UNIVERSITÉ DE MONTRÉAL

IDENTIFICATION DE RELATIONS ENTRE PERSONNES ET BIENS IMMOBILIERS À PARTIR DE DONNÉES TEXTUELLES

FRANÇOIS FERRY
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

IDENTIFICATION DE RELATIONS ENTRE PERSONNES ET BIENS IMMOBILIERS À PARTIR DE DONNÉES TEXTUELLES

présenté par : FERRY François
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

Mme BELLAÏCHE Martine, Ph. D., présidente
M. GAGNON Michel, Ph. D., membre et directeur de recherche
Mme ZOUAQ Amal, Ph. D., membre et codirectrice de recherche
M. BILODEAU Guillaume-Alexandre, Ph. D., membre

REMERCIEMENTS

Je tiens tout d'abord d'abord à remercier mes directeurs de recherches, Michel Gagnon et Amal Zouaq, pour leur soutien et l'aide qu'ils m'ont apportée dans la réalisation de ce projet.

Je remercie également le Ministère de la Culture et des Communications du Québec pour avoir financé mon travail, mais aussi fourni les données nécessaires à sa réalisation. Je remercie aussi Isabelle Jacques, de la direction générale du patrimoine, et Philippe Michon, consultant en informatique appliquée à l'histoire, pour m'avoir soutenu et aidé à mieux comprendre les enjeux propres au domaine patrimonial.

Enfin, je remercie mon école d'ingénieur, Grenoble INP - Esisar, pour m'avoir donné l'opportunité d'effectuer un double-diplôme en partenariat avec Polytechnique Montréal.

RÉSUMÉ

Une grande partie de l'information présente sur le web et dans les bases de données l'est sous forme de textes. Ces données sont difficilement exploitables de façon automatique et il est impossible de procéder à des requêtes particulières sur celles-ci, puisqu'elles ne sont pas décrites par des métadonnées. Structurer ces données est un enjeu de taille qui permettra de les rendre plus accessibles et exploitables. De nombreuses méthodes d'extraction d'informations à partir de textes bruts ont vu le jour. Les plus répandues reposent sur des algorithmes d'apprentissage automatique et font appel à différentes techniques pour représenter les mots. Ces techniques sont indispensables et permettent de mettre en valeur certaines informations, comme la nature des mots, leur fonction, leur répartition dans le corpus, ou encore leur sémantique.

Dans le cadre de ce projet, nous allons travailler avec les données du Répertoire du Patrimoine Culturel du Québec. Ce répertoire inventorie l'ensemble du patrimoine immobilier, mobilier et immatériel du Québec.Toutefois, la classification actuelle présente des problèmes majeurs et ne répond plus aux besoins du Ministère de la Culture et des Communications du Québec (MCC). C'est pourquoi, en vue d'une refonte de la base de connaissances, le MCC nous a proposé de nous intéresser aux relations pouvant exister entre des biens immobiliers et des personnes (physiques ou morales). Ces relations sont décrites dans les synthèses historiques des biens immobiliers ; des textes décrivant chacun l'histoire d'un bien immobilier. Il existe déjà des relations modélisées dans le répertoire, mais dans l'optique d'une refonte de la classification, nous proposons une application capable de peupler de façon automatique la future base de connaissances.

Les données d'entrée de notre problème sont donc, pour chaque bien immobilier, une synthèse historique relatant l'histoire du bien immobilier et une liste de personnes qui ont été en relation avec ledit bien. La question de recherche est de savoir si une approche basée sur l'apprentissage machine est suffisante pour extraire les relations à partir de ces synthèses.

Ainsi, pour chaque couple $\langle bien\ immobilier, personne \rangle$, nous allons isoler le contexte autour de chacune des mentions de la personne dans la synthèse historique du bien. L'idée étant que les informations relatives à sa relation avec le bien immobilier sont présentes à proximité de la mention. Ce contexte peut être une fenêtre de mots ou encore la phrase au complet. Ensuite, nous allons utiliser un modèle de représentation de mots pour transformer l'ensemble des mots des contextes en un seul et unique vecteur ; et ce, peu importe le nombre d'apparition de la personne dans le texte. Nous disposons donc d'un vecteur par

couple $\langle bien\ immobilier, personne \rangle$. Ensuite, nous utilisons un classifieur binaire par type de relation ou un classifieur multiclasse pour l'ensemble des types de relation. Ces classifieurs sont basés sur des algorithmes d'apprentissage supervisé (machine à vecteur de support ou perceptron multi-couche). Nous les entraînons sur une partie du Répertoire du Patrimoine Culturel du Québec (RPCQ) puis nous les testons sur un jeu de tests manuellement annoté par nos soins.

Avec cette architecture, nous obtenons des résultats assez bons, mais insuffisants pour une mise en production. Selon la relation à détecter, nous obtenons une F-mesure entre 90% et 95% avec le jeu d'entraînement et allant de 25% à 85% avec le jeu de test.

Malgré les limites de notre approche, elle pourra néanmoins trouver une place dans le processus de modernisation et de peuplement du Répertoire du Patrimoine Culturel du Québec, en assistant un professionnel du patrimoine dans son choix de relations. Il est en effet plus rapide de valider ou de corriger un champ plutôt que de devoir systématiquement ajouter le nom de la relation. De plus, nous concluons qu'une approche uniquement basée sur l'apprentissage machine n'est pas suffisante pour arriver à des performances convenables et traiter tous les cas. Il faut donc trouver d'autres systèmes pour minimiser les erreurs, comme un système de règles basées sur une analyse syntaxique ou l'utilisation d'une ontologie du domaine.

ABSTRACT

A lot of information on the Web and in databases is in raw texts. If the raw text is easily understandable for humans, it is more difficult to process it with machines. This is why structuring data is a big challenge, that will allow making data more accessible and exploitable. There are numerous information extraction methods from raw texts. The most popular are based on machine learning and word representation to take into account some information like semantic, word distribution, etc.

In this project, we will work with data from the Repertoire of Cultural Heritage of Quebec. This repertoire brings together real estates, person, movable heritage and intangible cultural heritage of Quebec. The current classification does no longer meet the needs of the Ministry of Culture and Communication of Quebec. This is why, to help to redesign the knowledge base, we propose an application to extract relations between real estates and persons or group of persons. Each real estate has a historical synthesis which describes its history, and cite persons who played some role in its history. Thus, our goal is to process these syntheses to extract these relations. Ultimately, this application should help to settle the future knowledge base.

Input data of our problem are, for each real estate, a historical synthesis and a list of persons who are in relation with this real estate. Our research question is to determine if a machine learning-based approach is enough to extract relations from the syntheses.

For each pair $\langle real\ estate, person \rangle$, we will first isolate the context around each mention of the person in the historical synthesis of the real estate. We found out, by browsing the data, that information describing relation is very often near the mention of the person. We define the context either by a fixed number of words surrounding the mention, either by the sentence containing the mention. Then we use a word representation model to transform context into a vector. Thus, we have a vector for each pair $\langle real\ estate, person \rangle$. This vector will then be given to a supervised machine learning algorithm (support vector machine or multilayer perceptron) to predict the relation it represents. These algorithms are trained on data extracted from the Repertoire of Cultural Heritage of Quebec, and are tested on a manually annotated corpus (extracted from the repertoire and annotated by us).

We obtain pretty good results. Depending on the relation, our F-measure is between 90% and 95% with the training set and between 25% and 85% for the testing set.

Despite the limits of our approach, it could be used to assist the heritage professional in the

choice of relations. It is indeed more convenient to correct than choose relations. Furthermore, by analyzing our results, we can conclude that an approach relying only on machine learning is not enough in some cases. We should use more advanced more complex techniques like syntax analysis.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

MCC	Ministère de la Culture et des Communications
MLP	Perceptron Multi-couches (MultiLayer Perceptron)
PG	Personne ou Groupe de personnes
RPCQ	Répertoire du Patrimoine Culturel du Québec
SVM	Machine à Vecteur de Support (Support Vector Machine)
TF-IDF	Term Frequency - Inverse Document Frequency
W2V	Word2Vec
W.W2V	Word2Vec pondéré ou Weighted Word2Vec

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Contexte

Le Ministère de la Culture et des Communications (MCC) du Québec consigne un grand nombre d'informations patrimoniales dans le Répertoire du Patrimoine Culturel du Québec (RPCQ). Ce dernier inventorie l'ensemble du patrimoine mobilier, immobilier et immatériel du Québec ainsi que les personnes, groupes de personnes ou événements qui y sont associés. Y sont donc recensés des personnages historiques, des bâtiments, mais aussi des plaques commémoratives, d'anciens documents ou encore des savoir-faire.

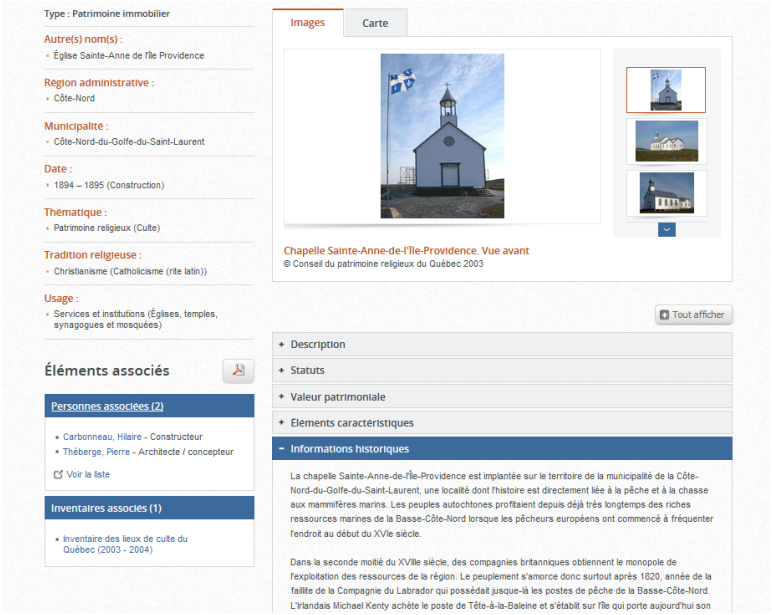

Figure 1.1 Exemple d'une fiche patrimoniale du RPCQ

Chaque entité du RPCQ est associée à une fiche. Un exemple est donné à la figure 1.1. Une fiche présente plusieurs types de champs :

— dans la partie supérieure gauche de la fiche, on peut trouver les différents noms de l'entité, son type (savoir-faire, bien immobilier, personne, etc.), et d'autres informations situationnelles, comme l'emplacement actuel du bien. Ces informations sont des méta-données structurées ;

— dans la partie inférieure gauche de la fiche, on trouve les entités associées à celle décrite par la présente fiche. Ces entités sont également recensées dans le RPCQ et sont associées de différentes manières. Ainsi, un bien immobilier peut faire partie d'un

autre bien immobilier, une personne peut être associée à un bien immobilier à différents titres (constructeur, architecte, etc.) ;

— dans la partie supérieure droite de la fiche, on peut trouver des images correspondant à l'entité et parfois une carte indiquant son emplacement (si applicable) ;

— dans la partie inférieure droite de la fiche, on trouve plusieurs champs contenant des textes précisant l'histoire de l'entité, sa valeur patrimoniale, le contexte de sa création, ses statuts patrimoniaux, etc.

Afin de moderniser ses bases de connaissances, le MCC expérimente les technologies associées au Web sémantique. Le Web sémantique consiste à relier les données afin que la machine puisse en exploiter le sens (nous détaillerons davantage ce point dans le chapitre suivant). C'est dans cette optique que le MCC nous a proposé de travailler sur les relations existant entre les biens immobiliers et les personnes ou groupes de personnes du RPCQ. Ces relations sont d'un type prédéfini. Le tableau 1.1 décrit l'ensemble des relations existant dans le RPCQ, ainsi que leur nombre d'occurrences dans le registre. Dans ce tableau, nous avons repris la terminologie du RPCQ. Si les relations sont nommées par des substantifs, il faut bien comprendre qu'elles ne désignent pas la profession ou l'occupation de la personne associée, mais bien son rôle dans l'histoire du bien immobilier.

Tableau 1.1 Répartitions des relations existant dans le RPCQ

Nom	effectif	% du total
Architecte / concepteur	5570	33%
Occupant	1846	10%
Constructeur	1795	9%
Artiste-Artisan	502	3%
Auteur	27	<1%
Collectionneur	1	<1%
Conseiller	11	<1%
Destinataire	42	<1%
Éditeur	2	<1%
Fabricant	28	<1%
Ingénieur	3	<1%
Inhumé en ce lieu	32	<1%
Producteur	1	<1%
Propriétaire	10	<1%
Sujet	5	<1%
Autre	7053	42%

1.2 Problématique

En analysant le tableau 1.1, on remarque donc que plus de 40% des instances de relations sont de type *Autre*, ce qui est problématique puisque ce type n'apporte pas d'information sur la nature de la relation existant entre la Personne ou Groupe de personnes (PG) et le bien immobilier. Sans trop de surprises, on trouve également beaucoup d'instances de relations *Architecte / concepteur*, *Occupant* et *Constructeur*. En revanche, on remarque que la relation *Propriétaire* est sous-représentée pour un corpus de ce type. On s'attend en effet à ce qu'elle soit bien plus présente dans un corpus traitant essentiellement de biens bâtis. Il y a par ailleurs certaines relations qui ne sont pas adaptées à ce corpus ou dont le sens est flou. Par exemple, la relation *Sujet* est difficilement explicable dans ce contexte : que signifie « un PG est le *Sujet* d'un bien immobilier » ?

Ces irrégularités nous ont donc amené à effectuer une analyse plus poussée des données. Les points suivants ont ainsi été mis en lumière :

- Dans la très grande majorité des cas, toutes les informations nécessaires à la qualification des relations se trouvent dans les synthèses historiques des biens. Une synthèse historique est un texte plus ou moins long relatant l'histoire d'un bien. Il y est donc fait mention de sa construction, des éventuelles modifications, des occupants notables, des propriétaires, etc. ;
- Les relations pouvant exister entre PG et biens immobiliers sont parfois complexes et donc difficilement représentables dans la classification actuelle. Par exemple, il n'est actuellement pas possible de représenter le cas où un PG a fait appel à un entrepreneur pour construire sa maison. Par ailleurs, la classification manque de clarté et parfois de précision. Il est en effet difficile d'appréhender le sens de la relation *Auteur* dans le cadre du patrimoine immobilier : que signifie qu'un PG est l'*auteur* d'un bien immobilier ? Est-ce qu'il est à l'origine de sa construction ? Est-ce lui qui l'a construit ? Ce manque de précision est d'ailleurs palpable lorsque l'on parcourt les données : ces instances sont peu nombreuses et désignent généralement non pas la relation mais l'activité du PG. De même, la relation *Artiste / Artisan* semble bien trop vague : est-ce que le PG qui a accompli la charpente d'une église est considéré comme un *constructeur* ou plutôt un *Artiste / Artisan* ? Est-ce que la réalisation d'une charpente doit être représentée par la même relation que la réalisation de sculptures ? Toutefois, ces considérations ne font pas partie du présent travail, mais elles justifient le besoin d'une nouvelle classification ;
- Il n'est pas possible de représenter le temps dans la présente classification. On ne peut donc pas dater les relations, pour dire par exemple que « X a construit le bâtiment

en 1874, puis en Y a agrandi la grange en 1896. » La plupart de nos relations sont limitées dans le temps, en particulier l'*occupation*, mais aussi, dépendant de comment on la considère, la *construction*. Ce problème est assez sérieux dans la mesure où nous avons un corpus fortement lié à l'histoire ;

— Il n'est actuellement pas permis à un PG d'avoir plus d'une relation avec un bien immobilier. Ainsi, il n'est pas possible de dire qu'un PG est à la fois le constructeur et le propriétaire d'un bien ; ou encore l'occupant et le propriétaire, ce qui est un cas assez commun ;

— De nouveaux types de relations ont été rajoutés a posteriori, comme c'est le cas de la relation *Propriétaire*, ce qui explique le faible nombre d'instances de cette relation ;

— La création de nouvelles fiches ainsi que l'ajout des relations appropriées est une tâche longue et fastidieuse.

Cette analyse a permis de mieux définir le besoin du MCC, à savoir une nouvelle classification, ainsi qu'une solution pour faciliter la saisie des relations, mais aussi le peuplement de la base de connaissances. En effet, puisqu'une nouvelle classification sera réalisée, il sera nécessaire de classer les relations selon ladite classification. Étant donné l'ampleur de la tâche, il est préférable de se tourner vers l'utilisation d'une méthode automatique. C'est pourquoi nous proposons au MCC de développer une application pour extraire les relations entre biens immobiliers et PG à partir des synthèses historiques des biens.

Lors du parcours du RPCQ, nous avons remarqué que les liens sont bien souvent décrits de façon explicite dans la synthèse historique. Nous avons donc formulé l'hypothèse suivante : il est possible d'identifier les liens entre un PG et un bien immobilier en utilisant une approche basée sur l'apprentissage supervisé, à l'instar du problème de classement de document. Ce qui nous amène aux questions de recherche suivantes :

Q1 Est-ce qu'une approche basée sur l'apprentissage supervisé est suffisante pour résoudre le problème en utilisant uniquement des outils indépendant de la langue ?

Q2 Est-ce que le type de relation a une influence sur les performances de détection ?

En effet, notre parcours des données nous laisse penser que nos relations sont facilement détectables en ne regardant que les mots composant nos phrases, sans que nous ayons à nous intéresser à la structure grammaticale de ces phrases. Il reste à vérifier si notre intuition est correcte. Nous allons donc extraire ces phrases, puis les confier à un algorithme d'apprentissage automatique et regarder si ces résultats sont probants. L'analyse d'erreurs nous permettra ensuite de déterminer la source des problèmes rencontrés par notre approche, et ainsi, répondre à notre question de recherche *Q1*.

Comme évoqué précédemment, certaines relations sont plus ou moins spécifiques. La relation

Propriétaire désigne uniquement qu'un PG possède le bien immobilier, alors que la relation *Artiste/Artisan* peut tout aussi bien être employée lorsque le PG a construit l'orgue de l'église, a sculpté les fonts baptismaux, ou a fondu une cloche. Fort de ceci, il nous apparaît raisonnable de supposer que cette différence de spécificité entre en jeu dans les performances de détection. Un autre point pouvant entrer en jeu est la variabilité du vocabulaire employé. Reprenons l'exemple de la relation *Propriétaire*. Dans les synthèses historiques, il y a plusieurs façons de dire qu'une personne a été propriétaire du bien immobilier :

— « X a hérité de la propriété » ;
— « la maison a été cédée à X » ;
— « X a acheté les locaux » ;
— « le premier propriétaire était X » ;
— « l'intendant attribua cette terre à X ».

La comparaison de nos différents résultats devrait nous permettre de constater l'influence du type de relation sur les performances de détection. L'analyse d'erreur devrait également apporter des pistes dans ce sens, notamment en ce qui concerne la variabilité du vocabulaire.

Dans ce mémoire, nous allons tenter de trouver une solution pour détecter les relations entre PG et bien immobilier en nous appuyant uniquement sur le contexte dans lequel le PG est mentionné dans la synthèse historique du bien immobilier. Ceci permettra alors de faciliter le peuplement de la base de connaissances du MCC en vue de sa reconception. La conception de cette nouvelle base de connaissances ne fait pas partie du présent travail, de même que la gestion de la temporalité. La gestion des cas où un même couple $\langle PG, Bien\ immobilier \rangle$ est lié à plusieurs relations sera pour le moment laissée de côté, n'ayant pas assez d'exemples pour le traiter avec une approche basée sur l'apprentissage supervisé. Cependant, nous évoquerons quelques pistes pour traiter ces cas-ci dans les travaux futurs.

1.3 Objectifs de recherche

Le premier objectif de notre recherche est d'analyser les données pour en comprendre les enjeux, mais aussi leur nature afin de relever les problèmes que pourrait rencontrer notre approche. Par la suite, il est nécessaire de faire un état de l'art des techniques couramment utilisées en matière d'extraction de relations, et en particulier celles basées sur l'apprentissage machine ; puis de voir comment il est possible de les appliquer dans le cadre de notre projet avec les ressources dont nous disposons. Le dernier objectif consiste à analyser les résultats des différentes méthodes testées pour pouvoir tirer des conclusions, mais également proposer des améliorations.

La principale contribution de ce projet est une méthode d'extraction de relations n'utilisant pas de base de connaissances ou de base de vocabulaires. En effet, comme nous allons le voir dans le chapitre suivant, l'extraction de relation fait souvent appel à une ontologie du domaine (Byrne and Klein (2009), Andreas and Douglas (2015)) ou une base de vocabulaire afin de déceler le sens des termes rencontrés dans les textes (Buranasing et al. (2016), Augenstein et al. (2012)). Ainsi, notre méthode devrait pouvoir s'adapter facilement à d'autres bases de connaissances patrimoniales, et faciliter ainsi le peuplement desdites bases par le traitement automatisé des données présentes dans des textes bruts. Une autre contribution est la production d'un modèle de plongement lexical utilisant l'architecture Skip-Gram de Word2Vec (plus de détails seront données en section 3.3.2). Ce modèle a été entraîné sur les données de Wikipédia-Fr.

1.4 Plan du mémoire

Le présent mémoire est tout d'abord constitué d'une revue de littérature faisant état de différents projets de numérisation de bases de connaissances, de techniques existantes en matière de représentation de mots et d'extraction de relations. Les chapitres suivants se concentrent sur la description de nos données ainsi que l'approche utilisée. Ensuite, nous présentons nos résultats et tentons de les expliquer. Enfin, nous concluons sur notre approche et proposons diverses voies d'amélioration.

