Développement d'un classifieur hybride pour le domaine des " Ressources Humaines "

Myriam Gafsi

Développement d'un classifieur hybride pour le domaine des « Ressources Humaines »

Myriam GAFSI

Sous la direction de Pr. LEBARBÉ Thomas

Réalisé au sein de la société : ELOQUANT
Sous la direction de : DUSSERRE Emmanuelle

UFR LLASIC
Département I3L

Mémoire de master 2 professionnel - Mention Sciences du langage - 20 crédits

Parcours : Industries de la langue

Année universitaire 2019-2020

Développement d'un classifieur hybride pour le domaine des « Ressources Humaines »

Myriam GAFSI

Sous la direction de Pr. LEBARBÉ Thomas

Réalisé au sein de la société : ELOQUANT
Sous la direction de : DUSSERRE Emmanuelle

UFR LLASIC
Département I3L

Mémoire de master 2 professionnel - Mention Sciences du langage - 20 crédits

Parcours : Industries de la langue

Année universitaire 2019-2020

Remerciements

Je profite de ces quelques lignes pour adresser mes remerciements à ceux qui, de près ou de loin, m'ont supporté tout au long de mon travail. Ce mémoire n'aurait pas pu être réalisé sans l'intervention des certaines personnes qui m'ont apporté une aide précieuse.

Tout d'abord, je tiens à exprimer ma gratitude et reconnaissance à Emmanuelle, mon maître de stage. Merci pour ton encadrement exceptionnel, ton suivi, ton soutien, ta confiance en moi et tes multiples conseils qui m'ont permis d'accomplir un travail dont je suis fière.

Je remercie aussi Ruslan qui a été généreux en conseils jusqu'à la fin. Merci pour tes remarques objectives apportées à ce travail.

J'adresse mes profonds remerciements à Mr Lebarbé pour son encadrement et son suivi au cours de ce travail de mémoire.

Un remerciement spécial à Myriam et David pour votre aide mais aussi pour vos blagues, pour toutes les rigolades et les parties de tennis. Tous ces moments ont illuminé mes journées et ont permis de rendre mon stage encore meilleur.

De plus, je remercie Mathieu et Munsta pour votre gentillesse et votre bonne humeur, ainsi que toute l'équipe ELOQUANT pour l'accueil et l'ambiance.

Je remercie vivement mes professeurs de Master IDL, qui tous se reconnaîtront je l'espère sans que j'aie besoin de les nommer.

Je voudrai aussi remercier mes amis et camarades de master : Lucie, Justine, Marie, Rachel, Florent, Vincent et Oussama, avec qui j'ai partagé deux années d'études inoubliables.

Enfin, mes derniers remerciements sont adressés à toute ma famille, en particulier mes parents, pour m'avoir toujours soutenu au cours de mes études. Sans leur soutien je n'aurai jamais pu atteindre mes objectifs.

DÉCLARATION ANTI-PLAGIAT

1. Ce travail est le fruit d'un travail personnel et constitue un document original.
2. Je sais que prétendre être l'auteur d'un travail écrit par une autre personne est une pratique sévèrement sanctionnée par la loi.
3. Personne d'autre que moi n'a le droit de faire valoir ce travail, en totalité ou en partie, comme le sien.
4. Les propos repris mot à mot à d'autres auteurs figurent entre guillemets (citations).
5. Les écrits sur lesquels je m'appuie dans ce mémoire sont systématiquement référencés selon un système de renvoi bibliographique clair et précis.

PRENOM : MYRIAM

NOM : GAFSI

DATE : 03/09/2020

Table des matières

INTRODUCTION

1. Contexte et problématique

L'intelligence artificielle (IA) s'impose dans le quotidien des particuliers et des entreprises. En effet, par le biais des nouvelles technologies, les données disponibles sur les consommateurs, les collaborateurs, etc., sont toujours plus accrues. Ces informations textuelles sont des mines d'or pour les entreprises car elles constituent un ensemble de ressources non négligeable pour leur développement. Cependant, aux vues de la quantité de données récoltées grâce à l'IA, il est essentiel que les entreprises soient formées efficacement sur le traitement de ces données « Big data[1] ». En effet, travailler avec l'IA nécessite deux qualités principales. Premièrement, il faut être expert en traitement des systèmes informatiques et deuxièmement il est essentiel de connaitre les différentes possibilités d'analyse du traitement du langage naturel. Pour le traitement informatique, l'IA peut faire appel à plusieurs méthodes, dans ce stage, nous avons travaillé avec les méthodes supervisées et non supervisées, dont nous verrons les définitions dans les prochaines parties. Pour le traitement du langage naturel, il est nécessaire, pour analyser les sorties de l'IA, de savoir traiter les informations brutes données par les différentes sorties du système. En effet, elles ne constituent pas, en elles même, des sources facilement analysables. Cela est d'autant plus vrai pour les données peu structurées, comme les textes « libres », qui sont difficiles à interpréter, soit par le volume des documents, soit par le coût du travail manuel. Celui-ci est fastidieux en termes de temps de travail, peu générique et relativement peu efficace.

Afin de tirer le meilleur parti de l'IA, nous avons donc besoin d'outils qui nous permettent de chercher, trier, classer et analyser les données accessibles. Mais aussi pour nous aider à rapidement trouver les informations dont nous avons besoin, à effectuer certaines tâches à notre place lorsque cela est possible, ou du moins à nous faciliter le travail. Dans ce contexte, le besoin d'une catégorisation[2] automatique de textes, définie comme un processus permettant d'assigner des textes à un ensemble prédéfini de catégories (ou classes), devient alors une nécessité presque absolue pour aider les entreprises que ce soient les équipes marketing du côté du service client ou les DRH[3] pour les enquêtes collaborateurs. Une telle aide permettrait d'automatiser des tâches de classification manuelle répétitives et

[1] **Big data** : Un ensemble très volumineux de données.
(http://www.ece.ubc.ca/~minchen/min_paper/BigDataSurvey2014.pdf)
[2] **Catégorisation** ou classification, lorsque les catégories sont mutuellement exclusives. Par abus de langage, nous utiliserons ces termes de manière interchangeable.
[3] **DRH** : Directeur, directrice des Ressources Humaines.

chronophages pour les utilisateurs. Qui plus est, ces derniers doivent traiter des centaines voire des milliers de documents, il est indispensable de s'équiper de technologies basées sur de l'intelligence artificielle pour pallier ces tâches chronophages et fastidieuses.

Nos utilisateurs concernés sont les clients d'Eloquant, à savoir des entreprises, qui souhaitent analyser des textes produits par leurs clients ou leurs collaborateurs, par exemple dans le cadre d'une enquête de satisfaction diffusée avec les outils d'Eloquant. Leur objectif étant de comprendre les sujets et causes de désenchantements de leurs clients afin d'améliorer le fonctionnement de leur entreprise. Pour ce faire ces entreprises ont besoin d'étudier manuellement ces données textuelles, appelées des verbatim,[4] souvent conséquents.

L'analyse automatique de texte nous permet aujourd'hui d'extraire et d'analyser les opinions et les thématiques issues d'enquêtes de satisfaction de plusieurs façons. L'objet de ce mémoire s'inscrit donc dans une perspective d'accompagner les utilisateurs afin de faciliter ces tâches d'analyses redondantes. Pour ce faire, nous avons créé un classifieur automatique de verbatim, contenant un nombre défini de thématiques, s'adressant aux entreprises réalisant des enquêtes d'écoute « collaborateurs ». A travers ces enquêtes, les entreprises ont un enjeu capital : savoir motiver, fidéliser et accroître l'engagement de leurs collaborateurs, c'est pourquoi ces enquêtes relatives au bien être des collaborateurs prennent une place de plus en plus accrue auprès des entreprises. Elles représentent un outil extrêmement précieux, puisqu'elles permettent aux collaborateurs de se sentir entendus et de prévenir le *turnover* ou encore les risques psychosociaux. Il est donc devenu quasi incontournable pour les entreprises de donner l'opportunité à leurs employés de s'exprimer via des réponses écrites libres (plutôt que des questionnaires de type Likert[5]), et de capitaliser sur ces réponses, qui peuvent être nombreuses, longues mais une mine d'or d'information. Ainsi, une analyse sémantique automatisée pour ces textes écrits permet de faciliter les tâches des DRH ou RRH[6] et d'avoir une vision exhaustive des problématiques évoquées.

Dans le cadre de l'analyse des données issues des enquêtes, nous avons mis en place un système de classification thématique. La méthode actuelle de la création du classifieur automatique de textes est directement liée à l'apprentissage automatique supervisé. Il s'agit d'un sous-domaine de l'intelligence artificielle qui vise à donner aux machines la

[4] **Verbatim** : La réponse textuelle à une question ouverte lors d'une enquête de satisfaction.
[5] **L'échelle de Likert** contient plusieurs choix de réponses permettant à la personne interrogée d'exprimer son degré d'accord ou de désaccord.
[6] **RRH** : Responsable Ressources Humaines

capacité d'apprendre à classer les documents. Dans le cadre de notre projet, le classifieur est développé de cette façon. À partir d'un ensemble de données déjà étiqueté et ajusté avec des enrichissements sémantiques par un expert de domaine, nous pouvons entraîner un système de classification automatique. À ce stade, la machine peut apprendre à effectuer la tâche de classification et à établir des liens entre les textes et les catégories grâce à l'analyse des exemples du travail d'annotation effectué par le linguiste qui fait partie de son entraînement. Après la phase d'entraînement, le classifieur sera capable de classer les documents textuels dans les catégories définies. Par la suite, nous évaluons la performance du classifieur en comparant ses propositions avec les catégories d'un ensemble de données réputé correctement annoté, mais qui n'a pas servi à l'entraînement. L'évaluation permet d'avoir un aperçu quantitatif et qualitatif des performances du classifieur et de faire émerger ses défauts, et lorsque cela est possible, de les corriger de diverses façons. Eventuellement, ces nouvelles connaissances permettront d'augmenter le taux de réussite du classifieur.

Pour la réalisation de ce travail, ce mémoire sera structuré en trois grandes parties : Nous présenterons dans la première partie l'état de l'art, nous mettrons notre travail en perspective par rapport aux travaux similaires qui ont été conduits autour de la catégorisation automatique de textes, des différents classifieurs d'apprentissage automatique, de la construction semi-automatique de taxonomies et nous finirons par des travaux menés sur la construction d'un classifieur hybride (combinaison d'un modèle statistique et symbolique). Nous démontrerons à la fin de cette partie comment notre travail apporte une dimension nouvelle au système de classification hybride développé par l'équipe HOLMES[7] en 2016.

La deuxième partie sera consacrée à la méthodologie de développement du système de classification hybride. Nous expliquerons donc les différentes étapes de la constitution des données afin qu'elles soient le plus compréhensibles possible par la machine et utilisables par les algorithmes d'apprentissage. Nous décrirons ensuite la constitution de la maquette de la catégorisation pour le domaine des enquêtes RH qui servira à l'annotation des données, et enfin le développement du système de classification hybride combinant apprentissage automatique (statistique) et règles symboliques.

Nous présenterons dans la troisième partie les mesures d'évaluation et d'analyse des résultats obtenus afin d'évaluer les performances de plusieurs configurations du classifieur de façon à pouvoir présenter celui-ci aux clients dits « bêta-testeurs ». Nous terminerons avec une

[7] **HOLMES**: Hybrid Operable platform for Language Management and Extensible Semantics.

conclusion générale dans laquelle nous étudierons la portée ainsi que les perspectives d'évolution du classifieur.

2. Présentation de l'entreprise

Ce mémoire présente le travail effectué au sein de l'entreprise Eloquant, dans le cadre d'un stage professionnel de deuxième année de master Industries de la Langue, à l'Université Grenoble Alpes.

Depuis 2001, Eloquant est spécialisée dans la relation client et développe des solutions globales en mode SaaS[8], dans le but de faciliter le dialogue et l'écoute des entreprises avec leurs clients.

L'entreprise fournit à ses clients trois grands services :

- Dialogue : Gestion unifiée des contacts entrants et sortants multicanal : téléphone, emails, SMS, chat, réseaux sociaux, etc.
- Écoute : Enquêtes multicanal de la mesure et du pilotage de la satisfaction.
- Sémantique : Analyse sémantique automatisée de données textuelles.

Dans le cadre de son offre EXPLORE, l'entreprise propose à ses clients d'effectuer une analyse sémantique des verbatim (commentaires, avis, mails et autres ressources textuelles) issus des échanges avec leur clientèle et/ou de leurs systèmes d'information.

L'analyse sémantique proposée permet, à partir d'un verbatim fourni en entrée, d'extraire diverses informations structurées :

- Les concepts détectés dans le verbatim.
- Les entités nommées (lieux, personnes, etc.) repérées dans le verbatim.
- Les catégories auxquelles le verbatim appartient.
- Les opinions avec leur polarité.

Les enquêtes effectuées par Eloquant sont des questionnaires mêlant questions fermées et ouvertes. Concernant notre projet, nous nous sommes consacrés aux questions ouvertes donnant le champ libre aux collaborateurs pour exprimer en quelques phrases leurs opinions sur leur vie d'entreprise.

[8] **SaaS** : *Software as a Service* ou Logiciel en tant que Service, est un modèle de distribution de logiciel à travers le Cloud. Les applications sont hébergées par le fournisseur de service.

Tous ces informations collectées seront traitées dans le moteur Holmes, qui est composé d'un ensemble de packages Java, pouvant être paramétrés selon le domaine, les besoins de l'utilisateur et les données à traiter, et d'une chaîne de traitement appelée *pipeline*[9], que nous pouvons trouver dans des outils comme STANFORD CORENLP[10] et MALLET[11]. Le pipeline Holmes permet :

- La normalisation orthographique des verbatim.
- La tokenisation : découpage des verbatim en *token*[12] (mots).
- La lemmatisation : attribution des part-of-speech (POS) et des informations morpho-syntaxiques (par exemple : genre, nombre, personne) à chaque token.

Lors de mon stage, j'appartenais à l'équipe CX (customer experience). Cette équipe est divisée en deux, d'une part les chefs de projet (CX consultant) chargés de mener à bien un projet et gérer la relation avec le client et d'autre part l'équipe technique (customer care) qui suivent au quotidien les clients (gestion des problèmes mineurs, mise en place des enquêtes de satisfactions ou de post appel, etc.).

3. Sujets et objectifs du stage

3.1 Tâches à réaliser

La mission principale durant ce stage était de développer un classifieur automatique dédié au domaine des ressources humaines. Les tâches que nous avons réalisées ont été les suivantes : la normalisation des données, la création de la maquette des catégories, le développement du système de classification hybride par apprentissage automatique et enrichissement sémantique, et pour finir une évaluation du classifieur. Enfin, nous avons appliqué le système développé pour une démonstration client.

