Détection et caractérisation d'événements dans des rapports de maintenance

MASTER TRAITEMENT AUTOMATIQUE DES LANGUES

PARCOURS : INGENIERIE MULTILINGUE

PAR RENISE PIERRE

DIRECTEUR DE MEMOIRE :

DAMIEN NOUVEL

ENCADREMENT DANS L'ENTREPRISE :

AUDE VINZERICH

ANNEE UNIVERSITAIRE 2016/2017

Résumé

Dans un contexte d'exploitation des retours d'expérience à des fins préventives, ce travail porte sur la détection de la réalisation d'événements dans des rapports de maintenance. Partant du constat qu'une simple recherche par mots-clés ne suffit à détecter leur réalisation, la chaîne de traitement présentée intègre des méthodes symboliques. Elle se concentre sur le problème du traitement de mention d'événements non-accomplis (négations et futur), ainsi que celui de la construction de ressources terminologiques (sous forme d'ontologie) adaptées à ces textes non formalisés. Ce travail porte également sur l'apport possible de méthodes statistiques en utilisant notamment l'apprentissage automatique pour la classification de ces événements selon un critère lié à l'anticipation des actions de maintenance.

Mots-clés : *recherche d'information, retour d'expérience, ontologie, terminologie, apprentissage automatique, classification*

Table des matières

I- Contexte

Ce travail s'inscrit dans le cadre de la capitalisation du retour d'expérience (**REX**) dans l'exploitation et la maintenance nucléaire de l'UNIE (Unité d'ingénierie d'exploitation) pour laquelle le pôle Intelligence Artificielle de la Direction des Partagés (DSP) et la branche R&D d'EDF ont été sollicité pour leur expertise en fouille de texte.

La capitalisation du retour d'expérience est le fruit chez EDF d'une « culture sureté » dont l'importance a été soulignée à la suite des événements de Tchernobyl. Elle repose sur une constante démarche d'interrogation, d'amélioration et de communication entre tous les acteurs du domaine.

II- Besoin

La base **SYGMA** recense des données de maintenances réalisées depuis quelques décennies déjà. Dans une optique d'amélioration continue de la performance, un retour d'expérience est réalisé. Il s'agit d'analyser les situations passées pour en partager les enseignements, anticiper des problèmes génériques et de restituer l'ensemble aux acteurs concernés. Il est nécessaire pour cela de parcourir des grandes quantités de comptes-rendus, écrits en langue naturelle. Il a donc été décidé d'avoir recours aux méthodes de fouille de texte pour la capitalisation des informations qui y sont contenues.

Le but de cette étude est, pour un premier cas d'application (le remplacement de roulements) de vérifier la réalisation de l'action. Si l'outil n'arrive pas à déterminer l'état de l'action, il doit être capable de placer le compte-rendu en zone grise. Des techniciens, familiers avec ces équipements, seront alors amenés à ajouter cette information manuellement. Bien sûr, cette zone grise ne doit pas être utilisée à outrance, le but étant avant tout d'alléger la charge de travail ces techniciens. Il serait contre-productif de leur imposer un long travail d'annotation avec un gain de temps par conséquent minime. Pour référence, le corpus annoté manuellement qui servira de référence d'évaluation aurait pris environ deux semaines à être construit.

L'information extraite pourra alors être intégrée aux éléments pris en compte dans les algorithmes visant à déterminer quand une pièce donnée doit être remplacée. L'enjeu étant de pouvoir anticiper la casse de ces composants identifiés comme risques potentiels de

défaillances (leur durée de vie serait en corrélation avec les bonnes conditions de fonctionnement d'un système) et ainsi éviter un manque à gagner conséquent qu'elle peut entrainer (immobilisation partielle ou totale de la production) tout en gardant un rythme de remplacement raisonnable. Ces informations serviront également à évaluer les coûts de maintenance imputable à chaque réacteur.

Deux autres études menées par une société privée et la R&D d'EDF et partageant le même objectif ont été lancées en parallèle afin de confronter plusieurs outils, d'en voir les forces et faiblesses et d'obtenir plusieurs points de vue. Les outils choisis côté EDF, **GATE et NLTK**, ont été sélectionnés suite à l'étude du département ICAME sur les outils de text-mining open-source. L'étude décrite ici porte sur le travail réalisé sous NLTK.

III- Contraintes

Les contraintes pour la réalisation du projet sont essentiellement liées à la plateforme utilisée: NLTK. Mais l'objectif final étant d'avoir un outil s'intégrant à la plateforme d'analyse existante et pouvant être éventuellement manipulé par les acteurs réalisant le REX, pour la plupart non-initiés au TAL, la facilité d'usage, d'édition et la rapidité de l'outil sont également pris en compte dans son évaluation.

IV- Description du corpus

SYGMA contient approximativement 7 millions d'entrées pour des interventions réalisées depuis 1989. Les expérimentations menées dans cette étude ont été réalisées sur un extrait contenant en 7900 entrées environ.

Ces items contiennent en tout 43 champs dont 8 rédigés en langue naturelle. Il s'agit notamment d'informations concernant la date de demande d'intervention, sa localisation et des détails sur le matériel et le réacteur concerné. L'analyse textuelle de ce corpus ne portera que sur les champs LIBELLE-DI, LIBELLE-OI (DI = demande d'intervention, OI = ordre d'intervention) ainsi que sur les comptes-rendus, répartis sur 4 champs. Il faut noter que ces entrées n'intègrent pas systématiquement un compte-rendu. Il arrive qu'il n'ait pas été rédigé car jugé inutile, par manque de temps ou parce que l'opération a été annulée.

Les contraintes de temps expliquent également le côté très synthétique des comptes-rendus effectivement présents. Les interventions le nécessitant seront complétées par un autre rapport. Il n'existe en revanche pas de contrainte liée à l'espace (nombre de caractères, etc.).

V- Définition du sujet

Cette étude se concentre sur l'application de méthodes symboliques pour la détection de réalisation de l'événement. Deux éléments principaux seront abordés : la détection de « mentions » grâce la construction de ressources terminologiques et d'autre part la vérification de la réalisation en s'intéressant l'emploi de la négation et du futur qui peuvent venir annuler la valeur de ces mentions.

Pour la construction des ressources, il s'agira de recenser et d'organier les expressions utilisées pour décrire ce qui est recherché. En effet, il existe pour un élément donné une multitude de façons de s'y référer. Ce nombre se voit augmenter de par la nature du texte qui n'est ni normalisé ni relu, ce qui rend la problématique de construction de ces ressources d'autant plus intéressante. Concernant la négation et le futur, il s'agira avant tout de construire des ressources linguistiques. On se concentrera sur les règles établies pour les modéliser aussi bien que sur le vocabulaire employé. Nous discuterons en premier lieu de travaux pertinents pour traiter ces points dans un état de l'art sur la recherche d'information et la négation.

Nous allons ensuite traiter de la caractérisation de ces événements selon un critère propre au projet dans lequel s'inscrit cette étude. Il s'agira d'extraire si, quand une action est décrite, elle a eu lieu a titre préventif ou « fortuit », une information qu'il sera plus difficile d'observer. Plusieurs méthodes seront présentées pour traiter de cette problématique de classification.

Cet état de l'art traitera de recherche d'information et, dans la mesure où il s'agit d'un des déclencheurs de ce projet, de la négation et de sa portée. Ces problématiques s'avérant pertinentes dans le cadre de sujet mais également au cœur des enjeux du traitement automatique des langues.

I- Recherche d'information

Les méthodes implémentées ici consisteront à retrouver des informations structurées depuis un corpus, soit de la recherche d'information (RI). On s'intéressera donc à des travaux concernant la construction de moteurs de recherche.

Automatiquement apporter une réponse à une question est une problématique difficile lorsque l'on dépasse les mécanismes à base de mots-clés. **[Zweigenbaum et al, 2008]** décrit un système de question-réponse dans lequel il définit 3 étapes: l'analyse de la question, le traitement des documents et l'extraction de la réponse. Le traitement des documents correspond à la sélection de ceux pouvant éventuellement contenir la réponse via un moteur de recherche grâce à une analyse linguistique de masse mais relativement courte. Il rappelle la nécessité d'avoir des connaissances sur la langue traitée pour fonctionner même dans le cadre d'approches comprenant peu d'analyse linguistique. Si la sélection de documents par des méthodes sacs de mots, via un calcul de distance (cosinus), est efficace, pour la sélection de phrases et par conséquent de réponses, elle filtre trop d'informations importantes**.** On peut aussi s'appuyer sur la redondance des données sans utiliser de connaissances linguistiques mais c'est une méthode qui s'avère limitée si ces données ne contiennent pas une multitude de reformulation de la réponse attendue. D'où la nécessité de cet apport sur la langue.

Parmi les méthodologies d'analyse linguistique traditionnellement proposées, on retrouve celles consistant à réaliser du regroupement sémantique. Pour **[Tannier, 2006]** le regroupement via racinisation (ou *stemming*) a une efficacité limitée car il a un fort risque d'effets indésirables réduisant la précision des résultats tels que des mots au sens éloignés avec une même racine (« chevalier » et « chevalet » par exemple). La lemmatisation en revanche, fournit une analyse plus fine mais ne résout toujours pas les problèmes de synonymie (échange = remplacement) et de polysémie (construction : bâtiment/action de construire). Pour pallier à ça, Tannier propose d'enrichir les requêtes en utilisant également des mots ayant des sens proche ou en lien avec les mots initiaux ou de les désambigüiser

grâce à des bases telles que Wordnet. Les résultats sont cependant peu satisfaisants, c'est pourquoi on utilisera une ontologie plutôt que ce processus expérimental.

