Un système intelligent pour l'optimisation du processus de e-recrutement

Halima Ramdani

▶ **To cite this version:**

Halima Ramdani. Un système intelligent pour l'optimisation du processus de e-recrutement. Intelligence artificielle [cs.AI]. Université de Lorraine, 2021. Français. NNT : 2021LORR0366 . tel-04213881

HAL Id: tel-04213881
https://hal.univ-lorraine.fr/tel-04213881

Submitted on 21 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale SIMPPÉ

Un système intelligent pour l'optimisation du processus de e-recrutement

THÈSE

présentée et soutenue publiquement le 14 Décembre 2021

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention Génie des systèmes industriels)

par

Halima Ramdani

Composition du jury

Président :	Yannick Toussaint, Professeur à l'Université de Lorraine
Rapporteurs :	Richard Chbeir, Professeur à l'Université de Pau Myriam Lamolle, Professeure à l'Université Paris 8
Examinateur :	Nicolas Lachiche, Maître de conférences à l'Université de Strasbourg
Encadrants :	Davy Monticolo, Professeur à l'Université de Lorraine Armelle Brun, Maître de conférences (HDR) à l'Université de Lorraine
Invités :	Eric Bonjour, Professeur à l'Université de Lorraine Xavier Ragage, Directeur de Xtramile

Equipe de Recherche sur les Processus Innovatifs

Résumé

Les systèmes d'aide à la décision sont largement utilisés pour résoudre les problèmes de sélection et de prise de décision dans de nombreux domaines. Ces systèmes aident les décideurs à prendre une décision lorsque cette sélection nécessite une expertise ou des connaissances. À mesure que le numérique et les systèmes informatiques évoluent, les environnements de décision sont moins connus par les décideurs entraînant (1) des décisions prises dans l'incertain et influencées par des facteurs externes, (2) des contextes de décision de nature hétérogène. Partant de ces faits, cette thèse propose un système d'aide à la décision générique qui peut être appliqué aux problèmes d'aide à la décision dont (1) l'environnement est incertain et évolutif dans le temps (2) les objectifs du décideur sont multiples (3) le contexte de décision est rédigé en langage naturel, chacun d'eux constituant un défi. Le système ADE^2 que nous proposons se base sur différents modules. Le premier est un module d'extraction et d'identification des informations présentes dans le contexte rédigé en langage naturel afin de le caractériser. Ce module fait l'objet d'une première contribution : DEEP, une méthodologie pour l'extraction d'entités en se basant sur le schéma organisationnel de textes rédigés en langage naturel. Le second module d'ADE^2 a pour objectif de créer des groupes de textes sémantiquement proches afin de pallier le manque de données pour certains contextes sous-représentés. Il fait l'objet d'une deuxième contribution : une approche d'appariement par type d'informations entre deux textes rédigés en langage naturel. Les résultats de cette contribution sont utilisés pour agréger les données temporelles liées aux contextes de décision sémantiquement proches afin de faire une prévision des facteurs de décision. Étant donné l'évolution de l'environnement et son incertitude, une architecture hybride de réseaux de neurones convolutifs et récurrents a été choisie pour capturer les tendances et les corrélations entre les items. Enfin, ces facteurs de décision sont utilisés dans une optimisation multi-objectifs et multi-périodes pour finalement recommander au décideur un ensemble optimal de décision pour ses objectifs et ses contraintes. Nous avons expérimenté ADE^2 dans le domaine du e-recrutement afin d'aider le recruteur (décideur) à choisir (décision) les médias (items évoluant sur le Web) optimaux (optimisation multi-objectifs) pour son offre d'emploi (contexte de décision). Pour ce faire, nous avons comparé les résultats obtenus suite à la mise en place d'une campagne de recrutement par un manager des campagnes et les résultats suite à la recommandation des canaux par ADE^2. Nos expérimentations ont montré que ADE^2 permet un gain de temps au recruteur sur (1) la préparation des données pour la diffusion des offres d'emploi sur les canaux en utilisant la contribution DEEP, (2) l'analyse des données anciennes (3) l'analyse des données actuelles (4) la prise de décision en utilisant les recommandations. ADE^2 permet aussi un gain d'argent, puisque la prévision temporelle et le système de renforcement qui repose sur une correction permanente des données économise de l'argent sur les périodes où les objectifs du recruteur ne peuvent pas être atteints.

Mots-clés: E-recrutement, Extraction d'entités, Appariement, Système d'aide à la décision

Abstract

Decision support systems are commonly used to solve selection and decision-making problems in a variety of domains. As digital and computer systems evolve, decision-making environments become less familiar to decision-makers, resulting in (1) decisions made under uncertainty and influenced by external factors, and (2) hybrid decision-making contexts. This thesis proposes a generic decision support system that can be used to solve problems with the following conditions : (1) the environment is uncertain and changes over time ; (2) the decision-objectives makers are multiple ; and (3) the decision-making context is written in natural language. The system ADE^2 we propose consists of several components. The first component extracts and identifies information from a natural language-written context in order to classify it. For this purpose, our first contribution is used : DEEP, a methodology for entity extraction based on the organizational patterns of a text written in natural language. The second component aims to create semantically comparable groups of texts in order to fill in data gaps for under-represented contexts. This component is our second contribution : a matching method based on the type of information contained in two natural-language texts. The results of this contribution are used to aggregate temporal data related to decision contexts that are semantically close in order to forecast decision-maker choice factors. Given dynamicity and uncertainty in the environment, a hybrid architecture of convolutional and recurrent neural networks was chosen to capture trends and correlations between items. Finally, these decision factors are used in a multi-objective, multi-period optimization to provide the decision maker with the best set of options based on his or her goals and constraints. ADE^2 is used in the e-recruitment domain to assist the recruiter (decision-maker) in selecting (decision) the most appropriate (multi-objective optimization) channels (items) for a job offer (context of decision). To do so, we compared the results obtained after a campaign manager implemented a recruitment campaign with the results obtained after ADE^2 recommended channels. The decision support system saves the recruiter time on (1) data preparation for job posting using the DEEP contribution, (2) data analysis of historical data, (3) data analysis of current data, and (4) decision-making using ADE^2 recommendations, according to our experiments. The time forecasting and reinforcement system, which is based on continuous data rectification, saves money during periods when the recruiter's goals are not met, so this approach saves money as well.

Keywords: E-recruitment, Entity extraction, Matching, Decision support system

Remerciements

J’aimerais montrer ma plus sincère reconnaissance à mes directeurs de thèse, les professeurs Davy Monticolo, Armelle Brun et Eric Bonjour pour leurs disponibilités, leurs rigueurs scientifiques et leurs conseils tout au long de ces trois années de thèse. Chacun de vous a su m’orienter et m’aider pour mener à bien cette thèse. Vous avez été complémentaire et je vous remercie de m’avoir guidé et d’avoir fait en sorte que je sois satisfaite du travail que j’ai réalisé grâce à vous.

Cette thèse n’aurait pas pu avoir lieu sans les deux fondateurs de l’entreprise Xtramile, Xavier Ragage et Stephanie Nenta, qui m’ont fait confiance pour mener à bien ce projet. Je souhaite souligner ma gratitude envers eux pour m’avoir permis d’apprendre beaucoup de choses à leurs côtés et au sein de leur entreprise.

J’adresse tous mes remerciements au Professeur Richard Chbeir, ainsi qu’au Professeure Myriam Lamolle, de l’honneur qu’ils m’ont fait en acceptant d’être rapporteurs de cette thèse. Je remercie également les professeurs Nicolas Lachiche et Yannick Toussaint pour participer au jury de thèse en tant qu’examinateurs.

Un grand merci également aux membres du comité annuel de thèse Professeur Yannick Toussaint et au Docteure Cecile Favre, dont les critiques et conseils m’ont permis d’orienter et de faire progresser ce travail de thèse de manière très efficace.

Je remercie les professeurs et chercheurs de l’ERPI et de l’ENSGSI, qui m’ont toujours soutenu et m’ont conseillé à chacune de mes présentations. Je tiens à remercier également tout le personnel de l’ERPI pour m’avoir facilité l’accès à différentes ressources me permettant ainsi d’avancer dans de très bonnes conditions.

Je tiens également à remercier tous mes collègues doctorants et post-doctorants pour tous les moments partagés ensemble, toutes nos discussions, activités, sorties et j’en passe. Merci pour leur motivation tout au long de ces trois années. Je tiens aussi à remercier tous mes collègues Xtramile, notamment l’équipe R&D qui m’a soutenu et encouragé et auprès de laquelle j’ai énormément appris.

Un grand merci à ma famille, qui m’a soutenue depuis le début et qui a toujours fait en sorte de me rendre les choses plus faciles. Je remercie mes parents et pour leur soutien infaillible. Merci de m’avoir sans cesse rassuré, aider à relativiser et conseiller. Vous avez été le pilier de toutes mes années d’études. Merci à ma sœur de m’avoir fait rire quand je doutais, de m’avoir rassuré et d’avoir fait en sorte que ces trois années se passent toujours dans la joie et la bonne humeur grâce à nos appels quasi quotidiens et malgré la distance. Merci à mon beau-frère d’avoir aussi partagé ces moments avec moi. Merci à ma tante et mon oncle pour leur soutien et leurs précieux conseils scientifiques. Je remercie aussi mes cousins Meryem et Haytham et mon grand-père pour leurs appels et leurs soutiens tout au long des trois années.

Enfin, j’exprime toute ma gratitude à mes amis pour nos nombrables sorties et voyages qui m’ont permis de penser à autre chose et de profiter de ces moments avec eux. Merci aussi pour votre soutien. Je partage aussi une pensée à mes entraîneurs de sport qui ont supporté mon énergie et mon stress pendant cette dernière année de thèse.

Table des matières

Liste des figures

Liste des tableaux

Bilan des formations et des conférences

1 Accompagnement de la thèse

1.1 Ethique et intégrité scientifique

AT3. MDD 11 - Culture de l'intégrité scientifique.

1.2 Formations disciplinaires

AT1. SLTC 07 - Rédiger la thèse ;
AJC CREM - Initiation a l'analyse de texte assistée par ordinateur et aux logiciels d'analyse de données textuelles : Cordial et Hyperbase.
AT1. SIMPPÉ 18 - Ecole d'été : Industry 4.0.

1.3 Formations transverses

AT2. MDD 31 - MOOC Gestion de projet (Certificat sur la gestion de projet) ;
AT2. MDD 42 - Prise de parole en public.

1.4 Langues

AT4. SLTC 05 - Surviving your First Presentation in English.

2 Valorisation scientifique

2.1 Conférences

2.1.1 Participations passives

AFIA 2017 – Plate-forme d'intelligence artificielle qui réunit des chercheurs, industriels et étudiants autour de conférences et d'ateliers consacrés à l'IA – Université de Caen Normandie ;
Journée scientifique de l'école doctorale SIMPPÉ 2019 : 'L'Économie Circulaire' ;
Journée scientifique de l'école doctorale SIMPPÉ 2020 : 'Les actions de l'ED SIMPPÉ dans la lutte contre le réchauffement climatique'.

2.1.2 Participations actives

APIA 2020 [1] : “Définition d’une méthodologie d’indexation de documents textuels par étiquetage de séquences : application aux offres d’emploi” [2].

Infosid 2020 [3] - Forum Jeunes Chercheurs : “Un système intelligent pour l’optimisation du e-recrutement” [4].

ICIKS 2021 [5] : "Decision support system for online recruitment".

2.2 Publications

Digital Technologies, Artificial Intelligence and Decision Making. Lecture Notes in Business Information Processing, vol 425 : "Decision support system for online recruitment" [6]

En cours d’évaluation dans le journal "Knowledge-based systems" : "DEEP : a Methodology for Entity Extraction using Organizational Patterns : Application to Job Offers"

2.3 Cours

Experfy - The Harvard Innovation Lab : Recurrent and Recursive Networks [7]- 2020.

1. http://pfia2020.fr/conferences/apia/programme-apia-2020/
2. https://hal.archives-ouvertes.fr/hal-02974679/
3. https://inforsid2021.sciencesconf.org/resource/page/id/16
4. https://hal.archives-ouvertes.fr/hal-02954812
5. https://iciks.org
6. https://doi.org/10.1007/978-3-030-85977-0_4
7. https://training.experfy.com/courses/recurrent-and-recursive-networks

Liste des notations

Notations	**Description**
$Q = \{q_j : j \in \{1, ..., U\}\}$	Ensemble des items
$\widetilde{D} = \{D_i : i \in \{1, ..., N\}\}$	Ensemble des contextes de décision textuels / Documents textuels
$Z = \{z_{D_i,q_j} : i \in \{1, ..., N\}, j \in \{1, ..., U\}\}$	Ensemble des variables de décision pour le contexte D_i et l'item q_j
F = $\{f_o : o \in \{1, ..., H\}\}$	Ensemble des objectifs du décideur
$c_{D_i,q_j} : i \in 1, ..., N, j \in \{1, ..., U\}$	Constante de contraintes liées à l'item q_j pour le contexte i
$B_{D_i} : i \in \{1, ..., N\}$	Constante de contraintes liées au contexte de décision D_i
$h_{D_i,q_j} : i \in \{1, ..., N\}, j \in \{1, ..., U\}$	Paramètre du contexte textuel D_i et de l'item q_j
$L = \{l_w : w \in \{1, ..., E\}\}$	Ensemble des étiquettes
$I_{q_j,l_m} : q_j \in Q, l_m \in L$	Matrice des caractéristiques de chaque item q_j pour chaque étiquette l_m
$ch_{q_j,l_m} : q_j \in Q, l_m \in L$	La séquence normalisée associée à la caractéristique de l'item q_j pour l'étiquette l_m
$K = \{q_j : j \in \{1, ..., M\}\}$	Ensemble des items résultants du module de filtrage c'est-à-dire les caractéristiques des items sont similaires au contexte de décision du décideur

Notations	**Description**
$X = \{x_{D_i,q_j} : i \in \{1,...,N\}, j :\in K\}$ avec $K = \{q_j : j \in \{1,...,M\}\}$	Ensemble des variables de décision utilisés dans le module d'optimisation
I_{q_j,l_m}	Matrice des caractéristiques de chaque item q_j pour chaque étiquette l_m
A_{D_i,l_m}	La matrice des séquences normalisées du contexte D_i pour chaque étiquette l_m
$r_{D_i,l_w} \in \mathbb{N}; D_i \in \widetilde{D}, l_w \in L$	Le nombre de séquences associées à l'étiquette l_w pour le contexte de décision textuel D_i
$\varphi_{D_i,l_w} = \{S_{D_i,l_w}, D_i \in \widetilde{D}, l_w \in L\}$, $card(\varphi_{D_i,l_w}) = r_{l_w,D_i}$	L'ensemble des séquences associées à l'étiquette l_w pour le contexte textuel D_i
$N\varphi_{D_i,l_w} : D_i \in \widetilde{D}, l_w \in L$	L'ensemble des séquences normalisées associées à l'étiquette l_w pour le contexte textuel D_i
$n_{D_i} = \{\{l_w, \varphi_{D_i,l_w}\}, l_w \in L, D_i \in \widetilde{D}\}$	Le contexte normalisé composé des paires étiquettes et ensemble des séquences associées à chaque étiquette pour le contexte textuel D_i
$S = \{CLS_d : d \in \{1,...,C\}\}$	Ensemble des classes sources
$\hat{T}$	Période courante (période de 7 jours)
$T' = \{\hat{T} : \hat{T} \in O,...,G\}$	Ensemble des périodes futures
$T'' = \{\hat{T} : \hat{T} \in \{-r,...,-1\}\}$	Ensemble des périodes passées
$\gamma_{CLS_d,q_j\,t} = \{\hat{\theta}_{CLS_d,q_j}(t) : CLS_d \in S, q_j \in Q, t \in T'\}$	La série chronologique du paramètre h_{D_i,q_j} pour la classe source CLS_d et l'item q_j pour les périodes futures appartenant à l'ensemble T'
D_v	Le nouveau contexte de décision inconnu par le système d'aide à la décision pour lequel le décideur souhaite se voir attribuer des recommandations

Notations	**Description**
n_v	Le contexte normalisé de D_v
CLS_{n_v}	La classe source la plus similaire au contexte de décision D_v
$\gamma_{CLS_{n_v},q_j\,t} : j \in K$	La série chronologique attribuée à la classe cible pour la suite du module pour l'item $q_j \in K$
x_{D_v,q_j} avec $q_j \in K$ et $1 \leq \sum_{j=1}^{M} x_{D_V,q_j}$	Variable de décision pour le problème mono-période.
B_{D_v}	Constante de la contrainte liée au contexte D_v.
c_{D_v,q_j}	Constante de la contrainte liée au coût de l'item q_j pour le contexte D_v.
$\begin{cases} \max f(x)_{D_v} = \sum_{j=1}^{M} \Theta_{D_v,q_j} * x_{D_v,q_j} \\ s.c \\ \sum_{j=1}^{M} c_{D_v,q_j} * x_{D_v,q_j} \leq B_{D_v} \end{cases}$	Problème d'optimisation mono-période et mono-objectif
$X_{\hat{T}} = \{x_{D_v,q_j,\hat{T}} : j \in K, \hat{T} \in T'\}$	Ensemble des variables de décision à des périodes de temps différentes
$\begin{cases} max f(x) = \sum_{\hat{T}=1}^{g} \sum_{j=1}^{M} \hat{\Theta}_{D_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ s.c \\ \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} c_{D_v,q_j} * x_{D_v,q_j\hat{T}} \leq B_{D_v} \end{cases}$	Problème d'optimisation multi-périodes et mono-objectif
$\tilde{P} = \{p_1, p_2, .., p_H\}$	Ensemble des paramètres liés à chacune des fonctions objectifs
$\hat{p_o}_{d_v,q_j}(t) : j \in K, t \in T'$	La prédiction du paramètre p_o pour le contexte de décision d_v à l'instant $t \in T'$ pour l'item q_j

Notations	Description
$\begin{cases} \max f_1(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p_1}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ \max f_2(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p_2}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ .... \\ \max f_o(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p_o}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ .... \\ \max f_H(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p_H}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ s.c \\ \sum_{\hat{T}=1}^{g} \sum_{j=0}^{M} c_{D_v,q_j} * x_{D_v,q_j\hat{T}} \leq B_{D_v} \end{cases}$	Problème multi-périodes et multi-objectif

Introduction générale

1 Contexte

1.1 Motivations

Le domaine du recrutement est un secteur où se développent des innovations en continu et qui évolue à l'ère de la révolution digitale. À l'origine, le recrutement se faisait par la méthode du bouche-à-oreille, puis a exploité par les petites annonces dans les journaux et aujourd'hui Internet. Cette nouvelle forme de recrutement en ligne, appelée "e-Recrutement", se focalise sur la publication des offres d'emploi sur des canaux de diffusion de typologies différentes : réseaux sociaux, métamoteurs, sites publicitaires.

L'e-recrutement a considérablement fait évoluer le paysage du recrutement tant pour les entreprises que pour les chercheurs d'emploi. En effet, l'e-recrutement permet de rechercher les candidats pour une offre d'emploi donnée dans un spectre de populations beaucoup plus large. Cela implique cependant la gestion d'une grande quantité d'informations. En effet, chaque canal possède une stratégie financière particulière et a pour objectif de cibler des profils de candidats particuliers. La diffusion des offres est également payante sur une grande partie des sites spécialisés. De ce fait, la tâche de recrutement en ligne devient de plus en plus difficile pour le recruteur. En effet, pour réussir son recrutement, il est nécessaire de connaître précisément les caractéristiques des canaux de diffusion et des offres d'emploi afin de choisir les canaux qui ont les meilleures chances de recrutement dans le respect des contraintes financières du recruteur. La tâche est d'autant plus difficile que les caractéristiques des canaux évoluent entraînant ainsi l'évolution de leurs rentabilités. Pour une entreprise, le coût annuel des recrutements peut donc être très élevé. En conséquence, il est devenu indispensable pour les recruteurs d'évaluer et d'analyser les performances des différents supports utilisés afin de les choisir objectivement lors de la diffusion d'une offre d'emploi. La performance d'une campagne de recrutement est généralement mesurée selon les objectifs du recruteur. Par exemple, maximiser la visibilité de son offre pour avoir le plus de candidatures possibles ou maximiser le nombre de candidatures en adéquation avec le profil recherché et minimiser les clics générés[8]. Face à cette multitude de sites et d'objectifs, la tâche de recrutement en ligne nécessite l'analyse de masse de données provenant de sources différentes (dédiés aux sites carrières, sites de recrutement, sites publicitaires, etc.) et nécessite une expertise dans le domaine pour optimiser le recrutement.

En prenant en compte ces éléments, notre objectif est de proposer un système d'aide

8. Sur beaucoup de sites le coût de la publication dépend du nombre de clics, il s'agit d'un forfait par coût par clic

à la décision à destination des recruteurs pour optimiser le recrutement de profil candidat pour une offre d'emploi donnée, en identifiant automatiquement les canaux permettant de répondre aux objectifs fixés par le recruteur. Un tel système permet trouver des candidats pour une offre d'emploi et de recommander les canaux permettant de satisfaire les objectifs du recruteur sur le court et le long terme tout en respectant ses contraintes budgétaires.

1.2 Contexte industriel

Nos travaux de recherche ont été effectués en collaboration avec l'entreprise Xtramile [9] dans le cadre d'une convention CIFRE. Xtramile est une entreprise du domaine du numérique visant à aider les entreprises à optimiser leur recrutement digital. Pour ce faire, Xtramile utilise une approche programmatique groupant des outils algorithmiques, de brassage de données (Big Data) et d'apprentissage automatique. Xtramile applique également des fondamentaux de techniques financières (coût-par-clic, coût par candidat, allocations de budget) au monde professionnel du recrutement.

2 Positionnement par rapport aux travaux de la littérature

La littérature s'est intéressée à la problématique de l'optimisation du e-recrutement en proposant un système de recommandation basé sur le contenu de l'offre d'emploi [130] pour estimer le rendement des canaux. Ce travail utilise un corpus de données contenant les offres d'emploi diffusées dans le passé et les actions de clics, vues, etc. effectués par les utilisateurs des canaux sur ces offres. Ces actions créent des évènements de clics, d'envoi de candidatures, etc. permettant de calculer des indicateurs de performance. Dans ce travail, seul le taux de conversion (le rapport entre le nombre de CV reçus et le nombre de clics) est utilisé comme indicateur et la dimension temporelle de l'information relative au canal de diffusion n'est pas prise en compte, alors qu'il est évident que le moment où une offre est diffusée sur un canal influence énormément l'impact de cette offre. Par exemple, il existe des périodes comme le printemps où les candidats sont plus actifs que d'autres périodes [51]. Les travaux de [136] ont permis de vérifier que la prise en compte de la temporalité permet d'améliorer les performances d'ADE^2. Le système proposé dans ces travaux recommande les canaux en fonction de la prédiction des clics dans le temps (séries temporelles), qui est considérée comme seul indicateur de performance d'un canal. Grâce à cette prédiction, le système recommande aux recruteurs les canaux permettant de maximiser le nombre de clics. Nous avons identifié plusieurs limites dans les travaux de la littérature dédiée à l'optimisation du choix des canaux de recrutement :

1. (L1) L'absence de l'identification du profil recherché à partir de l'offre d'emploi pour la recommandation des canaux, seule l'offre brute est exploitée. Le profil recherché peut être identifié au travers d'une formation requise, d'une expérience, etc. Notre intuition est que l'identification du profil recherché permettrait de mieux mettre en relation le profil recherché et le canal et ainsi améliorer les recommandations.
2. (L2) L'exploitation d'indicateurs de performance reposant uniquement sur le nombre de clics et le taux de conversion de candidats. D'autres indicateurs sont classiquement

9. https ://xtramile.io/home

utilisés par les recruteurs pour définir la performance d'une campagne de recrutement. Par exemple, le coût par candidat pertinent qui représente le coût pour obtenir un CV qui répond au profil recherché dans l'offre ou encore le coût par candidat.

3. (L3) La prise en compte d'un objectif unique pour la recommandation des canaux. En effet, ces travaux ne considèrent pas plusieurs objectifs à atteindre lors d'une campagne de recrutement. Pourtant, le recruteur peut évaluer la performance de sa campagne de recrutement en utilisant plusieurs indicateurs de performance et donc peut avoir plusieurs objectifs différents. Notre intuition se base sur le fait que la prise en compte de plusieurs indicateurs permettrait de mieux recommander et atteindre les objectifs du recruteur.
4. (L4) La caractérisation des canaux et des profils candidats que ces canaux véhiculent est étudiée uniquement sur la base du nombre de clics et CV reçus. Cependant, plusieurs facteurs externes (marché du travail, secteur spécifique, etc.) ou internes (diffusion plus fréquente sur certains canaux, biais humain sur le choix des canaux dans les données historiques) peuvent influencer la réception de clics ou de CV et donc la performance d'un canal. Ces facteurs reflètent un environnement évolutif et incertain. Le caractère incertain de l'environnement n'est aujourd'hui pas pris en compte dans la littérature.

3 Objectifs scientifiques et questions de recherche

L'objectif de cette thèse peut être résumé de la façon suivante :

Proposer un système d'aide à la décision pour répondre aux objectifs multiples du recruteur et à l'environnement évolutif du recrutement.

Cet objectif peut être décomposé en trois sous-objectifs. Le premier consiste à analyser et identifier le profil recherché à partir de l'offre d'emploi. Cet objectif répond à la première limite des travaux de la littérature (L1). Le second objectif consiste à analyser la pertinence d'un CV pour pouvoir introduire dans le système d'aide à la décision l'indicateur de performance basé sur la pertinence des CV. Ce deuxième objectif répond à la limite (L2). Enfin, le dernier objectif étant de considérer objectifs multiples du recruteur et l'environnement incertain du e-recrutement afin d'améliorer la prise de décision du recruteur. Ce dernier objectif répond aux limites (L3) et (L4).