3.2 Méthodes de suivi du travail

Avant de commencer le travail, nous nous sommes réunis pour déterminer les objectifs du travail et fixer les délais pour chaque tâche.

[9] Nous utiliserons l'anglicisme **pipeline** pour remplacer le mot chaîne de traitement.
[10] https://stanfordnlp.github.io/
[11] http://mallet.cs.umass.edu/
[12] **Token** est un anglicisme employé pour désigner une entité (ou unité) lexicale dans le cadre d'une analyse lexicale.

L’encadrement dans l’entreprise était assuré d’un côté par mon maître de stage, avec des points d’avancements hebdomadaires, et d’un autre côté, par l’équipe sémantique R&D (Recherche et Développement) lors de points d’équipe techniques. Afin d’atteindre notre objectif et d’être capable de livrer une première version à la fin du stage, nous avons fixé un planning au début du stage qui a été bien respecté et ce malgré les difficultés sanitaires rencontrées.

Partie 1.

ETAT DE L'ART

La catégorisation de textes est une des tâches indispensables dans le domaine du Traitement Automatique des Langues (TAL) et de la Recherche d'Information, qui repose souvent sur des Algorithmes d'Apprentissage.

Cet état de l'art est réalisé à partir de travaux de chercheurs et vise à faire un état des lieux bref et synthétique du sujet suivant : *« Le développement d'un classifieur automatique hybride ».* Cette notion d'hybride implique une part d'apprentissage automatique et une part experte, c'est-à-dire l'usage de connaissances et des procédures explicitement déclarées par des experts et non apprises empiriquement par la machine-même.

Afin de mieux décrire cette problématique, dans une première partie nous expliquerons le processus général de la catégorisation de textes. Dans une deuxième partie, nous présenterons les différentes méthodes d'apprentissage automatique. Dans la troisième partie, nous aborderons le sujet de la constitution d'une taxonomie semi-automatique et nous terminerons par présenter la méthode hybride du système de classification automatique.

1. Catégorisation automatique de textes

Comme il vient d'être mentionné, le but de la classification automatique de textes est de programmer la machine à classer un texte dans une ou plusieurs classes prédéfinies, en se basant sur des indices relevés dans son contenu. Les stratégies de classification peuvent soit être déclarées explicitement par les développeurs, soit apprises par la machine via des algorithmes d'apprentissage par observation de données annotées[13].

[Jalam, 2003] définit la catégorisation de textes comme « une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes) ». À partir de ces paires <document, catégorie> (ou <document, ensemble de catégories> pour la classification multi-label), il s'agit de construire, par apprentissage ou par déclaration explicite de procédures de classification, un modèle de prédiction qui, dans l'idéal, engendre le moins d'erreur possible, ce qui implique de disposer d'une ou plusieurs mesures d'erreur, dont le calcul devrait, si possible, être automatisable.

Un tel modèle prend en entrée un texte et lui attribue en sortie une ou plusieurs catégories. Le développement d'un système de catégorisation consiste en plusieurs étapes : (1) collecte et étude des données, qui permet de faire émerger les catégories pertinentes du domaine

[13] Nous parlons alors d'apprentissage « supervisé », puisque la machine dispose d'annotations lui indiquant la ou les classes attendues.

(et, au passage, de les normaliser[14] si besoin). (2) Annotation manuelle des données avec ces catégories pour constituer l'ensemble d'apprentissage (ou d'entraînement ou de développement, dans le cas de règles expertes). (3) Développement d'une algorithmique de classification. Cette algorithmique peut reposer sur des règles expertes permettant de définir automatiquement la catégorie, ou sur l'usage de méthodes d'apprentissage automatique, voire des deux. (4) Le système obtenu est ensuite évalué et ajusté s'il le faut, afin de converger vers un modèle de classification qui fait le moins d'erreurs en prédiction.

Afin de pouvoir s'assurer que le modèle génère le classement avec le classement de l'ensemble d'apprentissage fait par des experts humains, il faut appliquer une méthode d'évaluation et calculer un score de performance. Plusieurs de ces méthodes d'évaluations ont été proposées dans la littérature. Dans cette section, nous présenterons les méthodes qui sont couramment utilisés par les chercheurs dans le domaine de la classification automatique des documents. Pour mieux illustrer ces différentes mesures nous présentons les mesures suivantes :

- La précision P (precision en anglais) : soit le nombre de documents classés dans la catégorie par rapport au nombre total de documents classés.

$$\text{Précision} = \frac{\#\text{Nombre de document correctement attribués à la classe i}}{\#\text{Nombreb total de document attribué à la classe i}}$$

- Le rappel R (recall en anglais). : soit le nombre de documents classés dans la catégorie par rapport au nombre de documents appartenant à la catégorie.

$$\text{Rappel} = \frac{\#\text{Nombre de document correctement attribués à la classe i}}{\#\text{Nombre de document appartenant à la classe i}}$$

- F-mesure : La moyenne de la précision et du rappel. Il est calculé comme suit :

$$\text{F-mesure} = \frac{(2 * précision * rappel)}{précision + rappel}$$

Il existe aussi d'autres mesures, tels que le Kappa de Cohen[15], ou le *Matthews Correlation Coefficient*[16] *(MCC)*, qui dans certaines circonstances, reflètent mieux les performances des

[14] Il peut s'agir d'anonymisation, de correction orthographique ou typographique, etc.
[15] https://www.datanovia.com/en/fr/lessons/kappa-de-cohen-dans-r-pour-deux-variables-categorielles/
[16] https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

classifieurs. Cependant, pour notre travail la triade Précision—Rappel—F-Mesure est tout de même informative, et est si omniprésente dans la littérature que son usage reste pour l'instant inévitable.

La somme de la littérature que nous avons lue montre que la catégorisation de textes :

- Est généralement un problème de classification dite supervisée, c'est-à-dire fondée sur des jeux de données annotées, dont on vise à reproduire les annotations de façon automatique.
- Fait généralement appel à des techniques d'apprentissage automatique.

Nous abordons donc le sujet d'apprentissage automatique (dit *Machine Learning* en anglais) dans la partie suivante.

2. Apprentissage automatique : les classifieurs

Les approches d'apprentissage automatique appartiennent généralement à une de deux grandes classes d'approches : l'apprentissage supervisé et l'apprentissage non supervisé[17].

La différence entre ces types réside dans le fait que dans l'apprentissage supervisé les algorithmes apprennent à prédire les catégories des données d'entrée à partir d'exemples annotés, comme dit auparavant. En revanche, les techniques d'apprentissage non supervisées (par exemple le *clustering*) ne requièrent pas de données étiquetées. L'apprentissage s'effectue de façon totalement autonome et les algorithmes découvrent la structure inhérente des données à partir des propriétés statistiques des traits qui les caractérisent.

Naturellement, pour chacune de ces approches, plusieurs types de classifieurs ont été proposés au fil des années. Certaines en ont définitivement supplanté d'autres (par exemple les arbres de décision ne servent guère plus qu'à des fins didactiques), mais aujourd'hui plusieurs approches compétitives sont disponibles, avec forces et faiblesses qui leur sont propres. La tâche de classification automatique implique donc un choix de l'algorithme d'apprentissage (ou classifieur). Ce choix est déterminé selon l'objectif final à atteindre [Jalam, 2003].

Dans cette partie, nous présenterons les différentes méthodes d'apprentissage les plus souvent utilisées afin de réaliser la tâche de classification de textes. Chacun de ces classifieurs a des avantages et des inconvénients, cependant ils partagent des caractéristiques communes.

[17] Notons qu'il existe aussi des techniques d'apprentissage dit semi-supervisé, qui combine une petite quantité de données annotées et une grande quantité de données non-annotées, mais nous n'allons pas l'étudier ici.

Certains classifieurs consistent à identifier la classe d'appartenance d'un texte à partir de certains traits descriptifs. Ces traits (appelés aussi features, descripteurs ou caractéristiques, etc) sont extraits des données textuelles et sur lesquelles le processus de classification de textes en dépend directement. Les traits peuvent être des mots, des traits morphosyntaxiques, des traits sémantiques ou même des N-grammes, etc. Les performances du classifieurs dépendent crucialement de la pertinence de ces features.

La première méthode s'intitule **« Séparateur à Vaste Marge » (SVM[18]),** introduite par [Vapnik, 1995]. Proposé en 1995, ce classifieur conserve sa popularité grâce à ses bonnes performances dans différentes tâches d'apprentissage et a été reconnu pour sa performance dans l'application à la classification des textes [Dumais et al., 1998]. Cette méthode, conçue initialement pour la classification binaire, mais depuis étendue aux problèmes multiclasses via différentes techniques, est fondée sur la recherche d'un séparateur linéaire, appelé hyperplan, qui sert à séparer les classes des points projetés dans un espace multidimensionnel approprié[19], en se positionnant au plus loin des points appartenant aux classes distinctes (d'où la notion de « vaste marge »). Les points spécifiques qui déterminent la position de l'hyperplan sont appelés « vecteurs de support ». La complexité d'un classifieur SVM ne dépendra pas de la taille de l'espace de données, mais du nombre de vecteurs supports nécessaires pour effectuer la séparation.

La deuxième méthode est « **les *k* plus proches voisins** » [Yang et Chute, 1994] ou *K-nearest neighbors* en anglais (d'où l'appellation K-NN). Pour effectuer une prédiction, l'algorithme K-NN sera baser sur l'ensemble de données en entier. En effet, pour déterminer la classe d'une nouvelle donnée d'entrée, le modèle mémorise les observations de l'échantillon d'apprentissage. Il cherche les *k* points voisins les plus proches de la nouvelle donnée que l'on souhaite classer selon une certaine mesure de distance[20]. *K* est un nombre entier positif, défini par l'utilisateur, l'emploi de *k* voisins plutôt que d'un seul a pour but d'augmenter la fiabilité de la prédiction. La classe attribuée à l'instance à classifier est la classe majoritaire des *k* voisins. L'efficacité de la méthode découle de ces trois étapes et donne de bons résultats. K-

[18] Souvent maladroitement appelé Machine à Vecteurs de Support, traduction littérale de son nom original en anglais (*Support Vector Machine*).
[19] Par « approprié » on entend un espace de *features* de dimension plus grande que celui des données de départ, et dans lequel les classes, si elles n'étaient pas linéairement séparables dans leur espace d'origine, le deviennent. Les SVMs sont intéressants par le fait qu'ils ne projettent pas les données dans ces espaces, mais utilisent « l'astuce du noyau » pour calculer la similitude entre les points comme s'ils l'étaient (https://fr.wikipedia.org/wiki/Astuce_du_noyau).
[20] Souvent la distance euclidienne, parfois pondérée par la proximité des voisins.

NN est un algorithme assez simple à appréhender. Principalement, grâce au fait qu'il n'a pas besoin de modèle pour pouvoir effectuer une prédiction. En revanche, les résultats ont tendance à baisser en qualité lorsque le nombre de variables explicatives est grand.

Ensuite, **« la méthode de Rocchio »** proposée dans [Rocchio, 1971], est un classifieur linéaire conçu pour améliorer les systèmes de recherche documentaires. La méthode de Rocchio[21] est utilisée pour la création de profils prototypiques[22] pour chaque catégorie. Lors de la classification d'un nouveau document, un calcul de mesure de similarité est fait entre les profils des classes et le vecteur correspondant au nouveau document. Le document dont le profil est le plus proche au vecteur sera attribué à la classe. Pour une catégorisation dont le texte ne peut appartenir qu'à une seule catégorie, l'apprentissage de la méthode de Rocchio est rapide, et donne de bons résultats avec un bon taux de précision. En revanche, elle n'est pas très efficace si le texte peut appartenir à plusieurs catégories.

La méthode suivante est appelée « **les Bayes naïfs (NB)** » ou « classification naïve bayésienne[23] » [Lewis et al., 1994]. Elle permet de calculer les probabilités d'appartenance d'un document à une classe. C'est une méthode simple à appliquer et ne demande que peu d'informations pour l'estimation des paramètres. En effet, les données sont établies sur un simple calcul de cooccurrences sans aucune pondération de traits. Lors d'une classification de textes, sur la base des caractéristiques supposées statistiquement indépendantes (d'où le terme « naïf ») des descripteurs du corpus d'entraînement, le modèle calcule la catégorie la plus probable du document à classifier. Cette méthode est très rapide pour la classification, en effet les calculs de probabilités ne sont pas très coûteux et la classification peut être effectuée même avec un petit ensemble de données. En revanche, l'algorithme de cette méthode suppose l'indépendance des variables. C'est une hypothèse forte qui est abandonnée dans la plupart des situations pratiques.

La dernière méthode, que nous présenterons, est **« les réseaux de neurones »** [Wiener 1993, et Wiener et al. 1995], (ou *Artificial Neural Network* en anglais). C'est un modèle de

[21] https://en.wikipedia.org/wiki/Rocchio_algorithm
[22] Un profil prototype d'une classe est une liste de termes pondérés, dont la présence et l'absence discriminent au mieux cette classe. Pour un expert, un profil prototype est plus compréhensible qu'un réseau de neurones par exemple.
[23] https://towardsdatascience.com/naive-bayes-intuition-and-implementation-ac328f9c9718

calcul habituellement utilisé pour des tâches de classification, qui prend la forme d'un graphe orienté dont les nœuds sont appelés « neurones » et les arcs (pondérés) « synapses ». La phase d'apprentissage prend en entrée des données présentées sur l'une des couches terminales (appelée la couche d'entrée). Le principe de cette méthode est que chaque entrée est connectée à une ou plusieurs sorties par un chemin dans le graphe. Elle est traitée par un nombre paramétrable de couches de neurones intermédiaires ou couches cachées. Le résultat est la sortie produite par l'autre couche terminale (appelée la couche de sortie). Les neurones possèdent un ensemble d'arcs incidents ou sortants qui véhiculent des « signaux » numériques, et qui appliquent une certaine fonction mathématique (dite « fonction d'activation ») à la somme des signaux incidents, et dont le résultat est propagé hors du neurone via les synapses sortantes. Les poids de ces synapses sont des facteurs par lesquels est multipliée la valeur de la fonction. L'apprentissage consiste en l'ajustement de ces poids synaptiques. Il se fait sous le contrôle des associations prédéfinies entre documents (entrées du réseau) et classes (sorties du réseau) qui maintiennent le comportement du réseau souhaité. Ce modèle est simple à manier et a la capacité de représenter n'importe quelle fonction malgré le bruit ou le manque de fiabilité des données. Mais les réseaux de neurones peuvent vite être assimilés à une boîte noire en raison d'absence des explications concernant ses résultats. Par conséquent, ce système n'a que des capacités d'interprétations limitées, alors que d'autres systèmes experts peuvent être en mesure de retracer le raisonnement des résultats afin d'expliquer les conclusions tirées.