CHAPITRE 2 REVUE DE LITTÉRATURE

Afin de faciliter l'accessibilité et donc l'utilisation des données patrimoniales, de nombreux projets de numérisation de données et de création de bases de connaissances ont vu le jour, comme Europeana [1] ou encore ARIADNE [2]. Le peuplement de ces bases de connaissances est une tâche considérable, en particulier lorsqu'il s'agit d'ajouter des informations présentes dans des textes non structurés. En effet, si ces données sont facilement compréhensibles et utilisables par des humains, elles le sont beaucoup plus difficilement par des ordinateurs. Il est alors bien plus difficile d'effectuer des requêtes sur des informations contenues dans des textes que sur des informations structurées dans une base de connaissances.

Nous allons donc nous intéresser à des méthodes basées sur l'apprentissage machine. Dans la plupart des cas, ces méthodes font appel à ce que l'on appelle des représentations de mots. Ces mots ainsi représentés sont ensuite donnés à un classifieur basé sur un algorithme d'apprentissage machine. En analysant la littérature, il semblerait que le choix du classifieur n'ait pas une grande influence sur les résultats, contrairement à celui de la représentation de mots ; c'est d'ailleurs une hypothèse secondaire que nous vérifierons dans nos résultats.

Dans les parties suivantes, nous évoquons dans un premier temps les différents projets de numérisation de bases de connaissances existant dans le domaine culturel, puis nous parlons des différentes représentations de mots, et enfin nous évoquons les différents travaux déjà réalisés en matière d'extractions de relations.

2.1 Numérisation de bases de connaissances patrimoniales

Comme évoqué précédemment, plusieurs projets de numérisation ont vu le jour, notamment grâce à l'essor du Web sémantique. Le Web sémantique est un ensemble de technologies développé par le World Wide Web Consortium (W3C). L'objectif est le partage et la réutilisation des données par plusieurs applications et utilisateurs, en permettant à la machine de *comprendre* les données et faire les inférences qu'un humain ferait sans difficulté. Pour ce faire, les concepts sont liés et représentés de façon unique par un *Uniform Resource Identifier* (URI). Ces données ainsi structurées forment le *Web des données* (*linked data* en anglais). Lorsque ces données sont librement accessibles, on parle alors de *Linked Open Data*, communément abrégé LOD. Parmi les acteurs majeurs du LOD, nous avons DBpedia (Auer et al. (2007)) et WordNet (Miller (1995)). Le premier, considéré comme le centre du LOD, est une version

1. `https://www.europeana.eu/portal/fr`
2. `http://www.ariadne-infrastructure.eu/`

structurée des infoboîtes de Wikipedia. De fait, il couvre un nombre considérable de domaines et est donc bien souvent relié aux autres bases de connaissances. DBpedia est principalement construit par exploration et extraction automatique de pages Wikipedia, et ce, en plusieurs langues. WordNet, quant à lui, est une base de données développée par des linguistes de l'Université de Princeton depuis 1995. Le but de ce projet est de créer un dictionnaire regroupant l'ensemble des mots de la langue anglaise et de les lier en fonction de leur sémantique, de les regrouper par concept. WordNet est donc souvent utilisée pour l'extraction de relations afin de trouver le sens des termes utilisés, et donc identifier la relation associée.

Dans le domaine du patrimoine, des projets de création de bases de connaissances existent déjà et sont librement accessibles. Ainsi, Europeana [3] (Doerr et al. (2010)) propose des ressources numériques de plusieurs institutions culturelles de l'Union Européenne. À l'heure où nous écrivons ces lignes, cela représente plus de cinquante-cinq millions de ressources provenant de plus de trois mille cinq cents musées, galeries d'art, bibliothèques et archives d'Europe. Ces ressources sont de types divers (articles, contenu multimédia, photographies, ...) et sont regroupées selon plusieurs collections thématiques et peuvent également être accessibles par sujets, sources, par périodes de temps ou même par couleurs (pour les images). Il est également possible d'effectuer directement des requêtes sur les données en utilisant `http://sparql.europeana.eu/`.

Également financé par la Commission Européenne, ARIADNE [4] (Niccolucci and Richards (2013)) est un projet en version bêta ayant pour objectif de rassembler des données archéologiques. Il est ainsi possible d'effectuer des recherches par lieu, par période temporelle et par sujet. Ces données sont des rapports d'excavation, des études faites en laboratoire ou sur le terrain, etc., soit environ deux millions de ressources.

Mis en ligne en juillet 2011, Data.bnf.fr [5] regroupe les données de la Bibliothèque Nationale de France (BNF) afin d'en améliorer l'accès et la visibilité. Comme pour les deux projets précédemment cités, les données sont en accès libre soit par un portail web, soit par un point d'accès prenant en charge les requêtes [6]. Les données sont là encore assez diversifiées : on trouve différents auteurs, des œuvres, des fiches sur des spectacles, etc. Il est également possible de parcourir les fiches au moyens d'une carte, si l'on cherche par exemple des données sur un lieu particulier.

À l'instar de ces projets, le MCC veut structurer ses données selon les normes publiées par le W3C et les rendre accessibles à tous. L'objectif est double : faciliter l'accès pour le grand

3. `https://www.europeana.eu/portal/fr`
4. `http://www.ariadne-infrastructure.eu/`
5. `http://data.bnf.fr/fr/`
6. `http://data.bnf.fr/sparql/`

public, mais aussi permettre une meilleur expérience d'utilisation pour les professionnels ; notamment les chercheurs et les employés du MCC. Une des grandes différences avec les deux projets européens cités plus haut est que, dans le projet du MCC, toutes les données proviennent du même endroit, de la même base ; alors que les deux projets européens ont pour objectif premier de rassembler les données culturelles éparpillées dans des bases de données différentes.

2.2 Représentations de mots

La manière la plus simple de représenter un texte est la représentation par sac de mots (aussi connue sous le nom de *bag of words*). Le principe est de représenter un document par un vecteur de la même taille que le dictionnaire consignant l'ensemble des mots du corpus. Chaque composante i du vecteur représente la fréquence du i^{e} mot du dictionnaire dans le document. Le vecteur est ensuite traité selon l'utilisation qui va en être faite. Il peut alors être normalisé en divisant chaque composante par la norme du vecteur, ou encore binarisé en mettant les composantes non nulles à 1 (le vecteur indique alors la présence ou l'absence des mots). Il est également possible de diviser chaque composante par le logarithme en base dix de l'inverse de la proportion de documents contenant le terme associé, par rapport au nombre total de documents (on appelle alors ce nombre la fréquence inverse du document) ; on obtient alors le coefficient TF-IDF (Term Frequency - Inverse Document Frequency) (Salton and Buckley (1988)) pour ce terme dans le document associé au vecteur. Le principe de TF-IDF va donc être d'augmenter l'importance d'un terme en fonction de sa fréquence dans le document et de la diminuer en fonction de sa fréquence dans tout le corpus. En d'autres mots, un terme fréquent dans un document est significatif pour ce document, mais un terme fréquent dans l'ensemble du corpus perd son pouvoir discriminant. Le fonctionnement de TF-IDF sera évoqué plus en détail dans la présentation de notre méthodologie. Si les méthodes basées sur des sacs de mots sont relativement simples à mettre en place, un inconvénient majeur est le nombre élevé de dimensions du vecteur résultant, augmentant la difficulté de traitement ultérieur des vecteurs ainsi formés.

Pour pallier ce problème de dimensions, on utilise des méthodes de plongement lexical (aussi appelées *plongement de mots* ou encore *word embedding*). Le plongement lexical va permettre de faire correspondre chaque mot d'un dictionnaire à un vecteur de nombres réels de façon à ce que les mots apparaissant dans un contexte similaire soient représentés par des vecteurs proches dans leur espace vectoriel. La dimension de l'espace vectoriel est un paramètre de la création du modèle de plongement, et est généralement grandement inférieure au nombre de mots du dictionnaire. Ainsi, l'on travaille avec des données de plus faible dimension, ce qui

facilite l'utilisation d'algorithmes d'apprentissage machine.

Mikolov et al. (2013a) (ainsi que Mikolov et al. (2013b)) présentent Word2Vec, un modèle pouvant utiliser deux architectures différentes qui implémentent le plongement lexical en se basant sur des réseaux de neurones. Le principe est de confier à un réseau de neurones une tâche particulière sur un large corpus de documents ; puis de garder uniquement la première couche du réseau de neurones ainsi entraîné au lieu du modèle final. La tâche confiée pour créer le modèle dépend de l'architecture choisie : dans l'architecture *Continuous Bag Of Word* (CBOW), on va demander au réseau de neurones de prédire un mot à partir de son contexte (c'est-à-dire les mots qui l'entourent dans une fenêtre de taille donnée) ; dans l'architecture *Skip-Gram* (SG), l'on va demander au réseau de neurones de prédire le contexte à partir d'un mot donné. L'architecture SG fonctionne mieux avec peu de données d'entraînement et pour les mots peu fréquents, alors que l'architecture CBOW s'entraîne plus rapidement et fonctionne mieux pour les mots fréquents.

Pennington et al. (2014) soutiennent que Word2Vec ne fonctionne pas de façon optimale, car le modèle n'exploite pas les informations statistiques quant à la cooccurrence des mots. Mesurer la cooccurrence d'un mot avec ceux du corpus consiste à évaluer le nombre d'occurrences de chaque mot du corpus apparaissant dans le contexte du mot en question. Le contexte est alors généralement un certain nombre de mots autour du mot qui nous intéresse. Supposons par exemple que dans notre corpus, nous définissons le contexte comme étant la phrase et que nous ayons les deux phrases suivantes : « John Ostell a construit la maison en 1887, à proximité de la rivière. » et « David Ouellet a habité la maison en 1924. ». Alors « maison » cooccurre une fois avec « construit », une fois avec « habité », trois fois avec « la », etc. Pour répondre aux lacunes de Word2Vec, Pennington et al. (2014) proposent GloVe (pour Global Vector), une autre implémentation du plongement lexical combinant à la fois des avantages de Word2Vec et des méthodes exploitant la cooccurrence des mots. GloVe utilise les probabilités que deux mots cooccurent, c'est à dire la probabilité qu'un mot k apparaisse dans le contexte d'un mot w. Cette probabilité est calculée en faisant le rapport entre le nombre d'apparitions du mot k dans le contexte de w et le nombre de fois que chaque mot apparaît dans le contexte de w. L'information sémantique est alors obtenue en examinant le rapport entre ces probabilités. Pour reprendre l'exemple de l'article, soit $P(k|w)$ la probabilité que le mot k apparaisse dans le contexte du mot w. Considérons alors les mots « vapeur », « glace », « solide » et « eau ». Alors, $P(solide|glace)$ devrait être proche de 1, contrairement à $P(solide|vapeur)$ qui devrait être proche de 0. Donc le rapport $\frac{P(solide|glace)}{P(solide|vapeur)}$ devrait être bien supérieur à un. En effet, il est raisonnable de penser que la glace est plus souvent associée au terme « solide » que la vapeur. De la même façon, $P(eau|glace)$ et $P(eau|vapeur)$ devrait en être relativement proches, et donc leur rapport devrait être proche de 1, de même si à

la place de « eau » nous avions un terme aussi éloigné de « vapeur » que de « glace ». La conclusion est donc la suivante : soient trois mots k, a et b. Si le rapport $\frac{P(k|a)}{P(k|b)}$ est :

— Proche de 1, alors ni a ni b ne sont discriminants pour k ;
— Supérieur à 1, alors k est sémantiquement plus proche de a ;
— inférieur à 1, alors k est sémantiquement proche de b.

L'objectif de l'apprentissage du modèle GloVe va donc être d'apprendre des vecteurs de mots tels que leur produit scalaire est égal au logarithme de leur probabilité de cooccurrence. Comme le logarithme d'un rapport est égal à la différence des logarithmes, l'objectif va donc être d'associer des rapports de probabilités de co-occurrence (ou plus exactement leur logarithme) avec des différences de vecteurs dans l'espace de plongement lexical. En d'autres termes, l'information portée par le rapport de probabilités de cooccurence est représentée dans le modèle par une différence de vecteurs.

L'un des inconvénients des deux méthodes précédemment citées est qu'elles ne tiennent pas compte du contexte du document, si bien qu'il peut y avoir des ambiguïtés sur des termes pouvant être présents dans plusieurs domaines, ou sur des homographes. Par exemple, le mot « fils » aura toujours le même vecteur, peu importe si l'on parle de la filiation ou du pluriel de « fil ». Pour répondre à ce problème, Le and Mikolov (2014) proposent Doc2Vec, une modification de Word2Vec conçue pour prendre en compte le document, levant ainsi les possibles ambiguïtés. Le fonctionnement de Doc2Vec est similaire à celui de Word2Vec, sauf que cette fois-ci, en plus d'avoir une liste de vecteurs de mot, le modèle va également générer une liste de vecteurs de document. Chaque vecteur de document va représenter le sens général du document auquel il est associé et sera construit conjointement avec les vecteurs des mots composant ledit document : la tâche donnée au réseau de neurones sera alors de prédire un mot sachant son contexte et le document. Le vecteur de document sera donc modifié en conséquence. Ce vecteur pourra ensuite être utilisé en complément des vecteurs de mots ; ou seul, puisqu'il est censé représenter l'ensemble du document. Là encore, on obtient un vecteur dense d'une dimension bien plus faible à celle des vecteurs générés par une méthode à base de sac de mots. Néanmoins, un des inconvénients de Doc2Vec est que l'entraînement est long et nécessite un corpus conséquent. Par ailleurs, il n'est pas possible de construire un modèle pour ensuite le réutiliser sur un autre corpus, puisqu'il est nécessaire de construire les vecteurs de document.

Dans le même ordre d'idée que Doc2Vec, Kamkarhaghighi and Makrehchi (2017) proposent un système de représentation de documents, nécessitant moins de données d'entraînement que Doc2Vec, et également moins de calculs de la part du processeur. L'idée va être ici de tenir à la fois compte des données « locales » (c'est-à-dire les données que l'on veut exploiter) et de connaissances de bases, issues d'un modèle GloVe ou Word2Vec préalablement entraîné sur

un vaste corpus. Le point important est que le modèle fournissant les connaissances de base n'a pas besoin d'être entraîné sur les données locales, contrairement à Doc2Vec. À partir du modèle de connaissances de base, on construit pour chaque document un arbre représentant le contenu dudit document. Cet arbre va être construit en rattachant chaque nouveau mot à un mot de l'arbre en maximisant la corrélation entre le vecteur de ce nouveau mot et le vecteur du mot de l'arbre. Pour reprendre l'exemple de l'article, si dans l'arbre nous avons déjà les mots « histoire », « appartement » et « livre » ; et que le prochain mot du contexte est « chapitre », alors ce mot sera vraisemblablement rattaché à « livre », car c'est avec le vecteur de « livre » que le vecteur de « chapitre » aura le maximum de corrélation. Le vecteur de chaque nœud ainsi ajouté est alors modifié pour tenir compte du vecteur père en calculant une moyenne pondérée entre le vecteur du nœud et le vecteur de son père. Ainsi, on obtient pour chaque mot un vecteur tenant à la fois compte de la valeur sémantique et du contexte.

Li et al. (2016) proposent de combiner les architectures CBOW et SG de Word2Vec en partant du principe qu'elles se complètent puisqu'opposées. Cet article propose également de construire un vecteur représentant chaque document en faisant la moyenne pondérée des vecteurs composant ledit document. Ce vecteur de document est alors utilisé de façon similaire à celui de Doc2Vec lors de l'apprentissage des vecteurs de mots.

Garten et al. (2015) partent de l'intuition que les modèles de représentation de mots sont bien souvent linéaires (ce qui est un point soulevé par Mikolov et al. (2013b)), pour en explorer des combinaisons linéaires. Leur recherche montre que de telles combinaisons peuvent améliorer les performances. Ainsi, les auteurs ont combiné des vecteurs construits par une approche basée sur des réseaux de neurones (*Word2Vec*) avec des vecteurs construits par une approche basée sur l'analyse de cooccurrences. Pour combiner les vecteurs, deux approches ont été essayées : par addition vectorielle et par concaténation ; et il s'est avéré que la concaténation donne les meilleurs résultats. La question du pourquoi reste cependant assez floue.

Ainsi, il existe plusieurs façons de représenter les documents, chacune ayant des avantages et des inconvénients. Dans notre projet, nous ne disposons pas d'un corpus suffisamment important pour entraîner notre propre modèle basé sur des réseaux de neurones. Nous allons donc utiliser un modèle Word2Vec pré-entraîné utilisant l'architecture CBOW (Schöch (2016)). N'ayant pas trouvé de modèle utilisant l'architecture Skip-Gram, nous avons décidé d'en entraîner un nous-même, en utilisant le même protocole que celui qui a conduit à la production du modèle CBOW. Nous allons également utiliser TF-IDF, qui est bien souvent un point de références dans la littérature.

2.3 Apprentissage supervisé

L'apprentissage supervisé est une technique d'apprentissage automatique qui consiste à entraîner un modèle à effectuer une certaine tâche en lui fournissant des exemples. Une fois entraîné, ce modèle est alors à même de réaliser ladite tâche. L'apprentissage supervisé permet de traiter deux grandes catégories de problèmes : les problèmes de régression, consistant à prédire des valeurs ; et les problèmes de classification, consistant à classer des éléments. Dans notre cas, nous sommes dans un problème de classification ; les algorithmes d'apprentissage supervisé que nous allons utiliser sont donc des classifieurs.

Dans tous les articles passés en revue dans le présent chapitre, l'accent est principalement mis sur le prétraitement des données et les représentations de mots utilisés ; le choix du classifieur n'est généralement pas documenté. Il semblerait donc que ce choix n'a que peu d'importance. Dans notre projet, nous avons essayé plusieurs classifieurs : un arbre de décisions, la méthode des k plus proches voisins, une forêt d'arbres décisionnels (aussi appelée « Random Forest »), XGBoost, un perceptron multicouche et un séparateur à vaste marge. Nous allons maintenant expliquer, dans les grandes lignes, le fonctionnement de ces algorithmes.

Un arbre de décision (Breiman et al. (1984)) est un arbre dont les feuilles représentent les variables cibles (dans notre cas les feuilles représentent les relations que nous cherchons à détecter). Les nœuds correspondent alors à des conditions qui mènent à ces valeurs. Lors de l'apprentissage, l'algorithme va construire l'arbre et définir, à l'aide des données, les éléments qui conduisent aux différentes variables cibles. Considérons le cas simple où l'on veut détecter des relations de *construction* et de *conception*. Dans notre entraînement, l'algorithme remarque que pour toutes les instances de *construction*, il y a le mot « construit », et que toutes les autres instances sont des instances de *conception*. Alors on peut supposer que l'arbre contiendra un nœud « Est-ce que la phrase contient le mot « construit » ? » qui enverra vers la relation *construction* si c'est effectivement le cas, ou vers la relation *conception* si ce n'est pas le cas.

Plusieurs classifieurs utilisent et combinent des arbres de décisions. Parmi eux nous avons la forêt d'arbres décisionnels (Breiman (2001)) et XGBoost (Chen and Guestrin (2016)). Une forêt d'arbres décisionnels est un modèle composé de plusieurs arbres de décisions entraînés sur des sous-ensembles des données d'entraînement. La prédiction s'effectue alors sur chaque arbre et le résultat final correspond à celui trouvé par la majorité des arbres. XGBoost est une bibliothèque logicielle implémentant le « Gradient Boosting ». Le principe est là encore de produire un modèle composé de plusieurs autres modèles plus simples, généralement des arbres de décisions. L'ensemble est ensuite harmonisé en utilisant une fonction objectif (aussi

connue sous le terme anglais « loss function »).

La méthode des k plus proches voisins (Peterson (2009)) consiste à prédire un résultat en fonction de ses plus proches voisins. La notion de proximité dépend alors de comment l'on définit la distance dans l'espace d'entrée de nos données. Pour mieux comprendre le fonctionnement, considérons que nous avons une instance de relation à classer et que notre modèle est entraîné pour comparer l'entrée avec les 5 plus proches voisins. En comparant cette instance à celles de notre jeu d'entraînement, le modèle identifie que les plus proches voisins, donc d'une certaine façon les instances qui ressemblent le plus à notre entrée, sont trois instances de relation de *construction*, une instance de relation de *conception* et une instance de relation de *décoration*. Alors, l'instance en entrée sera identifiée comme étant une relation de *construction*, puisque parmi nos 5 plus proches voisins il y a une majorité de *construction*.

Un perceptron multicouche (Rosenblatt (1961)) est un réseau de neurones formels sur plusieurs couches et dans lequel l'information ne circule que dans un seul sens (le réseau est alors dit « à propagation directe »). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche représentant les valeurs cibles de notre modèle (donc dans notre cas, nos relations). Un neurone formel est une représentation mathématique qui possède plusieurs entrées, une sortie et qui est caractérisée par une fonction d'activation. Chaque liaison entre neurones est affectée par un coefficient, et c'est ce coefficient qui est modifié au cours de l'apprentissage afin de que les entrées conduisent bien aux sorties prévues, et donc conduisent à identifier correctement la classe.

Enfin, le séparateur à vaste marge (Boser et al. (1992)), aussi appelé machine à vecteurs de support, est une technique consistant à résoudre les problèmes de discrimination. Son principe est de trouver une frontière entre les données de différentes classes de façon à maximiser la distance entre cette frontière et les données les plus proches (cette distance est alors appelée « marge »). Pour trouver cette marge, le problème est reformulé en problème d'optimisation quadratique, type de problèmes pour lequel il existe des algorithmes de résolution connus. Dans le cas où les données ne sont pas séparables de façon linéaire, une fonction de noyau permet alors d'augmenter la dimension de l'espace des données dans l'espoir qu'une telle séparation existe dans l'espace ainsi formé.

