Classification multinomiale et supervisée de comptes rendus d'incidents

Master Thesis

Alina Ana-Maria PETRESCU

Université de Fribourg - Universität Freiburg
Faculté des sciences et de médecine

Sous la supervision de
Dr. Prof. Philippe CUDRÉ-MAUROUX, UNIFR

En collaboration avec l'entreprise
TECHWAN, Renens

16 avril 2019

Résumé

Mots-clé : *classification discrète multinomiale supervisée, BM25, Doc2Vec, Logistic Regression, Python*

Dans cette thèse, on s'intéresse à la classification discrète multinomiale supervisée des documents. Trois modèles sont proposés et testés grâce aux données mises à notre disposition par l'entreprise *TechWan* qui utilise actuellement une suite logicielle pour la gestion opérationnelle et la gestion de crises (Chapitre 1).

Dans ce contexte, les documents sont des comptes rendus ou exposés des faits décrivant des incidents et leur classification doit être faite en fonction d'un catalogue de catégories d'intervention bien précises. Le but recherché est de prédire la ou les catégories appropriées pour tout nouveau compte rendu à classifier.

L'implémentation des solutions envisagées est faite à l'aide du langage Python et de plusieurs librairies Python spécialisées dans le traitement du langage naturel.

Dans le Chapitre 2, une attention particulière est accordée au traitement préalable (pré-processing) des données dans le but de corriger les erreurs d'introduction de données. De plus, les 421 catégories (associées à environ 170'000 exposés des faits) sont brièvement présentées.

Dans le Chapitre 3, on propose un premier modèle basé sur la fonction d'ordonnancement *BM25*. Des définitions ad-hoc par catégorie sont élaborées et la catégorie proposée pour un nouvel exposé des faits est celle qui obtient le plus élevé score de similitude avec cet exposé.

Dans le Chapitre 4, on présente une deuxième approche basée sur le modèle *Doc2Vec* et ses deux versions *DBOW* et *DM*. Pour chaque version, on crée des modèles qui prennent en compte soit les 37 *catégories principales* (avec plus de 1000 exposés des faits par catégorie), soit 8 *super-catégories* regroupant des *catégories principales* apparentées. De plus, chaque tel modèle a deux variantes de labellisation de documents : par catégorie ou avec identifiants uniques. Une fois entraînés, ces modèles sont capables de vectoriser un nouvel exposé des faits et de lui associer une ou plusieurs catégories appropriées (en fonction de son degré de similitude avec les catégories ou les exposés des faits déjà labellisés).

Dans le Chapitre 5, on ajoute une troisième approche basée sur le modèle *LogisticRegression*. En fait, chaque modèle *Doc2Vec* déjà entraîné infère les vecteurs numériques correspondant aux documents à classifier qui constituent ensuite l'ensemble d'entraînement d'un modèle *LogisticRegression* associé. Entraîné à son tour, chaque modèle *LogisticRegression* réalise la classification multinomiale basée sur la prédiction des probabilités d'appartenance d'un nouvel exposé des faits (vectorisé) à chaque catégorie.

Finalement, dans le Chapitre 6, on compare les modèles proposés. Sans être aussi puissant que les deux autres, le modèle *BM25* a l'avantage qu'il peut être utilisé même pour des catégories avec un nombre "modeste" d'exposés des faits associés. Plus exacts, basés sur l'apprentissage automatique, les modèles *Doc2Vec* et *LogisticRegression* ont besoin d'un nombre important de données par catégorie, d'où la prise en compte seulement des *catégories principales* ou des *super-catégories*. Afin d'assouplir cette exigence, on peut imaginer une *classification en cascade* : les exposés des faits sont d'abord classifiés selon des super-catégories, ensuite selon des catégories principales dans le cadre d'une même super-catégorie et ainsi de suite. En outre, on énonce des recommandations concernant la rédaction des exposés des faits, la formation des opérateurs et la réorganisation de la taxonomie des exposés. Finalement, on présente des suites possibles du travail à la fois pour l'amélioration de la classification de documents et pour l'intégration des solutions proposées dans l'outil d'aide à l'engagement de *TechWan*.

Table des matières

1 Introduction

Selon la définition du ***Larousse***, les incidents sont des faits "de caractère secondaire" qui se produisent durant une action qui sera ainsi perturbée. Sans avoir une importance initiale excessive, les incidents peuvent générer des complications et des effets plutôt dommageables.

Par conséquent, la capacité d'identifier et de classifier de manière appropriée un incident est importante. D'un côté, elle permet de diminuer et, si possible, d'éviter les conséquences négatives et les dysfonctionnements produits par l'incident. D'un autre côté, elle permet de mieux comprendre la situation globale associée à l'incident (surtout si celui-ci se produit dans le cadre d'un événement majeur ou d'une crise). Donc, la classification adéquate des incidents par rapport à une taxonomie convenable devient essentielle pour la gestion efficace des incidents même et de l'environnement qui leur est associé.

D'une manière plus générale, développée au départ à cause du besoin de classification scientifique des espèces du monde vivant, la taxonomie est aujourd'hui utile, voire nécessaire, aussi pour d'autres sciences. En particulier, l'informatique a besoin et utilise des méthodes de classification d'informations afin de mettre en place des architectures basées sur des entités conceptuelles correspondant à des groupes hiérarchisés appelés taxons.

En revenant à la taxonomie des incidents, il convient de reconnaître qu'une classification à la fois détaillée et efficace est difficile à réaliser. Ainsi, l'optimisation de la classification peut avoir des conséquences bénéfiques pour la gestion d'événements car elle permet non seulement une compréhension rapide et juste de ce qui se passe mais aussi le choix prompt et adapté des moyens d'intervention pour le contrôle de la situation en cours.

Or, la société **TechWan** qui sera présentée ci-dessous a justement besoin d'optimiser la classification des incidents qu'elle gère à l'aide d'un produit logiciel spécialisé réalisé par elle-même.

1.1 La problématique

Aujourd'hui, à l'heure de l'apprentissage automatique (*Machine Learning*), de l'intelligence artificielle (AI) et des bases de données distribuées et massives (*Big Data*), la classification de documents est beaucoup étudiée, en général, et dans le domaine du traitement automatique du langage naturel (*Natural Language Processing - NLP*), en particulier.

Dans ce contexte, la classification des incidents en vue de la gestion opérationnelle et de crise peut être vue comme un cas particulier de classification discrète multinomiale supervisée de documents.

1.2 TechWan

Depuis 2000, la société **TechWan** a développé une suite logicielle appelée *SAGA Command & Control* qui est utilisée par des services publics (des polices cantonales et municipales, des pompiers, des services d'ambulance, des mairies, etc.) et des société privées (des aéroports, des festivals, etc.) impliqués dans la gestion opérationnelle et la gestion de crise (accidents, incendies, etc.). Le but du logiciel est de permettre à ces clients une intervention rapide des unités compétentes en cas d'incident.

Ainsi, le logiciel *SAGA* doit accompagner à la fois l'opérateur humain qui reçoit, à un numéro spécial, un appel concernant un incident produit, et les acteurs impliqués dans l'intervention proprement dite.

Grâce à ce logiciel et en fonction des spécificités du client, l'identité et la localisation de l'appelant sont obtenues automatiquement depuis la compagnie téléphonique. Ensuite, l'opérateur utilise le logiciel afin de rédiger un ***compte rendu d'incidents***, appelé ***exposé des faits***, sur la base des informations fournies par l'appelant.

Dans ce contexte, le compte-rendu sur un incident produit doit :

- être succinct mais assurer une description exacte, si possible exhaustive, de l'incident ;
- mentionner les détails pertinents dans la perspective où il sera la base des mesures à prendre pour régler l'incident ;
- éviter les ambiguïtés et utiliser des mots-clés et des abréviations établies à l'avance afin de faciliter le traitement rapide de son contenu, en vue de la préparation et de l'envoi automatiques des messages vers les intervenants à agir ;
- permettre le regroupement des informations au niveau "global" en cas de plusieurs appels concernant le même incident ou des incidents reliés.

Une fois ce rapport de synthèse rédigé, l'opérateur lui associe une ***catégorie*** appropriée (faisant partie d'un catalogue clairement défini) et utilise le logiciel *SAGA* afin d'annoncer (de dispatcher) les unités les mieux placées (par compétence, par disposition géographique, etc.) pour intervenir et gérer l'incident. Finalement, ces unités d'intervention emploient à leur tour ce même logiciel avant, durant et après l'engagement sur le terrain.

1.3 Étude de cas : SAGA, le logiciel d'aide à l'engagement

À présent, l'opérateur, un professionnel chevronné, avec des connaissances spécifiques assez poussées, utilise *SAGA* comme aide à l'engagement pour lui permettre d'accomplir sa tâche. Afin d'améliorer le travail de l'opérateur, il faudrait créer un nouveau service "d'aide à la décision" et l'intégrer dans la suite logicielle *SAGA* afin qu'il puisse accompagner l'opérateur dans le choix de la catégorie (voire des catégories) appropriée(s) pour l'incident produit et décrit dans l'exposé des faits. De plus, grâce à ce service, des personnes n'ayant pas forcément une expérience préalable liée aux services d'interventions pourraient être formées rapidement pour jouer le rôle de l'opérateur.

Dans cette logique, une taxonomie bien réfléchie des incidents est essentielle. Le service à ajouter doit ainsi repenser et restructurer les catégories d'incidents utilisées actuellement dans le logiciel *SAGA*.

Le principal défi du service est de proposer, en temps réel et de manière fiable, une liste restreinte de catégories convenables, voire la catégorie la plus appropriée, en fonction des données récoltées et stockées dans chaque nouvel exposé des faits. Le choix final sera normalement décidé ou seulement confirmé par l'opérateur humain.

1.4 La démarche

Sans trop d'hésitation, on a choisi comme langage de programmation pour l'implémentation des solutions proposées le langage Python [1] car il :

- est libre (free) et indépendant de la plateforme d'exécution (cross-platform) ;
- offre des structures de données élémentaires (comme les listes, sets et dictionnaires) ou élaborées (comme les DataFrame et Series de la librairie pandas [2]) appropriées à notre problème ;
- offre de nombreuses librairies optimisées (pour le calcul parallèle et concurrent) qui évitent de "réinventer la roue" ("batteries included") et qui permettent la manipulation rapide, efficace et fiable de grandes quantités de données ;
- est largement utilisé pour des problèmes de traitement automatique du langage naturel (Natural Language Processing - NLP) et d'apprentissage automatique (Machine Learning - ML).

Dans la suite du rapport, une méthode ou une fonction ***standard*** est une méthode ou une fonction qui existe dans une librairie Python et qui est prête à l'emploi, tandis qu'une méthode ***ad-hoc*** est une méthode écrite dans le cadre de cette thèse afin de résoudre un problème spécifique.

Par contre, pour la réalisation effective de la classification *discrète* (avec les catégories utilisées comme classes) *multinomiale* (avec bien plus de deux catégories différentes) *supervisée* (grâce aux exposés des faits déjà classifiés par l'opérateur humain) de documents, on a envisagé trois approches. Chaque fois, le but recherché est de prédire la ou les bonnes catégories pour tout nouvel exposé des faits (en se basant sur des exposés des faits labellisés au préalable).

La première approche exploite les fréquences d'apparition des mots dans les exposés de faits et s'appuie sur la fonction d'ordonnancement (ranking function) *BM25*.

Pour la deuxième approche, on a opté pour l'utilisation de la vectorisation (embedding) de documents. Le choix effectif a été assez difficile car il y a plusieurs de ces technologies : *bag-of-words*, *n-grams*, *Word2Vec*, *Doc2Vec*, etc. (voir Section 4.1). Finalement, on a préféré le modèle *Doc2Vec* qui est orienté documents et qui est basé sur le modèle *Word2Vec* (qui est un peu plus ancien et se concentre, de son côté, sur les mots individuels).

La troisième approche profite de la vectorisation réalisée par le modèle *Doc2Vec* et implémente la *régression logistique* afin de classifier les exposés des faits.

On a séparé le travail effectif en deux parties :

- le *traitement initial* (*preprocessing*) dont le but est de fournir des données de départ (contenant des exposés des faits déjà classifiés et labellisés) correctes (voir Chapitre 2) ;
- le *traitement à proprement dit* (*processing*) dont le but est d'exploiter les données de départ, en tant que données d'apprentissage et données de test, afin de mettre en place les algorithmes de prédiction ; cette partie représente le noyau de la thèse et elle conduit à trois solutions pour la classification des exposés des faits basées, comme mentionné ci-dessus, sur la fonction de classification *BM25* (voir Chapitre 3), le modèle *Doc2Vec* (voir Chapitre 4) et, respectivement, le modèle *LogisticRegression* (voir Chapitre 5).

Les solutions proposées, y compris les variantes d'une même solution, sont présentées et implémentées dans des fichiers sources correspondants à des carnets *Jupyter Notebooks* (voir Annexe A).

Il convient de mentionner que le code est abondamment commenté et pour cette raison, dans ce rapport, on ne donne pas de détails trop techniques concernant la programmation proprement dite.

2
Les données

Dans ce chapitre, on va présenter des informations concernant à la fois les données utilisées comme entrées (input) et les données calculées (output) par les programmes de classification supervisée proposés.

2.1 Données brutes et traitement préliminaire (preprocessing)

D'une manière générale, les *données de départ* sont essentielles pour la réussite du travail de classification car même un algorithme d'apprentissage automatique idéal conduirait à des résultats erronés s'il est entraîné sur des données erronées.

De plus, les données doivent être abondantes (si possible du genre Big Data) afin de permettre l'apprentissage et l'optimisation convenables. Il est vrai que des données massives impliquent des temps de traitement importants mais ce travail est fait en amont et non pas en temps réel durant le processus de décision.

En outre, les données doivent être mises à jour, corrigées et complétées en permanence.

Pour notre cas d'étude concret, TechWan a mis à notre disposition plus de 170'000 enregistrements, c'est-à-dire des couples formés des exposés des faits et les catégories qui leur ont été associées par un opérateur humain.

Le travail de traitement préliminaire a été fastidieux, il a pris un temps conséquent et il a été réalisé dans un Notebook Python à part (voir fichier 1 dans l'Annexe A).

Le but de ce pré-traitement a été d'assurer un ensemble de départ contenant des exposés des faits correctement labellisés.

