Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

MSc HES-SO en Business Administration

Orientation :
Management des Systèmes d'information

Classification automatique de contenus informationnels dans le domaine de la santé

Réalisé par

Maxime Beck

Sous la direction de
Prof. Henning Müller

Lausanne, Août 2019

Table des illustrations

Table des tableaux

Table des codes sources

Remerciements

Je tiens à adresser mes remerciements à toutes les personnes qui m'ont aidé et soutenu durant la réalisation de ce travail de Master. Je remercie ainsi tout particulièrement :

Prof. Dr. Henning Müller

Pour les conseils et avis critiques portés sur mon travail qui m'ont aidé à poursuivre cette étude dans le droit chemin. Je le remercie également de sa disponibilité durant toute la durée de réalisation de cette thèse.

M. Dini Grégoire

Pour sa disponibilité ainsi que son aide lié à la récolte de données.

L'expert

Pour prendre le temps de lire et d'analyse mon travail.

Mme Rosa Ammirata

Pour son soutien appuyé durant toutes mes études.

Mme Liliane Beck, M. Ludovic Beck ainsi que ma famille proche

Pour leurs encouragements.

Jean-Frédéric Clere et l'équipe de Red Hat Inc. Suisse

Pour leurs modularités, leurs bons conseils et leurs soutiens tout au long de ce travail.

Abréviations

API	Application Programming Interface
BLOB	Binary Large Object
BOW	Bag-of-Word
CT	Classification Textuelle
CTM	Classification Textuelle Multi-labelles
HTTP	HyperText Transfer Protocol
JSON	Format de représentation de données textuelle
LSI	Latent Semantic Indexing
MAA	Méthode d'Adaptation d'Algorithme
ML	Machine Learning
MTP	Méthode de Transformation de Problème
REST	REpresentational State Transfer
SGBD	Système de Gestion de base de données
TALN	Traitement Automatique du Langage Naturel
WSGI	Web Server Gateway Interface

Résumé

Dans le cadre de sa nouvelle stratégie, l'Hôpital du Valais a initié une refonte complète de son intranet. Parmi les nouvelles fonctionnalités proposées par cette refonte, un concept de “ciblage d'audience” y sera instauré afin d’offrir aux utilisateurs du contenu pertinent en fonction de leurs profils. Sa mise en œuvre repose sur l’indexation préalable des documents qui compose l’intranet actuel. Dans le cadre de cette thèse de Master, nous nous intéresserons à l’utilisation des principes d’intelligence artificielle dans la prédiction des labels relatifs à l’indexation de ces documents.

Pour ce faire, nous avons commencé par réaliser une étude qualitative sous forme d’interviews dans un contexte de récolte des besoins. S’en est suivie une revue de la littérature quant à la classification textuelle multilabélisée. Nous avons extrait de celle-ci, un processus standard de classification textuelle pour lequel les meilleures approches furent étudiées pour chaque étape. Nous en avons alors conclu qu’une analyse sémantique basée sur le principe de cooccurrence était la meilleure approche compte tenu de la faible quantité de documents prélabélisés à disposition. Pour ce faire, les modèles de prédictions d’indexation sémantique latente et Word2Vec ont été retenus. Notre évaluation de ces deux modèles démontre un *Hamming Loss* trois fois supérieur sur l’indexation sémantique latente (60,8% contre 18,3% pour Word2Vec). Le niveau de précision est cependant similaire pour les deux modèles (respectivement 14,3% et 16,2%). Nous avons finalement entrepris le développement d’un prototype fonctionnel basé sur la cooccurrence sémantique en utilisant le modèle Word2Vec.

Mots clés : Classification multilabels, Word2Vec, Indexation sémantique latent

1 Introduction

Cette étude est réalisée dans le cadre d'une thèse de Master en filière « Management des systèmes d'information » à la Haute École Spécialisé de Suisse Occidentale. Elle porte sur un travail de recherche mandaté par l'Hôpital du Valais et débouche sur l'obtention d'un diplôme de Master of Science in Business Administration (MScBA).

1.1 Motivations

Dans le cadre de sa nouvelle stratégie, l'Hôpital du Valais a initié une refonte complète de son intranet. Ce nouvel intranet sera son outil central de communication interne utilisé en deux langues, le français et l'allemand. Celui-ci sera en fonction sur tous ses sites géographiques, à savoir : Sion, Sierre, Martigny, Monthey, Saint-Maurice, Montana, Brigue et Viège. Parmi les nouvelles fonctionnalités proposées par cette refonte, un concept de ciblage d'audience y sera instauré afin de pouvoir offrir les informations les plus pertinentes à chaque sous-population. Le nouvel intranet accueillera ainsi les utilisateurs finaux grâce à un tableau de bord leur présentant des posts et documents susceptibles de les intéresser en fonction de dimensions tels que leur langue, leur site géographique de travail, leurs groupes métiers ou encore leur niveau hiérarchique. Ces dimensions seront extraites d'une taxonomie mise en place par le comité exécutif de l'Hôpital du Valais. Actuellement, ce projet est encore en phase conceptuelle. Sa mise en œuvre repose sur l'indexation préalable des documents et posts de l'intranet courant. Cependant, comme ceux-ci se comptent par millions, un processus manuel pour réaliser cette tâche n'est pas viable.

Cette étude s'intéresse ainsi aux possibilités d'automatiser l'indexation de ceux-ci en s'appuyant sur des concepts de traitement automatique du langage naturel (TALN).

1.2 Objectifs

Les objectifs principaux définissant le périmètre de cette étude sont les suivants :

1 Entreprendre une étude qualitative de récolte des besoins auprès des parties prenantes du projet ;

2 Revoir la littérature des solutions de traitement automatique du langage naturel dans le contexte de la classification de documents textuels ;

3 En se basant sur une taxonomie fournie par l'Hôpital du Valais, implémenter puis évaluer une série de solutions de classification ;

4 Sélectionner la solution d'implémentation la plus appropriée et en développer un prototype fournissant une API pour faciliter son accès par les applications de l'Hôpital du Valais ;

5 Proposer une méthode de déploiement de la solution.

Ces objectifs, au caractère générique, ne constituent pas une liste exhaustive du travail à réaliser. Ils fournissent cependant un cadre de travail et tracent les grandes étapes de réalisation. Ils ont été définis en collaboration avec le mandant et validés par le responsable de thèse. Dans le cadre de ce projet, aucune contrainte particulière n'a été établie quant aux outils et aux technologies à utiliser.

1.3 Questions de recherche et hypothèses

Dans le but de répondre à notre problématique, cette présente étude se repose sur la question de recherche suivante :

"Quels sont les facteurs impactant les labels liés aux documents constituant l'intranet de l'Hôpital du Valais ?"

Elle découle de la volonté de déterminer toutes manières possibles de résoudre la problématique technique de labélisation automatique. Ceci dans le but d'en déduire la meilleure approche. Nous exprimons ainsi, d'ores et déjà quatre hypothèses pouvant contribuer à sa résolution :

- *Hypothèse H_1* : Le contenu des documents a un impact significatif sur les labels ;
- *Hypothèse H_2* : Le nom des fichiers associé aux documents a un impact significatif sur les labels ;
- *Hypothèse H_3* : Les métadonnées des documents ont un impact significatif sur les labels;
- *Hypothèse H_4* : Le profil des utilisateurs publiant les documents à un impact significatif sur les labels.

Ces hypothèses ont été identifiées suite à une réflexion sur les données intrinsèques et extrinsèques des documents. En effet, nous entendons par données intrinsèques, toutes données propres à eux-mêmes. Ainsi, leurs contenus, leurs noms et leurs métadonnées (taille, nombres de mots, etc.) sont de potentiels facteurs impactant leur labélisation. Les données extrinsèques sont quant à eux les données reliées aux documents par un lien associatif. Dans notre contexte, le profil de l'utilisateur publiant le document dans l'intranet pourrait être une source de prédiction. Ce profil existant au sein du système interne est associé aux documents lors de la publication pour des raisons organisationnelles et structurelles. Nous pensons qu'il puisse contribuer à la prédiction des labels, car l'Hôpital du Valais est un établissement pluridisciplinaire avec de multiples domaines et sous domaines métiers. Selon nous, il est ainsi probable que les employés spécialisés publient majoritairement des documents en lien avec leurs domaines métiers.

1.4 Méthodologie de travail

Ce chapitre commence par présenter la méthodologie utilisée quant à l'acquisition des données puis à la population des études qualitatives et quantitatives entreprises pour satisfaire nos hypothèses. Puis, la fin de chapitre s'intéresse à la méthodologie de gestion de projet sur laquelle se base notre étude.

1.4.1 Acquisition des données

Une des difficultés auxquelles nous sommes confrontés dans la réalisation de cette étude est le manque de données prélabélisées à disposition. En effet, l'état de l'intranet au moment de l'étude est complètement dépourvu de toute classification. Ainsi, pour les besoins de l'étude, la mise en place d'un corpus labélisé est nécessaire. Cette opération aurait idéalement nécessité l'implication d'un collaborateur métier pour s'assurer de la justesse de cette classification. Cependant, celle-ci étant impossible dans le cadre de notre collaboration avec l'Hôpital du Valais, c'est le mandant du projet qui s'est chargé de l'opération. La qualité absolue de la classification est donc incertaine, mais le corpus labélisé nous permettra tout de même de débuter la phase d'analyse des données. Porter une étude sur un corpus si petit a pour risque que l'échantillon de documents résultant ne soit pas représentatif de l'ensemble des documents de l'intranet. Ce risque, inévitable compte tenu des circonstances, est à prendre en considération tout au long de l'étude. Dans le but de le limiter, nous travaillerons également sur un second corpus plus conséquent, mais non labélisé. L'analyse des deux de manière conjointe nous offre un certain contrôle sur le risque. Dans le cadre de la validation de nos hypothèses H_1, H_2 et H_3 c'est ainsi ces sources de données primaires qui seront utilisées. En revanche, en ce qui concerne la validation de notre hypothèse H_4, nous planifions de

réaliser une étude quantitative sous forme de sondage. Cette source de donnée primaire nous permettra de déterminer les types de documents que publient les différents profils utilisateurs. Finalement, une ultime source de donnée primaire sera utilisée dans le cadre d'interviews liés à l'élicitation des besoins. Ceux-ci nous permettront d'avoir une meilleure vue d'ensemble quant aux tenants et aboutissants du projet.

1.4.2 Population

Pour ce qui est de notre sondage, celui-ci constituera une analyse transversale sur la globalité des collaborateurs de l'Hôpital du Valais participants au partage de documents dans l'intranet. Un premier travail à l'interne doit être réalisé pour déterminer cette population. Le choix d'une population si vaste permet de réaliser une représentation des données la plus précise possible pour entraîner notre algorithme de manière optimale. De plus, les moyens de communication informatiques de l'établissement le permettent sans grandes difficultés. Le sondage sera disponible en ligne et nous tirerons profit d'une étude quantitative entreprise à l'interne par le maître d'œuvre pour le diffuser. Notre interview d'élicitation des besoins quant à lui, ciblera les membres à l'origine de l'initiative de refonte de l'intranet. Ceux-ci constituent ensemble le maître d'œuvre. Ils seront en mesure d'étendre notre compréhension de la problématique initiale de classification textuelle.

1.4.3 Gestion de projet

La méthodologie CRISP-DM est populairement utilisée dans la réalisation de projets liés à l'exploration de données. Elle fut introduite par Wirth et Hipp en 1995 [1] et gravite autour de six phases décrites ci-après (Figure 1). Nous choisissons d'utiliser cette méthodologie dans le cadre de notre étude dans le but de nous offrir une ligne directrice dans la résolution de la problématique.

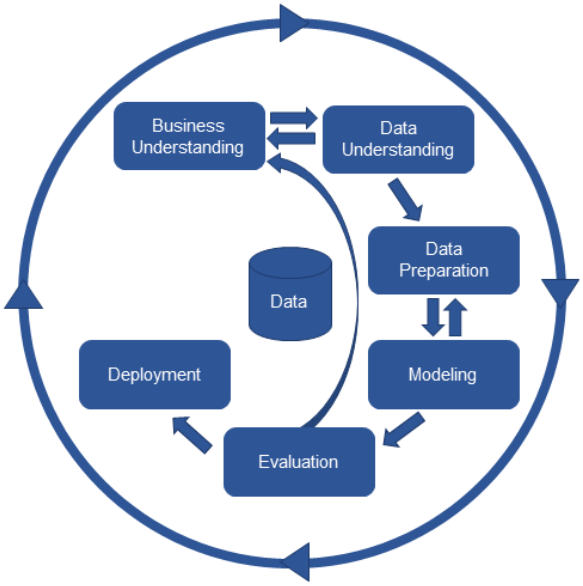


Figure 1 Processus de la méthodologie CRISP-DM

Compréhension de la problématique

Cette phase initiale, qui introduira notre étude, vise à comprendre les objectifs et les exigences du projet d'un point de vue métier. Elle établit le contexte de l'étude et y donne un but concret.

Compréhension des données

La compréhension des données commence par leurs récoltes. Dans le cadre de notre étude, celles-ci nous ont été fournies par le mandant. Elle se poursuit par leurs analyses pour se familiariser avec leurs natures statistiques et linguistiques en identifiant les potentiels problèmes de qualité. Cette phase se terminera par le choix des modèles de prédictions à implémenter.

Prétraitement des données

La phase de préparation des données couvre toutes les activités nécessaires à la construction du jeu de données final. Ainsi, nous utiliserons les outils préalablement sélectionnés pour exécuter les divers processus de nettoyage sur le contenu textuel des documents de notre corpus. Le but étant d'atteindre un niveau de qualité suffisante pour initier l'apprentissage de nos modèles de prédictions.

Modèles de prédictions

Cette phase constitue le cœur de l'apprentissage de nos modèles de prédictions. Nous y étudierons les APIs que proposent nos outils et adapterons leurs configurations à notre contexte spécifique.

Évaluation

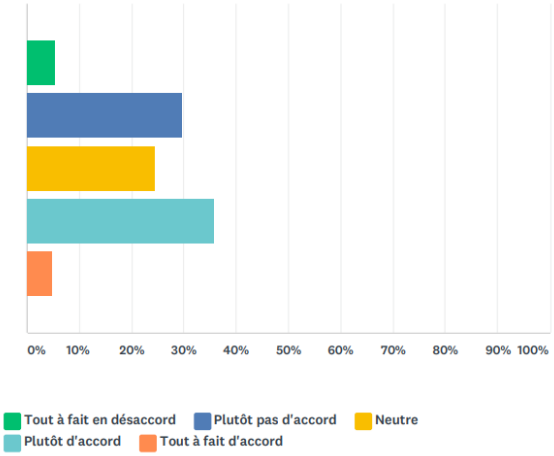
Une fois les modèles de prédictions entraîner, il est crucial de les évaluer pour mesurer leurs performances respectives. Dans cette phase, nous présenterons notre méthodologie d'évaluation ainsi que nos résultats.

Déploiement

La phase finale d'un projet d'exploration de données consiste au déploiement du modèle de prédiction retenue comme le plus performant pour résoudre la problématique en vigueur. Dans le cadre de notre étude, nous développerons au sein de cette phase un prototype fonctionnel de la solution proposée.

1.5 Compréhension de la problématique

Dans le but de concevoir une vue globale du projet et de ses besoins, nous avions pour objectifs de réaliser une série d'interviews qualitatifs avec les parties prenantes du processus de refonte de l'intranet de l'Hôpital du Valais. Malheureusement, par manque de disponibilités de leurs côtés, il fut impossible de s'entretenir avec eux dans le cadre de cette thèse. Le mandant, Monsieur Dini Grégoire, a cependant accepté de répondre à nos questions (cf. Annexe 1 – Guide d'entretien de l'étude qualitative de récolte des besoins). Engagé depuis septembre 2018, il dirige le projet de refonte à l'interne. Ce projet a pour but d'améliorer de manière significative la plateforme actuelle de partage d'informations. D'après un sondage comptant 422 réponses, réalisé dans le cadre de la refonte, 35% des collaborateurs seraient insatisfait des performances de l'intranet quant à la recherche d'information (Question 1).



Question 1 Considérez-vous que le temps nécessaire à trouver une information dans l'Intranet est acceptable ?

Les commentaires des participants font souvent référence au manque d'organisation au sein de sa structure de données. Ainsi, dans un premier temps, l'initiative de notre projet de classification automatique part de la volonté d'améliorer l'organisation des données. Dans un second temps, celle-ci permettra de fournir une interface dynamique aux utilisateurs en leur proposant de l'information pertinente en fonction de leurs profils. Actuellement, le projet de refonte en est encore à ces débuts. Initié en septembre 2018, il est actuellement en phase de récolte des besoins. Le développement de la solution sera ensuite externalisé. La mise en production du nouvel intranet est prévue pour mi-2020. Les parties prenantes au projet sont composées d'un groupe de travail d'une douzaine de personnes, ces personnes représentent les utilisateurs finaux et participent au projet dans le cadre d'interviews et de tests fonctionnels. La direction générale est également partie prenante, c'est son comité de pilotage stratégique qui a initié le projet de refonte. Certains collaborateurs externes interviennent de manière régulière dans le cadre, entre autres, de la conduite d'études qualitatives et quantitatives. Monsieur Dini endosse quant à lui, le rôle de chef de projet. Au sein de ce projet, notre étude quant à la classification textuelle est prévue d'y être intégrée lors de la phase de réalisation. Les spécifications fonctionnelles et technologiques liées à cette phase ne sont cependant pas encore définies. Ainsi, le mandant requiert que la solution de notre étude soit facilement intégrable à tout système et présente une rapidité d'exécution acceptable. Une interface REST est mentionnée. Notre travail se doit également d'être basé sur une taxonomie établie à l'interne (cf. Annexe 2 – Taxonomie de l'intranet). Cette taxonomie, multidimensionnelle, est évolutive dans le temps. Il est donc idéalement nécessaire que la solution gère cette évolutivité ainsi que la multidimensionnalité de celle-ci. L'Hôpital du valais, étant divisé entre des parties francophones et germanophones du canton Valaisant, il faudrait donc également tenir compte du caractère multilingual des documents de leur intranet. Pour le projet, aucun dégrée

d'importance n'a été mis en place sur les dimensions de la taxonomie. Cependant, le mandant propose de débuter par la prédiction des dimensions « Type de documents » et « Géographie ».

2 Revue de littérature

La *fouille de texte* (de l'angl. *Text Mining*) se réfère au processus d'extraction de connaissances ou de modèles à partir de documents textuels non structurés [2]. Il est considéré comme une extension de l'*exploration de données* (de l'angl. *Data Mining*). Le domaine du Machine Learning fait la distinction entre les problèmes de classifications monolabel, pour lesquels une prédiction peut être attribuée à une seule classe et multilabel, où celles-ci peuvent potentiellement appartenir à plusieurs classes [3]. [4] décrit le processus de classification textuel en cinq étapes chronologiques : la représentation des documents, la sélection des caractéristiques, la construction d'un modèle d'espace vectoriel, l'application d'un algorithme d'exploration de données et l'évaluation du classificateur (Figure 2).

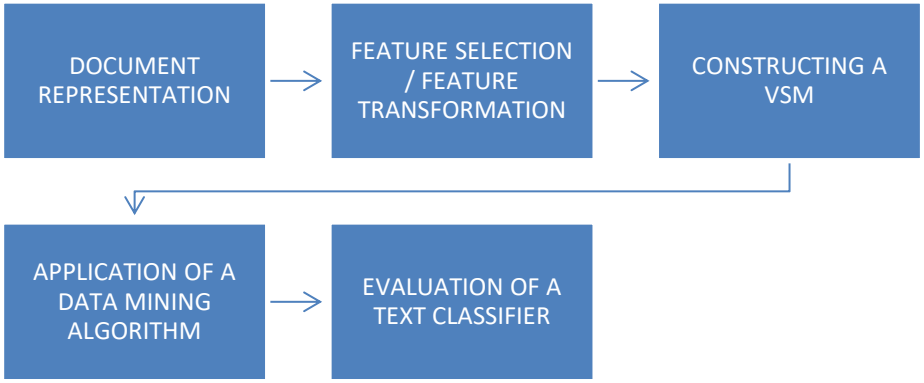


Figure 2 Processus de classification textuelle

Les sous-chapitres qui suivent présentent une revue de littérature des pratiques utilisées au sein de chacune de ces étapes.