2.4 Extraction de relations

Augenstein et al. (2012) proposent LODifier, une application permettant d'extraire des triplets RDF utilisant DBpedia et WordNet. Les entités correspondant à un URI de DBpedia sont dans un premier temps reconnues en utilisant *Wikifier* (Milne and Witten (2008)) ; un

outil qui permet, à partir d'un texte, de détecter la mention d'entités présentes dans DBpedia. Les relations entre les entités trouvées sont ensuite détectées grâce à un analyseur syntaxique. WordNet est ensuite utilisé pour détecter la sémantique des relations, permettant alors de construire un graphe RDF basé sur les vocabulaires de DBpedia et WordNet. Par exemple, si nous avons la phrase « John McCarthy a créé le langage de programmation LISP », alors Wikifier va dans un premier temps détecter les entités présentes dans DBpedia, à savoir « John McCarthy », « Langage de programmation » et « LISP » ; ensuite, l'analyseur syntaxique va permettre d'identifier que « John McCarthy » est le sujet de « a créé », ce dernier est un verbe dont l'objet est « Langage de programmation », dont le complément du nom est « LISP ». Les triplets résultants vont donc être : « John McCarthy créateur de LISP » et « LISP est un langage de programmation ». Si LODifier se veut utilisable peu importe le type de corpus, deux points sont bloquants dans notre projet. Premièrement, nos textes sont en langue française, langue qui n'est pas supportée par Wordnet. Deuxièmement, il n'y a que 5% des entités de notre corpus qui sont répertoriées dans DBpedia.

Byrne and Klein (2009) proposent d'extraire des événements à partir de textes provenant du *National Monument Record of Scotland*, une base de données relationnelle décrivant des sites archéologiques. Cette base de données décrit chaque site par des textes relatant notamment les différentes études qui y ont été faites, les fouilles archéologiques, etc. L'objectif est donc d'extraire ces événements et les informations qui y sont rattachées en utilisant divers procédés de traitement automatique de la langue naturelle. Dans un premier temps, les textes sont segmentés en phrases. Un analyseur syntaxique détermine ensuite la nature des mots de chacune de ses phrases. Les entités nommées sont ensuite identifiées et extraites ; puis chaque paire d'entités nommées est associée à une instance d'événement préalablement défini dans une ontologie. Ces deux dernières tâches sont réalisées à l'aide de techniques d'apprentissage supervisé, requérant donc un corpus d'entraînement manuellement annoté. Pour l'extraction de relation, les features sont au nombre de 17, et compte notamment le type des entités nommées, la distance entre ces entités et leur position dans la phrase. Les résultats sont bons avec une précision au-delà de 90% et une F-mesure d'environ 75%.

Odat et al. (2014) utilisent également une approche basée sur la détection d'entités nommées et l'analyse syntaxique pour extraire des relations à partir de textes bruts. Cette fois-ci, ce sont des publications traitant de la conservation des œuvres d'art qui sont analysées. Leur application se sert elle aussi d'une ontologie du domaine, ce qui permet d'avoir une meilleure précision dans les termes. C'est également dans une optique de peuplement que l'application a été développée.

Buranasing et al. (2016) proposent d'extraire les hyperonymes, synonymes et d'autres re-

lations sémantiques à partir de plusieurs sources de données patrimoniales. Cette approche se base sur l'utilisation de motifs sémantiques. En d'autres termes, elle consiste à analyser un texte pour découvrir s'il contient des motifs transportant une valeur sémantique. Par exemple, si dans un texte il est écrit « Jean-Baptiste Poquelin, aussi connu sous le nom de Molière », le motif « aussi connu sous le nom de » va permettre de détecter une relation de synonymie entre « Jean-Baptiste Poquelin » et « Molière ». Utilisant également une approche basée sur des motifs sémantiques, Andreas and Douglas (2015) proposent d'extraire des entités et des relations à partir de rapports d'archéologie en s'appuyant sur CIDOC-CRM [7], une ontologie spécialisée dans la représentation de concepts patrimoniaux. Pour extraire les relations, l'approche utilise un dictionnaire construit à partir de thesauri et de glossaires de termes du domaine patrimonial. De façon analogue, Vlachidis and Tudhope (2013) proposent d'extraire des informations sur des artefacts en analysant des textes. Cette approche se base sur l'utilisation de patrons sémantiques et de glossaires pour extraire la description de chaque artefact, sa forme, ses dimensions et son numéro de référence. L'objectif est de peupler une ontologie basée sur CIDOC-CRM avec ces informations. Afin de modéliser les règles, Vlachidis and Tudhope (2013) utilisent JAPE (Java Annotation Patterns Engine, Cunningham et al. (1999)), un traducteur à états finis basé sur des expressions régulières, qui permet d'écrire des règles pour extraire des informations à partir de textes annotés (les règles portent sur les annotations et non sur le texte). Par exemple, il est possible avec JAPE d'écrire une règle qui spécifie que toutes les personnes apparaissant après le mot « constructeur » sont des constructeurs. JAPE est un composant de l'environnement GATE (General Architecture for Text Engineering, Cunningham (2002)), une infrastructure logicielle possédant de nombreux composants spécialisés dans le traitement automatique de la langue naturelle, et qui est également utilisée par Vlachidis and Tudhope (2013) pour annoter le texte en vue de l'utilisation de JAPE.

Choi et al. (2016) proposent d'extraire les relations en deux étapes. La première étape consiste à déterminer si une phrase contient ou non une relation. Pour ce faire, le modèle va utiliser une analyse syntaxique pour extraire deux types de patrons : les patrons de dépendances les plus courtes et les patrons lexicaux. Le premier type de patron va trouver dans l'arbre syntaxique d'une phrase le plus petit sous-arbre liant deux entités nommées ; les termes appartenant à cette arbre décrivent donc potentiellement la relation entre les deux termes. Les patrons lexicaux, quant à eux, sont construit en analysant les mots se trouvant entre les entités nommées (les IN-Words) et les mots précédant la première entité ou suivant la seconde (les OUT-Words). Pour une phrase donnée, le modèle va donc extraire les patrons de dépendances et les patrons lexicaux, pour ensuite en calculer la similarité avec ceux des

7. `http://www.cidoc-crm.org/`

données d'entraînement afin de définir si la phrase décrit ou non une relation. La seconde étape va permettre ensuite de caractériser cette relation en utilisant des règles sur la position des mots et leur fonction grammaticale, ainsi que les mots en eux-mêmes.

Dans la plupart des cas, l'extraction de relations s'appuie sur des bases de connaissances déjà existantes, que ce soit DBpedia ou des ontologies spécialisées. Dans notre projet, nous n'avons pas à notre disposition de telles bases. Nous allons donc étudier s'il est possible de s'en passer et d'utiliser une approche uniquement basée sur des représentations de mots et des algorithmes d'apprentissage supervisé.

CHAPITRE 3 MÉTHODOLOGIE DE RECHERCHE

Pour notre problème, nous allons nous concentrer sur une approche basée sur l'utilisation de représentations vectorielles de mots et de classifieurs automatiques d'apprentissage supervisé. Notre tâche se découpe donc en plusieurs étapes. Premièrement, nous devons construire un jeu de données d'entraînement sur lequel nous effectuerons nos expérimentations. Ensuite, nous validerons nos résultats sur un jeu de données de test que nous aurons construit à partir des données du MCC. Nous utiliserons une validation croisée à dix échantillons sur nos données d'entraînement. En effet, puisque nous avons peu de données, nous n'allons pas utiliser de jeu de données de validation pour trouver les meilleurs paramètres de nos algorithmes.

3.1 Jeu de données

Le MCC a mis à notre disposition tout le contenu du RPCQ. Parmi ces données, on compte :

- la liste des biens immobiliers, chacun représenté par un identificateur unique, un nom et une synthèse historique ;
- la liste des Personnes et Groupes de personnes (PG), chacun représenté par un identificateur unique, un nom, un prénom, un secteur d'activité et une occupation (ou sous-secteur dans le cas d'un groupe de personnes).
- la liste des paires $\langle \text{PG}, \textit{Bien Immobilier} \rangle$, chacune représentée par un identificateur unique et une relation associée (pour rappel, il n'est actuellement pas possible dans le RPCQ d'avoir plusieurs instances de relation entre un PG et un bien immobilier, c'est pourquoi il n'y a qu'une seule relation associée à chaque couple) ;

Ces données sont le résultat direct de l'entrée de fiches patrimoniales dans le RPCQ par des professionnel(le)s du MCC, et ce, au fil des années. Elles ont donc été étiquetées par ces mêmes personnes. Dans notre approche, nous allons uniquement utiliser les synthèses historiques des biens immobiliers, les noms et prénoms des PG, ainsi que la liste des paires $\langle \text{PG}, \textit{Bien Immobilier} \rangle$.

3.1.1 Données d'entraînement

Afin de construire notre jeu de données d'entraînement, nous avons sélectionné les instances des relations les plus représentées dans le RPCQ, à savoir les relations d'*occupation*, de *conception*, de *construction* et de *décoration*. Nous avons dès le départ écarté la catégorie *autre* pour les raisons suivantes. Premièrement, cette catégorie n'apporte rien sur la nature de la

relation, si ce n'est son existence. Deuxièmement, nous savons, après un parcours de quelques instances de cette catégorie, que l'on peut y trouver aussi bien des instances de relations de *conception*, d'*occupation* ou de tout autre type, comme des instances de relations de *propriété*, de *rénovation*, etc. Nous ne pouvons donc pas l'utiliser comme relation particulière, puisque cela créerait de la confusion (on aurait par exemple des instances de relation d'*occupation* classées dans la catégorie *autre*, mais aussi dans la catégorie *occupation*, ce qui induit des confusions entre les deux catégories). En nous basant sur les relations sélectionnées, nous avons donc 9713 instances de relations.

À partir de là, nous éliminons toutes les instances pour lesquelles le bien immobilier n'a pas de synthèse historique, ce qui représente environ 47% des instances. En effet, il n'y a que les biens ayant été classés qui possèdent obligatoirement une synthèse historique. Il nous reste alors 5156 instances. Nous éliminons les instances pour lesquelles le PG n'est pas mentionné dans la synthèse historique du bien immobilier, puisque si le PG n'est pas mentionné, alors sa relation avec le bien n'est pas mentionnée non plus, donc nous ne pouvons pas l'extraire en analysant le texte. Il serait néanmoins possible de la récupérer en nous appuyant sur d'autres sources, ce qui dépasse le cadre du présent travail. Nous nous retrouvons alors avec 3399 instances, dont :

— 1686 instances de la relation de *conception* ;
— 1040 instances de la relation d'*occupation* ;
— 509 instances de la relation de *construction* ;
— 164 instances de la relation de *décoration*.

Note : La liste ci-dessus représente les effectifs finaux, c'est-à-dire après avoir mis de côté une partie des instances pour construire le jeu de tests (voir ci-après).

Comme l'a révélé une étude préliminaire sur une centaine de données choisies aléatoirement dans le corpus, nous savons qu'une partie de ces instances est potentiellement bruitée. Si nous ne sommes pas en mesure de quantifier ce bruit, nous savons que :

— Certains couples $\langle \text{PG}, \textit{Bien Immobilier} \rangle$ devraient être liés à plusieurs relations, ce qui n'est pas possible dans la classification actuelle. Par conséquent, il n'y a qu'une et une seule instance pour ce couple. La relation associée à ladite instance est alors à la discrétion de la personne qui a créé la fiche. Par exemple, nous pouvons avoir dans une fiche : « Guy Lasalle a construit une petite maison qu'il habite pendant plusieurs années », avec « Guy Lasalle » associé dans le RPCQ au bien immobilier par une relation de *construction* ; alors qu'en réalité, il est aussi lié au bien par une relation d'*occupation*. Des cas comme celui-ci engendrent du bruit, puisque nous avons dans le même contexte deux instances de relations différentes, mais ce contexte sera identifié comme ne représentant qu'une seule relation : le vocabulaire de l'autre relation va

donc créer de la confusion avec celui de la première. Nous estimons que ce problème touche entre 7% et 10% des données (estimation réalisée en parcourant environ 1400 instances choisies aléatoirement) ;

— Nous avons également le cas pour lequel nous avons deux instances de relations différentes portant sur une même phrase, ce que l'analyse d'erreur dans la section 4.6 identifiera comme une source commune de confusion entre les relations de *conception* et de *construction*, pour des raisons semblables au cas expliqué précédemment. Un exemple typique est « X a construit le bien Z selon les plans de Y ». « X » a donc construit « Z » et « Y » a conçu « Z ». En définissant donc le contexte comme étant une phrase ou une fenêtre de mots suffisamment large, nous aurons donc deux instances de relation qui pour des étiquettes différentes auront les mêmes données. Sur nos données, 27% des instances portent sur la même phrase qu'une autre instance ;

— Une autre source de bruit est la présence d'erreurs d'étiquetage, par exemple une instance de relation d'*occupation* qui a été étiquetée comme étant une instance de relation de *conception*. Lors du parcours des données pour la construction du jeu de tests, il s'est avéré que 10% des instances de *construction*, 16% des instances de *décoration* et 20% des instances de *conception* étaient mal attribuées (étant donné que nous n'avons pas utilisé de données appartenant initialement à la relation d'*occupation*, nous n'avons pas de données là dessus ; néanmoins, sur une centaine d'instances aléatoirement sélectionnées, il n'a été trouvé aucune erreur).

3.1.2 Construction d'un jeu de tests

Pour construire le jeu de tests, nous avons, avec l'aide d'un autre étudiant du laboratoire, manuellement annoté des instances du RPCQ en nous concertant dès lors que certaines instances occasionnaient des difficultés. Initialement, l'objectif était de former un jeu de données d'au moins 200 instances de chacune des relations, afin de travailler d'une part sur des données propres, et d'autre part avec un plus grand nombre de relations. Néanmoins, il est apparu que le RPCQ n'était pas aussi bruité que nous le pensions, et nos résultats n'étaient pas particulièrement meilleurs avec les données ainsi corrigées. Nous avons donc décidé de prendre comme données d'entraînement les données de bases non-corrigées provenant du RPCQ et comme jeu de données de tests une partie de ce que nous avions corrigé (nous avions également remis dans les données d'entraînement une partie des relations de *décoration* afin d'en avoir suffisamment pour l'entraînement).

Là encore, nous avons éliminé des données les instances pour lesquelles les biens immobiliers n'ont pas de synthèse historique, ainsi que celles impliquant un PG non mentionné dans la

synthèse du bien immobilier.

Notre jeu de tests comporte donc 509 instances réparties comme suit :

— 167 instances de relation de *conception* ;
— 96 instances de relation d'*occupation* ;
— 225 instances de relation de *construction* ;
— 21 instances de relation de *décoration*.

Le tableau 3.1 indique de quelles relations proviennent nos instances du jeu de test. Comme on peut le constater, les instances de *conception* proviennent en grande majorité de leur classe d'origine. Les instances d'*occupation* proviennent majoritairement de la classe *autre*. On constate que 16% des instances de *construction* proviennent de *Artiste-Artisan*, ce qui illustre bien la limite floue entre l'artisanat et la construction.

Outre ces relations, nous avons également noté la présence d'autres relations, en particulier en analysant les relations classées dans la catégorie *autres*. Parmi les plus nombreuses, nous avons :

— 120 instances de relation de *propriété* (on est donc bien loin des 10 instances actuellement étiquetées dans le RPCQ) ;
— 47 instance de relation de *demande de construction* (signifiant qu'un PG a commandité la construction du bien immobilier) ;
— 58 instance de relation de *modification* de bien (rénovation, agrandissement, et autres travaux).

Mise à part la relation de *propriété*, les autres relations ne comptent pas assez d'instances pour le moment. Nous avons choisi de ne pas utiliser la relation de *propriété* parce qu'elle est très souvent couplée avec la relation d'*occupation*. En d'autres termes, si un PG est le propriétaire d'un bien immobilier, il est souvent lié par une autre relation avec ce bien (généralement une relation d'*occupation*). Néanmoins, nous avons utilisé ces données pour expérimenter une approche basée sur une taxonomie de classifieurs, approche que nous détaillerons plus loin.

Tableau 3.1 Provenance des instances du jeu de tests

Relations du jeu de tests	Relations du RPCQ				
	Architecte	Constructeur	Artiste-Artisan	Auteur	Autre
Conception	161	1	1	0	4
Occupation	4	0	2	4	86
Construction	3	175	36	0	11
Décoration	0	0	9	0	12

Le parcours des données effectué lors de la réalisation du jeu de données de tests nous a permis d'effectuer les observations suivantes, qui nous ont guidé dans la suite de notre méthodologie.

Les relations que nous devons extraire lient un PG à un bien immobilier. Une des particularités des synthèses historiques est qu'elles ne mentionnent que très rarement le bien immobilier de façon explicite. Il est alors soit désigné par des termes génériques comme « le bâtiment », « l'église », etc. soit désigné de façon implicite, par exemple : « Les plans sont réalisés par David Ouellet ». Mais comme une synthèse historique ne traite que d'un seul et unique bien, nous avons donc fait l'hypothèse que si une phrase évoque une relation, alors celle-ci concerne le bien décrit par le texte.

Une synthèse historique décrit généralement plusieurs relations entre différents PG et le bien immobilier. C'est pourquoi nous allons pour chaque instance extraire uniquement le contexte de chaque mention du PG dans la synthèse du bien immobilier. L'application peut être décomposée en trois blocs : l'extraction du contexte, la représentation vectorielle dudit contexte, et la classification. L'extraction du contexte permet d'isoler la partie du texte qui nous intéresse pour le couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$ étudié. C'est d'ailleurs uniquement dans cette partie que nous nous servons du PG et du bien immobilier. En effet, une fois que nous trouvons le contexte de mention du PG dans la synthèse historique du bien immobilier, nous supprimons la mention du PG du contexte, puisque rien ne permettra de le représenter dans nos représentations vectorielles. La partie du texte ainsi extraite sera ensuite représentée de façon vectorielle, puis confiée à des classifieurs utilisant des algorithmes d'apprentissage supervisé. Les sections suivantes décrivent en détail le fonctionnement de chacun des blocs susmentionnés. Dans la mesure où la totalité des synthèses historiques est en langue française, nous avons essayé d'utiliser le plus possible des méthodes indépendantes de la langue.

3.2 Extraction du contexte

Cette étape consiste à éliminer, pour chaque couple, tout ce qui ne concerne pas le PG dans la synthèse historique du bien immobilier concerné. Pour cette étape, nous avons essayé deux méthodes distinctes :

- on ne garde que les phrases qui mentionnent le PG ;
- on ne garde qu'un certain nombre de mots autour de chaque mention du PG, on parle alors de fenêtre de mots. Ces fenêtres de mots ne tiennent pas compte des phrases. Ainsi, prendre une fenêtre de taille 4 consiste à prendre les 4 mots avant et après chaque mention du PG (dans le cas où les fenêtres se chevauchent, nous ne prenons qu'une seule fois chaque mot dans le chevauchement).

À l'issue de cette étape, nous avons donc, pour chaque couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$, une

liste de mots décrivant potentiellement la relation entre les deux parties, en faisant l'hypothèse que ladite relation est décrite dans le contexte où le PG apparaît. Bien évidement, il arrive que le PG soit aussi référé par un pronom dans la suite de la synthèse, mais nous avons décidé de ne pas nous y attarder pour plusieurs raisons. Premièrement, nous avons constaté que dans la grande majorité des cas, la relation est décrite dans la phrase où le PG est mentionné ; ajouter d'autres phrases ou d'autres mots risquerait d'apporter des données supplémentaires inutiles, et donc potentiellement du bruit. Deuxièmement, la récupération des mentions sous forme de pronoms implique d'effectuer de la résolution d'anaphores, qui est une méthode assez complexe dont il existe peu d'implémentations pour la langue française.

Le tableau 3.2 présente le nombre moyen de mots par contexte pour chaque relation (dans le cas où le contexte est défini comme les phrases mentionnant le PG de façon explicite). On remarque que, dans l'ensemble, la longueur est assez variable. En comparant nos deux jeux de données, nous remarquons que les phrases sont plus longues sur notre jeu de données d'entraînement pour les relations de *construction* et d'*occupation*. Ceci s'explique par la provenance des données pour la construction du jeu de tests : la majeure partie des instances d'*occupation* du jeu de tests proviennent de la catégorie *autre*, et une partie (environ 25%) des instances de *construction* proviennent d'ailleurs également. Toutefois, la différence de longueur de phrases entre les deux jeux de données pour les instances de construction reste importante, il se peut donc qu'il y ait eu un biais dans la création du jeu de tests lors de la sélection « aléatoire » des instances.

3.3 Représentations vectorielles

À l'issue de l'extraction du contexte, nous avons donc pour chaque couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$ une liste de mots. La partie suivante consiste donc à traiter ces listes de mots afin de les rendre utilisables par les classifieurs. À la fin de cette étape, nous aurons pour chaque couple un vecteur de nombres réels censé représenter la relation impliquant ledit couple. À cette fin, nous avons expérimenté deux méthodes de représentation de mots communément utilisées : les vecteurs TF-IDF et le plongement lexical (aussi appelé "Word Embedding"). Ces deux méthodes sont décrites ci-après.

