Jade MEKKI

Master TAL-IL

RAPPORT d'ALTERNANCE

[Effectué du 18 septembre 2017 au 31 aout 2018]

EDF

32 Avenue Pablo Picasso, 92000 Nanterre

Élaboration d'un outil d'aide à la décision : depuis sa base de connaissance jusqu'à l'interface utilisateur

Sous la direction de :

Mme. Battistelli

Soutenu le 06/07/2018 à l'UFR Phillia

Université Paris Ouest Nanterre La Défense

200 Avenue de la République 92001 Nanterre cedex

Année Universitaire 2018 - 2019

Remerciements

Mes premiers remerciements vont aux enseignants du master TAL-DEFI pour leurs cours de qualité et leurs pédagogies qui ont toujours su s'adapter à nos différents profils. Ils s'adressent particulièrement à madame Battistelli dont les conseils et l'écoute m'ont conduite à cette année d'alternance. Je la remercie plus largement pour tous nos échanges constructifs qui, je l'espère, continueront.

De plus, cette année n'aurait pu être aussi riche sans l'équipe du pôle. D'un point de vue professionnel, elle m'a permis d'apprendre de toutes les différences que nous pouvions avoir (que cela soit en termes de formations ou bien d'approches). De fait, l'ambiance bienveillante et la curiosité de chacun donnent à ce pôle une dynamique intellectuelle stimulante. D'un point de vue humain, je ne peux formuler tout ce que vous m'avez apporté. Aussi, je remercie Charlie, Christophe, Hicham, Kim, Laure, Joanne, Da Lan, Irène, Simon, Moez, Illyass, Mickael, Idriss, Aliénor et Monyrath. Je ne peux m'empêcher de remercier plus particulièrement Adèle pour son honnêteté, sa rigueur intellectuelle, son humour et tout le reste.

Enfin, j'aimerais conclure ces remerciements en les dédiant à Aude. Grâce à qui j'ai pu vivre une année d'alternance qui tint ses promesses et fut riche, intense et passionnante. Je continuerai ma vie professionnelle et académique accompagnée par toutes les valeurs que j'ai acquises au sein du pôle. Ces dernières dépassent largement la dimension professionnelle et je la remercie de les avoir cultivées.

Merci, je suis fière d'avoir participé au début de ce pôle d'intelligence artificielle à vos côtés.

Résumé :

EDF est un fournisseur d'énergie leader en France et en Europe. L'entreprise, forte de ~150 000 employé (2017), se divise en différentes branches elles-mêmes organisées en plusieurs unités. L'une d'entre elles est le Pôle Intelligence Artificielle. Ce dernier fut créé afin de répondre aux besoins métiers avec des solutions dites d'intelligence artificielle : les métiers producteurs ou les fonctions supports sont à l'origine de ces demandes. Ce rapport d'alternance porte sur les problématiques soulevées par le développement d'un outil d'aide à la décision dans le domaine du nucléaire. Un tel outil nécessite des travaux qui vont de l'élaboration d'une base de connaissance, au développement d'algorithmes de classification et de similarité ; de microservices, à la création d'une interface graphique pour l'utilisateur. En d'autres termes, nous verrons la conception de l'outil, sa mise en production et sa maintenance. Les langages majoritairement utilisés sont python et bash.

Mots-clés : base de connaissance, algorithmes de classification et de similarité, intelligence artificielle, mise en production, maintenance

Abstract :

EDF is a leading energy provider in France and Europe. The company, with ~150.000 employees (2017), is divided into different branches which are themselves organized into several units. One of which is the Artificial Intelligence Pole. The Pole aims to answer the business units needs with so called Artificial Intelligence solutions: the energy producer units and the corporate units are at the origin of these demands. This dual education system report deals with the issues raised by the development of a decision support system in the nuclear field. Such a tool requires work ranging from the development of a knowledge base, to the development of classification and similarity algorithms; from microservices, to the creation of a graphical user interface. In other words, we will see the design of such tool, its production and maintenance. The languages mainly used are Python and Bash.

Keywords : knowledge base, classification and similarity algorithms, artificial intelligence, production go-live, maintenance

Droits d’auteurs

TABLES DES MATIERES

Table des illustrations

Table des tableaux

Glossaire

- DSI : Direction des Systèmes d'Information
- DSP : Direction des Services Partagés
- Product Owner : porteur d'offre
- Scrum Master : le garant du respect de la méthode Agile Scrum
- Sprint : une phase de développement qui s'inscrit dans la méthode Agile
- EDF : Électricité de France
- Proof of Concept (POC) :

Introduction

Lorsqu'un enfant grandit, il apprend à parler, c'est à dire à décrire son environnement grâce au langage naturel. Pour cela il acquière un lexique qu'il va peu à peu étendre. Chaque mot de son vocabulaire va être associé à un élément extralinguistique. Prenons l'exemple du mot « chat ». Tout d'abord, l'enfant observe à plusieurs reprises différents chats : un chat noir, un chat roux et un chat tigré. Cependant il remarque des caractéristiques communes : des moustaches, un museau, deux yeux, quatre pattes et une queue. De plus, il comprend qu'à chaque fois qu'il pointe l'animal du doigts ses parents disent le mot « chat ». Il essaie ensuite de répéter ce mot en désignant une table ou bien une chaise, toutefois ses parents le corrigent. Finalement, il remarque que le terme « chat » n'est accepté uniquement lorsqu'il indique des animaux qui partagent les mêmes caractéristiques. Ainsi, aussi simple que cela puisse paraître l'action de nommer un chat convoque un procédé cognitif complexe :

- Identifier l'objet grâce à des caractéristiques récurrentes
- Connaître le concept associé à l'objet
- Mémoriser le terme pour le désigner

Autrement dit, l'enfant apprend à associer le signe, composé d'un signifiant (le mot « chat ») et d'un signifié (la représentation mentale du concept « chat »), à son référent extralinguistique (l'animal chat dans la réalité) (DE SAUSSURE et al, 1933). Il apprend à force de voir des chats, d'essayer de repérer des caractéristiques qui se répètent afin d'exclure les objets qui ne les possèdent pas. Il s'améliore car ses parents lui disent si les éléments qu'il désigne comme des « chats » sont effectivement des « chats » ou s'il se trompe. Ces étapes sont celles que suivent un algorithme d'apprentissage supervisé en informatique.

En effet, pour apprendre à identifier un texte poétique parmi différents genres littéraires, l'algorithme va « lire » beaucoup de textes dont certains seront de la poésie et que nous aurons annotés comme tels. Il va alors pouvoir identifier les motifs récurrents du genre car il sera corrigé lors de ses erreurs d'attribution. Ce type de modèles algorithmes mimétiques de l'apprentissage chez l'homme sont aujourd'hui beaucoup utilisés pour différentes tâches : analyser des images satellites pour prévenir la déforestation, trier des CV, identifier les « fakes news », conduire de façon autonome ...

Beaucoup d'entreprises amorcent une « transformation digitale », c'est-à-dire tirer parti de l'usage d'outils informatiques et de méthodologies associées. Dans ce rapport, nous verrons comment le pôle d'intelligence artificielle accompagne l'« Électricité De France » (EDF) dans cette transformation et les enjeux soulevés par une telle entreprise. En effet, l'industrialisation et la maintenance d'un modèle d'apprentissage ne vont pas de soi et posent des contraintes qu'il faut prendre en compte lors du développement.

En effet, la maintenance d'un outil apprenant est contradictoire puisque l'apprentissage est, d'après le dictionnaire du CNRTL, le fait « d'acquérir par l'entraînement la maîtrise des procédés ou des mécanismes permettant de se livrer à une activité déterminée générale ou professionnelle », ce qui implique un état changeant. Tandis que « Maintenir », toujours d'après le dictionnaire du CNRTL, se définit comme le fait de « faire en sorte que quelqu'un/quelque chose reste dans un état déterminé », c'est à dire inchangé.

Ainsi, nous nous demanderons comment maintenir un modèle d'apprentissage depuis sa base de connaissance jusqu'à son évaluation ? En d'autres termes, comment garder dans un état donné une solution qui doit s'améliorer ? Pour cela nous prendrons un cas d'outil d'aide à la décision dans le nucléaire (projet Gecko). Nous présenterons tout d'abord le projet et le cadre dans lequel ce dernier s'est déroulé, c'est-à-dire l'entreprise ainsi que le pôle d'Intelligence Artificielle (IA). Puis, nous évoquerons les questions soulevées par un tel projet avant d'exposer le travail réalisé. Enfin, nous mettrons en exergue les problèmes rencontrés et les solutions mises en place.

1. Environnement de travail

Mon alternance se déroule chez EDF. Dans un premier temps nous présenterons l'entreprise, puis nous introduirons le pôle IA avant de voir l'écosystème qui gravite autour de ce dernier.