En dépit des méthodes énoncées ci-dessus, le traitement automatique du langage dans la recherche d'informations reste souvent cantonné à des domaines bien spécifiques. Dans le cas de documents techniques comme c'est le cas ici, cela peut être avantageux mais il est bon de noter que le machine learning pourrait permettre de diminuer le temps d'adaptation à d'autres domaines, selon [Zweigenbaum].

Puisque le machine learning permet la prise en compte de beaucoup de critères en même temps, on songe donc à combiner cette approche avec la lemmatisation en vue de rassembler ce qu'il y a de meilleur dans chaque méthode. Mais, il faut noter que la lemmatisation présente toujours des résultats moins bons pour les textes contenant des scories des mots inconnus, son apport est donc limité. C'est néanmoins une approche efficacement appliquée aux extracteurs d'entités nommées à base de CRF[1]. Le texte est alors accompagné d'un set d'autres données dont les lemmes et les étiquettes morphosyntaxiques.

On retrouve pour toutes ces méthodes des problèmes similaires que sont la portée de la négation, la polysémie et les anaphores. Si la polysémie ne fait pas partie des grandes préoccupations de ce projet (dans ce contexte un pompe sera toujours un équipement, jamais un exercice physique), il sera important de s'attarder sur la problématique de la négation

II- Négation et sa portée

Ce projet se confronte à un cas particulier de recherche d'information qu'est la recherche d'information booléenne. Il est ainsi nécessaire de pouvoir jauger parfaitement si lorsqu'un document mentionne un événement, il s'agit d'une occurrence passée, future ou d'une occurrence niée.

On s'intéressera tout particulièrement aux occurrences niées et aux problèmes qu'elles soulèvent. Cet état de l'art ne prétend pas résoudre cette problématique de la négation mais simplement relever les travaux réalisés ayant permis d'avancer vers cette voie et qui auront participé à la résolution de ce cas d'application.

[1] CRF : *conditional random field*, champ aléatoire conditionnel ; un modèle statistique à base de graphes non-dirigés

On distinguera d'abord la négation syntaxique de la négation sémantique. Cette dernière se rapporte à l'antonymie : le plus souvent grâce à un préfixe, on nie le sens d'un mot souche (intéressant/inintéressant, bon/mauvais) **[Muller, 2008]**). Si les deux cas sont bien pertinents dans l'analyse à venir, la négation sémantique ne concerne qu'une part très réduite du vocabulaire utilisé (annulé, déprogrammé, etc.) qu'on aura préféré lister dans lexique par souci de temps de performance de l'outil. La négation syntaxique en revanche est une problématique non résolue et bien plus importante à cerner pour ce cas d'application.

On aura donc bien fait de recenser les marqueurs de négaton existants en français et d'observer s'ils se manifestent dans le corpus et le cas échéant de quelle façon pour les modéliser ensuite au mieux dans le programme. En raison du grand nombre de phrases nominales et du style d'écriture, la liste de marqueurs traités est finalement assez réduite : « ne », « non », « pas » et « aucun » ainsi que leurs variations. La négation ayant différentes **valeurs** ([Muller, 2008]), on se focalisera sur celle de fausseté, les valeurs de correction, réinjection, etc. étant plus difficiles à modéliser sur un texte non normalisé comme celui qui est traité.

La fausseté pose déjà un certains nombre de problèmes : la langue française est au cœur d'un processus de suppression du clitique préverbal « ne ». Cette disparition s'avère être constitutive d'un cycle qui nous ramène aux différents créoles qui sont à la base du français d'aujourd'hui **[Larrivée, 2004]**. Ce changement s'est d'abord opéré dans le langage familier et est en cours de généralisation. Il a pour conséquence de rendre impossible l'identification d'une phrase négative sur la base symétrie de la négation seule.

Il existe par ailleurs un autre cas de négation qu'il sera important de traiter mais qui risque d'être difficile à identifier également, la négation dite « externe » **[Moeschler, 2013]**. Jusqu'à présent on se concentrait sur les cas pour lequel l'adverbe de négation et ce sur quoi il porte étaient dans la même phrase. Il existe cependant des cas où la négation porte sur le contenu d'une phrase adjacente. C'est un cas que l'on rencontrera fréquemment dans la mesure où chaque compte-rendu a un libellé que le compte-rendu viendra parfois contredire**.**

« DEBRANCHMENT POUR VISITE.» (Libellé)

« NON REALISE IMPOSSIBLE (…)» (Rapport)

La seule solution pour traiter ces cas sera de s'appuyer sur une négation explicite dans le contenu d'autres énoncés.

Enfin, on mentionnera la négation métalinguistique, où un énoncé vient nier une présupposition *(« Anne n'a pas trois enfants, **elle en a quatre** »*) et qui peut aussi être considérée comme une forme de négation externe. Mais il ne s'agit pas là d'un cas particulièrement intéressant pour cette étude.

III- Positionnement et bilan

Tous les points relevés dans les travaux cités ne seront pas bons à prendre : sans être rédigé en langage SMS, le style d'écriture de ce corpus de travail est plutôt télégraphique, de plus il n'est pas relu ce qui rend difficile l'application de certains de ces modèles.

Par ailleurs, toutes les formes de négation n'y sont pas présentes. Les cas d'enchâssement où la négation est imbriquée dans des subordonnées (« Les enfants **qui ne sont pas gentils** ») sont rares. Idem pour la polyphonie : celle-ci exprimant un désaccord, elle n'a pas lieu d'être ici car on a généralement affaire à un seul et même auteur pour un rapport donné. Il s'agit aussi typiquement de textes dans lesquels on trouve des phrases nominales et peu de négations symétriques. Nous nous inspirerons néanmoins de ces analyses exhaustives pour développer une première méthodologie adaptée aux spécificités du corpus de travail et en recouvrant les cas récurrents.

I- Ressources linguistiques

a. NLTK - Grammaire

NLTK[2] rassemble plusieurs librairies Python open-source de traitement automatique du langage naturel, permettant ainsi de réaliser un grand nombre de manipulations de façon centralisée et dans un même langage.

Outre les paquets dont l'utilisation sera décrite plus loin dans ce papier, cette plateforme sera notamment utilisée pour écrire une grammaire du français qui servira à identifier quels patrons morphosyntaxique seront jugés comme appartenant à une même phrase ou à un même groupe.

b. TreeTagger - Etiquetage morphosyntaxique

Treetagger est un outil développé par l'université de Munich permettant d''annoter du texte avec les informations concernant les catégories grammaticales (accompagnées d'informations de temps pour les verbes conjugués) et les lemmes. Il est fondé sur un modèle probabiliste entrainé au préalable et a été choisi pour la qualité des ressources disponibles en français.

Pour les besoins de cette étude, la librairie Python **Treetaggerwrapper** a été utilisée afin de pourvoir communiquer avec NLTK et tous les autres éléments du script final. Elle reprend toutes les fonctionnalités de l'outil, modulo quelques différences. Ces différences concernent notamment l'annotation d'adresses e-mail, d'URL mais aussi, ce qui nous concernera davantage, le comportement de l'outil face a des mots inconnus.

Avec Treetaggerwrapper plus souvent qu'avec sa version originale, un mot qui « n'existe pas » ne sera presque jamais tagué comme *Unknown* (l'étiquette indiquant l'inconnu). L'outil va malgré tout apposer une étiquette morphosyntaxique qui sera malheureusement souvent erronée. En ce qui concerne la lemmatisation en revanche, les deux outils fonctionnent de façon identique.

[2] NLTK : *Natural Language Toolkit*

c. Interaction des outils

Treetagger permet de récupérer pour chaque mot du corpus sa catégorie grammaticale et ainsi utiliser une grammaire. Y sont défini les groupes verbaux, adverbiaux, adjectivaux, nominaux et prépositionnels. La définition des ces groupes est assez souple, en particulier pour les groupes nominaux qui peuvent avoir comme noyaux aussi bien des noms que des abréviations, symboles, etc.

La succession de l'ensemble de ces groupes syntagmatiques peut constituer un énoncé (CLAUSE). Ce découpage est déterminant, notamment pour décider de l'étendue des négations et des autres règles. Simplifiée, cette grammaire, conçue pour le projet, définit un énoncé comme une série groupe nominaux et verbaux se terminant par une ponctuation finale (., ?, !).

```
GADV: {<PUN>*<ADV>+<PUN>*}
GADJ: {<PUN>*(<GADV>*<ADJ><KON>?)+<PUN>*}

GN: {<PUN>*(<DET|NUM|PRO|PPER>?<GADJ>?<NOM|NUM|PRO|PPER|SYM|ABR><GADJ>?<KON>?)+<PUN>*}
GN: {<PUN>*<GN><KON><GN><PUN>*}
GP: {<PUN>*<PRP><GN|NUM|GV><PUN>*}

GV: {<PUN>*<VER><GN|GP|GADV|GADJ|CLAUSE>?<PUN>*}
CLAUSE: {<G.*>+<SENT>*}
```

« FABRICATION D'UN FLASQUE ET REMONTAGE SUR MOTEUR AVEC REMPLACEMENT DU ROULEMENT. » ne constitue ainsi qu'un seul énoncé malgré ses deux propositions bien délimitées. Si cela ne génère généralement pas d'erreur on pourrait tout de même obtenir un découpage plus raffiné en utilisant la proposition plutôt que la phrase comme unité avec une grammaire plus poussée. Cela permettrait de mieux traiter des cas d'énumération et de mieux cerner la portée des négations. Cependant, définir une proposition dans un tel contexte est d'un autre niveau de complexité et ne fera pas l'objet d'une étude ici.

II- Gestion des ontologies

Pour décrire les actions et éléments de maintenance, une ontologie est construite au format **OWL** (*Web Ontology Language*). Il s'agit d'une base de données structurant les concepts d'un domaine. Elle peut reprendre comme un thésaurus les relations de synonymie et d'hypéronymie mais elle structure surtout les relations entre ces différentes notions.

