pour y distinguer les titres et les paragraphes ou enfin (iv) l'étiquetage des pages html pour profiter du réseau d'information offert par la standardisation du web sémantique.
- Etiqueter les données du graphe pour les distinguer dans la recherche et l'analyse. Nous pourrions imaginer la classification par 'type' distinguant un item de nomenclature d'un item 3D, un document de type méthodologie d'une spécification, etc. Cette classification pourrait notamment s'appuyer sur la structuration du graphe.

Adaptation et interaction avec l'utilisateur

Au-delà d'adapter la visibilité de l'information selon les droits de l'utilisateur, il est intéressant d'intégrer un environnement de recherche personnalisé à l'instar de méthodes appliquées dans la recherche web (Micarelli et al. 2007). Le profil de l'utilisateur mais également ses habitudes et le processus dans lequel il se trouve permettrait de mieux cibler l'information et les suggestions de recherche qui lui sont proposées. D'autre part, la récupération de son retour sur la pertinence d'informations obtenues peut s'avérer essentielle pour optimiser l'efficacité de la solution. Ce retour peut être sur la pertinence d'un résultat vis-à-vis d'une requête ou la pertinence d'un élément du graphe.

Explorer de nouvelles méthodes d'interrogation

Dans la proposition de l'étude, le besoin d'information est traité selon la valorisation par des mots-clés de deux variables. L'exploitation directe de l'expression du besoin en langage naturel permettrait à l'utilisateur de gagner le temps de traduction nécessaire pour s'exprimer au système. De plus, l'ouverture vers des requêtes plus complexes mais restant simple de formulation permettrait d'exploiter davantage la structure du graphe (par exemple avec la recherche des documents liés à [...] et à [...]). Il est également envisageable d'explorer le graphe (ou un sous-graphe résultat de la recherche) manuellement en 'ouvrant' les sommets vers leurs sommets voisins, interroger leurs propriétés et ouvrir les documents associés. La conception d'une expérience utilisateur conviviale et soutenue par une proposition ergonomique serait importante à étudier. Enfin, si le langage d'interrogation du graphe est suffisamment standard, il permettrait à des systèmes d'information externes de venir y récupérer de l'information durant leurs propres processus d'activité.

Changement de paradigme - plus de relation

La dernière perspective est également la plus ambitieuse, celle d'un changement de paradigme où nous n'aurions plus besoin de créer les liens explicites dans les différents systèmes sources. Cette perspective permettrait de réduire le temps total nécessaire à la création et à la gestion des données. Pour cela, il serait nécessaire de poursuivre l'étude de la détection des liens afin d'y intégrer celle des liens d'arborescence (par exemple 'cet objet est classé sous ce parent') et des liens de relation originairement renseignés dans le système d'information (par exemple 'cet objet est lié à tel autre objet').

Bibliographie

10303-1:2021, ISO (2021). “Systèmes d’automatisation industrielle et intégration — Représentation et échange de données de produits — Partie 1: Aperçu et principes fondamentaux”. In :

10303-239:2012, ISO (2012). “Systèmes d’automatisation industrielle et intégration — Représentation et échange de données de produits — Partie 239: Protocole d’application : Cycle de vie du produit”. In :

Abadi, Daniel J (2008). “Query execution in column-oriented database systems”. Thèse de doct. Massachusetts Institute of Technology.

Abiteboul, Serge, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset et Pierre Senellart (2011). *Web data management*. Cambridge University Press.

Adger, David (2003). *Core syntax : A minimalist approach*. T. 20. Oxford University Press Oxford.

Aggarwal, Charu C et Haixun Wang (2010). “Graph data management and mining : A survey of algorithms and applications”. In : p. 13-68.

Ahn, Junwhan, Sungpack Hong, Sungjoo Yoo, Onur Mutlu et Kiyoung Choi (2015). “A scalable processing-in-memory accelerator for parallel graph processing”. In : *Proceedings of the 42nd Annual International Symposium on Computer Architecture*, p. 105-117.

Aizawa, Akiko (2003). “An information-theoretic perspective of tf–idf measures”. In : *Information Processing & Management* 39.1, p. 45-65.

Alhabashneh, Obada, Rahat Iqbal, Nazaraf Shah, Saad Amin et Anne James (2011). “Towards the development of an integrated framework for enhancing enterprise search using latent semantic indexing”. In : *International Conference on Conceptual Structures*. Springer, p. 346-352.

Alkilinç, Ahmet et Ahmet Arslan (2018). “A Comparison of Recent Information Retrieval Term-Weighting Models Using Ancient Datasets”. In : *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. IEEE, p. 1-4.

Amati, Giambattista (2003). “Probability models for information retrieval based on divergence from randomness”. Thèse de doct. University of Glasgow.

Amine Lies Benhenni, François-Xavier Bois (2016). *Bases de données orientées graphes avec Neo4j : Manipuler et exploiter vos bases de données orientées graphes*. Eyrolles.

Amzil, Kenza, Esma Yahia, Nathalie Klement et Lionel Roucoules (2021). “Neural networks based causality ranking for decision making in the context of Industry 4.0”. In : *In proceeding in International Journal of Computer Integrated Manufacturing* –, p. —.

Andreasen, Mogens Myrup et Lars Hein (2000). “Integrated product development”. In :

Angles, Renzo et Claudio Gutierrez (2008). "Survey of graph database models". In : *ACM Computing Surveys (CSUR)* 40.1, p. 1-39.

Arapakis, Ioannis, Xiao Bai et B Barla Cambazoglu (2014). "Impact of response latency on user behavior in web search". In : *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, p. 103-112.