3. Construction semi-automatique de taxonomies

Selon le dictionnaire Larousse, la taxonomie (ou taxinomie) est la « science des lois de la classification ». Dans notre contexte, une taxonomie décrit des catégories organisées hiérarchiquement et sert à classifier et à rassembler, sous une même annotation, des contenus ou des ressources documentaires selon les caractères qu'ils ont en commun, des plus généraux aux plus particuliers. Nous nous intéressons ici à l'usage de taxonomies pour la classification automatique, et à leur construction semi-automatique.

[Mustière et al., 2009] présentent dans leurs travaux une taxonomie de concepts topographiques construite à partir de l'analyse semi-automatique de documents textuels (environ 700 termes), dans l'objectif de créer une ontologie géographique. Pour ce faire, ils ont défini les termes de référence liés à la description de l'environnement et ont extrait du texte les concepts souhaités. Par la suite, ils ont effectué automatiquement une reconstitution de la hiérarchie de ces concepts à partir de leur localisation dans les textes. Enfin, une analyse

automatique du contenu des spécifications a été mise en place, ces spécifications décrivant le contenu de leurs données textuelles et servent à créer la taxonomie une fois formalisées.

D'autres publications sont également très adaptées à notre sujet comme [Mondary, 2008], qui aborde le sujet de la construction d'ontologie semi-automatique sur la base de corpus et insiste dans ses travaux sur l'emploi des usages attestés. Les étapes présentées par [Mondary, 2008] sont :

1) l'identification des termes.

2) le regroupement en classes sémantiques.

3) la structuration en réseau terminologique.

Ces étapes ont été suivis par notre collègue Armelle Ramond [Ramond, 2016][24], qui a élaboré une taxonomie spécifique au domaine de la relation client (RC) avec 20 catégories thématiques. Elle a développé un système de classification utilisant, d'une part, les différentes sources de la taxonomie (sous forme de gazetteers et tokensregex, qui seront expliqués dans le chapitre suivant) et d'autre part, un corpus de verbatims spécifique au domaine RC. Sa taxonomie est construite en une hiérarchie de concepts dans le but de relier les entrées lexicales entre elles, dans une représentation structurée basée sur leurs valeurs et leurs liens sémantiques. Ainsi, les termes sémantiques étroitement liés sont représentés par la même position dans la hiérarchie. [Ramond, 2016] nous montre qu'une taxonomie riche lui a permis de mieux représenter l'information sur le plan sémantique et d'obtenir ainsi un modèle d'apprentissage automatique plus solide.

4. Classification automatique de textes : méthode hybride

[Maurel S., Curtoni P., et Dini L., 2007][25] ont mis au point trois méthodes différentes permettant de réaliser une classification automatique de texte.

- La première méthode est symbolique, elle inclut un système d'extraction d'information propre aux corpus. Elle est basée sur des règles d'analyse syntactico-sémantique du texte. Le choix est donné au linguiste, à partie d'un système de base générique, de former une grammaire à sa façon et de la compléter ou l'enrichir avec de nouvelles règles afin d'extraire les relations syntactico-sémantiques qui l'intéressent.
- La seconde méthode est statistique, elle est basée sur des techniques d'apprentissage automatique. Elle prend en entrée des textes (la sortie de la pipeline) pour faire une première

[24] Il s'agit d'un travail fondamental pour la Sémantique à Eloquant, sur lequel repose le nôtre.

[25] Anciens de *Holmes Semantic Solutions*, entreprise acquise par Eloquant, et dont l'équipe Sémantique est héritière.

classification. La grammaire construite dans la méthode précédente peut être reprise et améliorée à cette étape pour obtenir de meilleurs résultats. Le travail prend alors la forme d'un cycle où les résultats s'améliorent constamment et en particulier les erreurs aperçues dans la classification. Ceci aide à corriger les erreurs de l'apprentissage automatique.

- Enfin, la troisième méthode est hybride, c'est une combinaison des deux premières méthodes. Elle calcule une moyenne (qui sera le résultat final) à partir des résultats des deux méthodes symbolique et statistique. « *C'est une approche qui permet de garder la robustesse de l'apprentissage automatique de la méthode statistique et d'orienter en même temps la base de l'entraînement sur une configuration manuelle de la méthode symbolique.* » [Maurel S., Curtoni P., et Dini L., 2007]

[Maurel S., Curtoni P., et Dini L., 2007] choisissent donc de calculer la classification finale à l'aide de la méthode hybride.

Ces approches ont fait l'objet de travaux dans le projet de [Ramond, 2016], qui a développé un système hybride pour la relation client. Nos travaux s'imbriqueront dans une démarche similaire en mettant à l'épreuve ce système pour un domaine spécifique aux ressources humaines avec un corpus au volume plus faible.

5. Paradigme de l'étude

Après avoir présenté les travaux similaires, ayant été conduit par différents chercheurs, notre objectif est de se reposer sur les technologies de classification automatique et d'enrichissement sémantique existants afin de proposer un classifieur robuste spécifique au domaine des ressources humaines. Pour cela, nous allons présenter les différents travaux réalisés lors de ce stage. L'originalité de ce sujet se révèle dans son coté novateur pour le domaine des ressources humaines mais aussi pour l'entreprise elle-même. En effet, c'est une première pour Eloquant qui, habituellement, oriente ses travaux vers l'analyse des relations clients. Ce travail a été réalisé selon plusieurs axes résumés dans cette partie.

Une maquette de catégories des ressources humaines a été le premier travail réalisé. Celle-ci a été présentée à l'expert du domaine qui l'a ensuite validée. Une fois la maquette constituée, nous avons donc fait appel aux traitements déjà mis en place par l'entreprise, cependant, le sujet, le lexique, les concepts et les catégories ont été modifiés et sont donc nouveaux et propres au domaine. En effet, la maquette représente la base du processus de classification. Une attention spécifique a été prêtée à cette tâche et notamment à la définition de la liste des catégories.

Un autre facteur très important pour la qualité des résultats est la présentation des données. Cependant, nous ne disposons pas d'un lexique des ressources humaines pour faire une annotation sémantique lexicale. Ce type de ressource est indispensable pour l'analyse statistique et symbolique afin de garantir une bonne classification. Dans un premier temps, afin d'annoter les données et d'entraîner le système, nous expérimenterons un nouvel outil, pour lequel nous avons été formés, sur la plateforme « Sherpa » de Kairntech[26] (Cf. chapitre 3, section 1.1). Ensuite, pour la méthode symbolique du système, nous construirons une grammaire générique destinée au domaine des ressources humaines ayant ses propres règles et ses concepts regroupant plusieurs termes sémantiques pour la constitution de la taxonomie. A cette phase, nous utiliserons Word2vec (Cf. chapitre 3, section 2.1.2) afin d'enrichir notre taxonomie.

Le classifieur hybride sera exploité pour une application d'entreprise et expérimenté selon les besoins des différents clients.

[26] **Kairntech** est fondée en 2019 et spécialisés dans l'intelligence artificielle, le traitement du langage naturel (NLP), l'ingénierie des connaissances et le développement de logiciels. https://www.kairntech.com/

Partie 2.

METHODOLOGIE

Chapitre 1. Structuration des données

La qualité de notre classifieur dépend avant tout des données textuelles que nous avons pour entraîner le modèle d'apprentissage automatique. Par conséquent, elles doivent être construites avec une attention particulière et faire l'objet d'un nettoyage minutieux.

Cette première partie est consacrée à la présentation des quatre étapes qui ont été nécessaires pour la constitution d'un corpus de données normalisé.

1. Les corpus d'étude

1.1. Collecte des données

La catégorisation a été élaborée à partir des données issues d'enquêtes réalisées auprès de collaborateurs. Ces enquêtes ont été présentées sous forme des questionnaires qui ont pour objectif de mesurer le niveau de satisfaction des collaborateurs et de leur permettre d'exprimer leurs opinions de l'entreprise et de leur situation professionnelle. Pour ce faire, nous avons travaillé sur deux corpus appartenant au domaine des ressources humaines.

La diffusion a été réalisée via des questionnaires papiers ainsi que par formulaire web envoyés par email. Les verbatim collectés sont répartis sur l'année 2019 et les questionnaires sont présentés en 14 langues (français, italien, polonais, anglais, espagnol, allemand, slovène, hongrois, turc, portugais, hindi, tchèque, chinois et japonais).

Les enquêtes diffusées prenaient la forme suivante :

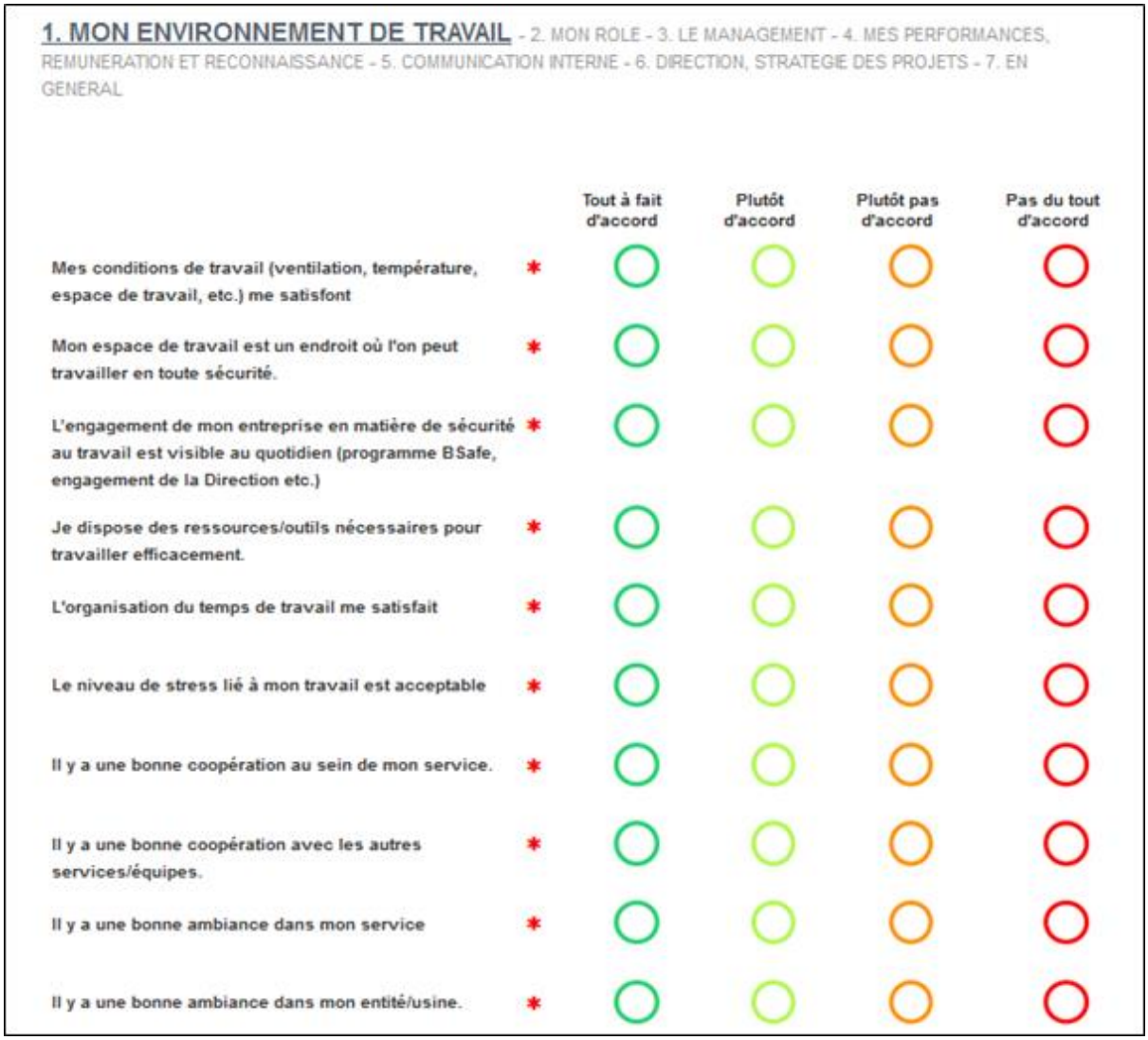

Figure 1. Exemple d'un questionnaire de satisfaction

La structure du questionnaire est composée de plusieurs thématiques. Ce formulaire, par exemple, porte sur le sujet « Mon environnement de travail » avec 10 questions et 4 choix de réponses.

Dans chaque partie, une question ouverte (Cf. figure 2) était présentée à la fin de chaque thématique aux collaborateurs permettant de recueillir leurs remarques et suggestions de façon libre. Les questions ouvertes sont un avantage pour l'extraction d'informations qualitatives et l'exploration du verbatim. En effet, les réponses aux questions ouvertes sont plus spontanées et riches en informations, en opposition aux questions fermées, qui reflètent plutôt les aspects quantitatifs. C'est donc cette partie des questionnaires qui nous intéresse pour l'analyse sémantique. Grâce aux questions ouvertes permettant de recueillir les avis des collaborateurs de façon détaillée, nous avons obtenu des données plus précises sur l'expérience vécue par les collaborateurs au sein des entreprises.

Figure 2. Exemple d'une partie d'un questionnaire de satisfaction

Les données collectées étaient strictement anonymes ce qui permet aux collaborateurs de s'exprimer sans crainte. Nous avons récolté au total 4237 réponses. Ces dernières ont représenté nos corpus, qui étaient classés sous format csv et en UTF8 comme standard d'encodage des caractères. Nous présentons dans le tableau suivant un extrait des réponses que nous avons récoltées : Le verbatim, les identifiants et la langue d'origine du verbatim.

Verbatim	Identifiants	Langue
I am satisfied with my employer.	7hv3k7q	en
Was and good opportunity at start but it' stagnant now	PulaqeA	en
great place to work	fHEbJQZ	en
Les propositions de formations professionnelle sont insuffisantes.	WgcAmUw	fr
il n y a pas de comunication	6YYr16e	fr
Dans l ensemble je suis tres satisfaite de mon travaille	avEPSOM	fr
Ritengo di non aver un inquadramento allineato con i lavori svolti.	ra9fD8R	it
distribuire equamente ipremi	HsgLbtP	it
la empresa valora a las personas	njDTR8Y	es
Demasiada carga de trabajo por el sueldo que tengo	C6No1jb	es

Tableau 1. Extrait du fichier CSV avec les verbatim des collaborateurs

1.2. Format des données

Afin de pouvoir exploiter au mieux nos corpus, nous avons sélectionné, à partir du fichier csv (Cf. tableau 1), les verbatim, les identifiants des collaborateurs et la langue originale

du verbatim. Ces données ont été traitées avec la classe Java « *DataImporter.java* », qui permet la conversion des fichiers csv en plusieurs documents XML permettant de structurer les données qui nous intéressent. Cette opération est appelée sérialisation et nous l'effectuons avec JAXB (Java Architecture for XML Binding).