2.1 Représentation des documents

La première phase du processus de classification textuel des données consiste à représenter les documents sous une forme qui convient à la fouille de texte, c'est-à-dire sous la forme d'instances constituant un nombre fixe d'attributs [4], [5]. Pour ce faire, le concept de tokenisation est la technique la plus rependue [4], [6]. Elle consiste en la décomposition du texte en phrases puis en mots en supprimant la ponctuation et en convertissant les lettres majuscules en minuscule. Le texte résultant est ensuite converti en *bag-of-words* de l'anglais *sac de mots*. Avec cette représentation, toutes les instances de mots parues au moins une fois au sein des documents sont recensées. Ceci peut cependant très vite générer de gros volumes de données complexifiant leurs exploitations. C'est la raison pour laquelle il est indispensable de réaliser quelques manipulations en amont de cette transformation dans le but de limiter le volume de données résultant. Une pratique commune dans la littérature est la suppression des mots qui n'apportent aucune pertinence à la compréhension du texte. Dans le domaine du TALN, ces mots portent le nom de *mots vides* [7] (*stop words* en anglais). Les déterminants et pronoms constituent ce type de mot. La racinisation et la lemmatisation font également partie des pratiques communes [8]. Celles-ci ont pour objectif de ramener toutes les variantes linguistiques (flexions) d'un mot à une forme générique. Leur procédé ainsi que leur résultat varient : la racinisation cherche à supprimer une série de préfixes, suffixes, postfixes ou antéfixes pour déterminer la racine (ou radial) d'un mot.

Ainsi, la racine des mots *recherchera*, *chercherons* et *cherchant* est *cherch*. Car *cherch* est une chaîne de caractère commun à toutes les variantes de cette même famille. La lemmatisation quant à elle, cherche à déterminer le lemme d'un mot, soit sa forme non conjuguée et non accordée. À titre d'exemple, les mots précédemment cités ont ainsi deux lemmes : *recherchera* devient l'infinitif *rechercher* et les deux flexions *chercherons* et *cherchant* se transforment en l'infinitif *chercher*. Comme nous pouvons le constater, contrairement à la racinisation, les mots résultant d'une lemmatisation ne perdent ainsi pas leurs significations d'origine. Cependant, cela engendre également un volume de données plus important que celui d'une racinisation. Le choix de l'une ou l'autre de ces pratiques dépend ainsi du contexte et de la problématique à résoudre.

2.2 Sélection des caractéristiques

La sélection des caractéristiques permet d'aller encore plus loin dans la réduction du volume de données correspondant à chaque document. La littérature parle de réduire la *dimensionnalité du set de caractéristiques* [9]. Dans notre contexte, le set de caractéristiques constitue tous les mots de nos documents (*sac de mots*) par labels et le terme *dimensionnalité* fait référence aux dimensions de l'espace vectoriel qui représentera ces mots au terme de nos transformations. Réduire la dimensionnalité est essentiel pour améliorer la qualité de la classification en réduisant les risques de surajustement lors de la phase d'apprentissage [10]. Le surajustement est une erreur de modélisation qui se produit lorsqu'une fonction est trop étroitement ajustée à un ensemble limité de points de données [11]. Cela a pour effet que le modèle résultant n'est pas assez générique pour classifier des documents dont il n'a pas précédemment appris. La sélection des caractéristiques se réalise en attribuant un score à chacun des mots des documents dans le but de ne garder que ceux dont le score est le meilleur [12]. Cela permet de retirer les mots considérés comme peu pertinents pour représenter les labels relatifs aux documents. Plusieurs méthodes d'attribution de score existent, la liste suivante expose les plus communément utilisés[4].

CHI-square (CHI)

Mesure le degré d'indépendance entre les mots et les labels.

Information Gain (IG)

Également nommé *information mutuelle attendue* [13], le gain d'information mesure le nombre de bits d'information obtenus par labels en fonction de la présence ou l'absence de mots au sein des documents.

Mutual Information (MI)

Mesure le degré de dépendance entre mots et les labels.

Analyse sémantique latente (LSA)

Établi des relations entre un ensemble de documents et leurs mots pour en ressortir des concepts. Pour se faire, l'analyse sémantique latente recense plusieurs algorithmes tels que les voisins proches (de l'angl. *Near neighbors*) ainsi que différentes comparaisons (matricielles, par paires, par phrases, etc.).

Document Frequency (DF)

Représente le nombre de documents dans lequel un mot apparaît.

Term Frequency (TF)

Représente la fréquence d'apparition d'un mot au sein d'un document.

Term Strength (TS)

Estime l'importance des mots basée sur leur probabilité d'apparition au sein de documents « étroitement liés ».

Odds Ratio (OR)

Exprime le ratio entre les probabilités qu'un mot soit associé à un label et celles dont il n'y fasse pas partie.

Selon l'étude comparative de Yang et Pedersen [14], toutes ces méthodes sont viables pour la sélection des caractéristiques. Cependant, celle-ci retient les mesures *Information Gain* (IG), *CHI-square* (CHI) et *Document Frequency* (DF) comme étant les plus performants dans leur expérience. L'étude a en effet évalué les cinq algorithmes *Document Frequency* (DF), *Information Gain* (IG), *Mutual Information* (MI), *CHI-square* (CHI) et *Term Strength* (TS) sur le corpus Reuters[1]. Leurs résultats démontrent que IG est parvenu à retirer jusqu'à 98% des mots uniques aboutissant à une amélioration significative de la précision de leur classificateur (k-NN). Ils ont également déterminé de fortes corrélations entre les scores des mesures IG, CHI et DF. Ceci suggère que DF, la méthode dont le coût de traitement est le plus faible, peut être utilisée au lieu de IG ou CHI lorsque le coût de traitement de ces méthodes est trop important.

Au-delà de la sélection des caractéristiques, des techniques d'extraction des caractéristiques permettent également la réduction de la dimensionnalité. Contrairement à la sélection des caractéristiques, cette approche n'attribue pas de score à des caractéristiques existantes, mais transforme celle-ci pour en générer de nouvelles tout réduisant leurs dimensionnalités [15]. L'analyse en composantes principales conventionnelle (ACP) est l'une des techniques de transformation des caractéristiques les plus couramment utilisées. Son rôle est de convertir un set d'observations dont les variables sont corrélées en un set dépourvu de corrélations intervariables [16]. Ceci dans le but d'atteindre une fonction linéaire non corrélée réduisant les risques de surajustement.

2.3 Construction du modèle vectoriel

Un modèle vectoriel est une forme de représentation algébrique d'un document permettant de prendre en considération sa sémantique [17]. Ainsi, chaque mot est transformé en vecteur et placé dans un espace vectoriel à N-dimensions. Au sein de cet espace vectoriel, la position des mots est relative à leurs poids. Ainsi, une méthode de pondération est nécessaire. TF-IDF (de l'angl. *Term Frequency-Inverse Document Frequency*) a été rapportée comme étant la plus efficiente [4]. Celle-ci calcule un indice pour chaque mot d'un document selon la proportion inverse de la fréquence d'un mot au sein d'un document donné par rapport au pourcentage de documents dans lequel le mot apparaît [18]. Les mots dont le score TF-IDF est élevé indiquent ainsi avoir une forte relation avec le document dans lequel ils apparaissent. Une fois le modèle vectoriel généré, il est impératif de normaliser les données avant d'appliquer l'algorithme d'exploration de données [19].

2.4 Application d'un algorithme d'exploration de données

Cette étape est le cœur du processus de classification. Elle consiste en la sélection et la mise en place d'un algorithme d'exploration de données. Nous ciblons, dans le cadre de cette partie de notre revue de littérature, les algorithmes susceptibles de répondre à notre problématique, soit la

[1] Reuters est une agence de presse britannique mettant à disposition une large collection d'articles pour la recherche et le développement dans le domaine du TALN - https://trec.nist.gov/data/reuters/reuters.html

classification textuelle multi-labelles (CTM). Dans le contexte de la Machine Learning, ces algorithmes peuvent correspondre à trois catégories distinctes : les algorithmes supervisés, semi-supervisés et non supervisés [20]. Les algorithmes supervisés apprennent sur la base d'un set de données prélabélisé. Ainsi, dans le cadre d'une classification, les labels de sorties (résultantes de l'algorithme) sont connus. Cette approche est particulièrement coûteuse, car elle nécessite la labélisation manuelle d'un grand nombre de données. À contrario, les algorithmes non supervisés apprennent en évaluant les similitudes statistiques trouvées au sein du set de données d'entraînement. Par mesures associatives, ceux-ci déterminent les labels de sorties automatiquement en créant des groupes au sein des données (nommés *clusters*). Finalement, l'approche semi-supervisée est une combinaison des deux précédentes approches. En effet, celle-ci commence par apprendre sur la base d'un set de données prélabélisées, puis, utilise le modèle de prédiction résultant pour automatiser la labélisation d'un set de données non labélisées. La littérature nomme cette pratique la *pseudo-labélisation* [21]. Une fois celle-ci terminée, les sets de données prélabélisés et pseudo-labélisés sont tous deux utilisés pour entraîner un nouveau modèle. En termes de qualité, le modèle résultant performe généralement mieux qu'un modèle entraîné uniquement sur des données non labélisées [22]. Cette approche est particulièrement intéressante lorsqu'un large set de données prélabélisé est difficilement récoltable.

Dans la littérature, il existe deux approches pour traiter les problèmes de CTM [3] : une approche indirecte, nommée *méthode de transformation de problème* (MTP)[2] et une approche directe intitulée *méthode d'adaptation d'algorithme* (MAA)[3]. Cette première approche (MTP) suppose l'indépendance des labels et transforme le problème de CTM en plusieurs problèmes de CT monolabel. Ainsi, un classificateur indépendant est modélisé pour chaque label. À l'opposé, l'approche MAA adapte les algorithmes pour traiter directement le problème de CTM. Avec cette approche, un seul classificateur est modélisé pour tous les labels. Les méthodes de transformation de problèmes actuels furent inspirées par deux méthodes populaires : *Binary Relevance* (BR) et *Label Powerset* (LP) [23]. La méthode BR transforme la problématique en une classification binaire. Son principe consiste à diviser le set de données multilabélisé en plusieurs sous-sets. Chacun de ces sous-sets contient toutes les instances positives et négatives de chaque label [24]. Il entraîne ensuite un classificateur binaire par sous-sets puis fusionne les prédictions de chacun d'entre eux pour déterminer la prédiction finale. BR a donc une complexité linéaire respectivement au nombre total de labels qui limite son utilisation lorsque le nombre de labels à traiter est élevé [25]. La méthode LP, quant à elle, transforme la problématique en une classification multilabels. Chaque label de sorties étant une combinaison possible de labels. Ainsi pour les labels A et B, LP les représente sous forme de quatre labels : [0, 0], [0, 1], [1, 0], [1, 1], créant un seul classificateur binaire pour toutes les combinaisons de labels [26].

Pour ce qui sont des méthodes d'adaptations d'algorithmes, Zhang et Zhou [27] proposent une approche nommée *ML-kNN* dérivée de la méthode des *k plus proches voisins* (abrégé *k-NN* de l'anglais *k-Nearest Neighbors*). Dans un contexte de classification, cette méthode a pour entrer les données d'apprentissages au sein de l'espace vectoriel le plus proche de la donnée à classifier et comme sortie le label le plus pertinent parmi ces données [28]. Cette méthode est basée sur la notion de *Lazy Learning*. Cette notion décrit une méthode d'apprentissage dans laquelle la généralisation des données d'apprentissage est retardée jusqu'à ce qu'une requête soit adressée au système [29]. Celle-ci s'oppose à *Eager Learning*, où le système tente de généraliser les données d'apprentissage avant de recevoir des requêtes. La méthode *ML-kNN* se distingue du traditionnel *k-NN* en évaluant, puis proposant plusieurs labels de sorties à la fois. D'où son préfixe *ML* (pour *multilabels*). Basé sur les résultats de l'étude de Yang et Pedersen relative aux méthodes de sélection des caractéristiques mentionnée précédemment [14], Zhang et al. conceptualisent en 2014 un algorithme visant à tirer profit simultanément des trois méthodes *Document Frequency* (DF), *Information Gain* (IG) et *CHI-Square* (CHI) [30]. Ils utilisent pour ce faire l'algorithme ML-kNN

[2] De l'anglais *Problem Transformation Method* (PTM).

[3] De l'anglais *Algorithm Adaptation Method* (AAM).

qu'ils renomment En-MLKNN (*Ensemble Multi-labels k-NN*). Leur étude démontre que l'utilisation simultanée des trois méthodes de sélection performe mieux que leur utilisation individuelle respective. Tout comme ML-kNN, l'approche Boostexter proposée par Schapire et Singer [31] exploitent les corrélations interlabels pour en généraliser leurs prédictions. Cette approche, basé sur la méthode d'apprentissage AdaBoost [32] se distingue de ML-kNN en conservant un ensemble de pondérations se référant à l'association des documents d'apprentissage et de leurs labels respectifs. Dans leur étude, Schapire et Singer ont cependant observés que l'algorithme peut potentiellement souffrir de surajustement lorsque la taille du set de données de tests est petite (< 1000 documents) [33]. Il en conclut l'importance du contrôle du système d'apprentissage dans son ensemble. Une approche basée sur l'algorithme classique SVM (*Support Vector Machine*) nommée RankSVM [34] cherche à corriger ces limitations en tenant de ce facteur. Pour se faire, l'algorithme optimise un set de classificateurs linéaires (SVM) dans le but de minimiser la perte de classement (de l'angl. *ranking loss*[4]) inhérente au système. Deux cas d'études, respectivement dans le contexte de la classification de textes et d'images, comparent les performances des algorithmes ML-kNN [33], BoosTexter [31], ADTBoost [35] et Rank-SVM [34]. Leurs résultats démontrent des performances supérieures pour l'algorithme ML-kNN sur les deux cas d'études [33].

Le défi fouille de texte (DEFT) est une campagne d'évaluation scientifique francophone portant sur le domaine du traitement automatique du langage naturel (TALN)[5]. Ainsi, chaque année, plusieurs équipes de recherches de la communauté scientifique participent à la résolution d'une problématique dans le but de faire avancer les recherches dans le domaine. Dans ce contexte, une étude de 2016 traite de l'indexation de documents scientifiques francophones multilabels [36]. Une de leurs approches est fondée sur la cooccurrence entre un concept (label) et les termes du document à labéliser. Ainsi, si dans un corpus de textes labélisés, un ou plusieurs termes cooccurrent souvent avec le même concept *c*, un document non labélisé contenant cette même combinaison de termes se voit attribué le concept *c*. Pour éviter que les cooccurrences soient dues au pur hasard, ils ne prennent en compte uniquement les concepts servis pour labéliser au moins cinq documents. L'étude utilise l'analyse sémantique latente (LSA) pour déterminer les cooccurrences à partir de leur corpus labélisé.

Dans la littérature, LSA est souvent mis en comparaison avec Word2Vec [37]–[40]. Ces deux modèles permettent l'extraction de cooccurrences sémantiques (de l'angl. *Word Embedding*) et sont tout deux les plus performants dans leurs approches respectives. Le premier (LSA) est un modèle basé sur le comptage (de l'angl. C*ounter-based*) [41]. Celui-ci commence par créer une matrice des termes des documents. Les enregistrements de cette matrice sont ensuite représentés sous forme de vecteurs dans un espace Euclidean. Finalement, une décomposition en valeurs singulières est réalisée dans le but de diminuer la dimensionnalité de celui-ci. Les distances entre vecteurs peuvent alors être calculées. Deux vecteurs proches signifient une grande similitude sémantique. Le modèle Word2Vec quant à lui, est un modèle prédictif. Il consiste en deux modèles de réseau neuronal : Continuous Bag-of-Words (CBOW) et Skip-gram [42]. Dans les deux modèles, une "fenêtre" de taille prédéfinie est déplacée tout le long du corpus. Pour chaque étape, le réseau est entraîné avec les mots présents en leur sein. La différence entre ces modèles est que CBOW est entraîné à prédire le mot au centre de la fenêtre basé sur les mots qui l'entourent alors que Skip-gram entraîné à prédire les mots entourant le mot central basé sur celui-ci [43]. La littérature était initialement divisée quant à la solution la plus performante entre les approches basées sur le comptage et celles prédictives [38]–[40]. Un point d'entente semble privilégier les solutions de comptage lors de l'utilisation d'un corpus de petite taille et les solutions prédictives pour les corpus de grandes tailles [37].

[4] Cette métrique est expliquée plus en détail dans le sous-chapitre 2.5

[5] Défi Fouille de Textes (DEFT) - https://deft.limsi.fr/

2.5 Évaluation du classificateur

La dernière étape du processus de classification textuel est l'évaluation du classificateur sélectionné. En contraste avec l'évaluation de classificateurs monolabel, évaluer un classificateur multilabels est plus complexe, car chaque instance de document peut être associée à plusieurs labels simultanément. Ainsi, distinguer le pire entre une instance associée à trois mauvais labels ou trois instances associées à un mauvais label est parfois difficile. Pour cela il existe différentes mesures de performances qui s'intéressent à différents aspects de l'évaluation.

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

	$\mathbf{y}^{(i)}$	$\mathbf{h}(\tilde{\mathbf{x}}^{(i)})$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.1 0.2]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]

Figure 3 Tableau d'indicateurs binaires de labels effectifs avec des prédictions binaires (à gauche) et probabilistes (à droite)

Pour commencer, il est nécessaire de représenter les labels effectifs $y^{(i)}$ de chaque document sous la forme d'un tableau de valeurs binaires (Figure 3). La valeur 1 indique qu'il y a une association du document au label. Le nombre d'entrés au sein du tableau correspond donc au nombre de labels possible [44]. Similairement, un deuxième tableau $\hat{y}^{(i)}$ correspondant aux labels prédits est également créé. Celui-ci peut être binaire ou probabiliste. Différentes mesures sont applicables pour l'une ou l'autre de ces formes. [45] fait la distinction entre les mesures basées sur les labels et celles basées sur les exemples. Les mesures basées sur les labels décomposent le processus d'évaluation en réalisant une évaluation distincte pour chaque label. À contrario, les mesures basées sur les exemples évaluent les performances en calculant la différence moyenne entre le set de labels effectif et celui prédit sur l'intégralité des exemples disponibles dans le set de données soumis à l'évaluation.

2.5.1 Mesures basées sur les exemples

Plusieurs mesures d'évaluation binaires ont été adaptées pour évaluer les performances de classificateurs multilabel.

$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Z_i\rvert}$	$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Y_i\rvert}$	$\frac{1}{m}\sum_{i=1}^{m}\frac{2\lvert Y_i \cap Z_i\rvert}{\lvert Z_i\rvert + \lvert Y_i\rvert}$
Précision	Rappel	Score F_1
$\frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap Z_i\rvert}{\lvert Y_i\rvert}$	$\frac{1}{m}\sum_{i=1}^{m}\frac{2\lvert Y_i \cap Z_i\rvert}{\lvert Z_i\rvert + \lvert Y_i\rvert}$	
Hamming Loss	*Exactitude*	

Tableau 1 Équations relatives aux mesures basées sur les exemples

Au sein des équations du Tableau 1 la précision désigne la proportion de labels correctement prédite parmi tous les labels prédits et le rappel est la proportion de labels correctement prédits parmi les labels qui auraient dû être prédits. Basé sur ces notions, le score F_1 se définit comme la moyenne harmonique pondérée de la précision et du rappel. Il traduit l'équilibre entre ces deux mesures. L'exactitude correspond quant à elle à la proportion de labels correctement attribués parmi la totalité des labels. Finalement, le *Hamming Loss* est la mesure d'évaluation la plus courante dans le contexte de classification multilabels [46], il se distingue de l'exactitude en mesurant la proportion de labels prédite de manière erronée parmi la totalité des labels. Ainsi, la performance évaluée est considérée parfaite lorsque le *Hamming Loss* équivaut à 0.