ARDOUIN, Etienne (2014). "Composites Design, Manufacturing and Simulation on the 3DEXPERIENCE Platform". In :

Al-Arfaj, Abeer et AbdulMalik Al-Salman (2015). "Ontology construction from text : challenges and trends". In : *International Journal of Artificial Intelligence and Expert Systems* 6.2, p. 15-26.

Arumugam, Subramanian, Andreas Brandstädt, Takao Nishizeki et Krishnaiyan Thulasiraman (2016). *Handbook of graph theory, combinatorial optimization, and algorithms*. T. 34. CRC Press.

Azad, Hiteshwar Kumar et Akshay Deepak (2019). "Query expansion techniques for information retrieval : a survey". In : *Information Processing & Management* 56.5, p. 1698-1735.

Baars, Henning et Hans-George Kemper (2008). "Management support with structured and unstructured data—an integrated business intelligence framework". In : *Information Systems Management* 25.2, p. 132-148.

Baeza-Yates, Ricardo, Berthier Ribeiro-Neto et al. (1999). *Modern information retrieval*. T. 463. ACM press New York.

Balalau, Oana, Helena Galhardas, Ioana Manolescu, Tayeb Merabti, Jingmao You, Youssr Youssef et al. (2020). "Graph integration of structured, semistructured and unstructured data for data journalism". In : *arXiv preprint arXiv :2007.12488*.

Balaneshinkordan, Saeid et Alexander Kotov (2016). "An empirical comparison of term association and knowledge graphs for query expansion". In : *European conference on information retrieval*. Springer, p. 761-767.

Barlaug, Nils et Jon Atle Gulla (avr. 2021). "Neural Networks for Entity Matching : A Survey". In : *ACM Trans. Knowl. Discov. Data* 15.3.

Barsalou, Matthew A (2014). *Root cause analysis : A step-by-step guide to using the right tool at the right time*. CRC Press.

Barth, Alex, Emmanuel Caillaud, Bertrand Rose et al. (2011). "How to validate research in engineering design ?" In : *DS 68-2: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 2: Design Theory and Research Methodology, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011*, p. 41-50.

Bedru, Hayat Dino, Shuo Yu, Xinru Xiao, Da Zhang, Liangtian Wan, He Guo et Feng Xia (2020). "Big networks : A survey". In : *Computer Science Review* 37, p. 100247.

Belkin, Nicholas J et W Bruce Croft (1992). "Information filtering and information retrieval : Two sides of the same coin ?" In : *Communications of the ACM* 35.12, p. 29-38.

Bennett, Mike (2013). "The financial industry business ontology : Best practice for big data". In : *Journal of Banking Regulation* 14.3, p. 255-268.

Berners-Lee, Tim, James Hendler et Ora Lassila (2001). "The semantic web". In : *Scientific american* 284.5, p. 34-43.

Beyssade, Claire (2006). "La structure de l'information dans les questions : quelques remarques sur la diversité des formes interrogatives en français". In : *Linx. Revue des linguistes de l'université Paris X Nanterre* 55, p. 173-193.

Bhogal, Jagdev, Andrew MacFarlane et Peter Smith (2007). "A review of ontology based query expansion". In : *Information processing & management* 43.4, p. 866-886.

Bishr, Yaser (1998). "Overcoming the semantic and other barriers to GIS interoperability". In : *International journal of geographical information science* 12.4, p. 299-314.

Bosc, Patrick, Vincent Claveau, Olivier Pivert et Laurent Ughetto (2009). "Graded-inclusion-based information retrieval systems". In : *European Conference on Information Retrieval*. Springer, p. 252-263.

Bouras, Abdelaziz, Benoit Eynard, Sebti Foufou et Klaus-Dieter Thoben (2016). *Product Lifecycle Management in the Era of Internet of Things : 12th IFIP WG 5.1 International Conference, PLM 2015, Doha, Qatar, October 19-21, 2015, Revised Selected Papers*. T. 467. Springer.

Bowman, Benjamin et H Howie Huang (2021). "Towards Next-Generation Cybersecurity with Graph AI". In : *ACM SIGOPS Operating Systems Review* 55.1, p. 61-67.

Boyer, Arthur et Aurelie Neveol (2018). "Détection automatique de phrases en domaine de spécialité en français (Sentence boundary detection for specialized domains in French)". In : *Actes de la Conférence TALN. Volume 1-Articles longs, articles courts de TALN*, p. 205-214.

Brauer, Falk, Michael Huber, Gregor Hackenbroich, Ulf Leser, Felix Naumann et Wojciech M Barczynski (2010). "Graph-based concept identification and disambiguation for enterprise search". In : *Proceedings of the 19th international conference on World wide web*, p. 171-180.

Bréal, Michel (1987). "Essai de sémantique. Science des signification. Chapitre XVIII". In : *Comment les noms sont donné aux choses*, p. 191-198.

Callan, James P, W Bruce Croft et John Broglio (1995). "TREC and TIPSTER experiments with INQUERY". In : *Information Processing & Management* 31.3, p. 327-343.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall et W Philip Kegelmeyer (2002). "SMOTE : synthetic minority over-sampling technique". In : *Journal of artificial intelligence research* 16, p. 321-357.

Chevallet, Jean-Pierre, Mathias Géry et Hatem Haddad (2000). "Campagne de tests Amaryllis II Expérimentations et résultats (Equipe MRIM-CLIPS, Grenoble)". In :

Christen, Peter (2012). *Data Matching : Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Publishing Company, Incorporated.

Christoforidis, Giannis, Pavlos Kefalas, Apostolos Papadopoulos et Yannis Manolopoulos (2018). "Recommendation of points-of-interest using graph embeddings". In : *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, p. 31-40.