3.1 Extraction d'entités à partir de textes rédigés en langage naturel

Les offres d'emploi et les CV sont rédigés en langage naturel. Le premier objectif de nos travaux est de concevoir des méthodes d'identification automatique des informations présentes dans ces deux documents, qui permettront d'identifier au mieux le profil souhaité et le profil candidat. Nous proposons de voir la tâche d'identification du profil comme un problème d'extraction d'entités, c'est-à-dire associer à chaque mot ou ensemble de mot du texte une catégorie d'informations. Notons que ces documents ont plusieurs caractéristiques. Ils sont rédigés librement en langage naturel, ont un schéma organisationnel et le

vocabulaire utilisé évolue au cours du temps. En effet, une offre et un CV représentent un enchaînement de sections représentant chacune un type d'information. Néanmoins, l'ordre des sections peut varier d'un document à un autre. De plus, avec l'apparition de nouveaux métiers et de nouvelles compétences, le vocabulaire utilisé tend à évoluer. Par ailleurs, certaines informations partagent un vocabulaire commun, c'est le cas des compétences techniques et des expériences, ce qui peut provoquer une ambiguïté lors de l'extraction d'entités. De ce fait, notre premier objectif renvoie à la problématique suivante :

QR1
Comment identifier et extraire de l'information à partir de textes rédigés en langage naturel en utilisant leur schéma organisationnel tout en étant robuste à l'évolution du vocabulaire ?

Cette contribution a fait l'objet d'une publication en conférence. La première publication "Définition d'une méthodologie d'indexation de documents textuels par étiquetage de séquences : application aux offres d'emploi" [119] est dans le cadre de la conférence nationale APIA 2020. Un article en anglais "DEEP : a Methodology for Entity Extraction using Organizational Patterns : Application to Job Offers" a été rédigé et est en cours d'évaluation dans le journal "Knowledge-based systems".

3.2 Analyse de la pertinence des CV

Le second objectif de nos travaux est d'analyser la pertinence d'un CV afin de considérer cette donnée comme un indicateur de performance d'une campagne de recrutement. Étant donné l'importance de cette information pour le recruteur, il est important de proposer un modèle transparent envers le recruteur, c'est-à-dire pouvoir lui proposer un système non opaque. De ce fait, à partir de l'extraction et l'identification des informations de l'offre et du CV, l'objectif est de les apparier et de comparer les deux profils associés pour attribuer un score de pertinence (basé sur un système non opaque) qui peut faire partie des indicateurs de performance utilisés pour le système d'aide à la décision. Un autre objectif serait d'être en mesure d'apparier les offres d'emploi entre elles par type d'informations pour la recommandation des canaux. De ce fait, notre second objectif renvoie à la problématique suivante :

QR2
Comment apparier deux textes rédigés en langage naturel ?

3.3 Système d'aide à la décision hybride s'adaptant à un environnement incertain

L'environnement du Web est incertain (QR3-C1). En effet, les conséquences des décisions d'un recruteur ne sont pas connues avec certitude. L'environnement du Web est aussi évolutif dans le temps (QR3-C2). Ceci entraîne l'évolution dans le temps des paramètres d'un recruteur, qui sont les indicateurs de performance tels que le nombre de candidatures reçu, le nombre de clics sur une offre d'emploi en ligne, etc. Le décideur du système

d'aide à la décision (le recruteur) souhaite optimiser des indicateurs de performance qui évoluent dans cet environnement. Ces caractéristiques rendent la tâche d'optimisation difficile d'autant plus lorsque les objectifs à atteindre sont multiples (QR3-C3) et le contexte d'optimisation du recruteur est décrit dans l'offre d'emploi (QR3-C4). L'évolution des données dans un environnement incertain dans le cas pratique du recrutement, entraîne la nécessité de la mise en place de prédictions.

Les méthodes classiques pour l'aide à la décision ne considèrent pas toutes ces caractéristiques. Le troisième objectif de cette thèse est donc de proposer un système d'aide à la décision permettant d'aider le recruteur à optimiser son recrutement avec les caractéristiques (QR3-C1), (QR3-C2), (QR3-C3) et (QR3-C4). De ce fait, notre dernier objectif correspond à la problématique suivante :

QR3
Comment concevoir un système d'aide à la décision qui s'adapte à un environnement incertain, évolutif et répondant à des objectifs multiples dont les paramètres sont variables dans le temps ?

Cette contribution a fait l'objet de deux publications d'article. La première "Un système intelligent pour l'optimisation du e-recrutement" [6] est dans le cadre du Forum Jeunes Chercheurs Inforsid. La seconde "Decision support system for online recruitment" est dans le cadre d'une conférence internationale ICIKS au travers laquelle l'article a été publié en chapitre du livre Lecture Notes in Business Information Processing, vol 425 [120].

4 Plan du document

Le plan de ce document de thèse est présenté dans la Figure 1. Le premier chapitre sera consacré à un état de l'art sur les travaux de la littérature sur les trois questions de recherche (QR-1), (QR-2) et (QR-3). Les trois chapitres suivants présenteront nos contributions pour chacune de ces questions. Le contexte expérimental et les contributions sont quant à eux présentés dans le chapitre 5. Enfin, nous conclurons ce mémoire en résumant les contributions, les résultats des expérimentations et enfin une discussion autour des perspectives de recherche.

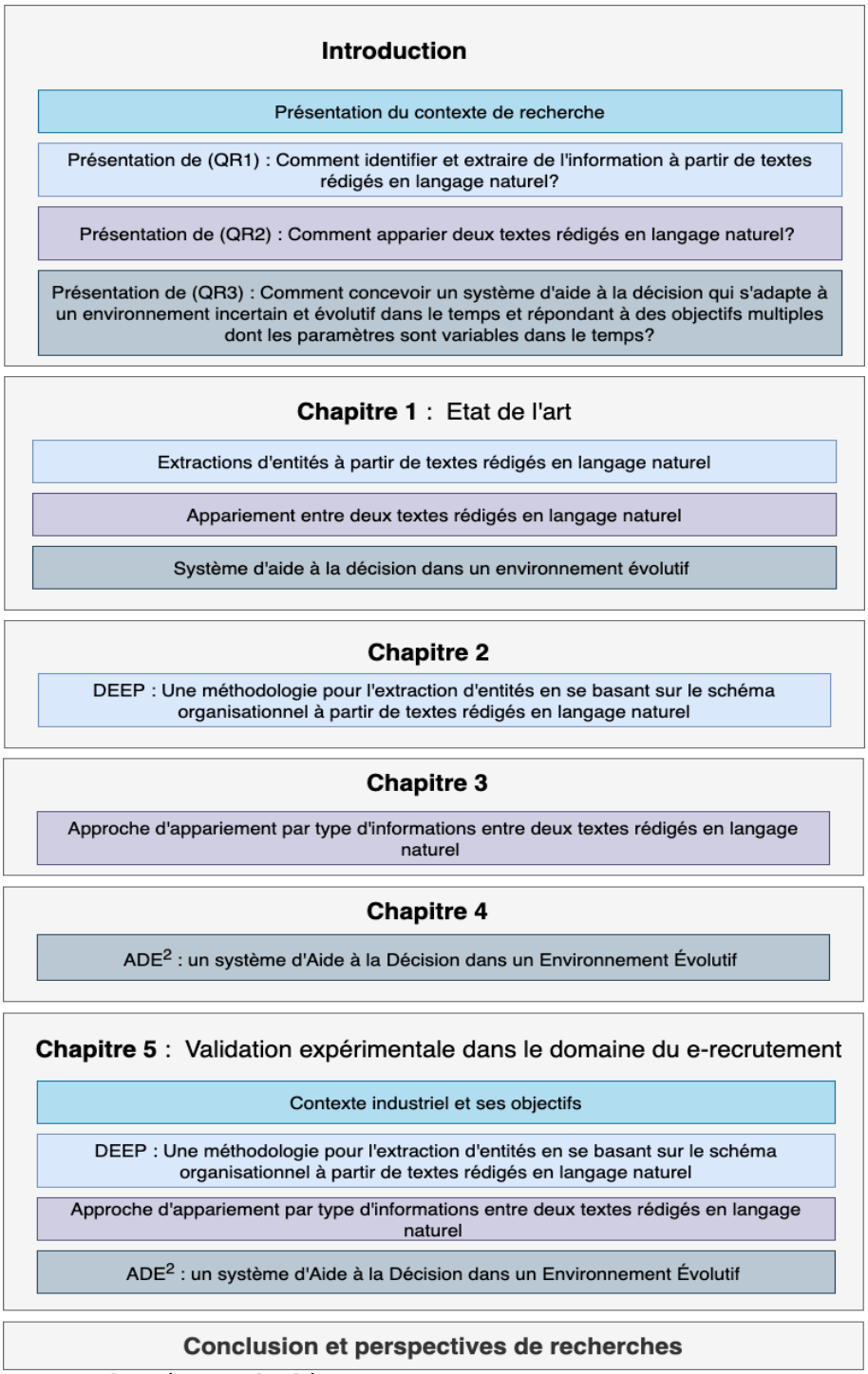


Fig. 1. Organisation du mémoire de thèse

1

État de l'art

Sommaire

Dans ce premier chapitre, nous présentons un état de l'art des travaux de la littérature pour chacune de nos questions de recherche (QR1), (QR2) et (QR3).

1.1 Extraction d'entités à partir de textes rédigés en langage naturel

Le premier objectif de nos travaux de recherche pour pallier la limite L1 est d'identifier le profil recherché dans une offre d'emploi. Une offre d'emploi est un texte rédigé en langage naturel, contenant différents types d'information. Rappelons que nous cherchons à répondre à la problématique suivante :

Pour répondre à cette problématique, nous nous sommes intéressés aux travaux de la littérature sur l'extraction d'entités à partir de textes rédigés en langage naturel. L'extraction d'entités est un processus de recherche d'informations dans un texte en identifiant, extrayant et déterminant toutes les étiquettes appropriées pour les mots ou séquences de mots de ce texte [148]. Cette tâche est critique, notamment pour les textes rédigés en langage naturel et est concernée par les spécificités du traitement du langage naturel, notamment : (1) l'ambiguïté du vocabulaire (2) l'évolution du vocabulaire (3) les fautes d'orthographe, etc. [109].

En outre, certains textes rédigés en langage naturel présentent des schémas organisationnels. Il s'agit de l'organisation des idées dans un texte [25, 56] en spécifiant les connexions logiques entre les idées et la subordination de certaines idées à d'autres [100]. De plus, le schéma organisationnel fournit un plan d'organisation qui non seulement guide l'auteur lors de la rédaction, mais aide également le lecteur à identifier les relations entre les idées dans le texte [100]. Les modèles organisationnels pour le texte sont également appelés structure du texte [132]. Les modèles organisationnels peuvent être utilisés comme une stratégie utile pour développer la compréhension des lecteurs [132, 25]. Nous choisissons d'aborder la tâche d'extraction d'entités sur des textes rédigés en langage naturel en tirant parti des schémas organisationnels d'un texte.

La littérature sur les approches d'extraction d'entités pour les textes rédigés en langage naturel considère généralement un sous-ensemble des caractéristiques du langage naturel (évolution du vocabulaire, ambiguïté, fautes d'orthographes, etc.), mais pas toutes à la fois [21, 44, 33]. Ce chapitre présente les différentes approches de la littérature permettant l'extraction d'entités :

— l'approche basée sur les règles
— l'approche basée sur l'ontologie
— l'approche basée sur l'étiquetage de séquences
— l'application de ces approches sur les offres d'emploi

Il se termine par une conclusion et pose une hypothèse pour répondre à l'objectif, ainsi qu'à la problématique que nous avons définie.

1.1.1 Approche basée sur les règles

Trois techniques fondées sur des règles sont classiquement utilisées :

Expressions régulières qui permettent d'identifier dans le document des chaînes de caractères, construites avec des caractères ou des métacaractères. C'est par exemple le cas des dates, codes postaux, expériences, etc. L'utilisation d'expressions régulières permet ensuite d'associer une chaîne identifiée à une catégorie ou à une étiquette. Cependant, toutes les séquences ne peuvent pas être représentées par une expression régulière. De ce fait, cette approche ne peut pas être généralisée.

Dictionnaire de mots est une énumération de mots associés à une étiquette. Par exemple, les mots "temps plein" et "temps partiel", etc. dans les offres d'emploi, sont associés à l'étiquette "type de contrat" [24]. Comme pour les expressions régulières, les dictionnaires de mots ont l'avantage d'être utilisés sur des documents textuels en langage naturel, quelle que soit leur structure. Cependant, cette technique ne peut pas être utilisée lorsque l'information à extraire ne peut être répertoriée de manière exhaustive dans un dictionnaire. Elle n'est pas non plus robuste aux ambiguïtés, notamment lorsqu'une séquence de mots peut être associée à deux étiquettes.

Les règles d'extraction. Contrairement aux deux approches précédentes, les règles d'extraction sont des techniques qui considèrent les schémas organisationnels du document pour générer les paires étiquette/séquence en associant un mot ou un ensemble de mots à une étiquette définie dans les règles d'extraction [57]. Les règles sont créées pour représenter le schéma organisationnel des textes par des règles représentant la séquence d'étiquettes. Un exemple de règle pourrait être : "Chaque section commence par un titre qui définit son contenu". Les règles d'extraction peuvent donc être utilisées sur des documents textuels écrits en langage naturel et, contrairement aux approches précédentes, cette approche utilise le schéma d'organisation du texte. Cependant, cette technique ne peut pas être appliquée lorsque le schéma organisationnel du texte est incertain.

L'approche à base de règles est utilisée dans divers domaines d'application, comme le recrutement [24] ou la santé [21]. Cependant, elle est limitée dans la mesure où elle ne peut pas prendre en compte les changements de vocabulaire (si un mot n'est pas représenté dans le dictionnaire des mots ou par une expression régulière, il ne peut pas être identifié et extrait). Elle ne prend pas non plus en compte l'ambiguïté du langage (si un mot ou une séquence peuvent être associés à deux étiquettes différentes). Enfin, les règles d'extraction ne peuvent pas être appliquées à des documents dont le schéma d'organisation est incertain.

1.1.2 Approche basée sur les ontologies

Une ontologie est un vocabulaire organisé représentant les connaissances d'un domaine. Ces connaissances sont souvent représentées comme un ensemble de concepts et d'instances (valeurs), organisés de manière hiérarchique et structurés par des relations [108]. L'extraction d'entités à partir de documents textuels à l'aide d'une ontologie consiste à associer une séquence de mots à un (des) concept(s) (étiquette). La littérature utilise soit des ontologies existantes [153] soit une ontologie dédiée conçue à cet effet [44].
La construction d'une ontologie consiste à créer des concepts et des relations spécifiques au domaine d'intérêt, et est très chronophage [3], car elle est manuelle et nécessite un expert du domaine. L'utilisation d'une ontologie existante permet donc de gagner du temps. Cependant, une telle ontologie peut ne pas contenir le niveau de détail souhaité ou fournir une correspondance parfaite avec le vocabulaire utilisé dans les textes, ce qui entraîne un manque de précision. Les deux types d'ontologies peuvent être appliqués à des documents en langage naturel. Cependant, chaque fois qu'un nouveau concept apparaît, il est nécessaire de l'ajouter à l'ontologie. Le coût associé à cette opération est important. Par

conséquent, l'approche ontologique ne permet pas une gestion automatique et simple de l'évolution du vocabulaire.

1.1.3 Approche basée sur l'étiquetage de séquences

1.1.3.1 Principe

L'approche d'étiquetage de séquences peut être appliquée à tout document textuel écrit en langage naturel [77, 40]. L'étiquetage de séquences considère un document comme une liste ordonnée de séquences, où une séquence est une série de mots. L'objectif est d'attribuer une étiquette par séquence [128]. Dans l'exemple "nous recherchons un développeur back-end pour notre application", la séquence "développeur back-end" est étiquetée "position". Cette approche comporte trois étapes [86] :

Constitution du corpus d'apprentissage Cette approche repose sur un corpus étiqueté, qui joue le rôle de corpus d'apprentissage. Cet étiquetage est généralement réalisé manuellement par des experts du domaine [40].

Enrichissement du corpus d'apprentissage est une étape qui fournit une représentation enrichie des mots. Cette étape effectue le traitement des données du corpus afin de retenir les informations utiles et de conserver les données précieuses [77]. Elle transforme les données dans un format qui sera d'autant plus efficace et adéquat. Un document en texte intégral écrit en langage naturel contient souvent des erreurs. Les données doivent être pré-traitées pour obtenir un modèle d'apprentissage robuste, plus précis et de haute qualité. Si les données du corpus ne sont pas de bonne qualité, des ambiguïtés peuvent apparaître et des résultats trompeurs peuvent être obtenus [41]. Le texte du corpus doit être tokenisé (découper le texte en unités lexicales élémentaires) et normalisé. La tokenisation divise les longues chaînes de texte en tokens (petits morceaux de mots). Les phrases plus longues peuvent être tokenisées en mots et les morceaux de texte peuvent être tokenisés en phrases, etc. La normalisation se réfère à une série de tâches qui rendent tous les textes égaux : conversion de tous les textes en majuscules ou minuscules, conversion des nombres en mots équivalents, etc. La normalisation permet au traitement de se dérouler de manière uniforme. La normalisation permet d'uniformiser le traitement. L'étymologie et la lemmatisation sont généralement utilisées pour la tâche de normalisation [145]. L'étape "enrichissement du corpus d'apprentissage" vise également à ajouter des informations supplémentaires, telles que l'étiquetage morphosyntaxique (marquage grammatical des mots) [138], l'estimation de la position du mot dans une phrase, ou de la position de la séquence dans le texte, etc. Ces informations permettent de considérer le contexte du mot dans le document et donc de maximiser la prise en compte de tout changement de vocabulaire.

L'application de l'algorithme d'apprentissage est l'étape qui permet d'apprendre les séquences et les étiquettes associées à partir du corpus d'apprentissage enrichi. Le modèle obtenu est ensuite utilisé afin d'étiqueter automatiquement de nouveaux documents [68]. L'étiquetage de séquences est abordé dans la littérature comme un problème d'apprentissage supervisé.

1.1.3.2 Choix de l'algorithme d'apprentissage

De nombreux travaux exploitent les machines à vecteurs de support (SVM) [77], ou des réseaux de neurones récurrents (RNN) [40]. Les SVM sont choisis pour leur simplicité. Ils sont utilisés, par exemple, pour classer les informations (code postal, salaire, etc.) d'une offre d'emploi [76]. Cependant, ils n'utilisent aucune information sur les schémas organisationnels du texte ou sur le contexte de chaque phrase. Pour relever le défi de capture de dépendances à court et à long terme entre les séquences, certains travaux se sont concentrés sur la mémoire à long terme (LSTM), un réseau de neurones récurrents avec une mémoire à court et à long terme. La capacité du RNN à traiter des séquences dans les deux sens (bi-LSTM) (enchaînement de gauche à droite et de droite à gauche) est avantageuse [84]. L'utilisation de la bidirectionnalité permet d'exécuter les séquences d'entrée de deux façons, l'une du passé vers le futur et l'autre, du futur vers le passé. Le modèle préserve ainsi les informations du passé et du futur, en considérant l'ordre des séquences. Plus récemment, certains travaux ont utilisé des algorithmes CRF (Conditional Random Field) [79]. Les CRF capturent les dépendances entre les étiquettes prédites à différentes étapes de l'apprentissage en analysant les probabilités de transition d'une étape à l'autre. De plus, plusieurs travaux ont utilisé des CRF couplés à des LSTM et des bi-LSTM pour leur adaptabilité à l'étiquetage de séquences. Les algorithmes CRF et les réseaux de neurones récurrents sont deux algorithmes qui ont permis une modélisation plus précise et efficace des étiquettes [40] par rapport à d'autres méthodes d'apprentissage, comme un simple réseau de neurones récurrents ou un SVM.

1.1.4 Application aux offres d'emploi

Rappelons que l'extraction d'entités dans une offre d'emploi a pour objectif d'identifier le profil souhaité dans l'offre, en attribuant des étiquettes aux informations de profil : type de contrat, compétences requises, formation requise, etc. Certaines des approches d'extraction d'entités mentionnées ci-dessus ont été étudiées dans le secteur du recrutement sur des offres d'emploi. C'est le cas de l'approche par règles utilisée avec des expressions régulières pour extraire les années d'expérience et avec un dictionnaire pour extraire les compétences [24]. Il a été montré que l'approche par règles permet de bien extraire les années d'expériences et les compétences contenues dans le dictionnaire. Néanmoins, pour les séquences non contenues, elle ne permet pas une extraction correcte.
La construction d'une ontologie destinée à la modélisation du vocabulaire des offres d'emploi a été proposée par Yahiaoui et al., 2006 [160]. Néanmoins, ce travail se limite à un secteur particulier du recrutement : informatique et télécommunications. Au vu de cet article, il est difficile d'en déduire la généricité de l'approche ontologique sur le domaine général du recrutement. Nous constatons cependant que cette démarche n'est pas facilement généralisable à l'ensemble des métiers et secteurs en raison de la complexité de construction d'une ontologie complète du domaine et qu'elle ne s'adapte pas à l'évolution du vocabulaire. Pourtant, cette caractéristique est importante dans le domaine du recrutement étant donné l'évolution très rapide des métiers et des compétences entraînant ainsi l'évolution du vocabulaire. Par ailleurs, Sidahmed et al., 2017 [137] ont proposé d'utiliser des référentiels

existants : ROME [10] pour pallier le problème de la complexité de construction. Ils ont utilisé le référentiel ROME pour attribuer une référence à une offre d'emploi rédigée en langage naturel. Cela offre un gain de temps significatif. Cependant, ces travaux se concentrent uniquement sur les métiers relatifs à l'expérience professionnelle, délaissant ainsi d'autres caractéristiques des offres telles que le savoir-être, le salaire, etc. Ces derniers éléments sont pourtant importants pour le recruteur. Nous pouvons dire que les travaux réalisés sur l'extraction d'entités dans les offres d'emploi à l'aide des approches par ontologie et par règles semblent pertinents. Néanmoins, ces travaux ne montrent pas l'adaptativité à l'évolution des métiers, des compétences, etc.

1.1.5 Conclusion

La revue de la littérature a montré que les approches basées sur les règles et les ontologies sont utiles dans la mesure où elles s'appuient sur des informations prédéfinies, telles que des modèles de recherche, des dictionnaires ou des règles. Cependant, elles ne peuvent pas prendre en compte l'évolution du vocabulaire, et elles n'utilisent pas (et ne peuvent pas utiliser) le schéma organisationnel du texte pour améliorer l'extraction. Ces approches nécessitent des itérations récurrentes dans le temps pour prendre en compte de nouveaux mots, de nouvelles phrases ou de nouvelles structures de documents. De son côté, l'étiquetage de séquences est une approche qui peut prendre en compte l'évolution du schéma organisationnel et du vocabulaire si le corpus d'apprentissage est bien créé, et si les algorithmes d'apprentissage sont bien choisis. Néanmoins, si le corpus utilisé pour l'étiquetage automatique de séquences n'est pas de haute qualité (un corpus est de haute qualité s'il contient le moins d'ambiguïtés possible), des résultats de moindre qualité peuvent être obtenus [41]. Notre première contribution a pour hypothèse : (RQ1-H) L'étiquetage de séquences est une approche qui permet de tirer parti des schémas organisationnels, tout en tenant compte de l'évolution et de l'ambiguïté du vocabulaire et en améliorant la qualité de l'extraction automatique des entités.

1.2 Appariement entre deux textes rédigés en langage naturel

L'appariement est la création de paire de textes qui se correspondent [160]. Dans la suite du document, nous voyons l'appariement de deux textes comme l'évaluation de la similarité entre eux. Plusieurs approches ont été étudiées dans les travaux de la littérature. Rappelons que notre problématique de recherche est :

QR2

Comment apparier deux textes rédigés en langage naturel ?

10. Répertoire opérationnel des métiers et des emplois `https://www.pole-emploi.fr/employeur/vos-recrutements/le-rome-et-les-fiches-metiers.html`

Pour répondre à cette problématique, nous nous sommes intéressés aux travaux de la littérature. Nous proposons dans la suite du document de faire un état des approches les plus populaires sur l'appariement entre des textes rédigés en langage naturel. Rappelons que l'appariement est la création de paire de textes qui se correspondent [160]. Dans la suite du document, nous voyons l'appariement de deux textes comme l'évaluation de la similarité entre eux. Ce chapitre est composé de cinq sections principales :

— l'approche lexicale,

— l'approche par ontologies,

— l'approche basée par apprentissage automatique,

— l'exploitation de ces approches au recrutement,

— conclusion et hypothèse pour répondre à l'objectif et la problématique définis.

1.2.1 Approche lexicale

La similarité par approche lexicale repose sur la comparaison des chaînes de caractères que composent un mot ou un ensemble de mots. Il s'agit d'une métrique qui mesure la similarité ou la dissimilarité entre deux ou plusieurs chaînes de caractères [24]. Par exemple, les chaînes de caractères "arbre" et "arbuste" peuvent être considérées comme similaires, alors que “conducteur” et “chauffeur” seront considérés comme différents. L'appariement de deux textes par approche lexicale repose sur deux étapes principales :

1. Représentation des textes dans un espace vectoriel : cette représentation débute par le prétraitement du texte [96]. Cette étape vise à éliminer les mots vides, la ponctuation, etc. [50]. Elle vise aussi à transformer les mots par leurs racines, etc. La seconde étape vise à calculer le poids des mots qui composent le texte. Différentes techniques sont possibles. Nous pouvons citer par exemple le TF-IDF (Term Frequency-Inverse Document Frequency) qui est une méthode de pondération qui permet de préciser l'importance d'un terme contenu dans un document relativement à un ensemble de documents [124, 50]. Nous pouvons aussi citer la méthode booléenne qui renseigne si un mot est contenu dans un document, ou encore la méthode de la fréquence des mots, qui représente la fréquence des mots dans un document [124, 50].
2. Calcul de la distance entre les vecteurs : cette étape vise à calculer la distance entre les vecteurs créés à l'étape précédente. Cette distance représente la similarité entre les textes. Nous pouvons citer le cosinus [10] comme mesure de similarité qui se calcule de la façon présentée dans la définition 1. Nous pouvons aussi citer, la distance euclidienne, le coefficient de Jaccard [55], distance de Levenshtein [101] ou encore l'indice de Dice [22] comme mesure de similarité. Ces mesures ont l'avantage d'être simples à développer néanmoins, elles peuvent éloigner des documents pourtant proches puisque la dimension sémantique n'est pas prise en compte comme vu dans l'exemple plus haut.