3.3.1 Vecteurs TF-IDF

La transformation Term Frequency - Inverse Document Frequency (TF-IDF) est une méthode de pondération. Elle évalue l'importance d'un terme d'un document par rapport à un ensemble de documents en s'appuyant sur l'observation empirique que l'importance d'un

Tableau 3.2 Longueur moyenne des phrases de chaque contexte pour chaque relation (en nombre de mots)

(a) Jeu de données d'entraînement

Relation	Moyenne	écart type
Conception	19,15	11,07
Décoration	19,79	10,10
Construction	20,42	12,62
Occupation	25,64	22,41
Toutes relations	21,36	15,85

(b) Jeu de données de test

Relation	Moyenne	écart type
Conception	18,32	10,43
Décoration	18,84	9,96
Construction	16,18	7,47
Occupation	22,78	16,76
Toutes relations	18,29	11,16

terme croit proportionnellement à sa fréquence dans le document et décroit proportionnellement à sa fréquence dans le corpus. En d'autres termes, plus le mot est fréquent dans un document, plus il sera significatif ; toutefois, s'il est également fréquent dans les autres documents, il sera alors peu significatif. Ainsi, les déterminants seront en toute logique associés à des poids faibles, contrairement à la plupart des noms communs. Plus formellement, le coefficient TF-IDF d'un terme t d'un document d est donné par :

$$tfidf(t,d) = tf(t,d) \times idf(t) \tag{3.1}$$

Avec $tf(t,d)$ le nombre d'occurrences du terme t dans le document d et $idf(t)$ défini comme suit :

$$idf(t) = log\left[\frac{Card(d)+1}{(df(t)+1)}\right] + 1 \tag{3.2}$$

Avec $Card(d)$ le nombre de documents dans lesquelles le terme t apparaît, et $df(t)$ la fréquence de t dans le corpus.

Dans notre cas, les documents sont les listes de mots obtenues par l'extraction du contexte puis une lemmatisation [1]. Afin de représenter chaque liste de mots sous forme vectorielle, nous allons considérer $Dict$, un dictionnaire contenant l'ensemble de mots lemmatisés existants dans notre corpus de documents (soit environ 3600 mots différents). Pour chaque liste de mots, c'est-à-dire pour chaque instance de relation, nous allons calculer un vecteur v de nombres réels tel que :

$$\forall i \in \mathbb{N}, v_i = tfidf(Dict[i], d) \tag{3.3}$$

Avec v_i la i^e composante du vecteur v ; d la liste de mot représentant le couple analysé ; $Dict[i]$ le i^e mot du dictionnaire $Dict$.

1. Nous avons utilisé ce lemmatiseur : `https://github.com/ClaudeCoulombe/FrenchLefffLemmatizer`

v est donc un vecteur dont la dimension est $Card(Dict)$ (c'est-à-dire le nombre de mots différents dans le corpus) et dont les composantes non nulles sont celles correspondant aux mots présents dans la liste de mots qui a permis de le générer. Il va donc de soi que v possède un très grand nombre de composantes nulles : on parle alors de vecteur creux. Dans notre cas, nous avons construit notre modèle TF-IDF sur l'ensemble des instances de notre jeu d'entraînement, après extraction du contexte. Nous n'avons pas procédé à des séparations entre nos relations.

Nous obtenons donc avec cette méthode un vecteur creux par couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$.

3.3.2 Plongement lexical

Le plongement lexical (ou Word Embedding) consiste à représenter dans un espace vectoriel continu un mot d'un dictionnaire en se basant sur l'apprentissage d'une représentation de mots. Cette représentation est construite de façon à ce que les mots apparaissant dans un contexte similaire soient représentés par des vecteurs proches dans l'espace vectoriel de destination. Ainsi, le mot « roi » est certainement plus proche de « reine » qu'il ne l'est de « moutarde ».

Dans notre travail, nous allons utiliser Word2Vec, dont le fonctionnement a été détaillé dans la revue de littérature 2.2.

Dans notre projet, nous avons utilisé un modèle déjà entraîné (Schöch (2016)) sur l'ensemble des données de Wikipédia-fr (à partir d'un dépot d'octobre 2016), ce qui représente environ cinq cents millions de mots et 3,5 Go de textes bruts. Il a été construit en utilisant l'implémentation de gensim [2] avec l'architecture CBOW en dimension cinq cents. Nous avons également entraîné nous même un modèle en gardant les mêmes paramètres mais en prenant l'architecture Skip-Gram, à titre de comparaison. Ce dernier modèle a également était entraîné sur l'ensemble des données de Wikipédia-fr (mais une version de juillet 2018 cette fois-ci).

Comme nos contextes n'ont pas toujours le même nombre de mots (en particulier lorsque l'on définit ce contexte comme étant les phrases dans lesquelles est mentionné le PG), il nous faut maintenant combiner les vecteurs obtenus pour chacun des mots de chacun des contextes. Pour ce faire, nous utilisons plusieurs méthodes, chacune étant décrite dans les paragraphes suivants.

La première méthode, la plus simple, consiste à calculer pour chaque contexte la moyenne des vecteurs de mots. Plus formellement, soient c le contexte d'une relation, c_i le i$^{\text{e}}$ mot du

2. `https://radimrehurek.com/gensim/models/word2vec.html`

contexte c et n la longueur dudit contexte, alors le vecteur de contexte v_{contexte} est défini par :

$$v_{\text{contexte}} = \frac{1}{n} \sum_{i=1}^{n} Word2Vec(c_i) \tag{3.4}$$

Note : Ici $Word2Vec(c_i)$ est un vecteur de nombres réels, par conséquent, la somme de la formule précédente est une somme vectorielle telle qu'elle est usuellement définie dans les $\mathbb{R}$-espaces vectoriels.

La deuxième méthode associe Word2Vec et TF-IDF. Le principe derrière cette association est de profiter des informations de chacune des deux méthodes : Word2Vec nous apporte des informations sémantiques tandis que TF-IDF nous apporte des informations sur la distribution des mots dans le corpus. Ainsi, pour chaque relation, nous allons calculer la moyenne des vecteurs de mots pondérés par leur coefficient TF-IDF (en d'autres termes, chaque vecteur de mot est pondéré par le coefficient TF-IDF dudit mot). Plus formellement, en reprenant les mêmes notations que précédemment et en considérant pour TF-IDF que le document d est le contexte c :

$$v_{\text{contexte}} = \frac{1}{n} \sum_{i=1}^{n} tfidf_d(c_i) \cdot Word2Vec(c_i) \tag{3.5}$$

3.4 Classification

Maintenant que nous avons pour chaque couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$ un vecteur censé représenter la relation existant entre les deux, il ne nous reste plus qu'à les classer. Pour ce faire, nous avons essayé trois approches : en utilisant un classifieur binaire par type de relation, en utilisant un classifieur multiclasse ou en utilisant une taxonomie de classifieurs.

Ces classifieurs font appel à des algorithmes d'apprentissage supervisé. Au cours du projet, nous avons essayé plusieurs algorithmes différents pour ne retenir que les deux nous offrant les meilleures performances en matières de F-mesure, précision et rappel ; à savoir un Séparateur à Vaste Marge (en anglais Support Vector Machine, SVM) et un Perceptron Multicouche (en anglais MultiLayer Perceptron, MLP), tous deux décrits dans le chapitre 2.3.

Du point de vue de l'implémentation, nous avons utilisé les classes de *Scikit-learn*[3] suivantes :

- *sklearn.svm.SVC*, avec un noyau linéaire ;
- *sklearn.neural_network.MLPClassifier*, avec la fonction d'activation linéaire rectifiée "relu", fonction définie par $\forall x \in \mathbb{R}, relu(x) = max(0, x)$, une seule couche cachée de 100 neurones, le solveur "adam" proposé par Kingma and Ba (2014), un taux d'apprentissage constant de 0,001 et un nombre maximum d'itérations positionné à

3. `http://scikit-learn.org/`

1000. La taille de la couche d'entrée est égale à la dimension des vecteurs utilisés (environ 3600 dans le cas de TF-IDF, contre 500 pour Word2Vec). Dans le cas de la classification binaire, il n'y a qu'un seul neurone à la couche de sortie. Dans le cas de la classification multiclasse, le nombre de neurones de la couche de sortie est égal au nombre de classes (ici, quatre), une fonction softmax est alors utilisée.

Le choix des paramètres a été fait de façon empirique, en choisissant ceux qui offraient les meilleures performances.

3.4.1 Classification binaire

Pour la classification binaire, nous allons créer un classifieur par type de relation. Chacun de ces classifieurs sera ensuite entraîné à détecter ce type de relation. Nos données d'entrée sont les vecteurs calculés lors de l'étape précédente. Ainsi, pour chaque vecteur, ces classifieurs retournerons chacun `Vrai` ou `Faux`, suivant la présence ou l'absence d'une instance de la relation à laquelle chaque classifieur est associé. Pour l'entraînement, nous avons pour chaque classifieur sélectionné aléatoirement autant d'instances positives que d'instances négatives parmi le corpus d'entraînement, et ce, de façon à avoir le plus d'instances possibles tout en gardant l'équilibre. Ainsi à l'issue de l'apprentissage, chaque classifieur devrait être en mesure d'identifier une relation donnée. Par exemple, dans le cas du classifieur pour la relation de *conception*, nous allons entraîner ce classifieur avec 1686 instances de relations de *conception* étiquetées `Vrai` et 1686 instances étiquetées `Faux`, sélectionnées aléatoirement parmi les instances n'appartenant pas à la relation de *conception*.

Étant donné que dans cette étude nous avons quatre relations, nous avons donc quatre classifieurs binaires.

3.4.2 Classification multiclasse

Pour la classification multiclasse, nous allons créer un seul et unique classifieur. Ce classifieur sera entraîné de façon à détecter nos différents types de relation. Ainsi, pour un vecteur donné, ce classifieur devra nous renvoyer une des quatre relations.

Là encore, nous équilibrons le jeu de données pour l'entraînement. Ainsi, nous sélectionnons aléatoirement 164 instances de chacune de nos quatre relations (164 étant le nombre d'instances de relation de *décoration*, notre relation la moins peuplée). Si nous avons choisi d'équilibrer nos données, c'est parce que nous avons observé que cela nous permet d'obtenir de meilleurs résultats, en particulier pour les relations les moins représentées.

3.4.3 Classification hiérarchique

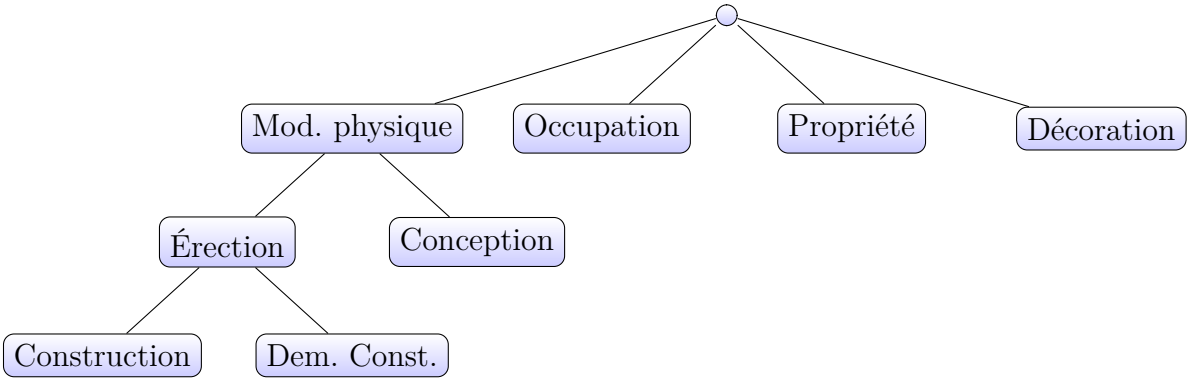

Figure 3.1 Ébauche de taxonomie

Sachant que l'extracteur que nous devions développer devait servir, entre autres, au peuplement d'une ontologie, nous avons expérimenté en ce sens une approche taxonomique. Après avoir manuellement annoté des données pour la construction de notre jeu de test, nous avions donc également à notre disposition d'autres types de relations, comme expliqué précédemment. Nous avions alors remarqué que certaines relations partageaient un vocabulaire proche, notamment les relations liées à l'érection d'un bien immobilier : les relations de *conception*, de *construction*, et de *demande de construction*. Nous avons donc établi une ébauche de taxonomie en regroupant des relations sur plusieurs niveaux. L'arbre de la figure 3.1 présente la taxonomie alors utilisée. Chaque nœud et chaque feuille représente une relation, et possède donc un classifieur binaire qui doit définir si une instance appartient ou non à ladite relation. Dans le cas d'un nœud, si une instance appartient à la relation associée à ce nœud (c'est-à-dire si le classifieur retourne `Vrai`), alors cette instance sera également testée par les nœuds fils, et ce, jusqu'à atteindre une feuille. La taxonomie de classifieur a été entièrement réalisée par nos soins.

Pour mieux illustrer le fonctionnement, considérons l'exemple suivant : « Thomas Baillairgé a construit l'édifice. ». Nous sommes donc en présence d'une relation de *construction*, et qui est aussi, d'après notre taxonomie, une relation de *modification physique* et une relation d'*érection*. Le premier niveau de classifieurs va donc déterminer si la relation appartient à une des quatre catégories suivantes : *modification physique*, *décoration*, *occupation* ou *propriété*. Supposons donc que tout se passe bien, le nœud permettant de détecter la *modification physique* va retourner `Vrai` et les autres nœuds `Faux`. Alors cette instance sera une instance de relation de *modification physique*. Or, cette relation a des « relations filles » : nous allons donc descendre au deuxième niveau de la taxonomie pour détecter si cette relation de *modification physique* est une relation d'*érection* ou de *conception*. Supposons maintenant que le classifieur

d'*érection* se trompe et renvoie `Faux` et que le classifieur de *conception* renvoie `Vrai`. Cette dernière n'ayant pas de « relation fille », l'algorithme prend fin, et il a donc été déterminé que « Thomas Baillairgé » a modifié physiquement le bâtiment en participant à sa conception. Ce qui n'est pas exact, mais pas complètement faux non plus : le PG a effectivement participé à la modification physique du bâtiment.

L'idée derrière cette approche est la suivante : il est plus simple de différencier deux relations sémantiquement proches si elles ne sont que toutes les deux, plutôt que si elles sont au milieu des autres. Bien évidement, nous n'avons pas pu prendre notre jeu d'entraînement pour tester cette approche par taxonomie, étant donné le faible nombre de classes qu'il contient. Nous sommes donc partis de notre jeu de tests, auquel nous avons rajouté des instances de relations de *propriété*, de *demande de construction*, et de *décoration*. Ainsi, notre jeu de données pour tester notre approche par taxonomie compte :

— 167 instances de relation de *conception* ;
— 96 instances de relation d'*occupation* ;
— 225 instances de relation de *construction* ;
— 174 instances de relation de *décoration* ;
— 120 instances de relation de *propriété* ;
— 47 instances de relation de *demande de construction*.

Du point de vue de l'entraînement, nous avons suivi la taxonomie et entraîné niveau par niveau, en équilibrant pour chaque classifieur les données, de façon à ce que chaque classifieur ait autant d'exemples positifs que d'exemples négatifs. Ainsi, pour entraîner le classifieur de la relation de *décoration*, nous avons fourni au classifieur 174 instances de *décoration* étiquetées `Vrai`, ainsi que 174 instances choisies aléatoirement parmi les relations de *propriété*, *occupation*, et *modification physique* étiquetées `Faux` (cette dernière relation regroupant donc les relations de *construction*, *conception* et *demande de construction*). Le classifieur de la relation de *conception*, quant à lui, sera entraîné avec 167 instances de *conception* étiquetées `Vrai` et 167 instances de relations choisies aléatoirement parmi les relations d'*érection*. Notons que pour nos résultats, nous allons effectuer une validation croisée stratifiée à 10 échantillons (« stratifiée » signifiant ici que chaque échantillon a les mêmes proportions d'instances de chaque relation que celles des données initiales), les chiffres donnés en exemple ci-dessus le sont donc à titre indicatif.

CHAPITRE 4 RÉSULTATS

Dans ce chapitre, nous présentons nos différents résultats. Nous commençons par choisir un modèle Word2Vec (Skip-Gram ou Continuous Bag of Words), puis une taille de contexte. Ensuite, nous effectuons des tests sur la classification binaire, la classification multiclasse et l'approche par taxonomie. Pour chacune de ces classifications, nous avons tout d'abord effectué une validation croisée à dix échantillons sur notre jeu d'entraînement. Nous avons donc pris notre jeu d'entraînement préalablement équilibré puis nous l'avons partagé en dix échantillons de façon à respecter les proportions entre les instances des différentes classes. Nous avons ensuite pris 9 échantillons pour l'entraînement et l'échantillon restant pour évaluer les performances ; et ce, avec les dix combinaisons possibles d'échantillons. Nous avons ensuite calculé la moyenne de chaque métrique sur ces dix combinaisons. Pour les classifications binaires et multiclasse, nous avons ensuite vérifié si les tendances étaient les mêmes sur notre jeu de test (nous n'avons en effet pas de jeu de test pour l'approche par taxonomie, faute de données). Pour nos résultats sur les jeux d'entraînement avec ces deux classifications, nous avons procédé à une analyse post-hoc (test de Tukey à 5%) afin de déterminer si les différences entre nos différentes méthodes étaient significatives. À la fin de ce chapitre, nous discuterons de nos résultats.

Dans les sections qui vont suivre, les abréviations suivantes seront utilisées : Machine à Vecteur de Support (SVM), Perceptron Multi-couches (MLP), TF-IDF, Word2Vec (W2V) et Word2Vec pondéré (W.W2V). Afin de comparer nos résultats, nous utilisons quatre métriques différentes : la précision, le rappel, la F-mesure et l'accuracy. Ainsi, pour une relation R donnée, nous définissons :

- la précision, comme étant le rapport entre le nombre d'instances correctement classées dans la relation R sur le nombre d'instances classées dans la relation R. La précision nous permet donc de connaître la proportion d'instances identifiées comme appartenant à la relation R qui ont été correctement assignées à la relation R ;
- le rappel, comme étant le rapport entre le nombre d'instances correctement classées dans la relation R sur le nombre d'instances de la relation R. Le rappel nous permet donc de connaître la proportion d'instances de R qui ont été assignées à la relation R ;
- la F-mesure, comme étant la moyenne harmonique de la précision et du rappel, soit : $F = 2 \times \frac{precision \times rappel}{precision + rappel}$. La F-mesure nous permet de mesurer si le système est capable de d'identifier correctement le plus d'instances de la relation R possible tout en gardant un bruit faible.
- l'accuracy, comme étant le nombre d'instances correctement classées sur le nombre

d'instances total. Dans le cas de la classification binaire, nous avons une valeur d'accuracy par relation, puisque nous avons un classifieur par relation. En revanche, nous n'avons qu'une valeur globale d'accuracy pour la classification multiclasse. L'accuracy va nous permettre de mesurer si le système parvient à maximiser les vrais positifs et les vrais négatifs.

4.1 Détermination du modèle Word2Vec

Pour déterminer quel modèle Word2Vec nous allons choisir, nous avons utilisé notre modèle préalablement construit avec l'architecture CBOW (Schöch (2016)) ainsi que celui construit par nos soins avec l'architecture SG. Pour rappel, ces deux modèles ont été construits à partir des données de Wikipédia-Fr et de la même façon (nous avons, pour nous en assurer, utilisé le même code que Schöch (2016), en changeant évidement le type d'architecture). Les tableaux 4.1 et 4.2 rassemblent ces résultats. Les abréviations suivantes y sont utilisées : SG pour *Skip-Gram*, CBOW pour *Continuous Bag of Words*, et W pour pondéré (*Weighted*) par TF-IDF avant le calcul de la moyenne des vecteurs Word2Vec (voir section 3.3.2 pour plus de détails). Les deux architectures sont à chaque fois comparées deux à deux pour les différentes méthodes, à savoir Word2Vec simple et Word2Vec pondéré par TF-IDF. La meilleure des valeurs pour les deux architectures est alors mise en gras.

Comme nous pouvons le constater, il y a assez peu de différences entre les deux architectures, et ces différences ne sont pas significatives. Nous ne pouvons donc rien en conclure quant au choix d'une architecture. Dans la suite de ce travail, nous allons donc utiliser l'architecture CBOW, qui semble obtenir des résultats légèrement supérieurs à ceux de l'architecture SG.

4.2 Détermination du contexte

Les tableaux 4.3 et 4.4 présentent les moyennes de F-mesure pour les différentes méthodes ; et ce, en prenant comme contexte les phrases ou une fenêtre de mots. Les résultats pour les autres métriques présentant les mêmes tendances, ils ont été mis en annexe, afin de ne pas surcharger cette partie. Comme on peut le constater, nous obtenons les meilleurs résultats en choisissant le contexte comme étant les phrases dans lesquelles apparaissent les PG plutôt qu'un nombre de mots autour desdites mentions (à l'exception de la relation de *décoration* pour la classification binaire). C'est pourquoi nous allons nous intéresser davantage à la première méthode. Une explication à cette différence de performances sera fournie dans la partie « Discussion ».