La structure XML du verbatim et ses métadonnées (les informations reliées au verbatim et à son auteur, par exemple : la date, l'origine de la personne, sa région, etc.) est la suivante :

```
<document analyzable="true" id="corpus1">
        <metadata url="url_du_corpus1.csv" corpus="nom_du_corpus1" language="fr"/>
        <text>manque de ressources</text>
    </document>
```

Figure 3. Exemple d'une structure XML

Id : identifiant du corpus d'origine

Url : chemin d'accès au fichier csv à l'emplacement où il est stocké

Corpus : nom du fichier csv

Language : version texte originale du verbatim

Text : texte du verbatim

2. Traduction des corpus internationaux

Les corpus que nous avons collectés sont en différentes langues étant donné que les questionnaires sont diffusés à l'international. Pour traduire les verbatim qui en sont issus, nous avons fait appel à l'API[27] Google Translate qui permet de traduire plus de 100 langues. L'API (interface de programmation d'application) est un ensemble de classes, de méthodes, de fonctions et de constantes standardisées. Elles constituent la structure de base à travers laquelle un logiciel peut fournir des services à d'autres logiciels.

Le tableau 2 indique d'une part les langues présentes dans les corpus (Source) et d'une autre part le code de langue proposée par Google (Target). Ces codes doivent être respectés afin que l'API détecte la langue et puisse traduire le texte. En effet, certaines langues ne sont pas acceptées par l'API Google.

[27] **API Google** : https://cloud.google.com/translate/docs/languages?hl=fr

source	target
fr	fr
en	en
es	es
nederlands	nl
italien	it
GERMANY	de

Tableau 2. Les codes de langue acceptés par Google

Dans le cadre de notre projet, nous avions 13 langues étrangères à traduire. Le pipeline prend en entrée les corpus internationaux au format XML et fournit en sortie des verbatim traduits. Afin d'être homogène avec le reste du pipeline, tous les codes utilisés étaient développés en JAVA.

La structure XML de chaque verbatim traduit est la suivante :

```
<document analyzable="true" id="id_corpus1">
    <metadata url="url_corpus1" corpus="corpus1" language="fr">
        <custom-fields>
            <field type="string" name="source-language" value="en"/>
            <field type="string" name="source-text" value="It is an honor to work in good environment."/>
        </custom-fields>
    </metadata>
    <text>C'est un honneur de travailler dans un bon environnement.</text>
    <analysis date="2020-07-22"/>
</document>
```

Figure 4. Exemple d'une structure XML d'un verbatim traduit

À la structure précédente s'ajoute la langue originale du verbatim, la langue vers laquelle il a été traduit et la date de la traduction. En ce qui concerne notre travail, nous avons traduit l'ensemble des verbatim vers le français, le pipeline que nous présenterons ultérieurement fonctionne sur la langue française.

3. Normalisation des données textuelles

La phase de normalisation orthographique et lexicale des données textuelles était indispensable pour notre travail. Le but consiste à rapprocher le verbatim d'un standard de la langue française pour qu'il soit compréhensible par la machine et utilisable par les algorithmes d'apprentissage.

La démarche de la normalisation que nous avons appliquée (appelée aussi Spell Correction) peut être résumée de la manière suivante :

- Nous avons tout d'abord nettoyé le texte en corrigeant les fautes d'orthographe des mots, les fautes de frappe et les abréviations (par exemple : tres -> très, tjr -> toujours, toute fois -> toutefois etc.)
- Ensuite, cette phase inclut également une normalisation des erreurs d'encodage, ponctuations, espacements, emojis, répétitions.
- Enfin, une deuxième phase de « Sentence Split » consiste à découper les phrases en fonction de la présence des ponctuations (reconnaissance des fins de phrase ou de paragraphe), mais aussi de caractères de séparation (de type « espace », « tabulation » ou « retour à la ligne »). Il s'agit d'un traitement de surface simple, néanmoins il est difficile de le réaliser avec précision sur des documents ayant beaucoup de bruits et des représentations très variées.

---	tr9CeUA	fr
? commenter quoi ?	WD2SG6X	fr
Globalement bon	WHcUePq	fr
)	ppG01em	fr
????????????????	ytSrNIm	fr
.	MKxSoWP	fr
Bien	ln3Hqcm	fr
rien a dire de plus	DO2QzTY	fr
confidentiel	5I0lbH1	fr
RAS	kIFH7fv	fr
Bienveillance	oaXEQxR	fr
mes reponses peuvent paraitre negatif	eyn9ZZ4	fr
Je suis stagiaire depuis 10 jours	EhoRGKr	fr

Figure 5. Extrait de certains verbatim après la phase de sentence split

Après toutes ces étapes de normalisation, nous avons éliminé également d'autres verbatim (présentés dans la figure 5). Par exemple, les verbatim « trop petits », c'est-à-dire se composant de moins de deux mots, les verbatim sans texte contenant uniquement des signes de ponctuation et les verbatim du type « RAS », « Pas de commentaires », « Rien à dire », etc. Ces verbatim ont été supprimés car ils ne peuvent pas être classés et n'apportent aucune information pour la catégorisation. À la suite de ce nettoyage, notre corpus était constitué, au total, de 1662 verbatim étiquetés.

4. Anonymisation des données confidentielles

L'anonymisation des données correspond à la suppression d'informations qui peuvent permettre d'identifier une personne ou une entreprise, par exemple les noms de lieux, les noms de personnes, les adresses, etc. Ce processus d'anonymisation garantit non seulement la sécurisation de l'utilisation des données personnelles, mais également le respect des droits fondamentaux des personnes dont les données personnelles sont traitées.

Eloquant a obtenu la norme ISO 27001 pour la sécurité de l'information de ses clients. Préserver et protéger la confidentialité des informations de ses clients est d'une importance primordiale, quelle que soit leur forme, contre l'accès, l'utilisation, la diffusion, la destruction et la modification non-autorisées (que ce soit par accident ou par malveillance).

Suivant le règlement général de protection des données (RGPD), nos corpus ont été anonymisés. Le RGPD responsabilise les organismes publics et privés, qui collectent et traitent leurs données, en mettant en place des règles sur la collecte et l'utilisation des données sur le territoire de l'Union Européenne.

Afin d'empêcher leur identification, toutes ces données ont été remplacées par CLIENT_ELQ_1, CLIENT_ELQ_2, etc.

Chapitre 2. Constitution de la maquette des catégories collaborateur

Dans ce chapitre, nous allons établir une maquette des catégories suite aux verbatim des collaborateurs. Pour ce faire, un algorithme de *clustering*[28] va alors venir en aide pour nous proposer une première partie sur les thématiques. À partir de ses propositions, nous allons élaborer et construire une maquette qui classe tous les verbatim dans leurs catégories respectives.

Nous présenterons, tout d'abord, l'outil de clustering et son fonctionnement, ensuite notre méthodologie sur la constitution de la maquette des ressources humaines.

1. Class4mass : Utilisation d'un outil de clustering

1.1 Fonctionnement de l'algorithme

La première étape qui nous a permis d'identifier les thématiques émergentes a été d'utiliser un algorithme de clustering par le classifieur Class4mass[29].

Class4mass est codé en java et est composé de modules conçus pour extraire, à partir du modèle d'apprentissage automatique non supervisé, des *clusters*. Ces derniers sont des regroupements de documents textuels qui ont des traits sémantiques communs.

Au début du clustering, l'algorithme de classification déterminera seul la définition des classes et leur nombre. C'est le concept de l'apprentissage non supervisé. Il ne nécessite pas non plus de données d'entrée classées, c'est également à l'outil class4mass lui-même de saisir la structure des données et former des groupes de mots ayant des caractéristiques communes.

Class4mass a été un outil indispensable qui nous a permis d'avoir une première vision sur les thématiques qui peuvent exister en rapport avec le domaine des ressources humaines. A partir de cela, nous avons sélectionné les clusters qui nous intéressent parmi ceux qui étaient présentés et s'en servir de socle pour identifier par la suite les catégories.

La classe de cet outil « *HrClassifierClustering.java* » ne prend qu'un seul argument, c'est-à-dire un fichier de propriétés qui doit être situé dans le chemin de classe. Les attributs les plus

[28] **Clustering** : Lorsque les catégories ne sont pas prédéfinies mais développées dans la tâche de classification.
[29] **Class4mass** est un outil développé par la Sémantique, il repose sur le topic modelling fourni par Mallet. http://mallet.cs.umass.edu/topics.php

importants du fichier sont : *base.baseDir* (le répertoire où se trouve tout le corpus) et *base.appName* (qui fait référence au sous-répertoire du client spécifique).

1.2 Résultat du clustering

Pour donner suite à l'application de l'algorithme du clustering, les résultats du class4mass sont affichés dans le fichier **«** *clustering.in.html* **»** qui est interprétable par le navigateur Firefox (fortement recommandé, car Chrome ne semble pas enregistrer correctement les annotations).

En sortie nous avons obtenu plusieurs propositions de clusters définis par les outils de class4mass (Cf. figure 6). Ce fichier était utilisé pour observer manuellement les résultats du clustering.

Cet extrait du fichier clustering.in.html est le suivant : (nous ne présentons ici qu'une partie des catégories, les groupes de verbatim ne seront pas affichés pour des raisons de protection des données.)

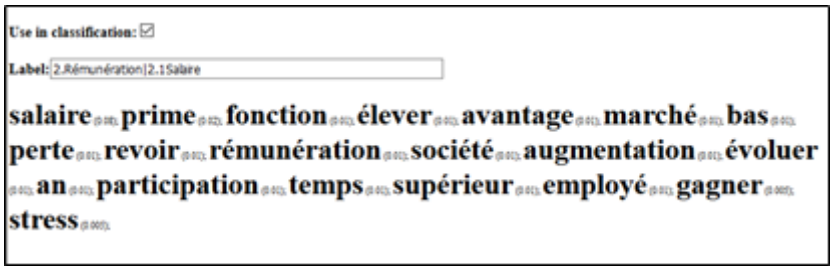

Figure 6. Une partie du fichier de sortie class4mass

Ces propositions nous permettent de trouver les noms pour chaque regroupement qui seront à l'origine de la taxonomie que nous expliquerons plus tard.
Cette étape va générer également un répertoire « *expansion* », contenant trois fichiers :

- Les expressions composées appelées *compounds*, par exemple esprit(\s)+d.(\s)+équipe, réseaux(\s)+sociaux ou open(\s)+source.
- Les mots non pertinents, *irrelevant words*, par exemple les mots « dommage, parfait, satisfaite, content ou super »,
- Les mots vides *Stopwords*, par exemple le, la, si, du, avec, vous.

Ces fichiers étaient utiles pour la réalisation du modèle d'entraînement lors de l'apprentissage automatique (Chapitre 3).

2. Elaboration de la maquette des catégories

La maquette des catégories a été mise en place afin d'assembler les catégories en une hiérarchie taxonomique < catégorie parent – catégorie enfant >. En effet, une catégorie parent, peut avoir des sous-catégories qui représentent des définitions plus spécifiques que la catégorie parent (appelé aussi classe supérieure). Pour notre travail, nous allons élaborer une maquette propre au domaine des ressources humaines.

2.1 Observation manuelle des résultats

La classification repose sur une maquette de catégories bien définie pour garantir une classification correcte à l'avenir. C'est donc un travail qui ne peut être ignoré car il affecte fortement l'exactitude des résultats. Au vue de ce besoin, nous proposons un travail d'analyse extrêmement précieux sur les catégories qui forment l'étape principale de la création de la maquette. Nous présentons la définition suivante pour la catégorie selon le dictionnaire Larousse : « Classe regroupant un ensemble d'éléments ayant un certain nombre de traits grammaticaux en commun et qui peuvent figurer dans un même environnement syntaxique. »

L'idée du notre travail était de répondre à la question : Quels sont les sujets évoqués par les collaborateurs ? Cette approche nous a permis, lors d'une première observation des résultats, de trouver les informations d'une thématique dans les réponses des collaborateurs, et ainsi regrouper un maximum de verbatim dans la maquette des catégories.

La constitution des catégories doit respecter les critères suivants :

- Catégories exhaustives : elles doivent couvrir l'ensemble des thématiques des verbatim.
- Catégories distinctes : elles ne doivent pas se recouper.
- Catégories objectives : les intitulés choisis doivent être clairs et concis car ils seront visibles dans l'interface. Pour chaque intitulé, nous avons ajouté une définition permettent de décrire ce que couvre une catégorie, et des exemples (entre 15 et 20 exemples) accompagnant chacune des catégories afin de faciliter leur interprétation. Les catégories sont susceptibles d'être réutilisées ou développées par d'autres linguistes qui doivent les interpréter de la même manière, ainsi que les clients.
- Nombre de catégories exploitables : la maquette doit avoir un maximum de 20 à 25 catégories.

- Catégories réalisables d'un point de vue technique : c'est-à-dire qu'elle doivent être linguistiquement concevables. C'est pour cela qu'un chef de projet linguiste peut être mis en relation avec le client pour pouvoir établir des catégories recevables.

2.2 Maquette des catégories « Ressources Humaines » obtenue

À partir des observations des résultats du clustering, nous avons fait des recherches sur le domaine des ressources humaines et nous nous sommes informés sur les sujets qui peuvent être traités dans ce domaine : les enjeux, les gestions et les tâches de la fonction ressource humaine, etc. Après avoir pris connaissance du domaine RH, les catégories ont été définies selon une approche linguistique. Cette approche consiste en un regroupement lexical dans le but de créer des catégories dont les mots sont proches sémantiquement et pour pouvoir distinguer ceux qui ne le sont pas. Par exemple, nous avons regroupé les verbatim abordant les thématiques similaires suivantes : le dialogue, la participation des collaborateurs, l'échange d'informations dans la catégorie commune « LA COMMUNICATION ». À ce stade, les catégories étaient définies avec soin afin de constituer une catégorisation pertinente.

Les catégories ont été assemblées dans un premier temps en catégories enfants représentant les sous-catégories, qui sont plus nombreuses et plus fines. Puis chaque catégorie enfant a été reliée à une catégorie parent plus générale qui regroupe les sous-catégories par thème. Par exemple, la catégorie parent « COMMUNICATION INTERNE » englobe les catégories enfants « COMMUNICATION D'ENTREPRISE » et « COMMUNICATION DES COLLABORATEURS ».