Définition 1 La similarité cosinus repose sur le calcul du cosinus de l'angle entre les représentations vectorielles des documents T1 et T2 à comparer. La similarité obtenue $Sim(T1, T2) \in [0; 1]$

$$Sim(T1, T2) = \frac{\overrightarrow{T1} \cdot \overrightarrow{T2}}{||\overrightarrow{T1}|| \cdot ||\overrightarrow{T2}||}$$

1.2.2 Approche sémantique

L'approche sémantique se base sur la ressemblance de la signification des concepts et le contenu sémantique [28]. Nous pouvons citer deux approches pour mesurer la similarité sémantique. L'approche vectorielle et l'approche statistique. Dans l'approche vectorielle nous pouvons citer deux approches populaires : le Bi-clustering qui est une technique non supervisée permettant de créer un vecteur sémantique pour chaque mot d'une phrase [39] ou encore les vecteurs sémantiques [28] qui créent un contexte vectoriel pour chaque texte. L'approche statistique diffère de l'approche vectorielle, car elle ne nécessite pas la compréhension du vocabulaire d'un texte. Les mesures les plus utilisées sont le LSA (Latent Semantic Analysis) et le LDA (Latent Dirichlet Allocation). LSA permet d'établir des relations entre un ensemble de documents et les termes qu'ils contiennent en créant un contexte, en construisant des « concepts » liés aux documents et aux termes [149]. Le LDA est un modèle probabiliste qui permet de décrire des collections de documents de texte ou d'autres types de données discrètes et qui cherche à découvrir des structures thématiques cachées dans les documents [15]. Cette approche considère la thématique sémantique du texte. Néanmoins, il est difficile de l'utiliser pour comparer deux mots ou deux séquences ayant moins de contextes [113]. Pour pallier ces limites, nous nous sommes spécifiquement intéressés aux représentations vectorielles de texte qui permettent de considérer le sens d'un mot ou d'un ensemble de mots. Il existe plusieurs modèles de représentation vectorielle sémantique dans la littérature. Pourtant, celui qui a permis d'avoir les meilleurs résultats pour la distance sémantique est BERT (Bidirectional Encoder Representations from Transformers) [39]. BERT est un modèle de langage développé par Google en 2018. Les travaux de la littérature ont démontré les performances significatives de ce modèle en traitement automatique des langues. BERT a les caractéristiques suivantes :

1. Performance : il est plus performant que les autres modèles de langages ;
2. Rapidité d'apprentissage : il est plus rapide que les autres modèles de langages ;
3. L'entraînement se fait de façon non supervisée. Il a été entraîné sur le corpus anglophone de Wikipédia ;
4. Les domaines spécifiques : il est nécessaire de faire un entraînement sur un vocabulaire spécifique au domaine ciblé.

Il existe d'autres modèles de vocabulaire pré-entraînés tels que GPT-2, XLNet, etc. Cependant, ils ne sont pas adaptés à la recherche de similarité sémantique. Une approche courante pour considérer la sémantique dans la tâche de similarité entre deux séquences consiste à représenter chaque séquence dans un espace vectoriel, de sorte à calculer la distance entre ces deux vecteurs. Néanmoins, le vecteur construit pour chaque séquence ne

contient pas les informations sémantiques [122] "Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks". En effet, pour deux séquences similaires, les vecteurs obtenus peuvent être très différents. En outre, non seulement il existe des différences significatives dans la sémantique de la représentation des vecteurs de séquences, mais cette méthode nécessite de trouver la paire de séquences la plus similaire à partir de l'ensemble des phrases du modèle pré-entraîné nécessitant un temps de calcul très haut. Face à ce problème pour la tâche de similarité sémantique de texte, [122] proposent Sentence-BERT (SBERT) pour modifier le BERT pré-entraîné : en utilisant des réseaux de neurones siamois. Cette structure du réseau permet de générer une représentation vectorielle pour chaque paire de séquences d'un texte permettant par la suite de calculer la similarité en cosinus ou en Manhatten/Euclidean entre deux séquences. Le modèle SBERT a été entraîné sur plusieurs sources de données telles que Wikipédia, etc. Néanmoins, sur des domaines ayant des vocabulaires spécifiques, ce modèle pré-entraîné peut ne pas suffire [146].

1.2.3 Approche par ontologies

L'appariement de deux textes par approche par ontologies est appelée alignement d'ontologies et repose sur le calcul de la similarité entre deux concepts ontologiques [159]. Cette approche nécessite d'avoir en amont représenté un texte en concepts ontologiques pour pouvoir appliquer l'alignement ontologique. L'alignement de deux ontologies revient à trouver une correspondance entre leurs entités qui sont sémantiquement similaires [48]. De façon plus formelle, il est défini comme suit [167] :

Définition 2 Soient O et O' deux ontologies à aligner, t le seuil minimal de similarité appartenant à l'intervalle $[0, 1]$. Soient $e_1 \in O$, $e'_1 \in O'$. Soient e_1 et e_2 les entités au niveau des deux ontologies. L'alignement est défini par la fonction map comme suit :

$$map : O \Leftrightarrow O' \quad \text{tel que} \quad map(e_1) = e'_1 \quad \text{si} \quad sim(e_1, e'_1) > t.$$

La littérature du domaine propose plusieurs méthodes d'alignement d'ontologies. Ces méthodes exploitent différents formats d'ontologies et reposent sur le calcul des méthodes de calcul de similarité. Dans les travaux de la littérature, nous retrouvons [167] :

- La méthode terminologique qui consiste à comparer les étiquettes des entités. Cette comparaison se base sur l'approche lexicale présentée dans la section précédente ;
- La méthode de comparaison qui consiste à comparer les structures internes (intervalle de valeurs, attributs, etc.) ou les structures externes (relations des entités avec d'autres, etc.) ;
- La méthode de comparaison des instances qui consiste à comparer les extensions des entités en comparant les autres entités qui lui sont attachées.

L'approche d'appariement par ontologies a l'avantage de considérer la sémantique des textes pour le calcul de leurs similarités. Néanmoins, cette approche nécessite dans un premier temps la représentation d'un texte en concept ontologique. Pourtant, cette étape est fastidieuse comme nous l'avons vu dans le chapitre précédent. De plus, si la représentation des textes est fausse cette étape entraînera des erreurs dans l'étape de calcul de similarité.

1.2.4 Approche par apprentissage automatique

Il existe deux types de méthodes possibles pour répondre à cet objectif. L'appariement par apprentissage non supervisé consiste à apprendre sans données d'entrées (données annotées : correspondance entre deux textes positive classe = 1 ou nulle classe = 0 par exemple). Si deux textes font partie de la même classe alors ils sont considérés comme similaires et donc proches. Cette approche est utilisée dans plusieurs domaines tels que le biomédical pour créer des groupes de données liées au cancer [111], la recherche d'informations dans les réseaux sociaux pour regrouper les tweets similaires [11], etc. Cette approche a l'avantage de ne pas nécessiter un corpus annoté. Nous pouvons citer le clustering K-means qui est l'un des algorithmes non supervisés les plus utilisés [4, 151]. Ces travaux ont montré l'intérêt et l'efficacité de cet algorithme à condition que le choix des clusters soit pertinent. En effet, l'algorithme K-means ne permet pas de développer un ensemble optimal de clusters et il est important d'attribuer le bon nombre de classes pour ne pas voir ses résultats de prédictions non satisfaisants.

L'appariement par apprentissage supervisé consiste à attribuer à une donnée d'entrée une annotation. Par exemple, attribuer à une paire de textes la classe 1 s'ils sont similaires et 0 s'ils ne le sont pas. Il s'agit ici d'un problème de classification binaire [20]. Plusieurs algorithmes peuvent être utilisés : SVM (que nous avons défini dans le chapitre d'extraction d'entités), les réseaux de neurones, etc. Cette approche s'est avérée efficace [32, 117]. En revanche, elle nécessite un corpus annoté, autrement, il n'est pas possible de l'utiliser.

1.2.5 Application au recrutement

Dans le domaine du recrutement l'appariement permet d'évaluer les aptitudes d'un candidat par rapport à une offre [54] ou d'une offre par rapport à une offre [130]. Plusieurs travaux ont reposé sur **l'approche lexicale**. Certains travaux ont associé une représentation vectorielle statistique utilisant l'approche par TF-IDF ou occurrences des mots à des mesures de similarité telles que le cosinus [75, 140], l'indice de Dice [22], Minkowski ou encore Okabiso [76] pour calculer la distance entre deux textes. Ces méthodes peuvent indiquer une première valeur de similarité néanmoins, comme mentionné précédemment, les mesures de similarités ne considèrent pas le domaine sémantique des mots dans le document entraînant ainsi la possibilité de mettre de côté des CV pertinents pour une offre d'emploi. Par exemple, un recruteur recherchant un "développeur " et un CV contenant le poste occupé "informaticien" ne vont pas être considérés comme proches en utilisant la méthode lexicale. Golec et Kahya, 2007 [54], proposent un modèle de recherche (fuzzy model) pour évaluer et sélectionner les candidats selon leurs compétences, dans une perspective de limitation de biais et de subjectivité dans le processus de recrutement. Leur modèle de correspondance est structuré en deux niveaux : (1) évaluation (2) sélection. Le niveau inférieur (d'évaluation) évalue l'employé selon des indicateurs pour les différents facteurs représentant les objectifs de l'organisation, en utilisant un algorithme heuristique. Le niveau supérieur sélectionne le candidat en utilisant l'approche "floue" à base de règles pour chacun des facteurs. Bien que cette approche présente une certaine logique et fournisse des justifications sur les choix de sélection des candidats, elle se base sur un élément clé qui est la définition des objectifs du recruteur. Ceci pose une limitation à cause de la spécificité de cette information clé, propre au recruteur, et implique une barrière sur le passage

à l'échelle et l'étendue du système puisqu'elle nécessite la connaissance des objectifs du recruteur. En revanche, cette approche peut être adaptée en faisant l'hypothèse que les objectifs du recruteur sont contenus dans l'offre d'emploi. La même approche est proposée dans la littérature [92, 78]. Celle-ci utilise des données spécifiques telles que l'expérience et les compétences pour sélectionner les candidats parmi un ensemble de CV. Les deux approches visent à accélérer l'efficacité en qualité et en rapidité de la recherche de candidats pour le poste à pourvoir. Une présélection, basée sur des critères indispensables par le recruteur, a été mise en place pour filtrer l'ensemble des CV afin qu'elles puissent aider à améliorer la rapidité du système.
Bien que l'approche par règles soit facile à mettre en œuvre, le contenu du texte brut apporte du bruit dans la recherche d'information, conduisant à une faible précision et à des résultats de classement insatisfaisants. Certaines approches ont préféré l'utilisation d'une **ontologie** pour l'appariement des candidats et des offres d'emploi. Un système de mapping automatique entre une ontologie pour les offres d'emploi et une autre pour les CV a été proposé [152]. Le classement des candidats est réalisé selon une fonction de similarité entre les propriétés des deux ontologies. Cette approche nécessite le traitement d'une large quantité d'offre d'emploi et de CV, avec toujours le risque de ne pas couvrir tous les métiers ou autres caractéristiques nécessaires au recrutement. De plus, une maintenance continue de l'ontologie sera nécessaire entraînant ainsi un temps supplémentaire dans la mise en place de l'appariement. Un autre type d'approche se base sur les systèmes de recommandation. Un autre système de recommandation bilatérale a été proposé pour faire correspondre des candidats à des emplois [94]. Ces travaux considèrent que la recommandation de personnes à des emplois est un processus bilatéral qui doit prendre en compte non seulement les préférences du recruteur, mais aussi celles du candidat. Pour cela, ils ont développé deux systèmes : un pour la recommandation des CV et un pour la recommandation des emplois. Ce système de recommandation hybride (basé sur le contenu et par filtrage collaboratif) est fondé sur un modèle de classe latente qui assimile les préférences individuelles en tant que combinaison convexe de certains facteurs de similarité. Les récents travaux de la littérature sur l'appariement des offres et des CV privilégient les approches basées sur **l'apprentissage automatique**, surtout l'utilisation de l'apprentissage profond (*deep learning*) pour la représentation des documents.
Dans les recherches de [93], le problème d'appariement s'inscrit dans le cadre d'un système d'apprentissage supervisé où sont utilisés des réseaux de neurones profonds (*Deep Neural Networks*) pour trouver les meilleurs candidats pour une offre d'emploi. Les auteurs proposent une adaptation siamoise d'un réseau de neurones convolutionnel. Les architectures siamoises permettent d'apprendre une mesure de similarité à partir de deux entrées indépendantes qui partagent une relation abstraite de similarité. Les réseaux de neurones siamois ont été premièrement utilisés dans [20] pour la vérification de signatures. Ce type d'architecture a la particularité de faire intervenir deux réseaux de neurones identiques, partageant les mêmes paramètres, qui prennent deux entrées indépendantes et qui se rejoignent finalement grâce à une fonction de pénalité. Cette fonction se base sur une distance calculée à partir des représentations de plus haut niveau des deux réseaux. Dans leur approche, les auteurs s'appuient sur un ensemble de données annotées par paire de documents (associer la classe 0 si l'offre d'emploi correspond au CV, la classe 1 sinon). Bien que les résultats exposés montrent une bonne performance, cette approche nécessite

d'avoir à disposition des paires de CV et offres d'emploi déjà mis en correspondance par un recruteur, qui est une limite pour reproduire ces expérimentations.
Pour surmonter le problème lié aux annotations, [14] ont formé des paires de descriptions d'expériences professionnelles extraites de CV en combinant les descriptions d'un même emploi dans un CV pour former des paires positives et les descriptions d'emplois de CV différents pour former les paires négatives. Le modèle BERT [39] de classification de paires de textes a ensuite été utilisé pour prédire le classement de candidats pour une offre d'emploi. Cette approche est intéressante puisque l'approche ne nécessite pas de données annotées : CV et offre préalablement mis en correspondance.

1.2.6 Conclusion

Les travaux de la littérature nous ont permis de faire les constats suivants :

— l'approche lexicale a l'avantage d'être simple à implémenter. Néanmoins, elle ne considère pas la sémantique des textes. Ceci peut être limitant dans le domaine du recrutement puisque les métiers évoluent, entraînant l'évolution du vocabulaire utilisé dans les CV et les offres d'emploi ;

— L'approche sémantique a l'avantage de considérer la sémantique des textes. Néanmoins, certaines techniques reposent sur une méthode statistique et thématique qui entraînent la difficulté de comparer sémantiquement deux mots ou deux séquences. La méthode basée sur un modèle de vocabulaire pré-entraîné afin de créer un vecteur sémantique est une solution. Cependant, ces modèles ne sont pas entraînés sur des domaines spécifiques, il est donc nécessaire de les adapter au domaine d'application souhaité ; dans notre cas le e-recrutement.

— L'approche par ontologies est coûteuse en temps. De surcroît, l'utilisation d'une ontologie et de bases de connaissance pour la recherche contraint l'exploitation de telles approches étant donné le besoin de grandes quantités de documents permettant de créer ces bases ;

— L'approche par apprentissage donne des résultats satisfaisants. Par ailleurs, elle nécessite un corpus annoté et les modèles utilisés sont opaques entraînant ainsi un manque de transparence envers le recruteur. Cependant, il est très important de mettre en avant la transparence du système d'appariement pour le recruteur ainsi que les critères permettant d'aboutir à une recherche réussie d'un CV ou d'une offre d'emploi.

1.3 Systèmes d'aide à la décision

Les systèmes d'aide à la décision (SAD) sont largement utilisés pour résoudre les problèmes de sélection et de prise de décision dans de nombreux domaines. Ces systèmes aident les décideurs à prendre une décision lorsque cette sélection nécessite l'intervention d'expertise, de connaissance ou de système intelligent [23]. On peut par exemple citer la santé pour améliorer les soins et les traitements prescrits aux patients [5]. Le domaine du recrutement s'y intéresse aussi pour la gestion du recrutement au sein d'une entreprise [51]. Certains travaux de la littérature se sont intéressés aux systèmes de recommandation

pour recommander les canaux de diffusion à des recruteurs [130, 137, 136]. Les systèmes de recommandation ont pour objectif de recommander à un utilisateur des items pertinents en fonction de ses choix antérieurs (préférences). Dans nos travaux, nous n'avons pas retenu une modélisation de notre problème sous la forme d'un système de recommandation. Nous avons tout d'abord écarté la mise en place d'un système de recommandation où le recruteur est considéré comme l'utilisateur. En effet, les préférences du recruteur dépendent de la rentabilité d'un canal, c'est-à-dire si le budget investi sur le canal a permis de répondre aux objectifs. Ceci supposerait que le recruteur connaisse suffisamment tous les canaux, y compris leur évolution, ce qui n'est pas le cas étant donné l'environnement incertain et évolutif du e-recrutement. Si d'un autre côté, nous considérons une offre d'emploi comme l'utilisateur d'un système de recommandations des canaux, nous écartons la possibilité de considérer les objectifs du recruteur.

Dans les travaux de [130] les auteurs considèrent les canaux de diffusion comme des utilisateurs et les offres d'emploi comme les items. Cependant, ce choix de modélisation ne permet pas de prendre en compte les objectifs du recruteur ainsi que ses contraintes et de baser les recommandations sur une fonction d'optimisation d'un ou plusieurs objectifs. De ce fait, dans le cadre de nos travaux, le problème de sélection des canaux ne peut pas être modélisé comme un système de recommandation, mais comme un problème d'optimisation.

Ainsi, nous avons procédé à une étude de l'état de l'art sur les SAD pour répondre à la problématique suivante :

QR3

Comment concevoir un système d'aide à la décision qui s'adapte à un environnement incertain, évolutif et répondant à des objectifs multiples dont les paramètres sont variables ?

Pour répondre à cette problématique, nous disposons de base de données contenant un historique de décision (sélection des items) et leurs résultats. Ainsi, nous sommes intéressés aux modèles de prédiction de série temporelles et aux algorithmes d'apprentissage.

Ce chapitre est composé de cinq sections principales :

— Définitions des entités d'un système d'aide à la décision ;
— Présentation des composantes d'un système d'aide à la décision ;
— Présentation des domaines d'application ;
— Modèles d'aide à la décision ;
— Conclusion et hypothèse pour répondre à l'objectif et la problématique définie.

1.3.1 Définitions

1.3.1.1 Une décision

Selon Mitroff et Betz, 1972 [103], le processus de décision est décrit comme une série d'étapes, commençant par la production et l'analyse d'informations et aboutissant à la résolution, à savoir une sélection parmi plusieurs alternatives disponibles. Dans le même ordre d'idée, Bernard et Denis, 1993 [125] estiment que la décision est le fait que des acteurs exercent librement un choix entre plusieurs possibilités d'actions à un moment donné dans le temps.

1.3.1.2 L'aide à la décision

L'aide à la décision [127] est généralement utilisée par des industriels lorsqu'ils sont confrontés à des problèmes complexes, par exemple, les investisseurs qui sont face à un problème de sélection d'actions parmi un ensemble d'actions. Nous pouvons citer un autre exemple plus actuel, qui implique des responsables d'une entreprise qui sont face à un problème de sélections de personnes pouvant faire du télétravail sans contraintes professionnelles ou personnelles. Ce type de problème induit donc une décision (ou un ensemble de décisions) lourde de conséquences [2]. Dans les travaux de la littérature, un modèle de prise de décision composé des trois étapes clés a été proposé [139] :

1. Identifier toutes les alternatives possibles ; Cette étape consiste à analyser et identifier les alternatives possibles parmi une série de décision ;
2. Déterminer toutes les conséquences possibles à ces alternatives ; Cette étape consiste à assigner à chaque alternative les conséquences possibles ;
3. Évaluer toutes les conséquences possibles.

Bernard et Denis, 1993 [125], définissent l'aide à la décision comme l'activité d'un acteur appelé analyste [150] à : (1) obtenir des éléments qui aident à la prise de décision d'un intervenant appelé décideur [95], (2) construire un modèle de décision cohérent par rapport aux objectifs et au besoin du décideur. Ce processus d'aide à la décision qui a pour objectif d'accompagner un décideur lorsqu'il est confronté à des problèmes complexes nécessitant de l'aide se décompose comme suit :

- Représentation du problème. Cette étape a pour objectif de déterminer (1) les acteurs principaux qui participent à ce processus d'aide à la décision : les décideurs et les analystes, (2) leurs rôles dans le processus, (3) les préoccupations et objectifs du décideur par rapport au problème posé, (4) l'importance de chaque préoccupation et leurs liens.

- Formulation du problème. Le décideur identifie lors de cette étape, l'ensemble des actions possibles (sur quoi va porter la décision). "Une action désigne un objet, une décision désigne un candidat" [35]. C'est sur cette action qu'un traitement de sélection ou classement est opéré [2]. L'ensemble des actions peut être, par exemple, dans le cadre d'aide à la décision pour des investisseurs de vendre et acheter. L'investisseur qui est le décideur, doit pouvoir sélectionner la meilleure alternative.

- Construction du modèle d'évaluation. Cette étape nécessite l'intervention du décideur afin de construire le modèle d'évaluation sur lequel il va se baser pour répondre au problème et construire ses recommandations. Cette étape passe par la définition et la caractérisation des différentes alternatives, des critères d'évaluation et des méthodes.

- Construction des recommandations finales. Cette étape nécessite l'intervention de l'analyste afin d'évaluer et vérifier si les recommandations finales sont cohérentes par rapport aux préoccupations et objectifs du décideur. Pour cela, l'analyste s'assure, premièrement, à travers une analyse de sensibilité, que le résultat n'est pas

trop sensible à des variations jugées non significatives des données utilisées pour la construction du modèle d'évaluation [2]. Ensuite, l'analyste interprète le résultat en prenant en considération des hypothèses sur les données, le modèle d'évaluation et l'évolution du processus de décision.

1.3.1.3 Un système d'aide à la décision

Un système d'aide à la décision (SAD) (Decision Support System (DSS) en anglais) est un système d'information qui aide un ensemble d'utilisateurs dans les activités de prise de décision. Le système d'information assiste les utilisateurs en analysant d'énormes volumes de données non structurés dont l'analyse par un humain est complexe et nécessite une expertise. Les DSS recueillent, analysent et synthétisent les données pour produire des rapports d'information complets permettant ainsi aux utilisateurs de prendre des décisions plus éclairées et plus rapidement. Un SAD est soit entièrement informatisé, soit alimenté par des humains. Dans certains cas, il peut combiner les deux. Les systèmes entièrement informatisés (autonomes) analysent les informations et prennent des décisions pour l'utilisateur. Enfin, les DSS hybrides utilisent les deux types de DSS. Tout DSS vise à fournir des avantages aux utilisateurs à différents niveaux, comme suit [65] :

1. améliorer l'efficacité personnelle,
2. accélérer le processus de prise de décision,
3. augmenter le contrôle de l'organisation,
4. encourager l'exploration et la découverte de la part du décideur,
5. accélérer la résolution des problèmes,
6. faciliter la communication interpersonnelle,
7. favoriser l'apprentissage ou la formation,
8. générer de nouvelles preuves à l'appui d'une décision,
9. créer un avantage concurrentiel par rapport à la concurrence,
10. révéler de nouvelles approches de réflexion sur l'espace du problème,
11. aider à automatiser les processus de gestion.

1.3.2 Les composantes d'un système d'aide à la décision

Les quatre composantes fondamentales de l'architecture d'un SAD [142, 65] sont :

1. La base de données (ou base de connaissance), dans laquelle toutes les données sont stockées ;
2. Le modèle (c'est-à-dire, le contexte de décision et les critères de l'utilisateur), qui peut comporter différentes étapes : (1) Définir le contexte de décision et les critères, (2) nettoyer, analyser et structurer les données (3) créer le modèle d'aide à la décision ;
3. l'interface utilisateur : une interface utilisateur est le système à travers lequel les utilisateurs interagissent avec une machine. L'interface utilisateur comprend des composants matériels (physiques) et logiciels (logiques). Les interfaces utilisateur existent pour divers systèmes et fournissent un moyen d'entrée permettant aux utilisateurs de manipuler un système ;

4. les utilisateurs : les utilisateurs sont également des composants importants du système. Ils sont appelés décideurs, car ce sont les personnes qui à partir des trois premiers composants prennent une décision.

1.3.3 Domaines d'applications

Dans le cadre du recrutement, des systèmes d'aide à la décision ont été mis en place pour répondre aux besoins tels que le choix de candidats pour un poste donné [123, 94, 74] ou inversement la recommandation des offres à des candidats [37, 42]. Néanmoins, pour l'aide aux choix des canaux de diffusion et leurs paramètres, les travaux de la littérature se sont intéressés aux systèmes de recommandation [13, 137, 130] qui sont difficilement modélisables lorsque les préférences du recruteur ne sont pas connues et qui ont des limites (présentées dans l'introduction de ce chapitre). Nous nous intéressons ici aux systèmes d'aide à la décision appliqués aux domaines connexes du e-recrutement, c'est-à-dire qui ont les caractéristiques présentées dans l'introduction de ce mémoire de thèse :

— (QR3-C1) L'environnement est partiellement connu, entraînant des conséquences incertaines. De ce fait, l'environnement est incertain entraînant une prise de décision dans l'incertain qui est définie comme suit :

> **Définition 3** La décision dans l'incertain traite des situations de choix où le décideur est amené à prendre des décisions dans un environnement incertain (c'est-à-dire, les conséquences des décisions ne sont pas connues avec certitude). Le décideur ne possède aucune information objective sur la vraisemblance des événements.