Ce format est indéniablement plus pratique qu'une structure plate telle qu'un lexique car hiérarchisé. De plus, les ontologies offrent la possibilité de créer des liens et inférences entre les éléments.

C'est avec le logiciel NEONToolkit que cette ontologie a été construite. Si Protégé a d'abord été considéré comme le logiciel par défaut car très ergonomique, il a été observé qu'il ajoutait fréquemment des balises superflues gênant le parsing par la librairie Owlready. NeonToolkit a donc été favorisé en dépit d'une gestion aléatoire des caractères non ASCII (qui auront finalement été ajoutés dans un éditeur de texte plutôt que dans le logiciel).

a. Ressources techniques

La base pour la création des ressources sémantiques est une note technique interne décrivant le programme de maintenance préventive des moto-ventilateurs du système élémentaire à l'étude. Cette note présente l'ensemble des procédures pouvant être mises en œuvre lors d'opérations de maintenance, leur périodicité, le matériel concerné et la durée desdites opérations. Ce sont les opérations et le matériel qui sont modélisés dans l'ontologie.

b. Description du modèle

Chaque action (Remplacement, Graissage, etc.) représente une classe dans l'ontologie et pour chacune de ces classes on retrouve des instances : des mots appartenant à la classe car possédant le même sens. Ce seront soit des synonymes, soit des variantes mal orthographiées de ces synonymes et du mot représentatif de la classe.

Le vocabulaire marquant l'absence d'actions notables pour le cas d'application (graissage, constat, visite de type A, etc.) est également consigné mais ne sera pas utilisé d'où l'absence pour ces éléments de développement en termes d'instances. Il est néanmoins conservé dans l'éventualité où elles feraient l'objet d'une recherche ultérieure. De plus il est envisageable de les utiliser pour arriver plus rapidement à un label **Non-réalisé**.

Pour le cas des termes englobant d'autres termes (typiquement moto-ventilateur qui inclus un moteur, un ventilateur et où ce dernier inclus un roulement par exemple) l'utilisation d'instances pose un problème car inexacte. On préfère alors créer une liste d'attributs créée spécialement pour l'ontologie. C'est d'ailleurs là tout l'intérêt d'OWL : étant un dérivé d'XML, la syntaxe est extrêmement souple.

Ces attributs que l'on a nommé « possede », ont donc été ajoutés pour chacune des instances de cette classe. Ils décrivent les relations de méronymie (partie-tout) entre les éléments de l'ontologie. Par exemple : « moteur-ventilateur » n'est pas une instance de *Roulement* mais « moteur-ventilateur » a *Roulement* parmi ses attributs « possede ».

c. OWLready

La librairie **Owlready** intervient ensuite. Elle permet d'utiliser et d'appeler le contenu d'une ontologie OWL comme s'il s'agissait d'objets ou de classes. Grâce à ce module, chaque opération constituant une classe dans l'ontologie constitue également une classe pour le script. Nous en respectons donc la notation traditionnelle. Cela peut sembler tenir de l'ordre du détail mais cela impacte la façon dont les actions et pièces recherchées doivent être donnés comme argument au script. Il utilisera le nom de la classe qui correspond à l'action recherchée avec une majuscule. Idem pour la pièce recherchée. Il sera ensuite aisé de vérifier l'appartenance de chaque mot aux classes créées, le contenu des attributs « possede » et ainsi enclencher l'étiquetage lorsque les conditions sont remplies.

III- Création de la base de connaissance

Le but de cette étape est de recenser le vocabulaire concernant les pièces, les actions et le type d'actions à travers la construction de lexiques et de l'intégrer dans la structure ontologique décrite précédemment. Il s'agit notamment de synonymes et d'abréviations, d'expressions (ex : « *checkup complet* », « ventilo *moteur* »).

Une revue de la documentation existante sur les appareils est réalisée. Des **entretiens** ont également été menés avec les experts à mêmes de pouvoir renseigner ces informations. Le cas sur lequel ce pilote est lancé se limitant à un équipement et un seul événement (remplacement de roulement), la contribution demandée est minime (deux entretiens environ) mais extrêmement importante pour la réussite du projet.

a. Manuelle - Etude préparatoire

TXM est un logiciel de textométrie qui prend en entrée un corpus (aux formats XML, txt, etc.) et réalise dessus un certain nombre d'opérations telles que de l'extraction terminologique.

Grâce à TXM, il est possible de classer les mots selon leur fréquence et de mettre en avant les collocations d'un mot donné. En se focalisant sur les termes de la recherche (remplacement et roulement), on peut dans un premier temps faire ressortir quelques synonymes et variations orthographiques.

Cela aura également permis de corroborer des hypothèses formulées à partir des informations données par le métier et de constater d'autres points ; notamment concernant les abréviations utilisées pour parler des actions et de l'équipement. Le métier nous aura par exemple listé « échange », « rplt ». En s'intéressant aux co-occurrents de roulement, on

constate que « rplt » n'est pas la seule abréviation utilisée pour remplacement mais que « rempl » et « rplct » aussi.

L'outil n'est pas destiné à faire une véritable étude des synonymes mais aura permis d'être mieux accompagné lors des entretiens avec le métier afin de savoir sur quels termes très fréquents il fallait absolument les interroger.

b. Via word-embedding (word2vec)

La méthodologie proposée ici ne concerne pour le moment qu'un cas d'application précis. Cependant il est amené à être élargi à d'autres. Le relevé manuel de termes à intégrer dans l'ontologie étant une étape particulièrement coûteuse en temps et à répéter pour chaque nouvel élément à détecter, on cherche à implémenter une solution de population de l'ontologie semi-automatique. Il sera donc nécessaire de traiter les problèmes de synonymie et de fautes d'orthographe automatiquement.

Des outils, tels que Word2Vec **[Mikolov et al., 2013]** se développent de plus en plus pour ce faire. Cet algorithme détermine pour chaque mot composant un document ou corpus volumineux un vecteur lui correspondant en fonction des contextes dans lesquels il apparait. Ainsi, il est possible pour un mot donné de récupérer tous ceux dont le vecteur est proche par calcul de distance.

Un tel outil risque cependant de ne pas faire de différence entre « courroie » et « roulement » par exemple puisqu'ils auront souvent des contextes similaires or ce n'est pas le genre de rapprochement qui est souhaité car ces mots ont des sens bien différents.

La connaissance métier reste donc la plus fiable. D'autant que les expressions peuvent varier selon la centrale ou le type d'équipement (ex : visite de type A vs visite de type 1). C'est cependant une méthode qui a été **expérimentée**[3] car elle peut faire gagner un temps considérable avec une certaine préparation.

La préparation consiste à présélectionner un certains nombre de termes désignant les objets et actions recherchés. Word2vec renvoie ensuite les 20 mots dont le vecteur est le plus proche pour chaque candidat entré. Ces candidats, une fois validés, sont intégrés à l'ontologie comme étant des instances du mot-candidat. Sont validés les mots pour lesquels la similitude est confirmée manuellement.

[3] L'ontologie qui en résulte n'est pas celle utilisée dans la version finale

L'implémentation de Word2Vec employée provient de la librairie gensim avec les paramètres suivants : on ne considère un mot que s'il apparait au moins **20 fois** dans le corpus (à l'exception des mots-outils), on considère les **15 mots voisins** dans la construction du vecteur, qui est de dimension **200** afin de conserver une grande quantité d'information. On utilise le modèle **Skipgram.** Le modèle aura été appliqué sur la totalité du corpus soit un total de 10000 documents.

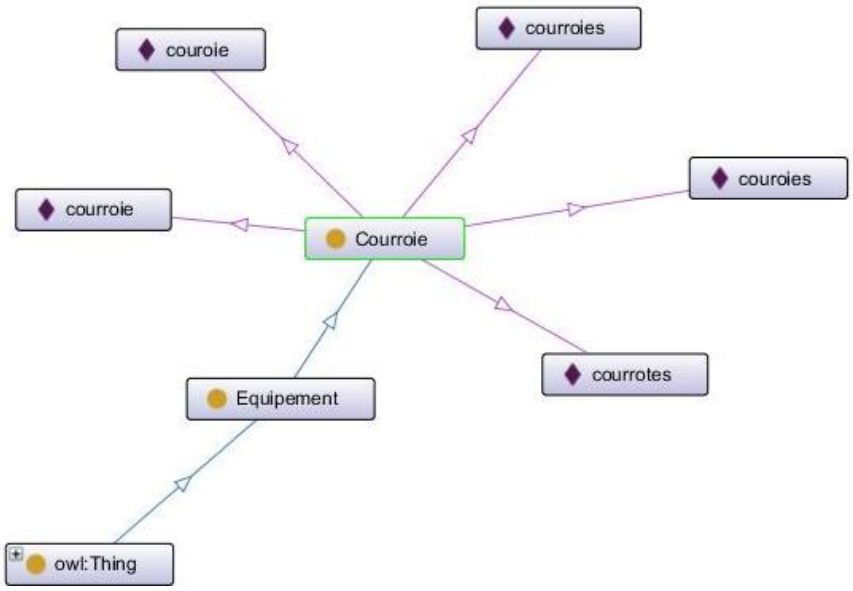
c. Apports

Avec une approche semi-automatique, la construction du plus gros de l'ontologie manuellement (par un humain) reste nécessaire. Cependant, le gain de performance est non négligeable compte tenu de l'absence de correction orthographique : Word2Vec permet de récupérer plus de variations orthographiques et plus rapidement quand l'homme en aurait laissé quelques unes passer (ex : courroie/courrote).