2.5.2 Mesures basées sur les labels

Toutes mesures connues pour l'évaluation binaire peuvent être utilisées dans leurs formes usuelles (monolabel), telles que l'exactitude, la précision et le rappel. Le calcul de ces mesures pour tous les labels peut être réalisé en utilisant deux opérations de moyennage, appelées macro-moyennage et micromoyennage [47].

$\frac{1}{q}\sum_{\lambda=1}^{q} B(tp_\lambda, fp_\lambda, tn_\lambda, fn_\lambda)$	$B\left(\sum_{\lambda=1}^{q} tp_\lambda, \sum_{\lambda=1}^{q} fp_\lambda, \sum_{\lambda=1}^{q} tn_\lambda, \sum_{\lambda=1}^{q} fn_\lambda\right)$
B_{macro}	B_{micro}

Tableau 2 Équations de mesures basées sur les labels

En considérant les équations du Tableau 2, $B(tp, tn, fp, fn)$ est une évaluation binaire calculée sur la base du nombre de vrais positifs (tp), faux positifs (tn), vrais négatifs (fp) et faux négatifs (fn). Les variables $tp\lambda,\ fp\lambda,\ tn\lambda$ et $fn\lambda$ se réfère à ces valeurs après l'évaluation du label λ. Ainsi, un macro-moyennage (B_{macro}) calcule la métrique indépendamment pour chaque label, puis évalue la moyenne traitant tous les labels de la même manière. Le micromoyennage (B_{micro}) quant à lui commence par agréger la métrique des variables de tous les labels, puis en calcul la moyenne. Dans le contexte d'une classification multilabels, le micromoyennage est donc préférable dans le cas où les classes présentent un déséquilibre.

3 Méthodologies

Basé sur notre revue de littérature, ce chapitre présente la méthodologie de résolution de notre problématique. Pour ce faire, nous commençons par réaliser une analyse complète des données à disposition pour en approfondir notre compréhension. Nous sélectionnons ensuite la méthodologie qui nous semble la plus adéquate puis les outils nécessaires pour sa mise en place.

3.1 Compréhension des données

Dans le cadre de cette étude, L'Hôpital du Valais nous a fourni un corpus comportant 1607 documents extraits de leurs intranets actuels. Sur ces documents, 122 ont été manuellement classifiés par le mandant du projet. Les classes assignées à ces documents sont celles de la dimension "Type de document". Elles se comptent au nombre de 12 et labélisent le contenu de 4 à 14 documents chacune. Les caractéristiques importantes de ces sets de données sont leurs multilinguismes ainsi que l'hétérogénéité de leurs formats.

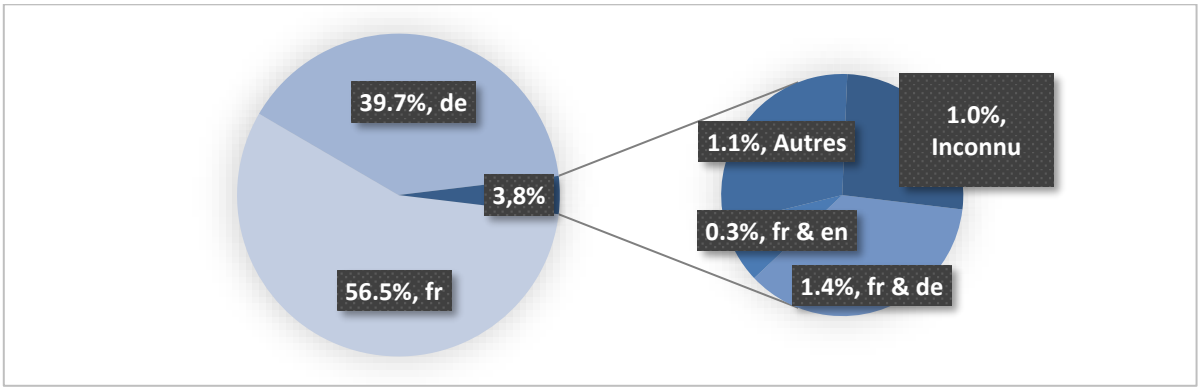


Figure 4 Répartition des langues

En effet, nous constatons dans la Figure 4 ci-dessus une répartition presque équitable entre les documents en français et ceux en allemand, représentant respectivement 56,5 et 39,7% du corpus. Le corpus sujet à ces statistiques est l'ensemble des données à disposition, soit 1607 documents. Au sein de ceux-ci, une minorité (3,8%) est rédigée en plusieurs langues. L'on trouve ainsi les combinaisons française et allemande (1,4%) ainsi que français et anglais (0,3%). D'autres, non répertoriés ici, présentent des combinaisons à trois langues tels que français, anglais et allemand ou encore parfois, des traces d'espagnole et d'italien (1,1%). La langue de 1% (soit 16 documents) des documents n'a pas pu être identifiée. Ceci, car ces documents ne contiennent pas de texte ou alors que le texte de ces documents n'est pas extractible pas un processus automatisé. Ce dernier cas survient lorsque le texte est encapsulé au sein d'images telles que des captures d'écrans ou des numérisations.

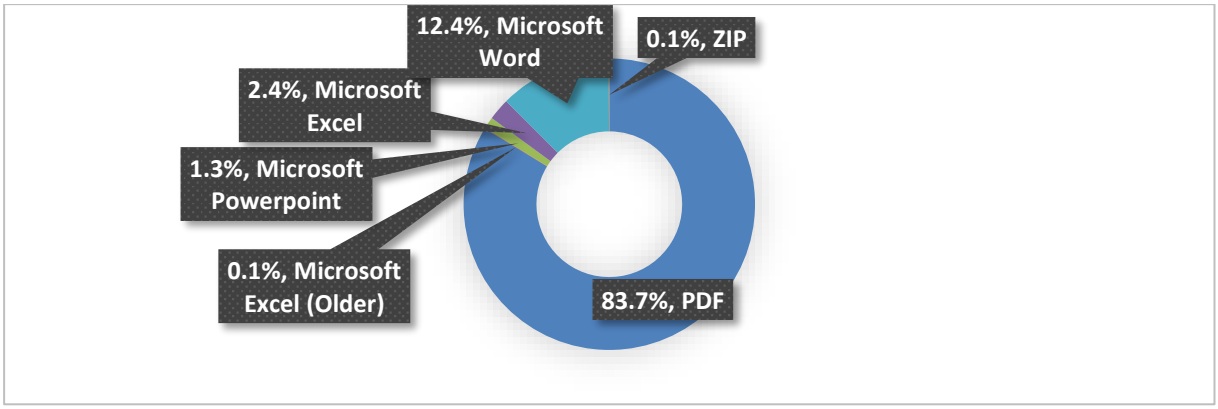


Figure 5 Répartition des formats de documents

La Figure 5 quant à elle, présente la répartition des formats de documents au sein du corpus. Le format PDF représente la majorité (83,7%). Les formats suivants sont ensuite dispersés à travers les logiciels de la suite Microsoft Office. Microsoft Word constitue une part non négligeable de documents (12,4%) suivis par Excel (2,4%) et PowerPoint (1,3%). Il est également important de tenir compte du changement de format au sein même de la suite Office. En effet, celle-ci est passée d'un format propriétaire à un format standardisé basé sur Office Open XML[6] lors de l'introduction de sa version Microsoft Office 2007. Il est très probable que certains documents disponibles dans l'intranet de l'Hôpital du Valais utilisent encore l'ancien format propriétaire. 0,1% de notre corpus est de cette nature pour le logiciel Microsoft Excel. Finalement, notre corpus présente également un pourcentage similaire de fichiers au format compressé ZIP.

Dans le but d'en apprendre plus sur les hypothèses que nous avions définies (voir, chapitre 1.3), les sous-chapitres suivent présentent une analyse des noms de fichiers (H_2), du contenu (H_1) ainsi que des métadonnées (H_3) des documents présents dans notre corpus de tests.

3.1.1 Noms de fichiers

En analysant les noms de fichiers des documents de notre corpus, nous constatons qu'il est indéniable que des conventions de nommages soient en vigueur au sein de la politique de publication de documents de l'Hôpital du Valais. Cependant, celles-ci semblent être liées aux départements, services et unités composant l'établissement et non directement aux types de documents résultants. Nous chercherons à déterminer les corrélations entre ces deux éléments lors de l'analyse des résultats de notre étude quantitative. Pour l'heure, nous nous intéressons dans ce sous-chapitre, à évaluer la fonction de recherche textuelle sur les noms de fichiers. Le principe de cette évaluation est de rechercher les mots qui composent nos labels au sein des noms de fichiers des documents à disposition afin d'en déterminer le label résultant.

Dimensions	Langues	Taille du corpus	Nb documents trouvés	Proportion de documents trouvés
Géographie	FR/DE	1607	82	5,1%
Type de documents	FR/DE	1607	81	5,0%

Tableau 3 Résultats d'une recherche textuelle simple sur les noms de fichiers

[6] Office Open XML - https://en.wikipedia.org/wiki/Office_Open_XML

Les résultats de cette évaluation (Tableau 3) montrent que le nom de fichier est une caractéristique des documents peu utilisable pour la prédiction de l'une ou l'autre de nos dimensions. En effet, respectivement 5,1 et 5,0% des documents de notre corpus contiennent des concepts de nos dimensions "Géographie" et "Type de documents". Bien que ces pourcentages soient faibles, nous pourrions hypothétiquement penser que lorsque trouvé dans le nom de fichier, cette information soit relativement fiable. Nous basons cette hypothèse sur le fait que la nature même d'un nom de fichier est la synthèse de son contenu. Ainsi, les quelques mots qui le composent ont un fort poids sur la classification définitive du document auquel il est attribué.

Notre choix d'évaluer ensemble les concepts français et allemand de notre taxonomie vient d'observations réalisées sur le corpus. Nous en avons effectivement conclu que certains noms de fichier étaient exprimés en français pour des documents au contenu allemand. Ainsi, les résultats de cette évaluation auraient été faussés si nous avions traité les deux langues séparément. Les Figures ci-dessous exposent les concepts émané des noms de fichiers pour nos dimensions « Géographie » (Figure 6) et « Type de documents » (Figure 7).

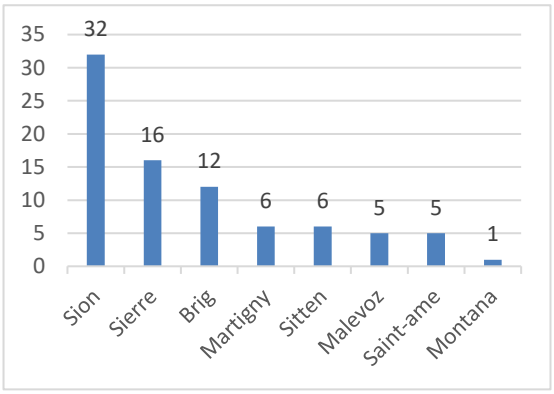


Figure 6 Occurrences par concepts sur les noms de fichiers (FR/DE) – Dimension « Géographie »

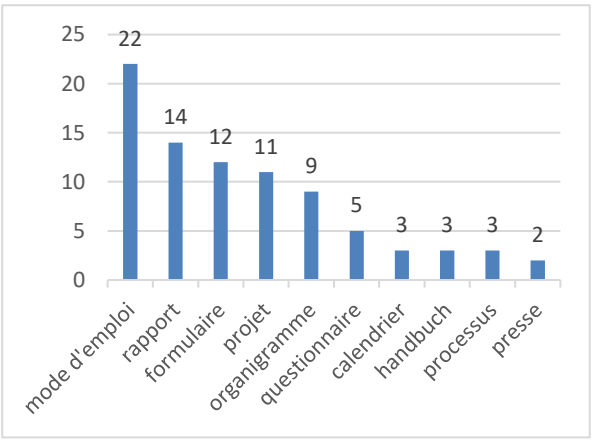


Figure 7 Occurrences par concepts sur les noms de fichiers (FR/DE) – Dimension « Type de documents »

En ce qui concerne la « Géographie », le site hospitalier le plus mentionné est Sion. Cela pourrait s'expliquer par sa taille et sa nature. Ce centre hospitalier est en effet le pilier central de l'Hôpital du Valais. Soignant environ 40% des patients somatiques aigus du canton[7], il est l'établissement employant le plus grand nombre de collaborateurs. Il fait donc sens de supposer qu'une majorité des documents y soient destinés. Pour ce qui est de l'occurrence des concepts liés à la dimension « Type de documents », nous constatons une certaine diversité des concepts. La Figure 7 présente les 10 premiers concepts en termes de nombre d'occurrences. Au total, 17 sur 40 furent identifiés au sein des noms de fichiers de notre corpus. Cela prouve une certaine qualité sémantique relative au choix des concepts. Ceux-ci semblent en effet être assez générique et commun pour décrire le contenu des documents de l'intranet.

3.1.2 Contenus textuels

Le contenu textuel est le cœur même des documents, il fournit toutes les informations nécessaires à la labélisation de ceux-ci. Cependant, il est également le plus complexe à traiter par un algorithme automatisé. La nature polysémique[8] des mots qui le compose nécessite une compréhension

[7] Hôpital de Sion - https://www.hopitalduvalais.ch/fr/lhopital-du-valais/sites/sion.html

[8] Qui a plusieurs sens.

contextuelle propre à chaque document. Ainsi, les concepts décrits dans notre taxonomie peuvent avoir une signification interdocuments variable qui ne se rapporte pas toujours aux labels associés. Similairement à l'analyse des noms de fichiers établis précédemment, nous commencerons cette examinassions par l'évaluation d'une recherche simple des concepts de notre taxonomie au sein des documents. Cela nous donne une première idée quant à la présence de ceux-ci au sein de notre corpus. Le Tableau 4 ci-après en présente nos résultats.

Dimensions	Langues	Taille du corpus	Nb documents trouvés	Proportion de documents trouvés
Géographie	FR	908	714	78,6%
Géographie	DE	638	460	72,1%
Type de documents	FR	908	448	49,3%
Type de documents	DE	638	265	41,5%

Tableau 4 Résultats d'une recherche textuelle simple sur le contenu des documents

Cette méthode, bien qu'imparfaite, pourrait couvrir jusqu'à 44,4% de la labélisation relative à la dimension « Type de documents » et 73% de la dimension « Géographie » (langues française et allemande confondues). Contrairement aux noms de fichiers, nous avons cette fois séparé l'analyse de nos dimensions par langues. Ceci pour évaluer l'efficience des concepts choisie pour chacune des langues traitées. Nous remarquons ainsi une proportion de documents trouvés relativement proche entre les concepts français et allemand : 78,6% contre 72,1% pour la dimension « Géographie » et 49,3% contre 41,5% pour la dimension « Type de documents ». Ces proportions pourraient démontrer une forte cohésion entre ces concepts indiquant que nos traductions sont de bonnes qualités.

Nous mentionnions précédemment que le choix des concepts semblait judicieux pour synthétiser le contenu des documents, cependant, offrent-ils des différences sémantiques substantielles permettant de les distinguer les uns des autres ? En effet, la présence de concepts sémantiquement proche au sein de nos dimensions pourrait générer des erreurs de classification. Heureusement, le domaine du traitement automatique du langage naturel (TALN) nous offre des outils permettant de mesurer cette similarité sémantique. Nous utilisons, pour ce faire, la librairie *Spacy* qui supporte nativement le calcul de la similarité sémantique. Nous l'exécutons sur leur modèle de réseau neuronal convolutif *fr_core_news_md*[9]. Celui-ci est leur modèle de taille moyenne en termes d'apprentissage sur la langue française, il s'oppose au *fr_core_news_sm* qui est plus petit et plus rapide. Leur documentation met cependant en garde que la précision quant à la similarité sémantique est meilleure plus le modèle utilisé est large. Une troisième taille de modèle plus large que les deux précédentes est disponible pour la langue anglaise uniquement. Il

[9] Celui-ci fut entraîné sur les corpus *French Sequoia* et *WikiNER*[54]. Spacy *fr_core_news_md* - https://spacy.io/models/fr#fr_core_news_md

est donc nécessaire d'être conscient de la précision imparfaite de ce modèle dans l'analyse des résultats présentés ci-après Tableau 5.

	catalogue	*communication*	*directive*	*formation*	*guide*	*menu*	*organigramme*	*procédure*	*processus*	*projet*	*rapport*	*questionnaire*
catalogue	1.00	0.13	0.00	0.08	0.05	0.02	0.01	0.08	0.08	0.11	0.08	0.08
communication	0.13	1.00	0.35	0.27	0.02	-0.03	0.04	0.27	0.50	0.17	0.50	0.27
directive	0.00	0.35	1.00	0.37	-0.03	0.04	0.15	0.37	0.41	0.15	0.41	0.37
formation	0.08	0.27	0.37	1.00	0.03	0.01	0.38	1.00	0.29	0.26	0.29	1.00
guide	0.05	0.02	-0.03	0.03	1.00	0.04	0.01	0.03	0.07	0.06	0.07	0.03
menu	0.02	-0.03	0.04	0.01	0.04	1.00	0.06	0.01	0.03	-0.10	0.03	0.01
organigramme	0.01	0.04	0.15	0.38	0.01	0.06	1.00	0.38	0.13	0.31	0.13	0.38
procédure	0.08	0.27	0.37	1.00	0.03	0.01	0.38	1.00	0.29	0.26	0.29	1.00
processus	0.08	0.50	0.41	0.29	0.07	0.03	0.13	0.29	1.00	0.19	1.00	0.29
projet	0.11	0.17	0.15	0.26	0.06	-0.10	0.31	0.26	0.19	1.00	0.19	0.26
rapport	0.08	0.50	0.41	0.29	0.07	0.03	0.13	0.29	1.00	0.19	1.00	0.29
questionnaire	0.08	0.27	0.37	1.00	0.03	0.01	0.38	1.00	0.29	0.26	0.29	1.00

Tableau 5 Matrice de corrélations de la similarité sémantique entre concepts des labels de la dimension "Type de documents"

À la vue de ces résultats, nous constatons une similarité sémantique élevée entre quatre groupes de concepts : *(procédure – formation)*, *(questionnaire – formation)*, *(questionnaire – procédure)* et *(rapport – processus)* qui rapportent une similarité parfaite de 100% (avec une certaine imprécision). Le deuxième taux de similarité le plus élevé est deux fois plus faible (50%), entre les groupes de concepts *(processus – communication)* et *(rapport – communication).* Ces distances sémantiques proches seront à prendre en considération lors de l'analyse des résultats. Le risque est que le classificateur retourne systématiquement les deux labels au lieu d'un, si leur sémantique est trop corrélée.

3.1.3 Métadonnées

Un ultime composant pouvant potentiellement améliorer la qualité de nos prédictions est les métadonnées des documents. Celles-ci sont leurs caractéristiques intrinsèques. Il s'agit d'informations telles que leur date de création et de dernières modifications, leur format et leurs auteurs. Dans le contexte de nos prédictions, nous supposions que leur taille pourrait avoir un impact sur leur labélisation respectif. Cependant, comme nous pouvons le constater sur la Figure 8, celle-ci semble difficilement diviser nos classes. Certains types de documents tels que "projets, plannings, calendrier" *(point 1)* ainsi que "communication, presse" *(point 2)* semblent se démarquer quelque peu des autres avec des documents généralement plus volumineux. Toutefois, l'écart-type de ces classes est significativement plus grand. Nous observons en effet que plusieurs de leurs documents ont également une taille tout à fait similaire aux autres classes rendant leurs prédictions basées sur cette seule caractéristique relativement incertaine.