1.1. Présentation de l'entreprise

EDF est une entreprise dont les activités vont de la production d'énergie à sa distribution. Elle fut la première fournisseuse d'énergie mondiale jusqu'en 2017 détrônée par la China General Nuclear Power Company. Créée en 1946 après la décision de nationaliser l'électricité et le gaz, EDF était un Établissement à Caractère Industriel et Commercial (EPIC). En 2004 suite à la volonté européenne d'ouvrir le marché énergétique français à la hauteur de 70%, EDF devint une Société Anonyme à conseil d'administration (SA). Elle ouvrit son capital et entra en bourse en 2005. Ce changement de statut juridique lui permit de varier ses activités (Figure 1). Ainsi, aujourd'hui le groupe d'Électricité de France reste un des leaders mondiaux dans le secteur de l'énergie, il compte ~152 000 employés (recensés en 2017) et sa production énergétique est dominée par le nucléaire (77%). Toutefois il engage désormais 37% de ses investissements bruts dans le développement des énergies renouvelables.

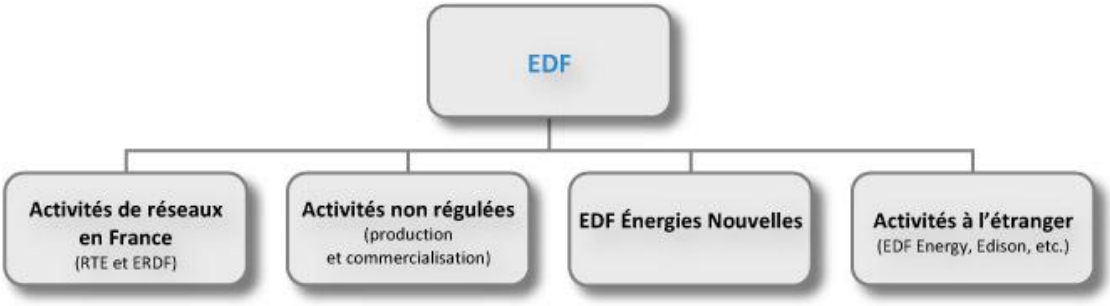

Figure 1 Activités d'EDF

Ses activités à l'étranger représentent une importante partie de sa production (Figure 2). De fait, le groupe compte en 2017 ~35,1 millions de comptes clients dans le monde. Bien qu'il soit leader sur le marché, EDF se trouve dans une situation économique complexe.

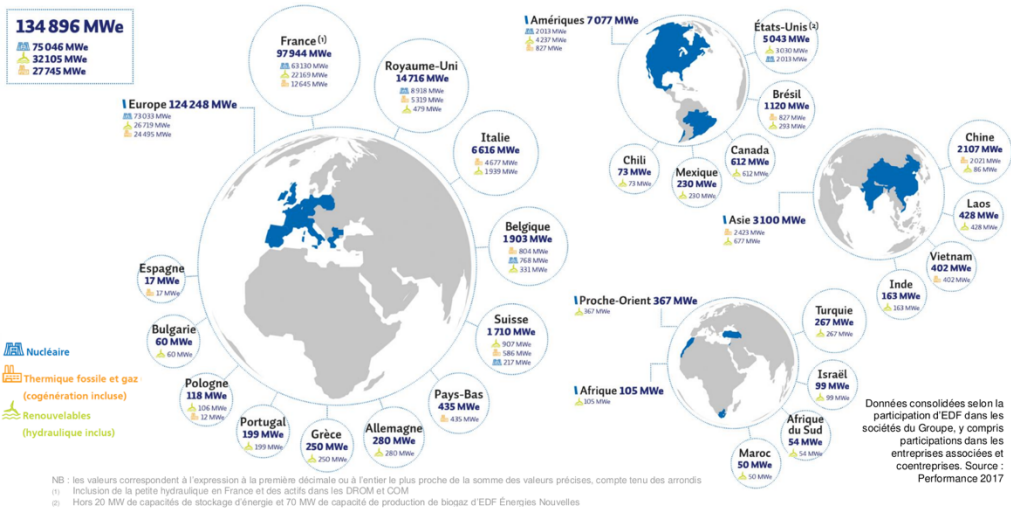

Figure 2 Capacités nettes installées du groupe EDF par pays en 2017

En effet, malgré un chiffre d'affaires de 69,6 Mds€ et un EBITDA de 13,7 Mds€, sa dette financière nette reste de 33,0 Mds€ et annonce en février 2017 en comité central d'entreprise une réduction de ses effectifs ~6% prévue entre 2017 et 2019.

Cette situation s'explique tout d'abord par l'absorption de la branche réacteurs d'Areva en 2016 afin de lui éviter la faillite. La fusion lui coûta 2,5 Mds€ et elle hérita de l'EPR de la centrale nucléaire d'Olkiluoto. Ce projet dont le lancement initialement prévu en 2009 sera finalement lancé en 2018 représente 8 Mds€.

En outre, EDF porte de son côté de nombreux chantiers coûteux. Tout d'abord, il vient d'engager un « grand carénage », c'est-à-dire un plan national qui tend à rénover les 58 réacteurs nucléaires afin de prolonger leurs activités de 40 à 60 ans. Ce dernier devrait prendre fin en 2025 et lui coûterait ~50 Mds€. De plus, il construit deux réacteurs EPR de nouvelle génération en Angleterre dont le prix est estimé à 22 Mds€ : 16 seraient à la charge d'EDF et 6 à celle de China General Nuclear Power Company. Par ailleurs, ces réacteurs à eau pressurisée constituent une nouvelle technologie remise en question depuis le projet de la centrale de Flamanville. Effectivement, les retards de livraison ont triplé le coût initial (désormais le coût est estimé à 10,5 Mds€) et la mise en service prévue pour 2007 est repoussée à 2018.

Enfin, EDF partageait avec Areva le projet Cigéo : un projet d'enfouissement des déchets nucléaires estimé entre 13 et 17 Mds€ et étalonné sur 100 ans. Toutefois ce dernier fut revu à la hausse par la ministre de l'Environnement en 2016 à 25 Mds€.

Ainsi, des chantiers de construction d'EPR, de rénovation ou bien d'enfouissement de déchets nucléaires coûteux ébranlent la situation économique d'EDF dans un contexte d'ouverture du marché énergétique français à la concurrence.

1.2. Présentation du pôle d'intelligence artificielle

C'est dans cette conjoncture économique que la Direction de la Transformation et Efficacité Opérationnelle (DTEO) qui, comme son nom l'indique, tend à accompagner les métiers dans leurs réalisations. Le Pôle Intelligence Artificielle appartient à cette entité (Figure 3). Autrement dit, le Pôle IA dépend de l'unité qui propose des services aux métiers afin de les accompagner dans leurs travaux en industrialisant les outils développés. En effet, ils peuvent être industrialisés puisque le Pôle se trouve dans le département de Développement et Maintenance Applicative (DMA) (Figure 3). Autrement dit, nous pouvons concevoir un modèle d'apprentissage, le développer mais également le déployer en production puis le maintenir grâce aux entités de DMA. Ce point fait notre spécificité par rapport aux autres entités chez EDF. Par exemple, la R&D ne conçoit pas ses outils à des fins d'industrialisation.

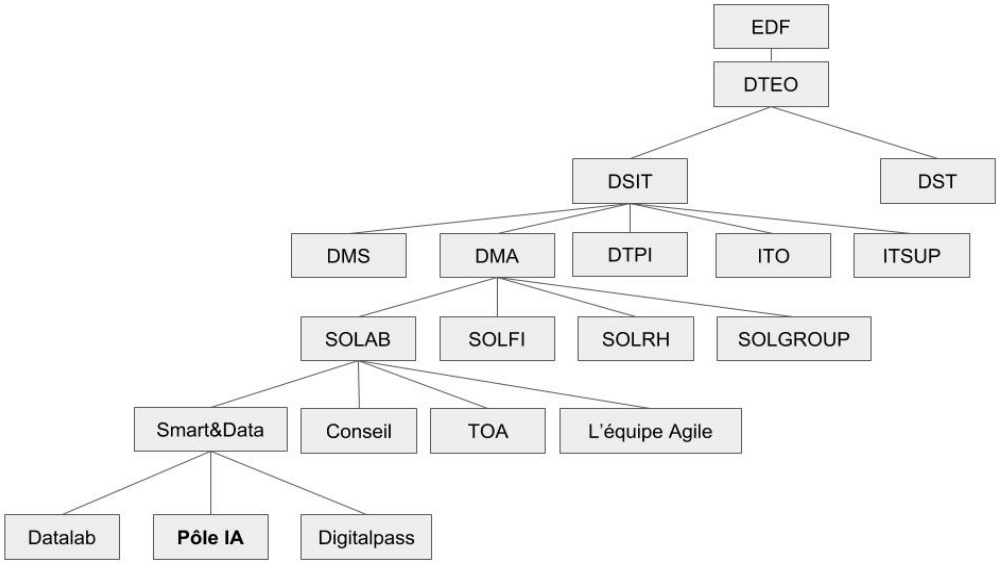

Figure 3 Structure simplifiée de la Direction de la Transformation et Efficacité Opérationnelle (DTEO)

Créé en avril 2017 le Pôle IA est dirigé par Aude Vinzerich. Il tend à concevoir et réaliser des outils dits d'intelligence artificielle. Dans un contexte économique qui impose la baisse des coûts les outils peuvent, par exemple, aider à diminuer les frais de maintenance ou bien à réduire les tâches répétitives à faibles valeurs ajoutées en les automatisant... Toutefois, pourquoi pouvons-nous les qualifier d'"intelligence artificielle" ?