EXTRAIT DES MOTS SELECTIONNES PAR WORD2VEC POUR « COURROIE » :

courroies	couroies	cassee	cassée
lignage	craquelees	poulies	descourroies
Tension	spz	detendue	spa

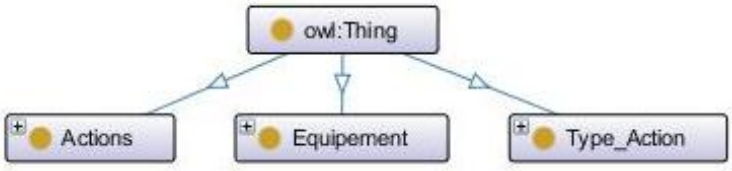
MOTS FINALEMENT SELECTIONNES :



IV- Conclusion

Une combinaison constituée du dictionnaire (format JSON) regroupant les expressions multi-mots (« mise en place », « visite type A », etc.) et de l'ontologie (format OWL) pour les équipements, actions, et type d'action a finalement été utilisée. **Les expressions multi-mots** ne sont conservées que pour être soudées afin de les traiter comme un seul token et ainsi simplifier le traitement. La hiérarchisation n'était donc pas nécessaire pour ces éléments qu'on a conservés au format JSON pour pouvoir en faire une variable Python. C'est aussi par nécessité que ce format a été conservé. En effet les éléments de l'ontologie étant ensuite traité comme des objets et des classes, il n'est pas possible d'y ajouter des éléments comprenant des espaces.

L'ensemble obtenu se compose alors d'une ontologie[4] de 22 classes réparties dans 3 « hyper-classes » représentées ci-dessous. Elles correspondent aux 3 éléments clés d'une recherche : les Actions, l'Equipement, et le Type_Action et rassemblent en tout 84 éléments (ou instances). Enfin, s'ajoute un dictionnaire de 21 expressions multi-mots. Grâce aux modules Python employés (NLTK et Owlready), tous ces éléments construits vont pouvoir interagir afin d'extraire les informations nous intéressant.



[4] Voir structure complète en annexe

I- Nettoyage et formatage du corpus

L'ensemble des données est récupéré au format JSON ou Excel. Chaque entrée a plusieurs attributs parmi 43 au total (dont LIBELLE-DI, LIBELLE-OI, SPECIALITE-OI, etc. écrits en langage naturel).

Un même compte-rendu est réparti sur 4 champs qu'il faut d'abord concaténer soit en les collant simplement, soit en ajoutant un espace entre chaque morceau. Dans les deux cas, on risque un mauvais formatage (mots collés ou tronqués). Ces 4 champs ne constituent pas différentes parties du compte-rendu qui aurait été séparé en amont mais des divisions réalisée a posteriori de façon arbitraire et en ne suivant aucun schéma régulier. On a parfois un trait de césure entre les morceaux, parfois rien, etc. mais jamais d'espace en fin ou en début de partie.

Après expérimentation, il a été décidé de séparer chaque partie par un espace car les résultats amenés ainsi étaient plus cohérents : la séparation au niveau d'un espace deu compte-rendu original étant le schéma le plus récurrent.

Quelques caractères accentués sont mal affichés (Ô et È par exemple). Transcoder le texte vers le format de travail (UTF-8) depuis un autre format (UTF-16, ISO etc.) n'a pas résolu le problème : ces caractères font partie du texte au même titre que les caractères correctement encodés, sans doute à cause de transcodages ayant eu lieu au préalable. Chacun de ces caractères étant toujours remplacé par une même suite de caractères (2 au plus), ils ont été substitués afin de pouvoir exploiter le texte dans son intégralité.

S'ajoute à cela un formatage du corpus. Les comptes-rendus étant exclusivement rédigés en majuscules, le texte est converti en minuscule afin de faciliter les traitements qui suivront. Pour anticiper la suite, on se charge aussi de transformer les points-virgules et espaces multiples en point. Les expressions formées de plusieurs mots et présentes dans la base de connaissance, décrite plus tôt dans ce document, sont également soudées.

Avant formatage	Après formatage
COURROIES **CASS{ES** (3 SUR 5 REMPLACEMENT DES COURROIES VENTILATEURS EXTRACTION(..)	courroies **cassées** (3 sur 5 remplacement des courroies ventilateurs extraction (..)
(…) GENERAL SATISFAISANT REQUALIFICATION SUITE @ **VISITE COMPLETE** SATISFAISANTE	(…) general satisfaisant requalification suite @ **visite_complete** satisfaisante

II- Détection de l'événement

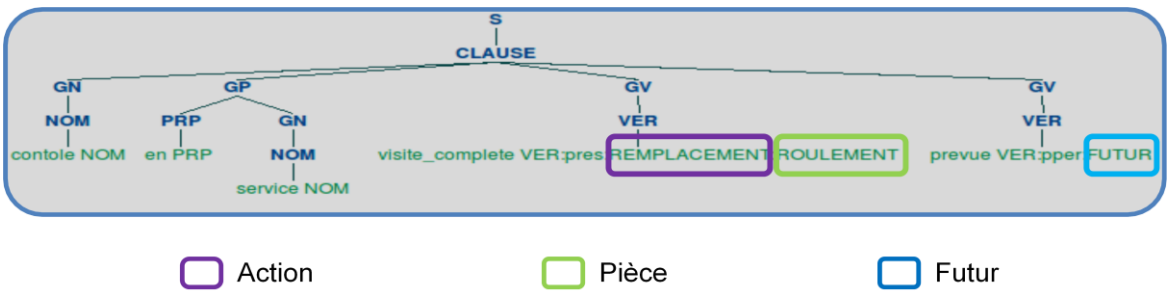
a. Annotations

Le lexique étant déjà construit, et le corpus plus propre, le but est à présent de pouvoir le combiner un ensemble de règles afin de vérifier si le rapport indique qu'un roulement a été changé. Ces règles se basent sur l'étiquetage des mots du compte-rendu et une grammaire du français. Le texte doit donc être annoté.

On utilise notamment la catégorie grammaticale et le nom de la classe de l'ontologie à laquelle appartient un mot comme étiquettes. D'autres étiquettes viendront s'ajouter pour garder en mémoire toutes dimensions associées à un mot. On aura quoiqu'il arrive la catégorie grammaticale accompagnée s'il y lieu des étiquettes :

- FUTUR, pour les mots indiquant une action à venir
- ACTION (l'action dont on veut vérifier la réalisation)
- PIECE (l'objet pour lequel on veut vérifier cette action) ou A_PIECE (s'il s'agit d'un élément comprenant celui recherché)
- NEG, si le mot se situe dans une portion négative
- OI (pour ordre d'intervention), s'il s'agit d'une référence à un autre OI au à un document externe
- ERREUR, s'il y a mention d'une erreur (qu'elle vienne du système d'entrée des comptes-rendus ou de l'intervention en soit)
- ANNULATION : pour les passages qui indiquent qu'une intervention a été déprogrammée

Ces annotations ont été modélisées comme des chaînes de caractères mais il pourrait tout aussi bien s'agir de tuples car leur ordre importe peu tant qu'il est fixe et connu lors de la création de patrons pour la recherche des étiquettes. L'étiquetage morphosyntaxique a cependant lieu en premier puisqu'il contribue à l'étiquetage d'autres propriétés. L'ordre doit néanmoins être fixe sans quoi ces patrons ne seront pas applicables.



La série d'étiquetages permet a priori de réduire le nombre de boucles dans le script pour qu'il soit plus rapide. On a ainsi accès en un seul passage à ce qu'implique un compte rendu, qu'il s'agisse de la réalisation de l'événement ou de son type. Quant au découpage, il évite d'obtenir le label **Réalisé** lorsque « remplacement » et « roulement » sont bien physiquement proches mais pas dans la même phrase.

On s'intéresse aussi aux références à d'autres ordres d'intervention car souvent ils signifient que le compte-rendu est incomplet et donc peu fiable pour la recherche effectuée. Un autre cas récurrent est celui dans lequel le compte rendu indique que l'action a été réalisée mais dans le cadre d'un autre ordre d'intervention. Associer le label **Réalisé** à l'ordre d'intervention à l'étude serait donc une erreur qui pourrait négativement impacter l'application pour laquelle ces résultats sont extraits. C'est aussi dans ce cas que s'inscrivent généralement les comptes-rendus comportant un mot avec l'étiquette ERREUR : le compte rendu a pu être tronqué par le système, ou le rapport lui-même indique un problème de référencement. Ce sont des rapports pour lesquels il a été jugé préférable de systématiquement remonter le label **Doute**.

i. Traitement de la négation

L'étiquetage de la négation est une libre adaptation de la fonction nltk.sentiment.util.**mark_negation** notamment utilisée dans la fouille d'opinion et de sentiments. Si on trouve une expression de négation, on étiquette toute la portion qui la suit comme négative et ce jusqu'à la fin de la phrase. Pour pouvoir l'utiliser il faut conserver les signes de ponctuations et considérer la fin du compte-rendu comme une ponctuation finale.

Voici les éléments changeant par rapport à la version d'origine :

- ⇨ Plutôt que de s'appliquer sur un token seul, l'adaptation s'applique sur une paire token/étiquettes. C'est aux étiquettes que viens s'ajouter « _NEG »
- ⇨ Les ressources rassemblant l'expression de la négation sont en français (et non en anglais) et s'appuient sur les points soulevés dans l'état de l'art de ce mémoire
- ⇨ L'option pour considérer qu'une double négation est une affirmation n'a pas été conservée. La négation en français étant fréquemment double (ne + pas) par défaut.

ii. Traitement du futur

Le traitement du futur est une tache plus ardue puisque le futur n'est pas le seul temps utilisé pour parler de l'avenir. Il y a aussi le présent, l'impératif, etc. Certains verbes ont pu néanmoins être isolés car très identifiables (prévoir, etc.). L'absence de ponctuation et

d'accents pose à nouveau problème car rendant plus difficile le repérage d'expressions comme « à remplacer » automatiquement.