— (QR3-C2) L'environnement est évolutif entraînant l'évolution dans le temps des variables du système entraînant un environnement non stationnaire. Définissons la stationnarité d'une série chronologique [60] :

> **Définition 4** Une série chronologique ou temporelle est non stationnaire, lorsque le processus qui la décrit ne vérifie pas au moins une des conditions d'un processus stationnaire, donnée par :
>
> — $E(Y_t) = m$ indépendant du temps
>
> — $V(Y_t) = \mu(0) < \infty, \mu(0)$ indépendant du temps
>
> — $Cov(Y_t, Y_{t-h}) = \mu(h)$ ne dépend pas de t

— (QR3-C3) Les objectifs du décideur à atteindre sont multiples entraînant la mise en place d'un système s'appuyant sur un modèle de connaissance représenté par le besoin du décideur et basé sur une optimisation multi-objectifs ;

— (QR3-C4) Contexte de décision rédigé en langage naturel. Le contexte de décision textuel est défini comme suit :

> **Définition 5** Un contexte de décision de nature textuelle est un contexte dans lequel sera prise la décision rédigée dans un texte.

L'état de l'art appliqué au recrutement pour la recommandation des canaux ne considère pas toutes ces caractéristiques (comme présenté dans l'introduction). De ce fait, nous avons décidé de nous focaliser sur les domaines connexes ayant une ou plusieurs caractéristiques en commun avec ceux définis ci-dessus. Nous nous sommes intéressés plus particulièrement aux travaux dans le domaine de la publicité et de la bourse :

- **Le marché de la publicité.** Les systèmes d'aide à la décision dans ce domaine ont pour objectif, par exemple, d'aider les entreprises à évaluer le prix optimal pour une publicité qui permettra de maximiser le revenu. D'autres travaux ont proposé d'aider les entreprises à décider du contenu de la publicité qui permettra de maximiser l'impact [115, 157]. Le marché de la publicité est influencé par différents facteurs économiques. De même que pour le recrutement et le marché de la bourse, le marché de la publicité, les objectifs peuvent être multiples (QR3-C3), les décisions sont prises dans l'incertain (QR3-C1), la résolution du problème nécessite une prédiction des paramètres (QR3-C3), ces paramètres évoluent dans le temps (QR3-C2), et les contextes de décision peuvent être textuels (QR3-C4) par exemple le contenu de la publicité, la description de l'entreprise pour le choix des actions en bourse.
- **Le marché de la bourse.** Les systèmes d'aide à la décision dans ce domaine consistent à aider un investisseur à anticiper les fluctuations des prix de l'action dans le temps pour l'aider à prendre une décision parmi, par exemple, l'achat et la vente. Le prix des actions évolue dans un environnement incertain qui est le marché de la finance et de la bourse. En effet, de nombreux facteurs sont à l'origine de la fluctuation du prix des actions par exemple l'évolution de l'entreprise qui émet ces actions, l'économie mondiale, etc. Ces facteurs de fluctuations difficilement modélisables rendent la tâche d'analyse des actions de plus en plus difficile (QR3-C1, QR3-C2). De plus, le contexte de choix des actions dépend des besoins et souhaits de l'investisseur (QR3-C3).

Les données du marché boursier ou de la publicité sont souvent traitées comme une série chronologique affectée par de multiples facteurs variables. Ces facteurs peuvent généralement être classés en deux types : les variables macroscopiques, qui affectent le marché à long terme (par exemple, les politiques économiques, l'évolution du marché, etc.), et les variables microscopiques, qui affectent le marché à une échelle microscopique (par exemple, les événements aléatoires, la subjectivité des investisseurs, etc.). En analysant les facteurs macroscopiques et microscopiques, des décisions peuvent être prises. Cependant, comme ces facteurs évoluent, nous pouvons considérer qu'ils sont influencés par un facteur inconnu. De plus, il est presque impossible de collecter toutes les variables macroscopiques et microscopiques et de déterminer l'étendue de ces variables. Par conséquent, la prédiction des séries chronologiques est souvent considérée comme l'un des domaines de recherche les plus difficiles dans la littérature relative aux séries temporelles et à l'apprentissage automatique. Dans la suite de ce chapitre nous allons nous intéresser aux modèles d'apprentissage utilisés dans la littérature pour l'aide à la décision dont les caractéristiques sont (QR3-C1) et (QR3-C2). Les travaux de la littérature sur l'extraction d'entités à partir de textes rédigés en langage naturel ont permis d'aborder les approches pour répondre à la caractéristique (QR3-C4).

1.3.4 Modèles d'aide à la décision

La conception d'un modèle d'un système d'aide à la décision dont les caractéristiques sont celles citées plus haut, est un défi extrêmement difficile [164]. En effet, les données récoltées pour ce type de problème contiennent généralement des informations complexes, incomplètes et floues, la prédiction de leurs tendances de développement est complexe. Pourtant, celle-ci a un impact significatif sur la prise de décision [155]. La prédiction des données futures a été largement utilisée dans divers domaines pour savoir comment prédire des événements incertains dans le futur et pour utiliser des informations déjà existantes pour projeter un avenir incertain [72]. La prévision de données évoluant dans le temps et dans l'incertain est l'un des objectifs fondamentaux de la modélisation statistique [72], et reste importante dans de nombreux domaines de recherche et d'application. Les fluctuations des données dépendent de facteurs difficilement modélisables. Dans la suite de ce chapitre, nous allons analyser les modèles d'aide à la décision utilisés dans les domaines cités précédemment qui ont pour objectif de prédire des évènements futurs en analysant l'historique de données afin de faire les recommandations au décideur. Ces évènements sont représentés par des variables dépendant du temps et de facteurs externes (séries temporelles).

Définition 6 Une série temporelle (ou chronologique) X_t est une suite d'observations $x_1, x_2, ..., x_n$ indexée par le temps. Une série temporelle X_t est communément décomposée en :

- une tendance T_t correspondant à une évolution à long terme de la série, par exemple :
  - tendance linéaire : $T_t = a + b_t$
  - tendance quadratique : $T_t = a + b_t + c_t^2$
  - tendance logarithmique : $T_t = log(t)$
- une saisonnalité S_t correspondant à un phénomène périodique de période identifiée
- une erreur ϵ_t qui est la partie aléatoire de la série (idéalement stationnaire)

Cette décomposition peut-être additive $X_t = T_t + S_t + \epsilon_t$ ou multiplicative $X_t = T_t \times S_t \times \epsilon_t$. Il est également possible de combiner ces deux décompositions : $X_t = (T_t + S_t) \times \epsilon_t$ ou $X_t = (T_t + S_t) + \epsilon_t$ etc.

Il existe deux types de séries temporelles :

1. La série temporelle univariée est une série comportant une seule variable dépendant du temps. Par exemple, l'échantillon de données dans la Figure 1.1 consiste en des valeurs de la température (chaque heure), pour les 2 dernières années. Ici, la température est la variable dépendante du temps. Les séries temporelles univariées ne considèrent pas les variables influençant potentiellement la température. Pourtant, dans le cas de plusieurs domaines : énergie, bourse, finance, recrutement, etc., des facteurs peuvent influencer la variable dépendant du temps.
2. La série temporelle multivariée est une série temporelle qui comporte plus d'une variable dépendant du temps. Chaque variable dépend non seulement de ses

valeurs passées, mais aussi d'autres variables. Par exemple, l'échantillon de données dans la Figure 1.2 inclut les autres variables pouvant influencer la température. Cette dépendance peut être utilisée pour prédire les valeurs futures.

Étant donné le caractère incertain de l'environnement de décision, notre intuition est que la modélisation de données temporelles en série temporelle multivariée permettrait de considérer des facteurs pouvant influencer la variation des données temporelles. Il existe plusieurs algorithmes qui ont été appliqués et analysés sur ce type de problème dans la littérature. Nous pouvons les classer comme suit : Modèles linéaires et un cas particulier de modèle non linéaire : l'apprentissage profond.

Table 1.1
Valeurs par heure de la température pour les deux dernières années

Temps	**Température (°C)**
5 am	10
6 am	12
7 am	14
1 pm	25
2 pm	28
3 pm	27
4 pm	26
8 pm	20

Table 1.2
Valeurs par heure de la température et caractéristiques de la météo pour les deux dernières années

Temps	**Température (°C)**	**Couverture nuageuse (%)**	**Humidité (%)**
5 am	10	30	45
6 am	12	29	42
7 am	14	29	43
1 pm	25	17	30
2 pm	28	13	30
3 pm	27	13	35
4 pm	26	13	32
8 pm	20	18	42

1.3.4.1 Modèles linéaires

Les modèles linéaires ont pour objectif (i) d'évaluer l'existence d'une relation entre les variables X et une ou plusieurs variables explicatives T, S, ϵ ; (ii) quantifier l'effet

des variables explicatives sur Y ; (iii) prédire X en fonction des variables explicatives. Dans la suite de cette section nous allons présenter quelques exemples de modèles linéaires qui ont été appliqués dans le domaine de la bourse ou de la publicité.

ARMA (Autoregressive–moving-average) est un modèle autorégressif et moyenne mobile qui combine le processus autorégressif (AR) et le processus de moyenne mobile (MA) et construit un modèle composite de la série chronologique. Le modèle ARMA est un cas particulier d'un modèle beaucoup plus général nommé ARIMA (en anglais) où le I désigne "Intégré" [59]. En effet, le modèle ARMA ne permet de traiter que les séries stationnaires [59]. Les modèles ARIMA permettent de traiter les séries non stationnaires après avoir déterminé le niveau d'intégration (le nombre de fois qu'il faut différencier la série avant de la rendre stationnaire) [135]. Comme l'indique l'acronyme, ARIMA(p,d,q) capture les éléments clés du modèle [135] :

— Auto-régression (AR). Un modèle de régression qui utilise les dépendances entre une observation et un certain nombre d'observations décalées (p).

— Intégré (I). Pour rendre la série temporelle stationnaire en mesurant les différences entre les observations à différents temps (d).

— Moyenne mobile (MA). Un type de moyenne statistique utilisé pour analyser les séries temporelles, en supprimant les fluctuations transitoires de façon à en souligner les tendances à plus long terme.

Les différents modèles linéaires ARMA et ARIMA et leurs variations [99, 29] utilisent des équations prédéfinies pour adapter un modèle mathématique à une série temporelle univariée. L'avantage de ces modèles est la qualité de leurs prédictions lorsque les séries sont relativement courtes et que le nombre d'observations n'est pas suffisant pour appliquer des méthodes plus flexibles.
Le principal inconvénient de ces modèles est qu'ils ne tiennent pas compte de la variation des données due à des facteurs externes et à un contexte. Comme ils ne considèrent que des séries temporelles univariées, les interdépendances entre les différents paramètres ne sont pas identifiées par ces modèles. De plus :

— les valeurs manquantes peuvent réellement affecter les performances des modèles ils ne sont pas capables de reconnaître des modèles complexes dans les données

— ils ne fonctionnent généralement bien que pour les prévisions à quelques étapes, et non pour les prévisions à long terme

— il est incapable de traiter simultanément plus d'une variable (série).

Pour ces raisons, il n'est pas possible d'identifier les modèles ou les dynamiques présents dans l'ensemble des données [131]. Une comparaison entre ces modèles linéaires et plus spécifiquement ARIMA et les réseaux de neurones récurrents dans les travaux de [135] a été effectuée. Ces expérimentations ont montré que sur des données où l'environnement est incertain comme en bourse et en finance, que les réseaux de neurones récurrents performent mieux que le modèle ARIMA. Plus précisément, l'algorithme basé sur les réseaux de neurones récurrents a amélioré la prédiction des données futures de 85% en moyenne par rapport à l'ARIMA.

1.3.4.2 Un cas particulier de modèle non linéaire : l'apprentissage profond

Afin de pallier les limites des modèles linéaires des travaux se sont intéressés à des modèles non linéaires. La classe des modèles autorégressifs conditionnellement hétéroscédastiques ("autoregressive conditional heteroscedastic" ou ARCH) a été proposée [49]. Ces modèles sont utilisés dans la modélisation de séries temporelles financières, qui comportent des volatilités variables et incertaines avec des périodes agitées suivies par des périodes de calme relatif. Ces modèles ont l'avantage de capturer les périodes de volatilité variables, néanmoins, lorsque les moments où se produisent ces périodes sont stochastiques (ils se produisent de façon aléatoire) ces modèles ne peuvent pas être appliqués. Ces évènements sont pourtant récurrents lorsque les environnements sont incertains. De ce fait, d'autres modèles ont été proposés [18] par exemple le modèle GARCH (generalized ARCH) afin de pallier les limites du modèle ARCH. D'autres travaux de la littérature se sont intéressés aux réseaux de neurones pour la prédiction du prix des actions [164, 141] dans le domaine de la bourse ou encore la prédiction du taux de clics pour une publicité [157]. Ces différents types de modèles non linéaires ont été comparés dans les travaux de [91]. Ces travaux ont montré qu'un modèle de type apprentissage profond par réseau neuronal appliqué à des séries temporelles ont une meilleure capacité de prévision que les modèles GARCH.
Pour la suite de l'état de l'art, nous avons donc choisi de nous focaliser sur les réseaux neuronaux profonds pour résoudre les problèmes non linéaires de manière plus satisfaisante que les modèles non linéaires et linéaires présentés ci-dessus.

L'apprentissage profond est un réseau neuronal comportant au moins une couche cachée. Les neurones sont divisés en couches d'entrée, couches cachées et couches de sortie. Chaque connexion entre les neurones a un poids entraînable correspondant. Cette architecture permet de modéliser des fonctions non linéaires complexes et possède une capacité d'abstraction de haut niveau, ce qui signifie que le pouvoir d'ajustement du modèle est considérablement amélioré [66]. Il s'agit d'un type de modèle discriminant qui peut être entraîné par l'algorithme de rétropropagation. Les réseaux de neurones ont une plus grande capacité d'analyse de données imprécises et bruitées et sont largement utilisés dans la littérature pour prédire des données dépendant du temps [66]. Les réseaux neuronaux profonds peuvent être considérés comme des approximateurs de fonctions non linéaires capables de modéliser des fonctions non linéaires. Selon le type d'application, différents types d'architectures de réseaux neuronaux profonds sont utilisés. Nous allons présenter différentes architectures d'apprentissage profond pour la prévision des séries temporelles :

1. les réseaux neuronaux récurrents (RNN), qui sont l'architecture la plus classique [31] ;
2. mémoire à long terme (LSTM), qui est une évolution des RNN développée afin de surmonter le problème de mémoire à long terme qui n'est pas considéré dans les réseaux de neurones récurrents [118]. Il s'agit de l'architecture la plus utilisée pour les problèmes de prévision des séries chronologiques dans le domaine de la finance [64] ;
3. les réseaux de neurones convolutifs très utilisés pour le traitement des images.

Cette architecture de réseaux a récemment reçu l'intérêt de quelques chercheurs, pour la prévision des séries temporelles, pour leurs potentiels d'extraction des caractéristiques de l'environnement grâce à son architecture sous forme de graphes de différents types de données en entrées [64].

Nous nous sommes intéressés à l'analyse de ces modèles, car ils peuvent tous répondre à notre problématique de modèle non linéaire, non stationnaire et de données temporelles évoluant dans un environnement incertain.

Réseaux de neurones récurrents Les réseaux neuronaux récurrents sont des réseaux de nœuds de type neurones organisés en couches successives, avec une architecture similaire à celle des réseaux neuronaux standard. La différence est que dans ce cas, chaque neurone est affecté à un pas de temps fixe [118]. Les neurones de la couche cachée sont également transmis dans une direction dépendante du temps, ce qui signifie que chacun d'entre eux n'est entièrement connecté qu'avec les neurones de la couche cachée ayant le même pas de temps assigné et est connecté avec une connexion à sens unique à chaque neurone assigné au pas de temps suivant. Les neurones d'entrée et de sortie sont connectés uniquement aux couches cachées avec le même pas de temps assigné. Puisque la sortie de la couche cachée d'un pas de temps fait partie de l'entrée du pas de temps suivant, l'activation des neurones est calculée dans l'ordre temporel : à tout pas de temps donné, seuls les neurones affectés à ce pas de temps calculent leur activation.

En général, les RNNs résolvent de nombreux problèmes des modèles traditionnels d'apprentissage automatique pour la prévision des séries temporelles [112, 166, 162, 133] :

— La performance des RNNs n'est pas affectée de manière significative par les valeurs manquantes [112] ;

— Les RNNs peuvent trouver des modèles complexes dans les séries temporelles d'entrée [112] ;

— Les RNNs donnent de bons résultats dans la prévision de plus de quelques étapes [162] ;

— Les RNN peuvent modéliser une séquence de données de sorte que chaque échantillon peut être supposé dépendre des précédents [166].

Néanmoins, lorsqu'ils sont entraînés sur de longues séries temporelles, les RNN souffrent généralement du problème du gradient qui disparaît ou du gradient qui explose, ce qui signifie que les paramètres des couches cachées ne changent pas beaucoup ou qu'ils entraînent une instabilité numérique et un comportement chaotique [64]. Cela se produit parce que le gradient de la fonction de coût inclut la puissance de W, ce qui affecte sa capacité de mémorisation. L'apprentissage d'un réseau neuronal récurrent est difficile à paralléliser et est également coûteux en calcul. Cette architecture souffre d'une mémoire faible : incapable de prendre en compte plusieurs éléments du passé dans la prédiction du futur [64]. Face à ces inconvénients, diverses extensions des RNN ont été conçues pour rogner sur la mémoire interne : réseaux de neurones bidirectionnels [129], LSTM [61], GRU [34], etc. L'élargissement de la mémoire peut

être crucial dans certains domaines tels que la finance, où il est fondamental de mémoriser le plus d'historique possible afin de prédire les prochaines étapes.

Réseaux de neurones récurrents à mémoire court et long terme Les réseaux de neurones récurrents à mémoire court et long terme (LSTM) ont été conçus afin de surmonter le problème de mémoire courte des RNN. Dans un réseau LSTM, l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. De ce fait, ils sont plus proches du vrai fonctionnement du système nerveux, qui n'est pas à sens unique. Ces réseaux possèdent des connexions récurrentes. Elles conservent des informations en mémoire : ils peuvent prendre en compte à un instant t un certain nombre d'états passés. Ceci est réalisé en utilisant une unité LSTM à la place de la couche cachée. Une unité LSTM est composée de :

— une cellule d'état, qui apporte des informations tout au long de la séquence et représente la mémoire du réseau. Cette cellule permet de maintenir un état aussi longtemps que nécessaire. Elle consiste en une valeur numérique que le réseau peut piloter en fonction des situations.

— une porte d'oubli, qui décide de ce qui est pertinent de conserver des étapes précédentes ;

— une porte d'entrée, qui décide si l'entrée doit modifier le contenu de la cellule ;

— une porte de sortie qui décide si le contenu de la cellule doit influer sur la sortie du neurone.

Cette architecture permet ainsi de pallier les problèmes du RNN.

Selvin et al., 2019 [131] ont utilisé les deux architectures LSTM et RNN pour prévoir le prix des entreprises cotées en bourse et ont comparé leurs performances. Les résultats finaux ont montré que le LSTM sur-performe le modèle RNN pour prédire le prix des actions, car il peut identifier la tendance du changement grâce à la mémoire court et long terme du LSTM. Dans le domaine de la publicité, cette architecture a été utilisée pour prédire le rapport entre le nombre d'utilisateurs qui cliquent sur l'affiche de la publicité et le nombre total d'utilisateurs qui consultent la page de la publicité (CTR) [115, 30]. Les expérimentations sur des corpus réels de contenu publicitaire démontrent que l'approche basée sur l'architecture LSTM peut augmenter la précision de la prédiction du CTR plus efficacement que les modèles traditionnels tels que les modèles linéaires ou des réseaux de neurones récurrents. En effet, ces résultats démontrent la performance supérieure des modèles LSTM dans la capture de la dynamique spatiale et temporelle des données, fournissant aux décideurs des modèles robustes pour leurs prises de décision. Dans le domaine du recrutement cette architecture a été utilisée [136] afin de prédire les valeurs futures pour une offre d'emploi sur des canaux de diffusion.

Réseaux de neurones convolutifs Le Réseau de neurones convolutif (Convolutional neural network - CNN) a été largement utilisé dans le domaine de la reconnaissance d'images en raison de sa puissante capacité de reconnaissance des formes [63] ; son utilisation a également été étendue au domaine de la prédiction des valeurs futures des séries chronologiques. Similaire au réseau neuronal traditionnel, le CNN

est composé de plusieurs neurones connectés par une structure hiérarchique, et les poids et les biais entre les couches peuvent être formés. Le CNN est différent de la structure d'un réseau entièrement connecté tel que le Deep Brief Network (DBN), le Sparse Autoencoder (SAE), la rétropropagation (BP), car le CNN peut partager le poids entre les neurones de chaque couche du réseau. De ce fait, le réseau CNN a le potentiel de tirer parti des caractéristiques implicites de données d'entrée et les représenter sous forme de graphes [63]. Les réseaux de neurones convolutifs ont été appliqués au domaine du marché boursier. En effet, les caractéristiques du marché boursier ont été considéré comme un graphique de caractéristiques [131]. Ainsi, le CNN a le potentiel d'extraire les caractéristiques du marché boursier à la période correspondante à partir de ces graphiques de caractéristiques. Le réseau de neurone convolutif performe bien pour les séries chronologiques lorsqu'elles sont multivariées [156]. Dans le domaine de la publicité, ces résultats ont été confirmés [64]. En outre, les annonceurs fournissent généralement des informations contextuelles utiles pour leurs campagnes de publicité, telles que des descriptions textuelles, des lieux de ciblage et des dispositifs, qui présentent une corrélation élevée avec le CTR mais qui ne sont pas utilisées dans les architectures présentées ci-dessus. Ces données sont utilisées comme contexte afin de capturer la forte non-linéarité et les informations locales des séries temporelles, ainsi que la corrélation sous-jacente entre les séries temporelles du CTR et les informations contextuelles [53]. L'efficacité du modèle avec une architecture CNN a été démontrée dans un ensemble de données publicitaires Yahoo du monde réel. Le modèle a été déployé en production avec un roulement quotidien de celui-ci.

1.3.4.3 Apprentissage par renforcement

L'apprentissage par renforcement est une méthode qui consiste à récompenser les comportements souhaités et/ou à sanctionner les comportements non désirés [144]. Il est parfois utilisé pour l'aide à la décision dans le domaine de la publicité [168] ou encore de la bourse [85]. Cette méthode d'apprentissage a été adoptée dans le domaine de l'intelligence artificielle afin de diriger l'apprentissage automatique non supervisé à l'aide de récompenses et de pénalités. Elle permet notamment de modéliser des problèmes décisionnels séquentiels [46]. Dans cette approche, un agent apprend à prendre des décisions optimales en interagissant avec l'environnement [107, 106]. Lorsqu'il effectue une action, l'état du système change et l'agent reçoit une valeur scalaire, appelée récompense, qui encode les informations sur la qualité de la transition. L'apprentissage par renforcement peut être caractérisé par les étapes suivantes :

1. l'agent observe un état d'entrée,
2. une action est déterminée par une fonction de prise de décision,
3. l'action est effectuée,
4. l'agent reçoit un résultat en fonction de son environnement,
5. les informations sur le résultat donné pour cet état ou action sont enregistrées,
6. la récompense est calculée selon le résultat des actions,

7. l'agent choisit une nouvelle action en fonction de la récompense des actions passées.

Parmi les premiers algorithmes d'apprentissage par renforcement, on compte le *Temporal difference learning* (TD-learning), proposé dans [144], et le *Q-learning* [158]. Ces algorithmes ont été évalués sur le domaine de la bourse et de la publicité [85, 134]. [85] a proposé trois méthodes différentes d'apprentissage par renforcement pour prédire le prix des actions. Les résultats ont montré que le modèle d'apprentissage par renforcement profond le plus performant est le *Deep-Q Network* (DQN). [134] a proposé un modèle d'apprentissage par renforcement combiné à LSTM et CNN. Le modèle a généré divers graphiques à partir de données boursières et les a utilisés comme entrées pour la couche CNN. Les caractéristiques extraites par la couche CNN ont été divisées en vecteurs colonnes et introduites dans la couche LSTM. L'apprentissage par renforcement a défini la structure, la récompense et l'action du réseau neuronal de la politique des agents et a fourni des probabilités d'achat, de vente et de maintien comme résultat final. Cette approche a permis de s'adapter aux fluctuations des données. L'apprentissage par renforcement a montré dans ces différents travaux sa capacité à s'adapter à l'environnement incertain. Cette approche ne nécessite pas non plus la caractérisation des items à recommander et de leur environnement puisque le choix des actions est tout d'abord effectué aléatoirement puis affiné au fur et à mesure de l'apprentissage. Les facteurs qui influencent l'environnement sont aussi gérés puisque le système apprend en temps réel. Néanmoins, cette approche est coûteuse en mémoire puisqu'elle doit stocker des valeurs à chaque itération. Sa mise en place nécessite la gestion des algorithmes en temps réel, ce qui pénalise les performances et l'interactivité. De plus, les premières itérations de cet algorithme ne sont pas pertinentes. Enfin, et c'est ce qui est le plus important, dans certains domaines tels que la bourse, la publicité ou le recrutement, cette méthode risque d'être fortement onéreuse en argent lors de l'initialisation de l'algorithme, en particulier pour les premières prises de décision, puisqu'une phase d'exploration plus ou moins longue est nécessaire pour que l'apprentissage converge vers un modèle qui répond aux besoins. De plus, lorsque les items sont multiples et le choix de ces items simultanément sans initialisation et pré-sélection impliquerait des surcoûts disproportionnés et des premières itérations avec des résultats mauvais.