L'origine du terme vient d'une conférence de Dartmouth en 1956. De fait, ce colloque acte la naissance de cette nouvelle discipline et ébauche les premières pistes de réflexions

(réseaux neuronaux, machine learning, étude de la créativité...). Quant à sa définition le dictionnaire en ligne du Centre National de Ressources Textuelles et Lexicales (CNTRL) la décrit comme la “recherche de moyens susceptibles de doter les systèmes informatiques de capacité intellectuelles comparables à celles des êtres humains.” (*La Recherche,* 1979). Dès lors l’objectif de l’IA est tout d’abord de comprendre et formaliser l’appareil cognitif humain afin de le reproduire et l’implémenter. L’ambition de cette discipline convoque différentes disciplines tels que les mathématiques, l’informatique, les sciences du langage, les sciences cognitives, ainsi que différentes techniques tels que les ontologies, l’apprentissage profond, l'apprentissage automatique, l’apprentissage adversarial… Cette interdisciplinarité se retrouve dans la composition de l’équipe ainsi que dans les projets menés par le Pôle. De fait, les solutions développées viennent en support afin d’augmenter l’intelligence des agents tel que l’identification d'événements similaires ou bien remplacent totalement l’intelligence humaine pour des tâches répétitives tel que le tri de mail. Ainsi, nous pouvons bien qualifier les solutions déployées par le Pôle comme étant de l’intelligence artificielle.

1.2.1. L’équipe du Pôle IA

L’interdisciplinarité que demande l’intelligence artificielle se retrouve dans la composition de l’équipe

Tableau 1 Composition de l'équipe du pôle IA

<table>
<tr><th rowspan="2">Métier \ Caractéristiques</th><th colspan="3">Expérience</th><th colspan="2">Statut</th></tr>
<tr><th>Junior</th><th colspan="2">Senior</th><th>Interne</th><th>Externe</th></tr>
<tr><td>Data Scientist</td><td>3</td><td colspan="2">1</td><td>3</td><td>1</td></tr>
<tr><td>Taliste</td><td>2</td><td colspan="2">0</td><td>2</td><td>0</td></tr>
<tr><td>Ingénieur télécom</td><td>2</td><td colspan="2">0</td><td>2</td><td>0</td></tr>
<tr><td>Ingénieur robotique</td><td>1</td><td colspan="2">0</td><td>1</td><td>0</td></tr>
<tr><td>Ingénieur cogniticien</td><td>1</td><td colspan="2">0</td><td>0</td><td>1</td></tr>
<tr><td>Ingénieur des connaissances</td><td>0</td><td colspan="2">1</td><td>0</td><td>1</td></tr>
<tr><td>Développeur web</td><td>0</td><td colspan="2">1</td><td>0</td><td>1</td></tr>
<tr><td>Chef de projet</td><td colspan="3">5</td><td>1</td><td>4</td></tr>
<tr><td rowspan="2">Total</td><td>9</td><td>5</td><td>3</td><td>9</td><td>8</td></tr>
<tr><td colspan="3">17</td><td colspan="2">17</td></tr>
</table>

A cela s’ajoute la responsable du Pôle qui manage toute l’équipe ainsi que du responsable du chantier d’intelligence artificielle chez EDF qui détecte les opportunités de projets auprès des métiers.

1.2.2. Projets menés par le pôle

Tableau 2 Projets menés par le pôle IA

Projets \ Secteur	Fonctions supports	Producteurs
Détection de niveau de Corrosion automatique		Nucléaire
Tri automatique de courriers électroniques	Compatibilité	
Chabot	Service juridique EDF obligation d'achat Service RH Support informatique DSI commerce	
Tri d'alertes machine		Hydraulique
Analyse vidéo du réseau hydraulique		Hydraulique
Identification d'événements similaires		Nucléaire
Prédiction de code matériel		Nucléaire

1.3. Écosystème autours du Pôle IA

Les différentes entités qui gravitent autour du pôle sont présents lors de différentes phrases d'un projet (Figure 4). Conseil accompagne les métiers pour cadrer leurs besoins : faire émerger leurs points de douleurs puis dessiner une solution pour y répondre. Une fois le besoin identifié et la solution imaginée, le Pôle IA la réalise. Devop's appuie ce dernier puisqu'il propose des outils afin de développer avec de bonnes pratiques. Dev@gil s'occupe de tout ce qui est développement web. Le groupe de pilotes opérationnels de projet sont quant à eux des chefs de projet experts du Système Informatique d'EDF qui permettent d'intégrer correctement les solutions dans ce dernier. Enfin, SOLGROUP et UNITEP permettent de les industrialiser selon leurs domaines métiers.

Cet écosystème n'est pas toujours entièrement mobilisé lors d'un projet. En effet, si ce dernier est un POC le Pôle IA a toutes les compétences nécessaires : les chefs de projets cadrent le besoin, les data scientists et les talistes développent la solution et les développeurs web réalisent le démonstrateur.

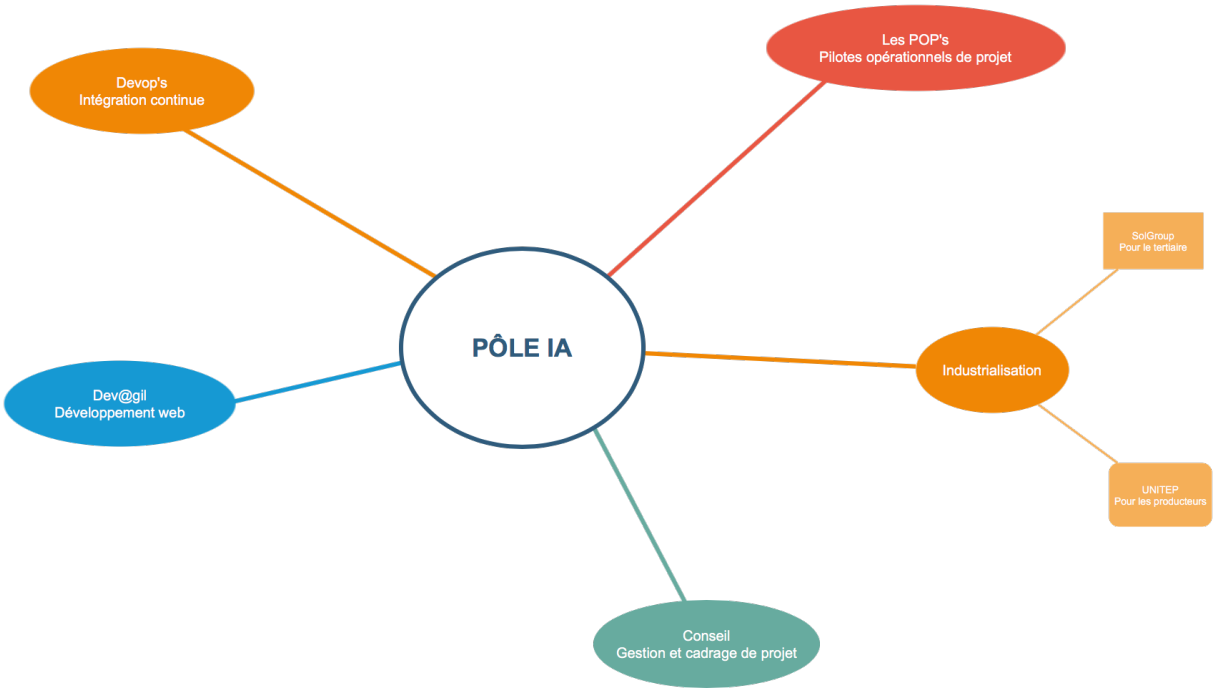

Figure 4 Écosystème autours du Pôle IA

En tant qu'alternante taliste, j'ai pris part à aux différentes étapes de plusieurs projets : de la phase de cadrage, en passant par les ateliers d'idéations jusqu'au développement de la solution et de sa maintenance. L'enjeux des premières étapes est de ne pas se focaliser sur la solution et de se détacher des intuitions que nous pourrions avoir quant au futur outil. Au contraire il faut essayer de s'abstraire de la solution outillée afin d'identifier les « points de douleurs » (« pain point ») du métier. La présence des experts TAL ou bien data scientists à ces réunions appuie le chef de projet afin de faire remonter chez le métier leurs points de douleurs afin, par exemple, d'évaluer la faisabilité des solutions.