Nous présenterons conjointement les définitions pour chaque catégorie. Le tableau suivant comporte (de gauche à droite) : la catégorie parent, sa définition, les catégories enfants et leurs définitions.

4. COMMUNICATION INTERNE	*La communication interne entre les collaborateurs, le partage de l'information avec l'équipe, l'efficacité des réunions et la communication globale de l'entreprise.*	**4.1 COMMUNICATION DES COLLABORATEURS**	*Concerne la communication et l'échange d'information entre les collaborateurs.*
		4.2 LE TRAVAIL D'EQUIPE	*Décrit l'aspect d'équipe, le travail entre collaborateurs, leurs relations, les réunions d'équipes, le partage d'informations entre les groupes, l'efficacité des réunions, etc.*
		4.3 COMMUNICATION D'ENTREPRISE	*Comporte sur la communication globale de l'entreprise.*

Tableau 3. La maquette des catégories

Lors de la création de la maquette des catégories, nous avons obtenu plusieurs versions différentes. Nous avons modifié la maquette selon une approche linguiste et nous apporterons ces modifications dans la partie suivante.

2.3 Révision finale de la maquette

L'objectif de ce travail de catégorisation était d'avoir une maquette des catégories bien définie avec un bon niveau de précision. Nous avons donc adopté une démarche linguistique sur les contenus des catégories. La maquette a été consultée par une experte en enquête collaborateurs qui a une bonne connaissance et plusieurs expériences d'application dans le domaine de catégorisation. Elle nous a donné également accès aux questionnaires diffusés auprès des collaborateurs. Cela nous a permis d'avoir d'autres thématiques ainsi que l'enchaînement des sujets, ce qui nous a aidé à réviser la maquette avec plus d'informations. Le but de cet échange et cet accompagnement avec la consultante était de parvenir à un consensus sur la répartition et la délimitation des catégories. De ce fait, nous avons effectué des suppressions des catégories, des fusions, et des ajouts sur la première version. Il a fallu trouver des compromis sur certaines catégories, entre ce qui peut être interprété facilement par la machine et ce que les clients attendent de la catégorisation.

Nous avons retravaillé les catégories de la façon suivante :

1) Concernant la sous-catégorie « 2.3.Suivi », nous l'avons supprimé du fait qu'elle représentait une faible catégorie et semblait peu pertinente car elle était abordée à travers la catégorie « 2.2.Manager ».
2) D'autres sous-catégories ont été fusionnées en une seule catégorie plus globale. D'un point de vue linguistique, le contenu des deux catégories était indissociable. Les autres catégories n'ont pas été modifiées.
3) Enfin, nous avons ajouté une dernière catégorie **«** Hors catégorie **»** pour les verbatim qui n'ont pas de rapport avec les thématiques RH (par exemple "merci", "satisfait", "aucun commentaire", etc.)

La version finale de la maquette a été validée par l'experte en enquête collaborateur et a été composée de 6 catégories parents subdivisées en 24 sous-catégories.

Chapitre 3. Développement du classifieur «Ressources Humaines»

Dans notre état de l'art, nous avons pu observer différentes méthodes permettant de classer des documents textuels de façon automatique : d'une part les méthodes statistiques et d'autre part les méthodes symboliques.

La méthode statistique est une technique d'apprentissage automatique qui se base sur l'entraînement d'un modèle. Le but est de créer un corpus d'entraînement qui permet au classifieur de créer un modèle appris sur ce corpus. Ainsi, il pourra classer par lui-même les nouveaux documents textuels dans les catégories qu'il aura appris dans le corpus d'entraînement.

La méthode symbolique, quant à elle, repose sur une analyse sémantique lexicale du texte, couplée à des règles linguistiques expertes développées manuellement. Ces règles permettent de détecter des catégories uniquement en se basant sur le lexique.

Nous avons utilisé une combinaison de ces deux méthodes afin d'utiliser un système dit hybride. Nous présenterons cela ultérieurement dans la partie 3.

1. Elaboration du modèle d'entraînement

Le principe de la réalisation d'un modèle d'entraînement est d'avoir un système d'apprentissage automatique dans lequel l'algorithme va apprendre à classer automatiquement des nouvelles données (dans notre cas des verbatim) dans une ou plusieurs catégories. Pour cela, il doit y avoir un ensemble de textes annotés, dit ensemble d'apprentissage. Cette phase « d'apprentissage » ou « d'entraînement » est généralement réalisée à l'utilisation pratique d'un modèle. Pour ce faire, la première étape que nous avons élaboré était d'annoter le corpus qui servira à l'algorithme pour générer le modèle d'entraînement. Ce modèle doit être capable de faire des prédictions non seulement sur les données que nous avons utilisées pour le construire, mais surtout sur de nouvelles données (d'où le terme apprentissage, plutôt que mémorisation).

Comme nous l'avons vu précédemment, il faut une étape de normalisation des données qui doit être effectuée à chaque fois avant l'apprentissage lui-même, quel que soit l'algorithme utilisé. Ensuite, l'algorithme étudie les données en fonction des configurations (Cf. section1.3) sélectionnées pour calculer un modèle de classification. Enfin, une phase de test nous permettra une évaluation et une visualisation des performances du modèle.

Pour nos travaux, nous commencerons donc avec la première étape d'annotation présentée dans le point suivant.

1.1. Annotation du corpus avec Sherpa

L'étape d'annotation est requise pour l'apprentissage supervisé afin de pouvoir entraîner les classifieurs. Dans notre cas, le système de classification automatique repose en grande partie sur la qualité du modèle d'entraînement, il est donc indispensable de le construire avec précision et d'avoir une annotation correcte, c'est-à-dire complète et cohérente.

L'annotation manuelle est faite à l'aide de la plate-forme web « Sherpa » développée par Kairntech[30], une startup de la région grenobloise. Leur application est destinée à quiconque souhaite exploiter son capital de données linguistiques pour des tâches de TAL, en particulier l'annotation automatique de séquences textuelles (par exemple l'extraction d'entités nommées) ou la classification multi catégorielle de textes.

Sherpa propose une interface simple et fluide pour annoter des corpus de textes, puis permet de fabriquer (et d'évaluer) des systèmes d'annotation ou de classification automatique à partir de ce corpus. L'intérêt de Sherpa est qu'elle apprend à annoter (pratiquement en temps réel) à l'aide d'algorithmes d'apprentissage automatique. Sherpa suggère des annotations, dans le but d'accélérer la tâche d'annotation, qui n'est donc plus nécessairement purement manuelle mais devient semi-automatique si l'utilisateur le souhaite. Nous avons eu l'occasion d'expérimenter Sherpa sur nos données. En général, il y avait plusieurs annotations à faire pour chaque suite de mots étant donné que notre catégorisation est multi-label. Cela s'oppose à la tâche traditionnelle de classification dans laquelle chaque phrase n'est associée qu'à une seule catégorie.

Notre méthode d'annotation a été la suivante : pour chaque phrase, une ou plusieurs catégories étaient attribuées en fonction du contenu représenté. Pour cela, deux contraintes ont dû être respectées. Premièrement, lorsqu'il existe plusieurs catégories il est nécessaire de commencer par identifier celle qui semble la plus importante car seule cette catégorie sera utilisée comme données d'apprentissage. De plus, il faut donc être précis que possible afin d'identifier un maximum de fragments porteurs d'une catégorie, ce qui permettra par la suite des évaluations précises et une performance du système.

[30] https://www.kairntech.com/

Par exemple, le verbatim « pas assez de reconnaissance..... pas assez à l'écoute..... manque d'informations… », évoque trois sujets de catégories : «RECONNAISSANCE », « ECOUTE » et « COMMUNICATION ». Les *snippets*[31] suivants de ce verbatim seront annotés de cette façon :

RECONNAISSANCE	*pas assez de reconnaissance*
ECOUTE	*pas assez à l'écoute*
COMMUNICATION	*manque d'informations*

Tableau 4. Exemple des snippets annotés

Hormis la vingtaine de catégories-métier que nous avons définies, nous avons utilisé une catégorie nommée « Hors-catégorie » dans laquelle se trouvent par défaut tous les verbatim qui n'étaient pas classés et qui ne seront pas annotés car ils sont inexploitables et ne portent aucun intérêt pour notre catégorisation. En revanche, l'annotation des autres catégories était bien répartie. La figure suivante présente la répartition que nous avons eue pour chaque catégorie sur l'ensemble du corpus :

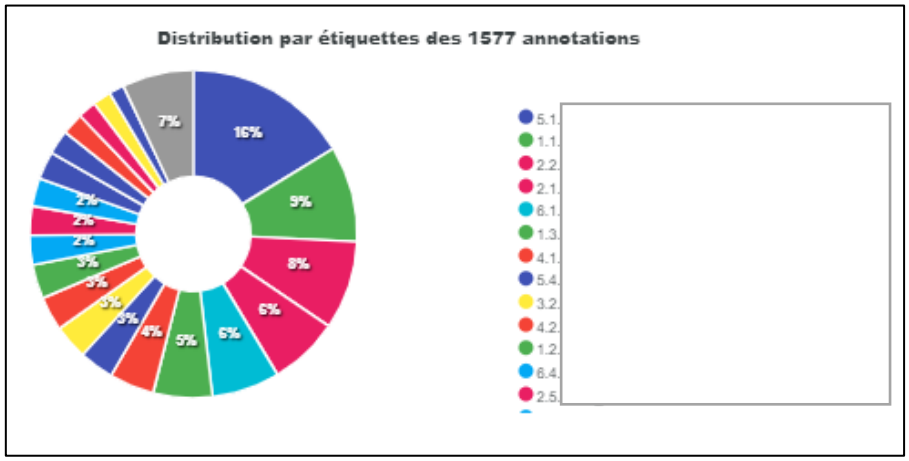

Figure 7. La répartition des catégories avec l'outil « Sherpa »

Trois catégories semblent être centrales : « REMUNERATION » (16%), « ENVIRONNEMENT DE TRAVAIL » (9%) et « LE MANAGER » (8%). Ces catégories présentent les sujets les plus abordés par les collaborateurs dans notre corpus.

En revanche, quelques catégories sont peu représentées, comme le cas de « COLLABORATEUR » ou « BIEN-ETRE » qui représentent chacun 2% des snippets.

[31] **Snippets** est un anglicisme employé pour désigner les fragments (continus ou discontinus) de texte.

1.2. Annotation du corpus avec Brat

Au cours de notre travail, nous avons eu une autre occasion de tester une autre approche d'annotation avec l'outil Brat[32], qui est un outil web conçu en particulier pour les annotations de texte. La différence entre les deux outils porte sur le fait que Brat permet de faire des annotations discontinues, cependant il n'inclut aucune composante d'apprentissage automatique. Ce type d'annotation peut être utilisée par exemple pour ce cas :

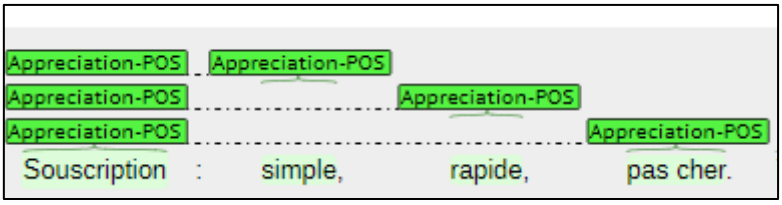

Figure 8. Exemple d'annotation sur Brat

Cette application, qui permet d'annoter les fragments pertinents au caractère près (plutôt qu'associer les étiquettes à tout le texte), facilite l'annotation car certaines formes lexicales sont plus importantes que d'autres surtout quand il s'agit d'identifier des classes différentes. Nous estimons qu'il n'est pas nécessaire de prendre la totalité de la phrase pour la classer. En effet, dans chaque verbatim il peut y avoir des parties qui ne sont pas très importantes pour la classification. Un des buts de notre approche consiste donc à sélectionner les segments les plus pertinents pour la classification des données et ignorer le reste. Cette étape d'annotation va nous permettre une sélection des verbatim qui expriment l'avis du client et une détection des alertes et de la polarité globale d'un verbatim : Neutre, Négative, Positive ou Mixte.

Les verbatim annotés peuvent être parfois courts et composés des mots simples ou bien des expressions plus longues et composées de plusieurs mots.

Une fois que la phase d'annotation est effectuée, nous entraînons notre modèle pour la classification de textes.

1.3. Réalisation finale du modèle d'entraînement

La construction des catégories est donc réalisée par apprentissage sur une partie des corpus d'entraînement. Plus le modèle est entraîné sur un grand ensemble de textes, plus le classifieur est susceptible de bien performer dans la classification de nouveaux documents, c'est-à-dire produire une classification similaire à celle qu'un humain pourrait produire. L'apprentissage

[32] **Brat :** https://www.aclweb.org/anthology/E12-2021.pdf

automatique est fait sur les données annotées avec un algorithme particulièrement adapté aux tâches de classification de texte choisi permettant de créer un modèle de classification.

À partir de notre corpus, nous avons éliminé les caractères à faible poids sémantique qui jouent rarement un rôle intéressant dans les recherches. Ces mots sont appelés les mots vides ou « Stopwords » et sont généralement des mots grammaticaux : les articles, les prépositions, les mots de liaisons, les déterminants, les conjonctions et les pronoms etc. Ces mots devaient être éliminés pour deux raisons : d'un point de vue linguistique, ces mots fournissaient peu (voire aucune) informations sur le sens d'un texte. Et d'un point de vue statistique, ces mots étaient nombreux sur l'ensemble des textes. Ils ne sont donc d'aucune aide pour la classification et pouvaient éventuellement ralentir le processus. À titre d'exemple, nous citerons, le cas des articles « le », « la », « les », des pronoms personnels « je », « nous », « vous » ou de certains mots de liaison « ainsi », « toutefois », etc. Les mots vides générés par notre système ont été créés pour la classification de relation client en amont de ce stage. Cette partie ne sera donc pas spécifique au domaine RH car elle n'a pas été développée par manque de temps. Cependant, nous aborderons les propositions possibles dans la partie des perspectives.

Le modèle s'est fait grâce à la classe « HrClassifierMainClassificationTraining.java », qui prend en entrée la taxonomie, un fichier d'entraînement annoté où pour chaque verbatim nous connaissons sa catégorie (Cf. tableau 5) et un fichier de configuration effectuant la représentation textuelle. Le but de ce fichier était de décrire le contenu du texte sous une forme compacte pouvant être utilisée par des algorithmes. Pour cela, nous avons choisi une liste de configurations (appelées « features » en anglais).