Cleverdon, Cyril (1967). "The Cranfield tests on index language devices". In : *Aslib proceedings*. MCB UP Ltd.

Consortium, Gene Ontology (2004). "The Gene Ontology (GO) database and informatics resource". In : *Nucleic acids research* 32.suppl_1, p. D258-D261.

Cortez, Eli, Philip A Bernstein, Yeye He et Lev Novik (2015). "Annotating database schemas to help enterprise search". In : *Proceedings of the VLDB Endowment* 8.12, p. 1936-1939.

Curran, James R et Marc Moens (2002). "Improvements in automatic thesaurus extraction". In : *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, p. 59-66.

Darwish, Kareem, David Doermann, Ryan Jones, Douglas Oard et Mika Rautiainen (2001). "TREC-10 Experiments at University of Maryland CLIR and Video." In : *TREC*. Citeseer.

Dawood, Harith A (2014). "Graph theory and cyber security". In : *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*. IEEE, p. 90-96.

De Meo, Pasquale, Emilio Ferrara, Giacomo Fiumara et Alessandro Provetti (2011). "Generalized louvain method for community detection in large networks". In : *2011 11th international conference on intelligent systems design and applications*. IEEE, p. 88-93.

Deolekar, Rugved et Akshay Dangare (2018). "Enterprise Search : A New Dimension in Information Retrieval". In : *2018 3rd International Conference for Convergence in Technology (I2CT)*. IEEE, p. 1-6.

Dinh, Ba-Duy (2012). "Accès à l'information biomédicale : vers une approche d'indexation et de recherche d'information conceptuelle basée sur la fusion de ressources termino-ontologiques". Thèse de doct. Université de Toulouse, Université Toulouse III-Paul Sabatier.

Dmitriev, Pavel A, Nadav Eiron, Marcus Fontoura et Eugene Shekita (2006). "Using annotations in enterprise search". In : *Proceedings of the 15th international conference on World Wide Web*, p. 811-817.

Doddington, George R, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel et Ralph M Weischedel (2004). "The automatic content extraction (ace) program-tasks, data, and evaluation." In : *Lrec*. T. 2. 1. Lisbon, p. 837-840.

Dong, Hai, Farookh Khadeer Hussain et Elizabeth Chang (2008). "A survey in traditional information retrieval models". In : *2008 2nd IEEE International Conference on Digital Ecosystems and Technologies*. IEEE, p. 397-402.

Efthimiadis, Efthimis N (1996). "Query Expansion." In : *Annual review of information science and technology (ARIST)* 31, p. 121-87.

Ehrlinger, Lisa et Wolfram Wöss (2016). "Towards a Definition of Knowledge Graphs." In : *SEMANTiCS (Posters, Demos, SuCCESS)* 48, p. 1-4.

Ershadi, Mohammad Javad, Roozbeh Aiasi et Shirin Kazemi (2018). "Root cause analysis in quality problem solving of research information systems : a case study". In : *International Journal of Productivity and Quality Management* 24.2, p. 284-299.

Erven, Gustavo CG van, Maristela Holanda et Rommel N Carvalho (2017). "Detecting evidence of fraud in the brazilian government using graph databases". In : *World conference on information systems and technologies*. Springer, p. 464-473.

Even, Fabrice (2005). "Extraction d'information et modélisation de connaissances à partir de notes de communication orale". Thèse de doct. Université de Nantes.

Feldman, S et J Duhl (2005). "The hidden costs of information work, IDC white paper". In : *Retrieved April* 18, p. 2011.

Fellegi, Ivan P et Alan B Sunter (1969). "A theory for record linkage". In : *Journal of the American Statistical Association* 64.328, p. 1183-1210.

Fernandes, Diogo et Jorge Bernardino (2018). "Graph Databases Comparison : AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB." In : *DATA*, p. 373-380.

Ferret, Olivier, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz, Christian Jacquemin, Nicolas Masson et Paule Lecuyer (2000). "QALC- The Question-Answering System of LIMSI-CNRS". In : *Proceedings of The Ninth Text REtrieval Conference, TREC 2000„ November 13-16, 2000*.

Fortineau, Virginie (2013). "Contribution à une modélisation ontologique des informations tout au long du cycle de vie du produit". Thèse de doct. Ecole nationale supérieure d'arts et métiers-ENSAM.

Fraga, Alvaro Luis, Marcela Vegetti et Horacio Pascual Leone (2020). "Ontology-based solutions for Interoperability among Product Lifecycle Management Systems : A Systematic Literature Review". In : *Journal of Industrial Information Integration*, p. 100176.

Freund, Luanne et Elaine G Toms (2006). "Enterprise search behaviour of software engineers". In : *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, p. 645-646.

Furnas, George W, Scott Deerwester, Susan T Durnais, Thomas K Landauer, Richard A Harshman, Lynn A Streeter et Karen E Lochbaum (2017). "Information retrieval using a singular value decomposition model of latent semantic structure". In : *ACM SIGIR Forum*. T. 51. 2. ACM New York, NY, USA, p. 90-105.

Gaizauskas, Robert et Yorick Wilks (1998). "Information extraction : Beyond document retrieval". In : *Journal of documentation*.

Galliano, Sylvain, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre et Guillaume Gravier (2005). "The ESTER phase II evaluation campaign for the rich transcription of French broadcast news". In : *Ninth European Conference on Speech Communication and Technology*.

Gaye, Mouhamadou (2017). "Estimation et propagation des effets de changement résultant de l'évolution d'une ontologie". Thèse de doct. Université Gaston Berger de Saint-Louis (Sénégal).