2. Présentation du projet Gecko

La Direction du Parc Nucléaire et Thermique (DPNT) et le Pôle IA travaillent ensembles sur des projets de traitement automatique des langues afin d'analyser les retours d'expériences (REX), c'est-à-dire le rapport rédigé après une intervention en centrale nucléaire.

Le premier cas d'usage qui est la recherche par similarité d'Événements Significatifs (ES) a fait l'objet d'un pilote et est actuellement en cours d'expérimentation auprès des ingénieurs de sûreté (Figure 5) : le projet Gecko. Autrement dit, des agents d'EDF ont accès à une interface graphique qui prend la forme d'un moteur de recherche où il peut naviguer entre événements similaires, afin d'évaluer la solution un système de feedbacks fut intégré au site. Le projet sera présenté sous forme de briques modulaires et mutualisables. Il a été conduit en parallèle la prédiction de l'équipement impacté (code équipement) pour un ES donné.

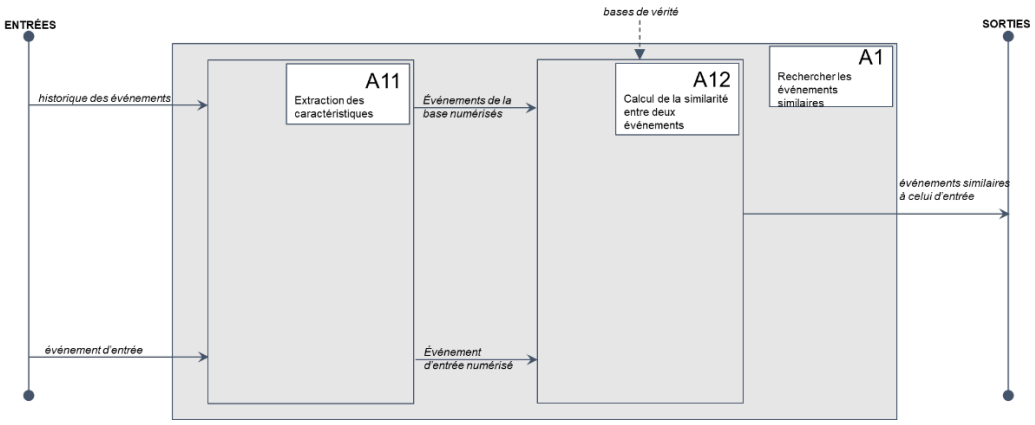

Figure 5 Vue algorithmique de la recherche par similarité d'ES

Le projet Gecko sera donc décrit dans son ensemble sous forme de briques puis nous présenterons le cadre dans lequel il se déroula. De plus, je n'illustrerai pas mes propos par des extraits de corpus pour des raisons de confidentialité.

2.1. Brique base de connaissance

Il a été choisi de formaliser le domaine métier en modélisant une ontologie. Cette dernière fut développée en OWL et hébergée sur RDF4J. Cette brique sera développée plus en avant par la suite.

2.2. Brique d'ingestion des données

2.2.1. Récupération des données

Le module permet la collecte de la donnée que la solution traitera et analysera. Dans notre cas, la donnée est issue de la base CID (la base CID est la base de donnée du nuéclaire). Durant la phase de pilote, les extraits sont faits toutes les semaines. Le format est un format tabulaire (csv).

2.2.2. Nature de la donnée

Les données récupérées sont essentiellement textuelles bien que certaines soit numériques. Les extraits de la base CID utilisés contiennent les détails de l'ensembles des REX rédigés depuis les années 90. Le format est tabulaire et contient des données de plusieurs types :

- Date : les dates de l'ES, de rédaction, d'intervention
- Textuelles : passages écrits en langue naturelle qui décrivent l'ES. Ce sont les données utilisées pour analyser et ressortir les ES similaires. Les champs utilisés sont les suivants :
    - Synthèse de l'événement : courte description de l'ES
    - Chronologie : description détaillée de l'ES
    - Libellé description brève de l'ES
    - Conséquence : description des conséquences de l'ES
    - Cause : description de la cause de l'ES
    - État initial : description de l'état physique du réacteur
- Métadonnées (variable catégorielles) : nous avons exploitées celles qui correspondent aux codifications de chaque REX, à savoir les champs :
    - Détection : code catégorisant la manière dont l'ES a été détecté
    - Code Acteur : code catégorisant les acteurs responsables de l'ES
    - Code Équipement : code catégorisant les équipements touchés par l'ES
    - Code Activité : code catégorisant l'activité ayant provoqué l'ES
    - Code Cause : code catégorisant la cause de l'ES
    - Code Évènement : code catégorisant l'événement ayant provoqué l'ES

2.2.3. Volumétrie de la donnée

Vous trouverez ci-dessous la volumétrie détaillée de la donnée traitée :

- Environ 27 000 entrées
- Dont 9 628 évènements significatifs
- Pour le calcul de similarité
  - 9 628 évènements sont comparés à 9331 évènements retenus
  - 90M de distances calculées et stockés (actuellement dans un csv, à priori dans la base de données cible).
- Pour l'ontologie
  - Pour évènement : ~30 triplets par évènement → 300K triplets
  - Similarité : 1 distance est décrite par 3 triplets → 270M triplets

2.3. Brique IA

2.3.1. Traitement automatique de la langue – TAL

Le TAL fut utilisé pour les 2 tâches : la recherche par similarité des ES et la prédiction de famille d'équipement. Comme nous l'avons dit précédemment en tant que taliste j'ai donc assisté aux réunions de cadrage du besoin afin de comprendre les points de douleurs : par la suite cela m'a permis d'extraire les entités textuelles les plus pertinentes quant à la tâche.

2.3.1.1. Recherche par similarité des ES

Numérisation du texte (Word embedding) : W2C pour l'embedding des mots (Figure 6).

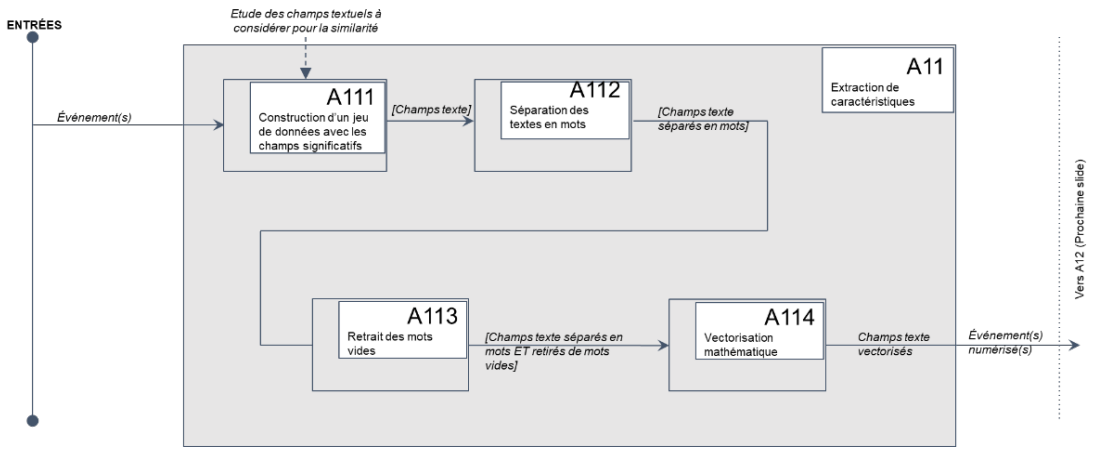

Figure 6 Pipeline de traitement pour la recherche par similarité des ES

2.3.1.2. Prédiction de famille d'équipement

Nous avons utilisé des syntagmes nominaux extraits grâce à des règles symboliques implémentées avec Unitex. Nous avons ensuite annoté le texte avec des balises qui encapsulaient les syntagmes identifiés afin de les extraire. Avec cette étape de prétraitement nous obtenons une meilleure représentation sémantique des textes : les bigrammes ou bien trigrammes permettent de garder du contexte.

2.3.2. Modèles d'apprentissages

La phase exploratoire, c'est-à-dire le temps où est recherché le modèle le plus pertinent pour la tâche qu'il doit remplir a été réalisée sur des machines locales. Cette phase est essentielle pour les deux tâches : recherche par similarité des ES et la prédiction de code famille. Le plus important fut de se concerter avec le métier afin de définir les critères selon lesquels nous allons évaluer les différents modèles :

- Sur l'ensemble des paires d'ES de la base des événements 2-0, nous avons calculé le pourcentage des évènements retrouvés dans les 15 premières sorties de l'algorithme. Le seuil à dépasser, défini par les métiers dans le cadre du pilote, est de 60%.
- Idem sur la base de vérité construite avec les métiers

Si les résultats des algorithmes implémentés n'atteignaient pas ces seuils alors ils étaient écartés. Notre travail fut donc d'optimiser les performances en jouant sur le choix du modèle et l'hyper-paramétrage des algorithmes. Dans l'optique d'une industrialisation de la solution (dans notre cas, le modèle a été déployé sur un serveur qui héberge la solution pilote), il est important de voir avec le métier les points suivants :

- La fréquence de mise à jour de la base de vérité
- La fréquence de calcul des indicateurs de performance
- La fréquence de lancement de phase exploratoire pour rechercher les modèles les plus performants
- La fréquence de déploiement de nouveau modèle.