1.3.5 Conclusion

Dans le cadre du recrutement, les systèmes d'aide à la décision de la littérature ne répondent pas aux caractéristiques de l'environnement du e-recrutement que nous avons défini en introduction (QR3-C1) à (QR3-C4). De ce fait, nous nous sommes intéressés aux domaines connexes qui ont des caractéristiques similaires. Dans le cas du marché boursier et de la publicité, les données générées sont énormes et hautement non linéaires. Pour modéliser ce type de données, des modèles capables d'analyser les modèles cachés et la dynamique sous-jacente sont nécessaires. Les algorithmes d'apprentissage profond sont capables d'identifier et d'exploiter les interactions et les modèles existant dans les données grâce à un processus d'auto-apprentissage. Contrairement à d'autres algorithmes linéaires (AR, ARMA, ARIMA, etc.) et non li-

néaires (ARCH, GARCH, etc.), les modèles d'apprentissage profond peuvent modéliser efficacement ce type de données et fournir de bons résultats. Nous nous sommes principalement intéressés aux RNN, LSTM et CNN. L'état de l'art effectué a montré que les CNN sont le modèle permettant le plus de considérer le contexte des séries temporelles permettant ainsi d'améliorer les résultats de prédiction. Néanmoins, ces modèles utilisent principalement les données historiques. Ces données peuvent pourtant contenir des biais, certaines classes de données peuvent être sur-représentées par rapport à d'autres entraînant ainsi une discrimination de certains items.

En conclusion, l'état de l'art montre que le système d'aide à la décision pour la recommandation d'items évoluant dans le temps et dans un environnement incertain, basé sur des modèles de réseaux de neurones profonds répondraient à notre problématique à condition d'avoir des données annotées, non biaisées et une quantité de données représentative pour la caractérisation des items et l'incertitude de l'environnement. Cette approche a par ailleurs l'avantage de générer un modèle qui permet de répondre à la problématique sans être coûteux ni en temps ni en argent. Quant à l'apprentissage par renforcement, il a l'avantage de s'adapter à un environnement incertain et de générer de nouvelles recommandations au décideur que l'apprentissage supervisé n'est pas capable de proposer si ces données ne sont pas annotées. De ce fait, dans la suite de nos travaux nous allons nous intéresser à la proposition d'un modèle hybride qui combine ces deux approches qui nous paraissent complémentaires.

2

DEEP : Une méthodologie pour l'extraction d'entités en se basant sur le schéma organisationnel à partir de textes rédigés en langage naturel

Sommaire

Les documents textuels constituent une partie essentielle et en croissance rapide de l'information en ligne dans divers domaines : biomédical, documents de recherche, recrutement, etc. [71, 143, 104]. Un simple texte brut peut contenir autant d'informations qu'une petite base de données structurée. L'extraction d'entités accélère le processus de recherche dans un texte en identifiant, en extrayant et en déterminant toutes les étiquettes appropriées pour les mots ou les séries de mots de ce texte (segment). Cette tâche est critique, en particulier pour les textes écrits en langage naturel et est de plus en plus concernée par les spécificités du traitement du langage naturel, y compris : 1) l'ambiguïté du vocabulaire 2) l'évolution du vocabulaire 3) les fautes d'orthographe, etc. La littérature sur les approches d'extraction d'entités pour les textes en langage naturel considère généralement un sous-ensemble de ces caractéristiques spécifiques, mais pas toutes à la fois [21, 44, 33]. D'ailleurs, la plupart des textes rédigés en langage naturel présentent des schémas organisationnels. Il s'agit

de l'organisation des idées dans un texte [25, 56] en spécifiant les connexions logiques entre les idées et la subordination de certaines idées à d'autres [100]. De plus, il fournit un plan d'organisation qui non seulement guide l'auteur lors de la rédaction, mais aide également le lecteur à identifier les relations entre les idées dans le texte. Les modèles organisationnels pour le texte sont également appelés structure du texte [132]. Les modèles organisationnels peuvent être utilisés comme une stratégie utile pour développer la compréhension des lecteurs [132, 25].

L'objectif de notre travail est d'automatiser la tâche d'extraction d'entités sur des textes rédigés en langage naturel. L'étiquetage des séquences présente depuis longtemps un intérêt particulier pour le traitement du langage naturel, comme le marquage des parties du discours ou l'annotation sémantique, etc. [121, 73, 43], pour sa capacité à considérer la série de mots comme une séquence et la série de séquences comme l'organisation du texte. La tâche d'étiquetage de séquences est une approche d'extraction d'entités qui attribue une étiquette catégorielle à chaque séquence en considérant un texte comme une séquence de mots sémantiques [88]. Par exemple, dans les recettes de cuisine, l'étiquetage typique du texte "Coupez la moitié de la banane dans le sens de la longueur. Si vous êtes vraiment courageux, vous pouvez aussi couper des tranches " pourrait attribuer l'étiquette " ingrédient " à la séquence " banane ", l'étiquette " quantité " à la séquence " moitié " et l'étiquette " action " aux séquences " couper dans le sens de la longueur " et " couper des tranches ". L'étiquetage automatique de séquences est une tâche d'extraction d'entités qui nécessite un corpus annoté [87] (séquences étiquetées à partir d'un corpus de texte brut) et qui part d'un corpus annoté, qui joue le rôle de corpus d'apprentissage. L'étiquetage des séquences pour l'extraction d'entités utilise un algorithme d'apprentissage supervisé pour automatiser davantage la tâche. Le corpus d'apprentissage, y compris l'étiquetage utilisé, est essentiel et influence la qualité de l'étiquetage des séquences [8, 52, 110]. Selon [40], l'étiquetage du corpus d'apprentissage est généralement effectué manuellement par des experts du domaine. Pourtant, cette tâche est coûteuse et longue, et elle est limitée par le besoin d'experts, qui peuvent ne pas être disponibles. Pour rendre la tâche d'étiquetage de corpus plus accessible, certaines plateformes Web (Kaggle[11], Data gouv[12], etc.) proposent des corpus prêts à l'emploi pour l'étiquetage de séquences. Cependant, l'utilisation de ces corpus présente certains inconvénients : 1) ils représentent des domaines spécifiques ; 2) ils ne sont pas personnalisables pour un besoin particulier ; 3) la taille du corpus peut ne pas être suffisante pour un besoin particulier.

Dans certains cas [105], le crowdsourcing peut être utilisé pour créer des corpus, nous pouvons citer, Amazon Mechanical Turk[13]. Le crowdsourcing est une activité participative en ligne dans laquelle un individu ou une entreprise, par exemple, propose à un groupe de personnes ayant des connaissances et des expériences diverses d'effectuer une tâche sur une base volontaire [105]. Néanmoins, la difficulté vient du fait d'avoir la même annotation entre annotateurs, en évitant la subjectivité dans la compréhension du texte brut et sans parti pris. Ces tâches de personnalisation

11. https ://www.kaggle.com/datasets
12. https ://www.data.gouv.fr/fr/datasets/all-datasets/
13. https ://www.mturk.com/

et de crowdsourcing nécessitent de donner des instructions claires qui peuvent être comprises et mises en œuvre indépendamment de la difficulté de cette tâche et du domaine d'expertise de chacun [105], surtout pour des textes rédigés en langage naturel.

Sur la base de ces contraintes, l'objectif de notre travail est de développer une méthodologie dédiée qui minimise le temps requis pour l'expert en tirant parti de ses connaissances du domaine pour créer des instructions claires pour les annotateurs, et qui contribue à réduire les incertitudes de l'annotation manuelle grâce au crowdsourcing, améliorant ainsi la qualité de l'extraction d'entités.

Comment identifier, extraire et normaliser de l'information à partir d'un texte rédigé en langage naturel ?

2.1 Extraction d'entités à partir d'un texte rédigé en langage naturel

2.1.1 Présentation des étapes de DEEP

DEEP vise à construire un corpus d'extraction d'entités de haute qualité afin d'éviter les problèmes liés à l'utilisation d'un corpus existant ou à la présence d'incertitudes qui surviennent lors de l'annotation manuelle d'un corpus. DEEP surmonte les caractéristiques spécifiques suivantes : *C1* : variabilité et incertitude de l'étiquetage manuel des séquences, notamment en raison des caractéristiques du langage naturel ; *C2* : ambiguïté entre les paires étiquette/séquence (incertitudes quant au choix entre deux étiquettes pour une séquence) ; *C3* : prise en compte de l'évolution du vocabulaire.

DEEP suit les trois étapes principales de la mise en œuvre de l'étiquetage des séquences, telles que définies dans la littérature :

Création du corpus d'apprentissage : Cette étape vise à créer le corpus étiqueté utilisé pour apprendre un modèle d'étiquetage automatique. La création du corpus est la première étape de l'étiquetage de séquences et a donc un impact sur les autres étapes et sur la qualité de l'extraction automatique d'entités. Cette étape constitue la principale contribution de DEEP. Elle sera développée dans les sections suivantes.

Enrichissement du corpus d'apprentissage : Cette étape sert à enrichir le corpus en appliquant un traitement de données pour ajouter des informations lexicales. La désambiguïsation lexicale est utilisée pour déterminer la catégorie grammaticale correcte de chaque mot en fonction de son contexte. Cette étape est réalisée par un développeur en apprentissage automatique.

Application de l'algorithme d'apprentissage : Cette étape vise à appliquer un algorithme d'apprentissage pour l'extraction automatique d'entités. Pour ce faire, elle exploite le corpus fourni par l'étape précédente. Quelques expériences seront donc nécessaires pour identifier les algorithmes les plus performants. Cette étape vise également à déterminer les entrées et sorties du modèle et à définir les méthodes d'évaluation et de validation des algorithmes choisis. La technique d'évaluation proposée

dans cette méthodologie est la validation croisée. Les ensembles de formation et de test sont sélectionnés de manière aléatoire. Les mesures d'évaluation proposées sont les mesures conventionnelles de la précision (P), du rappel (R) et du score F-mesure (F1). Cette étape est réalisée par un développeur d'apprentissage automatique.

2.1.2 Les acteurs impliqués

La méthodologie DEEP implique trois acteurs principaux ayant des expertises différentes. Ils sont généralement très occupés et ont peu de temps à consacrer à cette tâche, notre objectif est donc de minimiser le temps requis pour chacun d'entre eux, en particulier pour l'expert qui n'est impliqué que dans les tâches nécessitant une grande expertise. Ces acteurs sont présentés ci-après :
Le master gère la mise en œuvre de la méthodologie. Il/elle est familier(e) avec le domaine d'application spécifique et possède la connaissance et la compréhension de ses aspects essentiels.
L'expert possède une expertise d'un domaine d'application spécifique [98]. Le temps de l'expert est limité, et il ne peut pas consacrer beaucoup de temps aux annotations.
L'annotateur est une personne qui n'est pas spécifiquement familière avec le domaine. Cet acteur annote les documents manuellement en suivant les spécifications de l'expert et du master.
Le fait d'avoir ces trois acteurs différents permet également d'effectuer les différentes activités en parallèle, ce qui prend moins de temps. Nous proposons également d'impliquer des annotateurs qui ne sont pas familiers avec le domaine afin de faciliter et d'améliorer la vitesse de mise en œuvre du corpus. Cela dit, il est également possible d'avoir une seule personne qui réalise toutes les activités, mais le temps passé sera plus long.

2.1.3 La méthodologie : création du corpus d'apprentissage

Dans la méthodologie proposée, la première étape consiste à créer un corpus d'apprentissage composé de paires étiquette/séquence à partir des textes. La principale contribution de DEEP se situe dans cette étape qui est réalisée par les acteurs suivants : le master, l'expert et les annotateurs. Elle est représentée dans le diagramme d'activité de la Figure 2.1. Cette étape s'appuie sur un corpus de validation qui est utilisé pour préparer et valider les guides d'instruction (qui seront présentés ci-après) pour l'annotation manuelle. Le corpus de validation est également utilisé pour créer le corpus gold, comme on peut le voir sur la Figure 2.2. Le corpus gold (GSA) est un corpus annoté par un ou plusieurs experts. Le GSA est destiné à servir de corpus de référence pour évaluer l'annotation par des annotateurs non experts.
Cette étape est composée de trois activités principales qui sont décrites dans cette section et résumées dans la Figure 2.1.

2.1.3.1 Création du guide d'instruction pour les annotateurs (A1)

A1 contribue au développement d'une méthode unique d'étiquetage manuel d'un corpus en établissant un guide d'instruction pour les annotateurs. L'objectif est d'éviter,

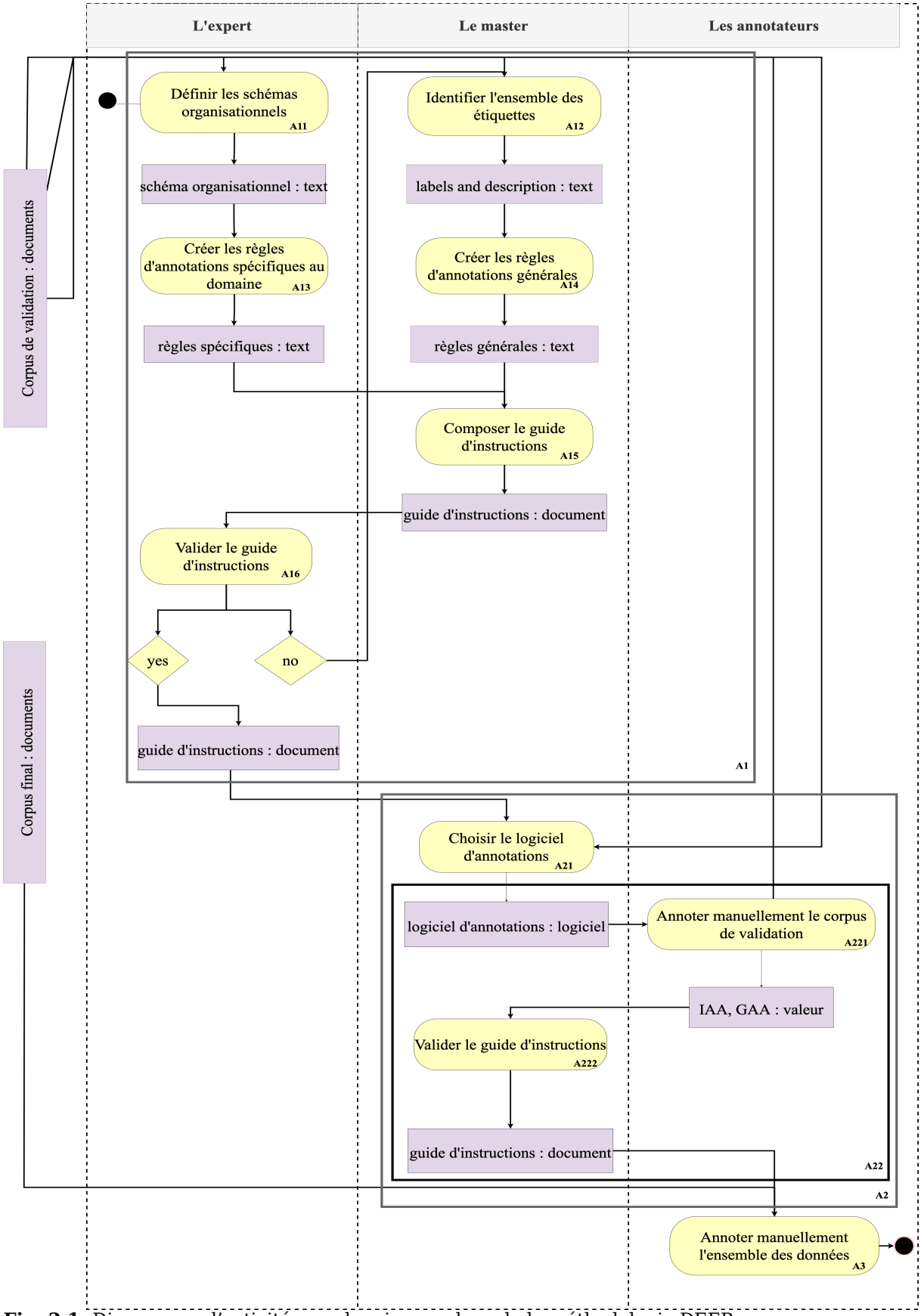


Fig. 2.1. Diagramme d'activité pour la mise en place de la méthodologie DEEP

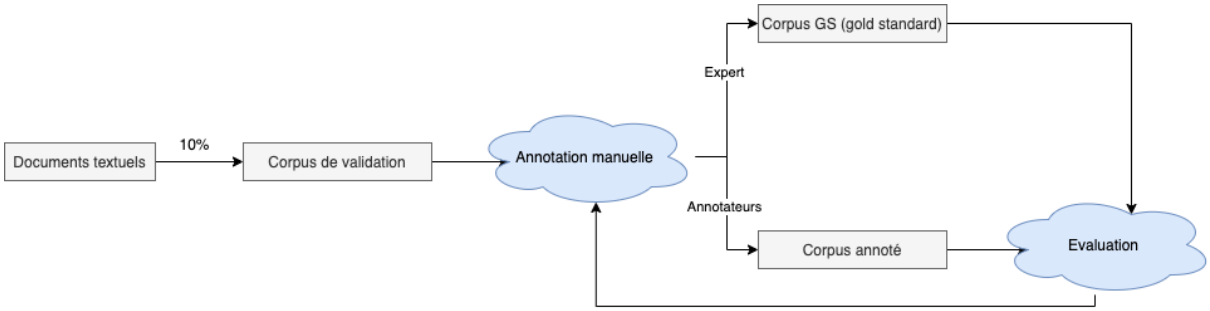


Fig. 2.2. Création du corpus de validation

dans la mesure du possible, les biais liés à la subjectivité de chaque annotateur et les ambiguïtés liées aux schémas organisationnels des textes. La création d'un tel guide d'instruction est d'autant plus critique lorsque l'étiquetage d'un corpus est réparti entre plusieurs annotateurs.

A1 est effectuée par le master et l'expert et comporte cinq sous-activités principales :

1. **A11. Définition des schémas organisationnels des textes**.
   A11 vise à définir un modèle de modèles organisationnels. Cette activité nécessite une connaissance du domaine et quelques années d'expérience pour le déterminer :
   - **L'ordre le plus courant des informations**. Par exemple, dans les recettes de cuisine, il existe un ordre séquentiel des informations. L'auteur suit un modèle étape par étape. Il commence par une liste d'ingrédients, et la suit avec une liste d'actions.
   - **Les différentes sections**. Un texte simple écrit en langage naturel est composé de différentes sections. Il est représenté par une série de sections sémantiques, où une section contient des types d'information spécifiques. Par exemple, dans un article scientifique, les sections sont les suivantes : Titre, Résumé, Introduction, Matériaux et méthodes, Discussions, Limitations, Remerciements, Références. Dans un CV, les sections sont : Identité, Éducation, Expériences, Loisirs.
   - **La description**. Il est fortement recommandé de décrire le contenu informatif de chaque section et le type de schéma organisationnel.
   - **Les indicateurs** sont les mots qui aident à identifier le type particulier de schéma organisationnel. Par exemple, dans l'introduction d'un article scientifique, les indicateurs peuvent être : contexte, problématiques, limites, difficulté, problème, défi, proposer, suggérer, résoudre, etc.
   - **Les exemples de séquences**. Par exemple, dans un résumé, un exemple de séquences peut être : "Dans l'ensemble, les changements qui ont eu lieu dans le hockey ont contribué à améliorer le jeu. Le hockey est plus passionnant grâce aux changements survenus au cours des 100 dernières années. Pour ces raisons, le hockey moderne est un meilleur jeu que le hockey des années 1880."

A11 est donc principalement menée par l'expert en interaction avec le master. L'expert propose un modèle d'organisation typique qui est la structure la plus commune entre les textes. Par exemple, dans l'article scientifique, "l'en-tête est toujours la première section. Il contient les noms, l'adresse, les affiliations, etc. Il est suivi par la section contenant le résumé ou l'introduction ", comme nous pouvons le voir dans l'exemple du guide d'instruction Annexe A. Nous supposons que l'exploitation des modèles organisationnels a l'avantage de limiter les ambiguïtés dans l'annotation manuelle, puisque chaque section représente un type d'information spécifique. Elle aide également les annotateurs à se familiariser avec ces modèles organisationnels afin de gagner du temps lors de l'étiquetage. En outre, l'activité **A11** est essentielle pour s'assurer que le document possède un modèle organisationnel permettant de prendre en compte l'évolution du vocabulaire, puisqu'il représente une série de séquences et de mots qui peuvent identifier la position d'une étiquette particulière dans le document afin de prendre en compte l'évolution du vocabulaire. Cependant, étant donné que les modèles organisationnels changent en fonction de l'auteur du texte, le modèle défini ne peut pas être généralisé à tous les textes, car le contenu et l'ordre des séquences peuvent différer. Par conséquent, les modèles organisationnels incertains devront être pris en compte dans l'annotation hétérogène du texte brut. En cas d'incertitudes, l'expert met en évidence l'ordre des sections qui peut générer des incertitudes (tout en annotant pour des personnes non expertes), afin de les considérer dans les activités suivantes comme des règles pour minimiser les ambiguïtés dans le guide d'instruction.

2. **A12. Identification de l'ensemble des étiquettes. A12** vise à identifier l'ensemble des étiquettes utiles pour l'extraction d'entités, leurs spécificités et leurs descriptions. Nous supposons que la liste des étiquettes est finie et peut être développée manuellement, en fonction d'un besoin spécifique et de la revue de la littérature. L'approche d'étiquetage de séquences permet d'identifier automatiquement les séquences associées à ces étiquettes. **A12** est effectuée par le master, qui a une vue d'ensemble des besoins. **A12** aide les annotateurs non experts à se familiariser avec les étiquettes et à associer les sections aux étiquettes.

3. **Création des règles d'annotation.**

   Cette sous-activité vise à définir une manière unique d'étiqueter en définissant des règles à la fois spécifiques au domaine et générales.

   (a) **A13. Création des règles d'annotation spécifiques au domaine**.
   A13 est pertinente lorsqu'il existe des règles spécifiques au domaine. **A13** est effectuée par l'expert pour son expertise dans le domaine et une vue détaillée des modèles organisationnels de **A11**. Il/elle crée des règles spécifiques au domaine qui peuvent aider à minimiser les ambiguïtés dans la structure incertaine lorsque les modèles organisationnels choisis ne peuvent pas être généralisés à tous les textes. La création de règles a l'avantage de minimiser les ambiguïtés pendant l'étiquetage manuel dans un domaine spécifique. Par exemple, dans certains cas, une séquence peut contenir deux étiquettes différentes. La séquence "développeur en python" est un métier,

mais contient une compétence technique. Cette séquence peut générer des incertitudes lors du choix entre les deux étiquettes. La règle associée pourrait être "si les séquences associées à une expérience contiennent une compétence technique ou générale, ne pas l'annoter comme une compétence, mais comme l'étiquette à laquelle elle est liée".

(b) **A14. Création de règles d'annotation générales**.

A14 ne nécessite pas d'expertise spécifique dans le domaine et est donc effectuée par le master. La création de règles générales permet de créer une manière commune d'étiqueter un document et d'éviter les ambiguïtés et la subjectivité. Les règles générales d'annotation sont classées en trois catégories :

1) Règles de ponctuation. L'objectif de cette catégorie est de créer des règles de ponctuation. Par exemple, "Inclure dans l'annotation la ponctuation des séquences". Cette catégorie de règles est importante pour s'assurer que les annotateurs ont la même façon d'annoter afin d'éviter les incertitudes.

2) Connecteurs de phrases. Cette catégorie de règles est utilisée pour niveler le statut de chaque étiquette/séquence : obligatoire ou non. Les connecteurs "ou" et "et" sont pratiques pour créer un profil à partir d'un texte en langage naturel.

3) Règles de langage naturel. Cette catégorie vise à définir quelques règles pour certaines séquences qui peuvent être écrites de différentes manières selon l'auteur du document. Par exemple, "Considérer dans la séquence associée à l'étiquette "Date" toutes les informations de l'année, du mois, du jour, même si elles sont écrites en lettres ou en unités". Cependant, les règles générales peuvent être exhaustives et dépendent des annotateurs. Certaines itérations peuvent être nécessaires pour prendre en compte les particularités du domaine.

(c) **A15. Composer le guide d'instruction.**
A15 est gérée par le master qui collecte les résultats des sous-activités précédentes et crée un document contenant ces informations. Ce document représente le guide d'instruction. Un exemple de guide d'instruction est disponible dans l'Annexe A.

(d) **A16. Validation du guide d'instruction**. **A16** peut être gérée par le master ou l'expert. Si l'expert ou le master valide le guide d'instruction, nous pouvons passer à l'activité suivante **A2**. Sinon, il y a une nouvelle itération de **A12** à **A14**.

2.1.3.2 Étiquetage manuel du corpus (A2)

A2 comprend trois sous-activités qui visent à valider chaque étape du guide d'instruction. L'objectif de l'activité **A2** est de valider le guide d'instruction en faisant en sorte que les annotateurs l'utilisent pour vérifier la cohérence et la clarté du guide d'instruction. Le corpus de validation est utilisé à cette fin.

1. **A21. Choix de l'outil d'annotation**.
   L'activité **A21** sert à choisir un outil d'annotation et est réalisée par le master. Des logiciels existent pour simplifier l'annotation en proposant des interfaces simples, incluant éventuellement une phase de pré-étiquetage automatique. On peut citer par exemple Dataturks[14] ou Gate[15]. Ces deux logiciels permettent une annotation manuelle en proposant une interface simple pour associer une étiquette à une séquence.
2. **A22. Pré-validation du guide d'instruction par les annotateurs**.
   (a) **A221. Étiquetage manuel du corpus de validation par les annotateurs**.
   A221 est supervisée par le master et réalisé par les annotateurs. En utilisant le logiciel choisi, les annotateurs étiquettent manuellement le même corpus de validation, en suivant le guide d'instruction de **A1**. Nous suggérons d'avoir au moins deux annotateurs pour pouvoir comparer les annotations.
   (b) **A222. Validation du guide d'instruction.**
   A222 est gérée par le master. A222 est basée sur l'évaluation de corpus étiquetés. Deux mesures sont utilisées : l'accord inter-annotateur (IAA), qui a été traditionnellement utilisé dans la littérature [7], et le gold accord inter-annotateur (GAA) que nous introduisons.