Goguen, J. A. (1973). "L. A. Zadeh. Fuzzy sets. Information and control, vol. 8 (1965), pp. 338–353. - L. A. Zadeh. Similarity relations and fuzzy orderings. Information sciences, vol. 3 (1971), pp. 177–200." In : *Journal of Symbolic Logic* 38.4, p. 656-657.

Goh, Cheng Hian, Stuart E Madnick, Michael D Siegel et al. (1995). "Ontologies, contexts, and mediation : representing and reasoning about semantics conflicts in heterogeneous and autonomous systems". In :

Grishman, Ralph et Beth M Sundheim (1996). "Message understanding conference-6: A brief history". In : *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Gruber, Thomas R (1993). "A translation approach to portable ontology specifications". In : *Knowledge acquisition* 5.2, p. 199-220.

Gualtieri, Mike et Scott Compton (2021). "Now Tech : Cognitive Search, Q2 2021 - Forrester's Overview Of 34 Cognitive Search Providers". In : *Forrester Report*.

Gunadi, Erwin, Till Plumbaum et Sahin Albayrak (2015). "Applied Distributed Information Retrieval in Enterprise Search." In : *SCST@ ECIR*.

Harraf, Abe, Isaac Wanasika, Kaylynn Tate et Kaitlyn Talbott (2015). "Organizational agility". In : *Journal of Applied Business Research (JABR)* 31.2, p. 675-686.

Hawking, David (2004). "Challenges in Enterprise Search." In : *ADC*. T. 4. Citeseer, p. 15-24.

Hersh, William R et Ravi Teja Bhupatiraju (2003). "TREC genomics track overview." In : *TREC*. T. 2003. Citeseer, p. 14-23.

Hevner, Alan R, Salvatore T March, Jinsoo Park et Sudha Ram (2004). "Design science in information systems research". In : *MIS quarterly*, p. 75-105.

Hirschberg, Julia et Christopher D Manning (2015). "Advances in natural language processing". In : *Science* 349.6245, p. 261-266.

Imhoff, Claudia et Colin White (2011). "Self-service business intelligence : Empowering users to generate insights". In : *TDWI best practices report* 40.

Jin, Wei et Rohini K Srihari (2007). "Graph-based text representation and knowledge discovery". In : *Proceedings of the 2007 ACM symposium on Applied computing*, p. 807-811.

Jones, David Edward, Yifan Xie, Chris McMahon, Marting Dotter, Nicolas Chanchevrier et Ben Hicks (2015). "Improving enterprise wide search in large engineering multinationals : A linguistic comparison of the structures of internet-search and enterprise-search queries". In : *Ifip international conference on product lifecycle management*. Springer, p. 216-226.

Jones, Rosie, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth JF Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy et Yongze Yu (2021). "Trec 2020 podcasts track overview". In : *arXiv preprint arXiv :2103.15953*.

Jouault, Frédéric et Ivan Kurtev (2005). "Transforming models with ATL". In : *International Conference on Model Driven Engineering Languages and Systems*. Springer, p. 128-138.

Kamel, Mohamed et Yuri Quintana (1990). "A graph based knowledge retrieval system". In : *1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*. IEEE, p. 269-275.

Kassner, Laura, Christoph Gröger, Bernhard Mitschang et Engelbert Westkämper (2015). "Product life cycle analytics–next generation data analytics on structured and unstructured data". In : *Procedia CIRP* 33, p. 35-40.

Kefi-Khelif, Leila (2006). "Un modèle général de recherche d ? information : Application à la recherche de documents techniques par des professionnels". Thèse de doct. Université Joseph-Fourier-Grenoble I.

Kherdekar, Vaishali A et Pravin S Metkewar (2016). "A technical comprehensive survey of ETL tools". In : *International Journal of Applied Engineering Research* 11.4, p. 2557-2559.

Khine, Pwint Phyu et Zhao Shun Wang (2018). "Data lake : a new ideology in big data era". In : *ITM web of conferences*. T. 17. EDP Sciences, p. 03025.

Kim, Lise, Esma Yahia, Frédéric Segonds, Philippe Véron et Victor FAU (2021). "Essential Issues to Consider for a Manufacturing Data Query System Based on Graph". In : *Advances on Mechanics, Design Engineering and Manufacturing III : Proceedings*

of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing, JCM 2020, June 2–4, 2020. Springer, p. 347.

Kim, Lise, Esma Yahia, Frédéric Segonds, Philippe Véron et Antoine Mallet (2020). "i-DATAQUEST : a Proposal for a Manufacturing Data Query System Based on a Graph". In : *IFIP International Conference on Product Lifecycle Management*. Springer, p. 227-238.

Kim, Soonho, Marta Iglesias-Sucasas et Virginie Viollier (2013). "The FAO Geopolitical Ontology : a reference for country-based information". In : *Journal of Agricultural & Food Information* 14.1, p. 50-65.

Kleinberg, Jon M, Mark Newman, Albert-László Barabási et Duncan J Watts (2011). *Authoritative sources in a hyperlinked environment*. Princeton University Press.

Koopman, Bevan, Guido Zuccon, Peter Bruza, Laurianne Sitbon et Michael Lawley (2012). "Graph-based concept weighting for medical information retrieval". In : *Proceedings of the Seventeenth Australasian Document Computing Symposium*, p. 80-87.

Kotov, Alexander et ChengXiang Zhai (2012). "Tapping into knowledge base for concept feedback : leveraging conceptnet to improve search results for difficult queries". In : *Proceedings of the fifth ACM international conference on Web search and data mining*, p. 403-412.