Concrètement, il a fallu corriger (voire éliminer) environ 10'000 enregistrements contenant des erreurs de différentes natures, par exemple :

- des enregistrements incomplets représentés par :
    - des exposés des faits sans catégorie ;

    - des catégories sans exposés des faits ;
- des inversions entre la colonne de l'exposé des faits et la colonne de la catégorie ;
- un même exposé des faits étalé sur plusieurs enregistrements ;
- des essais pour lesquels l'exposé des faits et/ou la catégorie des enregistrements sont formés du seul mot "test".

Dans la mesure du possible, on a essayé d'automatiser la découverte et la correction des erreurs. Ainsi, pour la suite du travail, on a gardé environ 158'000 couples exposés des faits-catégories associées.

2.1.1 Taxonomie des exposés des faits - les catégories

Les catégories représentent l'aspect clé de toutes les approches présentées par la suite car elles définissent les classes qui serviront comme étiquettes (labels) dans la classification discrète multinomiale supervisée.

À première vue, la taxonomie utilisée actuellement et basée sur plus de 400 catégories différentes (plus précisément 421) n'est pas idéale. En fait, même pour un opérateur humain chevronné, le choix de la "bonne" catégorie à attribuer à un certain exposé des faits n'est pas évident et l'opérateur aura tendance à proposer surtout les catégories qu'il aurait utilisées le plus souvent et dont il se rappellera le plus facilement. De plus, vu le nombre important de catégories, il y a dans les données à notre disposition des catégories avec peu d'exposés des faits associés. Cependant, toutes les catégories sont nécessaires car chacune d'entre elles correspond à une procédure en place bien spécifique.

Plus précisément, on a dans cette taxonomie des catégories *principales* avec des milliers d'exposés des faits, mais aussi des catégories "exotiques" avec à peine quelques exposés des faits associés.

Par exemple :

- il y a 37 catégories *principales* avec plus de 1000 exposés des faits chacune (voir Figure 2.1), parmi lesquelles 8 catégories avec plus de 5'000 exposés des faits (voir Tableau 2.1) ;
- dans l'autre sens, 69 catégories ont un seul exposé des faits associé (et regroupent donc seulement 69 exposés des faits) ;
- de même, 201 catégories ont moins de dix exposés des faits associés (et regroupent seulement 658 exposés des faits) ;
- les 37 catégories *principales* ont ensemble un peu plus de 122'000 exposés des faits associés, tandis que toutes les autres 384 catégories ont ensemble seulement environ 36'000 exposés des faits associés.

Si on garde à l'esprit le fait que les catégories sont les classes à associer aux nouveaux exposés des faits grâce au processus de classification automatique, il est important qu'elles soient clairement définies, en évitant les ambiguïtés et les superpositions. Dans ce sens, dans la taxonomie actuelle avec 421 catégories, il y a par exemple :

- 71 catégories dont les noms sont formés d'un seul mot ("ANIMAUX", "ARMES", "BRUIT", "DROGUE", "GAZ", "MEURTRE", "OVNI", "VIP", etc.) ;

	Catégories	Nombre exposés
1	DEMANDE - D'ASSISTANCE	17'968
2	TAPAGE NOCTURNE	7'182
3	ACCIDENT - DE CIRCULATION - DEGATS MATERIEL	7'091
4	ANIMAUX	6'847
5	VOL - PAR EFFRACTION	6'657
6	AR - DIVERS	5'617
7	INDIVIDU - SUSPECT	5'261
8	INFRACTION - LCR	5'141

TABLE 2.1 – Catégories avec plus de 5'000 exposés associés

- 350 catégories dont les noms sont formés de plusieurs mots, mais il y a seulement 154 premiers mots distincts.

La situation mentionnée au dernier point ci-dessus est due au fait que les noms composés des catégories suivent la logique d'une *classification de plus en plus précise*. Par exemple, il y a 26 catégories différentes dont les noms commencent par "ACCIDENT" parmi lesquelles un groupe de 5 catégories apparentées qui commencent par "ACCIDENT - DE CIRCULATION", à savoir :

- "ACCIDENT - DE CIRCULATION - DEGATS MATERIEL" - 7091 exposés associés ;
- "ACCIDENT - DE CIRCULATION - AVEC FUITE" - 2213 exposés associés ;
- "ACCIDENT - DE CIRCULATION - AVEC BLESSE" - 1960 exposés associés ;
- "ACCIDENT - DE CIRCULATION - AVEC ANIMAL" - 1714 exposés associés ;
- "ACCIDENT - DE CIRCULATION - MORTEL" - 47 exposés associés.

De même, on remarque une précision de plus en plus grande dans les catégories suivantes :

- "ACCIDENT" - 26 exposés associés ;
- "ACCIDENT - PERSONNE" - 2 exposés associés ;
- "ACCIDENT - PERSONNE - TRAIN" - 22 exposés associés.

Le revers de cette approche est qu'elle peut conduire à certaines catégories avec très peu d'exposés des faits associés. Dans l'exemple des catégories dont le nom commence par "ACCIDENT", il y a 7 telles catégories avec moins de 10 exposés des faits associés, à savoir : "ACCIDENT - DE TRAIN", "ACCIDENT - DE PLONGEE", "ACCIDENT - EN RIVIERE", "ACCIDENT - DE CHASSE", "ACCIDENT - DE PERSONNE", "ACCIDENT - AVEC DES EXPLOSIFS", "ACCIDENT - NOMBREUX BLESSES" et "ACCIDENT - DE TIR".

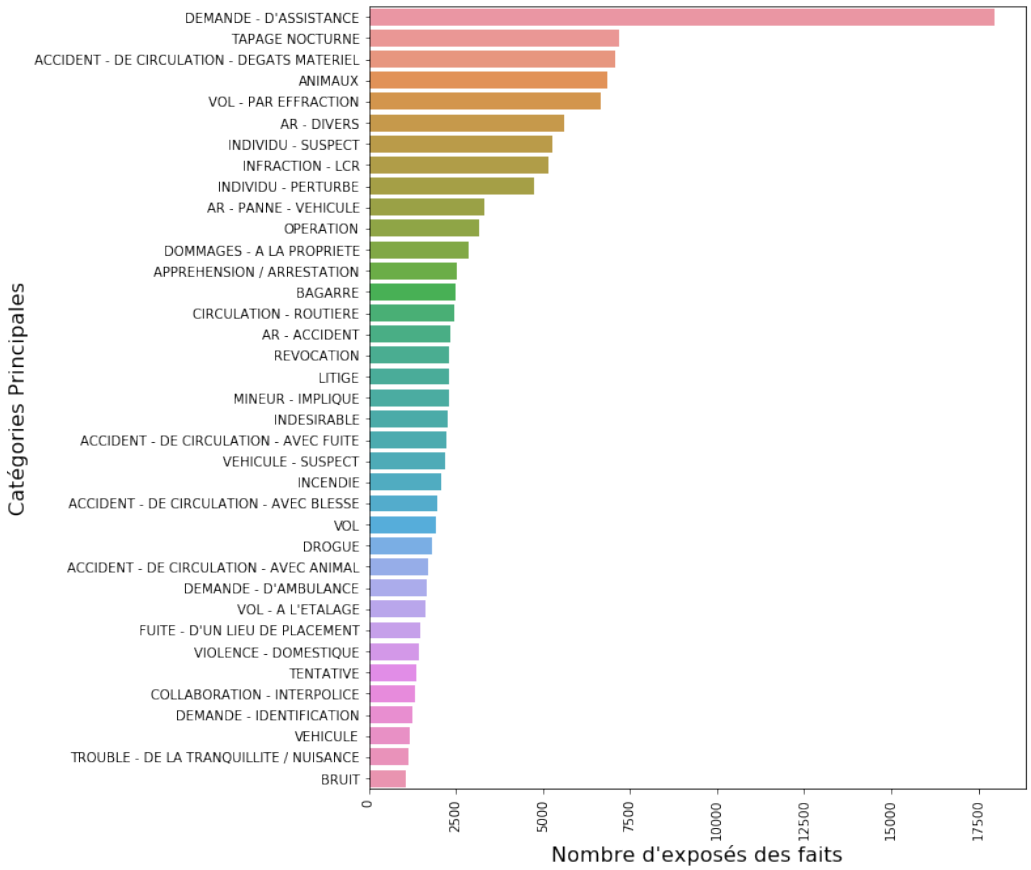

FIGURE 2.1 – Répartition des exposés des faits par catégories *principales*

Je pense qu'il faut réfléchir si on garde ces catégories distinctes ou si on les inclut, au moins dans un premier temps, dans la même catégorie "ACCIDENT" tout simplement.

Toujours dans cette perspective, il convient de mentionner que la taxonomie actuelle contient, par exemple :

- 100 catégories distinctes dont les noms commencent par "ALERTE" ;
- 25 catégorie distinctes dont les noms commencent par "PIEGE" ;
- 13 catégories distinctes dont les noms commencent par "OPERATION" ;
- 9 catégories distinctes dont les noms commencent par "VOL" ;
- 8 catégories distinctes dont les noms commencent, respectivement, par "AR", "CHUTE", "DISPARITION", "FUITE" ou "PANNE".

En outre, le choix de la bonne catégorie (par le programme ou par des opérateurs humains) n'est pas toujours univoque car plusieurs solutions peuvent s'avérer convenables, voire nécessaires.

En fait, il faut trouver un bon compromis entre le besoin de prendre en compte aussi les catégories avec un nombre réduit d'exposés des faits associés et la nécessité d'utiliser les catégories avec beaucoup d'exposés des faits associés afin de mettre en place un processus d'apprentissage automatique.

2.2 Données traitées

Il convient de mentionner que le traitement préliminaire n'a pas pu régler certains aspects qui peuvent influencer (parfois de manière défavorable) le traitement des données pré-processées, surtout dans la perspective de l'utilisation de modèles d'apprentissage automatique. Par exemple :

a. les tailles des exposés des faits sont bien différentes (à partir de 2 ou 3 mots, jusqu'à près de 300 mots) ;

b. le nombre d'exposés des faits par catégorie est aussi très variable (à partir de quelques exposés jusqu'à presque 20'000, ce qui donne un ensemble d'entraînement non équilibré) ;

c. les délimitations entre certaines catégories sont assez "perméables" ; il y a ainsi des intersections de catégories qui ne sont pas nulles, voire des relations d'inclusion entre certaines catégories, et ceci rend difficile et non univoque l'attribution des catégories pour les nouveaux exposés des faits à classifier ; par exemple, il y a dans la taxonomie actuelle une catégorie "ACCIDENT - DE NAVIGATION / NAUFRAGE" et une autre "ACCIDENT - DE NAVIGATION / NAUFRAGE - NAVIGATEUR EN DIFFICULTE" ;

d. un même exposé des faits peut apparaître plusieurs fois dans la base de données (parfois des dizaines de fois ou plus) et pas toujours avec la même catégorie associée (comme l'exposé "*2 voitures*" qui apparaît 105 fois avec comme catégorie "ACCIDENT - DE CIRCULATION - DEGATS MATERIEL" et "AR - ACCIDENT", mais aussi "ACCIDENT - DE CIRCULATION - AVEC BLESSE") ;

e. il y a des exposés des faits dont le contenu est trop court ou qui ne permettent pas d'en déduire la ou les catégories attribuées.

Concernant les points **d.** et **e.**, il convient d'ajouter quelques remarques :

- sur les quelques 158'000 exposés des faits retenus après le pré-processing, seulement environ 129'000 sont uniques (sans doublons) ;
- parmi les plus de 28'000 "doublons", on peut donner des exemples d'exposés des faits avec jusqu'à 5 catégories associées :
  - "*En cours de patrouille sont tombés sur un vhc étranger en panne. 4 personnes à bord et du matériel suspect dans le coffre. Personnes et vhc conduites au poste de douane*" s'est vu associer **5** catégories : "INFRACTION - LCR", "APPREHENSION / ARRESTATION", "INDIVIDU - SUSPECT", "VOL - A L'ETALAGE" et "AR - DIVERS" (sans que les deux dernières catégories soient vraiment déductibles à partir du contenu de l'exposé des faits) ;
  - "*son mari se bat avec un inconnu qui a essayé d'entrer dans la villa*" s'est vu associer **4** catégories : "INDIVIDU - SUSPECT", "BAGARRE", "APPREHENSION

/ ARRESTATION" et "DROGUE" (sans que les deux dernières catégories puissent être liées directement à l'exposé des faits) ;

- "*effraction dans sa cave. Attends sur place.*" s'est vu associer **3** catégories : "VOL - PAR EFFRACTION", "INDIVIDU - SUSPECT" et "VEHICULE" (sans que l'exposé des faits soit pertinent pour expliquer la dernière catégorie) ;
- "*une voiture en feu à l'extérieur des locaux*" s'est vu associer **2** catégories : "VEHICULE" et "INCENDIE" (et les deux catégories bien distinctes sont nécessaires ici pour caractériser correctement l'incident produit) ;
- "*direction Lausanne. Une femme enceinte et un homme qui saigne au visage.*" s'est vu associer **2** catégories : "ACCIDENT - DE CIRCULATION - AVEC BLESSE" et "ACCIDENT" (mais aucune indication claire dans l'exposé ne laisse penser aux deux catégories associées).

Les points **d.** et **e.** sont très sensibles pour l'apprentissage automatique qui essaie de trouver des liens pertinents entre des exposés des faits et leurs catégories. Voilà quelques exemples supplémentaires (à éviter dans le futur) :

- "*M. XYZ (07X XXX XX XX) signale un cornet noir contenant un bocal dans lequel se trouvent des sachets qui contiennent du cannabis. Tout ceci se trouve dans une haie située sur la gauche en entrant sur le parking.*" apparaît dans la catégorie "DROGUE" (ce qui est normal) mais aussi dans la catégorie "MILITAIRE IMPLIQUE" (sans savoir pourquoi) ;
- les exposés des faits "*Une voiture immobilisée sur la voie gauche.*" et "*1 voiture seule en cause sur voie gauche*" sont classifiés sous "AR – ACCIDENT" mais ces exposés ne laissent pas "deviner" l'accident et peuvent être facilement vus comme appartenant à la catégorie "AR - PANNE – VEHICULE" ;
- les exposés des faits "*Une voiture seule en cause, au milieu de la route. Route bloquée.*" ou "*Deux autos. Une voiture au milieu de la route.*" sont classifiés sous "ACCIDENT - DE CIRCULATION - DEGATS MATERIEL" mais ces exposés ne laissent "deviner" ni l'accident ni le dégât matériel, et pourraient être facilement vus comme appartenant à la catégorie "AR - PANNE - VEHICULE".