Les comptes-rendus pour lesquels le futur est marqué sont finalement ceux dans lesquels on trouve les mots « prévoir » (et ses formes conjuguées), « prochain » et les groupes infinitifs. On considère ici comme infinitif tout mot finissant par –ir, -er et –dre s'il est précédé de « a », « à » ou du raccourci « @ », également utilisé. Cela peut bien sûr amener des erreurs (« à partir » par exemple qui est renseigné comme une exception). Cependant au vu de l'absence d'accentuation et de normalisation, on ne peut s'appuyer sur les annotations réalisées par Treetagger. Pour ces raisons, et car il s'agit d'un cas de figure très rare dans le corpus, les formes de conditionnel (« il faudrait… ») n'ont pas été traitées.

b. Règles de décision

En parallèle des différents étiquetages, les comptes-rendus sont découpés en groupes verbaux, nominaux, etc. qui sont finalement rassemblés en phrases selon les règles définies par la grammaire du français qui a été crée dans NLTK. Les règles permettant de connaitre l'occurrence de l'évènement sont ensuite appliquées sur chaque énoncé puis synthétisées pour n'obtenir qu'un résultat final.

On considère d'abord **tout le compte-rendu**. On renvoi :

- **Doute :** S'il y mention d'une erreur (avec l'étiquette associée)
- **Non-réalisé** : Si l'intervention a été déprogrammée (tag ANNULATION)

Ensuite, si le compte rendu ne correspond à aucun des ces cas, **pour chaque phrase** :

- On considère qu'un remplacement a eu lieu (**Réalise**) si :
    - La phrase contient l'étiquette de la pièce (ici un roulement) **ET** de l'action (ici un remplacement). Ceci inclut la présence d'expressions signifiant un remplacement de la pièce donnée auquel cas, une seule unité porte les deux labels.
    - Il n'y a pas de négation ni de futur appliqué aux éléments énoncés plus haut, ou celle-ci n'est pas un co-occurrent proche (seuil de 3 unités).

  Si le compte-rendu entre dans ce cas de figure mais qu'il y a mention d'un autre ordre d'intervention, on renvoi systématiquement **Doute.**
- On considère qu'un remplacement n'a pas eu lieu (**Non-réalisé**) si :

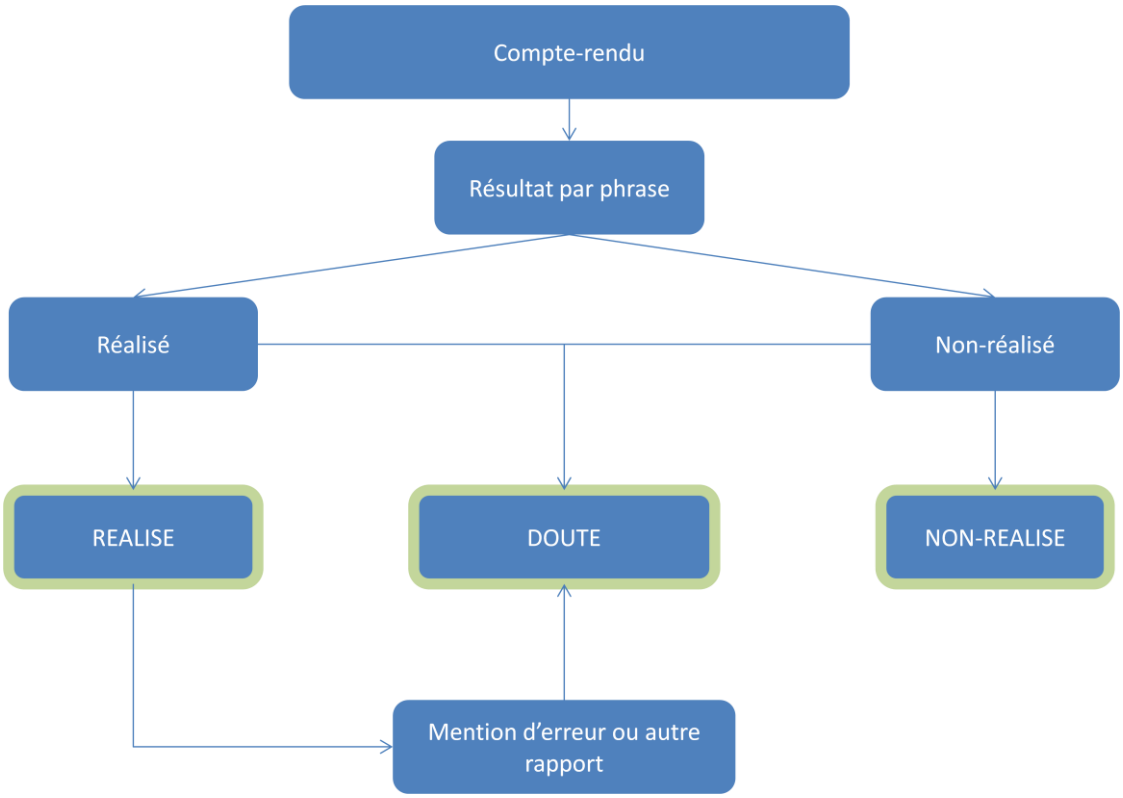
- Il y a présence de remplacement et de la pièce dans une phrase négative (fenêtre de 3 unités)
- Il y a présence d'une mention de l'action et de la pièce et d'un futur (fenêtre de 3 unités)

Les autres cas (pas de mention d'action, mention d'action mais pas de la pièce, etc.) ne sont pas étiquetés.

Dans le cas **de comptes-rendus longs** et sans la moindre ponctuation, on obtenait de mauvais résultats car l'application des règles se basant dessus, il n'était pas possible de les segmenter. Or il est plutôt rare de voir une négation porter sur un mot 10 unités plus loin. Pour éviter que ce type de mauvais résultats ne continue d'être amené, on a intégré aux règles un critère de distance. Pour les cas portant sur le **futur ou de la négation,** on observe une **amélioration avec un seuil de 3 unités**. En revanche en ce qui concerne l'espacement entre une pièce et une action, les distances constatées étant bien plus variables, ajouter ce critère de distance n'améliorait les résultats en rien quelque soit le seuil choisi. Dans le cas de l'étiquette PIECE, ce critère a donc été abandonné.

En revanche, pour le label A_PIECE, il a été conservé avec un seuil plus minimal (1 unité). On s'intéresse également à l'étiquette ACTION situé à deux unités si le mot séparant l'action de l'élément A_PIECE est une préposition. Une expression telle que *« remplacement du ventilateur »* peut ainsi être capturée mais pas *« remplacement de la manchette du ventilateur »,* car n'impliquant un changement de roulement. Les formes telles que *« ventilateur remplacé »* restent prises en compte.

c. Synthèse par phrase



Le schéma ci-dessus représente le cheminement suivi pour décider de si la réalisation de l'événement a été détectée. Lors de la synthèse des résultats pour l'ensemble du compte rendu, les portions non-annotées sont ignorées. Si aucune portion n'a été étiquetée, on renverra **Non-réalisé** pour ce compte-rendu. Si toutes les phrases étiquetées amènent à la même conclusion, c'est bien sûr celle-ci qui est conservée.

Tel que les règles sont conçues, tout Doute pour un énoncé concerne des remplacements de roulements et il suffit donc d'un seul pour renvoyer **Doute** pour tout le compte rendu. Le dernier cas possible est celui dans lequel dans un même compte-rendu on obtient au moins un label **Réalisé** et un **Non-réalisé** auquel cas on reverra **Doute** également.

III- Evaluation

a. Avec corpus de référence

i. Processus de test

Dans l'attente de l'échantillon annotés par le métier, les résultats ont étés d'abord été et évalués en se basant sur les connaissances métier acquises avec les précédents entretiens pour déterminer quels étaient les résultats attendus.

Les tests et améliorations ont été réalisés sur le corpus de test contenant 10 000 comptes-rendus. Compte-tenu de la taille de cet extract il n'a pas été analysé dans son entièreté à chaque itération. Seules certaines entrées filtrées ont fait l'objet d'une étude poussée.

Les comptes-rendus pour lesquels la méthode renvoyait **Réalisé** étant assez peu nombreux **(moins 5%)** mais essentiel pour le projet, c'est d'abord sur ceux qui auraient mal été attribués à cette catégorie qu'on a essayé d'identifier. Cela a permis de mettre en avant des erreurs causées par les règles déjà établies, de procéder à certains ajustements et d'itérer.

Non-réalisé constituant plus de 70% de résultats remontés, il a été plus délicat d'y repérer les erreurs. Les comptes rendus ne contenant pas le mot « roulement » (ou son abréviation « roult ») ont donc été filtrés ainsi que les interventions sans compte-rendu (soit avec des libellés d'intervention uniquement). Cet échantillon a ensuite été utilisé pour réaliser le même processus d'ajustement et d'itération tout en intégrant les nouvelles consignes et informations indiquées par le métier.

ii. Premières évaluation

C'est sur un échantillon moins large **de 1000 entrées (*échantillon A*)** annotées à la main que l'on s'est assuré que les chaque nouvelle version du script ramenait bien de meilleurs résultats que les précédentes, et qu'on a confirmé aussi les suppositions quant à l'ajout ou non d'espaces entre chaque colonnes des comptes-rendus. Ces annotations ne sont cependant pas de réelles références car elles ne sont pas réalisées par les experts. Elles incluent une dimension de doute en accord avec les besoins du projet mais qui n'existe pas dans le fichier de référence. Certains des comptes rendus ont été choisis pour cet échantillon car présentant des cas particuliers (comme l'absence totale de ponctuation), les autres ont été pris aléatoirement

C'est grâce à ces processus d'évaluation comparative qu'on a pu mettre en avant la nécessité d'ajouter des règles de distance. Sur cet échantillon on obtient près de 100% de bonnes annotations mais les tous cas complexes n'y sont pas présents. Cette **estimation** compare les résultats renvoyés par la méthode à ceux des annotations de l'échantillon A.