Kwok, KL (1989). "A neural network for probabilistic information retrieval". In : *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 21-30.

Lai, Darong, Hongtao Lu et Christine Nardini (2010). "Enhanced modularity-based community detection by random walk network preprocessing". In : *Physical Review E* 81.6, p. 066118.

Lake, Peter et Paul Crowther (2013). "Concise guide to databases". In : *A History of Databases*.

Larman, Craig (2004). *Agile and iterative development : a manager's guide*. Addison-Wesley Professional.

Latour, Marilyne (2014). "Du besoin d'informations à la formulation des requêtes : étude des usages de différents types d'utilisateurs visant l'amélioration d'un système de recherche d'informations". Thèse de doct. Université de Grenoble.

Leavitt, Neal (2010). "Will NoSQL databases live up to their promise ?" In : *Computer* 43.2, p. 12-14.

Levenshtein, Vladimir I (1966). "Binary codes capable of correcting deletions, insertions, and reversals". In : *Soviet physics doklady*. T. 10. 8. Soviet Union, p. 707-710.

Li, Jingran, Fei Tao, Ying Cheng et Liangjin Zhao (2015). "Big data in product lifecycle management". In : *The International Journal of Advanced Manufacturing Technology* 81.1, p. 667-684.

Li, Wen-Syan et Chris Clifton (2000). "SEMINT : A tool for identifying attribute correspondences in heterogeneous databases using neural networks". In : *Data & Knowledge Engineering* 33.1, p. 49-84.

Li, Yunyao, Ziyang Liu et Huaiyu Zhu (2014). "Enterprise search in the big data era : Recent developments and open challenges". In : *Proceedings of the VLDB Endowment* 7.13, p. 1717-1718.

Lina, Jia-Rui (2020). "OpenBridgeGraph : Integrating Open Government Data for Bridge Management". In : *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*. T. 37. IAARC Publications, p. 1255-1262.

Liu, Jiangzhou et Li Duan (2021). "A survey on knowledge graph-based recommender systems". In : *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. T. 5. IEEE, p. 2450-2453.

Liu, Jiaying, Xiangjie Kong, Xinyu Zhou, Lei Wang, Da Zhang, Ivan Lee, Bo Xu et Feng Xia (2019). "Data Mining and Information Retrieval in the 21st century : A bibliographic review". In : *Computer science review* 34, p. 100193.

Lu, SC-Y, Waguih ElMaraghy, Günther Schuh et Robert Wilhelm (2007). "A scientific foundation of collaborative engineering". In : *CIRP annals* 56.2, p. 605-634.

Lu, Yang (2017). "Industry 4.0: A survey on technologies, applications and open research issues". In : *Journal of industrial information integration* 6, p. 1-10.

Luhn, Hans Peter (1957). "A statistical approach to mechanized encoding and searching of literary information". In : *IBM Journal of research and development* 1.4, p. 309-317.

MacLean, Margaret et Ben H Davis (1998). *Time & bits : managing digital continuity*. Getty Publications.

Martínez-Bazan, Norbert, Victor Muntés-Mulero, Sergio Gómez-Villamor, Jordi Nin, Mario-A Sánchez-Martínez et Josep-L Larriba-Pey (2007). "Dex : high-performance exploration on large graphs for information retrieval". In : *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, p. 573-582.

Meier, Andreas et Michael Kaufmann (2019). "Nosql databases". In : p. 201-218.

Meski, Oussama, Farouk Belkadi, Florent Laroche et Benoit Furet (2019). "Towards a knowledge-based framework for digital chain monitoring within the industry 4.0 paradigm". In : *Procedia CIRP* 84, p. 118-123.

Micarelli, Alessandro, Fabio Gasparetti, Filippo Sciarrone et Susan Gauch (2007). "Personalized search on the world wide web". In : *The adaptive web*. Springer, p. 195-230.

Mika, Peter, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz et Carole Goble (2014). *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*. T. 8797. Springer.

Milanovic, Nikola et Miroslaw Malek (2004). "Current solutions for web service composition". In : *IEEE Internet Computing* 8.6, p. 51-59.

Miller, Justin J (2013). "Graph database applications and concepts with Neo4j". In : *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*. T. 2324. 36.

Mirza, Hamid Turab (2008). "Enterprise Information Retrieval : A Survey." In : *ICEIS (2)*, p. 141-148.

Mittal, Sameer, Muztoba Ahmad Khan, David Romero et Thorsten Wuest (2019). "Smart manufacturing : characteristics, technologies and enabling factors". In : *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture* 233.5, p. 1342-1361.

Miyamoto, Sadaaki (1990). *Fuzzy sets in information retrieval and cluster analysis*. T. 4. Springer Science & Business Media.

Modoni, Gianfranco E, Marco Sacco et Walter Terkaj (2014). "A semantic framework for graph-based enterprise search". In : *Applied Computer Science* 10.4.

Moniruzzaman, ABM et Syed Akhter Hossain (2013). "Nosql database : New era of databases for big data analytics-classification, characteristics and comparison". In : *arXiv preprint arXiv :1307.0191*.

Monnin, Pierre (2020). "Matching and mining in knowledge graphs of the Web of data-Applications in pharmacogenomics". Thèse de doct. Université de Lorraine.

Mordinyi, Richard, Philipp Schindler et Stefan Biffl (2015). "Evaluation of NoSQL graph databases for querying and versioning of engineering data in multi-disciplinary engineering environments". In : *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, p. 1-8.

Mukherjee, Rajat et Jianchang Mao (2004). "Enterprise Search : Tough Stuff : Why is it that searching an intranet is so much harder than searching the Web ?" In : *Queue* 2.2, p. 36-46.