Une fois le processus de pré-traitement achevé, les données traitées ainsi obtenues et formées des enregistrements *exposés des faits-catégories associées* sont utilisées pour apprendre à un modèle de prédiction comment choisir la ou les catégories appropriées pour de nouveaux exposés des faits. Dans les trois chapitres suivants, on présente trois tels modèles pour la classification supervisée des exposés des faits.

3
Approche basée sur le modèle BM25

Dans ce chapitre, on présente une première approche de classification discrète multinomiale supervisée basée sur le modèle *BM25*. Son but est surtout de mettre en évidence les facteurs qui interviennent dans le choix de la catégorie appropriée pour un nouvel exposé des faits, par l'intermédiaire d'une solution simple et construite de manière intuitive.

3.1 Le modèle BM25

À l'origine, *Okapi BM25* (où *BM* provient de "Best Matching") est une fonction d'ordonnancement (ranking function) utilisée par des moteurs de recherche pour classifier les documents en fonction de leur pertinence par rapport à une demande de recherche [3].

Cette fonction est implémentée en Python dans la librairie de même nom *BM25* [4].

En bref, à partir d'un corpus qui contient tous les mots utilisés par les documents à classifier, la librairie *BM25* permet de calculer surtout :

- la fréquence (ou plutôt le nombre d'occurrences) ***tf*** d'un mot dans un document donné ainsi que la fréquence relative ;
- la fréquence ***df*** d'un mot dans le corpus (vue comme le nombre de documents contenant au moins une fois ce mot) ;
- la fréquence inverse ***idf*** d'un mot défini comme le logarithme naturel d'un quotient, à savoir :

$$idf = log \frac{N - df + 0.5}{df + 0.5}$$

ou du quotient :

$$idf = log\frac{N}{df+1}$$

où N est le nombre total de documents dans le corpus ;

- le score d'un mot défini comme le produit :

$$tf \cdot idf$$

(afin de tenir compte du "rôle" de ce mot dans le document ainsi que de son "rôle" dans le corpus) ;

- le score d'un certain document D par rapport à un autre document Q calculé avec la formule :

$$score(D,Q) = \sum_{i=1}^{n} idf(mot_i) \cdot \frac{tf(mot_i,D)\cdot(k_1+1)}{tf(mot_i,D)+k_1\cdot(1-b+b\cdot\frac{len(D)}{avgdl})}$$

où :

  - n est le nombre de mots du document Q ;
  - mot_i est un mot courant du document Q ;
  - $k_1 \in [1.2, 2.0]$ est un paramètre libre ;
  - $b = 0.75$ est un autre paramètre libre ;
  - $len(D)$ est le nombre de mots du document D ;
  - $avgdl$ est la taille moyenne des documents dans le corpus.

3.2 Utilisation du modèle BM25 pour l'étude de cas

L'idée de l'approche proposée est d'obtenir une sorte de définitions par catégorie qui serviront ensuite à placer un nouvel exposé des faits dans une catégorie appropriée.

Plus précisément, une telle définition de catégorie sera formée par les mots qui apparaissent "le plus souvent" dans les exposés des faits associés à cette catégorie. En fait, la fréquence ***tf*** calculée avec le modèle *BM25* met l'accent sur "l'importance" d'un mot dans les exposés des faits d'une même catégorie, tandis que la fréquence inverse ***idf*** tient compte si un même mot apparaît dans les exposés correspondant à plusieurs catégories, en pénalisant les mots "trop communs" car ils ne permettent pas de distinguer les catégories entre elles. Ainsi, si un mot apparaît dans plus de la moitié des exposés des faits associé à toutes les catégories, il aura une fréquence inverse ***idf*** négative.

Finalement, le score ***tf*idf*** tâche de prendre en compte simultanément les deux aspects mentionnés.

En outre, un point important qui se retrouvera aussi dans les autres modèles proposés par la suite concerne l'élimination des mots "sans pertinence" dans les définition retenues pour les catégories. Il s'agit, par exemple, de "mots vides" (stop words) comme les prépositions, les

articles ou les pronoms qu'ils sont si communs qu'ils ne valent pas la peine d'être pris en compte, ou de certains nombres et dates ou des mots mélangeant lettres et chiffres, etc.

Le découpage des exposés des faits en mots (tokens) a été réalisé grâce à la libraire Python `text.blob` [5] et les mots vides ont été obtenus par l'intermédiaire de la librairie Python `spaCy` [6].

En tenant compte des aspects mentionnés auparavant, on présente ci-dessous les deux étapes principales de la solution ad-hoc proposée (voir Figure 3.1).

A. Dans un premier temps, on détermine les définitions des catégories en procédant ainsi :

- on lit les données de départ pré-traitées (preprocessed) et stockées sur le disque (en format `.csv` ou `.xls`) par un objet *DataFrame* de la librairie Python *pandas* [2] ;
- on organise ces données sous la forme d'un dictionnaire qui a comme clés les catégories et comme valeurs des listes de strings regroupant les mots (sauf les mots "sans pertinence") qui correspondent aux exposés des faits associés à une même catégorie ;
- on crée le ***corpus*** qui est une liste de listes de strings où chaque liste interne est une liste de strings avec tous les mots correspondants à une certaine catégorie ;
- on crée un objet de type classe *BM25* associé au ***corpus*** défini ci-dessus ;
- en appelant des méthodes standards pour cet objet, on peut obtenir des informations comme :
  - la taille du ***corpus*** ;
  - la longueur moyenne (en mots) d'un document du ***corpus*** ;
  - la fréquence ***tf*** (term frequency) d'apparition de chaque mot dans les exposés des faits pour une catégorie donnée ;
  - la fréquence ***df*** (document frequency) d'apparition de chaque mot dans le ***corpus*** ;
  - la fréquence inverse ***idf*** (inverse document frequency) d'apparition de chaque mot dans le ***corpus*** ;
- on calcule ensuite le score ***tf*idf*** pour chaque mot dans les exposés des faits d'une catégorie donnée ;
- pour chaque catégorie, on ordonne les scores ***tf*idf*** des mots correspondants (sans doublons) en ordre décroissant et on garde comme définition de cette catégorie une liste avec les mots qui ont les scores les plus élevés.

Il est clair que le choix du nombre de mots les plus fréquents retenus pour une catégorie donnée est arbitraire et il faut tester des valeurs différentes en tenant compte aussi des nombres totaux de mots par catégorie.

B. Dans un deuxième temps, on détermine la (ou les) catégorie(s) appropriée(s) pour un nouvel exposé des faits à classifier, en procédant ainsi :

- comme à l'étape **A**, on crée un objet de type classe *BM25* mais on lui associe un ***corpus*** formé avec les définitions des catégories obtenues à l'étape **A** (et il aura autant de documents que de catégories prises en compte) ;

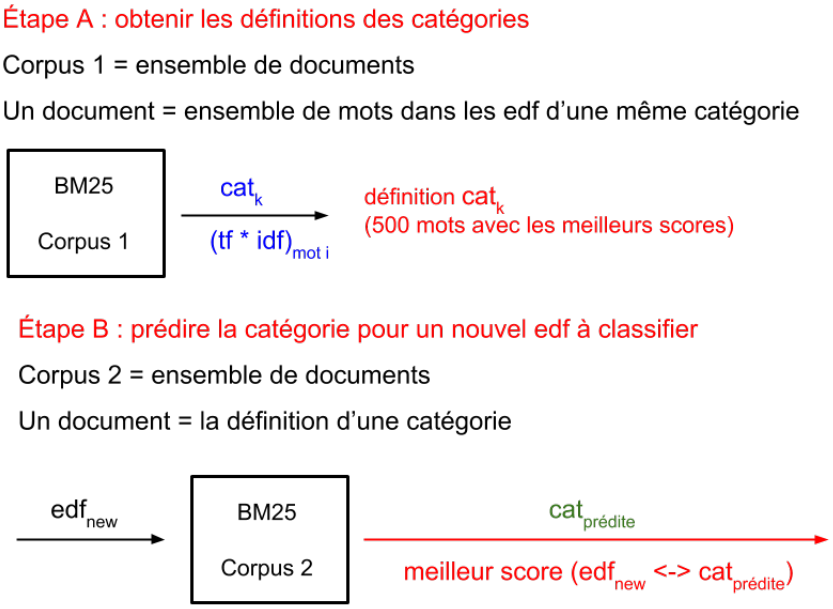

FIGURE 3.1 – Modèle BM25

- en appelant la méthode standard `get_bm25_weights` avec le ***corpus*** comme argument, on obtient les scores (les poids) de chaque document (en fait de chaque catégorie) par rapport à tous les autres documents (en fait toutes les autres catégories) ; ces scores de similitude nous permettent de comparer les catégories et de trouver les catégories qui sont éventuellement proches, donc difficilement différentiables (voir le Tableau 3.2) ;
- à partir d'un nouvel exposé des faits, on calcule la liste des mots pertinents qui lui sont associés (grâce à une méthode ad-hoc `clean`) ;
- on calcule ensuite les scores de similitude d'un nouvel exposé des faits par rapport aux définitions des catégories (grâce à la méthode standard `get_scores`) ;
- finalement, on propose comme catégorie celle (ou celles) ayant obtenu le meilleur score de similitude (en indiquant, éventuellement, aussi les catégories avec les scores suivants les plus élevés).

3.3 Résultats obtenus avec le modèle BM25

Afin de tester la solution proposée, on a mis d'abord au point une procédure pour calculer la *justesse* (appelée aussi exactitude ou accuracy) du modèle implémenté.

D'une manière générale, la justesse est largement utilisée pour évaluer et comparer des modèles de classification, et elle est définie de la manière suivante :

$$
\begin{aligned}
justesse &= \frac{\text{Nombre de prédictions justes}}{\text{Nombre total de prédictions}} \\
&= \frac{\text{Nombre de prédictions justes}}{\text{Nombre de prédictions justes + Nombre de prédictions fausses}} \\
&\leq 1
\end{aligned}
\tag{3.1}
$$

Dans notre cas, on a procédé ainsi :

- l'ensemble de données formé par les exposés des faits déjà labellisés avec la catégorie attribuée par l'opérateur humain, a été partagé de manière *aléatoire* en deux sous-ensembles (grâce à la méthode strandard `train_test_split` de la librairie Python `sklearn_model_selection`) :
  - un sous-ensemble d'entraînement (formé, par exemple, de 90% de données) ;
  - un sous-ensemble de test (formé, par exemple, du reste de 10% de données) ;
- les définitions des catégories ont été déterminées conformément au point **A.** ci-dessus et à partir du sous-ensemble d'entraînement ;
- pour chaque exposé des faits faisant partie du sous-ensemble de test, on a inféré la ou les bonnes catégories en suivant la démarche présentée au point **B.** ci-dessus ;
- les prédictions faites ainsi ont été comparées avec les catégories associées aux exposés des faits du sous-ensemble de test, ce qui a permis d'obtenir le nombre de prédictions justes et, respectivement fausses ;
- avec ces dernières valeurs ainsi obtenues, on a calculé la justesse du modèle selon la formule présentée auparavant.

La démarche indiquée ci-dessus est assez générale et elle est synthétisée dans la Figure 3.2 (qui reste valable aussi pour les modèles *Doc2Vec* et *LogisticRegression* qui seront introduits dans les deux chapitres suivants).

On peut souligner deux aspects intéressants :

- même si les catégories analysées ont des nombres d'exposés des faits associés très différents, les sous-ensembles d'entraînement et de test sont obtenus aléatoirement mais en respectant les proportions de l'ensemble total ;
- il y a des exposés des faits pour lesquels le modèle prédit plusieurs catégories dans le cas où ces catégories obtiennent le même score maximal.

L'intérêt de la solution basée *directement* sur le modèle *BM25* est double :

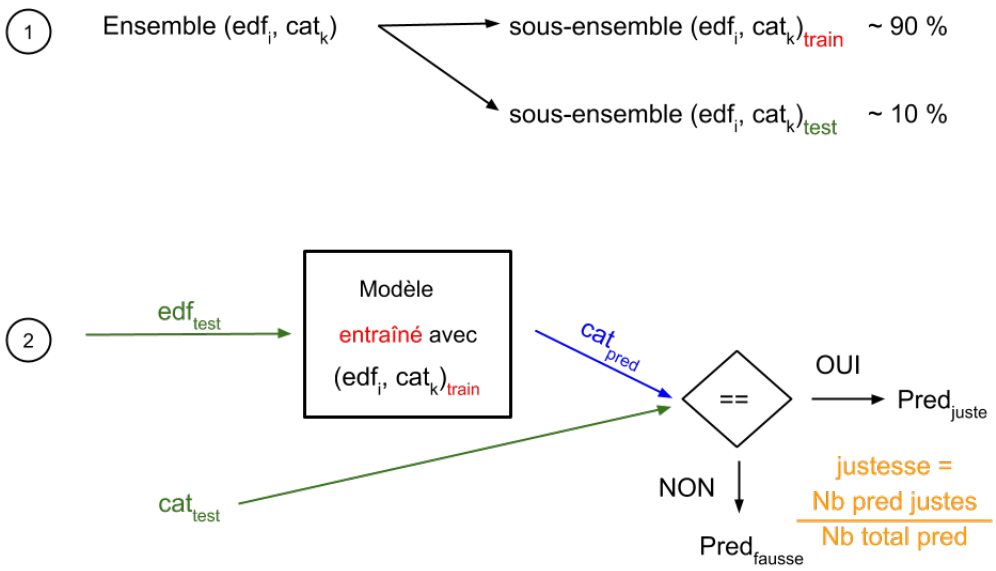

FIGURE 3.2 – La justesse d'un modèle

- d'un côté, l'exploitation directe des fréquences d'apparition des termes dans des documents individuels (chez nous des exposés des faits) et dans le vocabulaire global permet une meilleure compréhension du problème en vue de l'implémentation de solutions plus sophistiquées ;
- de l'autre côté, elle offre une solution simple pour la prédiction de la catégorie appropriée pour un nouvel exposé des faits dans le cas où les données à disposition sont trop "pauvres" pour permettre l'utilisation de méthodes d'apprentissage automatique.

Vu cette dernière remarque, on a testé la solution *BM25* surtout pour des catégories avec un nombre relativement réduit d'exposés des faits associés (donc pour des cas opposés à ceux qui seront pris en compte dans les deux chapitres suivants).