Tableau 4.1 Comparaison entre les architectures Skip-Gram (SG) et Continuous Bag of Words (CBOW) pour la classification binaire sur le jeu d'entraînement

F-mesure									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	0,881	**0,895**	**0,883**	0,835		**0,911**	0,897	**0,893**	0,866
occupation	0,914	**0,932**	**0,936**	0,930		**0,945**	0,944	0,916	**0,930**
conception	0,901	**0,918**	**0,911**	0,902		**0,926**	0,919	**0,910**	0,902
décoration	0,953	**0,961**	**0,948**	0,923		0,956	**0,961**	0,960	**0,964**
Précision									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	0,890	**0,910**	**0,901**	0,879		**0,917**	0,907	**0,901**	0,867
occupation	0,925	**0,931**	**0,927**	0,912		**0,946**	0,944	0,923	**0,927**
conception	0,898	**0,911**	**0,914**	0,912		**0,938**	0,910	**0,913**	0,911
décoration	0,964	**0,971**	0,953	**0,980**		**0,972**	0,971	**0,966**	**0,966**
Rappel									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	0,874	**0,881**	**0,867**	0,798		**0,908**	0,890	**0,888**	0,868
occupation	0,904	**0,934**	0,946	**0,949**		0,943	**0,946**	0,911	**0,933**
conception	0,904	**0,925**	**0,909**	0,893		0,920	**0,927**	**0,907**	0,893
décoration	0,944	**0,951**	**0,945**	0,872		0,944	**0,951**	0,957	**0,963**
Accuracy									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	0,881	**0,900**	**0,885**	0,843		**0,911**	0,897	**0,896**	0,865
occupation	0,915	**0,932**	**0,935**	0,928		**0,944**	**0,944**	0,917	**0,929**
conception	0,900	**0,915**	**0,911**	0,903		**0,927**	0,918	**0,910**	0,903
décoration	0,953	**0,965**	**0,948**	0,927		0,957	**0,960**	0,960	**0,963**

Tableau 4.2 Comparaison entre les architectures Skip-Gram (SG) et Continuous Bag of Words (CBOW) pour la classification multiclasse sur le jeu d'entraînement

F-mesure									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	**0,730**	0,726	**0,712**	0,706		**0,782**	0,737	**0,707**	0,706
occupation	0,789	**0,798**	**0,825**	0,809		0,812	**0,818**	0,776	**0,813**
conception	0,810	**0,832**	**0,901**	0,787		**0,851**	0,840	0,782	**0,794**
décoration	0,888	**0,901**	0,790	**0,917**		0,908	**0,912**	0,901	**0,917**
Précision									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	0,765	**0,771**	**0,739**	0,712		**0,810**	0,784	**0,722**	0,712
occupation	**0,806**	0,804	**0,812**	0,809		**0,807**	0,803	0,781	**0,809**
conception	0,768	**0,789**	0,814	**0,815**		**0,845**	0,837	0,811	**0,815**
décoration	0,902	**0,918**	0,903	**0,912**		0,914	**0,921**	0,896	**0,912**
Rappel									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
construction	**0,703**	0,676	0,697	**0,713**		**0,757**	0,707	0,710	**0,713**
occupation	0,774	**0,791**	**0,842**	0,817		0,824	**0,834**	0,775	**0,817**
conception	0,865	**0,879**	**0,903**	0,774		**0,866**	0,854	**0,778**	0,774
décoration	0,877	**0,884**	0,791	**0,922**		**0,908**	0,903	0,909	**0,922**
Accuracy									
	SVM					MLP			
Relation	CBOW	SG	W.CBOW	W.SG		CBOW	SG	W.CBOW	W.SG
toutes	0,805	**0,817**	**0,808**	0,806		**0,839**	0,831	**0,793**	0,751

Tableau 4.3 Classification binaire : F-mesure (sur le jeu d'entraînement)

F-mesure pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,908**	,771	,817	,838	,855	,837	,850	,860	,858	,861	,859	,856	,859	,845	,858
W2V-SVM	**,881**	,719	,771	,810	,816	,799	,808	,823	,829	,810	,799	,811	,821	,814	,835
W.W2V-SVM	**,883**	,715	,788	,812	,825	,800	,819	,823	,843	,832	,839	,823	,821	,824	,826
TF-IDF-MLP	**,908**	,769	,805	,834	,848	,834	,843	,838	,856	,860	,867	,872	,861	,862	,872
W2V-MLP	**,911**	,768	,833	,839	,842	,839	,844	,855	,854	,846	,852	,852	,843	,858	,853
W.W2V-MLP	**,893**	,767	,826	,833	,833	,836	,824	,837	,842	,839	,834	,840	,833	,825	,825
F-mesure pour la relation d'occupation															
TF-IDF-SVM	**,947**	,862	,895	,897	,910	,908	,907	,911	,908	,900	,902	,898	,900	,897	,896
W2V-SVM	**,914**	,831	,847	,851	,862	,865	,862	,875	,867	,871	,872	,864	,863	,861	,868
W.W2V-SVM	**,936**	,837	,862	,876	,892	,883	,885	,881	,883	,882	,878	,872	,863	,868	,866
TF-IDF-MLP	**,931**	,830	,860	,873	,872	,875	,882	,886	,879	,881	,878	,866	,875	,872	,877
W2V-MLP	**,945**	,874	,894	,905	,913	,902	,901	,906	,903	,901	,906	,886	,895	,890	,884
W.W2V-MLP	**,916**	,871	,892	,890	,894	,884	,884	,880	,877	,871	,868	,867	,869	,864	,856
F-mesure pour la relation de conception															
TF-IDF-SVM	**,934**	,793	,837	,859	,862	,868	,859	,856	,851	,845	,843	,843	,842	,836	,837
W2V-SVM	**,901**	,805	,824	,834	,838	,835	,828	,815	,822	,816	,805	,804	,798	,798	,801
W.W2V-SVM	**,911**	,801	,824	,828	,840	,826	,825	,816	,815	,813	,807	,802	,802	,802	,802
TF-IDF-MLP	**,887**	,785	,802	,817	,813	,808	,803	,800	,800	,808	,796	,799	,797	,796	,784
W2V-MLP	**,926**	,817	,836	,853	,862	,858	,858	,857	,853	,841	,845	,844	,836	,838	,838
W.W2V-MLP	**,910**	,818	,841	,844	,854	,843	,841	,834	,832	,827	,815	,815	,814	,817	,812
F-mesure pour la relation de décoration															
TF-IDF-SVM	,957	,801	,867	,915	,943	,961	**,975**	**,975**	,972	,963	,969	,972	,964	,966	,963
W2V-SVM	,953	,770	,916	,918	,938	,964	,943	,955	,947	,952	**,976**	,969	**,976**	,967	,958
W.W2V-SVM	,948	,791	,922	,941	,922	,961	,957	,951	,951	,963	,957	**,965**	**,965**	,963	,960
TF-IDF-MLP	,946	,795	,841	,910	,939	,957	,957	,975	,963	,969	,976	**,979**	,973	,973	,969
W2V-MLP	,956	,815	,920	,925	,927	,955	,951	,949	,957	,970	,976	,976	**,979**	,963	,962
W.W2V-MLP	,960	,824	,922	,945	,941	,955	,950	,941	,953	**,966**	,963	,965	,962	,960	,963

Tableau 4.4 Classification multiclasse : F-mesure (sur le jeu d'entraînement)

F-mesure pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,736**	,545	,618	,660	,643	,616	,673	,669	,622	,621	,643	,642	,639	,641	,605
W2V-SVM	**,704**	,517	,546	,594	,583	,560	,577	,557	,607	,613	,594	,572	,613	,580	,560
W.W2V-SVM	**,718**	,487	,531	,578	,555	,593	,608	,590	,568	,581	,587	,573	,591	,596	,589
TF-IDF-MLP	**,679**	,514	,625	,621	,660	,624	,619	,605	,572	,602	,619	,589	,620	,604	,602
W2V-MLP	**,737**	,590	,602	,677	,613	,601	,648	,614	,618	,638	,615	,632	,624	,597	,618
W.W2V-MLP	**,697**	,575	,576	,614	,585	,615	,648	,612	,601	,594	,585	,580	,556	,577	,598
F-mesure pour la relation d'occupation															
TF-IDF-SVM	**,842**	,629	,708	,767	,795	,782	,800	,793	,778	,809	,780	,755	,786	,771	,778
W2V-SVM	**,798**	,646	,746	,725	,756	,731	,705	,714	,735	,733	,677	,655	,696	,681	,690
W.W2V-SVM	**,817**	,624	,702	,744	,735	,748	,769	,747	,726	,724	,735	,742	,742	,722	,733
TF-IDF-MLP	**,778**	,552	,639	,685	,763	,711	,732	,716	,698	,727	,714	,705	,711	,711	,701
W2V-MLP	**,805**	,715	,782	,768	,777	,757	,789	,781	,732	,766	,717	,697	,712	,715	,736
W.W2V-MLP	**,789**	,667	,724	,750	,767	,763	,766	,745	,723	,690	,703	,709	,679	,670	,716
F-mesure pour la relation de conception															
TF-IDF-SVM	**,864**	,660	,656	,723	,724	,714	,726	,712	,697	,673	,711	,691	,701	,704	,697
W2V-SVM	**,805**	,605	,644	,693	,643	,609	,616	,603	,598	,645	,605	,594	,613	,616	,580
W.W2V-SVM	**,783**	,563	,641	,639	,647	,620	,620	,649	,600	,602	,613	,626	,599	,587	,602
TF-IDF-MLP	**,742**	,573	,571	,639	,660	,633	,648	,614	,604	,621	,662	,630	,621	,635	,647
W2V-MLP	**,832**	,678	,732	,747	,669	,660	,685	,670	,643	,640	,628	,611	,634	,616	,620
W.W2V-MLP	**,790**	,626	,707	,650	,679	,628	,622	,635	,631	,569	,582	,573	,560	,523	,574
F-mesure pour la relation de décoration															
TF-IDF-SVM	**,926**	,589	,690	,768	,805	,811	,861	,838	,854	,841	,839	,858	,842	,834	,817
W2V-SVM	**,880**	,551	,684	,766	,783	,800	,838	,814	,842	,800	,814	,831	,817	,811	,819
W.W2V-SVM	**,899**	,537	,715	,776	,808	,836	,821	,864	,860	,838	,845	,840	,838	,853	,834
TF-IDF-MLP	**,890**	,579	,669	,742	,787	,808	,835	,798	,810	,801	,806	,813	,819	,800	,793
W2V-MLP	**,891**	,625	,734	,808	,805	,847	,864	,847	,825	,815	,817	,858	,829	,820	,823
W.W2V-MLP	**,891**	,637	,696	,784	,800	,836	,853	,856	,827	,819	,840	,863	,844	,835	,818

4.3 Classification binaire

4.3.1 Résultats sur le jeu d'entraînement par validation croisée

Tableau 4.5 Résultats par relation avec le jeu de données d'entraînement (classification binaire).

F-mesure						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	0,905	0,881	0,883	0,902	**0,918**	0,885
occupation	$^{+}$**0,946**	0,914	$^{+}$0,936	$^{+}$0,933	$^{+}$0,941	0,935
conception	$^{+}$**0,935**	0,901	0,911	0,885	$^{+}$0,922	0,911
décoration	**0,959**	0,953	0,948	0,944	**0,959**	0.948
Précision						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	**0,939**	0,890	0,901	0,913	0,926	0,898
occupation	0,927	0,925	0,827	0,929	**0,944**	0,926
conception	$^{+}$**0,950**	0,898	0,914	0,887	$^{+}$0,925	0,911
décoration	**0,982**	0,964	0,953	0,958	0,965	0,960
Rappel						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	0,874	0,874	0,867	0,892	**0,912**	0,884
occupation	$^{+}$**0,966**	0,904	$^{+}$0,946	0,936	$^{+}$0,939	0,915
conception	$^{+}$**0,921**	$^{+}$0,904	$^{+}$0,909	0,883	$^{+}$**0,921**	$^{+}$0,907
décoration	0,939	0,944	0,945	0,939	0,951	**0,957**
Accuracy						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	0,909	0,881	0,885	0,904	**0,917**	0,892
occupation	$^{+}$**0,945**	0,915	$^{+}$0,935	$^{+}$0,932	$^{+}$0,944	$^{+}$0,923
conception	$^{+}$**0,936**	0,900	0,911	0,885	$^{+}$0,922	0,910
décoration	**0,960**	0,953	0,948	0,948	0,957	0,957

Le tableau 4.5 présente les moyennes des différentes métriques pour les différentes méthodes sur les dix échantillons de la validation croisée. Nous présentons ces résultats en plus de ceux de notre jeu de tests puisque nous savons que ces deux jeux de données sont assez différents, comme cela a été évoqué dans la section précédente. Pour chaque relation, la meilleure performance pour la métrique considérée est mise en gras. Les meilleurs méthodes qui forment

un groupe non significativement distincts sont indiquées par le symbole $^{+}$. Lorsqu'aucune n'est identifiée par le symbole $^{+}$, cela signifie qu'aucune ne se distingue statistiquement des autres.

En termes de F-mesure

On observe les meilleurs résultats pour TF-IDF-SVM et W2V-MLP. Ces deux méthodes ne semblent pas se distinguer de façon significative. Si l'on compare nos résultats par relation, nous obtenons les meilleures performances pour la relation de *décoration* et les moins bonnes pour la relation de *construction*.

En termes de précision

On observe là encore les meilleurs résultats pour TF-IDF-SVM et W2V-MLP, mais cette fois-ci les différences semblent un peu plus marquées bien que non significatives. Si l'on compare nos résultats par relation, nous observons là encore les meilleurs résultats pour la relation de *décoration* et les moins bons pour la relation de *construction*.

En termes de rappel

Sur une partie des relations, TF-IDF-SVM et MLP-W2V se distinguent des autres méthodes ; en revanche, pour la relation de *décoration*, toutes les méthodes sont assez proches et affichent un rappel entre 0,94 et 0,96 (pour W.W2V-MLP). Si l'on compare cette fois-ci nos résultats par relation, c'est pour les relations d'*occupation* et de *décoration* que nous observons le meilleur rappel, et pour la relation de *construction* que le rappel est au plus bas.

En termes d'accuracy

Tout comme pour la F-mesure, nous obtenons les meilleurs résultats avec TF-IDF-SVM et W2V-MLP, sans grandes différences entre les deux méthodes. Si l'on compare nos résultats par relations, nous arrivons aux mêmes conclusions : les performances sont les plus basses pour la relation de *construction* et les plus élevées pour la relation de *décoration*.

En résumé

Dans l'ensemble les résultats sont très bons (souvent au-dessus de 0,9). Nous obtenons les meilleures performances avec TF-IDF-SVM, suivi de près par W2V-MLP, voire dépassé dans

quelques cas. La relation de *décoration* semble la plus facile à détecter tandis que la relation de *construction* semble l'être beaucoup moins.

4.3.2 Résultats sur le jeu de test

Tableau 4.6 Résultats par relation avec le jeu de données de test (classification binaire).

F-mesure						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	**0,797**	0,752	0,738	0,736	0,770	0,743
occupation	0,692	0,718	0,759	**0,765**	0,730	0,698
conception	**0,844**	0,794	0,802	0,680	0,793	0,747
décoration	0,248	0,224	**0,253**	0,178	0,232	**0.253**
Précision						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	**0,792**	0,684	0,689	0,696	0,709	0,673
occupation	0,541	0,591	**0,638**	0,633	0,590	0,556
conception	**0,778**	0,704	0,710	0,584	0,706	0,633
décoration	**0,147**	0,126	0,145	0,099	0,131	0,145
Rappel						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	0,803	0,834	0,794	0,780	**0,843**	0,830
occupation	0,958	0,917	0,938	**0,969**	0,958	0,938
conception	**0,922**	0,910	0,922	0,814	0,904	0,910
décoration	0,789	**1,000**	**1,000**	0,842	**1,000**	**1,000**
Accuracy						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	**0,821**	0,758	0,752	0,754	0,780	0,749
occupation	0,839	0,864	**0,888**	**0,888**	0,866	0,847
conception	0,888	0,845	**0,985**	0,749	0,845	0,798
décoration	**0,821**	0,741	0,780	0,709	0,752	0,780

Les résultats sur le jeu de test sont donnés à la table 4.6. Dans l'ensemble, nous obtenons des performances moindres par rapport à nos résultats sur notre jeu d'entraînement.

En termes de F-mesure

Nos différentes méthodes ne semblent pas se distinguer de façon notable, mis à part pour les relations de *construction* et de *conception* où l'on observe des performances légèrement plus importantes pour TF-IDF-SVM, sans être significatives. Si l'on considère maintenant nos relations, nous observons des résultats très faibles pour la relation de *décoration*, alors que cette relation affichait les meilleurs résultats avec le jeu de données d'entraînement. Nos meilleurs résultats sont pour la relation de *conception*.

En terme de précision

À l'exception de la relation d'*occupation*, nous obtenons les meilleurs résultats avec TF-IDF-SVM. Là encore, contrairement à ce que nous avions avec nos données d'entraînement, nous avons les résultats les plus bas avec la relation de *décoration*, avec une précision autour de 10-15%, ce qui explique la faible F-mesure. Cette précision très faible s'explique par la très faible proportion d'instances de relation de *décoration* dans notre corpus de tests, qui représente 4,13% de l'ensemble de test, problème que nous n'avions pas à l'entraînement puisque nous avions équilibré les données. Ainsi, si nous avons environ 10% de nos instances de *construction* confondues avec la relation de *décoration*, cela représente déjà 22 instances, soit le même effectif que celui de nos instances de *décoration*. Les matrices de confusions présentées à la section 4.6 illustrent ce problème. La relation d'*occupation* affiche également des résultats assez faibles, aux alentours de 55-64%, contre 70-80% pour les deux autres relations.

En terme de rappel

Cette fois-ci, nous obtenons des résultats comparables (et même meilleurs parfois) à ceux obtenus avec notre jeu d'entraînement. Comme avec ces derniers, nous obtenons un meilleur rappel pour la relation de *décoration*, et un moins bon rappel pour la relation de *construction*. Avec W2V et W.W2V, nous obtenons un rappel de 100% pour la relation de *décoration*. Là encore, il est assez difficile de déterminer quelle est la meilleure méthode.

En terme d'accuracy

Pour les relations de *construction* et de *décoration*, les meilleures performances sont avec TF-IDF-SVM. Pour les relations d'*occupation* et de *conception*, c'est avec W.W2V-SVM que nous obtenons les meilleurs performances.

Pour résumer

Dans l'ensemble nous obtenons des performances inférieures à celles obtenues avec notre jeu de données d'entraînement. Il semblerait que notre approche soit trop sensible et pas assez spécifique pour détecter les relations de *décoration*.

4.4 Classification multiclasse

4.4.1 Résultat sur le jeu d'entraînement par validation croisée

Tableau 4.7 Résultats par relation avec le jeu de données d'entraînement (classification multi-classes).

F-mesure						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	$^{+}$0,758	$^{+}$0,730	$^{+}$0,712	0,674	$^{+}$**0,782**	$^{+}$0,707
occupation	**0,840**	0,789	0,825	0,761	0,812	0,776
conception	0,842	0,810	**0,901**	0,755	0,851	0,782
décoration	**0,925**	0,888	0,790	0,909	0,908	0.901
Précision						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	$^{+}$**0,847**	$^{+}$0,765	$^{+}$0,739	$^{+}$0,740	$^{+}$0,810	0,722
occupation	0,792	0,806	**0,812**	0,753	0,807	0,781
conception	**0,863**	0,768	0,814	0,744	0,845	0,811
décoration	**0,923**	0,902	0,903	0,896	0,914	0,896
Rappel						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	0,700	0,703	0,697	0,633	**0,757**	0,710
occupation	$^{+}$**0,915**	0,774	$^{+}$0,842	0,781	$^{+}$0,824	0,775
conception	0,828	0,865	**0,903**	0,773	0,866	0,778
décoration	**0,933**	0,877	0,791	0,926	0,908	0,909
Accuracy						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
toutes	$^{+}$**0,844**	$^{+}$0,805	$^{+}$0,808	0,778	$^{+}$0,839	$^{+}$0,793

Le tableau 4.7 présente les moyennes des différentes métriques pour les différentes méthodes sur les dix échantillons de la validation croisée. Les conventions utilisées sont les mêmes que

pour les précédents tableaux.

En termes de F-mesure

On peut difficilement conclure sur le choix d'une meilleure méthode. Toutefois, il semblerait que TF-IDF-SVM présente le plus souvent les meilleurs résultats, bien qu'il n'y ait pas vraiment de différence significative avec les autres. Si l'on regarde maintenant nos relations, la relation de *décoration* affiche les meilleures performances, tandis que la relation de *construction* affiche les plus basses : on retrouve la tendance que nous avions avec la classification binaire.

En termes de précision

Cette fois-ci, TF-IDF-SVM semble se démarquer davantage des autres méthodes (sauf pour la relation d'*occupation* où elle est dépassée par W.W2V-SVM). Néanmoins, les différences ne sont toujours pas statistiquement significatives. On observe également que TF-IDF-MLP et W.W2V-MLP affichent les moins bons résultats. De la même façon que dans la classification binaire, nous obtenons les meilleurs résultats pour la relation de *décoration*. Les autres relations, mis à part quelques exceptions, se démarquent moins entre elles et affichent des performances similaires.

En termes de rappel

À l'instar de la classification binaire, nous observons les meilleurs résultats pour TF-IDF-SVM et W2V-MLP (exception faite de la relation de *conception*). Là encore, les meilleurs résultats sont obtenus pour la relation de *décoration* tandis que les bons sont pour la relation de *construction*.

En termes d'accuracy

Là encore, les meilleurs résultats sont obtenus avec TF-IDF-SVM, suivie de près par W2V-MLP, comme c'était le cas pour la classification binaire.

En résumé

Nous obtenons le plus souvent les meilleurs résultats avec TF-IDF-SVM. La méthode W2V-MLP obtient également de bonnes performances. Dans l'ensemble, nous observons les mêmes tendances que pour la classification binaire.

4.4.2 Résultats sur le jeu de test

Tableau 4.8 Résultats par relation avec le jeu de données de test (classification multiclasse).