Navigli, Roberto et Simone Paolo Ponzetto (2010). "BabelNet : Building a very large multilingual semantic network". In : *Proceedings of the 48th annual meeting of the association for computational linguistics*, p. 216-225.

Nayak, Ameya, Anil Poriya et Dikshay Poojary (2013). "Type of NOSQL databases and its comparison with relational databases". In : *International Journal of Applied Information Systems* 5.4, p. 16-19.

Noel, Steven, Eric Harley, Kam Him Tam, Michael Limiero et Matthew Share (2016). "CyGraph : graph-based analytics and visualization for cybersecurity". In : *Handbook of Statistics*. T. 35. Elsevier, p. 117-167.

Noy, Natasha, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson et Jamie Taylor (2019). "Industry-scale knowledge graphs : lessons and challenges". In : *Communications of the ACM* 62.8, p. 36-43.

Olivares-Alarcos, Alberto, Daniel Bessler, Alaa Khamis, Paulo Goncalves, Maki K Habib, Julita Bermejo-Alonso, Marcos Barreto, Mohammed Diab, Jan Rosell, Joao Quintas et al. (2019). "A review and comparison of ontology-based approaches to robot autonomy". In :

Ooi, Jessie, Xiuqin Ma, Hongwu Qin et Siau Chuin Liew (2015). "A survey of query expansion, query suggestion and query refinement techniques". In : *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*. IEEE, p. 112-117.

Papadakis, George, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas et Manolis Koubarakis (2018). "The return of jedai : End-to-end entity resolution for structured and semi-structured data". In : *Proceedings of the VLDB Endowment, Vol. 11, No. 12* 11.12, p. 1950-1953.

Paradies, Marcus, Wolfgang Lehner et Christof Bornhövd (2015). "GRAPHITE : an extensible graph traversal framework for relational database management systems". In : *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, p. 1-12.

Patterson, Josh et Adam Gibson (2018). *Deep learning en action - La référence du praticien*. T. 6. Editions First, 2018.

Paulheim, Heiko (2017). "Knowledge graph refinement : A survey of approaches and evaluation methods". In : *Semantic web* 8.3, p. 489-508.

Pinquié, Romain (2016). “Proposition d’un environnement numérique dédié à la fouille et à la synthèse collaborative d’exigences en ingénierie de produits”. Thèse de doct. Paris, ENSAM.

Pinquié, Romain, Philippe Véron, Frédéric Segonds et Nicolas Croué (2016). “Requirement mining for model-based product design”. In : *International Journal of Product Lifecycle Management* 9.4, p. 305-332.

Piskorski, Jakub et Roman Yangarber (2013). “Information extraction : Past, present and future”. In : *Multi-source, multilingual information extraction and summarization*. Springer, p. 23-49.

Poibeau, Thierry (2003). “Extraction automatique d’information(du texte brut au web sémantique)”. In :

Pokornỳ, Jaroslav (2015). “Graph databases : their power and limitations”. In : *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, p. 58-69.

Ponte, Jay M et W Bruce Croft (1998). “A language modeling approach to information retrieval”. In : *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, p. 275-281.

Pradel, Camille (2013). “D’un langage de haut niveau à des requêtes graphes permettant d’interroger le web sémantique”. Thèse de doct. Université de Toulouse, Université Toulouse III-Paul Sabatier.

Prasad, Biren (1996). *Concurrent engineering fundamentals*. T. 1. Prentice Hall PTR NJ.

Presley, Adrien et Donald H Liles (1995). “The use of IDEF0 for the design and specification of methodologies”. In : *Proceedings of the 4th industrial engineering research conference*.

Qaiser, Shahzad et Ramsha Ali (2018). “Text mining : use of TF-IDF to examine the relevance of words to documents”. In : *International Journal of Computer Applications* 181.1, p. 25-29.

Rahm, Erhard et Philip A Bernstein (2001). “A survey of approaches to automatic schema matching”. In : *the VLDB Journal* 10.4, p. 334-350.

Ramar, Kaladevi et Geetha Gurunathan (2016). “Technical review on ontology mapping techniques”. In : *Asian Journal of Information Technology* 15.4, p. 676-688.

Raza, Muhammad Ahsan, Rahmah Mokhtar, Noraziah Ahmad, Maruf Pasha et Urooj Pasha (2019). “A taxonomy and survey of semantic approaches for query expansion”. In : *IEEE Access* 7, p. 17823-17833.

Régnier-Pécastaing, Franck, Michel Gabassi et Jacques Finet (2008). *MDM : Enjeux et méthodes de la gestion des données*. Dunod.

Reinanda, Ridho, Edgar Meij, Maarten de Rijke et al. (2020). “Knowledge graphs : An information retrieval perspective”. In : *Foundations and Trends in Information Retrieval* 14.4.

Robertson, Stephen E (1977). “The probability ranking principle in IR”. In : *Journal of documentation*.

Robertson, Stephen E, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford et al. (1995). “Okapi at TREC-3”. In : *Nist Special Publication Sp* 109, p. 109.

Robertson, Stephen et Hugo Zaragoza (2009). *The probabilistic relevance framework : BM25 and beyond*. Now Publishers Inc.

Rudolf, Michael, Marcus Paradies, Christof Bornhövd et Wolfgang Lehner (2013). "The graph story of the SAP HANA database". In : *Datenbanksysteme für Business, Technologie und Web (BTW) 2037*.

Russell-Rose, Tony, Jon Chamberlain et Leif Azzopardi (2018). "Information retrieval in the workplace : A comparison of professional search practices". In : *Information Processing & Management* 54.6, p. 1042-1057.