Catégories	Verbatim
1.1.ENVIRONNEMENT_DE_TRAVAIL	L'environnement de travail est très satisfaisant
1.1.ENVIRONNEMENT_DE_TRAVAIL	un endroit très équitable pour travailler
1.1.ENVIRONNEMENT_DE_TRAVAIL	amélioration de la ventilation

Tableau 5. Un extrait du fichier d'entraînement

L'entraînement de la méthode statistique se fait sur notre corpus en variant des différents features, qui peuvent être des mots, des groupes de mots, des expressions, des traits morphosyntaxiques, des traits sémantiques, etc., pouvant influencer les résultats de l'apprentissage automatique puisque le processus de classification de textes en dépend directement. Parmi les features sélectionnés, nous présentons :

- **UseNPs** : les groupes nominaux (dont la longueur peut varier entre deux et cinq tokens) issus de l'analyse syntaxique en dépendance.
- **UseRels** : les relations syntaxiques entre tokens.
- **UseSemFeat** : les traits sémantiques issus des tokenregex.
- **UseSemWord** : Les traits sémantiques issus des gazetteers.
- **USePOS** (PartOfSpeech) : la catégorie syntaxique des tokens.
- **UseLemma** : la forme lemmatisée des tokens. La lemmatisation ne retient pas le mot lui-même, mais la racine ou le lemme. Ce principe prend en compte les variations flexionnelles (singulier/pluriel, les conjugaisons, …) ou dérivationnelles (substantifs, verbes, adjectifs, …) en regroupant tous les mots d'une même famille sous un même terme, et donc ça contribue à l'amélioration de la classification.

Après l'étape de la méthode statistique et l'entraînement de la machine sur nos données annotées, nous verrons par la suite l'étape complémentaire sur les enrichissements sémantiques.

2. Enrichissement sémantique

La deuxième méthode que nous avons utilisée dans notre travail concerne l'enrichissement sémantique qui présente un feature d'apprentissage automatique. Nous allons tout d'abord extraire les termes avec lesquelles nous construisons la taxonomie semi-automatique. Et nous verrons par la suite les démarches adoptées pour la création des listes de termes et l'écriture des règles de classification.

2.1. Extraction semi-automatique de termes

2.1.1 Extraction des concepts

Compte tenu de notre taxonomie qui est basée sur le lexique, nous avons besoin d'extraire des concepts et construire des ressources sémantiques afin d'aider la machine à rapprocher les verbatim entre eux.

Pour ce faire, nous avons extrait les noms et groupes de noms qui sont spécifiques aux domaines étudiés à partir de nos deux corpus de départ. L'extraction a été faite par la classe Java « *XMLConceptOutput.java* » qui crée une liste dédoublonnée de concepts accompagnés de leurs scores (issus de chaque corpus). La sortie de cette classe est un fichier au format csv

(Cf. tableau 6). Nous avons obtenu une liste de 4116 concepts associés à leurs scores. Plus le score est élevé plus le concept est important dans le domaine.

Concept	Score
Production	9.7268
Communication	9.4593
Performance	5.7080
Haut direction	4.2786
Patron	0.9771
Reconnaissance	0.6538
Réduction	0.5493
Chauffage	0.5167
Apprentissage	0.4904
Travail d'équipe	0.4667
Salaire annuel	0.4554

Tableau 6. Exemple des concepts accompagnés de leurs scores

Après extraction des concepts, nous avons fait une sélection pour conserver les plus spécifiques au domaine des ressources humaines. Tout d'abord, nous avons sélectionné manuellement des concepts avec un score supérieur à 0.45 pour se limiter aux concepts qui se rapprochent le plus du domaine. Ensuite, nous avons regroupé les concepts ayant le même sens dans une liste regroupant tous les concepts triés. Par exemple, les mots « machines », « matériaux » et « outils » sont proches sémantiquement et renvoient à la même thématique, ils seront donc regroupés ensemble. Finalement, nous avons conservé 316 concepts qui ont constitué le socle de notre taxonomie que nous présenterons dans la section 2.2.

2.1.2 Word2vec

La limite de « l'extraction de concepts » est qu'elle porte uniquement sur des noms ou groupes de noms. Afin d'élargir notre liste, nous avons utilisé Word2vec développé par Mikolov et al (2013) pour obtenir des synonymes de verbes, adjectifs et adverbes sur notre liste de concepts.

Word2vec est un ensemble de modèles de langue utilisant des réseaux de neurones artificiels. Ces derniers sont entraînés à travers des exemples de mots et de leurs contextes à partir d'un corpus d'apprentissage pour créer des regroupements de mots entretenant des relations sémantiques. Ce modèle est prédictif, il attribue des probabilités aux mots et il est efficace dans les tâches de calcul de similarité.

Word2vec repose sur les méthodes distributionnelles. « *Ces méthodes permettent de mesurer la similarité distributionnelle indiquant le degré de cooccurrence entre un mot cible et son voisin apparaissant dans des contextes similaires.* » (Billami et Gala, 2016). Il y a plusieurs façons d'exprimer la même chose et un mot peut exprimer un même concept avec des expressions différentes. Pour extraire les synonymes, nous avons utilisé deux classes. La première est « *Word2vecCreateModel.java* » qui prend en entrée les corpus et fournit en sortie une matrice qui va nous servir pour la prochaine classe. La deuxième classe est « *GiveMeClosestNeighboursExcelFormat.java* ». Elle prend en entrée une liste de noms, pour en extraire tous les mots proches sémantiquement, et la matrice créée précédemment.

En sortie (figure 11), nous avons obtenu un fichier au format csv contenant une liste des mots (word), leurs voisins de mots qui sont sémantiquement proches et sélectionnés par word2vec (New word), et un score de cosinus. Plus le score du cosinus est grand plus le sens du mot est proche du mot cible. Par exemple, le mot « conditions » est plus proche du mot « environnement » que le mot « entreprises ».

Un tri manuel a été fait sur la liste des voisins. Voici un extrait des résultats :

Word	New Word	Cosinus	New Word	Cosinus	New Word	Cosinus	New Word	Cosinus	New Word	Cosinus
environnement	conditions	0,922429	flexibilité	0,915284	entreprises	0,892082				
salaire	payé	0,8685148	salariale	0,8621994	primes	0,8213886	besoins	0,8550488	moins	0,8352731
carrière	plan	0,9338451	opportunités	0,9268031	exigences	0,9042367	améliorer	0,9040549	compétences	0,8921616
patron	ressources	0,8937309	humaines	0,8796226	gestionnaires	0,8558515	former	0,6059387		
écoute	valoriser	0,8208068	savoir	0,7752202	personnes	0,7702606	responsables	0,7568141		
équipement	revoir	0,9481589	machines	0,9325276	manque	0,9324918	nécessaires	0,9253307	nouveau	0,9197538
charge	horaire	0,9830167	ans	0,9491993	bien-être	0,9229414	difficile	0,9467078	long	0,9416574

Figure 9. Extrait de l'extraction des concepts avec Word2vec

Nous avons pu obtenir des adverbes, des verbes et des adjectives afin d'enrichir notre liste de concepts. Cette méthode a comme avantage de regrouper les différents mots se rapportant au même concept. À titre d'exemple, le mot « salaire », le verbe « payer », l'adjectif « salariales » et l'adverbe « moins » seront rassemblés sous un concept commun nommé « #REMUNERATION ». Au total, nous avons obtenu 468 concepts pour la construction de la taxonomie que nous aborderons dans la partie suivante.

2.2. Construction de la taxonomie

Après avoir sélectionné les concepts propres au domaine des ressources humaines, nous avons élaboré deux autres éléments requis pour la construction de la taxonomie : la création des listes de termes et l'écriture des règles de classification.

Notre méthodologie pour la création de cette taxonomie s'appuie sur la constitution d'une ontologie. [Noy et McGuinness, 2000] définissent l'ontologie comme étant un vocabulaire commun destiné aux besoins des chercheurs pour partager une information. Nous avons alors mis en évidence les relations qui existaient dans un seul domaine.

Afin de mettre en place la taxonomie, nous avons commencé par construire un enrichissement sémantique qui peut être réalisé soit par enrichissement lexical à l'aide de gazetteers soit par des règles sous forme de tokenregex.

Tout d'abord, nous avons commencé par créer les gazetteers, qui sont présentés sous la forme de liste accompagnée d'annotations (tags sémantiques) et utilisés pour l'entraînement du modèle d'apprentissage pour aider la machine à corriger les erreurs et les lacunes des mots. Nous avons donc constitué des gazetteers pour chaque catégorie grammaticale (Part of speech). Pour cela, nous avons attribué manuellement un tag sémantique pour chaque concept. Les tags ont été choisis selon les relations sémantiques des concepts et regroupés par des relations hiérarchiques. Par exemple, le mot « manager » et « leadership » désignant un interlocuteur de la société peuvent être présentés dans la taxonomie accompagnés d'un même tag sémantique : SOCIETE#MANAGEMENT#INTERLOCUTEUR. Tandis que le mot « directeur » est présenté dans SOCIETE#DIRECTION#INTERLOCUTEUR, car il a une fonction différente des deux mots précédents. Chaque entrée contient donc deux parties, séparées par une tabulation, dans le format suivant : CONCEPT <TAB> TAG. Les gazetteers sont exprimés sous forme d'expression régulières JAVA définis manuellement. La figure 12 ci-dessous illustre la forme des gazetteers et quelques exemples d'expressions régulières.

```
directeur    SOCIETE#DIRECTION#INTERLOCUTEUR
direction    SOCIETE#DIRECTION#INTERLOCUTEUR
boss?e?s?    SOCIETE#DIRECTION#INTERLOCUTEUR
dirigeants?  SOCIETE#DIRECTION#INTERLOCUTEUR
ex-patron    SOCIETE#DIRECTION#INTERLOCUTEUR
patrons?     SOCIETE#DIRECTION#INTERLOCUTEUR
```

Figure 10. Exemple du gazetteers exprimés à l'aide des expressions régulières

Les gazetteers permettent de désambiguïser certains lemmes. Par exemple le mot « responsable » qui se trouve, à la fois, dans le fichier de gazetteers des adjectifs avec le tag sémantique TRAVAIL#MISSIONS et dans le fichier de gazetteers des noms avec le tag sémantique SOCIETE#MANAGEMENT#INTERLOCUTEUR . Le gazetteers est enrichi avec son

tag sémantique TRAVAIL#MISSIONS que lorsqu'il est analysé comme adjectif et non comme un nom. Ainsi, nous avons constitué quatre fichiers des gazetteers propres au domaine des ressources humaines : un contenant les noms, puis les adjectifs, les verbes et enfin les adverbes.

Ensuite, nous avons développé les *tokenregex*[33] qui sont des règles linguistiques définies manuellement par un linguiste, dont la syntaxe s'inspire des expressions régulières opérant sur des tokens et tenant compte des informations morphosyntaxiques.

Développés par le Natural Language Processing Group de l'université de Stanford[34], les tokenregex sont utilisés dès l'entraînement du modèle pour aider le machine learning à désambiguïser les groupes de mots, les mots composés ou les expressions polylexicales qui ne sont pas modélisables par les gazetteers. Cependant, les tokenregex se fondent sur les mêmes tags sémantiques que ceux définis lors des gazetteers. Le linguiste peut modeler ces règles selon le besoin final attendu et ajouter de nouvelles règles pour déterminer les relations qui l'intéressent, il peut les modifier, augmenter ou diminuer les traits sémantiques sur les mots, etc. Ce travail a pour but d'améliorer la pertinence des résultats.

Prenons l'exemple suivant (figure 13), qui montre une règle annotant le groupe de mots « équipe de direction ».

```
#entry_to_annotate    équipe de direction        SOCIETE#DIRECTION#INTERLOCUTEUR
{ pattern:   (   [  {lemma:/[éeè]quipe/}] /.+/{0,2}
          [  {  sa:/SOCIETE#DIRECTION#INTERLOCUTEUR/  }  ]  )  ,
  action: ( Annotate ( $0,sa,"SOCIETE#DIRECTION#INTERLOCUTEUR") )  }
```

Figure 11. Exemple de tokenregex

Cette tokenregex s'interprète de la façon suivante : si nous rencontrons le token « équipe » sous ses différentes formes, suivi de zéro à deux mots (pour les articles ou les prépositions qui peuvent intégrer la phrase) et suivi de n'importe quel terme appartenant à l'annotation sémantique SOCIETE#DIRECTION#INTERLOCUTEUR, nous mettons directement tout le verbatim dans cette annotation attribuée. Nous avons utilisé le « lemme » pour chaque token afin d'assurer la prise du lemme du mot et non seulement le mot lui-même.

[33] https://stanfordnlp.github.io/CoreNLP/tokensregex.html
[34] https://nlp.stanford.edu/software/tokensregex.html

Malgré toutes ces règles mise en place, certains verbatim restent non classés et à améliorer. Pour cela, un autre type de règle s'ajoute à nos règles lexicales.

2.3. Mise en place de règles « Boost »

Les règles Boost aident à résoudre les erreurs de catégorisation lors de la détection d'un certain mot spécifique et pour les catégories avec peu d'exemples classés. Pour les verbatim mal classés ou ceux se retrouvant en hors catégorie alors qu'ils devraient être classés, nous avons développé des tokenregex « Boost ». Ces règles ajoutent 1 au score de classification, d'où le terme boost. Ils présentent un impact très fort sur les résultats ce qui incite à ne les utiliser que lorsque ces règles s'avèrent indispensables car elles influencent fortement la catégorisation. Généralement, elles sont utilisées dans le cas des expressions idiomatiques[35] ou des catégories avec des représentations très faibles, lorsque les données d'entraînement ne suffisent pas pour construire un modèle solide en méthodes statistiques. Par exemple, les annotations CONTRAT#COLLABORATEURS et CONTRAT#RUPTURE#COLLABORATEURS doivent systématiquement être classées dans la catégorie « 6.4.COLLABORATEURS ». Cependant, la machine n'a pas su les classer correctement avec seulement l'apprentissage et l'enrichissement sémantique. Il est donc nécessaire dans ce cas d'utiliser les règles boost sur les deux annotations pour « forcer » leur classification.

L'écriture des règles boost est une étape qui demande de la rigueur. En effet, écrire des règles trop générales peut fausser les résultats. En revanche, établir des règles trop spécifiques pour certains verbatim peut ne pas couvrir assez d'information pour les nouvelles entrées. Par conséquent, il faut trouver un équilibre dans l'écriture des règles en essayant toujours de trouver la règle qui produit le moins de bruit. Plus les règles sémantiques sont bien définies, plus le modèle d'entraînement réussira à classer les verbatim et ainsi assurer une performance correcte.

[35] **Expression idiomatique** : une construction ou une locution particulière à une langue, qui porte un sens par son tout et non par chacun des mots qui la composent. Il peut s'agir de constructions grammaticales ou, le plus souvent, d'expressions imagées ou métaphoriques.
https://fr.wikipedia.org/wiki/Idiomatisme#Expression_idiomatique

Partie 3.