   L'IAA mesure à quel point plusieurs annotateurs prennent la même décision d'annotation (en suivant le guide d'instruction). Concrètement, il représente le pourcentage de choix qui se chevauchent entre les annotateurs. L'IAA représente le niveau de compréhension et d'accord dans l'annotation. Une valeur IAA faible signifie que les annotateurs ne sont pas d'accord sur les étiquettes à attribuer à une séquence. Cette conclusion signifie que les règles fournies dans le guide d'instruction peuvent créer une ambiguïté qui doit être modifiée pour cette étiquette.

   Le GAA est une mesure que nous avons introduite dans le cadre de ce travail. Elle compare le chevauchement entre les documents étiquetés par les annotateurs et le corpus gold (GSA), qui est un corpus annoté fiable provenant d'un expert [102]. Cette mesure est utilisée pour évaluer si les règles spécifiques au domaine sont bien définies dans le guide d'instruction. La mesure GAA permet de gagner du temps dans la création des règles en identifiant rapidement les ambiguïtés spécifiques au domaine. En fonction des valeurs IAA et GAA, le master valide ou modifie le guide d'instruction pour résoudre les ambiguïtés et les malentendus. Une valeur d'AAI supérieure à 79% (généralement considérée comme une valeur de citation élevée de l'AAI) et une valeur de GAA inférieure à l'AAI, par exemple, indiquent que les règles spécifiques au domaine dans le guide d'instruction ne sont pas bien définies. Le master doit donc les modifier et retourner à l'activité **A12**. Le master identifie les séquences qui ne chevauchent pas le corpus gold afin de créer une règle spécifique pour effectuer cette modification. Si le seuil est dépassé, le master valide le guide d'instruction.

14. Outil d'annotation : https ://dataturks.com/
15. Outil d'annotation : https ://gate.ac.uk/

2.1.3.3 Annotation manuelle du jeu de données final (A3)

Cette étape finale a pour but d'étiqueter le corpus final en utilisant le guide d'instruction produit dans **A1**. Comme le guide d'instruction a été validé, c'est-à-dire que l'IAA et le GAA sont maximaux, chaque document peut être annoté par un seul annotateur.

2.2 Normalisation des séquences

La normalisation textuelle a pour objectif d'obtenir une forme canonique pour les différents mots d'une même famille. La normalisation consiste à corriger certaines erreurs et à expliciter des informations manquantes ou complémentaires à l'aide parfois de ressources externes. Normaliser le texte avant de le stocker ou de le traiter permet d'effectuer des opérations sur ce texte. La normalisation de texte nécessite de savoir quel type de texte doit être normalisé et comment il doit être traité par la suite ; il n'existe pas de procédure de normalisation universelle.
Il existe deux types de tâche de normalisation :

— type primitif : date, langue et pays. Cette tâche est relativement aisée et peut s'appuyer sur des méthodes existantes.

  Elle est néanmoins indispensable pour pouvoir exploiter l'information extraite de manière fiable,

— type référence, qui consiste à associer la séquence à normaliser à une autre référence. Par exemple associer la séquence "volant de voiture" à la référence "système de directions". La normalisation de type référence permet d'attribuer une sémantique à une séquence pour faciliter les tâches de post-traitement de ces séquences. Néanmoins, cette normalisation nécessite la mise en place de différentes étapes.

2.2.1 Normalisation de type primitif

Plusieurs techniques peuvent être utilisées pour la normalisation de type primitif :

1. Systèmes itératifs à base de règles : qui consistent à établir une liste de règles. Nous pouvons visualiser quelques exemples dans la Table 2.1. Par exemple, lorsque nous souhaitons traiter des dates, il est important de leur attribuer la même normalisation.
2. Méthodes à base de dictionnaire de mots auxquelles sont associées leurs formes canoniques. Nous pouvons visualiser quelques exemples dans la Table 2.2. Dans cette table par exemple la forme canonique de la séquence "contrat à durée indéterminée" peut-être "CDI".

2.2.2 Normalisation de type référence

Il existe des techniques de normalisation de type référence qui consistent à attribuer une catégorie à un ensemble de mots ou de séquences. De la même façon que pour

Table 2.1
Normalisation de type primitif

Variantes	Exemple	Forme canonique
Ponctuations	U.S.A	USA
Diactriques	Tuebengen	Tubingen
Majuscules	mit	MIT
Dates	23/06/2021	2021-06-23
Prix/Salaires	35K eur	35 000 EUR

l'extraction d'entités nommées, il existe des approches à base de règles et les classifications. Les approches à base de règles sont très pertinentes lorsque les règles peuvent être généralisées. Dans le cas contraire, nous nous sommes principalement intéressés à la classification puisque cette approche permet de considérer l'évolution du vocabulaire et considère la sémantique.

1. Choix du référentiel. Cette première étape consiste à choisir le référentiel. Par exemple, ROME [16] est un référentiel de compétences dans le domaine du recrutement. Ce référentiel recense, référence et décrit la totalité des compétences d'une structure. Dans le domaine pharmaceutique il existe aussi un référentiel « liste en sus » répertoriant la dénomination commune internationale (DCI) associée au nom de marque de chaque médicament, le libellé des indications prises en charge ainsi qu'un code indication. Le choix du référentiel est important étant donné qu'il forme la base de la normalisation de type référence.
2. Annotation semi-automatique. Cette seconde étape consiste à répertorier les techniques permettant d'associer à chaque séquence sa référence en se basant sur le référentiel choisi. Plusieurs techniques peuvent être utilisées telles que :
   (a) la recherche par syntaxe,
   (b) l'utilisation d'api,
   (c) l'utilisation de modèle de similarité pré-entraîné.
3. Classification supervisée qui consiste dans notre cas à attribuer une classe sémantique à chaque séquence en utilisant les classes du référentiel choisi. Chaque fiche du référentiel est considérée comme une classe que le modèle de classification doit pouvoir attribuer à chaque séquence.

16. Répertoire opérationnel des métiers et des emplois

Table 2.2
Normalisation de type référence

Exemple	Forme canonique
contrat à durée indéterminée	CDI
Ferroviaire	Circulation du réseau ferré
Préparer les commandes et exécuter des opérations de réception, de stockage, de tenue des stocks	Magasinage et préparation de commandes

2.3 Conclusion

Pour répondre à la question de recherche (QR1) nous avons choisi l'approche par étiquetage de séquences puisque nous avons fait l'hypothèse que cette approche permet d'améliorer l'extraction d'entités à partir de textes rédigés en langage naturel. En revanche, cette approche nécessite la mise en place d'un corpus annoté de qualité pour l'apprentissage. Par conséquent, nous avons choisi d'intégrer le schéma organisationnel qui a pour avantage de simplifier la compréhension d'un texte et d'aider à considérer le contexte d'une séquence ou d'un ensemble de séquences. De ce fait, nous avons proposé une méthodologie d'étiquetage de séquences qui permet de considérer le schéma organisationnel tout en répondant aux différentes caractéristiques du langage naturel : ambiguïté et évolution du vocabulaire.

La méthodologie DEEP que nous proposons implique trois acteurs principaux ayant des compétences différentes. Les experts sont généralement occupés et ont peu de temps à consacrer à la tâche d'annotation pour la création de corpus d'étiquetage, notre objectif était donc de minimiser le temps nécessaire à chacun. Le fait d'avoir trois acteurs différents permet également d'effectuer les différentes activités en parallèle, ce qui entraîne un gain de temps. Nous avons également proposé d'impliquer des annotateurs qui n'étaient pas familiers avec le domaine afin de faciliter la mise en œuvre du corpus et d'améliorer la vitesse. Ceci étant dit, il est également possible d'avoir une seule personne qui réalise l'ensemble des activités, mais le temps passé sera plus long.

Ensuite, afin de traiter facilement ces informations identifiées et extraites à partir du texte, nous avons proposé deux types de normalisations des séquences : type primitif et type référence. Ces deux normalisations se font de deux façons différentes respectivement par approche par règles et par classification. La limite de l'approche par classification est qu'elle puisse nécessiter une masse de données lorsque le nombre de classes est très grand. De ce fait, il est nécessaire d'évaluer l'uniformité dans la représentativité de toutes les classes dans le corpus d'entraînement.

3

Appariement de deux textes rédigés en langage naturel

Sommaire

L'appariement de deux textes rédigés en langage naturel est l'approche qui consiste à mesurer la distance entre ces deux textes pour créer des paires de textes qui se correspondent sémantiquement. Comme nous l'avons vu dans le chapitre 1 de l'état de l'art, les approches actuelles décrites dans les travaux de la littérature, sont contraintes par leurs structures syntaxiques. En effet, ces approches délaissent la sémantique. Certaines approches sont en mesure de capturer la sémantique, en revanche, celles-ci sont coûteuses en temps ou nécessitent le temps d'un expert rendant ainsi leurs reproductions difficiles. Enfin la dernière limite des travaux de la littérature est l'opacité du système ne permettant pas d'expliquer sur quelles informations s'est basé le score de similarité. L'objectif de nos travaux est donc de (1) pouvoir profiter de la richesse des informations contenues dans un texte afin d'effectuer un appariement appliquée sur les informations les plus utiles d'un texte ; (2) capturer le contexte sémantique du texte ; (3) être en mesure d'expliquer sur quel ensemble d'informations s'est basé le modèle pour attribuer le score de distance sémantique entre deux textes et donc de leur appariement.

De ce fait, notre contribution repose sur l'hypothèse (RQ2-H) que (1) l'appariement des étiquettes et séquences extraites des textes permettrait d'éviter l'opacité du système et d'aider à l'explicabilité des résultats et (2) la représentation vectorielle en utilisant BERT-sentence permettrait d'améliorer la considération de la sémantique pour calculer la distance entre les séquences. Pour rappel, notre question de recherche est donc :

Comment apparier deux textes rédigés en langage naturel ?

Dans ce chapitre, nous allons proposer une approche se basant sur trois modules principaux (voir Figure 3.1). Le premier module utilise l'approche de l'extraction d'information et de normalisation présentée dans le chapitre précédent pour structurer et préparer les textes en identifiant et normalisant les séquences par type d'information. Le deuxième module utilise une représentation vectorielle qui permet de capturer la sémantique quel que soit le domaine d'application. Enfin, nous proposons un dernier module qui utilise cette représentation vectorielle pour calculer la distance entre les séquences de chaque texte.

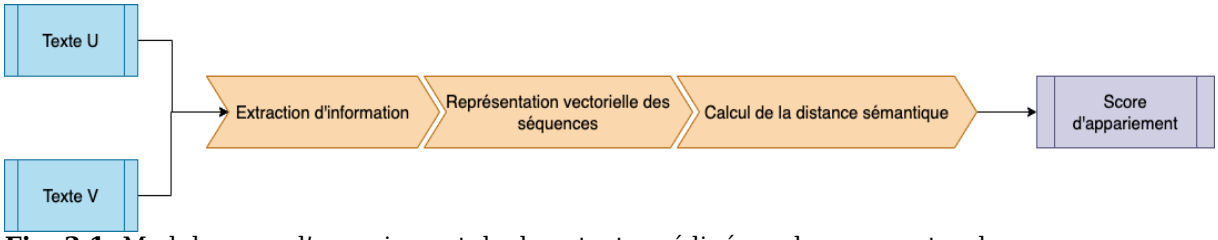


Fig. 3.1. Modules pour l'appariement de deux textes rédigés en langage naturel

3.1 Extraction d'entités par étiquetage de séquences

La revue de la littérature sur l'appariement de deux textes a montré qu'un texte peut contenir plusieurs informations essentielles et non essentielles. Si le texte entier est utilisé, l'appariement peut se baser sur des séquences qui ne sont pas utiles pour la similarité entraînant ainsi un score biaisé. Par exemple, pour la recommandation de films basée sur le genre, si la description entière des films est utilisée, le système d'appariement peut retrouver un vocabulaire commun qui est autre que ce qui caractérise le genre du film. De ce fait, nous proposons d'utiliser l'étiquetage de séquences afin d'identifier, extraire et catégoriser les informations. L'étiquetage de séquences permet de calculer une similarité basée sur un type d'information, palliant ainsi les limites de l'utilisation du texte entier. Cette étape consiste à attribuer à chaque séquence, composant un texte, une étiquette. Pour cela, notre première contribution sur l'extraction d'entités à partir d'un texte rédigé en langage naturel (présentée dans le chapitre précédent) est appliquée.

Soit $L = \{l_w : w \in \{1, ..., E\}\}$ l'ensemble des étiquettes qui représentent les types d'information à identifier et à extraire. L'étiquetage de séquences permet d'associer à chaque étiquette l_w la séquence S_{v,l_w} du texte v comme nous pouvons le voir dans la première colonne de la Figure 3.2.

La seconde limite des approches existantes est l'utilisation d'approches lexicales délaissant ainsi la sémantique des textes. Pour pallier cette limite, nous proposons d'introduire le calcul d'une distance sémantique qui permet de calculer le score de correspondance entre deux séquences tout en capturant la sémantique. Cependant celle-ci nécessite un pré-traitement des séquences. Pour cela, nous appliquons la normalisation de type primitif et de type référence introduite dans notre première contribution.

Cette normalisation permet d'attribuer à chaque séquence S_{v,l_w}, une séquence normalisée NS_{v,l_w} , comme nous pouvons le voir dans la seconde colonne de la Figure 3.2).

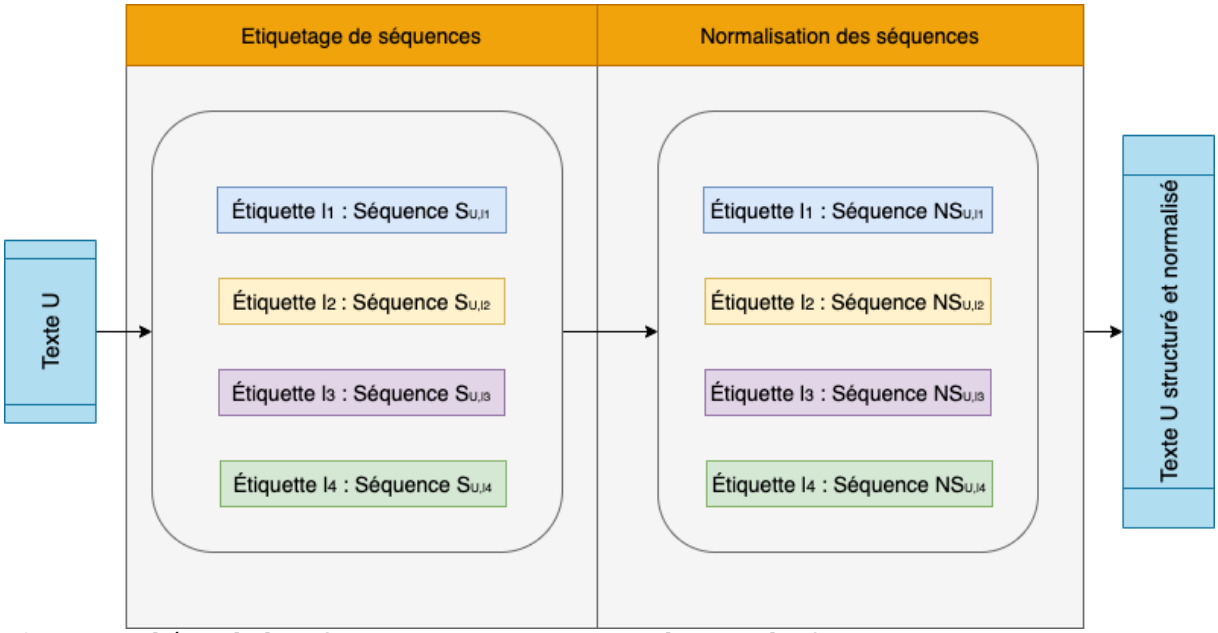


Fig. 3.2. Schéma d'identification, extraction et normalisation d'informations

3.2 Représentation vectorielle des séquences

Le calcul d'une distance entre deux textes nécessite une représentation vectorielle de ces textes. Comme nous l'avons vu dans les travaux de la littérature, plusieurs approches sont possibles. Pour ses limites sur la capture de la sémantique du texte, nous avons mis de côté l'approche lexicale (voir Chapitre 1, Section 1.2.1). Nous nous sommes ainsi intéressés à l'approche sémantique. Parmi les différentes méthodes de représentation vectorielle sémantique, celle qui a permis de représenter le mieux le sens du texte est la méthode par apprentissage profond. Pour ce faire, il existe des modèles pré-entraînés à l'aide d'apprentissage profond sur des corpus de taille très large.

3.2.1 Modèles pré-entraînés

Les travaux de la littérature ont montré que certains modèles existants de représentation vectorielle permettent de représenter sémantiquement les séquences sous forme de vecteurs afin de calculer la distance entre les vecteurs. Pour la construction de ces modèles plusieurs corpus de données de paires de séquences et leurs distances ont été utilisés pour entraîner des modèles d'apprentissage profond. Le premier et le plus largement utilisé est le corpus RG qui a été proposé par Rubenstein et Goodenough en 1965 [126] avec 65 paires de séquences annotées par 51 locuteurs natifs anglais

avec des valeurs de similarité comprises entre 0 et 4. D'autres modèles ont été entraînés plus récemment sur des corpus plus larges, par exemple "Multilingual knowledge distilled" [161], ou encore "Paraphrase-xlm-r-multilingual-v1" [122]. Rappelons que ces modèles ont été entraînés sur des séquences de mots contrairement aux modèles BERT [97], FlauBERT [81], etc. En effet, en étant entraîné sur des séquences, le contexte sémantique des mots (contexte relatif à la signification/au signifié des mots) est mieux capturé puisqu'il considère l'ensemble des mots constituant la séquence. Bien que de nombreux ensembles de données ont été publiés au fil des années comme référence pour les modèles mesurant la similarité sémantique au niveau d'une phrase, très peu sont spécifiques à un vocabulaire pour un domaine particulier. De ce fait, la construction d'un corpus peut être nécessaire dans certains domaines [27]. Dans la suite de ce chapitre, nous proposons une approche permettant d'alimenter un modèle pré-entraîné existant afin de compléter le modèle par un vocabulaire spécialisé.

3.2.2 Choix du modèle pré-entraîné

Chacun Des modèles pré-entraînés existants a ses particularités. Afin de choisir le modèle pré-entraîné qui construira la base de notre modèle de représentation vectorielle, nous avons comparé les modèles pré-entraînés sur des séquences ayant les caractéristiques suivantes :

1. corpus d'entraînement français et multilingue,
2. modèle ayant la meilleure précision sur un vocabulaire d'un domaine spécifique.

Dans la Table 3.1 nous présentons un exemple de modèles pré-entraînés avec leurs caractéristiques. Ces modèles sont entraînés sur des corpus très larges et dans différentes langues. En revanche, ces corpus ne sont pas spécialisés pour des domaines et des vocabulaires en particulier.

Table 3.1
Caractéristiques de certains modèles pré-entraînés

Modèle	Langues	Corpus d'entraînement
distiluse-base-multilingual-cased-v2	16	SNLI (Stanford Natural Language Inference) corpus [19]
stsb-bert-base	+50	Wikipédia, SNLI, etc. [122]
paraphrase-multilingual-MiniLM-L12-v2	+50	Wikipédia [122]

3.2.3 Pré-entraînement sur un vocabulaire spécifique

Rappelons que la particularité des modèles pré-entraînés est l'utilisation de la sémantique des séquences en utilisant la distance sémantique entre des paires de séquences. Afin de pré-entraîner un modèle sur un vocabulaire spécifique, il est né-

cessaire de créer des paires de séquences et leur attribuer un score de similarité sémantique. Par exemple, dans la Table 3.2, nous pouvons voir quatre exemples de paires de séquences et des distances sémantiques attribuées par un expert du domaine du recrutement. Par exemple, la distance sémantique choisie par un expert en recrutement entre la séquence "Cariste" et "Magasinier" est de 1. En revanche, le modèle pré-entraîné "distiluse-base-multilingual-cased-v2" a attribué à ces séquences une distance sémantique de 0.59.

Table 3.2
Paires de séquences et leurs distances sémantiques

Séquence A	**Séquence B**	**Distance sémantique (expert)**	**Distance sémantique (modèle distiluse-base-multilingual-cased-v2)**
Full stack	Back-end et front-end	1	0.2
Cariste	Magasinier	1	0.59
Manager une équipe de 5 personnes	Diriger une équipe	0.9	0.49

Afin de pré-entraîner un modèle de distance sémantique, il est important de suivre les étapes de création d'un modèle définies comme suit :

— création du corpus d'entraînement,

— choix de la fonction de perte,

— évaluation.

Ces trois étapes sont présentées dans la suite de ce chapitre.

3.2.3.1 A. Corpus d'entraînement

Cette étape consiste à construire un corpus ou à utiliser un corpus d'entraînement existant. Elle consiste en :

1. la sélection des séquences. Cette étape consiste à choisir les séquences sur lesquelles nous souhaitons effectuer l'entraînement. Par exemple, [27] a choisi 52 séquences dans le domaine de l'informatique et notamment les compétences liées au métier de l'informatique.
2. l'annotation de la valeur de similarité entre les séquences. Cette étape est la plus importante puisqu'elle nécessite d'attribuer à chaque séquence choisie dans l'étape précédente, une séquence et la distance sémantique entre cette paire de séquences. Des méthodes manuelles existent dans la littérature [27]. Néanmoins, ces méthodes peuvent nécessiter des ressources humaines et du temps.

Pour éviter une méthode manuelle, nous proposons une approche qui exploite les synonymes, antonymes et définitions des séquences choisies à l'étape précédente. Différentes sources de données pour la langue française peuvent être utilisées (voir Table 3.3). Les ressources permettant de définir et caractériser une séquence peuvent être utilisées pour construire les paires de séquences et leurs distances sémantiques (comme présenté en exemples dans la Table 3.2). Les ressources permettant de trouver les antonymes peuvent aider à la construction de paires sémantiquement opposées. Dans la Figure 3.3 nous pouvons voir un exemple de création de paires de séquences et leurs distances sémantiques en utilisant le référentiel de compétences ESCO. En effet, ce référentiel contient une hiérarchie de compétences par catégories. Huit catégories d'aptitudes sont disponibles et chacune contient une liste de sous aptitudes. Cette hiérarchie permet de créer de paires de séquences de compétences, avec une distance sémantique de 1 pour les compétences appartenant à la même catégorie. Inversement, les compétences n'appartenant pas à la même catégorie se voient attribuer une distance sémantique nulle. Dans l'exemple de la Figure 3.3, nous avons attribué automatiquement un score de 1 aux paires de séquences "Manipuler et déplacer" et "Trier et emballer des marchandises et des matériaux". Nous avons attribué un score de 0 aux paires de séquences "Laver et entretenir" et "Programmer des systèmes informatiques". En effet, ces deux séquences n'appartiennent pas à la même catégorie d'aptitudes. Sur un exemple basé sur une catégorie de métier issue d'un référentiel de métier, l'attribution automatique en suivant le modèle de création automatique de paires de séquences a généré 125 paires positives et 340 paires négatives.

Cette approche de création de corpus pour les paires de séquences et leurs distances sémantiques permet un gain de temps par rapport à une annotation manuelle. Une évaluation de ces paires de séquences et leur distance peut être faite par un expert pour améliorer la qualité du corpus qui sera utilisé dans la suite pour entraîner un modèle de similarité sémantique entre séquences.

3.2.3.2 B. Choix de la fonction de perte (Loss function et évaluation du modèle)

La fonction de perte, ou Loss function [62], est une fonction qui évalue l'écart entre les prédictions réalisées par le modèle d'apprentissage et les valeurs réelles des observations utilisées pendant l'apprentissage. Plusieurs fonctions de perte peuvent être utilisées par exemple, la distance euclidienne ou présentés dans l'état de l'art.