F-mesure						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	**0,782**	0,729	0,725	0,765	0,737	0,710
occupation	0,771	0,735	0,747	0,712	**0,777**	0,716
conception	**0,823**	0,778	0,770	0,716	0,793	0,780
décoration	**0,582**	0,421	0,436	0,492	0,557	0.464
Précision						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	**0,897**	0,832	0,853	0,808	0,840	0,738
occupation	0,650	0,689	0,694	0,643	**0,702**	0,351
conception	**0,839**	0,743	0,720	0,688	0,784	0,656
décoration	0,471	0,316	0,333	0,357	0,425	**0,835**
Rappel						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
construction	0,693	0,648	0,630	0,587	**0,702**	0,617
occupation	**0,948**	0,788	0,808	0,798	0,833	0,788
conception	**0,839**	0,816	0,828	0,747	0,802	0,828
décoration	0,762	0,632	0,632	0,789	**0,810**	0,684
Accuracy						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
toutes	**0,782**	0,720	0,687	0,670	0,764	0,670

Les résultats sont donnés dans le tableau 4.8. Comme pour les classifieurs binaires, on remarque une dégradation des performances, en particulier pour les relations de *décoration* et d'*occupation*. Un survol du tableau semble nous conforter dans l'idée que, comme pour les résultats sur le jeu d'entraînement, TF-IDF-SVM obtient de meilleures performances.

En termes de F-mesure

À l'exception de la relation d'*occupation*, TF-IDF-SVM se démarque plus ou moins nettement des autres méthodes, affichant des performances de 5% à 10% au-dessus. On remarque toutefois que W2V-MLP est assez proche. Si l'on regarde maintenant nos relations, la rela-

tion de *décoration* affiche les plus basses performances, contrairement à ce que les résultats de l'entraînement laissaient supposer. Nos meilleurs résultats sont avec la relation de *conception*. Ces deux comportements sont similaires à ceux observés avec les résultats des données de tests sur la classification binaire.

En termes de précision

Là encore, nous observons une baisse significative des performances pour la relation de *décoration* pour toutes nos méthodes, à l'exception de W.W2V-MLP, où l'on obtient une précision de 83,5% ; alors que les autres méthodes sont autour de 30-45%. Cette baisse de précision s'explique notamment par le fait que notre jeu de test contient très peu d'instances de la relation de *décoration*, donc dès lors que des instances de *construction* ou autre sont classées par erreur en *décoration*, la quantité de faux positifs pour cette dernière augmente significativement par rapport à la quantité de vrais positifs. Pour les relations de *construction* et de *conception*, nous obtenons les meilleurs résultats avec TF-IDF-SVM, et pour la relation d'*occupation*, nous obtenons le meilleur résultat avec W2V-MLP.

En termes de rappel

En comparant avec nos résultats sur les données d'entraînement, les performances pour la relation de *décoration* ont baissé ; ce qui n'était pas le cas avec la classification binaire (en termes de rappel tout du moins). La relation de *construction* affiche également des performances plus faibles. Les deux autres relations quant à elles affichent des résultats comparables. Nous observons les meilleurs résultats avec TF-IDF-SVM pour les relations de *conception* et d'*occupation*. Pour les relations *construction* et de *décoration*, c'est avec TF-IDF-MLP que nous obtenons les meilleurs résultats.

En termes d'accuracy

Conformément à ce que nous avions avec les données d'entraînement, nous obtenons la meilleure accuracy avec TF-IDF-SVM, suivi de près par W2V-MLP.

Pour résumer

Tout comme avec la classification binaire, nous observons une dégradation importante des performances pour la relation de *décoration*. Dans l'ensemble, c'est avec TF-IDF-SVM que nous obtenons les meilleurs résultats, ce qui était également le cas avec les données d'entraînement.

4.5 Approche Taxonomique

Bien que n'ayant que peu de données, nous avons tout de même réalisé des tests sur notre approche taxonomique avec le jeu de données que nous avons construit à cet effet. Le contexte utilisé est l'ensemble des phrases mentionnant le PG. Le tableau 4.9 présente les résultats obtenus par cette approche pour chaque relation. En plus des six relations de notre jeu de données, nous donnons également les résultats pour les relations « intermédiaires », à savoir les relations d'*érection* et de *modification physique*. Les résultats sont donnés en considérant la taxonomie de classifieurs dans sa globalité, et non en considérant chaque classifieur indépendamment. En d'autres termes, cela signifie par exemple que le rappel pour la relation de *conception* est défini comme étant le nombre de vrais positifs trouvés par le classifieur binaire associé divisé par le nombre total d'instances appartenant à cette relation, et non uniquement le nombre d'instances de cette relation arrivées jusqu'à ce classifieur. Nous prenons donc en compte pour ces métriques les instances de *conception* qui ont été mal classées dès le début et qui n'arrive pas au classifieur de la relation de *conception*.

En termes de F-mesure

On observe une F-mesure très mauvaise pour les relations d'*occupation*, de *propriété*, et de *demande de construction*. Pour les autres relations, la F-mesure se situe aux alentour de 0,8. En ce qui concerne les méthodes, W2V-MLP semble légèrement au-dessus des autres, même si ce n'est pas le cas pour toutes les relations.

En termes de précision

On observe une précision très mauvaise pour les relations d'*occupation*, de *propriété*, et de *demande de construction*, ce qui conduit aux faibles performances de F-mesure sur ces mêmes relations. Ces faibles performances s'expliquent par la faible quantité d'instances appartenant à ces relations. En effet, si les neuf échantillons d'entraînement sont équilibrés avant de procéder à l'entraînement, ce n'est pas le cas pour l'échantillon réservé au test, donc les erreurs des instances des classes les plus nombreuses font grandement baisser la précision pour les classes les moins nombreuses (c'est le même phénomène que nous avions observé dans les résultats précédents sur les jeux de données de tests). Cette fois-ci, c'est avec TF-IDF-SVM que nous obtenons les meilleurs performances, sauf pour les relations d'*occupation*, de *propriété*, et de *demande de construction*, qui ont les meilleurs résultats avec respectivement TF-IDF-MLP, W.W2V-SVM et W2V-MLP.

En termes de rappel

En termes de rappel, nous obtenons de très bons résultats pour les relations d'*occupation*, *propriété*, *décoration* et *modification physique*. Les performances se dégradent ensuite pour la *conception*, l'*érection* et davantage encore pour les relations de *construction* et de *demande de construction*. Ce n'est pas un hasard si l'on remarque que plus l'on descend l'arbre, plus le rappel chute. En effet, comme évoqué plus haut, le rappel est ici calculé non pas par classifieur, mais en considérant l'ensemble de la taxonomie de classifieurs, si bien que s'il y a des instances mal détectées en haut de l'arbre, les nœuds subséquents ne pourrons pas les classer puisque ces instances auront été éliminées par un classifieur plus général. Ainsi, si nous avons une instance de *construction* qui est mal classée par le classifieur de *modification physique*, alors cette instance ne passera jamais par les classifieurs des relations d'*érection* et de *construction*.

En termes d'accuracy

Les résultats sont ici plutôt bons et sont entre 0,8 et 0,9. Les résultats sont assez proches et il est difficile de dire qu'une méthode est meilleure qu'une autre tant les différences semblent peu significatives.

Pour résumer

Nous obtenons avec cette approche des résultats plutôt encourageants. L'avantage de cette approche est qu'elle pourra tirer parti de l'ontologie actuellement en développement au MCC. Un autre intérêt est la possibilité de classifier partiellement certaines instances. Supposons par exemple que nous ayons une instance de relation de *demande de construction*. Cette instance porte a priori sur un contexte contenant des termes liés à la construction, puisque c'est une relation assez proche. Il est donc raisonnable de penser que cette instance sera assez facilement classée en *modification physique* puis en *érection*. Néanmoins, si la différence entre les relation de *construction* et de *demande de construction* est trop subtile, les deux classifieurs associés vont probablement se tromper. Dans l'optique d'une aide à la saisie d'informations, cette approche va donc permettre de restreindre au maximum les choix possibles, en donnant des suggestions pertinentes.

Un inconvénient important de cette approche est qu'à mesure où l'on descend dans l'arbre, des erreurs apparaissent et réduisent peu à peu les performances, comme cela a été expliqué pour le rappel. Il faudrait donc réfléchir à un moyen de récupérer les instances qui sont égarées dans les niveaux supérieurs de l'arbre. Nous n'avons présentement pas assez de données pour

certifier la pertinence de nos résultats, en particulier dans certaines relations présentant peu d'instances, comme c'est le cas avec la relation de *demande de construction*.

Tableau 4.9 Résultats par relation pour l'approche par taxonomie.

F-mesure						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
occupation	0,512	0,520	0,533	**0,544**	0,530	0,533
propriété	0,603	0,647	**0,655**	0,578	0,650	0,636
décoration	**0,824**	0,767	0,807	0,771	0,810	0,795
mod. physique	0,900	0,880	0,895	0,886	**0,910**	0,897
conception	**0,821**	0,778	0,760	0,737	0,786	0,754
érection	0,792	0,767	0,778	0,738	**0,812**	0,779
construction	0,674	0,664	0,582	0,643	**0,711**	0,602
dem. const.	0,312	0,369	0,345	0,264	**0,448**	0,371
Précision						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
occupation	0,353	0,370	0,377	**0,387**	0,377	0,375
propriété	0,445	0,501	**0,505**	0,425	0,500	0,484
décoration	**0,729**	0,663	0,718	0,653	0,714	0,699
mod. physique	**0,944**	0,896	0,913	0,929	0,922	0,903
conception	**0,796**	0,752	0,712	0,725	0,755	0,710
érection	**0,890**	0,822	0,863	0,816	0,859	0,835
construction	**0,881**	0,799	0,825	0,788	0,818	0,778
dem. const.	0,278	0,329	0,271	0,243	**0,381**	0,284
Rappel						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
occupation	**0,947**	0,883	0,913	0,926	0,904	0,924
propriété	**0,942**	0,925	**0,942**	0,908	**0,942**	**0,942**
décoration	**0,949**	0,920	0,932	0,943	0,943	0,931
mod. physique	0,863	0,865	0,879	0,849	**0,901**	0,893
conception	0,851	0,815	0,821	0,761	**0,827**	0,809
érection	0,719	0,723	0,715	0,676	**0,774**	0,734
construction	0,554	0,578	0,463	0,549	**0,635**	0,501
dem. const.	0,405	0,465	0,500	0,300	**0,555**	0,550
Accuracy						
	SVM			MLP		
Relation	tf-idf	w2v	w. w2v	tf-idf	w2v	w. w2v
occupation	0,789	0,810	**0,814**	0,820	0,813	0,813
propriété	0,818	0,854	**0,856**	0,805	0,852	0,841
décoration	**0,917**	0,880	0,907	0,883	0,906	0,899
mod. physique	**0,904**	0,881	0,897	0,889	0,811	0,897
conception	0,871	0,858	0,854	0,833	**0,872**	0,856
érection	0,870	0,846	0,856	0,834	**0,877**	0,856
construction	0,821	0,810	0,793	0,795	**0,839**	0,802
dem. const.	0,828	0,829	0,804	0,815	**0,856**	0,815

4.6 Discussion

4.6.1 Analyse des matrices de confusion

Tableau 4.10 Matrices de Confusion par classifieur TF-IDF-SVM (classification binaire)

(a) Conception

	Vrai	Faux
Conception	156	11
Décoration	1	20
Construction	39	186
Occupation	4	92

(b) Décoration

	Vrai	Faux
Conception	19	148
Décoration	18	3
Construction	51	174
Occupation	11	85

(c) Construction

	Vrai	Faux
Conception	31	136
Décoration	12	9
Construction	182	43
Occupation	8	88

(d) Occupation

	Vrai	Faux
Conception	7	160
Décoration	8	13
Construction	58	167
Occupation	92	4

Tableau 4.11 Matrices de confusion par classifieur W2V-MLP (classification binaire)

(a) Conception

	Vrai	Faux
Conception	151	16
Décoration	3	18
Construction	68	157
Occupation	7	89

(b) Décoration

	Vrai	Faux
Conception	27	140
Décoration	21	0
Construction	80	145
Occupation	16	80

(c) Construction

	Vrai	Faux
Conception	44	123
Décoration	18	3
Construction	191	34
Occupation	13	83

(d) Occupation

	Vrai	Faux
Conception	8	159
Décoration	6	15
Construction	48	177
Occupation	92	4

L'analyse post-hoc a révélé qu'il existait quelques fois des différences significatives entre les différentes méthodes, mais tout porte à croire qu'il en existe également entre les relations. Afin de mieux comprendre pourquoi nous obtenons de telles différences entre nos relations, nous avons analysé les matrices de confusion sur nos données de tests en classification binaire et multiclasse. Afin de ne pas surcharger ce document, nous avons choisi de ne représenter que les matrices de confusion des méthodes TF-IDF-SVM et W2V-MLP, respectivement, dans les tableaux 4.10 et 4.11. Les lignes de ces matrices représentent les classes réelles tandis que les colonnes représentent les classes prédites. En temps normal, nous aurions aussi *Vrai* et

Faux dans les lignes ; toutefois, afin de mieux faire ressortir les confusions, nous avons séparé les *Faux* en leur classe d'origine. Ainsi, pour la matrice (a) du tableau 4.10 (donc la matrice pour le classifieur de la relation de *conception*), 156 instances de relation de *conception* ont correctement été identifiées ; en revanche, 39 instances de relation de *construction* ont été confondues avec la relation de *conception*.

Ainsi, nous remarquons pour la classification binaire que :

- le classifieur de la relation de *conception* évite sans difficulté la confusion avec la relation de *décoration* et d'*occupation*, mais a un peu plus de difficultés avec la relation de *construction* ;
- le classifieur de la relation de *décoration* présente également des difficultés avec la relation de *construction* ;
- le classifieur de la relation de *construction* a du mal à ne pas la confondre avec la relation de *décoration*, ainsi que de *conception* dans une moindre mesure ;
- le classifieur de la relation d'*occupation* est très performant pour discriminer la relation de *conception* (dans 95% des cas), en revanche, il l'est beaucoup moins pour la relation de *décoration* et de *construction* ;
- En comparant les deux méthodes, on remarque que pour W2V-MLP, les classifieurs ont plus de mal à ne pas confondre la relation de *conception* avec celle de *construction*, et la relation de *décoration* avec celle de *construction* (on le voit particulièrement bien en comparant les matrices (a) et (b) des deux méthodes). Cette différence provient de la proximité sémantique des termes employés dans les instances de ces relations, conduisant à des vecteurs relativement proches dans l'espace vectoriel du modèle *Word2Vec*. Nous n'avons pas ce problème avec *TF-IDF* puisque cette représentation de mot n'utilise pas la sémantique mais uniquement la répartition des mots dans le corpus. Entre les autres relations, il y a assez peu de différences de comportement entre les deux méthodes TF-IDF-SVM et W2V-MLP.

Tableau 4.12 Matrices de confusion pour la classification multiclasse

(a) TF-IDF-SVM

	Conc.	Déco.	Const.	Occ.
Conc.	135	10	13	9
Déco.	1	16	4	0
Const.	21	8	156	40
Occ.	4	0	1	91

(b) W2V-MLP

	Conc.	Déco.	Const.	Occ.
Conc.	134	12	18	3
Déco.	0	17	4	0
Const.	33	7	158	27
Occ.	4	4	8	80

Pour la classification multiclasse, les matrices sont présentées dans le tableau 4.12. On remarque que :

— les instances de la relation de *conception* sont très bien identifiées et rarement confondues avec les autres ;
— les instances de relation de *décoration* sont rarement confondues avec celles de *conception* et d'*occupation* et plus fréquemment avec celles de *construction* ;
— les instances de la relation de *construction* sont le plus souvent confondues avec celles de *conception* et d'*occupation*. Contrairement à ce que l'on a observé avec la classification binaire, il ne semble pas y avoir de confusion avec la relation de *décoration*, dans ce sens-là tout du moins ;
— les instances de relation d'*occupation* sont rarement confondues avec celles de *décoration*, de *construction* et de *conception* ;
— là encore, on remarque une plus grande confusion entre la relation de *conception* et celle de *construction*, et entre la relation de *décoration* et celle de *construction* pour W2V-MLP, mais dans une moindre mesure. Il y a, à part ce dernier point, assez peu de différences de comportement entre les deux méthodes TF-IDF-SVM et W2V-MLP.

Nous pouvons donc en conclure que les relations de *conception* et d'*occupation* sont très différentes dans les représentations de mots utilisées ; de même pour les relations de *conception* et *décoration*. En revanche, la relation de *construction* est assez mal discriminée, et est souvent confondue avec la relation de *décoration* et de *conception* (en particulier pour W2V-MLP). Ceci soulève donc la question suivante : pourquoi la relation de *construction* est-elle souvent confondue avec les autres relations ? Pour y répondre, nous nous sommes donc intéressé plus en détail aux mauvaises détections.

4.6.2 Analyse des mauvaises détections

Pour mieux comprendre l'origine des erreurs de nos classifieurs, nous avons analysé une à une chacune des erreurs des jeux de tests, et ce, pour les classifications binaire et multiclasse, pour les méthodes TF-IDF-SVM et W2V-MLP.

Ainsi, nous avons mis en lumière les points suivants :

— on retrouve sensiblement les mêmes types d'erreur et dans des proportions semblables, peu importe les méthodes et classifications utilisées ;
— dans 40% des erreurs, nous ne sommes pas parvenu à détecter la source de l'erreur : la relation est explicitement mentionnée, il n'y a pas de terme caractéristique d'une autre relation et pourtant le classifieur ne parvient pas à trouver la bonne relation. Par exemple, pour une centrale hydroélectrique [1], notre contexte est « La centrale

1. `http://www.patrimoine-culturel.gouv.qc.ca/rpcq/detail.do?methode=consulter&id=201908&type=bien`

hydroélectrique est construite en 1957 par la Eastern Mining and Smelting Company Limited. » et le PG est la « Eastern Mining and Smelting Company Limited ». La relation est assez explicitement décrite, pourtant notre classifieur multiclasse l'a identifiée comme une relation de *conception* et non de *construction*. Ces erreurs sont possiblement dues au fait que certaines de nos données d'entraînement sont bruitées, comme évoqué dans le chapitre 3.1.1 ;

— dans environ 35% des erreurs, le contexte comporte la mention de plusieurs instances de relation, le plus souvent des relations de *construction* et de *conception* (voir le point ci-après). Ce cas pose problème puisque le contexte présente donc le vocabulaire de plusieurs relations différentes. Or, le classifieur a été entraîné à ne reconnaître qu'une et une seule relation par contexte : il devra donc en choisir une seule. Son choix devrait dépendre du poids qu'il accorde aux mots des différents domaines. Considérons que le contexte contient les verbes « construire » et « occuper », si lors de l'apprentissage le classifieur a appris que « construire » conduit toujours à une relation de *construction* et que parfois « occuper » conduit à une relation d'*occupation*, alors il choisira la relation de *construction*. On pourrait penser que la classification binaire devrait résoudre ce problème, toutefois, ce n'est pas le cas. En effet, en reprenant l'exemple précédent, et en considérant le classifieur de la relation d'*occupation* : ce classifieur a appris que « construire » conduit toujours à une relation de *construction*, donc à `Faux` ; et que parfois « occuper » conduit à une relation d'*occupation*, donc à `Vrai`. Le classifier va donc choisir là encore le terme qui lui paraît le plus discriminant ;

— une grande partie (environ 75%) des confusions entre relations de *conception* et de *construction* provient de la présence d'instances de chacune de ces relations dans la même phrase. On retrouve en effet très souvent le schéma suivant : « X a construit la maison, selon les plans de l'architecte Y » ou un de ses dérivés. Ce type de cas pourrait être facilement traité en utilisant un analyseur syntaxique afin de ne sélectionner que la partie de la phrase contenant le PG. Afin de nous en assurer, nous avons confié chacun de ces cas à l'analyseur syntaxique de Google [2]. Ce dernier a procédé à ladite analyse sans aucune erreur : il a réussi systématiquement à séparer les différentes propositions de la phrase. Ceci nous laisse penser que, dans le futur, il serait possible de traiter ces cas en séparant les contextes. Ainsi, en donnant à l'analyseur : « X a construit Z, selon les plans de Y », nous obtenons l'arbre de syntaxe en figure 4.1. Comme on peut le constater, « X » est détecté comme étant le sujet de « construit » et « Y » comme étant un complément de « plan ». On peut donc supposément séparer de la phrase en « X a construit Z » et « les plans de Y », soit deux parties ne contenant chacune

2. `https://cloud.google.com/natural-language/`

qu'une seule et unique instance de relation ;

— on retrouve souvent les termes *maison*, *demeure*, *résidence* et leurs dérivés dans beaucoup de mauvaises assignations à la relation d'*occupation*. Ces termes semblent donc induire ce type de relation. Une des explications est que ces termes sont souvent des compléments d'objet de verbes liés à l'occupation. Par exemple pour le verbe « occuper » : « John occupe la maison durant plus de vingt ans. ». Dans ce cas-ci, nous sommes effectivement dans un cas d'occupation. Cependant, si nous considérons le contexte suivant : « Le bâtiment aurait été conçu par William Footner (vers 1799-1872).», extrait de la synthèse historique de l'Église anglicane Saint-Paul [3], nos classifieurs ont ici détecté une relation d'*occupation*, alors qu'il s'agit d'une relation de *conception*. « Concevoir » devrait renvoyer directement vers la bonne relation, mais il semblerait que « bâtiment » ait aiguillé le classifieur vers l'*occupation*. De façon analogue, on retrouve également des termes liés à la famille dans les assignations erronées à la relation d'*occupation*. Un autre cas courant d'erreur est la mention de la profession d'une personne logeant dans un bien immobilier. Ainsi, s'il est spécifié que le locataire d'un bien immobilier est un architecte, l'instance sera classée en tant que relation de *conception*, alors qu'il n'en est rien ;

— la confusion entre les relations de *décoration* et de *conception* est bien souvent due à une confusion dans la sémantique. On retrouve en effet des instances de relation de *conception* utilisant des termes liés au domaine artistique, comme c'est le cas dans le contexte de la relation entre « Louis-Zéphirin Gauthier » et « l'Église de La Présentation-de-la-Sainte-Vierge » [4] : « À l'intérieur, la fausse voûte est ornée de caissons dessinés par l'architecte Louis-Zéphirin Gauthier (1842-1922). » On retrouve en effet les termes « ornée » et « dessinés ». Par ailleurs, la profession d'architecte revêt une dimension artistique qui n'est pas à négliger, c'est pourquoi l'on retrouve dans bien des cas des architectes dans la rénovation ou la restauration de pièce d'art, dans les églises notamment. Par ailleurs, les termes liés au style sont sans doute associés à la décoration, alors qu'ils sont parfois présents dans les contextes d'instances de relation de *conception*, entraînant des erreurs de classification ;

— la longueur moyenne des contextes des instances présentant des erreurs est de 23 mots avec un écart type de 12,8 mots, contre 18 mots avec un écart type de 10 mots pour l'ensemble du corpus de tests. Un contexte comportant plus de mots tendrait donc à augmenter le risque d'erreur.