Russell-Rose, Tony, Joe Lamantia et Mark Burrell (2011). "A Taxonomy of Enterprise Search." In : *EuroHCIR*, p. 15-18.

Rydning, David Reinsel–John Gantz–John (2018). "The digitization of the world from edge to core". In : *Framingham : International Data Corporation*.

Salton, Gerald (1968). "Automatic information organization and retrieval". In :

Salton, Gerard (1989). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. USA : Addison-Wesley Longman Publishing Co., Inc.

Salton, Gerard, Edward A Fox et Harry Wu (1983a). "Extended boolean information retrieval". In : *Communications of the ACM* 26.11, p. 1022-1036.

Salton, Gerard et Michael McGill (1983b). *Introduction to modern information retrieval*. New York, NY : McGraw-Hill.

Salton, Gerard, Anita Wong et Chung-Shu Yang (1975). "A vector space model for automatic indexing". In : *Communications of the ACM* 18.11, p. 613-620.

Sammut, Claude et Geoffrey I Webb (2017). *Encyclopedia of machine learning and data mining*. Springer.

Sang, Erik F et Fien De Meulder (2003). "Introduction to the CoNLL-2003 shared task : Language-independent named entity recognition". In : *arXiv preprint cs/0306050*.

Sauvagnat, Karen (2005). "Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés". Thèse de doct. Université Paul Sabatier-Toulouse III.

Schabus, Stefan et Johannes Scholz (2017). "Spatially-linked manufacturing data to support data analysis". In : *Journal for Geographic Information Science* 1.15, p. 126-140.

Schütze, Hinrich, Christopher D Manning et Prabhakar Raghavan (2008). *Introduction to information retrieval*. T. 39. Cambridge University Press Cambridge.

Scott, Tyler A (2016). "Analyzing policy networks using valued exponential random graph models : Do government-sponsored collaborative groups enhance organizational networks ?" In : *Policy Studies Journal* 44.2, p. 215-244.

Seepersad, Carolyn C, Kjartan Pedersen, Jan Emblemsvåg, Reid Bailey, Janet K Allen et Farrokh Mistree (2006). "The validation square : how does one verify and validate a design method". In : *Decision making in engineering design*, p. 303-314.

Serrano, Laurie (2014). "Vers une capitalisation des connaissances orientée utilisateur : extraction et structuration automatiques de l'information issue de sources ouvertes". Thèse de doct. Universté de Caen.

Shafait, Faisal et Ray Smith (2010). "Table detection in heterogeneous documents". In : *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, p. 65-72.

Shafique, Umair et Haseeb Qaiser (2014). "A comparative study of data mining process models (KDD, CRISP-DM and SEMMA)". In : *International Journal of Innovation and Scientific Research* 12.1, p. 217-222.

Shahrokni, Ali et Jan Söderberg (2015). "Beyond Information Silos". In : *BigMDE 2015*, p. 63.

Shi, Chuan, Yitong Li, Jiawei Zhang, Yizhou Sun et S Yu Philip (2016). "A survey of heterogeneous information network analysis". In : *IEEE Transactions on Knowledge and Data Engineering* 29.1, p. 17-37.

Singhal, Amit (2012). "Introducing the knowledge graph : things, not strings". In : *Official google blog* 5, p. 16.

Singhal, Amit et al. (2001). "Modern information retrieval : A brief overview". In : *IEEE Data Eng. Bull.* 24.4, p. 35-43.

Singhal, Amit, Chris Buckley et Manclar Mitra (août 2017). "Pivoted Document Length Normalization". In : *SIGIR Forum* 51.2, p. 176-184.

Speer, Robyn, Joshua Chin et Catherine Havasi (2017). "Conceptnet 5.5: An open multilingual graph of general knowledge". In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 31. 1.

Stocker, Alexander, Alexander Richter, Christian Kaiser et Selver Softic (2015). "Exploring barriers of enterprise search implementation : a qualitative user study". In : *Aslib Journal of Information Management*.

Studer, Rudi, V Richard Benjamins et Dieter Fensel (1998). "Knowledge engineering : Principles and methods". In : *Data & knowledge engineering* 25.1-2, p. 161-197.

Sun, Yizhou et Jiawei Han (2013). "Mining heterogeneous information networks : a structural analysis approach". In : *Acm Sigkdd Explorations Newsletter* 14.2, p. 20-28.

Talburt, John R (2011). *Entity resolution and information quality*. Elsevier.

Tamine, Lynda (2000). "Optimisation de requêtes dans un système de recherche d'information approche basée sur l'exploitation de techniques avancées de l'algorithmique génétique". Thèse de doct. Université Paul Sabatier-Toulouse III.

Tarus, John K, Zhendong Niu et Ghulam Mustafa (2018). "Knowledge-based recommendation : a review of ontology-based recommender systems for e-learning". In : *Artificial intelligence review* 50.1, p. 21-48.

Tassinari, Robert (2006). *Pratique de l'analyse fonctionnelle*. Dunod.

Taylor, Chris (2019). "Harness The Value Of Your Data Capital To Drive Business Success". In : *Forrester Report*, p. 2.

Touré, Vasundra, Alexander Mazein, Dagmar Waltemath, Irina Balaur, Mansoor Saqi, Ron Henkel, Johann Pellet et Charles Auffray (2016). "STON : exploring biological pathways using the SBGN standard and graph databases". In : *BMC bioinformatics* 17.1, p. 1-9.

Trieu, Van-Hau (2017). "Getting value from Business Intelligence systems : A review and research agenda". In : *Decision Support Systems* 93, p. 111-124.