2.3.3. Langage de programmation et librairies

Nous avons utilisé que des librairies libres de droit et traditionnelles dans les domaines de la data science ou bien du TAL :

- Des librairies natives : « re »
- Pandas
- Numpy
- Scipy
- NLTK
- Gensim

Comme indiqué précédemment nous avons également utilisé Unitex pour extraire des syntagmes nominaux.

2.4. Cadre du projet

2.4.1. Composition de l'équipe

Le projet Gecko convoqua plusieurs compétences et donc différents profils d'agents. Voici la liste des participants au projet :

- 1 chef de projet scrum master : pour gérer la relation avec le client et faire l'articulation entre les différents membres de l'équipe
- 1 product owner : l'interlocuteur métier privilégié qui a commandé le projet
- 2 data scientists : pour la partie calcul de similarité
- 1 ingénieure cogniticienne : pour le développement de la base de connaissance ainsi que pour les ateliers d'idéation
- 1 expert en base de connaissances : pour le développement de la base de connaissance
- 2 talistes (dont moi-même) : la partie prédiction de code matériel, Microservices et maintenance de l'application

2.4.2. Fonctionnement de l'équipe

Nous avons utilisé la méthode agile avec des sprints de 2 semaines. Cela nous permit d'inclure les remarques du métier. De fait, la notion même de similarité est une notion

subjective, le retour des experts nous permit de mieux comprendre dans leurs contextes métiers ce qu'ils considéraient comme 2 évènements similaires : sur quels critères ils se basent pour établir une similarité ? A partir de quel moment un évènement n'est plus similaire à un autre ? Est-ce une question de gradation ou bien simplement de critères différents sur lesquels se baser ?

De plus, tous les lundis, nous avions un point d'équipe pour faire part de nos avancements ou problèmes rencontrés.

3. Travail réalisé

L'utilisation d'un algorithme d'apprentissage dans un cadre industriel permet de ne pas savoir a priori les natures des objets à identifier. Autrement dit, les entreprises tirent parti de la capacité des algorithmes à induire des règles de détection à partir de donnée et de leur capacité à généraliser afin de détecter des évènements encore inconnus (Beaugnon 2017). Toutefois plusieurs questions se posent (Beaugnon 2017) :

- Une solution qui se base sur du machine learning peut-il faire un traitement en temps réel ?
- Les utilisateurs de la solution ne seront pas des experts en machine learning : comment faire pour qu'ils aient confiance en la solution ? Surtout si les résultats retournés vont à l'encontre de leur propre expertise ?
- L'explicabilité d'une solution peut-elle être objectivement évaluée ? Autrement dit, la transparence d'un algorithme n'est-elle pas arbitraire selon les connaissances de chaque individu quant aux algorithmes d'apprentissage ? Pouvons-nous établir des « indicateurs d'explicabilité » ?

Lors du projet Gecko nous avons croisé ces questions. De fait, j'intervins sur toutes les phases de ce projet :

- La base de connaissance
- La prédiction de matériel
- La mise en production de la solution pilote
- La maintenance de cette solution

Nous détaillerons donc ces dernières dans l'ordre ci-dessus avant d'exposer quelques remarques rétrospectives.

3.1. Ontologie

Lors du projet Gecko l'ontologie fut choisie pour 2 raisons. La première est l'ouverture : pouvoir partager les référentiels (les ontologies) et partager les données (base de connaissances en RDF) au sein du groupe EDF. La seconde est de pouvoir s'adapter à de nouveaux usages : capacité de s'adapter à de nouvelles tâches. De fait, c'est pour prouver la plasticité d'une ontologie qu'il me fut demandé de modéliser une seconde ontologie : je devais pouvoir lier les deux ontologies et faire tourner deux scripts différents sur l'ontologie fusionnée.

Cela avait pour but de démontrer que l'on pouvait utiliser une seule grande ontologie qui formalisent plusieurs domaines métiers pour plusieurs projets producteurs.

Nous rappelons que la première ontologie modélisait un domaine du nucléaire (projet Gecko), plus précisément celui des retours d'expériences des agents de sureté qui rédigeaient un rapport pour rendre compte de leur intervention suite à un évènement significatif. Ainsi, l'ontologie tournait autour de la classe centrale qui était un « évènement significatif ».

La seconde ontologie devait modéliser un domaine différent du nucléaire : les interventions de maintenance (projet Macao). Après chaque intervention les agents rédigent également une synthèse. Les données sont exportées d'une base de données différente mais sont aussi sous forme tabulaire. La tâche de ce projet était d'identifier si un remplacement de matériel avait été effectué ou non en analysant et traitant les données non structurées du tableau. Ainsi, il générait des tags qui indiquaient si le remplacement avait été effectué ou non.

Mon travail devait dans un premier temps modéliser une ontologie commune (Figure 7) aux deux projets, puis l'instancier et l'insérer dans RDF4J avant d'adapter les scripts afin qu'ils prennent dorénavant une ontologie en entrée et qu'ils insèrent les tags sous formes de triplets dans RDF4J afin d'enrichir cette dernière.

Tout d'abord, j'ai rencontré des problèmes de modélisation. La première difficulté fut de modéliser un domaine métier dans lequel je n'avais aucune expertise. Autrement dit, le fait de n'avoir aucune expertise dans le domaine modélisé m'obligea à réfléchir en termes de référentiels et non d'instances.

Puis la seconde fut de respecter la modélisation de l'ontologie déjà développée. Autrement dit, mettre le même type d'information au même format, les mêmes formulations pour les mêmes types de relations, voici quelques exemples :

- Data Property : les dates, les n_commentaires sur l'évènement
- Object Property : « x a pour y », « n appartient à z »

De plus, la contrainte de devoir faire tourner un script sur l'ontologie fusionnée m'obligea à faire attention à la modélisation. Notamment pour tout ce qui était matériel. En effet, il ne fallait pas que lors de l'instanciation les matériels touchés dans les différentes ontologies soient confondus. Pour cela j'ai concaténé à chaque instance de la classe « matériel » l'identifiant de l'évènement rapporté.

La seconde fut de trouver un point pour lier les deux ontologies de manière « naturelle », du moins pour éviter une union artificielle. Après quelques échanges avec l'ingénieure cogniticienne qui avait développé la première ontologie nous trouvâmes que nous pouvions

relier les « Interventions Fortuites » (ontologie 2) avec « Évènement significatif » (ontologie 1) car cela renvoyait à un même type d'événement : un événement non prévu (Figure 7).