Figure 8 Taille des documents du corpus labélisé par classes

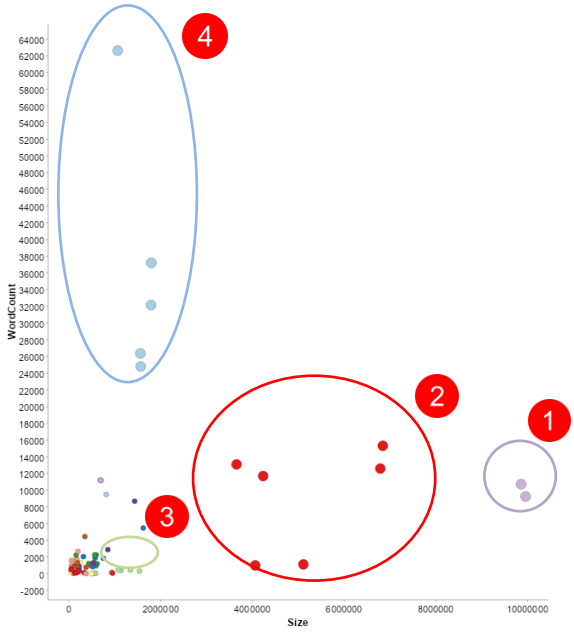


Figure 9 Taille des documents du corpus labélisé par nombre de mots

L'analyse des documents du corpus labélisé a cependant porté notre attention sur le fait que des documents appartenant à certaines classes semblaient contenir beaucoup d'images pour peu de texte. Les images occupant un espace considérable confronté au texte, celles-ci ont un impact significatif sur leurs tailles effectives. Pour cette raison, nous nous sommes intéressés au rapport entre la taille des documents et leur nombre de mots Figure 9. De cette analyse émane deux nouvelles classes créant leur propre cluster : "Organigrammes" *(point 3)* et "PV, rapports" *(point 4)*. Effectivement, nous constatons que le contenu des documents du type "PV, rapports" est généralement constitué de bien plus de mots que les autres pour une quantité d'images relativement faible n'impactant que très peu leurs tailles. À l'opposé, le type de document "Organigrammes" est lui généralement constituer de peu de mots pour beaucoup d'images.

Ces observations restent toutefois à prendre avec précaution, car elles sont réalisées sur notre corpus labélisé qui ne représente que 122 documents. Le risque que cet échantillon ne représente pas l'ensemble des documents qui constitue l'intranet de l'Hôpital du Valais est donc bien présent. Dans le but de limiter ce risque dans le cadre de cette analyse, nous tirons profit des documents du corpus non labélisé.

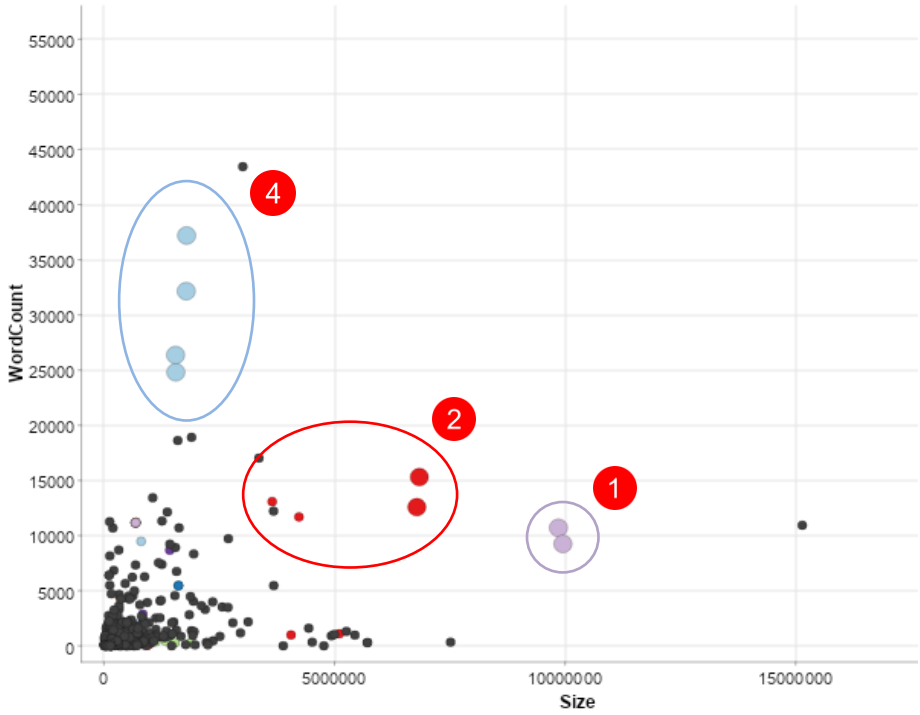


Figure 10 Taille des documents du corpus labélisé (couleurs) et non labélisé (noirs) par nombre de mots

Nous incorporons en effet ceux-ci au sein de notre nuage de points Figure 10 sous la forme de points noirs. De cette représentation, nous constatons que les caractéristiques évoquées précédemment semblent se confirmer pour les labels "Projets, plannings, calendrier" *(point 1)*, "Communication, presse" *(point 2)* et "PV, rapports" *(point 4)*.

3.1.4 Conclusions

Nous supposions en introduction de cette étude (cf. Chapitre 1.3), que le nom de fichier (H_2), le contenu textuel (H_1) et ainsi que les métadonnées (H_3) des documents pouvaient impacter les labels résultants. Dans l'analyse de nos corpus, nous en concluons que les trois hypothèses sont valides à un certain degré. En effet, bien que de manière très limitée, nous avons constaté que les concepts constituant nos labels sont présents sur ~5% des noms de fichiers des documents en moyenne. En ce qui concerne le contenu textuel, ces chiffres s'élèvent à 78,6% sur la dimension "Géographie" et 49,3% sur la dimension "Type de documents" (respectivement 72,1% et 41,5% sur les documents en allemand). Bien que toutes les instances de ces mots ne reflètent pas nécessairement le label associé par la taxonomie, il est indéniable qu'ils peuvent contribuer à leurs prédictions. En ce qui concerne leurs métadonnées, nos analyses semblent démontrer qu'il y a en effet une corrélation entre la taille des documents, leur nombre de mots et les labels associés. Pour la dimension "Type de documents" cette affirmation semble correcte pour les labels "Projets, plannings, calendrier", "Communication, presse", "PV, rapports".

3.2 Choix de la méthodologie

Compte tenu des informations relatives aux besoins, aux données et à la littérature que nous disposons à ce stade de l'étude, nous constatons des difficultés à proposer une solution de Machine Learning "classique". En effet, dans un contexte de méthodes d'adaptations d'algorithmes, les variantes de kNN et SVM pour gérer la problématique de multilabels (respectivement ML-kNN et RankSVM) semblent largement utilisées pour solutionner ce type de

problématiques. Cependant, entraîner ces algorithmes de manière supervisée requiert une quantité de données considérable. Ayant beaucoup de données non labélisées à disposition, une approche semi-supervisée pourrait être intéressante. Toutefois, là encore, les quelques documents disponibles par labels (0 à 12) permettront difficilement d'atteindre une qualité suffisante pour générer des pseudo-labels satisfaisants. De plus, les besoins du mandant vont au-delà de la dimension "Type de documents" pour laquelle nous disposons d'un corpus d'apprentissage. En effet, celui-ci souhaiterait pouvoir accélérer une grande partie du processus d'importation de documents dans l'intranet de l'Hôpital du Valais en automatisant la labélisation d'une grande partie des dimensions de leur taxonomie. Ainsi, la solution proposée part [48] quant à l'indexation par le principe de cooccurrence nous semble la meilleure alternative. Nos analyses statistiques révèlent qu'une simple recherche des concepts taxonomique dans le contenu textuel permet de déterminer au moins un label dans respectivement 78,6% des cas sur la dimension "Géographie" et 49,3% sur la dimension "Type de documents" (72,1% et 41,5% sur les documents en allemand). Il est cependant évident que rechercher uniquement les concepts directement dans le texte résulterait probablement en un taux de précision plutôt faible. Ceci, car il est nécessaire de tenir compte du contexte dans lequel ses mots sont trouvés pour déterminer s'ils correspondent à nos labels. La cooccurrence permet de faire exactement cela. Son principe est de rechercher non seulement les concepts, mais également des mots décrivant le contexte de ces concepts. En sémantique lexicale, le terme *descripteurs* est utilisé pour se référer à ces mots. Ainsi, chaque concept se voit attribuer un set de descripteurs. Lors de la prédiction, si le concept et au moins un de ces descripteurs associés sont trouvés au sein du contenu textuel, nous pouvons conclure avec un certain niveau de confiance que le label associé au concept est correct. Les principes de Machine Learning entrent en jeu pour déterminer les descripteurs de chacun des labels. De la littérature scientifique, nous extrayons l'algorithme Word2Vec ainsi que l'indexation sémantique latente (LSI). Nous avions en effet constaté que ces deux solutions sont largement utilisées dans la résolution de cette problématique. Une fois entraîné sur un corpus labélisé, il est possible d'interroger le modèle Word2Vec résultant pour récupérer les mots dont la distance sémantique est la plus proche pour chacun de nos concepts. Nous considérons ces mots comme nos descripteurs dans l'utilisation de ce modèle. LSI quant à lui, ne permet pas de calculer la distance sémantique entre deux mots. Son rôle est d'extraire les topiques (mots) qui décrivent l'ensemble de documents du corpus d'apprentissage de la manière la plus pertinente. Nous verrons lors de l'implémentation que cette différence impacte quelque peu la granularité de notre modèle de donnée relatif à l'évaluation des algorithmes.

Pour en revenir à nos hypothèses, nous concluions dans le sous-chapitre qui précède, que nos trois hypothèses liées aux données intrinsèques des documents sont véridiques. Dans le contexte de notre choix méthodologique, nous décidons d'utiliser uniquement deux de ces facteurs d'influences : le nom de fichier (H_2) ainsi que le contenu textuel (H_1) pour prédire nos labels. En effet, en essayant de tirer profit des métadonnées, nous retombons sur les mêmes problématiques précédemment établies quant au manque de données relatif à l'utilisation d'algorithmes de Machine Learning "classiques". Dans le but de rester fidèle au processus de classification textuelle présenté dans notre revue de littérature (cf. Chapitre 0), l'apprentissage des algorithmes sélectionnés suivra les recommandations mentionnées.

3.3 Outils

Dans le cadre de notre étude, nous utilisons un certain nombre d'outils au sein des différentes phases d'analyses et de réalisons. La liste de phases ci-dessous décrit ces outils et motive le choix de leurs utilisations :

Compréhension des données

Pour l'analyse de nos données, nous avions besoin d'outils à la fois efficients et faciles d'utilisation. Ceci dans le but de limiter les blocages liés à l'apprentissage des outils et ainsi se concentrer sur uniquement sur les données. Cela a accéléré grandement la vitesse de réalisation des premiers tests liés au processus de réalisation de la solution. Pour ce faire, l'outil Knime[10] a été utilisé. Celui-ci coïncide parfaitement avec l'expression de nos besoins. Effectivement, proposant une interface graphique basée sur la notion de "glisser-déposer", la réalisation de flux fut extrêmement efficiente par rapport à une solution programmatique. Son module de traitement automatique du langage naturel[11] nous a permis d'utiliser toute la puissance des algorithmes standard dans le cadre de nos analyses et de nos tests.

Prétraitement des données

Bien que Knime soit parfait dans le cadre de nos analyses, il n'est pas adapté pour la réalisation de notre prototype applicatif. En effet, fonctionnant sur le *Java Runtime (JRE)* son exécution est relativement lente comparée aux solutions Python. La rapidité d'exécution étant un facteur crucial pour le mandant (cf. Chapitre 1.5), un changement d'outil est nécessaire pour les phases de développement. Nous nous sommes ainsi tournés vers les diverses librairies du langage Python qui est le langage phare dans le domaine du Machine Learning [49]. Un article de O'Reilly paru l'année passée compare les deux leaders du marché : Spark-NLP[12] et spaCy[13], deux librairies open sources. Dans sa comparaison, l'auteur crée un scénario réaliste d'utilisation des librairies. Il en conclut que bien que Spark-NLP est le plus rapide en termes de performances, spaCy offre plus de fonctionnalités natives qui le rend "prêt-à-l'emploi". Entre autres, il offre des modèles préentraînés sur plusieurs langues, dont le français, l'allemand et l'anglais ainsi que des fonctions de prétraitement qui automatisent largement le flux de travail. Devant tenir compte de la problématique du multilingue dans le cadre de cette étude, spaCy semble ainsi parfaitement adapté. C'est cet outil que nous utiliserons pour tous nos traitements textuels.

Création des modèles de prédiction

Notre phase de création des modèles de prédiction requiert l'apprentissage des modèles Word2Vec et LSI. Pour se faire, spaCy[14] recommande de le coupler à la libraire Gensim[15]. Celle-ci est spécialisée et réputée dans le domaine de l'analyse sémantique. Elle permettra via son API Python, d'entraîner et d'extraire les descripteurs et topiques de nos modèles.

---

[10] Knime - https://www.knime.com/

[11] Knime Text Processing - https://www.knime.com/knime-text-processing

[12] Spark-NLP - https://nlp.johnsnowlabs.com/

[13] spaCy - https://spacy.io/

[14] spaCy Vectors Similarity - https://spacy.io/usage/vectors-similarity

[15] Gensim - https://radimrehurek.com/gensim/

Évaluation

Finalement, nos besoins quant à la phase d'évaluation gravitent autour du calcul des mesures relatives à celles-ci. Scikit-Learn[16] est une librairie Python de Machine Learning. Elle est reconnue pour sa facilité d'utilisation et sa large panoplie d'outils d'exploration et d'analyse de données. Quant aux mesures, elle propose plus d'une cinquantaine de métriques pour l'évaluation des modèles de régression, de classification et de clustering. Nous utiliserons cet outil pour calculer nos mesures d'évaluation.

[16] Scikit-learn - https://scikit-learn.org/

3. Analyses et développement

Ce chapitre s'intéresse à la mise en place de la solution sélectionnée. Nous commencerons donc par exposer le modèle de données que nous utiliserons tout au long de l'implémentation. Nous préparerons ensuite nos données avec la phase de prétraitement durant laquelle nous tenterons d'améliorer leurs qualités. Nos deux modèles retenus : Word2Vec et l'indexation sémantique latente seront alors créées et entraînées sur notre corpus d'apprentissage. Nous évaluerons leurs performances et finirons par implémenter un prototype fonctionnel incluant le modèle aux meilleures performances dans la phase de déploiement.

3.4 Modélisation de la problématique

Réalisant l'intégralité du processus d'apprentissage, de test puis d'évaluation au sein d'un environnement Python, nous remarquons que la nature de notre problématique est assez complexe pour nécessiter la mise en place d'une structure d'objets pour la représenter.

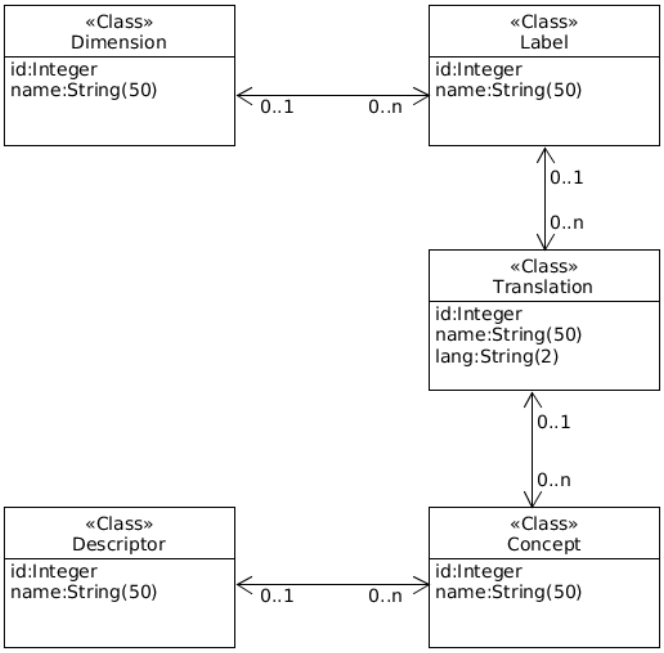


Figure 11 Diagrammes de classes (UML)

Le diagramme de classes de la Figure 11 présente ainsi nos choix fonctionnels quant à cette structure. Nous remarquons tout d'abord l'agencement en arbre des associations. Celles-ci représentent les nœuds de notre taxonomie. Ainsi, la classe *Dimension* est le nœud racine, celui-ci est constitué de zéro ou plusieurs *Label*. Un premier choix fonctionnel est de représenter les traductions des labels dans un objet à part entière. Pour rappel, un label peut être rédigé en plusieurs langues (cf. Chapitre 1.5). Dans le cadre de cette étude, nous traiterons les langues française et allemande. Nous justifions le choix de cette approche, car l'Hôpital du Valais utilisera un outil de traduction pour proposer leurs intranets en plusieurs langues. Ainsi, leur structure informationnelle sera composée d'une langue mère et de plusieurs traductions de celle-ci. Notre modèle ne fait que répliquer cette approche. Un deuxième choix fonctionnel est la décomposition des labels en concepts. En effet, nous notions qu'au sein de la taxonomie fournie, un label était parfois composé de plusieurs mots tels que *"Communication, presse"* ou encore *"Catalogues, glossaires".* Dans le cadre de notre solution d'analyse sémantique, il est nécessaire de réaliser celle-ci sur chacun de ces mots. Ainsi, nous les décomposons en instance d'objets *Concept* pour faciliter leurs traitements. Un *Concept* est finalement constitué de zéro ou plusieurs *Descriptor*

(descripteurs). Ces derniers seront utilisés uniquement dans le cadre de l'apprentissage et des tests des deux variantes des modèles Word2Vec. Les topiques du modèle d'indexation sémantique latente quant à eux, ne décrivent pas nos concepts, mais les documents associés à nos labels. Ainsi, nous considérons les topiques n'étant nul autre que des concepts. Au sein de notre modèle objet, ils seront stockés dans la classe *Concept*.

Dans le but de faciliter le transfert des données entre les différentes phases de notre processus, nous persistons celles-ci dans une base de données. Pour ce faire, nous utilisons le système de gestion de base de données (SGBD) SQLite[17]. La connexion à la base depuis l'environnement Python se fait quant à elle grâce à la libraire SQLAlchemy[18]. Ces tables et relations seront générées à partir de notre modèle objet. Au niveau de la base de données, nous avons défini les contraintes d'unicités suivantes :

- Deux traductions ne peuvent pas avoir le même nom et la même langue s'ils sont attribués au même label ;
- Deux labels ne peuvent pas avoir le même nom s'ils sont attribués à la même dimension.

3.5 Prétraitement des données

3.5.1 Processus de nettoyage

La phase de prétraitement des données consiste à préparer les données en vue d'entraîner nos modèles de prédictions. La première étape de ce prétraitement consiste à représenter les documents sous une forme qui convient à la fouille de texte.

```
# Mettre le tout en minuscule

content = content.lower()

# Remplace les retours à la ligne en espace

content = re.sub("\\r\\n|\\r|\\n", " ", content)

# Ne garder que les caractères des langues FR, DE et EN

content = re.sub("[a-zA-ZàâäôéèëêîïçùûüÿæœÀÂÄÔÉÈËÊÎÏŸÇÙÛÜÆŒäöüßÄÖÜß]", " ",
content).strip()

# Supprimer tous les mots de moins de 3 lettres

content = re.sub(" .{1,2} ", " ", content)

# Supprimer les espaces au début, à la fin et s'il y en a plus qu'un successivement

content = re.sub(" {2,}", " ", content).strip()

# Ajouter les mots d'arrêt français au vocabulaire

for w in spacy.lang.fr.stop_words.STOP_WORDS:

    nlp.vocab[w].is_stop = True
```

[17] SQLite - https://www.sqlite.org/index.html

[18] SQLAlchemy - https://www.sqlalchemy.org/

```
# Ajouter les mots d'arrêt allemand au vocabulaire

for w in spacy.lang.de.stop_words.STOP_WORDS:

    nlp.vocab[w].is_stop = True

# Conversion du contenu en Document

doc = nlp(content)

# Extraction des token

tokens = [token.lemma_ for token in doc if not token.is_stop]
```

Code 1 Script de nettoyage des données et d'extraction des tokens (Python)

Nous initions ainsi par quelques procédés visant à améliorer la qualité des données avant la tokenisation. Nous commençons par mettre l'intégralité du texte en minuscule. Au niveau informatique, les caractères minuscules et majuscules ont en effet une valeur ASCII différente. La même lettre sous ses deux différentes formes sera ainsi considérée comme deux lettres distinctes au sein d'un algorithme. Cette première opération est donc cruciale pour éviter tout problème par la suite. Les retours à la ligne au sein du texte génèrent du bruit et peuvent induire en erreur le processus de tokenisation. Ils sont donc remplacés par des espaces qui sont les séparateurs sur lesquels se base le processus. Du contenu, nous ne gardons ensuite que les caractères additionnels des langues de notre corpus, soit le français, l'allemand et l'anglais. Ceci permet de filtrer tous caractères spéciaux ainsi que la ponctuation et les chiffres qui n'apportent aucune plus-value à l'apprentissage de nos modèles. Pour cette même raison, les mots de moins de trois lettres sont également supprimés. Nous cherchons à ne garder que les mots offrant de l'information quant au contenu des documents. Ayant introduit des espaces au sein du texte, nous nous assurons ensuite que le contenu ne dispose pas plus d'un seul espace consécutif. Cela pour ne pas biaiser la tokenisation. Les mots d'arrêts sont également supprimés. Pour ce faire, nous utilisons la liste de mots d'arrêts de la langue française et allemande définie par la librairie *Spacy*, que nous ajoutons à son vocabulaire de base contenant la langue anglaise. Les deux dernières étapes consistent à convertir le contenu en *document* qui est une représentation qui facilite le traitement textuel par les algorithmes des librairies de TALN, puis exécuter le processus de tokenisation. Les tokens résultants sont finalement lemmatisé. Nous choisissons la lemmatisation plutôt que la racinisation, car nous ne voulons pas perdre la signification d'origine des tokens. Cela pourrait en effet compromettre l'analyse de similarité si les concepts de la taxonomie ne sont pas trouvés au sein du vocabulaire des documents d'apprentissage de nos modèles.