Il faut reconnaître, dès le début, que notre modèle est mis en défaut dans les cas des catégories *fortement apparentées* et/ou dans une *relation d'inclusion*. On a ainsi considéré le cas limite de deux catégories : "SIGNALISATION - ROUTIERE" et "SIGNALISATION ROUTIERE - FOURNITURE - PANNE" (voir Figure 3.3), pour lesquelles le modèle *BM25* a conduit à une surprenante justesse de 100%.

Ce résultat est dû au fait que, pour chaque exposé des faits du sous-ensemble de test, le modèle a prédit systématiquement les deux catégories car elles obtenaient le même score maximal. Plus précisément, le modèle ne pouvait pas distinguer entre les deux catégories dont les exposés des faits utilisaient presque les mêmes mots. Cette situation est illustrée dans le Tableau 3.1 qui montre les scores de similitude réciproques (évidemment négatifs) extrêmement proches pour les deux catégories.

Par contre, notre modèle *BM25* est très fiable si les catégories prises en compte sont bien distinctes. Pour justifier cette affirmation, on donne comme exemple un test effectué avec 5

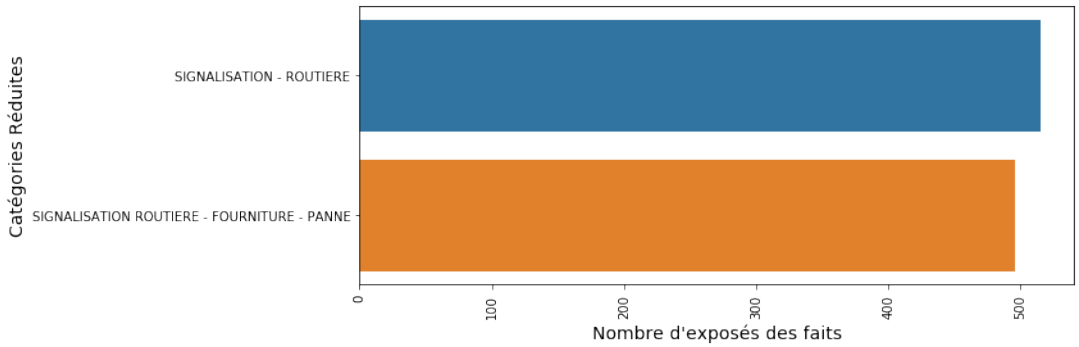

FIGURE 3.3 – Répartition des exposés des faits par catégories - BM25 - Cas 1

	SIGNALISATION - ROUTIERE	SIGNALISATION ROUTIERE - FOURNITURE - PANNE
SIGNALISATION - ROUTIERE	-331.1	-315.8
SIGNALISATION ROUTIERE - FOURNITURE - PANNE	-341.5	-362.7

TABLE 3.1 – Similitudes catégorie-catégories - BM25 - Cas 1

catégories, chacune avec moins de 1'000 exposés des faits associés (voir Figure 3.4).

Plus précisément, on a gardé la catégorie "SIGNALISATION - ROUTIERE" (515 exposés des faits et identifiant Cat_1 dans le Tableau 3.2) à laquelle on a ajouté les catégories : "MOEURS" (536 exposés des faits et identifiant Cat_2 dans le Tableau 3.2), "POLLUTION - HYDROCARBURES" (719 exposés des faits et identifiant Cat_3 dans le Tableau 3.2), "INDIVIDU - RECHERCHE" (869 exposés des faits identifiant et Cat_4 dans le Tableau 3.2) et "INONDATION" (476 exposés des faits et identifiant Cat_5 dans le Tableau 3.2).

Pour la définition de chaque catégorie, on a gardé les 500 mots les plus significatifs selon leur score ***tf*idf***.

Afin de s'assurer que les définitions ainsi obtenues sont bien distinctes, on a calculé leurs scores de similitudes réciproques (voir Tableau 3.2) qui sont tous positifs.

Par exemple, la catégorie "MOEURS" (qui a l'identifiant Cat_2) a le score maximal par rapport à elle-même, le score suivant relativement significatif par rapport à la catégorie "INDIVIDU - RECHERCHE" et des scores négligeables par rapport aux trois autres catégories, ce qui est tout à fait normal vu les similarités "naturelles" entre les cinq catégories.

Finalement, pour ce deuxième cas d'étude, notre modèle *BM25* a prédit les catégories appropriées pour les exposés des faits du sous-ensemble de test avec une justesse de **94.5%**. Plus précisément, il y a eu 273 prédictions justes et 19 prédictions fausses.

Il convient de mentionner que, malgré la non concordance avec les catégories associées par

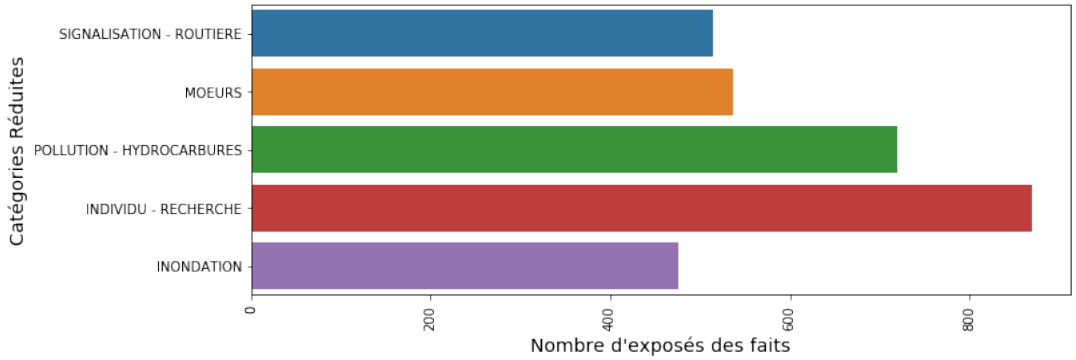

FIGURE 3.4 – Répartition des exposés des faits par catégories - BM25 - Cas 2

	Cat_1	**Cat_2**	**Cat_3**	**Cat_4**	**Cat_5**
Cat_1	2323.8	323.1	570.5	326	376.5
Cat_2	624.1	6091	848.2	2194.3	823.2
Cat_3	713.7	580.2	4893.2	620.7	954
Cat_4	822.5	2267.4	1031.2	7155.5	928
Cat_5	344.5	465.9	779	468.4	2096.1

TABLE 3.2 – Similitudes catégorie-catégories - BM25 - Cas 2

l'opérateur humain pour un même exposé des faits, certaines prédictions faites par le modèle sont aussi plausibles, surtout si on regarde les mots-clé soulignés ci-dessous. Voilà quelques exemples :

- l'exposé des faits "*Villa en feu. Pompiers roulent.*" a été labellisé avec "INONDATION", mais le modèle a prédit "POLLUTION - HYDROCARBURES" ;
- l'exposé des faits "*Mme a été retrouvée apparemment DCD dans son lit. Est arrivé à l'Hôtel le 18.07 et a payé en avance. Chambre 210.*" a été labellisé avec "INDIVIDU - RECHERCHE", mais le modèle a prédit "MOEURS" ;
- l'exposé des faits "*Probablement rupture d'une canalisation d'eau, à la hauteur des pompes à essence ! Beaucoup d'eau sur site, ainsi que sur la chaussée !*" a été labellisé avec "INONDATION", mais le modèle a prédit "POLLUTION - HYDROCARBURES".

Pour conclure, suite à nos tests, on peut considérer que la méthode proposée dans ce chapitre donne des résultats honnêtes (voire très bons si les catégories sont bien distinctes), et elle *reste envisageable surtout si le nombre de données disponibles par catégorie est relativement réduit*. Par contre, on ne s'attardera plus sur cette méthode car on lui a préféré les classes des méthodes présentées dans les deux chapitres suivants qui sont plus précises et mieux adaptées pour l'apprentissage automatique.

4
Modélisation avec Doc2Vec

Dans ce chapitre, on présente une modélisation du problème de la recherche de la classe (dans notre cas de la catégorie) appropriée pour un nouveau document (dans notre cas un nouvel exposé des faits) basée sur l'encapsulation et la vectorisation (embedding) des documents avec *Doc2Vec*.

Par rapport à la solution proposée dans le chapitre précédent, il s'agit cette fois d'un problème d'apprentissage automatique nécessitant un nombre important de données d'entraînement.

4.1 Le modèle Doc2Vec

Conformément aux [7] [8] [9], *Doc2Vec* est un outil de traitement du langage naturel (NLP) spécialement conçu pour l'encapsulation et la vectorisation des documents.

Il est basé sur le modèle *Word2Vec* et est implémenté en Python dans la librairie `gensim.models.doc2vec` [10].

Avant *Word2Vec*, il y avait deux approches principales pour représenter les documents comme des vecteurs numériques par rapport à un vocabulaire commun de base.

Le premier modèle *bag-of-words* représente chaque document par un vecteur de longueur fixe où les composantes du vecteur sont les fréquences d'apparition de chaque mot dans le document correspondant. Malgré son efficacité, ce modèle ne prend pas en compte l'ordre des mots dans les documents.

Afin de palier à cet inconvénient, le deuxième modèle, appelé *n-grams*, représente aussi chaque document par un vecteur mais de longueur fixe plus grande. Les composantes de ces vecteurs sont cette fois les indices (les identifiants uniques) des mots dans l'ordre de leur apparition dans les documents. Le prix à payer pour cette information supplémentaire est que les vecteurs ainsi obtenus ont de tailles plus élevée (haute dimensionnalité) et ils contiennent beaucoup de zéros (données creuses).

Le modèle *Word2Vec* utilise les idées évoquées ci-dessus mais il est focalisé sur les mots individuels. Son but est de calculer des *vecteurs de mots* de sorte que, si de tels vecteurs sont proches dans l'espace vectoriel correspondant, alors les mots qui ont servi à leur génération ont aussi des significations proches. À l'envers, des mots dont les significations sont bien différentes seront représentées par des vecteurs aussi éloignés dans l'espace vectoriel.

La vectorisation des mots avec *Word2Vec* conduit cette fois à un espace vectoriel de dimension réduite grâce à l'utilisation d'un réseau neuronal superficiel (shallow neural network) formé seulement de trois couches : la couche d'entrée (input layer), la couche interne ou cachée (hidden layer) et la couche de sortie (output layer). En fait, la dimension des vecteurs de mots est donnée par le nombre de neurones utilisés dans la couche cachée.

Word2Vec est implémenté en deux versions :

- le modèle *SG* (*Skip-Gram*) ;
- le modèle *CBOW* (*Continuous-bag-of-words*).

Vu que notre objectif est la classification des documents, plus précisément des exposés des faits selon leur catégorie, on a préféré une approche orientée documents et non pas orientée mots individuels. Par conséquent, on a choisi le modèle *Doc2Vec* qui est basé sur *Word2Vec* mais qui réalise la vectorisation des documents (ou paragraphes) entiers. Ainsi, dans notre cas, pour chaque exposé des faits, le modèle *Doc2Vec* calcule un vecteur numérique correspondant.

À nouveau, l'approche *Doc2Vec* est aussi implémentée en deux versions :

- le modèle *DBOW* (*Distributed Bag of Words*) qui correspond au modèle *SG* (*Skip-Gram*) de l'approche *Word2Vec* ;
- le modèle *DM* (*Distributed Memory*) qui correspond au modèle *CBOW* (*Continuous-bag-of-words*) de l'approche *Word2Vec*.

Avec *DBOW*, les vecteurs correspondant aux documents sont obtenus en entraînant un réseau neuronal dont la tâche est de prédire la distribution de probabilité des mots dans un document (paragraphe) basé sur un échantillonnage aléatoire des mots du paragraphe.

L'algorithme *DM* se comporte comme une mémoire qui se souvient du sujet du document vectorisé.

4.2 Utilisation du modèle Doc2Vec pour l'étude de cas

D'une manière générale, pour chaque variante d'implémentation du modèle *Doc2Vec* présentée par la suite, on a structuré le travail dans les phases suivantes :

a. *réalisation* du modèle :

- *création* du modèle abstrait (en précisant des méta-paramètres et, tout particulièrement, le choix entre les algorithmes *DBOW* et *DM*) ;
- *concrétisation* du modèle abstrait en lui associant un vocabulaire (grâce à la méthode standard `build_vocab`);
- *entraînement* du modèle pour son vocabulaire (à l'aide de la méthode standard `train` et pour un certain nombre d'époques).

b. *vectorisation* de nouveaux exposés des faits ;

c. *exploitation* du modèle.

a. Dans la première phase, la *concrétisation* du modèle abstrait suppose la création du vocabulaire obtenu par la transformation de chaque document, i.e. de chaque exposé des faits, en un objet de type `TaggedDocument`. Plus précisément, chaque exposé des faits est décomposé en une liste de mots (il est tokenisé) et on lui associe un identifiant (un tag) unique ou pas. À cette occasion, comme expliqué dans le dernier chapitre, certains mots sans pertinence sont éliminés et les traitements impliqués ont été regroupés dans une fonction ad-hoc `clean`.

Cette première phase de l'implémentation du modèle *Doc2Vec* est essentielle car, en fonction des choix opérés, plusieurs variantes ont été étudiées par la suite :

- d'un côté, des modèles distincts basés sur les algorithmes *DBOW* et *DM* ont été réalisés et comparés comme performances ;
- de l'autre côté, deux stratégies pour la labellisation (l'étiquetage ou le tagging) des documents (durant la création d'objets `TaggedDocument`) ont été proposés : avec *identifiant par catégorie* ou avec *identifiant unique par exposé des faits*.

Concernant le dernier aspect, il convient de rappeler que nos données d'entraînement contiennent des exposés des faits qui peuvent apparaître plusieurs fois et, éventuellement, avec des catégories associées différentes. Par conséquent, la labellisation des exposés des faits avec des ***identifiants selon les catégories*** semble la plus naturelle (malgré le fait que l'utilisation initiale de l'approche *Doc2Vec* employait des identifiants uniques par document). Dans cette variante, le modèle entraîné contient autant de vecteurs numériques que de catégories distinctes et tous les exposés des faits appartenant à une même catégorie contribuent au calcul du vecteur numérique correspondant. Ainsi, un exposé des faits qui apparaît plusieurs fois avec la même catégorie renforce simplement sa prise en compte dans le calcul du vecteur numérique correspondant à cette catégorie. De plus, le fait qu'un même exposé des faits puisse avoir plusieurs catégories associées a (probablement) comme effet d'approcher, d'une certaine façon, les orientations des vecteurs numériques calculés pour ces catégories.