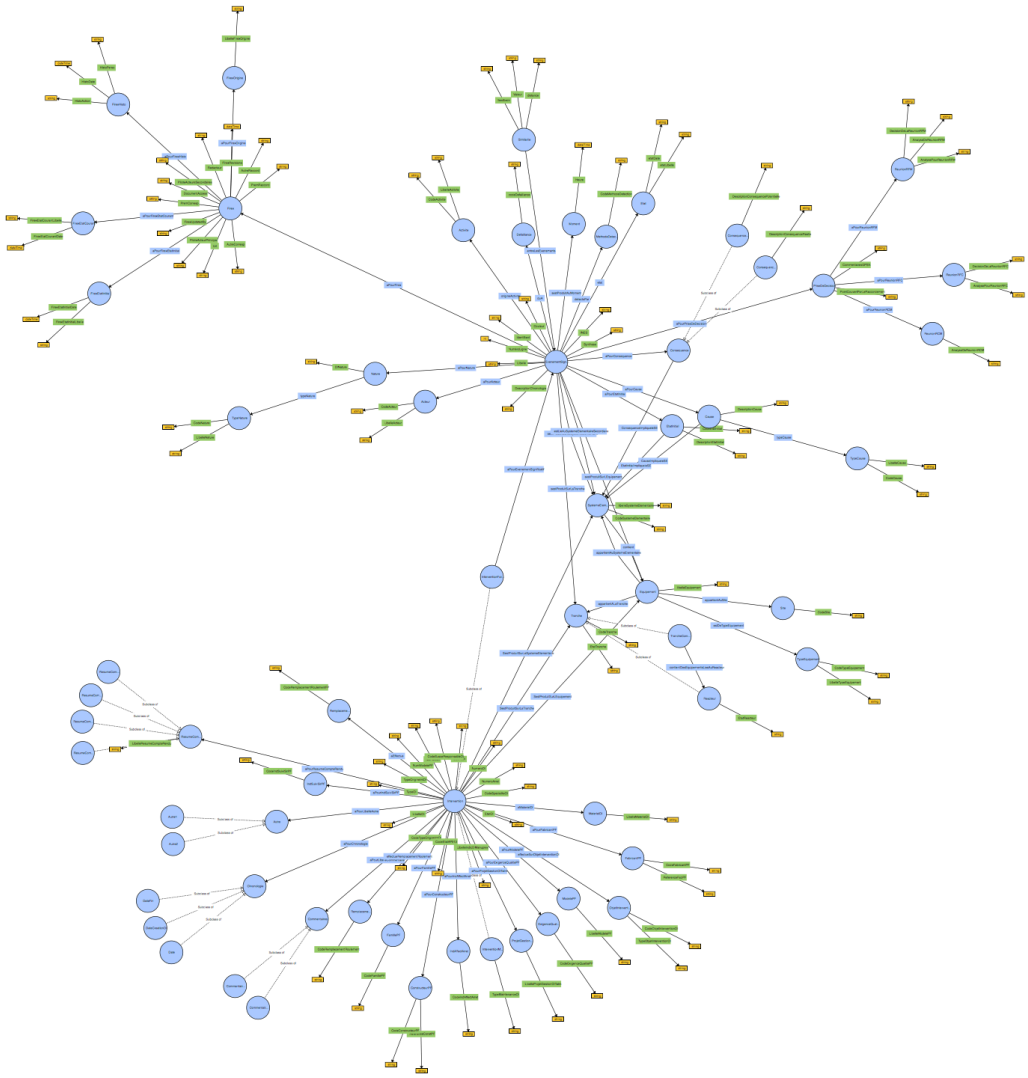

Figure 7 Les deux ontologies fusionnées

Une fois l'ontologie modélisée il fallut l'écrire et l'instancier automatiquement en l'hébergeant sur RDF4J, c'est-à-dire écrire automatiquement l'ontologie en triplet RDF. Pour cela j'ai extrait les données d'ElasticSearch (Figure 8). La seconde étape était donc d'adapter les scripts afin qu'ils prennent en entrée l'ontologie, qu'ils la requêtent (en SPARQL) pour obtenir les données textuelles nécessaires aux algorithmes puis qu'ils retournent les tags sous forme de triplets RDF avant de les insérer sur RDF4J afin d'enrichir l'ontologie (Figure 9).

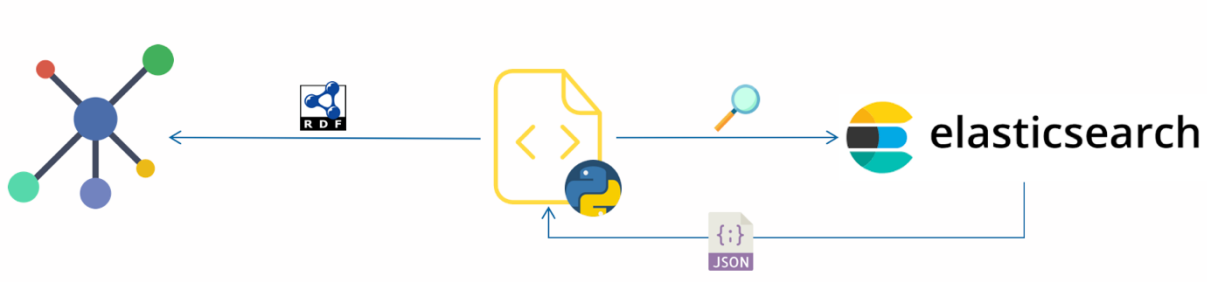

Figure 8 Instanciation de l'ontologie à partir d'ElasticSearch

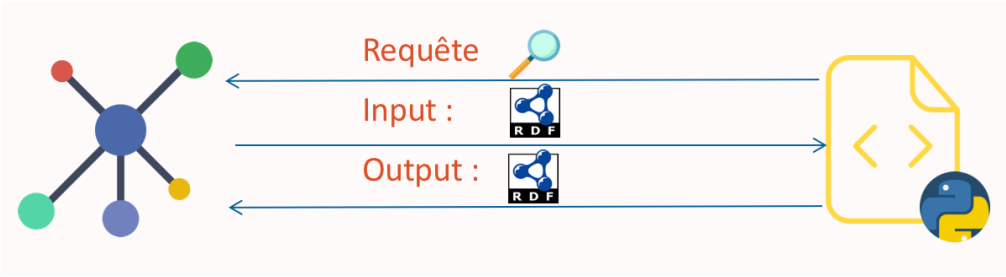

Figure 9 Enrichissement de l'ontologie avec les tags qui indiquent si un matériel a été remplacé ou non

Cette seconde partie du projet mit en lumière la difficulté de reprendre un travail déjà développé. En effet, adapter un script demande de comprendre son architecture et sa logique. De plus, l'utilisation de plusieurs logiciels et services tels qu'ElasticSearch et RDF4J sollicite la compréhension de leurs API (notamment pour la construction de requêtes) et la maîtrise de leurs « wrappers » en Python afin d'interagir avec ces dernières. Cela me permit de comprendre l'importance de documenter un script et de rester explicite lors de l'instanciation des variables. D'un point de vue méthodologique, reprendre de longs scripts avec plusieurs définitions qui s'appelaient entre elles et s'exécutaient seulement à la fin, me fit comprendre l'importance de ne définir qu'une action par fonction.

Ainsi, afin de prouver l'adaptabilité d'une ontologie dans un cadre industriel j'ai rencontré des difficultés de modélisation et d'ordre technique. Ce travail se situe dans la partie base de connaissance : comment modéliser un domaine métier ? La partie suivante exposera mon travail sur la partie prédiction de code matériel du projet.

3.2. Prédiction de code matériel

La partie prédiction de code tend à prédire quel matériel est concerné lors d'un évènement significatif. Pour cette tâche nous avons eu deux démarches en parallèle. Une première démarche était statistique avec l'utilisation d'une Machine à Support de Vecteur (SVM). La seconde était symbolique avec l'utilisation de règles expertes. Le but était de

comparer les deux afin de voir si une meilleure représentation sémantique avec des règles expertes pouvait avoir de meilleurs résultats qu'un outillage statistique plus lourd.

Nous voulions tester une méthode symbolique. De fait, j'ai écrit des règles expertes basées sur les connaissances métiers. Pour cela j'ai extrait des éléments de matériels sous forme de syntagmes (par exemple, « robinet de vannes », « le placage de l'opercule », « la commande de la vanne »). Les règles ont été implémentées avec l'aide d'Unitex (Figure 10) qui encapsulaient les entités à extraire. Ensuite j'ai extrait les entités entre les balises « <SN> » et « </SN> ». Puis, j'ai gardé que les syntagmes représentatifs de leur classe, c'est-à-dire tous les éléments qui n'étaient pas dans l'intersection des classes (Figure 11), avant de réaliser un simple compteur qui totalisait le nombre de syntagmes présents pour chaque code famille : le code famille attribué était le code qui avait le plus grand compteur.

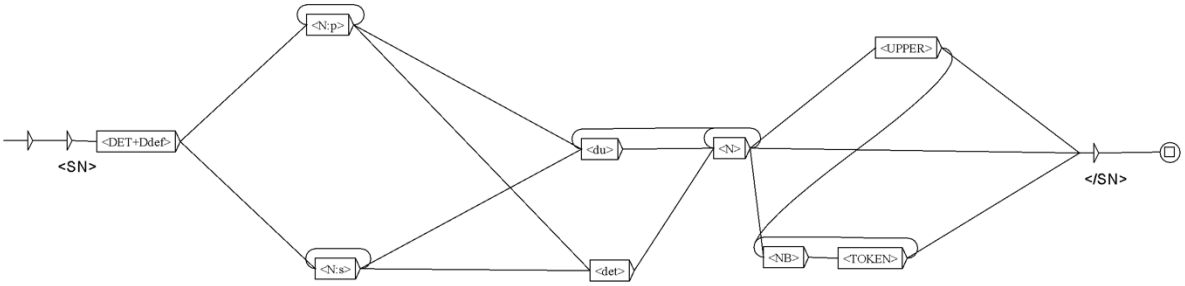

Figure 10 Automate qui encapsule un syntagme nominal

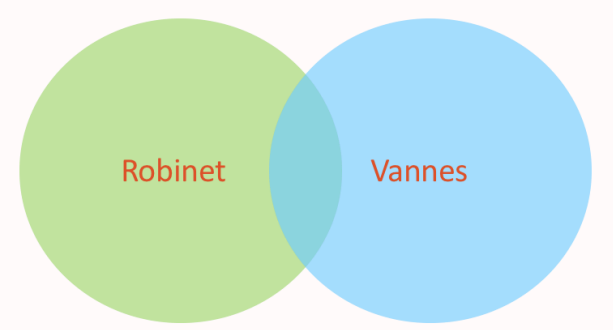

Figure 11 Extraction du lexique en dehors de l'intersection des deux vocabulaires

Les familles de codes matériels sont hiérarchiques (voir exemple ci-dessous). Pour le test, nous avons uniquement considéré la famille « robinetterie ».