3.5.2 Partitionnement

Comme précédemment mentionné, notre corpus labélisé est relativement petit (122 documents). Le processus d'apprentissage d'un algorithme de Machine Learning requiert le partitionnement de ces documents en deux sets de données qui serviront respectivement à l'entraînement puis aux tests de l'algorithme [50]. Utiliser les mêmes documents pour ces deux phases distinctes est considéré comme une faute critique, car biaisant inévitablement l'évaluation de l'algorithme (taux de précision et d'exactitude anormalement haut). La littérature préconise donc un partitionnement allant de 70 à 90% pour le set d'entraînement [51]. Le reste constituant le set de tests.

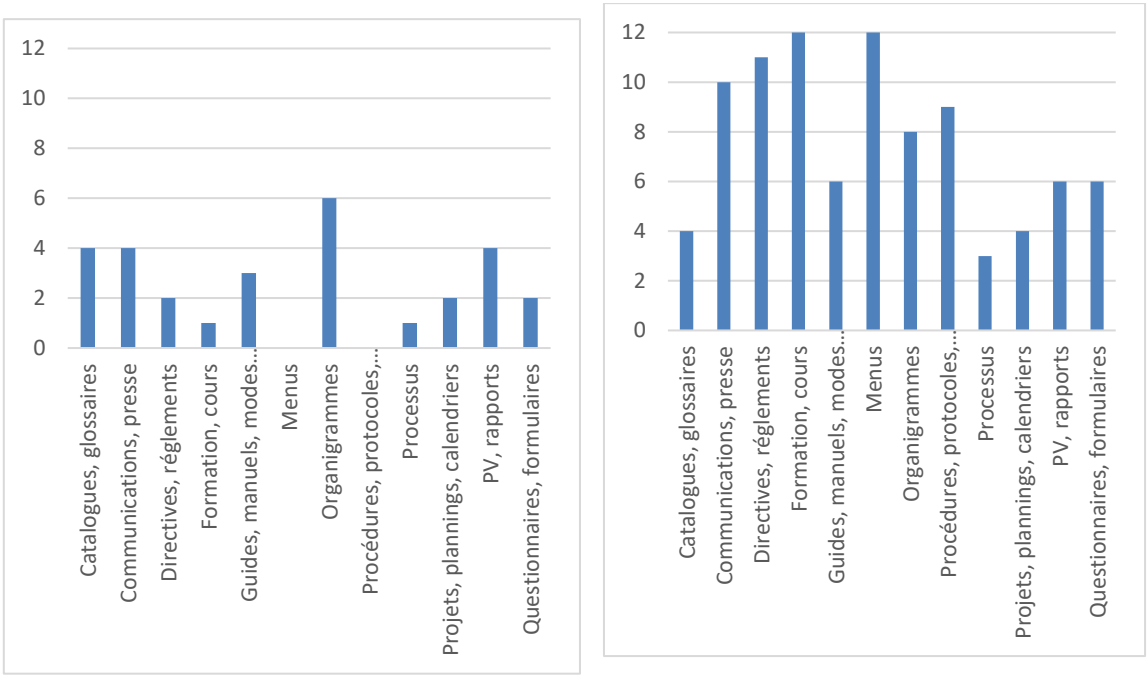


Figure 12 Répartition des fichiers par classes pour la langue allemande (gauche) et française (droite)

Notre corpus labélisé contient des documents en français (76%) et en allemand (24%) réparti entre douze labels qui constitue la dimension "Type de documents". La Figure 12 présente la répartition des fichiers au sein de ces labels par langues. Nous constatons l'absence de documents en allemand pour les labels "Menus" et "Procédures, protocoles, instructions" ainsi que le faible nombre de documents relatifs à cette langue. Respecter le partitionnement recommandé par la littérature est ainsi parfois impossible. C'est la raison pour laquelle nous avons décidé de n'utiliser que les labels pourvus d'au moins trois documents. Dans ces cas-ci, deux documents sont utilisés pour l'entraînement (66,7%) et un seul pour les tests (33,3%). Seuls cinq labels sur douze seront ainsi utilisés pour l'entraînement des algorithmes relatifs aux documents en allemand. Pour ce qui est des documents en français, tous seront pris en compte. L'étude de [X] utilise les labels entraînés sur au moins cinq documents pour éviter que les topiques extraits par l'indexation sémantique latente (LSI) soient dus au pur hasard. En choisissant cette même contrainte, nous n'évaluerions qu'un seul label pour la langue allemande. Lors de l'analyse des résultats des performances de nos modèles de prédictions, nous tiendrons compte de ce manque de documents. Également, une autre caractéristique de notre corpus labélisé est qu'aucun de ses documents n'est associé à plusieurs label. Dans le contexte de l'utilisation de nos modèles d'extractions de caractéristiques sémantiques, cela est plutôt positif. En effet, cela permet de limiter les interdépendances des descripteurs et topiques entre concepts qui pourraient fausser la classification.

3.5.3 Gestion des documents hétérogènes et multilingues

Nous avions constaté l'hétérogénéité des formats de documents lors de nos différentes analyses statistiques liées à la compréhension des données (cf. chapitre 3.1). La librairie *Apache Tika*[19] nous semble parfaitement appropriée pour répondre à cette problématique. En effet, cette librairie détecte et extrait les données et métadonnées de plus d'un millier de formats de documents. Les différentes versions de la suite Microsoft Office ainsi que les PDF (semblent être les formats majoritairement présents au sein de l'intranet de l'Hôpital du Valais) sont donc supportées. En plus

[19] Apache Tika - https://tika.apache.org/

du chargement de documents hétérogènes, la librairie *Tika* permet également de déterminer la ou les langues de ceux-ci. Originalement accessible via une interface de programmation REST, nous utiliserons, dans le cadre de cette étude, une version portée de la librairie pour fonctionner nativement grâce au langage Python[20].

3.6 Modèles de prédiction

La création de nos modèles de prédiction fut triviale grâce à la librairie Gensim[21]. Ce chapitre s'intéresse à la création de nos modèles de prédiction pour l'indexation sémantique latente (LSI) et le Word2Vec. Également, celui-ci présente la logique de notre algorithme de prédiction.

3.6.1 Création du modèle d'indexation sémantique latente (LSI)

Au sein du code source présenté ci-dessus (Code 2), chaque ligne de notre liste *list_tokens* correspond au texte tokenizé d'un document de notre corpus d'entraînement.

```
# Création d'un dictionnaire Corpera

dct = corpora.Dictionary(list_tokens)

# Conversion des documents en BOW

corpusLSI = [dct.doc2bow(line) for line in list_tokens]

# Création du modèle LSI

model = LsiModel(corpus=corpusLSI, id2word=dct, num_topics=10)

# Extraction des topiques

model.print_topics(-1)
```

Code 2 Créations du modèle d'indexation sémantique latente

Gensim met à disposition des structures de données adapter aux opérations de traitement de données textuelles. *Corpera.Dictionary* est une de ces structures représentant un dictionnaire (ensemble clé-valeur). Nous convertissons ensuite chaque ligne de notre dictionnaire en représentation Bag-of-Words (sac de mots) avant de débuter l'entraînement de notre modèle d'indexation sémantique latente. Quatre paramètres sont passés à la fonction de création du modèle. *corpus* correspond au corpus d'entraînement, *id2word* permet de mapper les mots présents dans nos documents aux token extrait par la représentation BOW. Ce mappage permet d'accélérer l'entraînement en associant chaque mot déjà appris à un seul et même token. Ainsi, celui-ci n'a pas à être régénéré à chaque instance diminuant les coûts de temps et de mémoire. Finalement, *num_topics* n'est autre que le nombre de topiques à découvrir au sein du corpus. Une fois le modèle entraîné, la méthode *print_topics* permet d'en extraire les topiques. Son paramètre indique le nombre de topiques à extraire, une valeur de *-1* les extrait tous.

[20] Apache *Tika* pour python - https://github.com/chrismattmann/tika-python

[21] Gensim - https://radimrehurek.com/gensim/

3.6.2 Création du modèle Word2Vec

Le code source lié à l'entraînement du modèle Word2Vec est extrêmement similaire au précédent. Il est même plus intuitif, car les opérations de conversions en représentation BOW réalisées précédemment sont ici gérées par Gensim (Code 3).

```
# Création du modèle Word2Vec

model = Word2Vec([tokens], size=128, window=5, min_count=3, workers=4, sg=1)

# Récupérer les mots les plus proches

similarities = model.wv.most_similar("foo")
```

Code 3 Créations du modèle d'indexation sémantique latente

En effet, comme Gensim permet le choix quant à l'entraînement d'un modèle Word2Vec avec sa représentation CBOW ou Skip-gram, c'est la fonction même qui prend en charge les opérations de conversion. Ainsi, nous lui passons simplement la liste de tokens de nos documents en paramètres. Le paramètre *size* correspond au nombre de dimensions au sein de l'espace vectoriel (cf. Chapitre 2.3). [52] propose d'y attribuer un facteur de 32 et mentionne la taille 128 comme adapter à grand nombre de modèles standard. C'est cette taille que nous utiliserons dans le cadre de notre apprentissage. Lors du traitement, *window* désigne la distance maximale entre le mot courant et le mot prédit au sein d'une phrase. Travaillant avec des documents de tailles variables, nous attribuons la valeur qu'a définie Gensim par défaut, soit 5. Tous les mots dont la fréquence d'occurrence est plus basse que l'attribut *min_count* ne sera pas pris en compte lors de l'apprentissage. Ce paramètre permet donc d'appliquer un filtre supplémentaire de sélection des caractéristiques avant le passage de CBOW ou Skip-gram. Travaillant sur un corpus très petit, il est impératif de garder cette valeur relativement basse pour éviter que nos concepts soient indisponibles dans le vocabulaire final du modèle. Cela rendrait l'extraction de leurs descripteurs impossible. Finalement, le paramètre *worker* est le nombre de processus d'exécution en parallèle lors de l'apprentissage. Celui-ci est paramétrable lorsque l'algorithme fonctionne sur des machines multi-core pour augmenter sa rapidité d'exécution. Quant à *sg*, il indique à la fonction si elle doit exécuter le processus avec CBOW (*sg* à 0) ou Skip-gram (*sg* à 1).

3.7 Évaluation

Suite au partitionnement de notre corpus labélisé (cf. Chapitre 3.5.2), la Figure 13 expose les documents à notre disposition pour la phase d'évaluation.

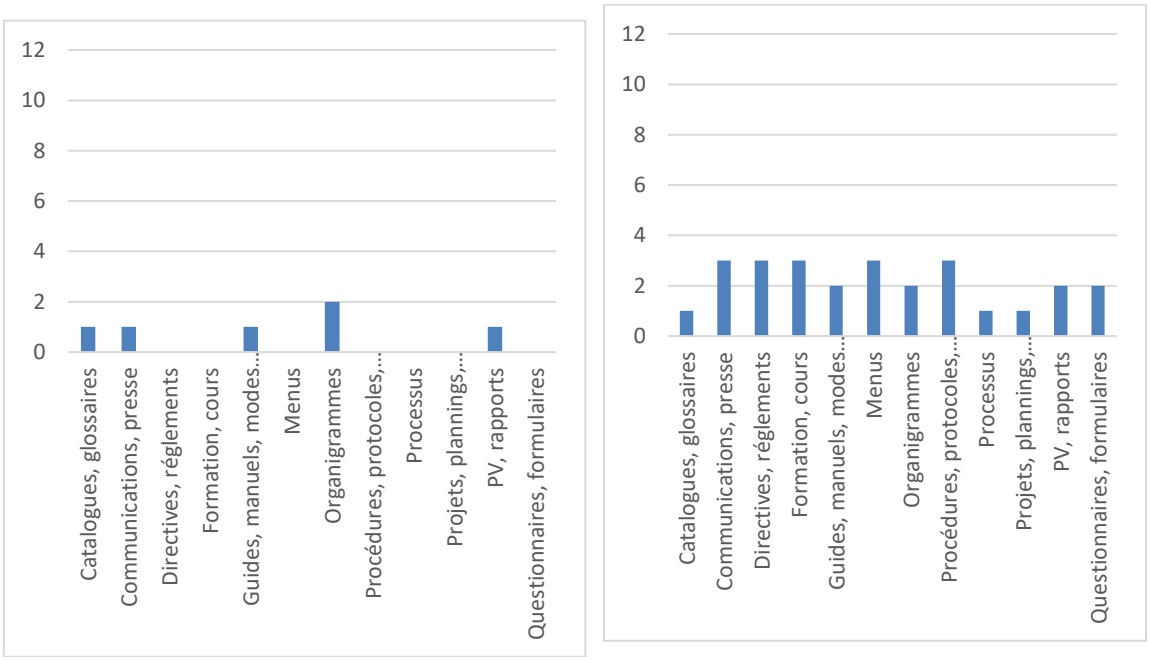


Figure 13 Répartition des documents d'évaluation par classes pour la langue allemande (gauche) et française (droite)

Notre stratégie quant à l'évaluation de nos modèles de prédictions va s'intéresser à comparer leurs performances avec les performances d'une recherche simple des concepts au sein des noms de fichiers et du contenu des documents. Nous pensons que cela est indispensable pour déterminer la pertinence des descripteurs et topiques proposés par nos modèles. Ainsi, nous avons imaginé deux tests d'évaluation :

- Recherche des concepts dans le texte pour récupérer une évaluation de base. La prédiction consiste à associer le label lorsqu'au moins un ces concepts est trouvé ;
- Recherche des concepts et des descripteurs. Le label est associé lorsqu'au moins un de ces concepts et un de ces descripteurs est trouvé. Pour LSI la notion de concepts et topiques étant mergée, le test se basera sur la découverte d'au moins deux concepts ou topiques.

Les sous-chapitres qui suivent présentent le choix des mesures d'évaluation ainsi que les résultats. Une conclusion, en fin de chapitre, fera la transition vers l'implémentation de notre prototype fonctionnel.

3.7.1 Définition des mesures

Dans le cadre de l'évaluation de nos modèles de prédiction, nous utiliserons les mesures proposées par [53] pour comparer leurs performances respectives. Ces mesures sont la précision, le rappel, le score F_1 et le *Hamming Loss.* Dans le cadre de leur étude, le *Hamming Loss* est utilisé en tant que mesure d'exactitude. C'est cette mesure qui prime quant à la finalité de l'évaluation des modèles. Nous ferons de même pour notre étude.

```
precision_score(y_true=np.array(actual_label_matrix), y_pred=np.array(predicted_label_matrix), average='weighted') # Précision
```

```
recall_score(y_true=np.array(actual_label_matrix), y_pred=np.array(predicted_label_matrix), average='weighted') # Rappel

f1_score(y_true=np.array(actual_label_matrix), y_pred=np.array(predicted_label_matrix), average='weighted') # Score F1

hamming_loss(y_true=np.array(actual_label_matrix), y_pred=np.array(predicted_label_matrix)) # Hamming Loss
```

Code 4 Calculs des mesures d'évaluation

Le Code 4 ci-dessus présente les fonctions de Scikit-Learn utilisées pour le calcul de nos métriques. Chacune de celles-ci se construit de la même manière. Nous avions mentionné la nécessité d'une matrice de prédiction et d'une matrice effective lors de notre revue de littérature relative à l'évaluation []. Les attributs *y_pred* et *y_true* correspondent respectivement à ces deux matrices, elles attendent un objet de type *np.array*. Celui-ci n'est d'autres qu'un tableau personnalisé avec une structure d'objets spécifique au calcul matriciel. Nous utilisons sur nos mesures un moyennage *weighted* qui calcul les métriques pour chaque label et détermine leurs moyennes pondérées sur l'ensemble du nombre d'instances vraies pour chacun d'entre eux. Nous justifions ce choix, car il tient compte du déséquilibre entre labels qui sont proéminents dans notre corpus.

3.7.2 Présentation des résultats

Comme précédemment mentionnés, nous avons commencé par nous intéresser aux résultats d'une recherche simple des concepts au sein du contenu et du nom de fichier des documents de notre corpus de test. Le Tableau 6 en expose les résultats.

	Français	**Allemand**
Score F1	0.207	0.333
Précision	0.169	0.300
Rappel	0.350	0.400
Hamming loss	0.279	0.150

Tableau 6 Résultats pour la recherche des concepts uniquement

La différence trop importante quant au nombre de documents relatif aux labels français et allemand nous empêche de comparer les résultats des deux langues entre elles. Ceci, car les constatations résultantes seraient totalement biaisées. Ainsi, nous comparerons les performances des modèles de prédictions pour ces deux langues de manière indépendante. Le Tableau 6 constitue nos résultats de références. Sur les documents français, nous constatons un rappel global de base à 35% pour une précision de 16,9%. 4/12 labels présentent un score F1 positif (cf. Annexe X pour le détail des résultats) : "Communication, Presse", "Cours, Formation", "Menus" et "PV, Rapport". Après analyses, ces labels font pour la majorité partie de ceux accumulant le plus de mots compte tenu de la taille de leurs corpus respectifs. Il fait en effet sens de supposer que plus les mots à disposition sont nombreux, plus les chances d'y trouver nos concepts augmentent. Ceci, peu importe l'origine sémantique du mot et de son rapport avec les labels de la taxonomie. L'exception semble être le label "Menus" dont les documents sont systématiquement les mêmes, soit très condensée et comprenant toujours le mot "menu". Cela lui octroie un score F_1 de 66,6%. Nos constatations sont les mêmes pour les documents de langue allemande, dont 2/5, labels présentent un score F_1 positif : "Guides, Manuels, Modes d'emploi" et "PV, Rapport".