Cependant, on a aussi proposé des modèles avec des ***identifiants uniques par exposé des faits***. Bien évidemment, on labellise seulement les exposés des faits sans doublons, tout en gardant une liste des catégories associées pour chaque document apparaissant plusieurs fois dans les données d'entraînement mais avec des catégories associées différentes. Par conséquent, le modèle entraîné contient autant de vecteurs numériques que d'exposés des faits sans doublons. Dans cette variante, on respecte l'esprit de départ de l'approche *Doc2Vec* et, comme on le montre plus loin, on pourra proposer plus d'une catégorie convenable pour un nouvel exposé des faits à classifier.

b. Dans la deuxième phase de l'implémentation *Doc2Vec*, le modèle entraîné est utilisé pour la *vectorisation* de nouveaux exposés des faits qui ne font pas partie du vocabulaire et pour lesquels on calcule des vecteurs numériques représentatifs. Cette opération est implémentée par l'intermédiaire d'une fonction ad-hoc `edf2vec` qui s'appuie sur la méthode standard `infer_vector` mais qui réalise aussi d'autres tâches, y compris la transformation d'un nouvel exposé des faits en un objet `TaggedDocument`.

c. Dans la troisième phase de l'implémentation *Doc2Vec*, l'*exploitation* du modèle entraîné permet de mettre en évidence les similarités entre, d'un côté, ses propres vecteurs numériques et, de l'autre côté :

- soit certains de ses propres vecteurs numériques (correspondant au vocabulaire même) ;
- soit de nouveaux vecteurs numériques correspondant à de nouveaux documents (n'appartenant pas au vocabulaire).

Plus précisément, pour les modèles utilisant des ***identifiants par catégorie*** (voir Figure 4.1), on peut ainsi :

- trouver les catégories du modèle les plus similaires avec une catégorie donnée du modèle - grâce à la fonction ad-hoc `show_similar_cats_inner` (qui utilise la méthode standard `most_similar`) ;
- proposer les catégories du modèle les plus appropriées pour un nouvel exposé des faits à classifier et qui ne fait pas partie du vocabulaire - grâce à la fonction ad-hoc `show_proposed_cats` (qui utilise également, mais dans un autre contexte, la méthode standard `most_similar`).

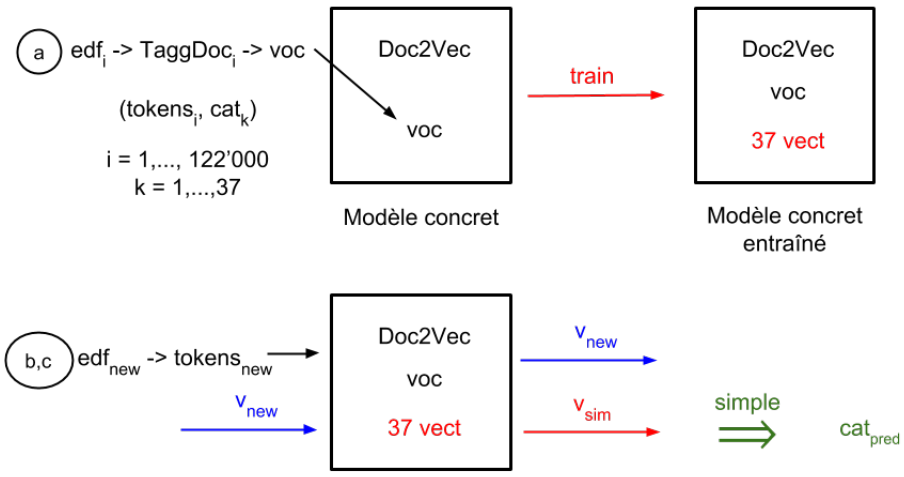

FIGURE 4.1 – Modèle Doc2Vec (DBOW ou DM) - Labellisation par catégories principales

Pouvoir comparer les similarités entre les catégories d'un même modèle peut s'avérer très utile pour la compréhension des erreurs de prédiction (des *faux-positifs* et des *faux-négatifs*) du modèle ainsi que pour une éventuelle réorganisation de la ***taxonomie des catégories utilisées***. D'une part, une grande similarité entre deux catégories rend l'apprentissage plus difficile et favorise les erreurs de classification. D'autre part, des catégories trop similaires pourraient être regroupées, au moins dans un premier temps, dans une seule super-catégorie.

Ainsi, on peut imaginer une première classification utilisant des super-catégories suivie par de nouvelles classifications distinctes, plus fines, à l'intérieur des super-catégories. De cette façon, on obtiendrait une ***classification en cascade*** nécessitant une implémentation plus laborieuse mais, probablement, capable de prédictions plus justes.

Quant à la similarité entre le vecteur numérique calculé par la vectorisation d'un nouvel exposé des faits et les vecteurs numériques correspondants aux catégories du modèle, elle nous permet de résoudre le problème de classification car, les catégories les plus similaires sont aussi les plus appropriées pour être associées à un exposé des faits.

De manière analogue, pour les modèles utilisant des ***identifiants uniques par exposé des faits*** (voir Figure 4.2), on peut :

- trouver les exposés des faits du modèle les plus similaires avec un exposé des faits donné et appartenant au modèle - grâce à la fonction ad-hoc `show_similar_expos_inner` (qui utilise la méthode standard `most_similar`) ;
- afficher les exposés des faits du modèle les plus similaires avec un nouvel exposé des faits qui ne fait pas partie du vocabulaire - grâce à la fonction ad-hoc `show_similar_expos` (qui utilise également, mais dans un autre contexte, la méthode standard `most_similar`).
- proposer les catégories du modèle les plus appropriées pour un nouvel exposé des faits à classifier et qui ne fait pas partie du vocabulaire - grâce aux fonctions ad-hoc `infer_cats` et `show_proposed_cats`.

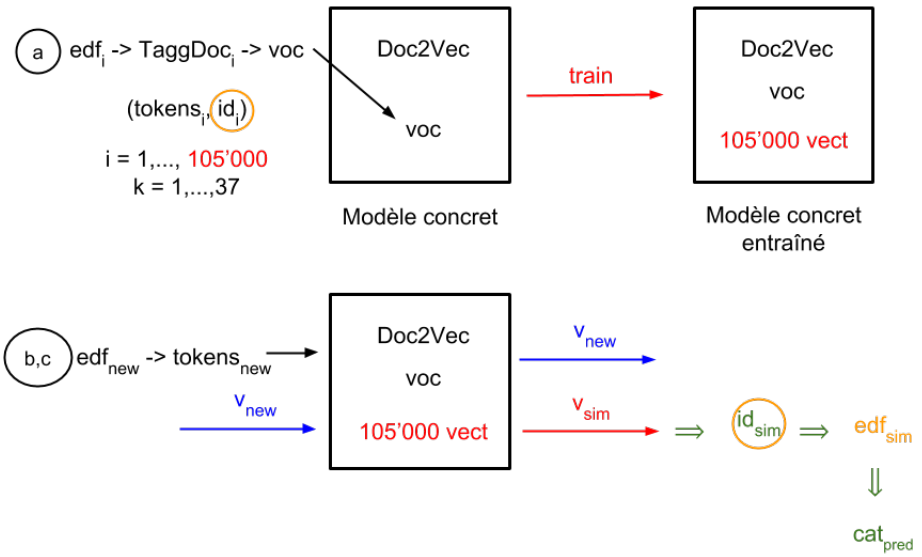

FIGURE 4.2 – Modèle Doc2Vec (DBOW ou DM) - Labellisation par identifiants uniques

Cette approche ressemble à la précédente, mais, vu que le modèle travaille avec des exposés des faits sans doublons, il est essentiel de pouvoir retrouver pour chaque tel exposé la ou les catégories qui lui étaient associées selon la classification supervisée de départ (d'entraînement). C'est justement la fonction ad-hoc `infer_cats` qui retourne cette ou ces catégories pour un exposé des faits faisant partie du modèle.

4.3 Résultats obtenus avec le modèle Doc2Vec

Afin de tester les approches proposées dans ce chapitre, on a mis d'abord au point une procédure pour déterminer la *justesse* des modèles implémentés. Il y a des fonctions standards (comme la fonction `accuracy_score` de la librairie `sklearn.metrics`) qui retournent directement la justesse mais, on a préféré un calcul ad-hoc (basée finalement sur la formule 3.1) car, dans notre cas, un même document (exposé des faits) peut apparaître plusieurs fois dans les données d'entraînement ou de test et il peut appartenir, éventuellement, à plusieurs classes (catégories).

Plus précisément, pour chaque version de notre modèle *Doc2Vec*, on a procédé ainsi :

- l'ensemble d'objets de type `TaggedDocument` (correspondant aux exposés des faits labellisés selon l'une des deux variantes mentionnées auparavant) a été partagé de manière *aléatoire* en deux sous-ensembles (grâce à la méthode strandard `train_test_split` de la librairie Python `sklearn_model_selection`) :
  - un sous-ensemble d'entraînement (formé, par exemple, de 90% de données) ;
  - un sous-ensemble de test (formé, par exemple, du reste de 10% de données) ;
- un modèle *Doc2Vec* abstrait destiné au calcul de la justesse a été créé et concrétisé en lui associant comme vocabulaire le sous-ensemble d'entraînement ;
- le modèle concret a été entraîné toujours avec le sous-ensemble d'entraînement ;
- pour chaque `TaggedDocument`, cette fois, du sous-ensemble de test :
  - on a inféré un vecteur numérique grâce au modèle qu'on venait d'entraîner ;
  - on a récupéré la ou les catégories qui lui étaient associées par l'opérateur humain (une seule catégorie pour la labellisation avec identifiant par catégorie, voire, éventuellement, plusieurs catégories pour la labellisation avec identifiant unique par exposé) ;
- pour chacun de ces vecteurs numériques inférés pour le sous-ensemble de test, on a utilisé le modèle *Doc2Vec* entraîné afin d'obtenir les vecteurs du modèle les plus similaires avec lui-même (dans nos tests les trois premiers vecteurs les plus similaires) ;
- on a récupéré les catégories associées aux vecteurs les plus similaires et on a obtenu ainsi les prédictions du modèle ;
- pour chaque exposé des faits qui a généré un `TaggedDocument` dans l'ensemble de test, on a comparé les catégories prédites par le modèle avec la ou les catégories attribuées par l'opérateur humain afin d'obtenir le nombre de prédictions justes et fausses ;
- en appliquant la formule 3.1, on a finalement obtenu la justesse du modèle *Doc2Vec*.

L'implémentation des approches *Doc2Vec* présentées dans la section précédente a été réalisée dans plusieurs fichiers *Jupyter Notebook* (voir dans l'Annexe A les noms des codes sources 3, 4, 5, 6).

Comme déjà mentionné dans la sous-section 2.1.1, les données brutes à notre disposition contiennent un nombre important de catégories qui ont très peu d'exposés des faits associés. Par conséquent, on a procédé en deux étapes.

Dans un premier temps, afin de rendre le processus d'apprentissage automatique de nos modèles pertinent, on a gardé parmi les 421 catégories seulement les 37 catégories qui ont au moins 1000 exposés des faits associés et qui correspondent à plus de trois-quart du nombre total d'exposés des faits. Elles sont appelées par la suite *catégories principales* (voir Figure 2.1).

Pour la *base de données principale* contenant seulement les exposés des faits dont les catégories sont des catégories principales, on a proposé deux classes de modèles *Doc2Vec* :

- des modèles basés sur une ***labellisation selon les catégories*** et utilisant les algorithmes *DBOW* et *DM* ;
- des modèles basés sur une ***labellisation selon les exposés des faits sans doublons*** et utilisant, à nouveau, les algorithmes *DBOW* et *DM*.

Dans le Tableau 4.1, on indique des valeurs typiques de justesse pour les premières quatre variantes correspondant au modèle *Doc2Vec* utilisé pour les catégories principales.

	Labellisation par catégorie	**Labellisation par identifiants uniques**
DBOW	65.5%	68.9%
DM	31.6%	69%

TABLE 4.1 – Doc2Vec - Catégories principales - Justesse des modèles

En utilisant la base de données principale, grâce au Tableau 4.1, il convient de remarquer que :

- pour la labellisation des exposés des faits avec les catégories attribuées par l'opérateur humain, cas où le modèle *Doc2Vec* contient seulement 37 vecteurs numériques, la variante *DBOW* est nettement meilleure que la variante *DM* ;
- par contre, pour la labellisation des exposés des faits sans doublons avec des identifiants uniques, cas où le modèle *Doc2Vec* contient un peu plus de 105'000 vecteurs numériques, les deux variantes *DBOW* et *DM* donnent des valeurs de justesse très proches ;
- quant aux deux façons de labelliser les documents, la labellisation avec des identifiants uniques par documents sans doublons conduit à des valeurs de justesse légèrement supérieures dans la variante *DBOW* ou bien supérieures dans la variante *DM*.

Dans un deuxième temps, afin d'améliorer la justesse (accuracy) des prédictions, on a créé des super-catégories, chacune regroupant plusieurs catégories principales apparentées. Plus précisément, les 37 catégories principales ont été regroupées dans 8 super-catégories (voir Figure 4.3).