	Réalisé	Non-réalisé	Doute
Précision	97%	93%	97%
Rappel	83%	96%	100%
F-mesure	89%	94%	98%

iii. Evaluations réalisées à l'obtention du fichier de référence :

Le fichier final de référence contient également 1000 comptes-rendus. Ils ont été annotés à la main par des ingénieurs sûreté de l'UNIE. Ces compte-rendus sont différents des 1000 utilisés précédemment mais certains ont pu être revus lors de la création des règles. Seuls 930 des 1000 comptes-rendus du fichier de référence ont été utilisés pour le calcul des résultats finaux, car pour les 70 restants, il existe encore un doute métier. Ces doutes correspondent à des cas où le contenu du compte-rendu est jugé insuffisant pour déterminer si l'action donnée a été réalisée. N'étant pas de même nature que les doutes décelés pas les règles implémentées (contradiction, etc.), ils sont ignorés. On retrouve dans ce fichier de référence seulement 10% de labels **Réalisé** pour 90% de **Non-réalisé**.

En comparaison à l'évaluation performée *sur l'échantillon A*, il y a une perte de performance pour le label **Réalisé** et notamment en précision. Ce qui est gênant dans la mesure où c'est pour ce cas qu'il y a le plus d'attente de qualité de la part du métier. Mais compte tenu du rapport entre le nombre de d'intervention où le remplacement réalisé et celui où il n'a pas été réalisé (ce dernier étant prépondérant), c'était attendu.

Certain échecs sont liés à un manque dans la base de connaissance (informations métier non fournies en amont) et peuvent donc aisément être corrigés (en ajoutant par exemple dans l'ontologie que lorsqu'un rapport parle d' « expertise » ou de « rebranchement », cela implique le remplacement de roulement ou ajoutant davantage de variations orthographiques).

D'autres sont plus difficiles à traiter car directement liés à la qualité du corpus. Implémenter des règles qui restent génériques mais peuvent les traiter fait augmenter le nombre de faux positifs notamment pour des comptes-rendus mieux rédigés. On préfère les ignorer pour conserver le meilleur taux de performance.

On constate néanmoins des résultats similaires entre GATE et NLTK. Pour rappel, l'étude sous GATE a été réalisée en parallèle par la R&D et partage le même objectif. Ses résultats sont donc présentés ici à titre comparatif. Après discussions, les quelques différences sont davantage liées à la méthode appliquée qu'à la puissance des outils. Nous avons recherché un cas ne pouvant être modélisé par un outil mais pouvant l'être par l'autre, en vain

EVALUATION DES METHODES :

	NLTK		GATE	
	Réalisé	Non-réalisé	Réalisé	Non-réalise
Précision	84%	98%	89%	98%
Rappel	95%	100%	92%	99%
F-mesure	89%	99%	91%	98%

b. Qualitative – critères d'industrialisation

Les activités de la DSP étant liées aux problématiques d'industrialisation d'outils au sein des différentes entités d'EDF, l'évaluation de l'outil réalisé repose également sur des critères d'industrialisation que nous allons introduire ici.

i. Temps d'exécution

Le temps d'exécution est un critère clé car déterminant le mode d'utilisation de l'outil. Il est assez long cependant : pour l'ensemble des rapports concernant les systèmes de ventilation on l'estime à environ une demi-journée. C'est certes plus court que la même opération réalisée par un humain mais étant donné la quantité de rapports (>100 000), il est difficile de voir le produit fini être utilisé comme un web service. Le délai de réponse, trop long, pourrait mener à une interruption même sans dysfonctionnement. Cela fonctionne néanmoins lorsque le jeu de données à analyser est limité. L'application a donc néanmoins été packagée en web service (Flask) appelé la plateforme générale, qui fonctionne avec Elastic Search notamment.

Une utilisation en *one-shot*, avec un système d'enregistrement des résultats, reste à envisager pour n'être relancée que pour les nouvelles données ou en cas de grande modification du script et/ou de la base de connaissance.

ii. Modularité

Une autre préoccupation pour ce projet aura été la modularité. La demande nécessitait que l'application soit finalement suffisamment générique pour aisément être applicable à d'autres rapports que ceux concernant les systèmes élémentaires DVK.

Pour ce faire, on a opéré la séparation des éléments de la brique linguistique afin de faciliter l'adaptation du processus à d'autres actions, d'autres pièces, équipement, centrales, etc.

L'ensemble de l'outil se décompose ainsi en 4 éléments :

- **L'ONTOLOGIE:**
  - **VOCABULAIRE POUR EQUIPEMENT**
  - **VOCABULAIRE POUR ACTIONS**
  - **VOCABULAIRE PAR TYPE D'ACTION (FORTUIT/PREVENTIF)**
- **LA LISTE D'EXPRESSIONS MULTI-MOTS**
- **LA GRAMMAIRE**
- **LE SCRIPT (PRENANT EN ENTREE UN FICHIER JSON OU EXCEL ET AVEC ACTION ET PIECE COMME VARIABLE)**

Si a priori peu d'efforts sont nécessaires pour adapter cette méthodologie, des ajustements doivent être effectués. Les variables utilisées par exemple (par défaut « remplacement » et « roulement ») devront s'adapter à l'objet de la nouvelle recherche selon le même format. Il faudra néanmoins s'assurer de l'existence de ces éléments dans l'ontologie. Dans le cas contraire il faudra actualiser l'ontologie existante ou en intégrer une nouvelle avec les forme pouvant être décrite ainsi que leurs liens avec les autres éléments le cas échéant pour permettre au script d'y « piocher » le vocabulaire nécessaire.

Caractérisation de l'événement

I. Introduction

Dans le cadre du projet, il a également été demandé de déterminer si une intervention était de nature préventive ou fortuite. On parle d'intervention fortuite si elle a lieu des suites d'une anomalie, d'une panne, d'une casse ou si les signes annonciateurs d'un des deux cas précédents étaient présents. Dans ce cas, si un remplacement a bien eu lieu, il donnera davantage d'information sur la durée de vie d'une pièce que lorsqu'il aura été anticipé. C'est ce type de caractérisation que nous allons traiter dorénavant.

Cette information fait déjà partie des attributs pour chaque intervention. Mais si ce besoin a été formulé néanmoins c'est parce que ce champ n'est toujours pas rempli et que le cas échéant, il est souvent mal renseigné : des visites sont marquées comme préventives car réalisée à la date prévue alors qu'elles auraient du être marquées comme fortuites car la panne ou la casse avait déjà eu lieu. Ou inversement marque comme fortuite car réalisées un peu plus tard que prévue mais prévue et pas de panne.

On cherche donc également à savoir si en analysant le rapport de maintenance avec des méthodes de TAL on peut obtenir des informations plus fiables.

II. Retour rapide sur l'approche symbolique

Jusque là, comme pour la détection de l'opération, les méthodes symboliques ont été utilisées. L'étiquetage est alors réalisé selon la présence des mots parmi les classes de l'ontologie Fortuit et Préventif comprises dans Type_Action. Les règles utilisées alors sont les suivantes : on considère qu'un rapport indique

- Une intervention fortuite si :
    - S'il y a au moins un mot étiqueté fortuit
- Une intervention préventive si :
    - S'il y a au moins un mot étiqueté préventif
- Un doute :
    - Si on y trouve les deux types d'annotations

Dans ce cas, la détection est très efficace en termes de précision (97% de comptes-rendus classifiés le sont correctement) mais remonte beaucoup de silence (environ 92% des items ne sont pas classifiés). Les comptes-rendus ne sont pas aussi explicites quant au type d'intervention réalisée que pour les actions réalisées. Utiliser des méthodes statistiques et numériques s'avère donc être la solution une solution envisagée pour pallier au manque d'informations dans le texte.

Pour cette étape, on décide donc de la création de deux corpus : un corpus de rapports d'interventions purement planifiées et un corpus de d'intervention à la suite de panne casse, etc. On a une première classification fiable, et une portion non annotée qu'il aurait été intéressant de pouvoir classifier également. Pour ce faire, nous allons donc nous intéresser à la définition de critères discriminants pouvant être définis à partir du corpus de référence annoté.

A titre indicatif, on trouve 47 **Fortuits** 31 **Préventifs** et 922 rapports non-labélisés dans le fichier de référence. Les rapports pour lesquels cette information est inconnue n'ont pas été distingués des non-concernés. Ces annotations sont indépendantes de celles inclues dans les données d'entrées et qu'on a déjà qualifiées comme manquant de fiabilité et qui sont finalement ignorées. Le fichier annoté à l'aide de méthode symbolique comprend 44 **Fortuits** et 29 **Préventifs**.

III. Classification automatique

De la même façon que les algorithmes de prédiction vont s'intéresser aux données standardisées du rapport (type intervention, date, site, etc.), nous allons sélectionner pour chaque rapport un set d'informations jugé pertinent concernant les champs textuels. On s'intéresse à des points de nature linguistique tandis que les algorithmes prennent en comptes d'autres éléments pouvant potentiellement affecter la nature du rapport tel que sa longueur.

a. Division apprentissage entrainement

On décide de diviser le texte annoté en deux portions pour l'apprentissage et le test de ces algorithmes. Il est cependant important de noter que les rapports annotés selon le types d'intervention sont peu nombreux (79). 60% d'entre eux serviront de base d'apprentissage, le restant servira au test.

b. Classification via LSA

La méthode LSA[5] **[Landauer, et al., 1997**]ou analyse sémantique latente est au carrefour de la sémantique et des statistiques et fréquemment utilisé en recherche d'information.

i. Prétraitements linguistiques

Le côté sémantique de la méthode consiste à enrichir le texte d'information concernant les sens des mots, notamment en utilisant leurs racines. Pour ce processus de classification, le corpus subira donc une étape de prétraitement plus lourde. On utilise la racinisation pour obtenir des unités minimales de sens pouvant être plus pertinentes que les lemmes dans le cadre de ce type d'expérimentation. Le choix du stemmer peut faire varier la qualité des résultats mais Snowball a été choisi car sous NLTK, c'est le seul applicable au français.