3. `http://www.patrimoine-culturel.gouv.qc.ca/rpcq/detail.do?methode=consulter&id=93508&type=bien#.WzpjRhduOV4`

4. `http://www.patrimoine-culturel.gouv.qc.ca/rpcq/detail.do?methode=consulter&id=92766&type=bien#.Wzp5qBduOV4`

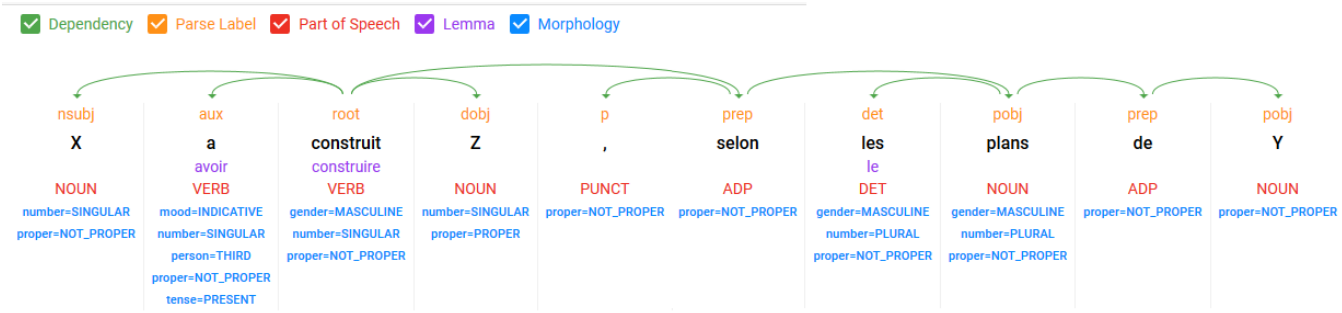

Figure 4.1 Arbre syntaxique fourni par l'analyseur de Google

4.6.3 Le contexte : fenêtre ou phrase ?

Nos résultats nous conduisent à conclure que définir le contexte de la relation comme une phrase amène de meilleures performances que sa définition comme une fenêtre de mots autour de la mention du PG. L'analyse des données nous apporte une explication à ce phénomène : si les termes nous permettant de décider de la relation associée au contexte sont très souvent dans la phrase, ils sont à une position variable, de même que la mention du PG qui peut aussi bien être tout au début de la phrase, au milieu ou à la fin. Ainsi, avec une taille de fenêtre fixe, nous risquons à la fois de manquer des mots qui seront plus loin dans la phrase, mais aussi de prendre des mots qui ne concerne pas la relation en prenant ceux de la phrase suivante ou précédente.

Toutefois, comme nous avons pu le remarquer dans le tableau 4.3, il semblerait que la relation de *décoration* fasse exception et affiche même de meilleurs résultats pour des fenêtres de grande taille (de 6 à 12 mots à gauche et à droite de chaque mention du PG dans la synthèse historique). Une explication à ce phénomène est que lorsqu'il est question de réalisation artistique dans la synthèse historique d'un bien immobilier, la totalité des éléments artistiques du bien immobilier est décrite dans le paragraphe. Par exemple, dans la synthèse historique de la Chapelle Notre-Dame-de-Bon-Secours [5], nous avons :

« Le peintre François-Édouard Meloche (1855-1914) peint un décor en trompe-l'oeil et en grisaille. Il conçoit aussi, en 1892, le campanile de la chapelle aérienne consacrée à la Sainte Famille, orné notamment d'une imposante statue de la Vierge exécutée par Philippe Banlier dit Laperle (1860-1934). Dès le début du XXe siècle, le décor de Meloche est vivement critiqué. En 1908, l'artiste décorateur Delphis-Adolphe Beaulieu (1849-1928) obtient le mandat de refaire un décor intérieur plus clair et plus sobre. »

5. `http://www.patrimoine-culturel.gouv.qc.ca/rpcq/detail.do?methode=consulter&id=96643&type=bien`

Comme on peut le constater, les quatre phrases ci-dessus se suivent et évoquent toutes des relations de *décoration*. Ainsi, si nous prenons une fenêtre de grande taille, nous augmentons les chances d'avoir dans notre contexte de nombreux mots liés au domaine artistique, et donc nous augmentons les chances qu'une instance portant sur un tel contexte soit assignée à une relation de *décoration*.

4.6.4 L'influence du classifieur et de la représentation de mots

Comme nos résultats tendent à le montrer, nous obtenons de meilleurs performances avec TF-IDF-SVM et W2V-MLP. Si l'on doit comparer SVM avec MLP, on constate que MLP semble mieux fonctionner pour Word2Vec alors que SVM obtient de meilleurs résultats avec TF-IDF qu'avec Word2Vec et Word2Vec pondéré. Mais les différences entre les méthodes sont assez peu significatives, si bien que nous ne sommes pas en mesure d'affirmer avec certitude que le choix des classifieurs ait un réel impact sur les performances.

4.6.5 Approche Multiclasse ou Binaire ?

À la vue de nos résultats, nous observons une meilleure précision avec l'approche multi-classe, tandis que l'approche binaire nous donne un meilleur rappel. Le choix de l'approche dépendrait donc de l'aspect que l'on souhaite privilégier.

Cependant, nous n'avons testé ces approches que sur un jeu restreint de relations. Dans le cas d'une mise en production, il y aurait bien plus de relations possibles. On peut donc imaginer que l'approche multiclasse rencontrerait davantage de difficultés avec autant de relations. Par ailleurs, cette approche ne permet pas de donner plusieurs relations possibles pour une seule instance, alors que c'est le cas de l'approche binaire, où l'on aura un résultat par classifieur. Cette dernière approche semble donc être davantage indiquée dans l'optique de réaliser une application suggérant les relations possibles pour une instance donnée. On peut aussi envisager, dans le futur, un système composé de multiples classifieurs binaires. Ce système pourra alors déterminer la relation par un système de vote.

CHAPITRE 5 CONCLUSION

5.1 Synthèse des travaux

Nous avons dans ce projet expérimenté la combinaison de plusieurs méthodes pour arriver à identifier des instances de relations entre biens immobiliers et personnes ou groupes de personnes, et ce, en prenant comme données de départ le nom de la personne ou groupe de personnes et la synthèse historique du bien immobilier. Nous nous intéressons dans ce travail à quatre relations : les relations de *conception*, *décoration*, *construction* et *occupation*. Une étude empirique des données nous a conduit à faire les hypothèses suivantes :

— les relations sont décrites dans la synthèse du bien immobilier lors de la mention de la personne qui y est associée ;

— il est possible d'extraire les relations en faisant appel à des représentations de mots et des algorithmes d'apprentissage supervisé, sans utiliser de bases de connaissances ou d'outils d'analyse syntaxique.

Notre approche se divise en plusieurs parties. Dans un premier temps, pour chaque relation, nous extrayons le contexte autour des mentions du PG dans la synthèse historique du bien immobilier. Ce contexte est soit défini comme étant les mots des phrases dans lesquelles le PG est mentionné, soit une fenêtre de mots autour de chaque mention du PG. Ces listes de mots doivent, d'après notre hypothèse, contenir les informations nécessaires à la détermination de la relation entre le PG et le bien immobilier. Ces listes de mots sont alors transformées en vecteur en utilisant un système de représentation de mot. Là encore, nous avons expérimenté plusieurs méthodes :

— le sac de mots (bag of word), en utilisant la pondération TF-IDF ;

— le plongement lexical en utilisant Word2Vec, en calculant l'isobarycentre des vecteurs obtenus pour chaque mot ;

— le plongement lexical en utilisant Word2Vec, en calculant le barycentre des vecteurs obtenus pour chaque mot et pondérés par le coefficient TF-IDF dudit mot.

Ces vecteurs sont ensuite confiés à un algorithme d'apprentissage supervisé (perceptron multicouche ou séparateur à vaste marge) qui aura pour tâche de classer chaque vecteur en une relation. Trois types de classifications ont été utilisés : une approche binaire, en utilisant un classifieur binaire par relation ; une approche multiclasse, en utilisant un seul classifieur multiclasse pour toutes nos relations ; ainsi qu'une approche par taxonomie, sur un jeu comportant moins de données mais plus de relations différentes.

Dans le meilleur des cas, nous obtenons une F-mesure entre 90% et 95% avec le jeu d'en-

traînement et allant de 25% à 85% avec le jeu de test, suivant la relation. Nous obtenons les meilleurs résultats en utilisant comme contexte les phrases et avec TF-IDF combiné à un séparateur à vaste marge ou Word2Vec combiné à un perceptron multicouche. Nous ne sommes pas vraiment en mesure de dire si la classification multiclasse est meilleure que la classification binaire.

Bien qu'insuffisants pour une mise en production, nos résultats n'en sont pas moins encourageants. En effet, nous n'avons pour l'instant utilisé que des méthodes de représentation de mots et des algorithmes d'apprentissage supervisé. Nous n'avons utilisé ni bases de connaissances du domaine, ni outils d'analyse syntaxique, ni systèmes élaborés de reconnaissance d'entités nommées. Il nous reste donc des voies d'amélioration qui seront discutées dans une section subséquente. Par ailleurs, nos travaux nous ont mené à la construction d'un modèle Word2Vec utilisant l'architecture Skip-Gram. Ce modèle a été entraîné en français sur l'ensemble de Wikipédia-Fr, et fait donc partie des contributions de notre projet.

5.2 Limitations de la solution proposée

Une des premières limitations provient de notre choix initial de ne nous concentrer que sur un jeu restreint de relations, en ne prenant que les relations présentes en plus grand nombre (soit en plus de cinquante exemplaires). Ce choix découle directement de l'état de la base de données alors à notre disposition. En effet, comme expliqué dans l'introduction, cette base de données a évolué au cours du temps et s'est vue dotée de nouvelles relations, comme la relation de *propriété*. Si bien que l'on se retrouve avec une grande quantité de relations classées dans la catégorie *autre*, donc non classées. Le parcours des données pour la création du jeu de test a révélé la présence d'une quantité non négligeable d'instances de relations de *propriété*, de *demande de construction*, de *rénovation* ... et d'autres très certainement cachées dans les *autres*. Un parcours en profondeur de ces sept mille instances permettrait sans nul doute d'ajouter plus de relations à notre approche.

Une autre limitation découlant de l'état actuel de la classification est la présence d'un bruit dans nos données d'entraînement. Ainsi, en construisant le jeu de test, il s'avère qu'entre 10% et 20% des relations sont mal attribuées. Il y a également un flou entre la notion de conception et de décoration, puisque bien souvent, c'est un architecte qui est chargé de la rénovation de décors intérieurs ou d'autres pièces "artistiques". Cette confusion entre la profession du PG et son rôle dans la relation est également présente dans des instances de relation d'*occupation*, où l'on nomme la personne puis précise le métier qu'elle exerce ; ce qui est alors source de confusion dans notre approche.

L'étude des erreurs présentée dans le chapitre précédent met également une limitation importante de notre approche, à savoir la présence de plusieurs PG et donc potentiellement plusieurs relations dans une même phrase. Ce dernier point entraîne les difficultés suivantes :

— pour un même contexte, nous allons apprendre une première fois à notre modèle que ledit contexte correspond à une relation X et une seconde fois qu'il correspond à une relation Y, entraînant potentiellement une confusion entre les deux ;

— lors de la phase de prédiction, le modèle va prédire une des deux relations, en fonction de l'importance qu'il accorde à un terme plutôt qu'un autre. Ainsi, dans nos données de tests, les contextes contenant à la fois des relations de *conception* et de *construction* ont souvent été associés à la relation de *conception*.

Une autre limitation liée au point précédent est la gestion de plusieurs instances de relations entre un même couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$. Ce problème est issu de l'impossibilité de représenter plusieurs relations entre un même couple $\langle \text{PG}, \textit{Bien Immobilier} \rangle$ dans la classification actuelle du Répertoire du Patrimoine Culturel du Québec, si bien que nous n'avons pas ou peu d'exemples pour traiter ces cas-ci avec notre approche. En matière de volume, sur un parcours d'environ 1400 couples, 9% devraient être impliqués dans plusieurs relations (dans 60% des cas, une des relations était une relation de *propriété*).

Enfin, notre approche nécessite de connaître le nom du PG pour pouvoir identifier la relation dans laquelle il est impliqué. Dans l'état actuel des choses, nous n'effectuons pas de reconnaissance d'entité nommée. De plus, nous supposons que le PG est toujours mentionné par son nom complet ou un de ses alias, ce qui n'est pas toujours le cas.

5.3 Améliorations futures

Tout d'abord il faudrait essayer d'entraîner et de tester notre approche sur des données corrigées et non bruitées, notamment pour vérifier si TFIDF-SVM et W2V-MLP ont toujours tendance à fournir les meilleurs résultats. Par ailleurs, plus de données permettraient de constater si les performances sont similaires pour d'autres types de relations, mais aussi de vérifier si les différences de performance entre relations ne sont pas dues à des niveaux de bruits différents entre celles-ci.

Afin de contourner les différentes limitations de notre approche, nous proposons plusieurs axes d'améliorations. Premièrement, l'utilisation d'outils d'analyse syntaxique permettrait de résoudre les problèmes survenant lors de la présence de plusieurs PG dans la même phrase. Ainsi, nous serions en mesure de ne garder que les termes concernant un seul des PG, et donc de réduire les possibles risques de confusion. Néanmoins, cela implique d'avoir à disposition un analyseur syntaxique pour la langue française suffisamment fiable. Il pourrait également

être pertinent d'utiliser la résolution d'anaphores pour traiter les cas où le PG n'est pas mentionné explicitement dans la synthèse historique. Par exemple, pour l'instance de relation de *propriété* suivante entre « Mary-Lauretta Stuart » et la « Maison Henry-Stuart » [1], nous avons la phrase : « En 1918, l'avocat Gustavus George Stuart (1855-1918) acquiert la propriété pour la donner à ses nièces Mary-Lauretta (1884-1974) et Adèle (1889-1987) Stuart. » Comme on peut le constater, la relation de *propriété* est assez clairement exprimée. Néanmoins, notre approche n'est pas en mesure de trouver que « Mary-Lauretta » correspond à « Mary-Lauretta Stuart », et donc ne peut pas traiter cette instance. La résolution d'anaphores permettrait donc de corriger ce problème.

Le parcours des données ainsi que l'analyse d'erreurs ont révélé la présence de motifs récurrents pour décrire les relations, comme « X a construit le bâtiment selon les plans de l'architecte Y ». On pourrait donc envisager un système de règles permettant de traiter ces cas, mais aussi les relations moins fréquentes et décrites par un vocabulaire très précis, comme les relations d'*inauguration*.

Une partie des améliorations futures dépendra également de la restructuration des données du RPCQ, et nécessitera également de corriger suffisamment de données pour utiliser l'apprentissage supervisé. Avec cette restructuration, il faudra également réfléchir à une façon de gérer les cas où nous aurons des couples $\langle \text{PG}, \textit{Bien Immobilier} \rangle$ associés à plusieurs relations. Une approche sans doute possible serait de séparer les propositions plutôt que les phrases, ce qui permettrait notamment de résoudre le cas assez commun « X a construit Z, selon les plans de Y ».

Par ailleurs, il est envisageable de nous appuyer sur l'ontologie en développement au MCC pour poursuivre les expérimentations de classification taxonomique. Il serait également intéressant d'essayer des approches spécialement conçues pour construire des classifieurs hiérarchiques, comme HSVM (Chen et al. (2004)). Le principe de HSVM est de construire un arbre de décisions possédant à chaque nœud un classifieur binaire SVM qui a donc pour tâche de séparer les données en deux pour continuer la descente de l'arbre. La création de la hiérarchie repose alors sur la résolution de problèmes de coupe maximum afin de séparer l'ensemble des classes en deux, et ce, de façon récursive jusqu'à n'avoir plus qu'une classe à chaque feuille. On obtient alors un arbre binaire, et l'on s'affranchit du besoin d'une taxonomie construite au préalable. Toujours dans la construction automatique de taxonomie, il serait également intéressant d'expérimenter les approches de Du et al. (2018) et de Marszałek and Schmid (2008), qui proposent de reporter à plus tard les classes difficiles à discriminer. Les instances

1. `http://www.patrimoine-culturel.gouv.qc.ca/rpcq/detail.do?methode=consulter&id=92466&type=bien`

appartenant à ces classes sont alors propagées vers chacun des nœuds fils plutôt que vers un seul. La hiérarchie ainsi construite ne sera donc pas un arbre, mais un graphe orienté acyclique.

Il nous reste également à enquêter sur l'impact du modèle préentraîné pour Word2Vec. Ainsi, il serait intéressant d'étudier l'influence d'un changement de corpus d'entraînement, par exemple si l'on entraîne notre modèle uniquement sur des données patrimoniales. Un travail de ce type a déjà été effectué par un autre étudiant du laboratoire, mais les résultats n'ont pas été très concluant. On pourrait aussi changer les paramètres du modèle et pas uniquement l'architecture utilisée. Par ailleurs, nous pourrions essayer d'autres modèles de plongement lexical, comme GloVe (Pennington et al. (2014)). Néanmoins, nous ne pensons pas que des recherches en ce sens amélioreraient les résultats de façon significative, puisque, comme l'a révélé l'analyse d'erreurs, une grande partie des problèmes de classifications découlent de nos données et de notre approche plutôt que du plongement lexical en lui-même. C'est d'ailleurs sans doute pour cela que nous obtenons des résultats similaires entre les deux architectures Word2Vec utilisées, ainsi qu'entre TF-IDF-SVM et W2V-MLP.

L'utilisation de bases de connaissances existantes est également une approche envisageable. Cependant, notre corpus est relativement spécifique, aussi il est possible que nous ne trouvions pas ou peu de références à nos entités dans d'autres bases de connaissances. Nous avons songé à utiliser DBpedia, mais nous avons constaté que moins de 5% des biens immobiliers et des PG du RPCQ y sont référencés. Il faudrait sans doute creuser dans les bases de données spécialisées. Cela nous permettrait alors de résoudre des cas où le bien immobilier ne présente pas de synthèse historique dans le RPCQ ou lorsque le PG n'y est pas mentionné : nous pourrions alors nous appuyer sur une source externe pour compléter les informations manquantes, et ainsi enrichir le RPCQ.

Enfin, notre approche ne se contente que d'extraire des relations entre PG et biens immobiliers, mais comme évoqué en introduction, le RPCQ compte également des données d'autres types, comme des biens mobiliers ou des savoir-faire. Il reste donc à voir si notre approche utilisée présentement fonctionne également pour ces données. Un autre travail à faire pour répondre pleinement aux besoins du MCC est l'extraction de davantage d'informations sur les relations, notamment l'aspect temporel ; ce qui est une tâche bien plus complexe, bien que l'on puisse sans doute s'appuyer sur le fait que les synthèses historiques sont généralement organisées de façon chronologique.

RÉFÉRENCES

V. Andreas et T. Douglas, “A knowledge-based approach to information extraction for semantic interoperability in the archaeology domain”, *Journal of the Association for Information Science and Technology*, vol. 67, no. 5, pp. 1138–1152, 2015. DOI : 10.1002/asi.23485. En ligne : https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23485

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, et Z. Ives, “Dbpedia : A nucleus for a web of open data”, dans *The semantic web*. Springer, 2007, pp. 722–735.

I. Augenstein, S. Padó, et S. Rudolph, “Lodifier : Generating linked data from unstructured text”, dans *The Semantic Web : Research and Applications*, E. Simperl, P. Cimiano, A. Polleres, O. Corcho, et V. Presutti, éds. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012, pp. 210–224.

B. E. Boser, I. M. Guyon, et V. N. Vapnik, “A training algorithm for optimal margin classifiers”, dans *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

L. Breiman, “Random forests”, *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. DOI : 10.1023/A:1010933404324. En ligne : https://doi.org/10.1023/A:1010933404324

L. Breiman, J. H. Friedman, R. A. Olshen, et C. J. Stone, “Classification and regression trees”, 1984.

W. Buranasing, S. Phoomvuthisarn, et M. Buranarach, “Information extraction and integration for enriching cultural heritage collections”, dans *2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, Nov 2016, pp. 1–6. DOI : 10.1109/KICSS.2016.7951425

K. Byrne et E. Klein, “Automatic extraction of archaeological events from text”, 04 2009.

T. Chen et C. Guestrin, “Xgboost : A scalable tree boosting system”, dans *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

Y. Chen, M. M. Crawford, et J. Ghosh, “Integrating support vector machines in a hierarchical output space decomposition framework”, dans *Geoscience and Remote Sensing*

Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International, vol. 2. IEEE, 2004, pp. 949–952.