Par conséquent, dans la base de données principale, on a gardé tous les exposés de faits mais on a ajouté une colonne supplémentaire correspondant aux nouvelles super-catégories qui représentent les ensembles suivants :

- la super-catégorie "CIRCULATION" regroupe les catégories principales suivantes : 'ACCIDENT - DE CIRCULATION - DEGATS MATERIEL', 'AR - DIVERS', 'INFRACTION - LCR', 'AR - PANNE - VEHICULE', 'CIRCULATION - ROUTIERE', 'AR - ACCIDENT',

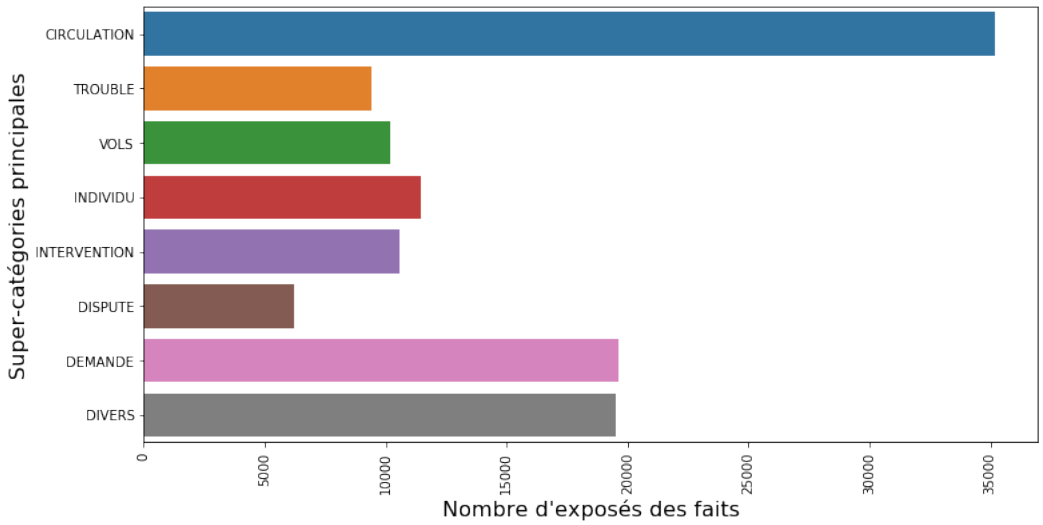

FIGURE 4.3 – Répartition des exposés des faits par super-catégories

'ACCIDENT - DE CIRCULATION - AVEC FUITE', 'ACCIDENT - DE CIRCULATION - AVEC BLESSE', 'ACCIDENT - DE CIRCULATION - AVEC ANIMAL', 'VEHICULE - SUSPECT', 'VEHICULE' ;

- la super-catégorie "TROUBLE" regroupe les catégories principales suivantes : 'TAPAGE NOCTURNE', 'TROUBLE - DE LA TRANQUILLITE / NUISANCE', 'BRUIT' ;
- la super-catégorie "VOLS" regroupe les catégories principales suivantes : 'VOL - PAR EFFRACTION', 'VOL', 'VOL - A L'ETALAGE' ;
- la super-catégorie "INDIVIDU" regroupe les catégories principales suivantes : 'INDIVIDU - SUSPECT', 'INDIVIDU - PERTURBE', 'FUITE - D'UN LIEU DE PLACEMENT' ;
- la super-catégorie "INTERVENTION" regroupe les catégories principales suivantes : 'OPERATION', 'REVOCATION', 'APPREHENSION / ARRESTATION', 'COLLABORATION - INTERPOLICE', 'DEMANDE - IDENTIFICATION' ;
- la super-catégorie "DISPUTE" regroupe les catégories principales suivantes : 'BAGARRE', 'LITIGE', 'VIOLENCE - DOMESTIQUE' ;
- la super-catégorie "DEMANDE" regroupe les catégories principales suivantes : 'DEMANDE - D'AMBULANCE', 'DEMANDE - D'ASSISTANCE' ;
- la super-catégorie "DIVERS" regroupe les catégories principales suivantes : 'ANIMAUX', 'INCENDIE', 'DROGUE', 'MINEUR - IMPLIQUE', 'TENTATIVE', 'DOMMAGES - A LA PROPRIETE', 'INDESIRABLE'.

L'introduction de super-catégories offre un double avantage :

- d'une part, les super-catégories sont mieux séparées entre elles, en réduisant le risque d'erreur de prédiction (donc moins de faux-positifs ou faux-négatifs) ;

- d'autre part, le nombre d'exposés des faits par super-catégories augmente, ce qui améliore le processus d'apprentissage.

À nouveau, on a proposé pour le cas des super-catégories les deux mêmes classes de modèles *Doc2Vec* comme pour les catégories principales.

Dans le Tableau 4.2, on indique des valeurs typiques de justesse pour les autres quatre variantes correspondant au modèle *Doc2Vec* utilisé cette fois pour les super-catégories.

	Labellisation par catégorie	**Labellisation par identifiants uniques**
DBOW	78.4%	81.7%
DM	57.4%	81.4%

TABLE 4.2 – Doc2Vec - Super-Catégories - Justesse des modèles

Pour les super-catégories, grâce au Tableau 4.2, on observe que les trois remarques faites pour les catégories principales restent valables. De plus, on constate une nette amélioration de la justesse dans le cas des super-catégories pour chacune des quatre paires de variantes. Ceci confirme l'intérêt du regroupement, au moins dans un premier temps, des catégories apparentées qui deviennent mieux définies et plus facilement distinguables.

Les valeurs de justesse obtenues pour les modèles *Doc2Vec* sont plutôt bonnes pour ce genre de classification de documents, surtout si on pense au fait que la classification est multinomiale (avec bien plus que deux classes différentes) et que les documents (les exposés des faits) ne sont pas forcément uniques et, en outre, peuvent appartenir à plusieurs classes (catégories).

Cependant, dans le chapitre suivant, on essaie d'améliorer nos résultats grâce à des modèles de régression logistique.

5
Classification multinomiale supervisée

Les modèles *Doc2Vec* proposés dans le chapitre précédent, permettaient déjà de trouver et de proposer une ou des catégories appropriées pour de nouveaux exposés de faits à classifier. Cependant, on a essayé d'améliorer ces prédictions en utilisant pour chaque modèle *Doc2Vec* un modèle associé de type *LogisticRegression* auquel on confie la tâche de la classification discrète multinomiale supervisée.

5.1 Le modèle LogisticRegression

Conformément aux [11] [12], la *régression logistique* désigne à la base un modèle statistique qui utilise une fonction logistique afin de modéliser une variable dépendante binaire. Le modèle binaire peut être étendu aussi pour des variables dépendantes ayant plusieurs valeurs possibles et c'est exactement cette situation qui nous intéresse. Dans de tels cas, on parle plutôt d'une *régression logistique multinomiale* et les valeurs possibles pour la variable dépendante sont, pour notre problème, les catégories associées aux exposés des faits.

Le modèle *LogisticRegression* est implémenté en Python dans la librairie `sklearn.linear_model.LogisticRegression` [13].

À l'heure actuelle, beaucoup d'applications utilisées pour la classification des documents sont basées sur des modèles *LogisticRegression*. Le plus souvent, la classification est binaire. De plus, si la classification est multinomiale, il est supposé qu'un même document ne peut appartenir qu'à une et une seule classe (ce qui n'est pas le cas pour nous).

Dans notre cas, on utilise des modèles *LogisticRegression* afin de réaliser la classification discrète (car les catégories associées aux exposés des faits forment bien un ensemble discret) multinomiale (car on a plus de deux catégories différentes) supervisée (car on dispose d'un ensemble d'exposés des faits labellisé "correctement" permettant l'apprentissage) des exposés

des faits. En fait, chaque tel modèle est basé sur un modèle *Doc2Vec* qui réalise la vectorisation des documents, opération indispensable au modèle *LogisticRegression*.

5.2 Utilisation du modèle LogisticRegression pour l'étude de cas

D'une manière générale, pour chaque variante d'implémentation du modèle *LogisticRegression* présentée par la suite, on a structuré le travail dans les phases suivantes :

a. *réalisation* du modèle :

b. *exploitation* du modèle.

Par la suite, on détaille les phases mentionnées ci-dessus (voir Figure 5.1).

a. La première phase, celle de la *réalisation* du modèle *LogisticRegression*, suppose, à son tour, trois étapes :

- la *création* du modèle abstrait (en précisant des méta-paramètres et, tout particulièrement, le fait qu'il s'agit d'une classification *multinomiale*) ;
- la *génération* de l'ensemble d'entraînement (train set) formé des vecteurs numériques correspondant aux exposés des faits (vecteurs qui sont obtenus grâce à un modèle *Doc2Vec* associé) et du vecteur cible (target vector) correspondant aux catégories associées (celles qui ont été attribuées par l'opérateur humain) ;
- l'*entraînement* du modèle (avec les données préparées ci-dessus et à l'aide de la méthode standard `fit`).

Un aspect clé de cette phase concerne la vectorisation des exposés des faits labellisés avec les catégories appropriées (ce qui donne la dimension *supervisée* de notre classification). Plus précisément, cette opération est réalisée par la fonction ad-hoc `edf2vec` à l'aide d'un modèle *Doc2Vec* entraîné au préalable et qui infère les vecteurs numériques (feature vectors) correspondants aux exposés des faits.

b. La deuxième phase, celle de l'*exploitation* du modèle *LogisticRegression*, est la plus importante car c'est elle qui réalise effectivement la classification discrète multinomiale supervisée des documents.

Son but est bien de prédire les catégories appropriées pour de nouveaux exposés des faits et elle implique les étapes suivantes :

- d'abord, un nouvel exposé des faits à classifier est vectorisé (avec la méthode ad-hoc `edf2vec` qui exploite le modèle *Doc2Vec* choisi) ;
- ensuite, une catégorie appropriée est proposée pour l'exposé des faits (grâce à la fonction ad-hoc `infer_cat` qui utilise la méthode standard `predict` afin de trouver la catégorie la plus probable qui correspond à l'exposé des faits) ;
- en outre, plusieurs catégories convenables peuvent être proposées pour un même exposé des faits (grâce à la fonction ad-hoc `show_infer_cats` qui utilise la méthode standard `predict_proba` afin de trouver les catégories avec les plus grandes probabilités de correspondre à l'exposé des faits à classifier).

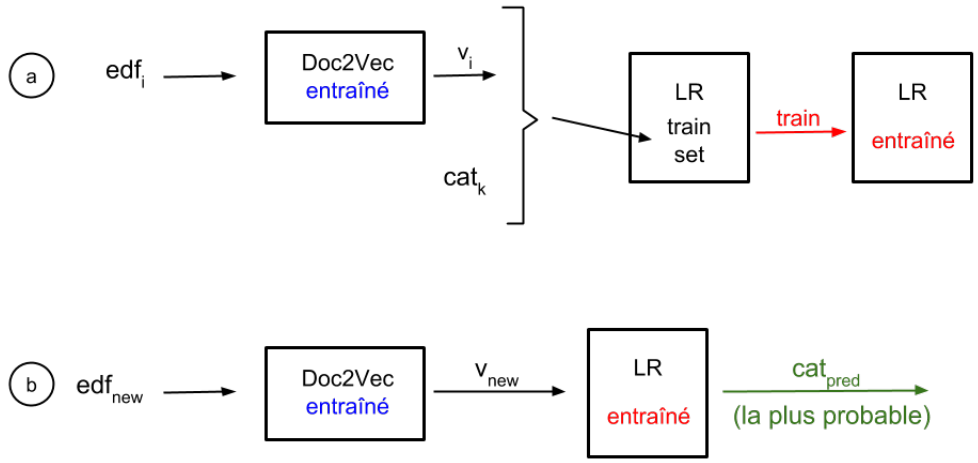

FIGURE 5.1 – Le modèle LogisticRegression

5.3 Résultats obtenus avec le modèle LogisticRegression

Afin d'évaluer les performances des solutions proposées dans ce chapitre, on a mis d'abord au point une procédure pour déterminer la justesse des modèles implémentés. Comme dans la Section 4.3, au lieu d'utiliser des fonctions standards qui retournent directement la justesse, on a préféré tenir compte, dans notre calcul de la justesse, du fait que certains exposés des faits apparaissent plusieurs fois et même avec plusieurs catégories associées (dû au fait que les frontières entre certaines catégories sont assez perméables - voir aussi la Section 2.1.1).

Plus précisément, pour chaque modèle *LogisticRegression* (qui utilise un modèle *Doc2Vec* spécifique déjà entraîné afin de vectoriser les exposés des faits), on procède ainsi :

- on partage de manière *aléatoire* (à l'aide de la fonction standard `train_test_split` de la librairie `sklearn.model_selection`) les ensembles des vecteurs numériques et des labels associés en sous-ensembles d'*entraînement* (par exemple 90% d'éléments) et de *test* (par exemple 10% d'éléments) ;
- on crée un modèle *LogisticRegression* abstrait destiné au calcul de la justesse ;
- on entraîne ce modèle avec les vecteurs numériques et les labels d'entraînement (toujours avec la méthode standard `fit`) ;
- pour chaque vecteur numérique du sous-ensemble de test, on prédit les catégories (les labels) les plus probables (dans nos tests, les trois meilleures catégories), grâce à la fonction ad-hoc `get_infer_cats` qui utilise la méthode standard `predict_proba` ;
- pour chaque exposé des faits qui a généré un vecteur numérique dans le sous-ensemble de test, on compare les catégories prédites avec le label de l'ensemble de test (qui est, en fait, la catégorie attribuée par l'opérateur humain et considérée comme juste) afin d'obtenir le nombre de prédictions justes et fausses ;
- on calcule finalement la justesse du modèle *LogisticRegression* à l'aide de la formule 3.1.

Comme déjà mentionné, dans notre cas, chaque modèle *LogisticRegression* utilise les vecteurs numériques inférés par un modèle *Doc2Vec* associé. Par conséquent, on a implémenté et testé d'abord 8 modèles *LogisticRegression* correspondant aux 8 variantes *Doc2Vec* présentées dans le Chapitre 4.

Dans le Tableau 5.1, on indique des valeurs typiques de justesse pour les quatre premiers modèles *LogisticRegression* correspondant aux 37 catégories principales.

	Labellisation par catégorie	**Labellisation par identifiants uniques**
DBOW	89%	66%
DM	71.2%	61.2%

TABLE 5.1 – Régression Logistique - Catégories principales - Justesse des modèles

Grâce au Tableau 5.1, on constate que :

- pour les deux façons de labelliser les documents par le modèle *Doc2Vec* associé, les modèles *LogisticRegression* correspondant à la version *DBOW* donnent des meilleurs résultats que ceux correspondant à la version *DM* ;
- si le modèle *Doc2Vec* associé a utilisé une labellisation par catégorie, les modèles *LogisticRegression* correspondant donnent des meilleurs résultats (et ceci à la fois pour la version *DBOW* et pour la version *DM*) ;
- la meilleure justesse est obtenue pour un modèle *LogisticRegression* basé sur un modèle *Doc2Vec* avec labellisation par catégorie et en version *DBOW*.

En comparant les Tableaux 4.1 et 5.1, on peut remarquer que :

- les modèles *LogisticRegression* basés sur des modèles *Doc2Vec* avec labellisation par catégorie conduisent à des valeurs de justesse plus grandes que celles obtenues en utilisant seulement les modèles *Doc2Vec* correspondants ;
- par contre, pour les modèles *LogisticRegression* basés sur des modèles *Doc2Vec* avec labellisation par identifiants uniques, la justesse diminue ;
- par conséquent, l'utilisation des modèles *LogisticRegression* en plus des modèles *Doc2Vec* se justifie seulement dans le cas où ces derniers labellisent les documents par catégorie.

Dans le Tableau 5.2, on indique des valeurs typiques de justesse pour les autres quatre modèles *LogisticRegression* correspondant aux 8 super-catégories.