Turney, Peter D et Patrick Pantel (2010). "From frequency to meaning : Vector space models of semantics". In : *Journal of artificial intelligence research* 37, p. 141-188.

Turtle, Howard et W Bruce Croft (1989). "Inference networks for document retrieval". In : *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 1-24.

Uddin, Shahadat, Arif Khan, Md Ekramul Hossain et Mohammad Ali Moni (2019). "Comparing different supervised machine learning algorithms for disease prediction". In : *BMC medical informatics and decision making* 19.1, p. 1-16.

Villela, Thyrso, Cesar A Costa, Alessandra M Brandão, Fernando T Bueno et Rodrigo Leonardi (2019). "Towards the thousandth CubeSat : A statistical overview". In : *International Journal of Aerospace Engineering* 2019.

Voorhees, Ellen M et al. (1999). "The TREC-8 question answering track report". In : *Trec*. T. 99. Citeseer, p. 77-82.

Voorhees, Ellen, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff et Lucy Lu Wang (2021). "TREC-COVID : constructing a pandemic information retrieval test collection". In : *ACM SIGIR Forum*. T. 54. 1. ACM New York, NY, USA, p. 1-12.

Wang, Hai, Zeshui Xu et Witold Pedrycz (2017). "An overview on the roles of fuzzy set techniques in big data processing : Trends, challenges and opportunities". In : *Knowledge-Based Systems* 118, p. 15-30.

Weill, Peter, Mani Subramani et Marianne Broadbent (2002). "IT infrastructure for strategic agility". In :

Wieczerniak, Sebastian, Piotr Cyplik et Jarosław Milczarek (2017). "Root cause analysis methods as a tool of effective change". In : *Business Logistics in Modern Management*.

Winkler, William E (1990). "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." In :

Wong, SK Michael, Wojciech Ziarko et Patrick CN Wong (1985). "Generalized vector spaces model in information retrieval". In : *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 18-25.

Yahia, Esma (2011). "Contribution à l'évaluation de l'interopérabilité sémantique entre systèmes d'information d'entreprise : Application aux systèmes d'information de pilotage de la production". Thèse de doct. Université Henri Poincaré-Nancy 1.

Yahia, Esma, Mario Lezoche, Alexis Aubry et Hervé Panetto (2012). "Semantics enactment for interoperability assessment in Enterprise Information Systems". In : *Annual Reviews in Control* 36.1, p. 101-117.

Yoon, Byoung-Ha, Seon-Kyu Kim et Seon-Young Kim (2017). "Use of graph database for the integration of heterogeneous biological data". In : *Genomics & informatics* 15.1, p. 19.

Zhang, Yingfeng, Shan Ren, Yang Liu et Shubin Si (2017). "A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products". In : *Journal of cleaner production* 142, p. 626-641.

Zipf, George Kingsley (1949). "Human behaviour and the principle of least-effort. Cambridge MA edn". In : *Reading : Addison-Wesley*.

Zou, Xiaohan (2020). "A survey on application of knowledge graph". In : *Journal of Physics : Conference Series*. T. 1487. 1. IOP Publishing, p. 012016.

Lise KIM

Proposition d'un système de recherche d'information dans un environnement numérique distribué et hétérogène : application à l'industrie manufacturière

La valorisation du patrimoine informationnel dans l'entreprise de l'industrie manufacturière est un enjeu important. Elle permet la prise de décisions éclairées et de détecter de nouvelles opportunités à valeur ajoutée. Lorsqu'il est retranscrit numériquement, ce patrimoine informationnel est composé de données hétérogènes et distribuées dans les différents silos de l'entreprise rendant la vision holistique de l'information difficile. La thèse propose d'accéder à l'information hétérogène et distribuée de l'entreprise par un système de recherche d'information. L'originalité de la proposition consiste à considérer et modéliser l'ensemble des données structurées et non structurés de l'entreprise dans un graphe unique. L'application de l'approche sur un cas d'étude a permis de détecter une liste d'enjeux clés à traiter pour améliorer les critères de performances usuels en recherche d'information. Les quatre enjeux considérés sont : (i) le traitement des spécificités syntaxiques des données, (ii) l'extension sémantiquement des termes utilisés dans la recherche, (iii) le filtrage les résultats peu pertinents et (iv) la détection de liens implicites entre les données. Enfin, l'approche enrichie des propositions à ces enjeux est confrontée à un second cas d'étude afin de valider la proposition.

Mots clés : Recherche d'information, Base de données orientée graphe, Industrie manufacturière, Extension sémantique de requête

Proposal of an information retrieval system in a distributed and heterogeneous digital environment: application to the manufacturing industry

The valorisation of information in the manufacturing industry is an important issue. It enables informed decisions to be made and new value-added opportunities to be detected. When it is digitally transcribed, this information is composed of heterogeneous data distributed in the different silos of the company, making a holistic view of the information difficult. The thesis proposes to access the heterogeneous and distributed information of the company through an information retrieval system. The originality of the proposal consists in considering and modelling all the structured and unstructured data of the company in a single graph. The application of the approach on a case study has allowed to detect a list of key issues to be addressed to improve the usual performance criteria in information retrieval. The four issues to be considered are (i) the treatment of syntactic specificities of the data, (ii) the semantic extension of the terms used in the search, (iii) the filtering of irrelevant results and (iv) the detection of implicit links between the data. Finally, the approach enriched with proposals to these issues is confronted with a second case study in order to validate the proposal.

Keywords : Information retrieval, graph-oriented databases, Manufacturing Industry, Semantic expansion of queries