	Word2Vec (FR)	LSI (FR)	Word2Vec (DE)	LSI (DE)
Score F1	0.186	0.230	0.000	0.313
Précision	0.162	0.143	0.000	0.197
Rappel	0.250	0.750	0.000	0.800
Hamming loss	0.183	0.608	0.000	0.617

Tableau 7 Résultats pour la recherche des concepts et descripteurs/topiques des modèles Word2Vec et LSI (10 topics)

En ajoutant la recherche de nos descripteurs (Word2Vec) et de nos topiques (LSI), nous remarquons que sur cette faible quantité de documents, nos résultats présentent une exacte correspondance quant aux spécifications CBOW et Skip-gram de notre modèle Word2Vec sur les deux langues de notre corpus. Dans le cadre de cette évaluation, nous considérerons donc les résultats de ce modèle indépendamment ses spécifications. Ainsi, en comparaison avec nos prédictions de références, nous notons une amélioration de –9,6% quant au *Hamming Loss* pour notre modèle Word2Vec français. Le score F_1 sur ce modèle est plus accru de respectivement 1,1 et 16,7% sur la prédiction des labels "Cours, Formation" et "PV, Rapport". Celui-ci a cependant diminué de 21,2% sur la prédiction du label "Communication, Presse", ce qui explique le score F_1 global plus bas. Le modèle LSI quant à lui, performe de manière bien pire que notre référence avec une augmentation de 32,9% relative au *Hamming Loss.* Nous remarquons sur ce modèle, un meilleur rappel (+%35) pour une moins bonne précision (-15,7%) qui démontre que le classificateur associe plus de labels aux documents, mais que ceux-ci sont moins pertinents (il ajoute du bruit au sein des prédictions). Le constat quant au modèle LSI relatif à la langue allemande est le même. Celle-ci présente en effet un *Hamming Loss* supérieur de 46,7%. Une observation intéressante est les mesures du modèle Word2Vec allemand. En effet, en exécutant son apprentissage, nous avons remarqué que celui-ci n'a généré aucun descripteur. Pour cette raison, la condition de notre test (au moins un concept et un descripteur) ne peut jamais être vraie, résultant en une série de mesures à 0. Après analyses, nous avons déterminé qu'aucun descripteur n'a pu être découvert, car aucun de nos concepts n'a été trouvé dans le vocabulaire résultant de la phase d'apprentissage du modèle. Notre calcul de la similarité sémantique ne peut effectivement pas retourner de descripteurs lorsque le concept n'a jamais été appris (vectorisé). Ceci est indéniablement dû au manque de documents relatifs à notre corpus d'entraînement en allemand.
Nous doutons que les performances désavantageuses de nos modèles LSI soient dues à la différence dans le nombre de topiques qu'ils découvrent par rapport aux descripteurs pour notre modèle Word2Vec. Dans le cadre de notre évaluation, LSI a en effet été configuré pour découvrir dix topiques alors que Word2Vec ne découvre en moyenne que six descripteurs par concepts. Nous avons par conséquent entrepris un dernier test qui consiste à diminuer le nombre de topiques relatifs à chaque label à 5 pour le modèle LSI (Tableau 8).

	LSI (FR)	LSI (DE)
Score F1	0.226	0.340
Précision	0.145	0.217
Rappel	0.650	0.800
Hamming loss	0.513	0.483

Tableau 8 Résultats pour la recherche des concepts et topiques du modèle LSI (5 topics)

En faisant cela, nous observons une évolution positive des mesures : le *Hamming Loss* a diminué de respectivement 9,5 et 13,4% sur nos modèles français et allemand tout en gardant un rappel similaire sur les documents en allemand. Le modèle français a cependant connu une diminution de rappel de 10%.

3.7.3 Conclusions

À la vue de nos résultats, distinguer la meilleure solution n'est pas tâche aisée. Effectivement, sur le peu de données à disposition, les deux modèles performent de manière similaire en termes de précision (respectivement 14,5 et 16,2% pour LSI et Word2Vec sur le corpus français). La différence majeure entre les deux réside dans le nombre de labels résultant leurs prédictions (Tableau 9).

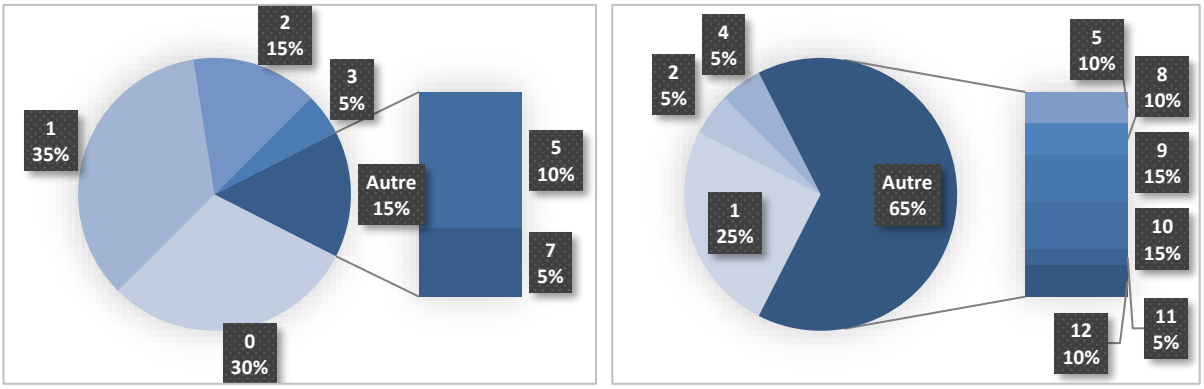


Tableau 9 Nombres de classes résultantes des prédictions françaises relatives aux modèles Word2Vec (gauche) et LSI (droite)

D'un côté, Word2Vec prédit relativement peu de labels (entre un et trois sur 55% du corpus) alors que LSI est son opposé (plus de quatre sur 65% du corpus). D'un point de vue rationnel, en analysant les labels de la dimension « Type de documents », nous pouvons supposer qu'un document appartenant à plus de trois labels est peu probable. Ceci, car sémantiquement, certains d'entre eux semblent présenter une mutuelle exclusion. De plus, tous les documents de notre corpus sont monolabélisés. La qualité de la classification étant cependant incertaine, nous ne pouvons affirmer que c'est effectivement systématiquement le cas dans la réalité. Mais cela nous indique toutefois que les documents de cette dimension présentent généralement peu (voire un seul) label. Pour une précision similaire, les deux modèles constituent ainsi deux approches distinctes de résolution de la problématique. Dans la première (Word2Vec), l'on accepte que le modèle de prédiction ne retourne parfois aucune valeur (c'est le cas sur 30% de notre corpus de tests) et dans la seconde (LSI), l'on accepte que le modèle nous retourne systématiquement trop de labels dont le bon dans la majorité des cas (65% de rappel). Le choix de cette approche dépend de l'utilisation finale de la solution. Une discussion avec le mandant nous informe que celle proposant peu de labels est préférable dans leur contexte. C'est ainsi sur le modèle Word2Vec que ça basera l'implémentation de notre prototype fonctionnel.

3.8 Déploiement

Au sein de cette ultime phase, nous présentons la réalisation d'un prototype fonctionnel basé sur le modèle de prédiction Word2Vec. Ce prototype réutilise le modèle de données présenté dans le chapitre précédent (cf. Chapitre 3.4). Il présente la création d'une API REST, notre stratégie

d'amélioration continue puis une proposition de déploiement de la solution basé sur la conteneurisation.

3.8.1 Création de l'API REST

Dans un premier temps, la réalisation de notre prototype fonctionnel à requit la création d'un service REST[22] pour permettre à l'utilisateur d'exécuter les opérations de création, lecture, mise à jour et suppression des données de notre modèle objets (cf. Chapitre 3.4). Ces opérations seront dictées par des requêtes au format JSON[23].

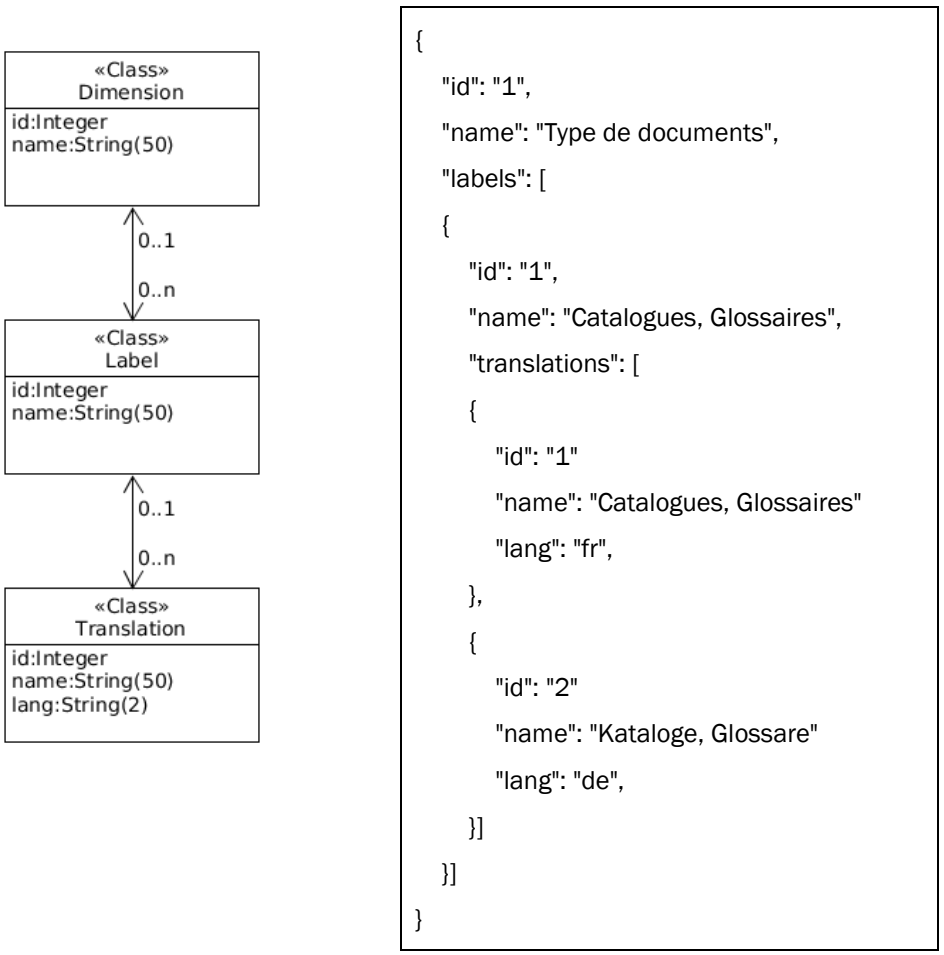


Figure 14 Modèle de classes (à gauche) et sa représentation JSON (à droite)

La Figure 14 compare une partie de notre diagramme de classes avec la représentation JSON de quelques-unes de ces ressources. Notez que les classes exposées au sein de cette figure

[22] REST (representational state transfer) est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web. Ceux-ci permettent aux systèmes effectuant des requêtes de manipuler des ressources web via leurs représentations textuelles à travers un ensemble d'opérations uniformes et prédéfinies sans état. - Wikipédia.org

[23] JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée - Wikipedia.org

définissent la granularité la plus fine des ressources que l'on décide de partager avec l'utilisateur. En effet, dans notre modèle de données (cf. Chapitre 3.4), les traductions (Translation) contiennent une liste de concepts (Concept) qui contiennent eux-mêmes une liste de descripteurs (Descriptor). Cependant, ceux-ci sont autogénérés respectivement par des déclencheurs en base de données et par nos modèles Word2Vec. Ainsi, il ne fait de sens de les présenter à l'utilisateur, nous les utiliserons uniquement dans le cadre de nos prédictions. En ce qui concerne la conversion des objets Python en JSON, nous utilisons le Framework Marshmallow[24]. Celui-ci nécessite de crée de classes dites "schéma" pour définir les attributs à sérialiser[25] ainsi que leurs types respectifs. Un schéma est défini pour chaque classe que l'on souhaite pouvoir sérialiser.

3.8.1.1 Déclencheurs SQL

En base de données, les déclencheurs décrivent des morceaux de code source pouvant être exécutés avant ou après un événement. Dans notre cas, notre déclencheur est exécuté après l'insertion d'une nouvelle traduction.

```
CREATE TRIGGER IF NOT EXISTS tg_after_insert_translation
AFTER INSERT ON translation
FOR EACH ROW
BEGIN
INSERT INTO concept (name, translation_id)
WITH split(word, str) AS (
SELECT '', NEW.name||','
UNION ALL SELECT
substr(str, 0, instr(str, ',')),
substr(str, instr(str, ',')+1)
FROM split WHERE str!=''
) SELECT trim(word), NEW.id FROM split WHERE word!='';
END;
```

Code 5 Déclencheurs SQL d'extraction des concepts

Comme nous avons déterminé que labels relatifs à la taxonomie sont parfois une série de mots séparés par des virgules, le rôle de ce déclencheur est d'extraire les mots du label en utilisant ce délimiteur. Les mots résultants sont alors directement insérés dans la table relative aux concepts. L'avantage de réaliser cette logique au niveau de la base de données plutôt que du code Python est de garantir qu'elle sera systématiquement exécutée après l'insertion d'une traduction. Ainsi, même si l'utilisateur décide pour une quelconque raison d'insérer une traduction directement dans la base de données sans passer par le service REST, la génération des concepts reste

[24] Marshmallow - https://marshmallow.readthedocs.io/en/stable/

[25] Processus de conversion des structures de données dans un format pouvant être stocké ou transmis et reconstruit ultérieurement (désérialisé) - Wikipedia.org

fonctionnelle. Un second déclencheur a également été créé pour gérer l'opération de mise à jour d'une traduction.

3.8.1.2 Service de requêtage

La dernière étape de création de notre service REST est la création du service de requêtage qui va nous permettre de recevoir les requêtes de l'utilisateur. Pour ce faire, nous avons choisi d'utiliser le Framework Flask-RESTPlus[26]. Fonctionnant sur le serveur web WSGI (Web Server Gateway Interface), celui-ci est optimisé pour faciliter la création d'un tel service. Le Tableau 10 ci-dessous présente les URLs d'accès des différentes ressources classés par méthode HTTP[27].

GET	/dimensions /dimensions/{pk_dimension} /dimensions/{dimension_name} /dimensions/{pk_dimension}/labels /dimensions/{dimension_name}/labels /dimensions/{pk_dimension}/labels/{pk_label} /dimensions/{dimension_name}/labels/{pk_label} /dimensions/{pk_dimension}/labels/{pk_label}/translations /dimensions/{dimension_name}/labels/{pk_label}/translations /dimensions/{pk_dimension}/labels/{pk_label}/translations/{pk_translation} /dimensions/{dimension_name}/labels/{pk_label}/translations/{pk_translation}
POST	/predict /true-labels /dimensions /dimensions/{pk_dimension}/labels /dimensions/{dimension_name}/labels /dimensions/{pk_dimension}/labels/{pk_label}/translations /dimensions/{dimension_name}/labels/{pk_label}/translations
PUT	/dimensions/{pk_dimension} /dimensions/{dimension_name} /dimensions/{pk_dimension}/labels/{pk_label} /dimensions/{dimension_name}/labels/{pk_label} /dimensions/{pk_dimension}/labels/{pk_label}/translations/{pk_translation} /dimensions/{dimension_name}/labels/{pk_label}/translations/{pk_translation}

[26] Flask-RESTPlus - https://flask-restplus.readthedocs.io/en/stable/

[27] L'Hypertext Transfer Protocol (HTTP) est un protocole de communication client-serveur développé pour le World Wide Web. - Wikipedia.org

DELETE	/dimensions/{pk_dimension} /dimensions/{dimension_name} /dimensions/{pk_dimension}/labels/{pk_label} /dimensions/{dimension_name}/labels/{pk_label} /dimensions/{pk_dimension}/labels/{pk_label}/translations/{pk_translation} /dimensions/{dimension_name}/labels/{pk_label}/translations/{pk_translation}

Tableau 10 URLs d'accès aux ressources par méthodes HTTP

Le protocole HTTP définit dans sa spécification un certain nombre de méthodes. Dans le cadre de la réalisation d'un service REST, quelques-unes de ces méthodes sont utilisées pour différencier les opérations de lecture, d'ajout, de modification ou de suppression des données. Ainsi, nous utilisons la méthode GET pour l'accès des données en lecture, POST pour l'ajout, PUT pour la mise à jour et DELETE pour la suppression. Dans le contexte de GET, La sémantique d'accès aux ressources est le nom de la classe aux pluriels (/dimensions) pour récupérer toutes les dimensions. En ajoutant ensuite l'identifiant de la ressource ou son nom, l'on obtient une seule dimension. Ainsi, /dimensions/1 retourne la dimension avec l'identifier (ID) numéro 1 et /dimensions/Géographie retourne la dimension dont le nom est "Géographie". Les ressources peuvent ensuite s'enchaîner pour récupérer les labels d'une dimension puis les traductions d'un label. La méthode d'ajout POST s'exécute quant à elle sur les noms de classes aux pluriels uniquement et les méthodes de mise à jour (PUT) et de suppressions (DELETE) sur les ressources singulières. Par commodité, nous proposons l'accès à nos dimensions par identifiant ou par nom. Le Framework Flask-RESTPlus a pour avantage d'intégrer la surcouche Swagger[28]. Celle-ci déploie un assistant de requêtage directement à la racine du service REST (/). Nous proposons cette interface dans le cadre de notre prototype applicatif (Figure 15).

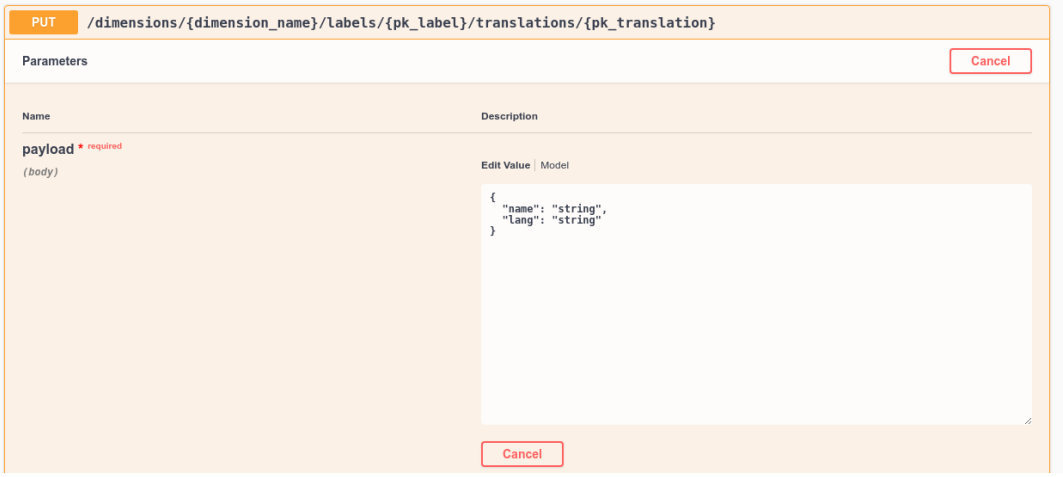


Figure 15 Interface de requêtage Swagger

[28] Swagger - https://swagger.io/

3.8.2 Amélioration continue

À la fin de la phase d'entraînement, nos modèles Word2Vec souffraient d'une précision relativement faible (cf. Chapitre 3.7.2). Nous avons pensé à une solution pour pallier au problème : l'amélioration continue.