Pour les super-catégories, grâce au Tableau 5.2, on observe que les trois remarques faites pour les catégories principales (grâce au Tableau 5.1) restent valables. De plus, on constate une nette amélioration de la justesse dans le cas des super-catégories par rapport au cas des catégories principales, pour chacune des quatre paires de variantes (ce qui était aussi le cas dans le Chapitre 4).

	Labellisation par catégorie	Labellisation par identifiants uniques
DBOW	94%	82.6%
DM	80.8%	78.4%

TABLE 5.2 – Régression Logistique - Super-Catégories - Justesse des modèles

En comparant les Tableaux 4.2 et 5.2, on constate que les remarques faites suite à la comparaison des Tableaux 4.1 et 5.1 restent en général valable (sauf pour la labellisation par identifiants uniques en variante *DBOW* où le modèle *LogisticRegression* améliore très faiblement la justesse).

Toujours à la recherche d'une augmentation de la justesse des modèles, on a essayé d'alimenter des modèles *LogisticRegression* avec des vecteurs numériques *étendus*, obtenus par la concaténation des vecteurs inférés par les versions *DBOW* et *DM* d'un même modèle *Doc2Vec*.

On a ainsi implémenté et testé quatre nouveaux modèles *LogisticRegression* et des valeurs typiques de justesse sont présentées dans le Tableau 5.3.

	Labellisation par catégorie	Labellisation par identifiants uniques
Catégories Principales	68.8%	60.5%
Super-Catégories	80.8%	78.4%

TABLE 5.3 – Régression Logistique - Justesse des modèles DBOW+DM

En comparant les Tableaux 5.1, 5.2 et 5.3, on constate que, dans notre cas, l'utilisation des vecteurs numériques *étendus* ne se justifie pas car les valeurs de justesse ainsi obtenues sont plus petites ou, au plus, égales aux valeurs de justesse obtenues par les versions *DBOW* et *DM* séparément.

Le modèle *LogisticRegression* (avec ses 12 variantes) a été le dernier modèle étudié et dans le chapitre suivant, on présente les conclusions de notre travail.

6
Conclusions

Un travail que je pensais initialement impliquant surtout de la programmation pure et dure s'est avéré au fur et à mesure un sujet complexe, pouvant être abordé de différentes manières et nécessitant une quantité importante de documentation et de réflexion.

En outre, il convient de mentionner que l'élaboration de la thèse a été faite à partir de zéro (from scratch) et les solutions retenues ont été imaginées et parachevées progressivement.

6.1 Résumé du travail

Le sujet de cette thèse concerne un problème actuel du traitement automatique du langage naturel, à savoir la classification discrète multinomiale supervisée de documents ou, plus précisément, des comptes rendu d'incidents appelés aussi exposés des faits.

On a proposé trois modèles (déclinées en plusieurs versions) dont le but commun a été de prédire, en fonction d'un catalogue précis de procédures, la ou les catégories d'intervention appropriées pour tout nouveau compte rendu d'incident à classifier.

On a testé nos solutions avec des données mises à notre disposition par l'entreprise *TechWan* qui utilise actuellement une suite logicielle pour la gestion opérationnelle et la gestion de crises. Cette base de données est formée de plus de 170'000 enregistrements représentant des couples qui associent à chaque exposé des faits la catégorie appropriée, déterminée par un opérateur humain.

D'abord, on a fait un pré-traitement de ces données afin d'éliminer les erreurs d'introduction de données et on a pu garder environ 158'000 enregistrements (voir Chapitre 2.1). Ensuite, on a systématiquement tokenisé les exposés des faits en éliminant les mots sans pertinence (comme les "mots vides", certains nombres et dates, les mots mélangeant lettres et chiffres, etc.).

Il est important de mentionner le fait qu'il y a plusieurs aspects qui ont rendu difficile la mise en place des modèles d'*apprentissage automatique* efficaces pour la classification des exposés des faits. Par exemple :

- il y a des exposés des faits qui apparaissent plusieurs fois dans la base de données avec la même catégorie associée ;
- il y a des exposés des faits qui apparaissent dans des enregistrements différents avec des catégories différentes ;
- parmi les 437 catégories, il y en a beaucoup qui ont très peu d'exposés des faits associés ;
- il y a des exposés des faits qui sont extrêmement courts (d'autant plus après l'élimination des mots sans pertinence).

En tenant compte de ces aspects, nous avons proposé trois modèles de base, chacun avec plusieurs variantes. Leur implémentation a été faite à l'aide du langage Python et de plusieurs librairies spécialisées dans le traitement du langage naturel. Les trois modèles de base exploitent des librairies Python qui implémentent :

- la fonction d'ordonnancement (ou classement) *BM25* ;
- le modèle *Doc2Vec* ;
- le modèle *LogisticRegression*.

Une fois implémentée, chaque variante de solution a été testée, grâce à la base de données *TechWan*, afin d'obtenir des valeurs de référence pour la justesse des modèles.

6.2 Comparaison des résultats

L'idée de notre modèle *BM25* (voir Chapitre 3) est d'associer d'abord à chaque catégorie une définition ad-hoc formée avec les mots apparaissant dans les exposés des faits associés à cette catégorie et ayant obtenu les scores $tf \cdot idf$ les plus élevés. Ensuite, pour un nouvel exposé des faits à classifier, le modèle propose comme catégorie(s) appropriée(s) celle(s) ayant la (les) plus grande(s) similitude(s) avec cet exposé des faits.

Ce modèle a été testé avec plusieurs jeux de catégories, notamment avec des catégories apparentées très similaires (voir Figure 3.3 et Tableau 3.1) et avec des catégories bien distinctes (voir Figure 3.4 et Tableau 3.2).

Le modèle *BM25* n'est pas approprié dans des situations comme celles du premier cas, mais il donne de bons résultats pour des situations comme celles du deuxième cas où nous avons obtenu une justesse de 94.5%.

Il convient de mentionner le fait que notre approche *BM25* peut s'avérer particulièrement intéressante pour des catégories avec un nombre réduit d'exposés des faits associés et pour lesquelles les données ne sont pas suffisantes pour envisager un modèle d'apprentissage automatique comme ceux présentés par la suite.

Notre modèle de base *Doc2Vec* (voir Chapitre 4) est un modèle d'apprentissage automatique dont le but principal est de réaliser la vectorisation (embedding) des documents (chez nous des exposés des faits). De plus, il est aussi utilisé afin de trouver la ou les catégories appropriées pour de nouveaux exposés des faits. Ce modèle nécessite beaucoup de données d'entrainement et on a proposé *8 variantes principales* pour son implémentation. Les deux premiers critères pris en compte ont été :

- le nombre de catégories traitées :
  - variante pour les 37 *catégories principales* (avec plus de 1'000 exposés des faits par catégorie) ;
  - variante pour 8 *super-catégories* (obtenues par le regroupement des catégories principales selon des relations de parenté) ;
- la façon de créer le *TaggedDocument* correspondant à chaque exposé des faits :
  - labellisation par catégorie (ce qui conduit à un nombre de vecteurs numériques par modèle égal au nombre de catégories, voir la Figure 4.1) ;
  - labellisation par identifiants uniques (ce qui conduit à un nombre de vecteurs numériques par modèle égal au nombre d'exposés des faits sans doublons, voir la Figure 4.2).

De plus, pour chacune de ces quatre variantes, on a implémenté le modèle *Doc2Vec* :

- soit en version *DBOW* (*Distributed Bag of Words*) ;
- soit en version *DM* (*Distributed Memory*).

Bien sûr, pour chaque variante principale, on peut encore choisir des méta-paramètres (par exemple la taille des vecteurs numériques, les taux initial et minimal d'apprentissage, le nombre de cœurs du processeur à utiliser, etc.) au moment de la création du modèle *Doc2Vec abstrait de départ*.

Concrètement, les valeurs des justesses (accuracies) obtenues grâce aux 8 variantes principales du modèle *Doc2Vec* pour les mêmes données de départ (à savoir les plus de 120'000 enregistrements correspondant aux catégories principales) sont présentées dans le Tableau 6.1 (qui regroupe en fait les données des Tableaux 4.1 et 4.2). Il convient de mentionner que pour chaque exposé des faits faisant partie de l'ensemble de test qui a servi au calcul de la justesse, les modèles ont inféré trois "meilleures" catégories (pas toujours différentes) et cette proposition a été considérée correcte dès qu'une de ces catégories était justement la catégorie associée par l'opérateur humain à cet exposé des faits. Les vecteurs numériques ont tous eu 300 composantes (car les tests avec des vecteurs de tailles inférieures ont conduit à des justesses plus basses, tandis que des tailles supérieures ont augmenté les temps d'exécution sans apporter une amélioration significative de la justesse).

		Labellisation par catégorie	**Labellisation par identifiants uniques**
Catégories Principales	**DBOW**	65.5%	68.9%
	DM	31.6%	69%
Super-Catégories	**DBOW**	78.4%	81.7%
	DM	57.4%	81.4%

TABLE 6.1 – Doc2Vec - Justesse des modèles

Sans reprendre toutes les observations faites à la Section 4.3, en étudiant ce tableau de synthèse, on remarque que pour notre approche *Doc2Vec* :

- si la labellisation se fait par identifiants uniques, les modèles *DBOW* et *DM* donnent des résultats similaires ; par contre, si la labellisation se fait par catégorie, le modèle *DBOW* est plus approprié que le modèle *DM* ; ce dernier comportement peut s'expliquer par le fait que le modèle *DBOW* standard travaille seulement avec des vecteurs de documents, tandis que le modèle *DM* calcule et tient compte aussi des vecteurs supplémentaires pour les mots individuels ; or, dans le cas de la labellisation par catégorie, les vecteurs "de documents" sont en fait les vecteurs correspondant aux catégories et ce sont justement eux qui permettent la prédiction directe de la catégorie appropriée ;
- le regroupement des catégories principales en super-catégories améliore la justesse des modèles (et cet effet était attendu et voulu car, en regroupant des catégories principales apparentées, les super-catégories sont plus clairement délimitées et donc mieux distinguables et, en outre, elles ont plus d'exposés des faits associés) ;
- la création d'objets *TaggedDocument* basée sur la labellisation des exposés des faits avec identifiants uniques conduit à des justesses supérieures (légèrement pour le modèle *DBOW* et fortement pour le modèle *DM*) par rapport à la labellisation utilisant des catégories ; cette différence peut s'expliquer par le fait que l'approche *Doc2Vec* a été conçue initialement (et elle est restée probablement optimale) pour la classification des documents sans doublons, avec des identifiants uniques et n'intervenant qu'une seule fois dans la création du vocabulaire associé au modèle.

Ainsi, si on veut utiliser une approche Doc2Vec, on recommande le choix de la labellisation des documents sans doublons par identifiants uniques.

Notre modèle de base *LogisticRegression* (voir Chapitre 5) est aussi un modèle d'apprentissage automatique, mais qui est spécialement conçu pour la classification discrète multinomiale et supervisée des documents. Il est décliné en 12 variantes :

- 8 variantes sont obtenues à partir de chaque variante du modèle *Doc2Vec* qui est utilisée pour inférer les vecteurs numériques correspondant aux exposés des faits (à la fois ceux déjà labellisés et ceux à classifier) ;
- 4 variantes supplémentaires ont été proposées en considérant comme vecteurs numériques dans l'ensemble d'entraînement (de différentes variantes du modèle *LogisticRegression*) des vecteurs *étendus*, obtenus en concaténant ceux inférés par les variantes *DBOW* et *DM* d'un même modèle *Doc2Vec*.

Concrètement, les valeurs de justesse obtenues grâce aux 12 variantes du modèle *LogisticRegression* pour les données de départ mentionnées ci-dessus sont présentées dans le Tableau 6.2 (qui regroupe en fait les données des Tableaux 5.1, 5.2 et 5.3).

Sans reprendre toutes les observations faites à la Section 5.3, en étudiant ce tableau de synthèse, on remarque que pour notre approche *LogisticRegression* :

- à nouveau, le modèle *DBOW* est plus approprié que le modèle *DM* ;
- à nouveau, le regroupement des catégories principales en super-catégories améliore la justesse des modèles ;
- par contre, cette fois, la création d'objets *TaggedDocument* basée sur la labellisation des exposés des faits avec les catégories conduit à des justesses bien supérieures par rapport

		Labellisation par catégorie	Labellisation par identifiants uniques
Catégories Principales	**DBOW**	89%	66%
	DM	71.2%	61.2%
	DBOW+DM	68.8%	60.5%
Super-Catégories	**DBOW**	94%	82.6%
	DM	80.8%	78.4%
	DBOW+DM	80.8%	78.4%

TABLE 6.2 – Régression Logistique - Justesse des modèles

à la labellisation utilisant des identifiants uniques ; ce changement n'est pas étonnant car le modèle *LogisticRegression* est justement conçu pour réaliser la classification par classes qui, dans notre cas, sont justement les catégories ; or, la labellisation par identifiants uniques fausse ce comportement, en mettant d'abord l'accent sur les exposés sans doublons (et non pas sur les catégories) ; en termes imagés, on peut dire que, dans la version avec labellisation par catégorie, le modèle *Doc2Vec* découvre *le premier*, durant son entraînement, la logique de la classification qui est transmise après, durant le processus de vectorisation, au modèle *LogisticRegression* associé ; celui-ci s'entraîne ensuite de façon supervisée pour comprendre, *à son tour*, de manière encore plus poussée, la même logique de classification par catégorie ;

- en outre, la concaténation des vecteurs numériques inférés par les modèles *DBOW* et *DM* de *Doc2Vec* n'améliorent pas la justesse (car les performances moins bonnes du modèle *DM* gâchent les valeurs de justesse plus élevées obtenues par le modèle *DBOW* seul).

Ainsi, si on veut utiliser une approche LogisticRegression, on recommande le choix de la labellisation par catégorie et de la variante DBOW pour le modèle Doc2Vec associé.

De plus, on recommande plutôt l'utilisation d'une approche LogisticRegression basée sur un modèle Doc2Vec approprié au lieu d'une simple utilisation de l'approche Doc2Vec.

Pour conclure, on rappelle que notre but était de trouver des solutions pour la classification discrète multinomiale supervisée des documents, qui dans l'étude de cas chez *TechWan* sont des exposés des faits avec des catégories associées. Suite à notre travail et aux modèles conçus, on peut affirmer que :

- pour des groupes de catégorie avec peu d'exposés des faits associés, on peut envisager l'utilisation du modèle de base *BM25* ;
- pour les autres catégories avec un nombre d'exposés des faits par catégorie suffisant pour faire de l'apprentissage automatique, on propose plutôt une ***approche en cascade*** : super-catégories -> catégories principales -> catégories ordinaires.