SYNTHESE POUR LES PRETRAITEMENTS LINGUISTIQUES :

	Forme pleine seule	+ POS	+ Lemme	+ POS +Lemme	+ POS + lemme +racinisation
Correctement classés	46,875%	46,875%	65,625%	68,75%	59,75%

On observe, en dépit de ce que LSA propose, de meilleurs résultats avec la lemmatisation sans racinisation. Tout comme pour la lemmatisation, avoir un texte aussi peu normalisé est pénalisant. Mais à l'origine de cela se trouve également un vocabulaire sans doute trop spécialisé et qui n'est pas associé à la bonne racine car absent de la base utilisée par Snowball.

[5] *Latent Semantic Analysis*

ii. Conversion matricielle et réduction

L'aspect mathématique quant lui consiste à construire un modèle vectoriel (*vector space model*) du corpus via des matrices « termes par document » se présentant sous la forme suivante:

	Mot1	Mot2	Mot3	Mot4	etc.
Doc 1	2	1	1	3	…
Doc 2	0	2	5	1	…

Ces valeurs, fréquentielles dans un premier temps, sont ensuite normalisées : on utilise le TF-IDF[6] (*term frequency – inverse document frequency*) de chaque mot plutôt que leur fréquence car cette mesure prend mieux en compte la disparité en termes de taille de chaque document.

Les matrices sont ensuite décomposées en valeurs singulières (SVD) puis **réduite** pour ainsi se concentrer sur les valeurs les plus fortes. C'est le modèle matriciel issu de ces étapes qui va servir à prédire la nature des éléments du corpus de test qui auront alors subit les mêmes étapes. Ils sont alors classifiés en utilisant les k-plus proches voisins (avec distance cosinus)

iii. Paramétrages et classification

Cette méthode nécessite de paramétrer un certain nombre de champs :

- Le nombre de traits (**100**) sachant, qu'a priori plus on en a moins on perd d'information mais plus le traitement est long
- On ne s'intéresse pas au mots-outils du français ni aux mots-outils propres au discours technique (soit ceux **apparaissant dans plus de 70% des comptes-rendus**)
- On ne s'intéresse pas aux mots apparaissant dans moins de **3** comptes-rendus (ce qui permet de filtrer les ID notamment : numéro d'OI, etc.)

Le risque cependant en filtrant ces mots (presque des hapax dans les faits) est d'aussi supprimer les fautes d'orthographes uniques ou de perdre un vocabulaire technique ayant

[6] $tfidf_{ij} = tf_{ij} \cdot \log \frac{N}{df_i}$
Où tf_{ij} est la fréquence du mot *i* dans *j*
df_i le nombre de documents contenant *i*
N le nombre total de documents

peu d'occurrences. A ne donc appliquer que sur un corpus conséquent ou, à défaut, en connaissance de cause.

Le modèle LSA est cependant remis en question à la vue de la perte de performance à l'ajout des informations quand à la racine des mots pourtant essentiel à la dimension sémantique du processus. Par ailleurs le principe même consistant à traiter des sacs de mots, le texte n'est pas vu comme une suite de mots mais comme un ensemble et il est ainsi inévitable de perdre de l'information sur le texte.

Nous allons donc essayer d'appliquer des méthodes d'apprentissage automatique et notamment utilisées en data-science pour analyser des séquences de données.

b. Classification par apprentissage automatique

L'apprentissage automatique consiste pour une machine à correctement prédire un événement ou comportement et à éventuellement agir en conséquence sans intervention humaine, en analysant des résultats précédents. Il nous servira ici à réaliser une classification des rapports.

Les algorithmes présentés ici seront testés à l'aide du logiciel WEKA[7] . Ils utilisent des fichiers ARFF également converti en matrices terme par documents.

- Naive Bayes
- Modèle de Markov caché

Outre ses résultats, Naive Bayes a été choisi car c'est un modèle où les traits sont analysés indépendamment contrastant ainsi avec le modèle de Markov caché (HMM) qui s'intéresse aux séquences. Ce dernier a été jugé pertinent car on sait que des expressions multi-mots et patrons morphosyntaxique notamment peuvent constituer des critères de classification.

[7] *Waikato Environment for Knowledge Analysis*

IV. Evaluation et conclusions

	LSA	Machine Learning	
		HMM	Naive Bayes
Correctement classés	59,75%	65%	90%

On obtient des résultats décents sans l'utilisation des relations et lexiques construits dans l'ontologie des étapes précédentes, preuve que la modélisation linguistique peut être supplantée par des modèles mathématiques pour cette tâche.

S'il n'est en revanche pas possible d'évaluer les résultats de ces approches pour les rapports qui n'ont pas été annotés, on peut néanmoins s'attendre à ce qu'ils aient une fiabilité proche de la précision obtenue.

Le risque cependant est de classer l'ensemble du corpus selon ces deux caractéristiques quand parfois un rapport ne devrait être associé à aucune car ne décrivant aucune action menée. Pour gagner en fiabilité, il eut été idéal d'obtenir un échantillon de ce type de rapport également avec une codification associée.

Conclusion

I- Conclusion

Au cours de ce travail, nous avons traité la problématique de détection de la réalisation d'événements dans des rapports de maintenance. Ces rapports se distinguent de la plupart des corpus étudiés de par leur style laconique pourtant très technique également. Outre le repérage du vocabulaire utilisé lors de la description d'une action, nous avons dégagé deux problématiques clés pour la détection de sa réalisation : l'impact du futur et celui de la négation. Ces événements auront ensuite du être caractérisés selon leur caractère fortuit ou préventif.

La constitution de ressources linguistiques et terminologiques complètes et adaptées au style de rédaction des rapports aura été un pan important de cette démarche et aura permis l'application de méthodes symboliques. Si nous avons démontré que ces méthodes pouvaient suffire pour réaliser cette première tâche de détection, nous avons en revanche vu que pour la caractérisation les modèles linguistiques pouvaient s'avérer insuffisants. En effet, ce caractère, aussi important soit-il, est rarement explicité dans les rapports. Ces méthodes peuvent néanmoins être supplantées par des modèles mathématiques qui présentent d'excellentes mesures de qualité (avec jusqu'à 90% de classifications correctes avec l'algorithme de classification bayesien naïf).

II- Perspectives

Cette étude justifiant la collaboration entre traitement automatique du langage et data science pour l'exploitation du retour d'expérience, elle sera amenée à être approfondie. Les futurs travaux impliquent cependant des textes normalisés ; la définition et la modélisation de propositions dans la grammaire employée, qui n'ont pas été accomplis, ici y sera possible et peut-être même nécessaire. Ces travaux devront aussi tenter de résoudre le manque d'optimisation de l'outil décrit ici dont la performance est finalement assez lente. A terme, le résultat de ce type de méthodes sera combiné au restant des données structurées pour participer à l'amélioration d'algorithmes prédictifs.

Annexe

EXTRAIT DES RAPPORTS ANALYSES :