M. Choi, H.-g. Lee, et H. Kim, "Relation extraction based on two-step classification with distant supervision", *The Journal of Supercomputing*, vol. 72, no. 7, pp. 2609–2622, 2016.

H. Cunningham, "Gate, a general architecture for text engineering", *Computers and the Humanities*, vol. 36, no. 2, pp. 223–254, 2002.

H. Cunningham, D. Maynard, et V. Tablan, "Jape : a java annotation patterns engine", 1999.

M. Doerr, S. Gradmann, S. Hennicke, A. Isaac, C. Meghini, et H. Van de Sompel, "The europeana data model (edm)", dans *World Library and Information Congress : 76th IFLA general conference and assembly*, 2010, pp. 10–15.

Y. Du, J. Liu, W. Ke, et X. Gong, "Hierarchy construction and text classification based on the relaxation strategy and least information model", *Expert Systems with Applications*, vol. 100, pp. 157–164, 2018.

J. Garten, K. Sagae, V. Ustun, et M. Dehghani, "Combining Distributed Vector Representations for Words", dans *Proceedings of NAACL-HLT 2015*. Denver, Colorado : Association for Computational Linguistics, Juin 2015, pp. 95–101. En ligne : `http://ict.usc.edu/pubs/Combining%20Distributed%20Vector%20Representations%20for%20Words.pdf`

M. Kamkarhaghighi et M. Makrehchi, "Content tree word embedding for document representation", *Expert Systems with Applications*, vol. 90, pp. 241 – 249, 2017. DOI : `https://doi.org/10.1016/j.eswa.2017.08.021`. En ligne : `http://www.sciencedirect.com/science/article/pii/S0957417417305596`

D. P. Kingma et J. Ba, "Adam : A method for stochastic optimization", *arXiv preprint arXiv :1412.6980*, 2014.

Q. Le et T. Mikolov, "Distributed representations of sentences and documents", dans *International Conference on Machine Learning*, 2014, pp. 1188–1196.

J. Li, J. Li, X. Fu, M. A. Masud, et J. Z. Huang, "Learning distributed word representation with multi-contextual mixed embedding", *Knowledge-Based Systems*, vol. 106, pp. 220–230, 2016.

M. Marszałek et C. Schmid, “Constructing category hierarchies for visual recognition”, dans *European conference on computer vision*. Springer, 2008, pp. 479–491.

T. Mikolov, K. Chen, G. Corrado, et J. Dean, “Efficient estimation of word representations in vector space”, *arXiv preprint arXiv :1301.3781*, 2013.

T. Mikolov, W.-t. Yih, et G. Zweig, “Linguistic regularities in continuous space word representations”, dans *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, 2013, pp. 746–751.

G. A. Miller, “Wordnet : a lexical database for english”, *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

D. Milne et I. H. Witten, “Learning to link with wikipedia”, dans *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 509–518.

F. Niccolucci et J. D. Richards, “Ariadne : advanced research infrastructures for archaeological dataset networking in europe”, *International Journal of Humanities and Arts Computing*, vol. 7, no. 1-2, pp. 70–88, 2013.

S. Odat, T. Groza, et J. Hunter, “Extracting structured data from publications in the art conservation domain”, *Digital Scholarship in the Humanities*, vol. 30, no. 2, pp. 225–245, 2014.

J. Pennington, R. Socher, et C. Manning, “Glove : Global vectors for word representation”, dans *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

L. E. Peterson, “K-nearest neighbor”, *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

F. Rosenblatt, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms”, CORNELL AERONAUTICAL LAB INC BUFFALO NY, Rapp. tech., 1961.

G. Salton et C. Buckley, “Term-weighting approaches in automatic text retrieval”, *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

C. Schöch, “A word2vec model file built from the French Wikipedia XML Dump using gensim.” Oct. 2016. DOI : `10.5281/zenodo.162792`. En ligne : `https://doi.org/10.5281/zenodo.162792`

A. Vlachidis et D. Tudhope, “Classical art semantics information extraction : Casie pilot project”, 2013.

ANNEXE A Résultats supplémentaires de l'étude de la taille du contexte

Tableau A.1 Classification binaire : rappel (sur le jeu d'entraînement)

Rappel pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,878**	,719	,770	,788	,829	,797	,809	,831	,823	,825	,821	,821	,829	,815	,827
W2V-SVM	**,874**	,722	,764	,819	,825	,799	,805	,815	,839	,799	,803	,809	,823	,821	,842
W.W2V-SVM	**,867**	,714	,781	,817	,827	,789	,819	,813	,831	,825	,835	,819	,817	,811	,815
TF-IDF-MLP	**,892**	,799	,783	,829	,846	,823	,835	,837	,856	,852	,866	,874	,864	,852	,860
W2V-MLP	**,908**	,756	,827	,825	,840	,846	,839	,843	,843	,833	,842	,844	,836	,850	,838
W.W2V-MLP	**,888**	,754	,815	,827	,827	,835	,827	,833	,841	,842	,829	,840	,831	,825	,827
Rappel pour la relation d'occupation															
TF-IDF-SVM	**,966**	,924	,938	,926	,933	,929	,920	,920	,916	,910	,905	,901	,908	,908	,909
W2V-SVM	**,904**	,826	,846	,849	,860	,857	,864	,868	,864	,874	,874	,863	,865	,862	,870
W.W2V-SVM	**,946**	,837	,865	,882	,897	,892	,891	,888	,891	,890	,880	,880	,868	,873	,871
TF-IDF-MLP	**,933**	,803	,855	,871	,880	,878	,884	,883	,880	,878	,876	,868	,885	,883	,882
W2V-MLP	**,943**	,889	,904	,905	,921	,908	,906	,906	,909	,908	,911	,888	,894	,894	,878
W.W2V-MLP	**,911**	,880	,896	,889	,896	,886	,886	,881	,884	,868	,867	,869	,869	,860	,849
Rappel pour la relation de conception															
TF-IDF-SVM	**,920**	,726	,784	,825	,832	,851	,838	,843	,837	,835	,836	,837	,838	,831	,834
W2V-SVM	**,904**	,803	,819	,831	,832	,838	,830	,815	,819	,817	,809	,795	,791	,797	,801
W.W2V-SVM	**,909**	,794	,812	,815	,830	,816	,815	,804	,811	,810	,803	,801	,795	,805	,804
TF-IDF-MLP	**,887**	,807	,813	,829	,812	,814	,806	,796	,801	,811	,799	,802	,797	,797	,783
W2V-MLP	**,920**	,816	,830	,854	,860	,860	,860	,860	,861	,847	,856	,845	,834	,839	,847
W.W2V-MLP	**,907**	,807	,837	,844	,860	,847	,850	,838	,833	,837	,820	,823	,820	,823	,817
Rappel pour la relation de décoration															
TF-IDF-SVM	,951	,874	,885	,909	,940	,952	,958	**,964**	,946	,940	,951	,958	,952	,940	,940
W2V-SVM	,944	,760	,897	,909	,921	,957	,939	,958	,952	,952	,970	,964	**,976**	,958	,946
W.W2V-SVM	,945	,772	,897	,921	,903	,957	,951	,940	,933	,951	,945	,951	,951	,951	**,952**
TF-IDF-MLP	,951	,861	,836	,890	,927	,952	,946	,970	,946	,958	**,982**	,970	,970	,970	,952
W2V-MLP	,944	,819	,904	,927	,927	,958	,939	,946	,946	,964	**,970**	**,970**	**,970**	,952	,946
W.W2V-MLP	**,957**	,808	,903	,945	,932	,951	,933	,928	,933	,951	,945	,946	,945	,945	,951

Tableau A.2 Classification binaire : précision (sur le jeu d'entraînement)

Précision pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,942**	,836	,875	,899	,887	,886	,898	,894	,899	,902	,902	,896	,893	,880	,893
W2V-SVM	**,890**	,716	,779	,802	,811	,803	,813	,831	,822	,822	,797	,814	,820	,809	,829
W.W2V-SVM	**,901**	,717	,796	,810	,824	,813	,822	,833	,859	,841	,845	,829	,828	,841	,840
TF-IDF-MLP	**,925**	,742	,830	,839	,852	,848	,856	,842	,859	,870	,872	,871	,858	,874	,886
W2V-MLP	**,917**	,783	,841	,852	,851	,840	,854	,869	,869	,866	,865	,859	,851	,866	,865
W.W2V-MLP	**,901**	,781	,831	,846	,843	,843	,828	,842	,847	,841	,839	,839	,836	,828	,831
Précision pour la relation d'occupation															
TF-IDF-SVM	**,928**	,808	,857	,871	,889	,889	,896	,902	,901	,892	,899	,895	,892	,888	,885
W2V-SVM	**,925**	,836	,848	,853	,866	,875	,861	,883	,874	,869	,870	,867	,863	,862	,867
W.W2V-SVM	**,927**	,839	,860	,872	,887	,875	,881	,875	,876	,874	,876	,865	,859	,864	,861
TF-IDF-MLP	**,929**	,859	,865	,878	,870	,874	,882	,890	,878	,883	,880	,866	,869	,863	,871
W2V-MLP	**,946**	,863	,883	,903	,904	,899	,902	,904	,899	,897	,899	,887	,895	,886	,891
W.W2V-MLP	**,923**	,859	,887	,888	,892	,882	,885	,882	,872	,876	,868	,865	,871	,868	,864
Précision pour la relation de conception															
TF-IDF-SVM	**,949**	,874	,897	,896	,895	,886	,882	,871	,867	,856	,851	,849	,848	,843	,840
W2V-SVM	**,898**	,809	,829	,839	,845	,834	,828	,817	,827	,816	,802	,815	,806	,801	,803
W.W2V-SVM	**,914**	,810	,838	,843	,851	,838	,836	,829	,820	,818	,813	,805	,811	,801	,802
TF-IDF-MLP	**,891**	,763	,790	,805	,812	,804	,801	,804	,800	,804	,794	,797	,798	,796	,793
W2V-MLP	**,938**	,819	,843	,854	,867	,857	,858	,855	,847	,837	,836	,844	,839	,839	,834
W.W2V-MLP	**,913**	,831	,846	,841	,849	,841	,834	,833	,832	,816	,815	,808	,808	,811	,808
Précision pour la relation de décoration															
TF-IDF-SVM	,966	,745	,853	,925	,949	,971	,994	,988	**1,00**	,989	,989	,989	,979	,994	,989
W2V-SVM	,964	,787	,938	,930	,957	,972	,950	,954	,942	,953	,983	,977	**,978**	**,978**	,973
W.W2V-SVM	,953	,818	,949	,962	,944	,967	,964	,964	,970	,976	,972	**,983**	,982	,977	,972
TF-IDF-MLP	,942	,745	,848	,930	,953	,964	,971	,982	,982	,983	,972	**,989**	,978	,978	,988
W2V-MLP	,972	,826	,935	,920	,930	,953	,970	,953	,970	,977	,983	,978	**,989**	,978	,983
W.W2V-MLP	,966	,849	,945	,947	,952	,959	,969	,957	,976	,982	,983	**,989**	,982	,978	,977

Tableau A.3 Classification binaire : Accuracy (sur le jeu d'entraînement)

Accuracy pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,912**	,787	,828	,849	,859	,845	,858	,865	,864	,867	,865	,862	,864	,851	,863
W2V-SVM	**,881**	,717	,773	,808	,814	,800	,809	,825	,828	,813	,798	,811	,820	,812	,834
W.W2V-SVM	**,885**	,715	,788	,811	,824	,803	,820	,825	,846	,834	,841	,825	,823	,828	,829
TF-IDF-MLP	**,910**	,759	,810	,836	,848	,837	,846	,839	,856	,860	,869	,872	,860	,863	,874
W2V-MLP	**,911**	,772	,835	,841	,843	,839	,845	,858	,856	,851	,853	,852	,847	,859	,853
W.W2V-MLP	**,896**	,771	,827	,839	,835	,836	,825	,838	,845	,840	,835	,840	,832	,826	,827
Accuracy pour la relation d'occupation															
TF-IDF-SVM	**,945**	,852	,890	,894	,907	,906	,906	,909	,907	,899	,901	,897	,899	,896	,894
W2V-SVM	**,915**	,832	,847	,851	,863	,867	,861	,876	,868	,870	,871	,864	,863	,861	,868
W.W2V-SVM	**,935**	,837	,861	,875	,891	,882	,885	,880	,882	,881	,878	,871	,862	,868	,865
TF-IDF-MLP	**,930**	,836	,860	,874	,875	,875	,881	,886	,879	,881	,878	,867	,875	,871	,876
W2V-MLP	**,944**	,873	,892	,903	,912	,902	,905	,904	,902	,901	,903	,886	,893	,889	,885
W.W2V-MLP	**,917**	,868	,891	,888	,894	,883	,884	,881	,877	,872	,867	,866	,869	,864	,857
Accuracy pour la relation de conception															
TF-IDF-SVM	**,936**	,810	,847	,864	,867	,870	,863	,858	,854	,847	,844	,843	,843	,837	,837
W2V-SVM	**,900**	,806	,825	,835	,839	,834	,828	,815	,823	,816	,804	,807	,800	,799	,802
W.W2V-SVM	**,911**	,803	,826	,831	,841	,828	,827	,818	,816	,814	,808	,803	,804	,802	,802
TF-IDF-MLP	**,889**	,778	,798	,814	,812	,807	,802	,801	,799	,806	,795	,798	,796	,795	,788
W2V-MLP	**,927**	,817	,837	,853	,862	,857	,858	,857	,852	,840	,843	,843	,837	,838	,836
W.W2V-MLP	**,910**	,820	,842	,841	,853	,842	,839	,833	,831	,823	,814	,813	,812	,816	,811
Accuracy pour la relation de décoration															
TF-IDF-SVM	,957	,780	,864	,916	,943	,961	**,976**	**,976**	,973	,964	,970	,973	,964	,967	,964
W2V-SVM	,953	,777	,918	,918	,939	,964	,943	,955	,946	,952	**,976**	,970	**,976**	,967	,958
W.W2V-SVM	,948	,798	,924	,942	,924	,961	,958	,952	,952	,964	,958	**,967**	**,967**	,964	,961
TF-IDF-MLP	,945	,777	,840	,913	,940	,958	,958	**,976**	,964	,970	**,976**	**,976**	,973	**,976**	,970
W2V-MLP	,957	,813	,921	,918	,927	,955	,952	,949	,958	,970	,976	,973	**,979**	,964	,964
W.W2V-MLP	,960	,828	,924	,945	,942	,955	,951	,943	,955	**,967**	,964	**,967**	,964	,961	,964

Tableau A.4 Classification multiclasse : Rappel (sur le jeu d'entraînement)

Rappel pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,671**	,522	,608	,626	,608	,590	,619	,633	,571	,582	,588	,606	,600	,600	,562
W2V-SVM	**,689**	,503	,533	,564	,582	,534	,564	,539	,589	,593	,610	,566	,606	,577	,544
W.W2V-SVM	**,699**	,472	,507	,551	,542	,581	,607	,585	,558	,551	,553	,546	,553	,564	,553
TF-IDF-MLP	**,634**	,492	,637	,637	,638	,612	,600	,601	,546	,581	,582	,576	,619	,593	,582
W2V-MLP	**,724**	,562	,599	,662	,594	,607	,636	,611	,600	,630	,598	,625	,624	,600	,605
W.W2V-MLP	**,700**	,570	,568	,599	,583	,626	,656	,610	,589	,593	,594	,595	,553	,601	,601
Rappel pour la relation d'occupation															
TF-IDF-SVM	**,920**	,616	,769	,855	,861	,842	,855	,847	,825	,849	,838	,819	,849	,811	,842
W2V-SVM	**,798**	,625	,751	,740	,739	,720	,684	,727	,744	,713	,659	,650	,680	,680	,697
W.W2V-SVM	**,837**	,600	,690	,771	,734	,764	,800	,782	,746	,758	,770	,776	,789	,769	,775
TF-IDF-MLP	**,786**	,489	,593	,667	,757	,709	,729	,720	,709	,750	,746	,728	,732	,746	,745
W2V-MLP	,804	,775	,825	,800	,781	,768	,805	**,806**	,757	,782	,738	,736	,727	,740	,769
W.W2V-MLP	**,775**	,684	,769	,801	,776	,788	,793	**,775**	,753	,717	,721	,716	,704	,678	,728
Rappel pour la relation de conception															
TF-IDF-SVM	**,853**	,582	,587	,672	,698	,707	,715	,709	,705	,689	,721	,671	,680	,709	,693
W2V-SVM	**,838**	,685	,691	,746	,685	,672	,666	,650	,623	,690	,625	,617	,632	,631	,601
W.W2V-SVM	**,793**	,636	,696	,679	,679	,642	,619	,630	,599	,593	,615	,638	,594	,581	,607
TF-IDF-MLP	**,767**	,570	,575	,656	,686	,649	,655	,618	,614	,618	,660	,624	,613	,624	,619
W2V-MLP	**,843**	,697	,715	,740	,693	,649	,678	,662	,640	,630	,629	,604	,626	,594	,613
W.W2V-MLP	**,805**	,643	,697	,655	,684	,610	,606	,623	,637	,557	,565	,564	,564	,503	,571
Rappel pour la relation de décoration															
TF-IDF-SVM	**,938**	,715	,721	,781	,813	,805	,880	,837	,866	,842	,837	,866	,848	,848	,829
W2V-SVM	**,872**	,522	,668	,740	,757	,782	,812	,781	,829	,792	,807	,825	,818	,810	,807
W.W2V-SVM	**,896**	,505	,703	,752	,789	,805	,794	,860	,866	,862	,861	,831	,854	,854	,836
TF-IDF-MLP	**,919**	,691	,708	,726	,789	,811	,855	,801	,841	,811	,818	,824	,812	,800	,811
W2V-MLP	**,896**	,594	,723	,807	,806	,847	,874	,848	,829	,822	,825	,844	,830	,822	,813
W.W2V-MLP	**,892**	,607	,673	,751	,789	,818	,843	,849	,817	,825	,843	,855	,824	,831	,806

Tableau A.5 Classification multiclasse : précision (sur le jeu d'entraînement)

Précision pour la relation de construction															
		Taille de fenêtre utilisée													
Méthode	Phrase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TF-IDF-SVM	**,830**	,621	,646	,713	,714	,674	,753	,731	,691	,689	,730	,695	,694	,691	,684
W2V-SVM	**,732**	,538	,571	,658	,598	,608	,596	,603	,636	,649	,609	,597	,628	,599	,590
W.W2V-SVM	**,751**	,514	,570	,631	,580	,617	,632	,608	,593	,624	,639	,609	,643	,640	,644
TF-IDF-MLP	**,755**	,567	,635	,612	,696	,664	,657	,616	,610	,637	,679	,612	,627	,624	,641
W2V-MLP	**,756**	,635	,621	,708	,643	,608	,673	,628	,642	,664	,646	,653	,630	,610	,641
W.W2V-MLP	**,704**	,591	,596	,643	,600	,611	,663	,624	,628	,604	,584	,573	,565	,568	,606
Précision pour la relation d'occupation															
TF-IDF-SVM	**,780**	,670	,660	,701	,743	,734	,755	,759	,753	,776	,741	,707	,738	,740	,728
W2V-SVM	**,812**	,690	,755	,717	,789	,753	,741	,711	,734	,771	,713	,668	,722	,695	,692
W.W2V-SVM	**,807**	,663	,728	,723	,747	,744	,751	,728	,712	,702	,717	,712	,703	,691	,708
TF-IDF-MLP	**,782**	,668	,703	,712	,783	,729	,741	,734	,715	,716	,694	,689	,697	,685	,666
W2V-MLP	**,811**	,678	,749	,742	,786	,754	,793	,764	,721	,759	,709	,668	,709	,698	,710
W.W2V-MLP	**,807**	,658	,688	,713	,763	,741	,748	,727	,703	,678	,702	,708	,666	,675	,715
Précision pour la relation de conception															
TF-IDF-SVM	**,881**	,777	,768	,793	,764	,730	,743	,741	,700	,659	,707	,722	,731	,725	,714
W2V-SVM	**,790**	,547	,609	,653	,612	,559	,576	,581	,579	,610	,599	,585	,604	,632	,567
W.W2V-SVM	**,788**	,508	,597	,614	,625	,609	,636	,682	,604	,617	,628	,625	,614	,597	,613
TF-IDF-MLP	**,729**	,578	,575	,631	,644	,632	,647	,629	,612	,649	,673	,641	,636	,663	,687
W2V-MLP	**,837**	,666	,760	,763	,660	,681	,699	,695	,660	,667	,634	,640	,663	,658	,643
W.W2V-MLP	**,780**	,623	,722	,665	,682	,663	,670	,666	,634	,592	,615	,601	,571	,551	,585
Précision pour la relation de décoration															
TF-IDF-SVM	**,918**	,508	,672	,773	,825	,827	,850	,844	,850	,846	,845	,858	,840	,828	,818
W2V-SVM	**,895**	,604	,732	,801	,820	,826	,871	,860	,859	,820	,829	,847	,832	,822	,836
W.W2V-SVM	**,907**	,576	,749	,809	,834	,873	,854	,874	,863	,824	,837	,861	,832	,861	,836
TF-IDF-MLP	**,866**	,502	,644	,767	,801	,811	,821	,803	,794	,799	,799	,807	,838	,819	,786
W2V-MLP	**,889**	,672	,763	,814	,810	,854	,864	,858	,826	,815	,819	,879	,842	,828	,838
W.W2V-MLP	**,897**	,677	,741	,829	,816	,856	,868	,872	,846	,821	,841	,874	,871	,845	,834