3.2.3.3 C. Méthode d'évaluation

L'évaluation du modèle d'apprentissage est basée sur le calcul du coefficient de Pearson et Spearman qui sont deux coefficients très utilisés pour évaluer la corrélation entre deux variables [9]. Le coefficient de Pearson est un indice reflétant une relation linéaire entre deux variables continues. Le coefficient de Spearman quant à lui reflète

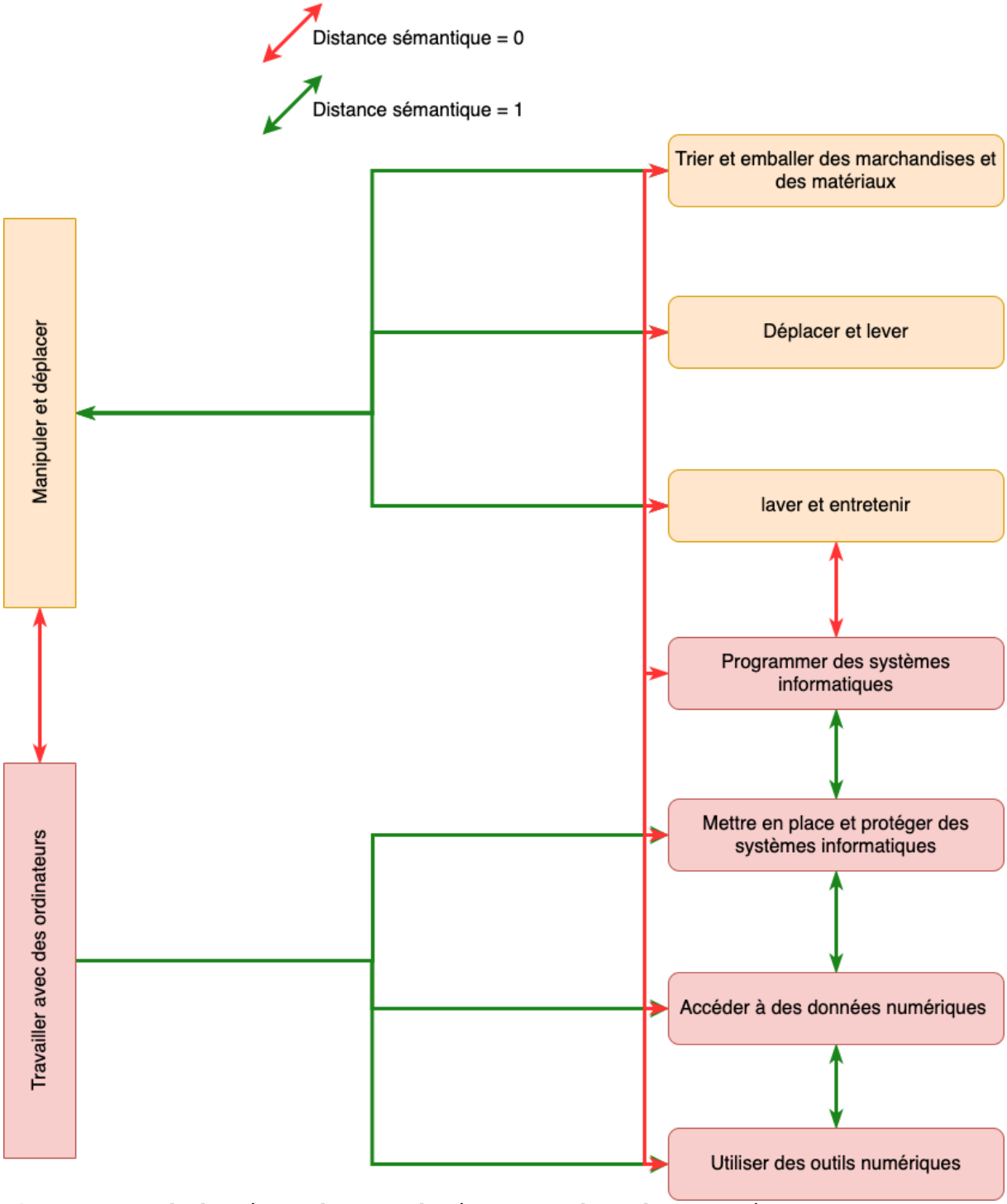


Fig. 3.3. Exemple de création de paires de séquences et leurs distances sémantiques

Table 3.3
Descriptions de quelques référentiels existants

Type	Source	Nom	Contenu	Langues
Thesaurus	Non spécifique	WordsAPI[1]	Définitions, synonymes, rhymes	Anglais
Thesaurus	Non spécifique	WikiSynonyms[2]	Synonymes, définitions	Français et anglais
Base de données lexicale	Non spécifique	Wordnet[3]	Synonymes, définitions	Anglais
Base de données lexicale	Non spécifique	WOLF[4]	Synonymes, Antonymes, définitions	Français
Ontologies	Recrutement	ESCO[5]	Synonymes, définitions	Français et anglais
Référentiels	Recrutement	ROME	Synonymes, définitions	Français et anglais

[1] https://www.wordsapi.com/
[2] https://github.com/ipeirotis/WikiSynonyms
[3] https://wordnet.princeton.edu/
[4] http://pauillac.inria.fr/~sagot/index.html#wolf
[5] https://ec.europa.eu/esco/portal/occupation

la relation monotone entre deux variables continues ou ordinales. Dans une relation monotone, les variables ont tendance à changer ensemble, mais pas toujours à une vitesse constante. Le coefficient de corrélation de Spearman est basé sur les valeurs classées pour chaque variable plutôt que sur les données brutes [9]. Le coefficient de corrélation varie entre -1 et +1 :

— La valeur 0 reflète une relation nulle entre les deux variables,

— la valeur négative signifie que lorsqu'une des variables augmente, l'autre diminue,

— la valeur positive signifie que les deux variables varient ensemble dans le même sens.

Ce coefficient est utilisé pour évaluer la corrélation entre deux séquences d'après le modèle entraîné sur un corpus d'entraînement en comparaison au corpus gold (corpus de test). Cette corrélation permet ainsi d'évaluer le modèle d'apprentissage. Comme nous pouvons le voir dans la Figure 3.4, le modèle entraîné de représentation vectoriel est appliqué aux séquences du corpus de test. La distance Cosinus est ensuite appliquée sur ces deux vecteurs. Ce score représente la distance sémantique entre les deux vecteurs et est comparé à la vraie distance (distance attribuée automatiquement et validée par l'expert) entre ces séquences en utilisant la corrélation de Pearson et Spearman.

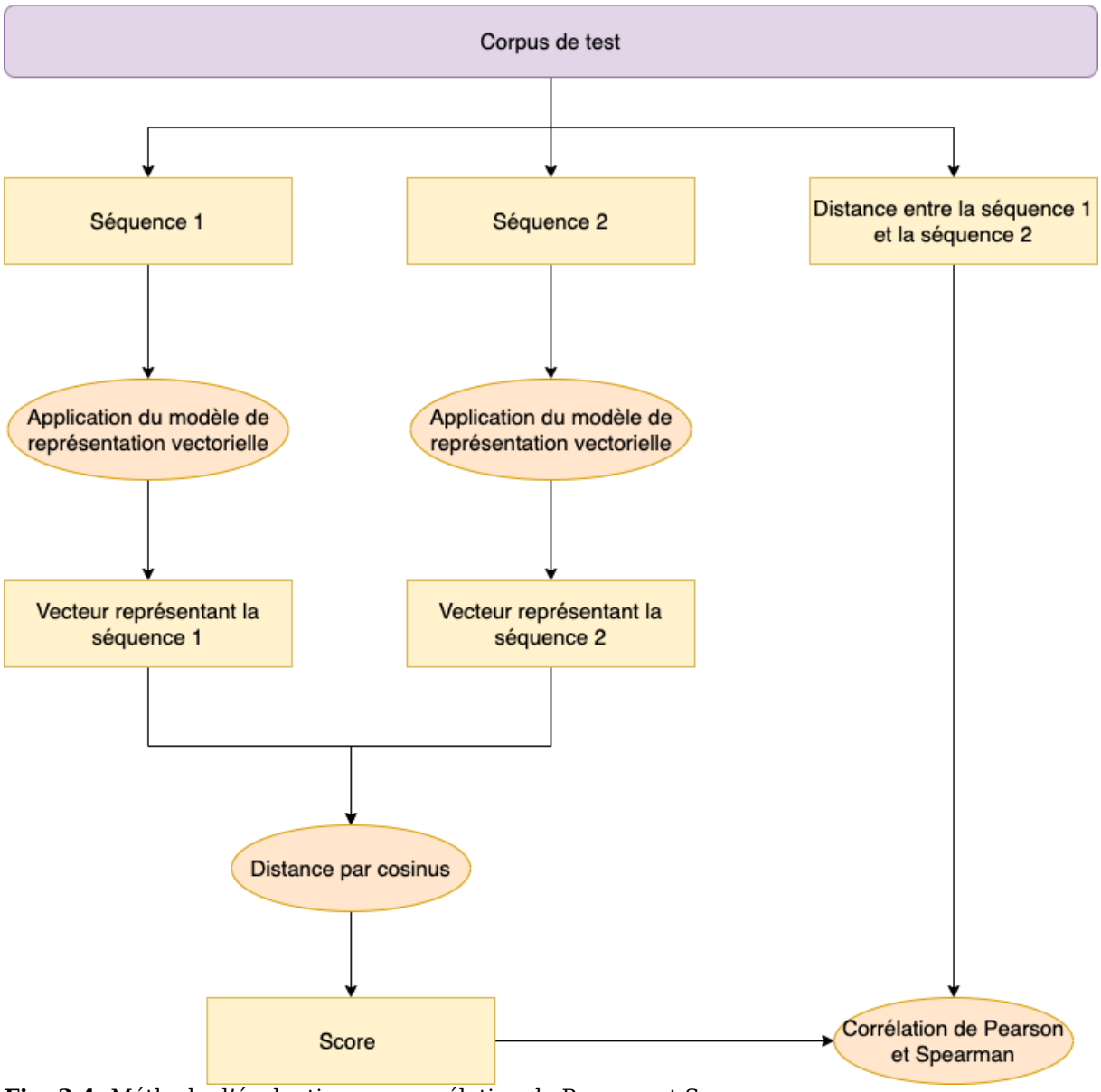


Fig. 3.4. Méthode d'évaluation par corrélation de Pearson et Spearman

3.3 Calcul de la distance sémantique

Le calcul de la distance sémantique entre deux textes pour l'appariement se fait en deux étapes :

1. Distance sémantique par paires de séquences. Le modèle entraîné dans l'étape précédente permet donc de calculer la distance sémantique entre deux séquences comme nous pouvons le voir dans la Figure 3.5. L'algorithme est décrit dans 1.
2. Distance finale par moyenne de la distance par paires de séquences comme nous pouvons le voir dans la Figure 3.5. L'avantage de cette approche de similarité par étiquette/séquences est d'attribuer un poids à chaque type d'information si nous souhaitons baser notre similarité sur un type d'information plus qu'un autre. Par exemple, si un recruteur souhaite comparer un CV et une offre d'emploi et qu'il souhaite mettre plus de poids sur le nombre d'années d'expérience, cette approche de similarité par séquence permet justement de laisser ce choix au recruteur. Les poids sont donc une donnée d'entrée au module d'appariement qui, si elle n'est pas ajoutée, alors par défaut le poids est distribué de façon égale pour toutes les étiquettes. Définissons alors l'ensemble W contenant p_{l_w} le poids attribué à l'étiquette l_w. L'algorithme est décrit dans 2.

Dans la Figure 3.5, nous présentons les étapes pour l'approche d'appariement proposée dans ce chapitre. Celle-ci suit les étapes expliquées précédemment :

— Extraction d'information à partir des textes en utilisant l'étiquetage de séquence. Ces séquences sont normalisées afin d'attribuer une catégorie sémantique à chaque séquence.

— Représentation vectorielle se basant sur un modèle pré-entraîné sur des données du domaine en utilisant des référentiels existants.

— Distance entre les vecteurs de chaque séquence, permettant d'avoir un score de correspondance par type d'information

— Distance finale correspondant à la moyenne de tous les scores de distance. L'avantage de cette approche est la possibilité d'attribuer un poids par type d'information, si nous souhaitons attribuer plus d'importance à une information en particulier.

3.4 Conclusion

Les travaux de la littérature ont des limites telles que l'utilisation d'une représentation vectorielle lexicale ou l'utilisation de modèle de similarité opaque délaissant l'explicabilité des résultats. Puisqu'un texte est composé de différentes séquences, nous avons fait l'hypothèse que la similarité par type de séquences pouvait améliorer la qualité de l'appariement tout en permettant une transparence pour l'explication des résultats. Notre contribution sur l'extraction d'entités et la normalisation des séquences nous a permis d'associer à chaque texte un ensemble d'étiquette/séquence et de normaliser ces séquences. La distance entre deux textes est transformée en distance entre chaque séquence associée à chaque étiquette pour chaque texte. Notre

Algorithm 1 $computeSim(U, V)$: Mesure de la distance entre chaque séquence de U and V

$S \leftarrow \emptyset$ // Initialiser l'ensemble qui contiendra les scores de similarité pour chaque étiquette
for each l_w in L **do**
  $\vec{S_{U,l_w}} \leftarrow SBertVec(S_{U,l_w})$ // Appliquer la représentation vectorielle pour les séquences du texte U pour chaque étiquette
  $\vec{S_{V,l_w}} \leftarrow SBertVec(S_{V,l_w})$ // Appliquer la représentation vectorielle pour les séquences du texte V pour chaque étiquette
  $sim_{score}(\vec{S_{U,l_w}}, \vec{S_{V,l_w}}) \leftarrow \frac{\vec{S_{U,l_w}}.\vec{S_{V,l_w}}}{||\vec{S_{U,l_w}}||.||\vec{S_{V,l_w}}||}$ // Appliquer la distance cosinus pour chaque étiquette entre les vecteurs de U et V
  $score_{S_{U,l_w},S_{V,l_w}} \leftarrow sim_{score}(\vec{S_{U,l}}, \vec{S_{V,l_w}})$ // Assigner à $score_{S_{U,l_w},S_{V,l_w}}$ le résultat de la distance
  $S \leftarrow S \cup \{score_{S_{U,l_w},S_{V,l_w}}\}$ // Ajouter à l'ensemble S le score d'appariemment entre les séquences de chaque étiquette U et V
end for

return S // Retourner l'ensemble S

Algorithm 2 $finalSim(U, V, W)$: Mesure de la distance entre les textes U et V

$S \leftarrow computeSim(U, V)$ // Ensemble des scores d'appariemment pour chaque étiquette résultant de l'algorithme $computeSim(U, V)$
$sc \leftarrow 0$ // Initialiser la valeur qui contiendra la somme des scores de toutes les étiquettes
$pc \leftarrow 0$ // Initialiser la somme totale du poids de toutes les étiquettes
for each i in S, p in W **do**
  $sc \leftarrow sc + p * i$ // Ajouter à sc le score total final pour chaque étiquette et le poids qui lui est attribué
  $pc \leftarrow pc + p$ // Sommer les poids
end for
$final_{U,V} \leftarrow \frac{sc}{len(L)-1+pc}$
$sc \leftarrow sc + p * i$ // Faire la moyenne des scores obtenus pour toutes les étiquettes

return $final_{U,V}$ // Retourner le score final de l'appariement

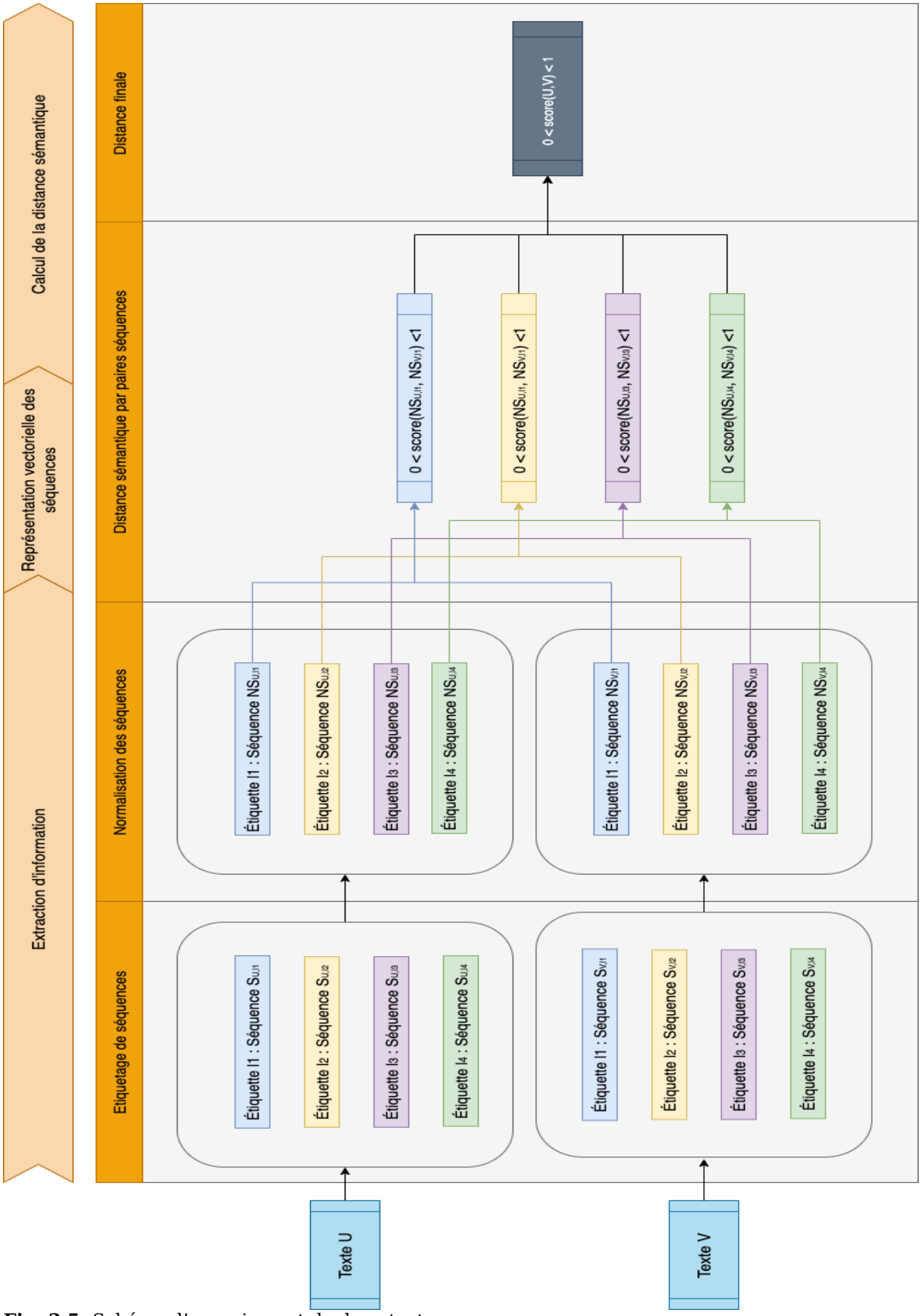


Fig. 3.5. Schéma d'appariement de deux textes

travail s'est aussi basé sur l'entraînement d'un modèle de similarité sémantique en ayant comme base des modèles déjà pré-entraînés utilisant le modèle de langage BERT. Ces modèles pré-entraînés le sont sur des données non spécifiques à des domaines, entraînant une similarité moins qualitative. De ce fait, nous avons proposé une approche pour construire automatiquement des paires de séquences similaires et non similaires pour compléter ce modèle d'entraînement. Cette approche se base sur l'utilisation de référentiels existants sur le domaine applicatif souhaité. Une des limites de cette approche est la représentation vectorielle basée sur des séquences construites automatiquement qui à notre sens nécessite une validation préalable par un expert. En effet, les référentiels utilisés peuvent parfois contenir des erreurs qui peuvent entraîner des distances sémantiques fausses.

4

ADE^2 : un système d'Aide à la Décision dans un Environnement Évolutif

Sommaire

Dans ce chapitre notre objectif de recherche est de répondre à la question de recherche suivante :

QR3
Comment concevoir un système d'aide à la décision qui s'adapte à un environnement incertain, évolutif et répondant à des objectifs multiples dont les paramètres sont variables ?

Cette question de recherche concerne les verrous suivants que nous avons définis auparavant :

1. (QR3-C1) L'environnement est incertain ;
2. (QR3-C2) L'environnement est évolutif entraînant l'évolution dans le temps des variables du système et donc un problème non stationnaire.
3. (QR3-C3) Les objectifs du décideur à atteindre sont multiples ;
4. (QR3-C4) Le contexte de décision est rédigé en langage naturel ;

Pour répondre aux verrous définis ci-dessus, nous proposons ADE^2 un système d'aide à la décision qui a les caractéristiques suivantes : (1) Le décideur peut être un expert du domaine ou non ;(2) La décision du décideur est le choix d'un ensemble d'items pour un contexte de décision donné ; (3) Le décideur souhaite atteindre un ensemble d'objectifs en choisissant les items ; (4) Le contexte de décision est caractérisé par un contenu textuel rédigé en langage naturel (document textuel) ; (5) Les paramètres évoluent dans le temps et dans un environnement incertain. L'objectif de ce système d'aide à la décision est de recommander à un décideur des items qui évoluent dans un environnement incertain générant des paramètres temporels dépendant d'acteurs externes inconnus. Le décideur a un ou plusieurs contextes de décision rédigés en langage naturel, des objectifs et des contraintes.

Le système proposé dans ce chapitre est composé de différents modules :

— (1) Module de traitement de l'information
— (2) Module d'apprentissage
— (3) Module de filtrage
— (4) Module d'optimisation
— (5) Module d'apprentissage par renforcement

Ces modules ont pour objectif d'utiliser les données de la base de données (BDD) issus d'évènements provenant des items afin d'aider le décideur à choisir un ensemble d'items étant donné un contexte de décision et des objectifs à atteindre. La Figure 4.1 décrit ce processus d'aide à la décision. La prise de décision dépend du contexte de décision textuel et des estimations des fonctions objectifs dont les valeurs des paramètres sont les conséquences du choix des paires contexte/item. Dans le diagramme d'activité Figure 4.1, nous pouvons visualiser les composants principaux d'un SAD : Le décideur, la base de données, les modèles de connaissance. Ces modèles sont construits grâce aux modules cités ci-dessus et que nous allons décrire dans la suite de ce chapitre.

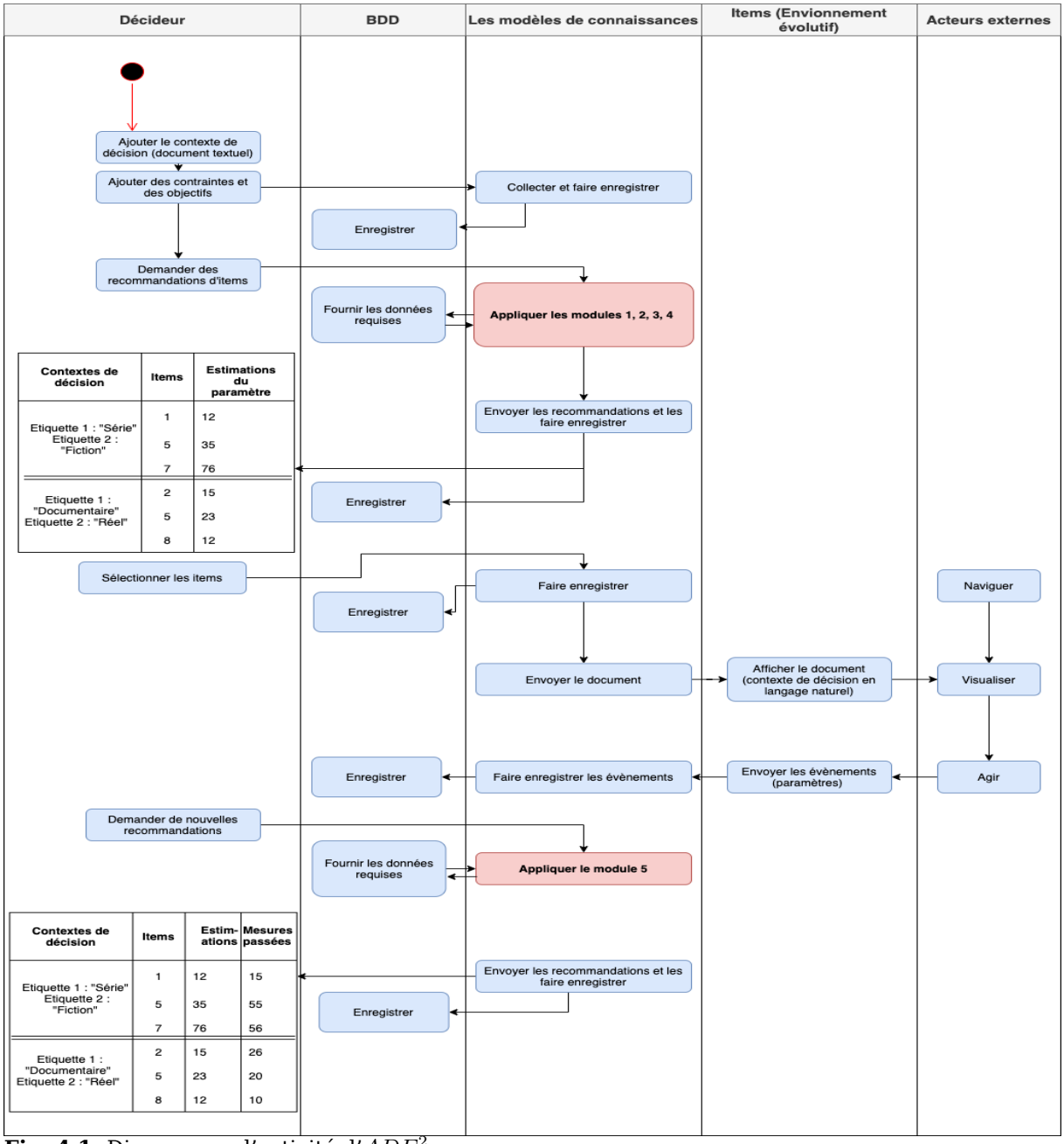


Fig. 4.1. Diagramme d'activité d'ADE^2

4.1 Représentation et formulation du problème

4.1.1 Items

Pour rappel, un item est le terme général désignant ce qui est recommandé pour le décideur afin de l'aider à faire un choix. Les items peuvent être de diverses natures (films, vidéos, restaurants, lieux d'activité, etc.). Pour la suite, nous définissons l'ensemble des items comme suit : Q=$\{q_j : j \in \{1, ..., U\}\}$.

Note : dans notre étude de cas sur l'optimisation du e-recrutement, un item sera un canal.

4.1.2 Objectifs

Les objectifs que le décideur souhaite atteindre peuvent être multiples. Dans la suite du document, nous appelons $\{f_o : o \in \{1, ..., H\}\}$ l'ensemble des objectifs du décideur.

Note : dans notre étude de cas sur l'optimisation du e-recrutement, un objectif sera la maximisation du nombre de conversions.

4.1.3 Contexte de décision

Pour rappel, un contexte de décision, est le contexte dans lequel sera pris la décision du décideur. Dans la suite du document, nous nous intéressons aux contextes de décision rédigés en langage naturel représenté à travers un document textuel. Nous définissons cet ensemble comme suit : $\widetilde{D} = \{D_i : i \in \{1, ..., N\}\}$

Note : dans notre étude de cas sur l'optimisation du e-recrutement, un contexte de décision correspond à une offre d'emploi.

4.1.4 Variables de décision

Nous définissons l'ensemble des variables de décision comme suit : $Z = \{z_{ij} \in \{0, 1\}; i \in \{1, ..., N\}, j \in \{1, ..., U\}\}$, avec $z_{ij} = 1$ si pour le contexte de décision i, l'item j est recommandé au décideur ; 0 sinon.