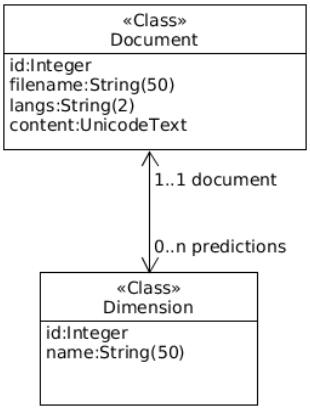


Figure 16 Diagramme de la classe Document (UML)

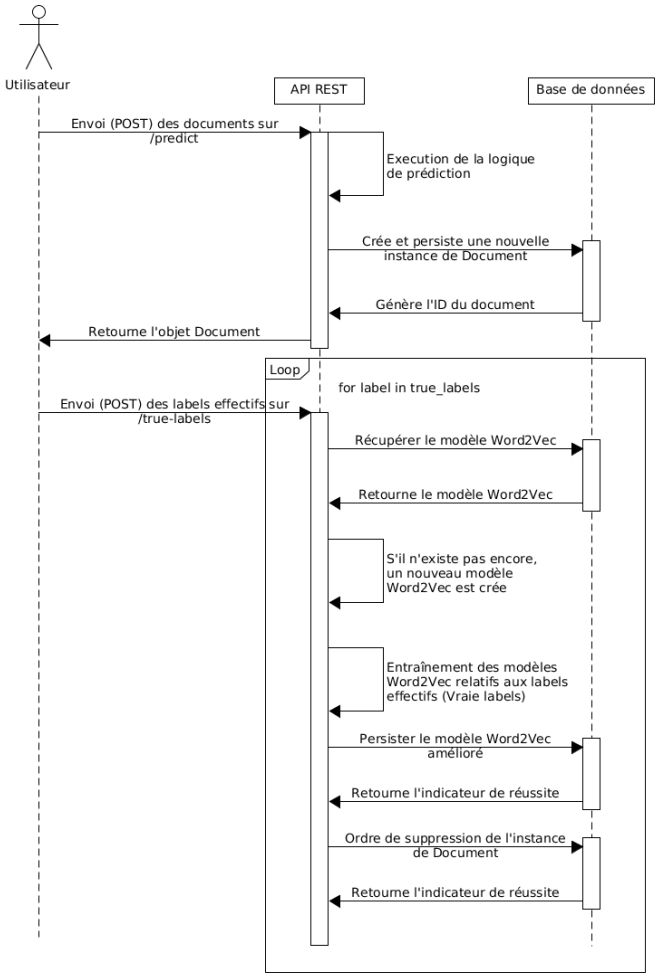


Figure 17 Diagrammes de séquence du processus de prédiction

Dans ce contexte, le diagramme de la Figure 17 présente les interactions entre l'utilisateur et notre système lors du processus de prédiction. Celui-ci commence par envoyer les documents à prédire à l'URL /predict via une requête POST à notre interface REST. Notre système exécute alors la logique de prédiction pour en découvrir les labels. Celle-ci consiste à trouver au moins un concept et un descripteur associé au sein du contenu textuel ou du nom de fichier. Lorsque les concepts d'un label n'ont pas encore de descripteurs (nouveaux labels), la logique de prédiction retourne le label associé si au moins un de ses concepts est découvert. Les prédictions résultantes sont ajoutées à une nouvelle instance de la classe Document (Figure 16) en plus du nom de fichier, de la/des langue(s) et du contenu extrait. Au sein de notre modèle objet, une prédiction correspond à une dimension comprenant les labels prédits. Ainsi, la classe Document peut contenir zéro ou plusieurs prédictions. Celles-ci ne sont bien entendu jamais persisté, car elles viendraient chambouler la structure taxonomique en place. Nous persistons donc uniquement les attributs *filename* et *langs* en base de données. Le SGBD retourne ensuite l'identifiant de la nouvelle entrée qui est ajouté à l'instance de Document, converti en JSON et retourné à l'utilisateur. Celui-ci peut alors corriger les potentielles erreurs de labélisation et renvoyer les labels effectifs en réalisant une nouvelle requête POST à l'URL /true-labels. Lorsque notre système reçoit les labels effectifs, ils les

utilisent pour améliorer les performances des modèles Word2Vec associés. Dans le cas où le label n'a pas encore de modèle associé (cela sera le cas des nouveaux labels), c'est à ce moment que celui-ci est créé. De notre modèle de données présenté au chapitre 3.4, nous avons donc ajouté un attribut *word2vec_model* à la classe Label pour y stocker leur modèle respectif. En base de données, celui-ci constitue un champ de type BLOB (de l'anglais *Binary Large Object*). Finalement, une fois avoir tiré profit des nouveaux documents dans le cadre de notre apprentissage continu, nous n'en avons plus d'utilité au sein de notre logique. Nous les supprimons de la base de données dans le but de limiter l'espace disque utilisé.

3.8.3 Stratégie de déploiement

L'unique condition quant à la stratégie de déploiement utilisée dans le cadre de cette thèse était que celle-ci soit facile à mettre en œuvre (cf. Chapitre 1.5). Nous avons donc pensé à la conteneurisation de la solution. Comme celle-ci n'est encore qu'un prototype, la conteneuriser permet de l'exécuter sans peines sur n'importe quelle machine pour y effectuer des tests fonctionnels. Dans le cadre de notre solution de déploiement, nous encapsulons donc notre application au sein d'un conteneur Docker[29]. Notre Dockerfile[30] est relativement simple :

```
# Image de base

FROM python:3.7.4

# Crée un dossier /app

RUN mkdir /app

# Copie l'intégralité du dossier courant dans /app

COPY . /app

WORKDIR /app

# Installe les dépendances

RUN pip install flask_sqlalchemy

RUN pip install marshmallow

....

# Expose le port 5000

EXPOSE 5000

# Lance le script main.py comme point d'entrée de l'image

ENTRYPOINT [ "python" ]
```

[29] Docker - https://www.docker.com/

[30] Fichier de configuration de Docker.

```
CMD [ "main.py" ]
```

Code 6 Logique du Dockerfile

Nous nous basons sur l'image python:3.7.4. Ceci pour faciliter l'installation de nos dépendances, qui repose essentiellement sur Python. Nous copions ensuite l'intégralité du répertoire courant dans le container. Ce répertoire est généralement la racine du projet. Il contient tous nos scripts Python de la logique applicative ainsi que nos modèles Word2Vec préentraînés. La suite du Dockerfile consiste simplement à installer les dépendances Python une à une puis à exposer le port 5000 et lancer le script *main.py* lors du démarrage de l'image. Le port 5000 est celui sur lequel fonctionne le serveur WSGI à l'intérieur du container. En exposant ce port, nous le rendant accessible depuis l'extérieur de l'image.

3.8.4 Interface de tests

Dans le but d'évaluer la rapidité des réponses de notre service REST quant aux prédictions de nos documents, nous avons réalisé une interface de tests en HTML/CSS et JavaScript.

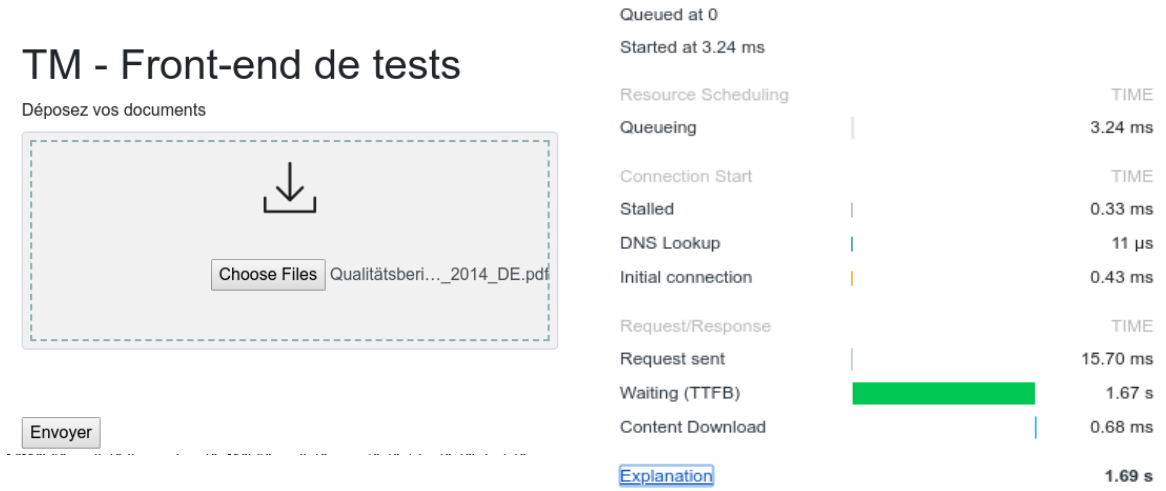


Tableau 11 Interface de tests (à gauche) et détails de la réponse (à droite)

L'évaluation a porté sur la prédiction d'un document en allemand de 34'334 mots pour une taille de 1,8 Mégaoctet. La réponse a été reçue après 1,69 seconde (Tableau 11).

4 Discussion

Nos hypothèses, au caractère très générique, furent validées assez rapidement au cours de notre étude. Nous le rappelons, celles-ci cherchent à répondre à notre question de recherche : ***quels sont les facteurs impactant les labels liés aux documents constituant l'intranet de l'Hôpital du Valais ?***

En concevant nos hypothèses, nous étions confient que notre hypothèse H_1 serait valide. Celle-ci suppose que le contenu des documents peut impacter leurs labélisations. La littérature scientifique proposant de multiples méthodes pour y arriver, cela ne faisait aucun doute. Cependant, en récoltant plus de détails quant à la problématique, nous avons constaté que les circonstances de l'étude ajoutaient une complexité non négligeable à l'application de ces méthodes. La nature de cette complexité provient de la volonté de satisfaire au mieux les besoins du mandant tout en se contentant des données à disposition. En début d'étude, aucune donnée prélabélisée n'était à disposition. La conception de notre corpus d'apprentissage fut difficile, nous avions initialement dans l'idée que des collaborateurs métiers se charge de la classification manuelle d'un set de documents. Il s'est avéré que cela fut impossible. Selon le mandant, personne n'était disposé à se charger de la tâche en interne. Ainsi, c'est lui-même qui a entrepris la création du corpus. Nous l'avons vu lors de l'analyse des données, celui-ci est relativement petit (122 documents divisés sur deux langues) et ne contient que les labels d'une seule dimension taxonomique. Certains labels ne présentent que deux ou trois documents et tous les documents sont monolabellisés. De plus, comme le mandant ne fait pas partie du métier, nous ne pouvons affirmer avec certitude que la classification réalisée est correcte. Toutefois, nous n'avions de choix que de conduire l'étude avec ce corpus. Le besoin, quant à lui, s'adresse à toutes les dimensions de la taxonomie. L'Hôpital du Valais ayant à disposition un large set de données non labélisé, l'apprentissage semi-supervisé semblait initialement une bonne approche. Cependant, nous pensons qu'il est encore trop tôt pour se tourner vers une telle solution. En effet, la qualité et la taille du corpus prélabélisé doivent être suffisantes pour entraîner un premier classificateur avant de déterminer les pseudo-labels des documents non labélisés. Dans notre contexte, aucune de ces deux contraintes n'est satisfaite. Nous pensons ainsi que la solution proposée, basée sur l'analyse sémantique et la cooccurrence soit la meilleure approche compte tenu des circonstances. Les résultats sur nos modèles de prédictions divergent quelque peu de la littérature scientifique. Celle-ci affirme en effet que le modèle LSI performe généralement mieux que le modèle Word2Vec lorsque le corpus d'entraînement est petit. Nous constatons dans notre étude les résultats inverses. Notre système de cooccurrence entre concepts et topiques lié à notre évaluation du modèle LSI retourne systématiquement trop de labels. Cela signifie que les topiques qu'il découvre sont communs à la plupart des documents, peu importe leurs labels. Il est toutefois possible qu'en ajoutant une méthode de sélection des caractéristiques avant la représentation en BOW cela améliore les résultats. Le taux de précision résultant de nos modèles est malheureusement relativement faible. Nous pensons que le nombre de labels à prédire pour la dimension "Type de document" est trop important compte tenu des documents à disposition pour l'apprentissage. Les documents étant monolabélisés, le classificateur n'a qu'une chance sur douze d'obtenir le 100% de précision sur chaque document. Nous soupçonnons également que les liens de corrélations sémantiques entre concepts soient à l'origine du manque de précision. En effet, basé sur notre solution d'analyse sémantique, des concepts trop fortement corrélés à travers plusieurs labels peuvent biaiser la classification. Dans le cadre de notre prototype, nous proposons toutefois un système d'apprentissage continu pour pallier au problème sur le long terme. Nous espérons qu'au fur et à mesure de l'apprentissage de notre modèle Word2Vec avec de nouveaux documents, sa précision augmente.

Notre seconde hypothèse (H_2) suppose que les noms des fichiers liés aux documents peuvent impacter leurs labélisations. Nous en avons conclu lors de notre phase d'analyse que cela est effectivement le cas. Les documents qui émanent de certains départements semblent être soumis

à des conventions de nommages, d'autres pas. Lorsque les conventions sont appliquées, celles-ci ne sont pas homogènes. Cela nous laisse penser que chaque département ou unité à sa propre stratégie quant aux conventions de nommages de leurs documents. Dans certains cas, nous sommes cependant parvenus à soutirer de l'information au sein de ces noms de fichiers. Cette hypothèse est donc validée. Comme la méthode utilisée pour soutirer ces informations est similaire que celle relative au contenu textuel, nous ne faisons qu'ajouter le texte des noms de fichiers au contenu et exécuter notre analyse de cooccurrence. Il pourrait être intéressant dans le futur, de déterminer si un concept trouvé au sein d'un nom de fichier à un plus grand impact sur la labélisation du document que s'il est trouvé au sein du contenu. Le nom de fichier étant parfois un condensé du contenu, nous pourrions en effet naturellement penser qu'il porte plus de poids sur la prédiction.

Nous supposions avec notre hypothèse (H_3) que les métadonnées des documents pouvaient impacter leurs labélisations. D'après nos analyses, il semblerait que cela soit effectivement le cas. Nous avions découvert une corrélation entre leurs tailles et leurs labels. Leurs nombres de mots respectifs semblaient également jouer un rôle au sein de l'équation. Le risque en analysant un corpus aussi petit est cependant que notre échantillon ne représente pas correctement l'intégralité des documents de l'intranet. Pour limiter ce risque, nous avions essayé de tirer profit de notre corpus non labélisé en le fusionnant avec les résultats de notre corpus labélisé. Bien que moins évidents, certains clusters semblaient rester isolés. Nous pouvons donc considérer que cette hypothèse soit validée bien qu'évidemment, le niveau de l'impact dépend de la nature du label comme pour les deux précédentes hypothèses.

Finalement, en ce qui concerne la validation de notre hypothèse H_4, nous avions planifié la réalisation d'une étude quantitative sous forme de sondage pour déterminer si les profils des utilisateurs ont un impact sur la labélisation des documents. La mise en place de cette source de données primaire a malheureusement été infructueuse dans le cadre de cette étude. Nous étions supposés tirer profit de la publication à l'interne d'un sondage entrepris par le mandant. Celui-ci, relatif à la l'élicitation des besoins liés à la refonte de l'intranet, devait initialement nous servir de support pour y ajouter les questions spécifiques à notre étude. Cependant, la décision d'un niveau hiérarchique supérieur a empêché la modification du sondage en cours. Ce qui a rendu la validation de cette dernière hypothèse impossible dans le cadre de cette étude.

5 Conclusion

Dans le cadre de cette thèse de Master, nous nous sommes intéressés à la prédiction des labels relatifs aux documents de l'intranet de l'Hôpital du Valais. Nous avons ainsi commencé par réaliser une étude qualitative sous forme d'interviews dans un contexte de récolte des besoins. L'élément principal qui en est ressorti est le besoin de tenir compte d'une taxonomie définie à l'interne dans le cadre des prédictions. Cette taxonomie, multidimensionnel est fortement susceptible d'évoluer dans le temps. Également, l'Hôpital du Valais étant divisé en huit sites d'origines linguistiques variés. La solution proposée doit tenir compte de la nature multilinguiste des documents qui composent leur intranet. S'en est suivie une revue de la littérature quant à la classification textuelle multilabile. Nous avons extrait de celle-ci, un processus standard de classification textuelle pour lequel les meilleures approches furent étudiées pour chaque étape. Nous en avons alors conclu qu'une analyse sémantique basée sur le principe de cooccurrence était la meilleure approche compte tenu de la faible quantité de documents prélabélisés à disposition. Pour ce faire, les modèles de prédictions d'indexation sémantique latente et Word2Vec ont été retenus. Notre évaluation de ces deux modèles démontre un *Hamming Loss* trois fois supérieur sur l'indexation sémantique latente (60,8% contre 18,3% pour Word2Vec). Le niveau de précision est cependant similaire pour les deux modèles (respectivement 14,3% et 16,2%). Nous avons finalement entrepris le développement d'un prototype fonctionnel basé sur la cooccurrence sémantique en utilisant le modèle Word2Vec. Notre étude démontre que nos hypothèses H_1, H_2 et H_3 sont valides. Soit, que le contenu textuel, les noms de fichier et les métadonnées des documents peuvent impacter leurs labélisations. Notre hypothèse H_4 qui supposait que les profils des utilisateurs ont un impact sur la labélisation des documents fut abandonnée en cours d'étude pour cause de conflit hiérarchique.

Après la mise en production de la solution proposée, il serait intéressant de réévaluer nos modèles Word2Vec de manière période. Cela pour déterminer si l'apprentissage continue à un impact positif sur les prédictions. Également, tester la solution sur toutes les dimensions taxonomiques permettrait d'améliorer notre compréhension des données et si nécessaire, d'adapter notre méthodologie de résolution. Finalement, lorsque l'Hôpital du Valais aura amassé suffisamment de documents labélisés par les utilisateurs, une nouvelle étude pourrait être conduite. Celle-ci pourrait évaluer les performances de solutions de classification multilabel semi-supervisée ou supervisée pour déterminer si leurs résultats sont meilleurs que la solution proposée.

6 Références

[1] R. Wirth et J. Hipp, « CRISP-DM : Towards a Standard Process Model for Data Mining », *Proc. Fourth Int. Conf. Pract. Appl. Knowl. Discov. Data Min.*, nº 24959, p. 29-39, 1995.

[2] G. Piatetsky-Shapiro, U. Fayyad, et P. Smith, « From data mining to knowledge discovery: An overview », *Adv. Knowl. Discov. data Min.*, vol. 1, p. 35, 1996.

[3] G. Tsoumakas et I. Katakis, « Multi-Label Classification: An Overview », *Dept. Informatics, Aristotle Univ. Thessaloniki*, vol. 48, nº 6, p. 1-10, 2007.

[4] R. Jindal, R. Malhotra, et A. Jain, « Techniques for text classification: Literature review and current trends », *Webology*, vol. 12, nº 2, p. 1-28, 2015.

[5] C. N. Mahender et V. Korde, « Text Classification and classifiers: a Survey », *Int. J. Artif. Intell. Appl.*, vol. 3, nº 2, p. 85-99, 2012.

[6] N. Singh et M. Devi, « Document representation techniques and their effect on the document Clustering and Classification: A Review », *Int. J. Adv. Res. Comput. Sci.*, vol. 8, nº 5, p. 1780-1784, 2017.

[7] W. J. Wilbur et K. Sirotkin, « The automatic identification of stop words », *J. Inf. Sci.*, vol. 18, nº 1, p. 45-55, 1992.

[8] K. Tuomo *et al.*, « Stemming and lemmatization in the clustering of finnish text documents », *Conf. Inf. Knowl. Manag.*, 2004.

[9] L. van der Maaten, E. Postma, et J. van den Herik, « Dimensionality reduction: A comparative review », *Tilbg. Univ. Cent. Creat. Comput.*, 2009.

[10] W. Wang et M. Á. Carreira-Perpiñán, « The role of dimensionality reduction in linear classification », nº 2, 2014.

[11] D. M. Hawkins, « The Problem of Overfitting », *J. Chem. Inf. Comput. Sci.*, nº 44, p. 1-12, 2004.

[12] J. Tang, S. Alelyani, et L. Huan, « Feature Selection for Classification: A Review », *Data Classif. Algorithms Appl.*, p. 511-536, 2014.

[13] D. D. Lewis, « An Evaluation of Phrasal and Clustered Representations Task on a Text Categorization », *Proc. SIGIR 1992, 15th ACM Int. Conf. Res. Dev. Inf. Retr.*, p. 37-50, 1992.

[14] Y. Yang et J. O. Pedersen, « A comparative Study on Feature Selection in Text Categorization », *ICML*, 1997.

[15] M. Pechenizkiy, A. Tsymbal, et S. Puuronen, « PCA-based feature transformation for classification: issues in medical diagnostics », nº June, p. 535-540, 2004.

[16] I. T. Jolliffe, « Principal Component Analysis », *Springer*, 2002.

[17] G. Salton, A. Wong, et C. S. Yang, « Vector Space Model for Automatic Indexing. Information Retrieval and Language Processing », *Commun. ACM*, vol. 18, nº 11, p. 613-620, 1975.

[18] J. Ramos, « Using TF-IDF to dertermine word relevance in document queries », *Zeitschrift für Anorg. und Allg. Chemie*, vol. 164, p. 45-56, 2003.