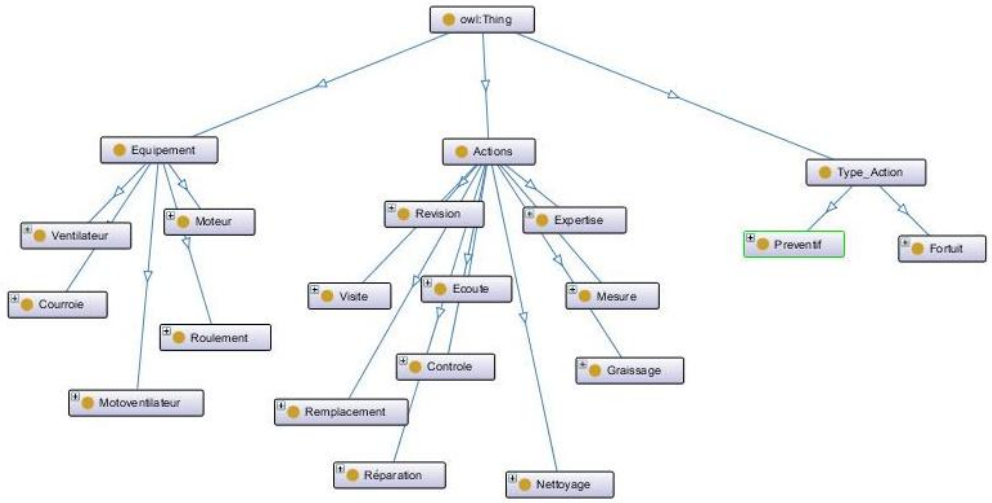
Texte analysé
LA CELLULE DU VENTILATEUR EST EN DEFAUT DIAG ET REMISE EN ETAT CELLULE (MOTEUR A REMPLACER DI990460) ENTR.COTE CELLULE: REMPLACES FUSIBLES & COMPTEUR HORAIRE. COTE MOTEUR: REMPLACEMENT DU MOTEUR PAR UN MOTEUR NEUF . ENROULEMENT ET ISOLEMENTS ===) CONFORMES. . SENS DE ROTATION ET INTENSITE ===)
MOTEUR HS - ECHANGE STANDARD DU MOTEUR - DEPUIS 1993 PAS DE REMPLACEMENT PB DE MAUVAISE ADAPTATION DES MOTEURS EN STOCK MAGASIN POUR UN MOTEUR DE ZV EN ZC IL FAUT 1)UNE PEINTURE DECONTA MINABLE 2)UN ROULEMENT NU COTE POULIE 3)GRAISSE SHC100
USURE DE L ARBRE . V.I VENTILO MOR 80 SUITE A EXPERTISE (OIN0483162) ARBRE H.S REMPLACEMENTARBRE,ROULEMENTS PALIERS,VIS FIXATION MO CONTROLE DE BON FONCTIONNEMENT OK
DECLENCHEMENT 5 DVK 001 ZV DIAGNOSTIC DE NON FONCTIONNEMENT REMPLACEMENT DU MOTEUR DANS OIN0537848 REMPLACEMENT DES FUSIBLES UNELEC 63 AMPERES SUR CELLULE D E PUISSANCE.ISOLEMENT 0 MOTEUR A LA MASSE
DEBRANCHEMENT REBRANCH POUR VISITE COMPLETE MOTOVENTILATEUR 4 DVK 006 ZV:MOTEUR 380V VISITE DEBRAN./REBRANCHEMENT DEBRANCHEMENT EFFECTUE CE JOUR. REBRANCHEMENT DU MOTEUR. ISOL >100 MOHM. MOTEUR BON
DEBRANCHEMENT REBRANCH.POUR VISITE COMPLETE MOTOVENTILATEUR 4 DVK 003 ZV:MOTEUR 380V VISITE DEBRAN./REBRANCHEMENT DEBR + REBR ET ISOL > 100MOHMS
VISITE COMPL.MOTOVENT.EXTRACT.A DEBIT REDUIT PIEGE IODE BK VISITE COMPL.MOTOVENT.EXTRACT.A DEBIT REDUIT PIEGE IODE BK REMPLACEMENT ROULEMENTS BON ETAT GENERAL DU MOTOVENTILATEUR REQUALIFICATION EFFECTUEE SOUS L'OI N0175359
VISITE ANNUELLE VENTILATEUR SOUFFLAGE BATIMENT COMBUSTIBLE VISITE ANNUELLE VENTILATEUR SOUFFLAGE BATIMENT COMBUSTIBLE VISITE ANNUELLE DU MOTO-VENTILATEUR 4 DVK 001 ZV REALISEE REMPLACEMENT DES COURROIES, REMPLACEMENT D'UN PALIER DU ZV. REMONTAGE ET REQUALIFICATION OK. TRAVAUX NEU 2M EN CAS 1, VOIR RFI Nø 3334.M3.04 HISTORISE AVEC CET OI.
CONTROLE MECANIQUE DES PALIERS VENTILATEUR VISITE PALIER DU VENTILATEUR VISITE PALIERS SUITE CONSTAT TøC ELEVEE LORS RMS. EXCES DE GRAISSE. MANQUE 1 DEFLECTEUR COTE POULIE. REMPLA RLTS 22313 ET MANCHONS. CTRL ARBRE: CORRECT. RFI NEU 31012.G1. 09. RMS SATISFAISANTE.
MOTEUR EN COURT-CIRCUIT ECHANGE STANDART MOTEUR.
VISITE TYPE B. CONTROLES MECANIQUES, GRAISSAGE. VISITE TYPE B. CTRL MECANIQUES, GRAISSAGE. ||| + FORTUIT |||

GRAMMAIRE :

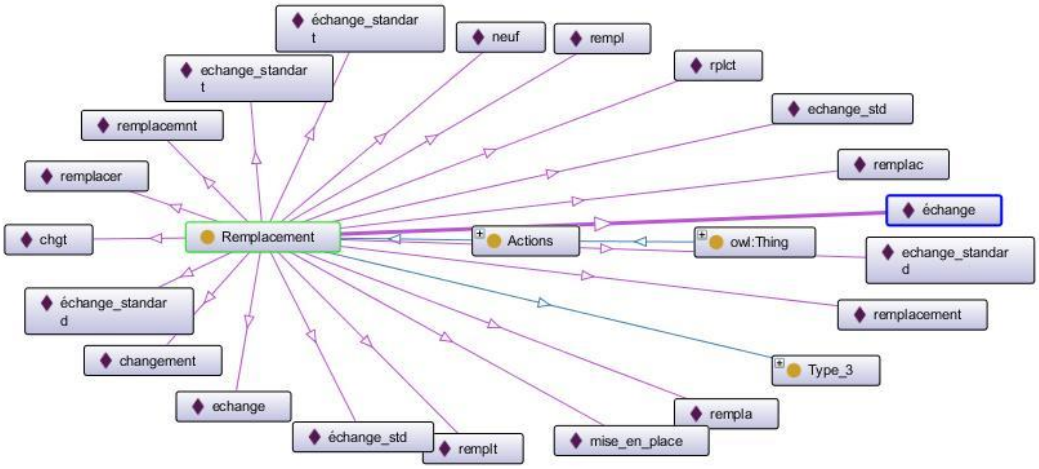
```
NOM : {<NOM.*>}
DET : {<DET.*>}
ADJ : {<ADJ.*>}
ADV : {<ADV.*>}
PRP : {<PRP.*>}
VER : {<VER.*>}
KON : {<KON.*>}
NEG : {<NEG.*>}
PPER : {<PRO:PER>}
NUM : {<NUM.*>}
PRO : {<PRO.*>}
PUN : {<PUN>}
ABR : {<ABR.*>}
SYM : {<SYM.*>}

GADV: {<PUN>*<ADV>+<PUN>*}
GADJ: {<PUN>*(<GADV>*<ADJ><KON>?)+<PUN>*}

GN: {<PUN>*(<DET|NUM|PRO|PPER>?<GADJ>?<NOM|NUM|PRO|PPER|SYM|ABR><GADJ>?<KON>?)+<PUN>*}
GN: {<PUN>*<GN><KON><GN><PUN>*}
GP: {<PUN>*<PRP><GN|NUM|GV><PUN>*}

GV: {<PUN>*<VER><GN|GP|GADV|GADJ|CLAUSE>?<PUN>*}
CLAUSE: {<G.*>+<SENT>*}
```

ONTOLOGIE:

- **CLASSES**



- **FOCUS SUR LES INSTANCES DE LA CLASSE REMPLACEMENT :**



Références

Bird, Steven, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.", 2009.

Cunningham, Hamish, et al. "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics." PLoS computational biology 9.2 (2013): e1002854.

Bontcheva, Kalina, et al. "Evolving GATE to meet new challenges in language engineering." Natural Language Engineering 10.3-4 (2004): 349-373.

Leopold, Edda, and Jörg Kindermann. "Text categorization with support vector machines. How to represent texts in input space?." *Machine Learning* 46.1-3 (2002): 423-444.

Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

Constant, Matthieu, et al. "Intégrer des connaissances linguistiques dans un CRF: application à l'apprentissage d'un segmenteur-étiqueteur du français." *TALN*. Vol. 1. 2011.

Moeschler, Jacques. "Négation, portée, et la distinction négation descriptive/métalinguistique." *J. François, P. Larrivée, D. Legallois et F. Neveu (éds), La Linguistique de la contradiction, Berne, Peter Lang* (2013): 163-179.

Longhi, Julien, Claudia Marinica, and Haddioui Naoual. "Extraction automatique de phénomènes linguistiques dans un corpus de tweets politiques: quelques éléments méthodologiques et applicatifs à propos de la négation." *Res per Nomen V*. Presses universitaires de Reims, 2015.

Muller, Claude, « La négation, opérateur transversal dans la construction des énoncés. », Revue de linguistique latine du Centre Alfred Ernout, 2008

Larrivée, Pierre. L'association négative: depuis la syntaxe jusqu'à l'interprétation. Vol. 35. Librairie Droz, 2004.

Zweigenbaum, Pierre, et al. "Apports de la linguistique dans les systèmes de recherche d'informations précises." *Revue française de linguistique appliquée* 13.1 (2008): 41-62..

Tannier, Xavier. *Extraction et recherche d'information en langage naturel dans les documents semi-structurés*. Diss. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2006.

Vanrullen, Tristan. "Analyse syntaxique et granularité variable." *Actes, Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL)*. LPL, 2004.

Heiden, S., Magué, J-P., Pincemin, B. (2010a). TXM : Une plateforme logicielle open-source pour la textométrie – conception et développement. In Sergio Bolasco, Isabella Chiari, Luca Giuliano (Ed.), *Proc. of 10th International Conference on the Statistical Analysis of Textual Data -)* (Vol. 2, p. 1021-1032). *Edizioni Universitarie di Lettere Economia Diritto*, Roma, Italy.

Lamy JB. Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. Artificial Intelligence In Medicine 2017;80:11-28

Helmut Schmid (1994): . Proceedings of International Conference on New Methods in Language Processing, Manchester, UK.

Toussaint, Yannick. Fouille de textes: des méthodes symboliques pour la construction d'ontologies et l'annotation sémantique guidée par les connaissances. Diss. Université Henri Poincaré-Nancy I, 2011.

Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.

Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*(2013).

Muneeb, T. H., Sunil Kumar Sahu, and Ashish Anand. "Evaluating distributed word representations for capturing semantics of biomedical concepts." *Proceedings of ACL-IJCNLP*(2015): 158.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.

Landauer, T. K., and Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240

Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." *The American Statistician*46.3 (1992): 175-185.

Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11.1 (2009): 10-18.

Yi, Kwan, and Jamshid Beheshti. "A hidden Markov model-based text classification of medical documents." Journal of Information Science 35.1 (2009): 67-81.

Denoyer, L., Zaragoza, H., and Gallinari, P. 2001. HMM-based passage models for document classification and ranking. In Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research (Darmstadt, DE, 2001).