- Famille robinetterie
  - Vannes
  - Robinet
  - Tuyauterie

Les résultats (Tableau 3) montre que le système expert est meilleur en rappel et précision par rapport à la SVM. Afin de nuancer ces résultats nous pouvons rappeler qu'un système expert est plus pertinent car plus spécifique. Au contraire, il ne pourra pas être facilement réutilisé pour un autre domaine métier ou bien pour une autre tâche. En d'autres termes, la SVM est moins bonne mais elle permettrait d'être plus généralisable. Lors du choix de la solution il faut donc arbitrer avec notamment la prise en compte de la tâche : cette dernière va-t-elle rester la même et les données vont-elles rester du même type ? Si oui, alors le système expert peut être mis en place, si non la SVM peut être plus adaptée.

Tableau 3 Métriques d'évaluation de la SVM et du système expert

Métriques **Synthèse de l'événement**

Champs textuel	Rappel	Précision
SVM	60	30
Règles expertes	97	65

Le pilote mis en production ne prend pas en compte cette partie : les résultats n'étaient pas suffisamment bons, mais il est envisagé à terme de l'intégrer après quelques améliorations.

3.3. Microservices

Une fois les algorithmes développés (de recherche par similarité et de prédiction de code) le Pôle a mis en place une interface graphique pour que les ingénieures de sûreté puissent tester les résultats et donner leurs retours (Figure 12, Figure 13).

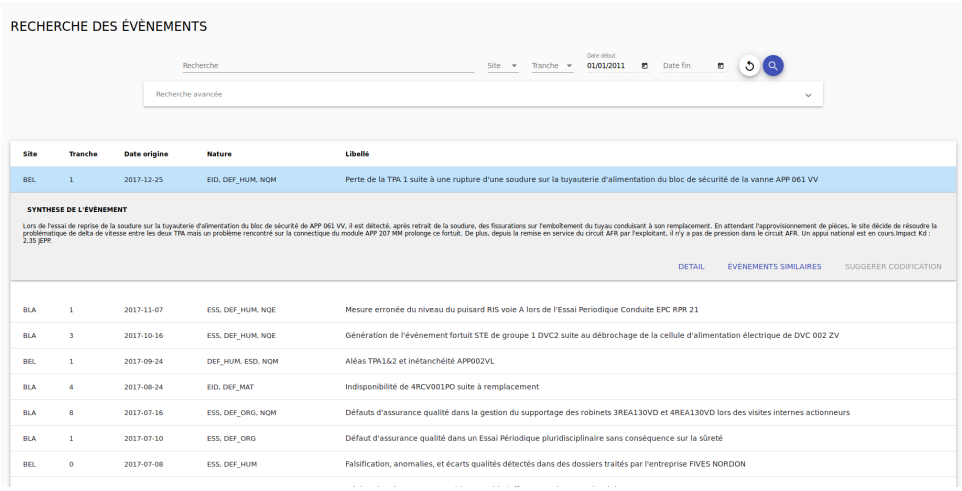

Figure 12 Application pour la recherche par similarité d'évènements significatifs

Figure 13 Interface graphique : l'option des notes utilisateurs et le choix d'affichage par taux de similarité entre des évènements

Afin d'assurer cette interface graphique le Pôle a mis en place des microservices : c'est-à-dire une architecture logicielle où les services sont indépendants les uns des autres et modulaires.

Ainsi, l'utilisateur rentre une requête avec les différents filtres proposés (date, tranche, site…) puis navigue entre les évènements similaires en jouant sur le taux de similarité (Figure 13). Aussi comment fonctionnent les scripts entre eux et comment les différents acteurs s'articulent ils ? Voici les étapes lorsqu'un agent de sûreté rentre une requête (Figure 14) :

1. Un utilisateur rentre une requête
2. Une fonction récupère la requête
3. Une fonction réécrit la requête afin de rédiger une requête ElasticSearch bien formée

4. Une fonction récupère les identifiants d'ElasticSearch et les retourne : ce sont les identifiants des évènements qui correspondent aux critères de la requête
5. Une fonction écrit une requête SPARQL afin de récupérer le contenu textuel de tous les évènements dont les identifiants sont dans la liste retournées par ElasticSearch
6. Une fonction récupère les données textuelles
7. Une fonction transmet les informations textuelles à afficher sur l'interface graphique

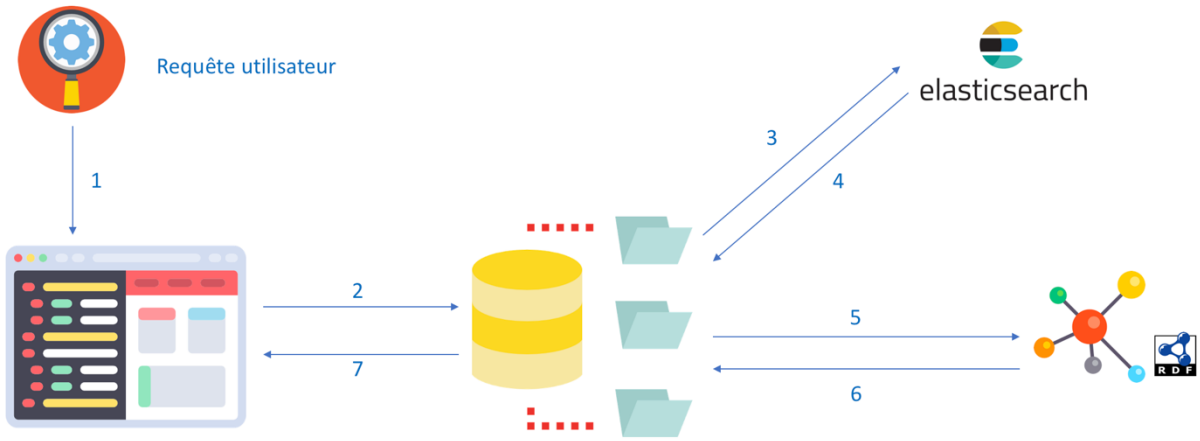

Figure 14 Architecture des microservices

3.4. Maintenance de la solution pilote

Chaque semaine un nouvel export de la base nous est envoyé. Il faut donc toutes les semaines recalculer le delta et les insérer sur RDF4J pour qu'ils s'affichent sur l'interface graphique. Pour cela plusieurs fonctions ont été écrites :

1. Une fonction qui génère un fichier avec les évènements delta (c'est-à-dire nouveaux)
2. Une fonction qui génère les triplets afin d'instancier l'ontologie avec les nouveaux documents
3. Une fonction qui réindexe le répertoire dans ElasticSearch
4. Une fonction qui retourne l'information textuelle vers l'interface graphique

3.5. Retours sur le projet Gecko

Participer au projet gecko depuis la base de connaissance jusqu'à sa maintenance de l'interface utilisateur me permit de vraiment voir chaque brique d'un outil informatique. Mon expertise en traitement automatique des langues naturelles met en avant l'importance du travail en amont de tout ce qui est algorithmique. En effet, j'ai pu prouver qu'une belle représentation sémantique des données nous permettait d'avoir des résultats corrects même avec un simple compteur. De fait, il me semble qu'avoir une sensibilité aux sciences du langage permet de mieux sélectionner les éléments à représenter par la suite et d'augmenter la performance des algorithmes.

En outre, le Pôle IA se différencie du DataLab car il traite des données non structurées, ainsi participer à l'industrialisation d'un tel projet de TAL permet de légitimer l'expertise mise en avant par le Pôle et de le démarquer d'autres unités d'EDF. Enfin, le projet Gecko donna aux agents de sûreté un outil d'aide à la décision testé qui sera mis en production en fin d'année. Nous avons d'ailleurs gagné le trophée de l'innovation aux prix DTEO EDF. En effet, le projet contient des technologies encore nouvelles et le fait d'aboutir à une solution qui fonctionne, malgré certaines difficultés, est un succès.

Par ailleurs voici quelques éléments de réponse quant aux questions de la mise en production bien qu'elles soient détaillées dans la partie suivante :

- Notre outil ne calcul pas en temps réel les nouvelles données mais est mis à jour avec une insertion hebdomadaire de ces dernières
- Les agents de sûreté semblent avoir confiance en la solution : l'interface graphique leur permet de tester les résultats et donc, de manière empirique, de comprendre comment marche l'algorithme de similarité
- Les retours utilisateurs sont stockés et permettront dans de futurs travaux une évaluation plus juste des résultats et donc de les améliorer

4. Limites et perspectives

Aussi, même si la solution mise en production fonctionne cela n'a pas été sans modifier plusieurs briques. Nous verrons tout d'abord les limites rencontrées puis comment nous les avons parées avant de terminer sur les perspectives envisagées.

4.1. Limites

Nous verrons dans cette partie les limites rencontrées lors de la mise en production de l'outil d'aide à la prédiction basée sur un algorithme de calcul de similarité qui lui-même prend en entrée des données textuelles stockées dans une ontologie hébergée sur RDF4J. Les limites quant à l'ontologie seront détaillées avant celles du modèle algorithmique.