EVALUATIONS ET RESULTATS

Pour déterminer les performances du classifieur sur nos corpus, nous avons effectué plusieurs mesures d'évaluation. Nous avons utilisé les calculs classiques du traitement automatique des langues : la précision, le rappel et le F-mesure. Les évaluations ont été conduites avec différentes configurations telles que la variation des features de l'algorithme du machine learning ou l'ajout ou le retrait des enrichissements sémantiques.

Nous présenterons tout d'abord les méthodes de calcul, ensuite les différents features testés et nous finirons par la présentation des résultats obtenus.

Chapitre 1. Evaluation du classifieur « Ressources Humaines »

1. Méthodes de calcul

Afin de prendre conscience de l'impact des différents composants du système de classification automatique (notamment l'enrichissement sémantique), nous avons réalisé une série d'évaluations sur la classification des catégories.

L'outil d'évaluation que nous avons utilisé est développé en Java et permet de calculer certaines mesures dont : la précision, le rappel et la F-mesure. Ces mesures sont expliquées dans l'état de l'art. À cela s'ajoute l'accuracy[36] , l'écart type[37] du F-mesure (*Standard Deviation*) et la médiane[38] du F-mesure.

La classe d'évaluation utilise 80% du corpus pour créer un gold standard qui sert d'étalon d'entraînement sur lequel le classifieur fait son apprentissage et les 20% serviront d'ensemble de test sur lequel nous pouvons évaluer sa performance. Ces dernières seront utilisées de façon aléatoire lors de la phase de test. Le principe du système est d'attribuer une probabilité d'appartenance (un score) pour toutes les catégories. Pour chaque catégorie, nous avons calculé une moyenne des scores sur cinq itérations d'entraînement et d'évaluation. C'est cette moyenne qui sera prise en compte lors des résultats. La valeur de la F-mesure peut aller de 0 à 1 (ou en pourcentages de 0% à 100%). Nous avons répété ce processus de calcul (appelé cross-validation) entre trois à cinq fois, afin de lisser les erreurs attribuables aux aléas de l'échantillonnage et nous assurer d'avoir une estimation plus fiable. Enfin, nous avons fait

[36] L'accuracy est la proportion de documents correctement classés.
[37] L'écart-type sert à mesurer la dispersion autour de la moyenne.
[38] Pour calculer la médiane, il faut d'abord ordonner les données (les trier dans l'ordre croissant). La médiane sera le nombre qui se situe au point milieu.

varier les features décrit dans le chapitre 3 section 1.3, afin d'évaluer et améliorer le classifieur à la suite des résultats obtenus.

2. Configurations évaluées

Pour l'évaluation, nous avons utilisé les features déjà définis dans le chapitre 3 et les enrichissements sémantiques : gazetteers, tokenregex et règles boost. Pour chaque série de calcul, nous avons alterné l'application des différents features comme nous pouvons le voir dans le tableau suivant (Cf. figure 14). Cela nous a permis d'observer pour chaque configuration testée les conséquences que cela avait sur la moyenne.

useNPs	gazetteers	TRs	BOOST	useRels	useSemFeat	useSemWord	usePOS	useLemma
-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
3	+	-	-	-	+	+	-	-
3	+	+	-	-	+	+	-	-
3	+	+	+	-	+	+	-	-
3	+	+	+	+	+	+	-	-
3	+	+	+	+	+	+	+	-
3	+	+	+	+	+	+	+	+
3	+	+	-	+	+	+	+	+
2	+	-	+	+	+	+	+	+
3	+	-	-	+	+	+	+	+
4	-	-	-	+	-	-	+	+
3	+	+	+	+	+	+	+	-
3	+	+	+	+	+	+	+	no Stopwords
3	+	+	+	+	+	+	+	no Stropwords no Synonyms

Figure 12. Exemple des features

Dans un premier temps, nous avons commencé par évaluer notre corpus sans aucun feature, ensuite nous avons ajouté au fur et à mesure les différents features en variant les croisements de ces derniers. Les (+) montrent la présence du features, ce qui équivaut à « True », en revanche les (-) présentent l'absence du feature, ce qui équivaut à « False ». Les traits sémantiques SemFeat et SemWord correspondent à l'ajout des gazetteers. La forme lemmatisée des tokens (useLemma) peut être variée avec ou sans les stopwords et les synonymes, afin de voir l'impact de ceux-ci sur nos résultats. Les groupes nominaux (useNPS), dont la longueur varie entre 2 et 5, sont généralement fixés sur le 3 car il s'agit du cas de figure le plus fréquent.

Nous verrons lors du point suivant les impacts de chaque modification apportée au système.

Chapitre 2. Résultats

1. Observation et bilan des premiers résultats

Après la phase des évaluations, voici dans cette partie les résultats obtenus. La figure 15 présente un aperçu de ce que nous avons obtenu pour le score de F-mesure, ainsi que la précision, le rappel et la moyenne.

useNPs	gazetteers	TRs	BOOST	useRels	useSemFeat	useSemWord	usePOS	useLemma	Precision	Recall	Accuracy	F-mesure
-	-	-	-	-	-	-	-	-	XX%	XX%	XX%	XX%
3	+	+	+	+	+	+	+	+	XX%	XX%	XX%	XX%
3	+	+	+	-	+	+	-	+	XX%	XX%	XX%	XX%
3	-	+	+	+	+	+	+	+	XX%	XX%	XX%	XX%
3	-	-	-	+	+	+	+	+	XX%	XX%	XX%	XX%

39

Figure 13. Résultats de la F-mesure, la précision, le rappel et la moyenne avec la variation des features

Lors des premières évaluations, nous avons remarqué que l'ajout de l'ensemble des features permettent d'améliorer les résultats. En revanche d'autres features ne contribuent pas à cette amélioration tel que le *POS* (part-of-speech) et les relations de dépendances syntaxiques. Nous avons noté un meilleur F-mesure avec la présence des enrichissements sémantiques (gazetteers, tokenregex et boost), une précision de XX% et un rappel de XX%. La précision caractérise un niveau de fiabilité très important, cela signifie que les catégories ont été bien classées mais peuvent être améliorées. Nous notons qu'une faible précision indique que le système génère du bruit : C'est-à-dire qu'il y a des verbatim qui ne sont pas classés dans la bonne catégorie. Et un rappel faible se manifeste par du silence, ce qui signifie que certains verbatim qui relèvent d'une catégorie ne sont pas détectés dans celle-ci. Nous avons également remarqué la grande différence entre les deux scores XX%, sans enrichissements sémantiques, et XX% avec enrichissements sémantiques. Nous avons donc évalué ces enrichissements séparément, le tableau suivant nous montre ces résultats :

[39] Les chiffres sont anonymisés pour des raisons de privacité.

	Résultats		
	Precision	**Recall**	**F-mesure**
Gazetteers	XX%	XX%	XX%
Tokenregex	XX%	XX%	XX%
Boost	XX%	XX%	XX%
Gazetteers + Tokenregex	XX%	XX%	XX%
Gazetteers + Boost	XX%	XX%	XX%
Tokenregex + Boost	XX%	XX%	XX%

Tableau 7. Les résultats de la première évaluation avec les enrichissements sémantiques

Nous avons remarqué que le meilleur score est obtenu par la présence des gazetteers et les tokenregex, mais à l'absence des gazetteers le score baisse de 10 points de pourcentage. Ceci nous montre que les gazetteers et les tokenregex sont plus perfectibles qu'avec les règles boost. Afin d'améliorer et de rendre plus performant notre classifieur, nous avons étudié les autres features en les ajustant et en renforçant les règles établies.

2. Ajustements des paramètres

À partir des premiers résultats obtenus, nous avons remarqué que certaines catégories se retrouvent avec un faible ou zéro score. En effet, le classifieur produit un classement de catégories par score de confiance (pouvant varier entre 0 et 1), nous retiendrons uniquement les catégories dont le score dépasse un seuil fixé empiriquement. Pour cela, certaines améliorations ont été apportées à la fois aux scores et aux enrichissements sémantiques.

À partir de nos résultats, un seuil a été fixé pour chacune de nos 24 catégories. (Cf. tableau 8). Il s'agit d'appliquer un seuil qui est un paramètre afin de limiter le score au-delà duquel la catégorie doit être attribuée. Les seuils que nous avons appliqués varient entre 0.1 et 0.65 (Cf. Annexe 1 : les seuils attribués aux catégories).

1.1.ENVIRONNEMENT_DE_TRAVAIL= 0.6
1.2.RESSOURCES_POUR_LE_TRAVAIL= 0.25
1.3.HORAIRES_CHARGES_DE_TRAVAIL= 0.35
2.1.LA_DIRECTION= 0.4
2.2.LE_MANAGER= 0.5

Tableau 8. Extrait des seuils attribués aux catégories

Tous les autres verbatim en-dessous du seuil seront classés automatiquement dans « Hors catégorie ». Cette procédure nous permet de limiter les verbatim et de garder ceux qui sont pertinents.

La deuxième amélioration concerne les enrichissements sémantiques, nous avons ajouté des tags sémantiques plus spécifiques à notre liste de gazetteers et également des règles de boost sur les verbatim (Cf. tableau 9) ayant du mal à se classer dans leur propre catégorie.

#pas de sourires { pattern : ([{sa:/AMBIANCE/}]), result : (HolmesGroup($0, "command", "action:BOOST; target_cat:1.5.AMBIANCE")), Name: "1.5.AMBIANCE" }
#aucun signe de tête { pattern : ([{sa:/AMBIANCE/}]), result : (HolmesGroup($0, "command", "action:BOOST; target_cat:1.5.AMBIANCE")), Name: "1.5.AMBIANCE" }

Tableau 9. Exemples d'amélioration pour les règles boost

Pour chaque modification apportée, nous avons revu en parallèle le modèle du machine learning en modifiant certaines annotations. Par exemple, en améliorant les scores des deux verbatim « pas de sourires » et « aucun signe de tête », nous avons annotés ceux-ci dans la catégorie « 1.5.AMBIANCE », afin d'être pris en compte dans le modèle d'entraînement.

Pour apporter toutes ces améliorations, nous avons utilisé une démarche itérative pour réentraîner notre corpus et nous avons relancé à chaque fois les calculs. Il a fallu plusieurs itérations afin d'améliorer les résultats. Voici les résultats finaux obtenus à la suite de nos évolutions :

useNPs	gazetteers	TRs	BOOST	useRels	useSemFeat	useSemWord	usePOS	useLemma	Precision	Recall	Accuracy	F-mesure
-	-	-	-	-	-	-	-	-	XX%	XX%	XX%	XX%
3	+	+	+	+	+	+	+	+	XX%	XX%	XX%	XX%
3	+	+	+	-	+	+	-	+	XX%	XX%	XX%	XX%
3	-	+	+	+	+	+	+	no stopwords no synonyms	XX%	XX%	XX%	XX%
+	-	-	-	+	+	+	+	+	XX%	XX%	XX%	XX%

Figure 14. La variation des features ainsi que la F-mesure, la précision, le rappel et la moyenne après les améliorations

La première remarque d'ensemble que nous pouvons faire, est que les résultats de chaque feature se sont améliorés en comparaison avec les premiers résultats et le meilleur score obtenu de F-mesure est passé de XX% à XX% actuellement. Cette augmentation montre que la manipulation des seuils et l'utilisation des ressources linguistiques pour l'enrichissement sémantique ont un impact positif sur la catégorisation effectuée par notre système.

Voici les nouveaux résultats obtenus sur les mesures des enrichissements sémantiques testés séparément :

	Résultats		
	Precision	**Recall**	**F-mesure**
Gazetteers	XX%	XX%	XX%
Tokenregex	XX%	XX%	XX%
Boost	XX%	XX%	XX%
Gazetteers + Tokenregex	XX%	XX%	XX%
Gazetteers + Boost	XX%	XX%	XX%
Tokenregex + Boost	XX%	XX%	XX%

Tableau 10. Résultats de la deuxième évaluation avec les enrichissements sémantiques

Nous avons remarqué que le plus faible score de F-mesure est toujours en absence des gazetteers. En revanche, après les améliorations ajoutées par la suite des premiers tests d'évaluations, nous avons remarqué que les règles boost se sont améliorées d'où le meilleur score de F-mesure XX% en présence des gazetteers et des boost.

Nous avons constaté que les scores du rappel et de la précision présentent une tendance intéressante : pour chaque calcul, nous avons observé une plus grande augmentation du rappel que de la précision. Cela montre que nous pouvons effectuer une amélioration plus facile sur

le rappel que sur la précision, en ajoutant des entrées de la taxonomie et des règles de classification pour détecter des documents plus pertinents.

Nous confirmons que l'amélioration des performances du classifieur que nous avons mis en place est due en grande partie à l'enrichissement sémantique et chaque règle symbolique avait apporté de meilleurs résultats. Nous avons par la suite testé le classifieur sur d'autres données clients pour évaluer sa robustesse lorsqu'il est testé sur des nouvelles données.

3. Conclusion

Le travail sur l'évaluation était une tâche longue. Notamment à cause des différents cycles de modification et d'exécution d'un grand nombre de features. Cependant, ce travail était nécessaire car, comme nous avons pu le remarquer lors des évaluations, nous avons pu observer une net amélioration des résultats issus des enrichissements sémantiques ainsi que la qualité et les performances du classifieur. Le travail n'est toutefois pas terminé, certaines modifications des features doivent encore être apportées, notamment sur les enrichissements sémantiques. Au niveau des gazetteers, certains tags sémantiques peuvent être développés. Par exemple le tag #CLIENT peut être affinés selon le contexte en #ENTREPRISE#CLIENT#RECLAMATION ou #ENTREPRISE#CLIENT#LIVRAISON. Enrichir par la suite les règles sémantiques et varier les seuils d'attribution à chaque cycle de modification en fonction des performances des catégories.

Chapitre 3. Application du classifieur et perspectives d'amélioration

1. Application du classifieur sur des cas spécifiques clients

L'évaluation de notre système nous a montré des résultats encourageants en termes de rappel et de précision. Nous avons donc appliqué notre classifieur sur deux autres corpus issus d'enquêtes « collaborateurs ».

Le premier était semblable aux corpus utilisés pour générer le modèle, les verbatim ont donc bien été classés malgré la petite taille du corpus. Nous avons détecté quelques nuances dans les contenus. Par exemple, dans la sous-catégorie « 6.1.Image », les sujets sur la responsabilité sociétale des entreprises, les égalités homme / femme et le plan de carrière de l'entreprise n'ont pas été évoqués par les collaborateurs.