Le but de cette dernière approche est d'améliorer la justesse et l'idée est de partir d'un nombre réduit de super-catégories, en faisant une première classification avec un bon modèle *LogisticRegression* basé sur un modèle *Doc2Vec* approprié. À la limite, on pourrait regrouper les catégories avec peu d'exposés des faits dans une même super-catégorie (qui aurait dans ce cas un nombre suffisant d'exposés des faits pour envisager un apprentissage automatique). Une fois que chaque exposé des faits s'est vu associé une super-catégorie appropriée, on peut faire une nouvelle classification en fonction des catégories principales regroupées dans les super-catégories. Cette nouvelle classification se ferait, à nouveau, avec un bon modèle *LogisticRegression*, sauf pour la super-catégorie regroupant les catégories avec peu d'exposés des faits pour laquelle on pourrait utiliser le modèle *BM25*. Bien sûr, on peut éventuellement imaginer d'autres niveaux intermédiaires de classification.

6.3 Recommandations

Un des problèmes principaux qu'il fallait clarifier dans cette thèse était de savoir si la classification des comptes rendus d'incidents pouvait être réalisée de manière automatique. Comme on vient de le montrer dans la section précédente, les trois modèles proposés et déclinés en plusieurs variantes ont prouvé leur capacité de prédire la ou les catégories appropriées pour de nouveaux exposés des faits à classifier.

Cet aspect est essentiel et le choix d'une catégorie à partir d'un catalogue bien précis doit être rapide et fiable, car c'est justement la catégorie qui sera déterminante dans la suite de la gestion de l'incident. Ainsi, afin de faciliter la juste association automatique *catégorie-exposé des faits*, le travail côté TechWan pourrait suivre trois axes :

a. élaborer un recueil de bonnes pratiques (*best practices*) pour les opérateurs ;

b. former les opérateurs afin de rédiger de manière pertinente les exposés des faits ;

c. réorganiser la taxonomie des exposés des faits afin de définir des catégories mieux distinguables.

Pour le point **a.**, on peut énoncer des consignes du genre :

- garder des noms de catégorie cohérents (car, dans les données utilisées, on a rencontré des catégories distinctes comme "DISPARITION - PERSONNE **-** AGEE" et "DISPARITION - PERSONNE AGEE" ou "ACCIDENT - D'ASCEN**C**EUR" et "ACCIDENT - D'ASCEN**S**EUR") ;
- éviter les exposés des faits trop courts (car, dans les données utilisées, se trouvent des exposés comme "*2 voitures*" qui apparaît 105 fois ou "*crevaison*" qui apparaît 45 fois) ;
- utiliser des abréviations "normalisées" et éviter les abréviations personnalisées ou aléatoires (car, dans les données utilisées, on a rencontré, mais pas de manière systématique, des abréviations comme "DCD", "SIG" ou "VHC") ;
- éviter les exposés dont le contenu ne permet pas d'en déduire la catégorie correspondante (ce qui est important afin de donner la possibilité aux modèles de classification automatique d'associer de manière pertinente les exposés des faits avec leurs catégories durant le processus d'apprentissage).

Le point **b.** est intimement lié au point **a.** car la formation (de base ou continue) des opérateurs doit être réalisée dans le sens du respect des consignes comme celles mentionnées ci-dessus. Il convient de souligner le fait qu'en utilisant des modèles de prédiction automatique, les opérateurs n'auront pas besoin d'une expérience préalable dans la gestion des incidents, mais devront être rigoureux dans la rédaction des comptes rendus. La condition indispensable pour un bon exposé des faits est sa pertinence dans le sens où il doit contenir les informations clés permettant le choix clair (humain ou automatique) de la catégorie appropriée, tout en éliminant les détails inutiles.

Les modèles de classification élaborés dans cette thèse ont montré leur fiabilité mais l'amélioration de leur justesse dépend surtout de la qualité des données d'apprentissage utilisées, car ce sont elles qui permettent aux modèles de découvrir la logique d'association *exposés-catégories* durant l'entraînement.

Il faut avouer qu'on aurait pu augmenter les performances des solutions proposées en complétant (à la main) les exposés des faits de la base de données avec des détails manquants. Ainsi, en utilisant des exemples vus dans la Section 2.2 :

- dans l'exposé des faits "*son mari se bat avec un inconnu qui a essayé d'entrer dans la villa*", on aurait pu préciser "*un inconnu sous l'emprise de la drogue*" afin de faire comprendre au modèle pourquoi l'une des catégories associée à cet exposé a été "DROGUE" ;
- à la fin de l'exposé des faits "*Une voiture immobilisée sur la voie gauche.*", on aurait pu ajouter l'information pertinente "*suite à un accident sur l'autoroute*" qui permettrait au modèle de faire un lien logique avec la catégorie attribuée, à savoir "AR - ACCIDENT" ;
- quant à l'exposé des faits "*M. XYZ (07X XXX XX XX) signale un cornet noir contenant un bocal dans lequel se trouvent des sachets qui contiennent du cannabis. Tout ceci se trouve dans une haie située sur la gauche en entrant sur le parking.*", quatre mots supplémentaires, à savoir "*cornet noir jeté par un militaire*", suffiraient pour créer le lien avec la catégorie "MILITAIRE IMPLIQUE" associée à l'exposé.

Cependant, de telles modifications ultérieures des exposés des faits n'auraient pas été honnêtes. De plus, elles auraient pu seulement conduire à une justesse plus élevée pour les données existantes et *modifiées*, mais elles n'auraient pas eu d'effet positif significatif avec de nouveaux exposés des faits rédigés toujours de la manière actuelle.

Quant au point **c.**, repenser la taxonomie des exposés des faits dans la perspective d'une classification supervisée automatique n'aurait pas eu de sens avant de savoir si un telle classification soit possible. En outre, l'éventuelle réorganisation de cette taxonomie n'aurait pas pu être faite avant d'avoir plus de détails sur la façon dont la prédiction automatique agit.

Par contre, maintenant, on peut affirmer que les éventuelles modifications au niveau des catégories doivent avoir comme but de les rendre aussi distinguables que possible. Bien évidemment, tout changement du catalogue de catégories doit être fait en collaboration avec les spécialistes qui ont mis en œuvre les procédures d'intervention spécifiques à chaque type d'incident.

En reprenant des catégories concrètes mentionnées dans la Section 2.2, voilà un exemple de réflexion à envisager :

- il y a deux catégories concernant la navigation et les naufrages, à savoir "ACCIDENT - DE NAVIGATION / NAUFRAGE" et "ACCIDENT - DE NAVIGATION / NAUFRAGE - NAVIGATEUR EN DIFFICULTE" :

- avec ces noms, la deuxième catégorie est un cas particulier de la première (et elle est donc, d'une certaine façon, incluse dans la première car les exposés des faits de la deuxième catégorie sont aussi des exposés des faits de la première) ;
- il serait plus judicieux de les renommer en "ACCIDENT - DE NAVIGATION / NAUFRAGE - SANS NAVIGATEUR EN DIFFICULTE" et "ACCIDENT - DE NAVIGATION / NAUFRAGE - AVEC NAVIGATEUR EN DIFFICULTE" ;
- les deux nouveaux noms correspondent cette fois à des catégories disjointes, d'intersection nulle (et les exposés des faits devraient être rédigés en conséquence) ;

- l'exposé des faits "*direction Lausanne. Une femme enceinte et un homme qui saigne au visage.*" a eu **2** catégories associés, à savoir "ACCIDENT - DE CIRCULATION - AVEC BLESSE" et "ACCIDENT" :
  - comme dans l'exemple antérieur, la première catégorie est incluse dans la deuxième et l'opérateur humain a labellisé l'exposé avec les deux catégories au lieu de garder seulement la catégorie la plus spécifique ;
  - plus grave, cet exposé est un parfait exemple de compte rendu mal rédigé du point de vue de l'apprentissage automatique ; en fait, il ne contient pas d'élément pertinent permettant au modèle qui s'entraîne de comprendre le choix de la catégorie retenue.

En général, il faudra expliquer aux opérateurs que, malgré le stress inhérent à leur tâche, ils ne doivent pas seulement décider correctement et rapidement la catégorie, mais aussi inclure les bons indices dans l'exposé des faits pour qu'on puisse saisir leur logique.

De toute façon, la révision de la taxonomie des exposés des faits est un sujet complexe qui mérite une recherche à part entière, orientée et menée dans la perspective de l'apprentissage automatique.

6.4 Suite du travail

Une fois que des modèles fonctionnels de classification multinomiale supervisée des exposés des faits ont été implémentés et testés, le travail actuel pourra être continué dans deux directions principales afin d'assurer :

- d'un côté, l'amélioration au niveau de la programmation et des modèles de classification automatique des documents ;
- de l'autre côté, l'intégration des solutions proposées dans l'outil d'aide à l'engagement (*SAGA*).

Pour la première direction, on peut penser, par exemple, à l'extension de la base de données d'apprentissage (car un nombre plus grand de données cohérentes dans l'ensemble d'entraînement améliore la qualité de l'apprentissage automatique) ou à la recherche des méta-paramètres optimaux pour les modèles *Doc2Vec* et *LogisticRegression* utilisés.

En outre, la transformation des exposés des faits en documents tokenisés et labellisés (TaggedDocument) pourrait être rendue plus efficace :

- en choisissant de manière personnalisée les mots à éliminer, en commençant par les "mots vides" (stop words) ;

- en découvrant et corrigeant les mots mal orthographiés ;
- en regroupant des mots par famille linguistique (grâce, par exemple, aux librairies *WordNet*).

De plus, on pourrait voir aussi si de nouveaux modèles basés sur des réseaux neuronaux profonds (Deep Neural Network - DNN) conduisent à de meilleures prédictions.

Pour l'autre direction, il s'agit surtout d'ajouter de nouveaux éléments à l'interface graphique utilisateur (Graphical User Interface - GUI) du logiciel *SAGA* afin d'implémenter la nouvelle dimension "d'aide à la décision" basée sur le présent travail.

6.5 Remerciements

Maintenant, à la fin de ce travail de Master, je peux dire que j'ai eu beaucoup de plaisir à travailler dessus. J'ai rencontré maintes difficultés mais je les ai regardées comme des défis et j'ai réussi à les surmonter.

Dans ce contexte, j'aimerais remercier tout d'abord le superviseur de ce projet, Monsieur le Professeur Philippe Cudré-Mauroux, dont la compétence, la disponibilité et la gentillesse m'ont beaucoup impressionnée et aidée tout au long de mon travail.

Je suis également bien reconnaissante vis-à-vis de l'entreprise TechWan, en général, et de Monsieur l'Ingénieur Toufic Saad, en particulier, pour l'étude de cas proposée ainsi que pour le soutien financier de ma recherche.

En outre, je suis reconnaissante vis-à-vis de ma famille et de mes amis pour leurs soutien et encouragements. Aussi, je sais bien gré à ma sœur et à Monsieur Cédric Leroux, pour la lecture de ce projet et pour les diverses idées proposées quant à une quelconque amélioration.

Table des figures

Liste des tableaux

A
Annexe

On présente ci-dessous la liste des codes sources Python associés à ce travail de Master. Chaque élément de cette liste correspond à un cahier *Jupyter Notebook*.

1. `ap_preprocessing.ipynb` (implémentation du traitement préalable des données)
2. `ap_bm25.ipynb` (implémentation de la classification basée sur le modèle *BM25*)
3. `ap_doc2vec_1.ipynb` (implémentation de la classification basée sur le modèle *Doc2Vec* pour les catégories principales et labellisation des exposés des faits par catégorie)
4. `ap_doc2vec_2.ipynb` (implémentation de la classification basée sur le modèle *Doc2Vec* pour les catégories principales et labellisation des exposés des faits par identifiants uniques)
5. `ap_doc2vec_3.ipynb` (implémentation de la classification basée sur le modèle *Doc2Vec* pour les super-catégories et labellisation des exposés des faits par catégorie)
6. `ap_doc2vec_4.ipynb` (implémentation de la classification basée sur le modèle *Doc2Vec* pour les super-catégories et labellisation des exposés des faits par identifiants uniques)
7. `ap_lr_1.ipynb` (implémentation de la classification discrète multinomiale supervisée basée sur le modèle *LogisticRegression* et la vectorisation grâce au modèle *Doc2Vec* 3.)
8. `ap_lr_2.ipynb` (implémentation de la classification discrète multinomiale supervisée basée sur le modèle *LogisticRegression* et la vectorisation grâce au modèle *Doc2Vec* 4.)
9. `ap_lr_3.ipynb` (implémentation de la classification discrète multinomiale supervisée basée sur le modèle *LogisticRegression* et la vectorisation grâce au modèle *Doc2Vec* 5.)
10. `ap_lr_4.ipynb` (implémentation de la classification discrète multinomiale supervisée basée sur le modèle *LogisticRegression* et la vectorisation grâce au modèle *Doc2Vec* 6.)

Bibliographie

[1] Python. https://www.python.org/.

[2] pandas - python data analysis library. https://pandas.pydata.org/.

[3] Okapi bm25. https://en.wikipedia.org/wiki/Okapi_BM25,.

[4] summarization.bm25 – bm25 ranking function. https://radimrehurek.com/gensim/summarization/bm25.html,.

[5] Textblob : Simplified text processing. https://textblob.readthedocs.io/en/dev/.

[6] Industrial-strength natural language processing. https://spacy.io/.

[7] Doc2vec tutorial on the lee dataset. https://github.com/RaRe-Technologies/gensim/blob/develop/docs/notebooks/doc2vec-lee.ipynb,.

[8] Gensim doc2vec tutorial on the imdb sentiment dataset. https://github.com/RaRe-Technologies/gensim/blob/3c3506d51a2caf6b890de3b1b32a8b85f7566ca5/docs/notebooks/doc2vec-IMDB.ipynb,.

[9] Multi-class text classification with doc2vec & logistic regression. https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4.

[10] models.doc2vec – doc2vec paragraph embeddings. https://radimrehurek.com/gensim/models/doc2vec.html,.

[11] Logistic regression. https://en.wikipedia.org/wiki/Logistic_regression.

[12] Multi-class text classification with doc2vec & logistic regression. https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f.

[13] sklearn.linear_model.logisticregression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.