4.1.1. Ontologie

L'ontologie ne fut finalement pas utilisée dans le projet gecko car elle était trop longue à requêter. Aussi, cela n'était pas un problème de technologie utilisée mais de serveur d'hébergement à savoir RDF4J. Les requêtes pour insérer les distances entre chaque évènement étaient trop longues. De fait, le volume de données traité est un volume à échelle industrielle donc important. Aussi, je n'enrichie l'ontologie qu'avec les données textuelles (c'est-à-dire les nouvelles instances) mais pas avec de nouvelles relations entre les classes (« x a pour distance y avec n »). De fait, insérer les triplets de manière hebdomadaire pour les distances prenait plus de 12h.

En outre, RDF4J est instable, c'est-à-dire qu'il s'arrêtait sans raison apparente et je devais le relancer manuellement dès qu'un utilisateur remontait le problème. Lorsqu'il fallait faire les mises à jour hebdomadaires avec les nouvelles données les requêtes d'insertion s'interrompaient de manière aléatoire.

4.1.2. Modèle utilisé

La distance utilisée pour l'algorithme de similarité se base sur la distance « Word Mover Distance » qui calcule la distance entre chaque mot des deux textes pour tous les textes… Dès lors, le temps de calcul était conséquent. Nous ne pouvions donc pas proposer un calcul de distance en temps réel aux utilisateurs via l'interface.

4.2. Solutions

4.2.1. Ontologie

Pour contrecarrer l'instabilité de RDF4J, j'ai installé un système de « print » et de compteur afin de savoir dans quelle classe nous étions et quel était de l'index de l'instance. Dès lors, il m'était possible de reprendre l'insertion là où elle s'était interrompue. Cela comporte une contrainte majeure : l'insertion ne peut être totalement automatisée, il faut toujours une surveillance humaine.

En outre, pour pallier de manière plus générale à son instabilité il a été envisagé d'installer une sonde qui relancerait RDF4J si ce dernier tombait.

4.2.2. Modèle utilisé

Afin d'éviter le temps de calcul trop long, nous les effectuons en local toutes les semaines et nous les déployons sur le serveur sous forme tabulaire. De fait, le format tabulaire a été choisi avant d'éviter l'insertion de nouvelles relations dans RDF4J : encore une fois le temps était trop important pour cette action. J'ai donc adapté les fonctions pour qu'elles puissent prendre en entrée des dataframes.

4.3. Perspectives

Nous venons de voir qu'à toutes les étapes d'un projet de TAL, que cela soit la base de connaissance, la mise à jour des calculs de distance ou bien la maintenance de l'interface graphique les contraintes industrielles posent problèmes.

Ces dernières sont principalement dues aux volumes de données traitées ainsi qu'au temps de calcul exigé (ce dernier doit être minime). Ce qui ressort du projet Gecko est son besoin de stabilité. Effectivement, nous avons mis en place des technologies nouvelles qui ont été peu industrialisées : la documentation et les retours d'expériences sont donc également peu nombreux. La solution est donc innovante mais instable. Cette première expérience permet de répondre à la problématique de maintenir un modèle d'apprentissage en production : nous pouvons le maintenir car ce n'est pas son état qui assure sa stabilité mais les technologies qui le supportent.

Cet axe nous permet alors de réfléchir en termes de « cycle de vie » de modèle : si la structure est fixe et les technologies stables alors le modèle peut être envisagé comme

cyclique même mis en production. L'élément clé est de bien tester le modèle dans un environnement de recette (c'est-à-dire un environnement qui permet d'évaluer les développements en vue de la mise en production) avant de le déployer en production. Ainsi nous dépassons la contradiction entre « maintenir » et « apprentissage » : il peut apprendre car ce qui est doit être maintenu dans un état donné ce n'est pas l'algorithme mais ce qui lui permet de fonctionner, l'environnement applicatif (software et hardware).

Cette réflexion a amené le Pôle à concevoir à une « plateforme de TAL », c'est-à-dire une chaine de traitement modulaire et modulable selon les projets où des briques seraient préécrites mais toujours adaptables (Figure 15).

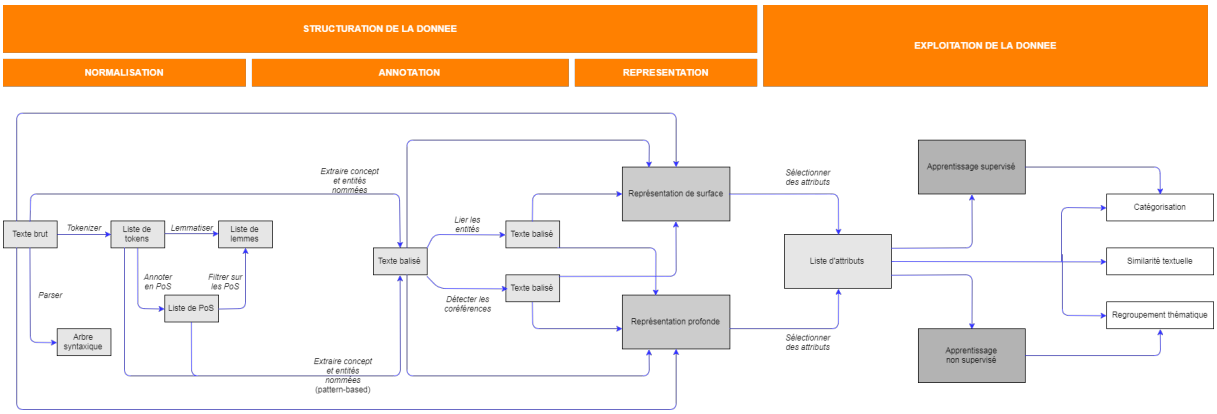

Figure 15 Modélisation d'une chaine de traitement en Traitement Automatique des Langues pour la « Plateforme TAL »

Une telle plateforme permettrait alors d'avoir les meilleures pratiques avec les bonnes technologies pour construire un outil de TAL robuste et industrialisable.

Conclusion

Nous nous demandions comment maintenir dans un état donné un modèle évolutif. Pour cela nous avons pris le cas d'usage Gecko puisque mon projet d'alternance couvrait toutes ses étapes : depuis sa base de connaissance jusqu'à sa maintenance en production. Suite à la description des problèmes rencontrés nous concluons sur l'importance du socle technologique. De fait, nous pouvons dépasser la contradiction entre « maintenir » et « apprentissage » car ce qu'il faut garder inchangé c'est l'infrastructure technique afin d'avoir la stabilité nécessaire à l'industrialisation. En revanche, le modèle lui peut évoluer.

De plus, le projet Gecko est apprécié par les utilisateurs et représente un véritable outil pour les agents de sûreté. En outre, il me semble, que nous avons su développer une solution innovante qui permet de faire rayonner le Pôle au sein de l'entreprise et de participer à sa transformation digitale. Il me semble que les réflexions engagées lors de ce projet par les membres de l'équipes aboutiront à un outil d'aide aux développements pertinents (la plateforme TAL).

De manière plus générale, ce projet me permit de comprendre l'importance d'une bonne relation avec le client car c'est lui qui nous transmet le savoir à modéliser et c'est pour lui que nous développons l'outil final. En outre, évoluer dans le Pôle Intelligence Artificielle mit en exergue l'articulation entre la data science et le TAL : le TAL intervient beaucoup durant les phases de pré-traitement afin d'avoir une belle représentation sémantique des données ainsi que dans la partie ingénierie des connaissances. La partie data science, davantage itérative, développe et compare plusieurs modèles afin de sélectionner le meilleur.

D'un point de vue scolaire, j'ai pu durant ces derniers mois articuler différents cours que nous avons eus (ingénierie des connaissances, programmation algorithmique et objet, base de données et web dynamique…). Pouvoir les appliquer concrètement me donna l'occasion d'en comprendre les enjeux.

Enfin, cette année d'alternance m'apprit autant sur le plan humain que professionnel. Évoluer dans une équipe pluridisciplinaire exige d'être à l'écoute et à toujours remettre en question ses approches : j'ai beaucoup aimé défendre mes points de vue et apprendre de ceux des autres. Je repars non seulement avec une vision riche de ce qu'est le traitement automatique des langues naturelles, mais également avec la capacité de projeter ses application concrètes et la manière de les industrialiser.

Références bibliographiques

DE SAUSSURE, Ferdinand, BALLY, Charles, et KOBAYASHI, Hideo. *Cours de linguistique générale*. impr. Darantière, 1933.

BEAUGNON, Anaël, HUSSON, Antoine Husson. *Le Machine Learning confronté aux contraintes opérationnelles des systèmes de détection*. SSTIC 2017: Symposium sur la sécurité des technologies de l'information et des communications, Jun 2017, Rennes, France. pp.317-346. <hal-01636303>