En revanche, le deuxième corpus a été issus des enquêtes qui sont envoyées fréquemment et de manière régulière aux collaborateurs, contrairement aux enquêtes classiques annuelles. L'avantage de ces enquêtes est de suivre les collaborateurs tous les mois ou trimestres et ainsi obtenir davantage d'informations précises sur l'expérience en entreprise vécue par les collaborateurs**.** Ainsi, la maquette qui a été faite pour ces nouvelles enquêtes nous a révélée quelques améliorations au niveau des catégories. En effet, les enquêtes de ce corpus ont été faites entre la période du confinement et du déconfinement. Les collaborateurs ont donc évoqué plusieurs fois le sujet du « télétravail » et de la « sociabilité », qui étaient jusqu'alors absents de notre catégorisation. Néanmoins, le classifieur a pu classer avec succès les nouveaux verbatim parmi les catégories existantes.

2. Perspectives d'évolution du classifieur

Après l'application du classifieur sur d'autres données clients, nous prévoyons d'apporter plusieurs modifications afin de faire évoluer positivement le classifieur sur le plan précision et interprétabilité.

Nous souhaiterons intégrer les deux nouveaux corpus dans le système, ceci permettra d'augmenter la taille du corpus et ainsi avoir plus de verbatim à classer. Plus il y a de données injectées dans le système Machine Learning, plus il peut apprendre et fournir des résultats de qualité supérieure.

Comme nous avons pu le remarquer, nous avons utilisé certains fichiers à disposition dans la plateforme Holmes, qui ont été développés en amont de ce stage. Il s'agit principalement des fichiers des expressions composées « compounds », les mots non pertinents « irrelevant words » et les mots vides « stopwords», ainsi que le fichier de synonymes utilisé par le machine learning. Cependant, une perspective que nous envisageons serait de créer des fichiers propres au domaine des ressources humaines car ils peuvent avoir une influence sur les performances du classifieur.

De même, nous souhaiterions améliorer la maquette des catégories. Tout d'abord, la catégorie « 3.1.COMPETENCES » qui peut intégrer certains de ces verbatim dans d'autres catégories. Par exemple, les verbatim liés aux compétences du manager ou de la direction « *le manager n'est pas compétent* ». D'autres pouvant être dans la catégorie « 5.4.RECONNAISSANCE » comme le cas de « *faire valoir les compétences* ». Certain verbatim de cette catégorie sont très génériques et cela reste difficile pour la machine de les classer. Ensuite, nous pouvons améliorer la catégorie « 4.2.TRAVAIL D'EQUIPE ». Nous avons remarqué une confusion due à l'ambiguïté des deux mots 'groupe' et 'équipe'. En effet, le mot 'groupe' employé par les collaborateurs désignait quelquefois la – société – entreprise, alors que dans d'autres cas il désignait l'équipe dans le verbatim « *travail de groupe* ». Enfin, comme nous avons pu le voir grâce à d'autres enquêtes collectées, nous pouvons inclure de nouvelles catégories, telles que le « télétravail », la « sociabilité », la « mobilité », etc., ou améliorer certaines catégories existantes comme la « 1.4.SANTE_SECURITE » pouvant être liée à ce sujet.

Nous pensons qu'il serait envisageable d'améliorer les évaluations et avoir un système avec un score de F-mesure supérieur à 80%. Ceci demandera d'améliorer les enrichissements sémantiques ainsi que le modèle d'entraînement.

CONCLUSION GENERALE

1. Conclusion et perspectives

Les classifieurs représentent un outil pratique pour les entreprises, leur avantage réside dans leur manipulation intuitive et leur simplicité. Ils se composent de règles linguistiques qui sont faciles à interpréter par l'expert humain, ce qui est l'un des facteurs importants dans l'acceptation et l'utilisation pour des solutions de satisfaction client.

L'objectif de ce stage a été de développer un classifieur automatique multi-catégoriel destiné aux enquêtes collaborateurs et basé sur un système hybride. C'est une approche nouvelle qui apporte beaucoup au domaine de l'apprentissage automatique. En effet, elle permet de mettre en place une méthode statistique robuste tout en orientant l'entraînement basée sur la configuration manuelle de la méthode symbolique. Comme nous l'avons souligné tout au long de ce travail, ce type d'approche permet également de corriger de manière significative les erreurs et les lacunes de l'apprentissage automatique sur les catégories les moins représentées du corpus d'entraînement. Ce travail a donné des résultats prometteurs, le classifieur a pu catégoriser les verbatim selon une vingtaine de catégories génériques au domaine des ressources humaines (elles-mêmes regroupées en six catégories de niveau supérieur). Nous avons également montré que tous ces apports sont positifs car ils contribuent aux performances de celui-ci. Les différentes étapes de ce travail nous ont permis d'acquérir méthodologie et expertise dans plusieurs domaines. Tout d'abord, notre processus de classification est basé sur la maquette des catégories, il a fallu que ces dernières soient représentatives. Ce travail était très minutieux et n'a été abouti qu'après nombreuses heures. Mais les résultats obtenus ont été très satisfaisants. D'autre part, une des problématiques de cette méthode a été de construire un corpus d'entraînement qui soit représentatif de notre base de travail. La tâche d'annotation sur Sherpa a été une première expérience à la fois pour moi et pour l'entreprise. Par la suite, le développement de la méthode symbolique a été effectué avec succès. Nous avons enrichi les données existantes avec des données externes afin de fournir plus d'informations à l'algorithme, ce qui améliore le modèle. Nous sommes satisfaits des enrichissements sémantiques que nous avons exploités. Ceux-ci se sont avérés être performants après les expérimentations que nous avons réalisées. Concernant les évaluations faites sur notre système, nous avons varié les features dans le but d'évaluer les performances du classifieur. Nous pouvons dire que les résultats des évaluations en termes de précision et de rappel peuvent être jugées comme satisfaisants. Enfin, la manipulation de toutes ces données linguistiques, qui nous ont permis de travailler sur la mise en place d'une plateforme Eloquant, nous ont permis de mettre en évidence l'importance du travail de linguiste au sein même de l'entreprise. Pour

conclure, ce projet est encore perfectible dans son ensemble et reste utilisable pour de futurs développements. Comme nous l'avons déjà évoqué, en implémentant de nouvelles données les résultats obtenus pourraient être meilleurs. Cependant, à l'issu de ce stage, celui-ci reste utilisable en tant que tel.

2. Bilan et acquis

Travailler chez Eloquant a été très enrichissant. Ce stage a été, pour moi, une première expérience personnelle dans le monde du TAL en entreprise. Ce stage m'a énormément appris sur le métier du linguiste informaticien et surtout m'a conforté dans mon choix de continuer dans cette voie. Ces 6 mois ont été très bénéfiques car j'ai beaucoup appris et j'ai pu mettre en pratique les connaissances acquises tout au long de ces années d'études. J'ai créé des ressources linguistiques, élaboré une taxonomie, travaillé en Machine Learning et développé un outil de traitement automatique de la langue, tout en étant confrontée aux difficultés réelles du monde du travail. De plus, des nouvelles connaissances ont été acquises sur le domaine des ressources humaines et les enquêtes collaborateurs. J'ai appris à développer mon esprit critique, pratiquer mon sens du partage, améliorer mes capacités d'organisation et avoir l'aspect d'équipe avec une réelle liberté d'expression et des personnes en face attentives et disponibles. J'ai également pu participer aux réunions hebdomadaires de l'équipe, voir un autre côté du travail à la fois clients et produits et j'ai vraiment apprécié le fait d'être incluse dans l'entreprise. D'autre part, j'ai pu découvrir le télétravail durant la crise sanitaire du COVID-19. Cette expérience m'a appris avant tout l'autonomie et le développement personnel. Ce changement imprévu ne m'a pas trop affecté dans mon travail car l'équipe Eloquant a très bien su gérer cette période et a su humaniser nos échanges malgré la distance.

Pour terminer, je pense que tous ces acquis me permettront de démarrer en bien ma carrière de linguiste informaticienne avec beaucoup de positif en tête et plus de confiance en moi.

Bibliographie

Abadie, N., Mustière, S. (2008). *Constitution d'une taxonomie géographique à partir des spécifications de bases de données.* Colloque International de Géomatique et d'Analyse Spatiale SAGEO 2008, Montpellier, France. ffhal-02411372

Acosta, A., Bittar, A. (2007). *LAGRATOUNETTE : classification automatique générique de textes d'opinion*. LATTICE-CNRS (UMR 8094), Université Paris 7.

Bachimont, B. (2000). *Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances*. Ingénierie des connaissances : évolutions récentes et nouveaux défis. Paris: Eyrolles.

Billami, B. M., Gala, N. (2016). *Approches d'analyse distributionnelle pour améliorer la désambiguïsation sémantique. Journées internationales d'Analyse statistique des Données Textuelles (JADT)*. Nice, France.

Caillet, M., Pessiot, J.-F., Amini, M.-R., Gallinari, P. (2004). Unsupervised Learning with Term Clustering for Thematic Segmentation of Texts. Paris, France.

Geibler, S. (2020). *The Kairntech Sherpa – An ML Platform and API for the Enrichment of (not only) Scientific Content*. Meylan, France.

Grouin, C. (2013). *Anonymisation de documents cliniques : performances et limites des méthodes symboliques et par apprentissage statistique*. Bio-informatique [q-bio.QM]. Université Pierre et Marie Curie - Paris VI. Français. fftel-00848672

Jalam, R. (2003). *Apprentissage automatique et catégorisation de textes multilingues.* Thèse de doctorat, Université Lumière Lyon 2, France.

Kessler, R., Torres-Moreno, JM., El-Beze, M. (2004). *CLASSIFICATION THÉMATIQUE DE COURRIELS AVEC APPRENTISSAGE SUPERVISÉ, SEMI SUPERVISÉ ET NON SUPERVISÉ*. Université d'Avignon et des Pays de Vaucluse - Avignon, France.

Korde, V. (2012). *TEXT CLASSIFICATION AND CLASSIFIERS: A SURVEY*. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2. Sardar Vallabhbhai National Institute of Technology, Surat.

Maurel, S., Curtoni, P., Dini, L. (2007). *Classification d'opinions par méthodes symbolique, statistique et hybride*. Actes de DEFT'07, Grenoble, France.

Maurel, S., Dini, L. (2009). *Exploration de corpus pour l'analyse de sentiments,* « DÉfi Fouille de Textes ». Atelier de clôture, Paris.

Mondary, T., Després, S., Nazarenko, A., Szulman, S. (2008). Construction d'ontologies à partir de textes : la phase de conceptualisation. *19èmes Journées Francophones d'Ingénierie des Connaissances* (IC 2008). Nancy, France.

Mustière, S., Abadie, N., Aussenac-Gilles, N., Bassagnet, MN., Kamel, M., et al. (2009). GéOnto : *Enrichissement d'une taxonomie de concepts topographiques*. Spatial Analysis and GEOmatics Sageo 2009, Paris, France. ffinria-00432628

Noy, N., McGuinness, D. (2000). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA.

Ouali, C. (2014). *Classification automatique de textes*. Mémoire de master, Université de M'SILA, Algérie.

Poudat, C., Cleuziou, G., Clavier, V. (2006). *Catégorisation de textes en domaines et genres*. Document numérique, Vol. 9 2006/1, 61-76.

Ramond, A. (2016). *Intérêt de l'enrichissement sémantique pour une tâche de catégorisation de textes courts par méthode hybride avec peu de données d'entraînement*. Grenoble, France.

Réhel, S. (2005). *Catégorisation automatique de textes et Cooccurrence de mots provenant de documents non étiquetés*. Mémoire, Université Laval Québec, Canada.

Toussaint, Y. (2011). *Fouille de textes : des méthodes symboliques pour la construction d'ontologies et l'annotation sémantique guidée par les connaissances.* Traitement du texte et du document. Université Henri Poincaré - Nancy I. fftel-00764162

VINOT, R., GRABAR, N., Valette, M. (2003). *Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'Internet*. Université Paris, France.

Glossaire

Pipeline	Un élément du processeur où l'exécution des instructions est divisée en plusieurs étapes.
Snippet	Un terme de programmation informatique désignant un fragment de texte.
Gazetteers	Une liste de mots avec des annotations (tags sémantiques).
Tokenregex	Des règles linguistiques exprimées avec des expressions régulières et qui agissent sur des token en effectuant des actions.
Token	Une entité lexicale.
Clustering	Appelé classification non supervisée. C'est un processus qui permet de rassembler des données similaires.
Clusters	Des regroupements de documents textuels qui ont des traits sémantiques communs.

Sigles et abréviations utilisés

IA	Intelligence Artificielle
RH	Ressources Humaines
DRH	Directeur, Directrice des Ressources Humaines
RRH	Responsable Ressources Humaines
RC	Relation Client
SaaS	Software-as-a-service
RGPD	Règlement Général de Protection des Données
TAL	Traitement Automatique des Langues
ML	Machine Learning
API	Application Programming Interface
NLP	Natural Language Processing

Table des illustrations

Liste des tableaux

Table des matières

MOTS-CLÉS : méthodes symbolique, méthode statistique, méthode hybride, catégorisation, taxonomie semi-automatique, enrichissement sémantique, apprentissage automatique, ressources humaines.

RÉSUMÉ

Dans ce travail, nous avons cherché à présenter une application d'un système de classification automatique de textes. Pour cela, nous avons développé un produit pour une entreprise à des fins applicatives pour un client : un classifieur automatique pour des réponse d'enquêtes de ressources humaines, fondé sur une méthode hybride (apprentissage automatique sur des données annotées, couplé avec des règles expertes et des lexiques). La première approche est basée sur des techniques statistiques. Elle consiste à effectuer une annotation multi-catégoriel permettant d'entraîner le modèle afin de classer le texte. La deuxième approche est symbolique. Une taxonomie du domaine a été élaborée dans un but d'annotation lexicale sémantique des textes. Par la suite, nous avons développé un système de classification hybride. Nous avons présenté à la fin les résultats des mesures d'évaluation pour déterminer les performances du classifieur sur nos corpus.

KEYWORDS : symbolic, statistical and hybrid methods, categorization, semi-automatic taxonomy, semantic enrichment, machine learning, human resources.

ABSTRACT

In this work, we present an application of an automatic text classification system. In order to do so, we developed a product for a company that will be applied to a customer case : an automatic classifier for human resources survey responses, based on a hybrid system (machine learning on annotated data, coupled with expert rules and lexicons). The first approach is based on statistical techniques. It consists on performing multi-categorical annotation to train the model in order to classify the text. The second approach is symbolic. A taxonomy of the humain ressources domain has been developed for the establishment of semantic annotation on texts. We, then, developed a hybrid classification system. At the end, we present the results of evaluation measures to determine the performance of the classifier on our corpus.