[19] R. E. Atani et M. Yamaghani, « The Significance of Normalization Factor of Documents to Enhance the Quality of Search in Information Retrieval Systems », vol. 2, p. 91-97, 2014.

[20] Fabrizio Sebastiani, « Machine learning in automated text categorization », *ACM Comput. Surv.*, vol. 34, nº 1, p. 1–47, 2002.

[21] D.-H. Lee, « Pseudo-labeling: efficient and simple semi-supervised learning method for Deep NN », *ICML 2013 Work. Challenges Represent. Learn.*, 2013.

[22] S. Zhong, « Semi-supervised model-based document clustering: A comparative study », *Mach. Learn.*, vol. 65, nº 1, p. 3-29, 2006.

[23] J. Read, « Advances in Multi-label Classification ». 2011.

[24] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, et A. Bahamonde, « Binary relevance efficacy for multilabel classification », *Prog. Artif. Intell.*, vol. 1, nº 4, p. 303-313, 2012.

[25] M. Zhang et K. Zhang, « Multi-Label Learning by Exploiting Label Dependency », *Int. Conf. Knowl. Discov. Data Min.*, 2010.

[26] N. Spolaôr, E. A. Cherman, M. C. Monard, et H. D. Lee, « A comparison of multi-label feature selection methods using the problem transformation approach », *Electron. Notes Theor. Comput. Sci.*, vol. 292, p. 135-151, 2013.

[27] M. Sorower, « A literature survey on algorithms for multi-label learning », *Oregon State Univ. Corvallis*, p. 1-25, 2010.

[28] P. Soucy et G. W. Mineau, « A simple KNN algorithm for text categorization », *Proc. - IEEE Int. Conf. Data Mining, ICDM*, p. 647-648, 2001.

[29] E. K. Garcia, S. Feldman, M. R. Gupta, et S. Srivastava, « Completely lazy learning », *IEEE Trans. Knowl. Data Eng.*, vol. 22, nº 9, p. 1274-1285, 2010.

[30] T. Zhang, J. Wu, et H. Hu, « Text Classification Based on Novel Ensemble Multi-Label Learning Method », *Int. Conf. Syst. Informatics*, 2014.

[31] R. E. Schapire et Y. Singer, « BoosTexter: A boosting-based system for text categorization », *J. High Energy Phys.*, vol. 39, nº 11, p. 299-312, 2000.

[32] A. Esuli, T. Fagni, et F. Sebastiani, « Boosting multi-label hierarchical text categorization », *Inf. Retr. Boston.*, vol. 11, nº 4, p. 287-313, 2008.

[33] M. Zhang et Z. Zhou, « Ml-knn: A Lazy Learning Approach to Multi-Label Learning », *Natl. Lab. Nov. Softw. Technol.*, 2007.

[34] J. Weston et A. Elisseeff, « A kernel method for multi-labelled classification », *Adv. Neural Inf. Process. Syst. 14*, 2018.

[35] M. Drauschke, « Multi-class ADTboost », *Tech. Rep. Nr. 6, Dep. Photogramm. Inst. Geod. Geoinf. Univ. Bonn*, nº 6, 2008.

[36] D. Buscaldi, « Annotation de documents en utilisant l'Information Mutuelle Algorithmes d'annotation », 2016.

[37] E. Altszyler, M. Sigman, S. Ribeiro, et D. F. Slezak, « Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database », p. 1-14, 2016.

[38] M. Baroni et G. Dinu, « Baroni - Don't Count, Predict! », p. 238-247, 2014.

[39] O. Levy et Y. Goldberg, « Neural word embedding as implicit matrix factorization », *Adv. Neural Inf. Process. Syst.*, vol. 3, nº January, p. 2177-2185, 2014.

[40] O. Levy, Y. Goldberg, et I. Dagan, « Improving Distributional Similarity with Lessons Learned from Word Embeddings », *Trans. Assoc. Comput. Linguist.*, vol. 3, p. 211-225, 2015.

[41] S. Deerwester, G. W. Furnas, T. K. Landauer, et R. Harshman, « Indexing by Latent Semantic Analysis », *Kehidupan*, vol. 3, nº 12, p. 34, 2015.

[42] Y. Goldberg et O. Levy, « word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method », nº 2, p. 1-5, 2014.

[43] T. Mikolov, K. Chen, G. Corrado, et J. Dean, « Efficient Estimation of Word Representations in Vector Space », p. 1-12, 2013.

[44] J. Read, « Multi-label Classification - Course Material », 2013.

[45] G. Tsoumakas, I. Katakis, et I. Vlahavas, « Mining Multi-label Data », *Data Min. Knowl. Discov. Handb.*, p. 667-685, 2009.

[46] M. Pushpa et S. Karpagavalli, « Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification », *Procedia Comput. Sci.*, vol. 115, p. 572-579, 2017.

[47] Y. Yang, « An evaluation of statistical approaches to text categorization », *Inf. Retr. Boston.*, vol. 1, nº 1-2, p. 69-90, 1999.

[48] T. Hamon, « Indexation automatique de notices bibliographiques à l ' aide d ' approches d ' acquisition terminologique Matériel », vol. d, nº 1, 2016.

[49] D. Economics, « 16 Edition - State of the Developer », 2018.

[50] S. Marsland, *Machine Learning*. 2014.

[51] G. M. Foody, A. Mathur, C. Sanchez-Hernandez, et D. S. Boyd, « Training set size requirements for the classification of a specific class », *Remote Sens. Environ.*, vol. 104, nº 1, p. 1-14, 2006.

[52] T. Team, « Introducing TensorFlow Feature Columns ». [En ligne]. Disponible sur: https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html.

[53] S. Godbole et S. Sarawagi, « Discriminative methods for multi-labeled classification », *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3056, p. 22-30, 2004.

[54] V. Nastase, M. Strube, B. Börschinger, C. Zirn, et A. Elghafari, « WikiNet: A very large scale multi-lingual concept network », *Proc. 7th Int. Conf. Lang. Resour. Eval. Lr. 2010*, nº 2007, p. 1015-1022, 2010.

7 Annexes

7.1 Annexe 1 – Guide d'entretien de l'étude qualitative de récolte des besoins

Méthodologie

Cette interview est basée sur la méthode des 5V.

Introduction du contexte

Dans le cadre de ma thèse de Master, je cherche à réaliser cette interview dans le but d'apporter de la profondeur à la problématique pour comprendre son contexte ainsi que ses facteurs environnementaux. C'est un interview de récolte des besoins assez classique. Je vais ainsi vous poser des questions relatives au projet de refonte. Cette interview sera enregistrée à titre d'utilisation académique uniquement.

Présentation de la personne interviewée

- Pourriez-vous vous présenter en deux mots, dire depuis combien de temps vous travaillez à l'Hôpital du Valais et quel est ton rôle ?

Pourquoi

- Pourriez-vous décrire le/les processus actuels de partage(s) de documents au sein de l'intranet ?
- Quelles sont les difficultés/problèmes que vous rencontrez avec ces processus ?
- Quels sont les impacts de ces difficultés/problèmes sur le travail des collaborateurs ?

Quoi

- Quelles sont les solutions envisagées pour résoudre ces problèmes/difficultés ?

Qui

- Pourriez-vous me parler des différentes parties prenantes associées au projet ? Qui travaille sur le projet à l'interne/externe, qui sont les partenaires et qui seront les utilisateurs finaux ?

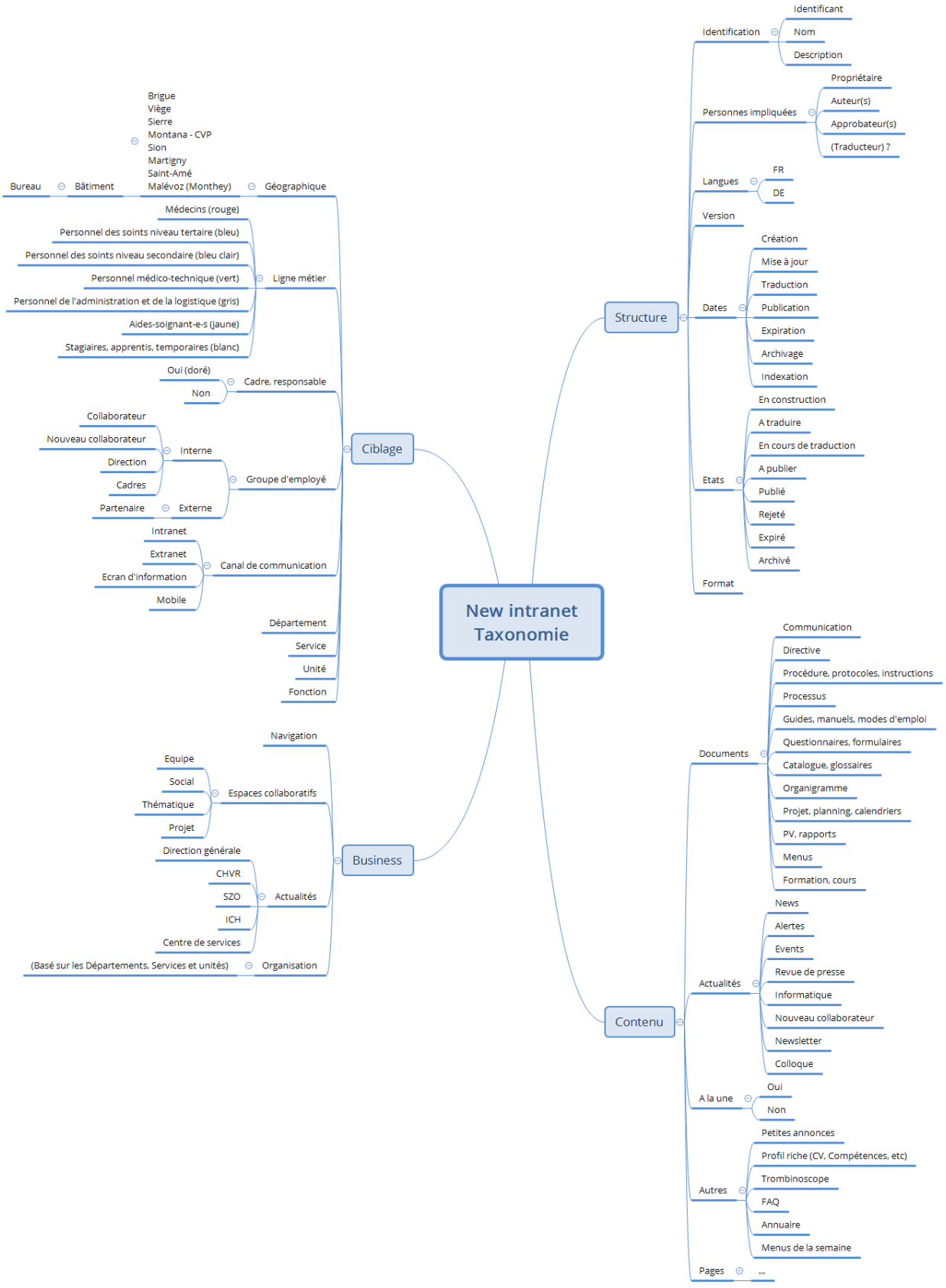
Quand

- Depuis combien de temps avez-vous entrepris la refonte de l'intranet ? Où en est-elle ?
- Avez-vous un délai de mise en production du nouvel intranet ? Et qu'en est-il de la partie relative à ma thèse ?
- Quelles sont vos attentes quant à l'algorithme d'intelligence artificielle dans le cadre de ma thèse ?

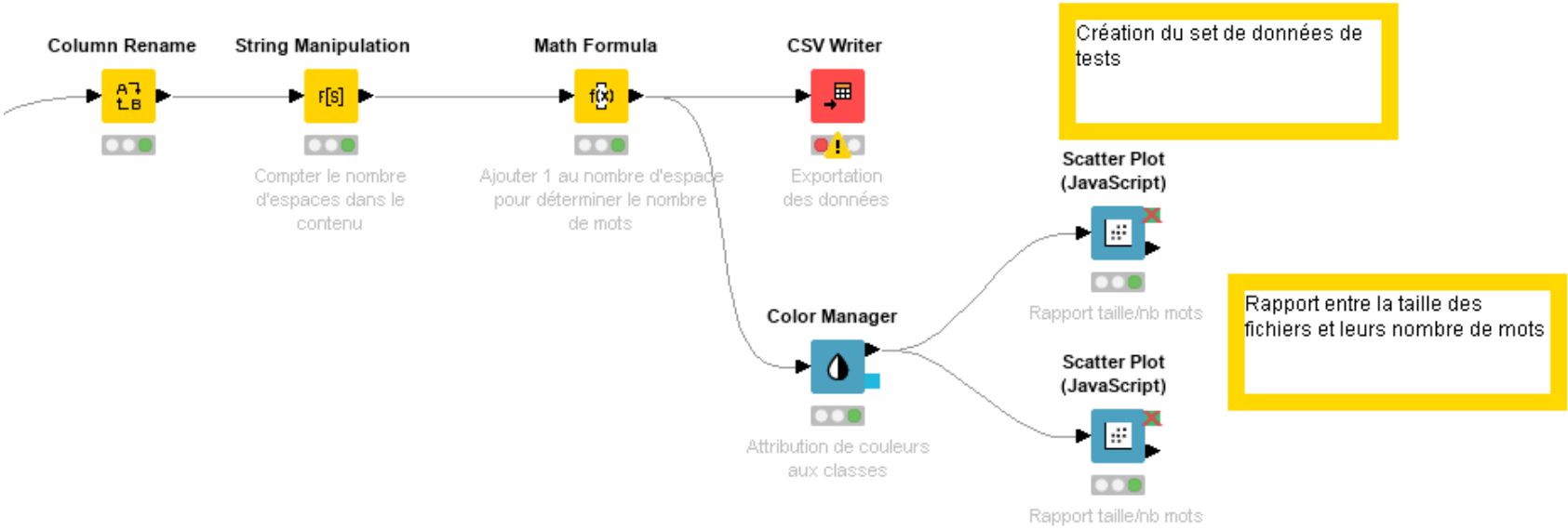
Où

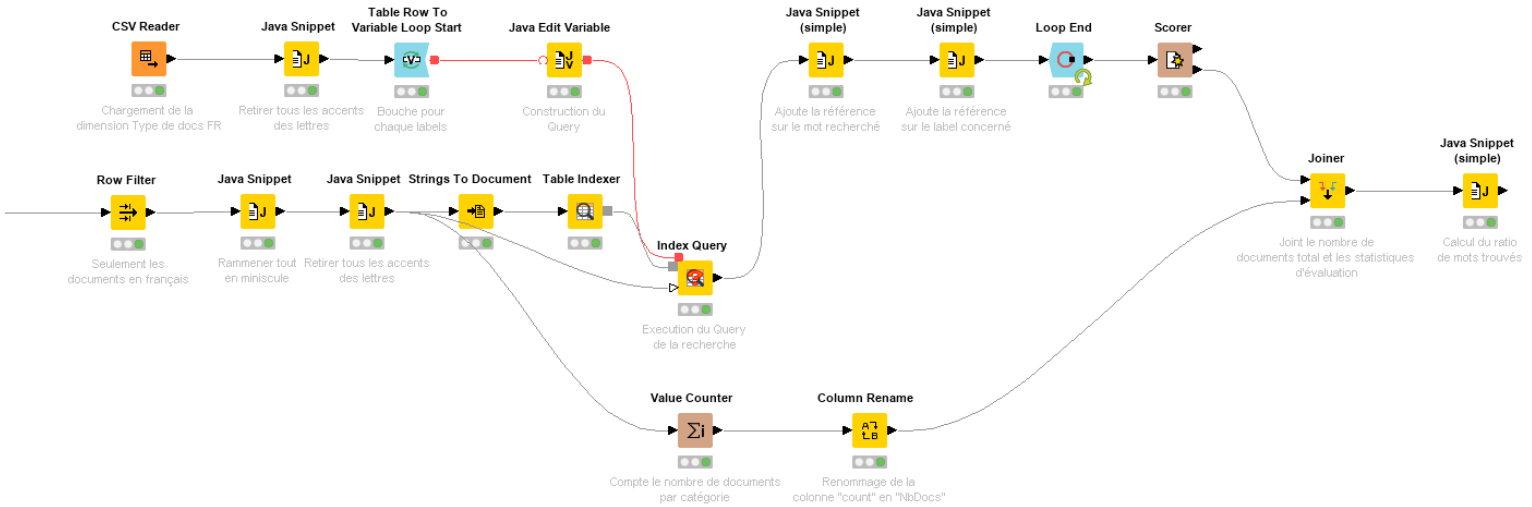
- L'Hôpital du Valais est divisé en huit établissements répartis dans huit villes du canton du Valais. Quelle est l'infrastructure informatique derrière l'hébergement de l'intranet ? Où sont les serveurs et comment est-ce qu'ils opèrent ?
- Avez-vous des préférences quant au moyen de déploiement de mon algorithme ? Containerisation ?

7.2 Annexe 2 – Taxonomie de l'intranet

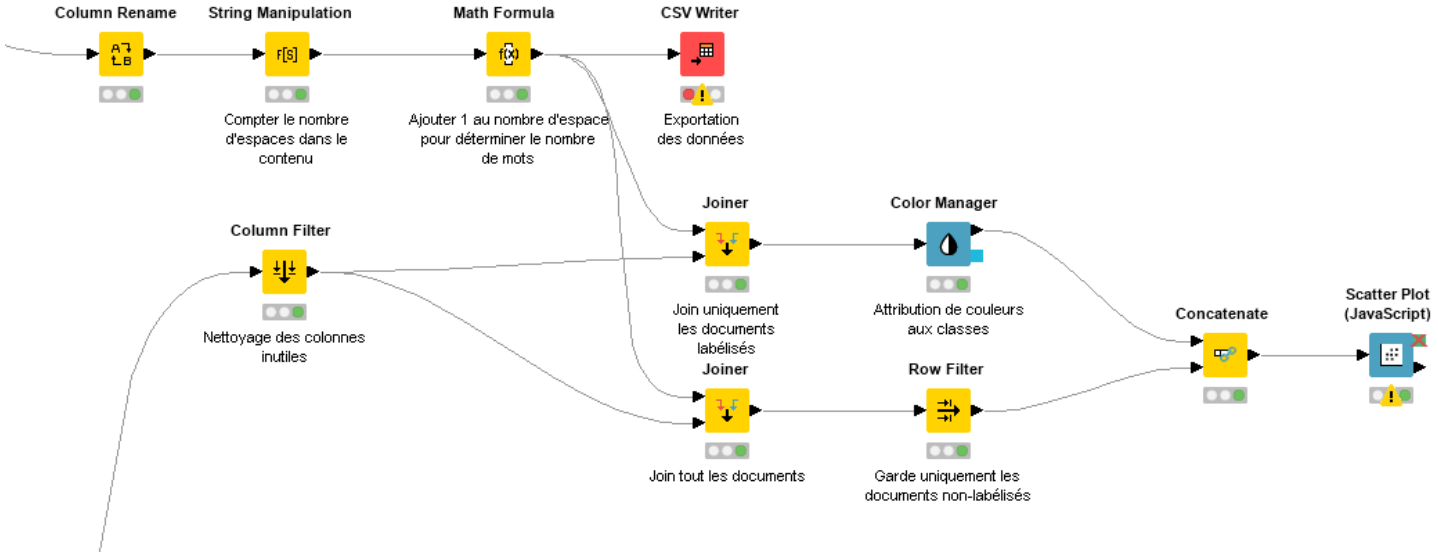


7.3 Annexe 3 – Flux Knime





Proportion de mots composant les labelles trouvés dans le contenu des documents associés en FRANCAIS



Rapport entre la taille des fichiers et leurs nombre de mots sur l'intégralité du corpus. Mise en évidence des données

Attestation

Je déclare sur l'honneur, que j'ai effectué ce Travail de Master seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du Responsable de l'Orientation et du Professeur chargé du suivi du Travail de Master et de l'institution ou entreprise pour laquelle ce travail a été effectué.

XXXX, February 15, 2017.

Maxime Beck