Note : dans notre étude de cas sur l'optimisation du e-recrutement, $z_{ij} = 1$ si pour une offre d'emploi, le canal j est recommandé au décideur ; 0 sinon.

4.1.5 Paramètres

Les paramètres sont les constantes associées aux contraintes et à la fonction objectif. Une des caractéristiques de notre problème est que les paramètres évoluent dans le temps.

Par exemple, à chaque item j à un instant t dans le contexte i est associé le paramètre h_{D_i,q_j}. Ce paramètre varie dans le temps et les données récoltées sont des données passées. Un exemple est présenté dans la Table 4.1. $h_{21}(t)$ est la série chronologique du contexte 2 pour l'item 1.

Note : dans notre étude de cas sur l'optimisation du e-recrutement, $h_{D_i,q_j}(t)$ sera le nombre de clicks obtenus pour l'offre d'emploi D_i sur le canal q_j à l'instant t.

4.1.6 Contraintes

Nous définissons deux types de contraintes :

Table 4.1
Exemple de séries chronologiques du paramètre h pour les contextes 1 et 2 et l'item 1

$\mathbf{t}$	$\mathbf{h_{11}(t)}$	$\mathbf{h_{21}(t)}$
10/04/2021	18	-
11/04/2021	06	-
12/04/2021	45	-
13/04/2021	42	-
14/04/2021	32	-
15/04/2021	35	-
16/04/2021	-	15
17/04/2021	-	35
18/04/2021	-	2

— c_{D_i,q_j} : constante de contraintes liées à l'item q_j pour le contexte de décision D_i

— B_{D_i} : constante de contraintes liées au contexte de décision D_i

Note : dans notre étude de cas sur l'optimisation du e-recrutement, c_{D_i,q_j} correspond au coût d'un click pour l'offre d'emploi D_i sur la canal q_j. B_{D_i} sera le budget alloué au recrutement pour l'offre D_i.

4.1.7 Problème d'optimisation

L'objectif d'ADE^2 est d'aider le décideur à affecter à chacun de ses contextes de décision rédigés en langage naturel un ou plusieurs items : $1 \leq \sum_{j=1}^{U} z_{ij}$. Le problème d'optimisation qui en découle consiste à minimiser ou maximiser un ensemble de fonctions (décrivant les objectifs du décideur) sollicitant des paramètres évoluant dans le temps.

Note : dans notre étude de cas sur l'optimisation du e-recrutement, le problème consiste à maximiser le nombre de conversions pour une offre d'emploi sur un canal sous contrainte d'un budget donné et en fonction d'un historique de conversions passées.

4.1.8 Analogie avec des cas réels d'aide à la décision

Afin de permettre une meilleure compréhension des paramètres et variables définis ci-dessus, nous présentons une analogie avec des cas concrets d'aide à la décision dans le monde réel.

4.1.8.1 Publicité

— **Items** : Sites de publicité en ligne.

— **Objectif du décideur** : maximiser la visibilité de son produit.

— **Contexte de décision** : La description du produit possiblement décrit sur une fiche de description.

— **Variables de décision** : $z_{ij} = 1$ si le spot de publicité du produit i (contexte de décision) est déposé (décision) sur le site de publicité j (item).

— **Paramètres** : Nombre de vues à la période t (période passée).

— **Contraintes** : Financières.

— **Données** : Historique de données sur les diffusions des spots publicitaires sur les médias.

4.1.8.2 Bourse

— **Items** : Entreprises en bourse.

— **Objectif du décideur** : maximiser le retour sur investissement[17].

— **Contexte de décision** : Les articles d'actualités.

— **Variables de décision** : $z_{ij} = 1$ s'il y a achat de l'action (décision) de l'entreprise j (item) étant donné les actualités i (contexte de décision).

— **Paramètres** : Les bénéfices à la période t (période passée).

— **Contraintes** : Financières.

— **Données** : Historique des fluctuation du cours des actions par entreprise.

4.1.8.3 E-recrutement

— **Items** : Canaux de diffusion.

— **Objectif du décideur** : maximiser le nombre de CV reçus.

— **Contexte de décision** : Offre d'emploi.

— **Variables de décision** : $z_{ij} = 1$ si l'offre d'emploi i (contexte de décision) est déposée (décision) sur le canal de diffusion j (item).

— **Paramètres** : Nombre de CV à la période t (période passée).

— **Contraintes** : Financières.

— **Données** : Historique de données sur les diffusions des offres d'emploi sur les canaux.

4.2 Module de traitement de l'information

Les données à traiter pour la conception d'ADE^2 sont de nature hétérogènes :

— le contexte de décision est textuel,

— les paramètres évoluent au cours du temps,

— les contraintes de décision sont numériques.

17. Le retour sur investissement est un indicateur qui permet de comparer des investissements en considérant l'argent investi et l'argent gagné (ou perdu)

Afin de construire un modèle de décision, il est important d'unifier et traiter ces données. Le premier module de notre modèle de décision consiste à traiter les différentes informations récoltées. Ce module a pour objectif de transformer les différentes sources d'informations pour rendre les données plus facilement traitables dans la suite des modules. Un de nos verrous est le contexte qui est textuel et rédigé en langage naturel. Il est donc difficile de comprendre et interpréter ce contexte si les informations de ce texte ne sont pas identifiées et extraites. De ce fait, les modules d'extraction d'entités et de normalisation présentés dans le Chapitre 2 sont utilisés dans ce module afin de caractériser le contexte de décision qui est présent dans le texte concerné.

4.2.1 Extraction d'entités et normalisation des données textuelles

Ce module a pour objectif de répondre au verrou suivant :

QR3-C4
Le contexte de décision est rédigé en langage naturel

Pour rappel, le module d'extraction d'entités présenté dans le chapitre 2 a pour objectif d'identifier et extraire les informations importantes à partir d'un texte rédigé en langage naturel.
Rappelons que :

1. D_i est le contexte textuel rédigé,
2. le module d'extraction d'entités permet de :
   (a) définir les informations importantes à identifier que nous appelons "étiquettes". Notons $L = \{l_w : w \in \{1, ..., N\}$ l'ensemble de ces "étiquettes", valable pour tous les contextes,
   (b) associer à chaque étiquette l_w pour le contexte D_i l'ensemble des séquences $\varphi_{D_i,l_w} = \{S_{D_i,l_w}, D_i \in \widetilde{D}, l_w \in L\}$, $card(\varphi_{D_i,l_w}) = r_{l_w,D_i}$ avec $r_{D_i,l_w} \in \mathbb{N}$ est le nombre de séquences associées à l'étiquette l_w pour le contexte de décision textuel D_i avec $D_i \in \widetilde{D}$.
3. le module de normalisation permet :
   (a) de normaliser chaque séquence pour lui attribuer une séquence normalisée ou catégorie sémantique. Nous définissons alors $N\varphi_{D_i,l_w}$ l'ensemble des séquences normalisées associées à l'étiquette l_w pour le contexte textuel D_i, avec $D_i \in \widetilde{D}, l_w \in L$
   . Rappelons que la normalisation a pour objectif de faciliter la lecture et le traitement des séquences en leur attribuant une catégorie sémantique.

Ces deux modules permettent donc de faciliter l'interprétation et l'utilisation des informations contenues dans le contexte de décision et pouvoir les utiliser dans les modèles. Ce module permet donc à partir d'un contexte rédigé dans un document textuel d'attribuer un contexte normalisé :

$$n_{D_i} = \{\{l_w, \varphi_{D_i,l_w}\}, l_w \in L, D_i \in \widetilde{D}\}$$

Le contexte normalisé est composé des paires étiquettes et séquences associées pour le contexte textuel D_i.

$NO = \{n_{D_i}, D_i \in L\}$ est l'ensemble de ces contextes normalisés.

4.2.2 Uniformisation des données

Lorsque les données sont récoltées de différentes sources d'informations, les formats, les types de ces dernières sont différents. Par exemple, dans la Table 4.2 nous pouvons visualiser les statistiques pour une publicité diffusée sur le site publicitaire A et dans la Table 4.3 les statistiques pour la même publicité diffusée sur le site publicitaire B. Les noms des données, le format et le type sont différents. De ce fait, il est important de créer un format de table de données unique dans lequel toutes ces données seront exportées. Les données sont donc transformées et unifiées pour générer une table de données facilement interprétable pour la suite du modèle de décision (Table 4.4). Nous supposons dans cet exemple que les étiquettes du contexte publicitaire "Abonnez-vous pour 15 euros par mois et recevez 2 produits ménagers pour une valeur minimale de 10 euros chacun", sont

• le type de souscription • le prix • la catégorie (pour désigner la catégorie du produit) • Valeurs (pour désigner les valeurs du produit).

À ces étiquettes sont associées respectivement les séquences suivantes : • {Abonnez-vous, par mois} • 15 euros • produits ménagers • 10 euros chacun.

À ces séquences sont associées respectivement les séquences normalisées suivantes : • Abonnement • 15 € • Ménages • 10 €

En conclusion, ces deux premières étapes de traitement permettent donc de :

1. remplacer le contexte textuel en catégories d'informations (auxquelles sont associées des séquences sémantiques),
2. uniformiser le nom des paramètres,
3. créer une table de données pour les différentes sources d'informations.

4.2.3 Création des classes sources

Une fois les données uniformisées l'objectif de l'étape de "création des classes sources" est de préparer les données pour la prévision des valeurs des paramètres puisque le décideur base sa décision sur ces paramètres. Pour cela les travaux de la littérature ont montré que pour gagner en qualité dans les prévisions d'une série chronologique il est important que la taille des données soient importantes. Étant donné que les contextes de décision peuvent être multiples, si nous créons un modèle de prévision pour chaque contexte, la qualité du modèle peut être de moindre qualité que si nous regroupons les contextes sémantiquement proche.

Table 4.2
Exemple d'évènements sur le site publicitaire A

Contenu publicitaire	Date	Clics	Impr.[1]	Conv. [2]	Répét. [3]	CTR[4]	Budget dépensé
"Abonnez-vous pour 15 euros par mois et recevez 2 produits ménagers pour une valeur minimale de 10 euros chacun"	11/04/21	12	102	0	1.06	0.11	3.6 €
"Abonnez-vous pour 15 euros par mois et recevez 2 produits ménagers pour une valeur minimale de 10 euros chacun"	12/04/21	21	94	2	1.02	0.22	6.3 €

[1] L'impression publicitaire correspond à un affichage de l'élément publicitaire
[2] Le taux de conversions est le rapport entre les individus qui ont réalisé l'action finalement recherchée dans le cadre de la campagne publicitaire et le nombre total d'individus touchés par la campagne
[3] La répétition est l'estimation du nombre de fois que chaque individu a vu la publicité
[4] Taux de clics (Click Through Rate) est le rapport entre le nombre de clics et le nombre d'impressions de la publicité

Table 4.3
Exemple d'évènements sur le site publicitaire B

Contenu publicitaire	Date	Clics	Impr.	Conver.	Répét.	CTR	Budget
"Abonnez-vous pour 15 euros par mois et recevez 2 produits ménagers pour une valeur minimale de 10 euros chacun"	2021-04-11	16	160	0	1.3	0.1	3.2
"Abonnez-vous pour 15 euros par mois et recevez 2 produits ménagers pour une valeur minimale de 10 euros chacun"	2021-04-12	12	190	0	1.02	0.06	2.4

De ce fait, nous proposons de pallier cette limite en créant des classes sémantiques pour représenter les données. Ces classes sémantiques permettent :

1. de considérer la sémantique des contextes
2. de pallier le manque de données,
3. de créer des classes qui faciliteront la considération de nouveaux contextes.

Nous appelons ces classes dans la suite du chapitre les classes sources normalisées :

Table 4.4
Table de données en sortie du module de traitement de l'information

Type de souscription	Prix (€)	Catég.	Val. (€)	Date	Site	Action	Val. de l'action	Budget dépensé
Abonnement	15	Ménages	10	11/04/2021	A	clic	12	3.6
Abonnement	15	Ménages	10	11/04/2021	B	clic	16	3.2
Abonnement	15	Ménages	10	12/04/2021	A	conv	2	6.3
Abonnement	15	Ménages	10	12/04/2021	A	clic	21	6.3
Abonnement	15	Ménages	10	12/04/2021	B	clic	12	2.4

— $S = \{CLS_d : d \in \{1, ..., e\}\}$ l'ensemble des classes sources ;

— n_{CLS_d} est le contexte normalisé de la classe source CLS_d avec $CLS_d \in S$;

— $h_{CLS_d, q_j(t)}$ est l'historique du paramètre h (représentant une série chronologique) évoluant dans le temps par le passé pour le contexte normalisé de la classe source d et pour l'item q_j.

Par exemple, dans le domaine de la publicité, supposons que nous avons les deux contenus publicitaires suivants :

1. P1. "Abonnez-vous pour 15 euros par mois et recevez 2 produits ménagers pour une valeur minimale de 10 euros chacun"
2. P2. "Pour un abonnement mensuel de 15 euros, vous pouvez recevoir 2 produits ménagers pour une valeur minimale de 10 euros chacun"

Ces deux publicités sont considérées comme deux contextes de décision différents : $P1$ et $P2$. Ils sont lexicalement différents, mais sémantiquement très proches. Nous supposons donc que ces deux publicités peuvent appartenir à une même classe source puisque leur contexte normalisé sont similaires.

4.2.3.1 Approche

L'approche de création des classes sources est composée de plusieurs étapes :

1. **Regroupement des contextes similaires.** Cette étape regroupe les contextes normalisés similaires n_{D_i}. Les contextes normalisés ayant les mêmes séquences normalisées sont regroupés dans une même "classe source" CLS_d. Par exemple, la séquence "contrat à durée indéterminée" a comme séquence normalisée "CDI" et "cdi" a comme séquence normalisée "CDI" ; Leurs séquences normalisées étant similaires, ces deux contextes (certes élémentaires) font partie de la même "classe source".
2. **Agrégation des paramètres évoluant dans le temps** est une étape qui agrège l'historique des données des paramètres pour les contextes normalisés appartement à la même "classe source". Nous proposons un algorithme qui calcule l'ensemble des séries chronologiques pour les classes sources (contenant les contextes normalisés similaires).

L'algorithme 3 que nous proposons suit les étapes suivantes :

(i) création des classes sources en regroupant les contextes normalisés similaires. La similarité se calcule en comparant les séquences normalisées de chaque étiquette associée. Si ces séquences sont identiques alors les deux contextes normalisés font partie de la même classe source.

(ii) agrégation des paramètres qui sont des données temporelles. Pour ce faire, supposons que nous avons deux contextes de décision 1 et 2 qui ont respectivement le contexte normalisé n_1 et n_2. Supposons que n_1 et n_2 appartiennent à la même classe source CLS_1. Dans ce cas, nous associons à la classe source CLS_1 la série chronologique qui agrège la série chronologique $h_{11}(t)$ du paramètre h, du contexte de décision 1 et de l'item 1 et la série chronologique $h_{21}(t)$ du paramètre h, du contexte de décision 2 et de l'item 1. Ces deux séries chronologiques sont représentées dans la table 4.1. Le résultat de l'agrégation est affiché dans la table 4.5. Si les deux séries chronologiques se chevauchent nous avons fait le choix d'appliquer la fonction moyenne des séries. *Note : L'agrégation consiste à construire une matrice de série chronologique pour les différents contextes normalisés qui composent la classe source.*

Une classe source est donc la classe qui regroupe les contextes similaires et agrège les séries chronologiques de chacun de ces contextes normalisés.

Table 4.5
Résultat de la série chronologique agrégée à partir des séries h_{11} et h_{21} appartenant à la nouvelle classe source

t	$\mathbf{h_{11}(t)}$	$\mathbf{h_{21}(t)}$	**Série chronologique agrégée pour la nouvelle classe source**
10/04/2021	18	-	19
11/04/2021	6	-	6
12/04/2021	45	-	45
13/04/2021	42	-	42
14/04/2021	32	-	32
15/04/2021	35	-	35
16/04/2021	-	15	15
17/04/2021	-	35	35
18/04/2021	-	2	2

Algorithm 3 Group-Normalized(Q)

$S \leftarrow \emptyset$ // Ensemble de classes sources vide
$H \leftarrow \emptyset$ // Ensemble de séries chronologiques vide
Initialisation de l'ensemble des classes sources et des séries chronologiques associées
$CLS_1 \leftarrow n_{D_1}$ // Attribuer à la première classe source, le premier contexte normalisé de l'ensemble NO
$S \leftarrow S \cup CLS_1$ // Ajouter CLS_1 à l'ensemble S
while $j \in \{1, .., U\}$ **do**
  $h_{CLS_1,q_j} \leftarrow h_{n_{D_1},q_j}$ // Attribuer à CLS_1 , la série chronologique du contexte normalisé n_{D_1} pour chaque item q_j
  $H \leftarrow H \cup h_{CLS_1,q_j}$ // Ajouter la série chronologique de la première classe source dans l'ensemble H
end while
Parcourir tous les contextes normalisés restant pour la création des classes sources
for each $i \in \{2, .., N\}$ **do**
  $exist \leftarrow 0$
  Cas où la classe normalisée est similaire à une classe source existante
  for each $CLS_d \in S$ **do**
    if $n_{D_i} == CLS_d$ **then**
      $exist \leftarrow 1$ // Attribuer une valeur de 1 à exist pour garder en mémoire que le contexte normalisé n_{D_i} lui est déjà associée une classe source
      while $j \in \{1, .., U\}$ **do**
        $h_{CLS_d,q_j} \leftarrow agreg(h_{n_{D_i},q_j}, h_{CLS_d,q_j})$ // Agréger la série chronologique du contexte similaire à la classe source
        $H \leftarrow H \cup h_{CLS_d,q_j}$ // Ajouter la nouvelle série chronologique de la classe source CLS_d dans l'ensemble H
      end while
    end if
  end for
  Cas où la classe normalisée n'est similaire à aucune classe source existante
  if $exist == 0$ **then**
    $id \leftarrow len(S) + 1$ // Si exist est égal à 0 c'est que le contexte de décision ne s'est vu attribuer aucune classe source existant dans l'étape précédente. Ceci entraîne la nécessité de créer une nouvelle classe source adaptée. Pour cela, nous récupérons la longueur de l'ensemble S
    $CLS_{id} \leftarrow n_{D_i}$ // Créer une nouvelle classe source qui est égale au contexte normalisé qui n'est similaire à aucune autre classe source existante dans l'ensemble S
    $S \leftarrow S \cup CLS_{id}$ // Ajouter la nouvelle classe source dans l'ensemble S
    while $j \in \{1, .., U\}$ **do**
      $h_{CLS_{id},q_j} \leftarrow h_{n_{D_i},q_j}$
      $H \leftarrow H \cup h_{CLS_{id},q_j}$ // Ajouter la série chronologique associée à la classe source CLS_{id} dans l'ensemble H
    end while
  end if
end for

return S,H // Retourner l'ensemble S des classes sources et l'ensemble H des séries chronologiques pour chaque classe sources et pour tous les items

4.2.4 Préparation des classes cibles

Une classe cible est la classe pour laquelle le décideur souhaite se voir recommander un ensemble d'items. Au contexte normalisé n_v est associé la classe cible représentée par un contexte rédigé en langage naturel D_v. Cette classe cible se voit attribuer pour laquelle le modèle d'aide à la décision va devoir attribuer une série chronologique prédite à partir des classes sources existantes Pour cela, un calcul de similarité utilisant l'algorithme 1 est appliqué entre cette classe cible et chacune des classes sources. La classe cible se voit attribuée la série chronologique de la classe source la plus similaire. L'algorithme utilisé pour préparer la classe cible est décrit dans l'algorithme 4.

Algorithm 4 Create-cible-classe(S, n_v)

$maxScore \leftarrow 0$ // Initialiser maxScore à 0
for each $CLS_d \in S$ **do**
  if $maxScore < computeSim(CLS_d, n_v)$ **then**
    $maxScore \leftarrow computeSim(CLS_d, n_v)$ // Si le score de similarité entre la classe source CLS_d et la classe cible n_v est supérieur à maxScore alors on attribue à maxScore cette nouvelle valeur
    $class \leftarrow CLS_d$ // On attribue à la variable $class$ la classe source qui est la plus similaire
  end if
end for

return class // On retourne la classe source la plus similaire à notre classe cible

4.3 Module d'apprentissage

4.3.1 Approche

Le module d'apprentissage a pour objectif de faire une prévision des séries chronologiques. Étant donné l'environnement incertain (QR3-C1) et les variables évoluant dans le temps (QR3-C2) les travaux de la littérature nous ont permis de considérer les modèles non linéaires et notamment le modèle d'apprentissage CNN-LSTM. Celui-ci étant le modèle permettant le plus de considérer le contexte des séries temporelles (voir Section 1.3.5).

Nous avons donc choisi de considérer ce modèle dans ce module. Le module d'apprentissage suit les étapes suivantes :

(a) Préparer le corpus d'apprentissage. Cette étape consiste à transformer la série chronologique en données d'entrée pour appliquer une prévision des données futures. La première étape de la préparation du corpus est tout d'abord l'encodage des données catégoriques. Pour cela, il existe plusieurs techniques qui peuvent être utilisées. La plus utilisée dans la littérature [80, 69] est l'encodage avec une valeur comprise entre 0 et le nombre de

classe-1. La seconde étape est la transformation des données stockées sous forme d'évènements par date en données d'apprentissage supervisé. Pour cela, chaque ligne de la table de donnée représente les k derniers évènements. Dans la Figure 4.2, nous pouvons voir l'exemple où les lignes de la table de départ sont transformées en variables d'entrée du module d'apprentissage en suivant la chronologie. Le modèle CNN choisi permet de prendre en compte le contexte de décision pour améliorer la qualité de la prévision et le LSTM permet de considérer les séries chronologiques représentées pour chaque donnée d'entrée du modèle.

Item	Contexte	Date	Parm_Des
0	1	04/07/2019	69
0	1	05/07/2019	74
0	1	06/07/2019	72
0	1	07/07/2019	66
0	1	08/07/2019	70
0	1	09/07/2019	75
0	1	10/07/2019	14

Item(t-1)	Contexte(t-1)	Parm_Des(t-1)	Item(t)	Contexte(t)	Parm_Des(t)	Item(t+1)	Contexte(t+1)	Parm_Des(t+1)
0	1	69	0	1	74	0	1	72
0	1	74	0	1	72	0	1	66
0	1	72	0	1	66	0	1	70

Fig. 4.2. Création du corpus de données pour la prévision des paramètres de décision afin d'appliquer l'algorithme CNN-LSTM

Le modèle de données préparé dans le module 1 est utilisé pour transformer cette table en données d'entrées pour le module d'apprentissage. Dans la Figure 4.2 nous pouvons visualiser cette transformation et le résultat.

(b) Appliquer l'algorithme d'apprentissage. Cette étape nécessite d'abord la mise en place de l'algorithme d'apprentissage qui suit l'architecture suivante (présentée aussi dans la Figure 4.3) :

— En entrée nous avons les données préparées dans l'étape précédente, composées des contextes normalisés des classes sources et des séries chronologiques associées à chacune d'elle. Comme nous l'avons vu dans l'état de l'art, l'intégration de variables qui potentiellement peuvent influencer les variations des paramètres de décision est importante lorsque l'environnement est incertain. De ce fait, l'entrée de notre algorithme d'apprentissage est une série multivariée temporelle.

— La première couche de notre architecture est une couche convolutive. Elle est utilisée pour extraire les caractéristiques des données d'entrée et notamment des variables du contexte normalisé. Une première étape de cette couche consiste à réaliser un filtrage par convolution : le principe

est de faire "glisser" une fenêtre représentant la caractéristique (feature) sur les entrées, et ensuite de calculer le produit de convolution entre la caractéristique et chaque portion de l'entrée balayée. Une caractéristique est alors vue comme un filtre. En d'autres termes, nous voulons construire un modèle capable de prédire les paramètres de décision dans n'importe quel contexte normalisé pour toutes les classes sources, étant donné les données historiques récoltées dans le passé. Une couche de pooling (mise en commun) est intégrée pour découper la série de rectangles (comme nous pouvons le voir dans la Figure 4.3) en plus petit rectangle. Le pooling réduit la taille spatiale des données d'entrées, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Des noyaux de taille [longueur de la série chronologique * le filtre] passent en entrée de la couche convolutive. Ainsi, lorsque chaque filtre glisse sur les données d'entrée, il produit un tableau 1D de longueur q.

— Chaque unité de ce tableau est utilisée comme entrée d'un réseau de neurones récurrent long terme. Afin de maximiser la capture de tendance sous-jacente dans les données, le LSTM est appliqué.

— Après plusieurs couches de neurones, le raisonnement dans le réseau neuronal se fait à travers des couches entièrement connectées. Les neurones sont connectées à toutes les sorties de la couche précédente. Leurs fonctions d'activation sont calculées avec une multiplication matricielle suivie d'un décalage de polarisation [154]. Ainsi, la prédiction des valeurs futures pour le paramètre de chaque classe source sur chaque item est produite comme sortie.

Il était possible pour nous d'appliquer uniquement un réseau de neurones récurrent pour chaque classe source indépendamment, comme dans les travaux [115, 30]. En revanche, nous aurions perdu toute potentielle corrélation entre ces classes sources. De plus, cette approche est coûteuse en espace mémoire et en temps de réponse. Nous avons donc fait l'hypothèse que la mise en place d'un modèle de prédiction de paramètres intégrant toutes les classes sources sera de meilleure qualité que plusieurs modèles indépendants pour chaque classe source.

La mesure d'évaluation du modèle utilisée très souvent dans la littérature est l'erreur quadratique moyenne (MSE). [83]. Elle se calcule comme suit :

Définition 7 L'erreur quadratique moyenne d'un estimateur $\hat{Y}$ d'un paramètre Y de dimension n est une mesure caractérisant la « précision » de cet estimateur calculée comme suit :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

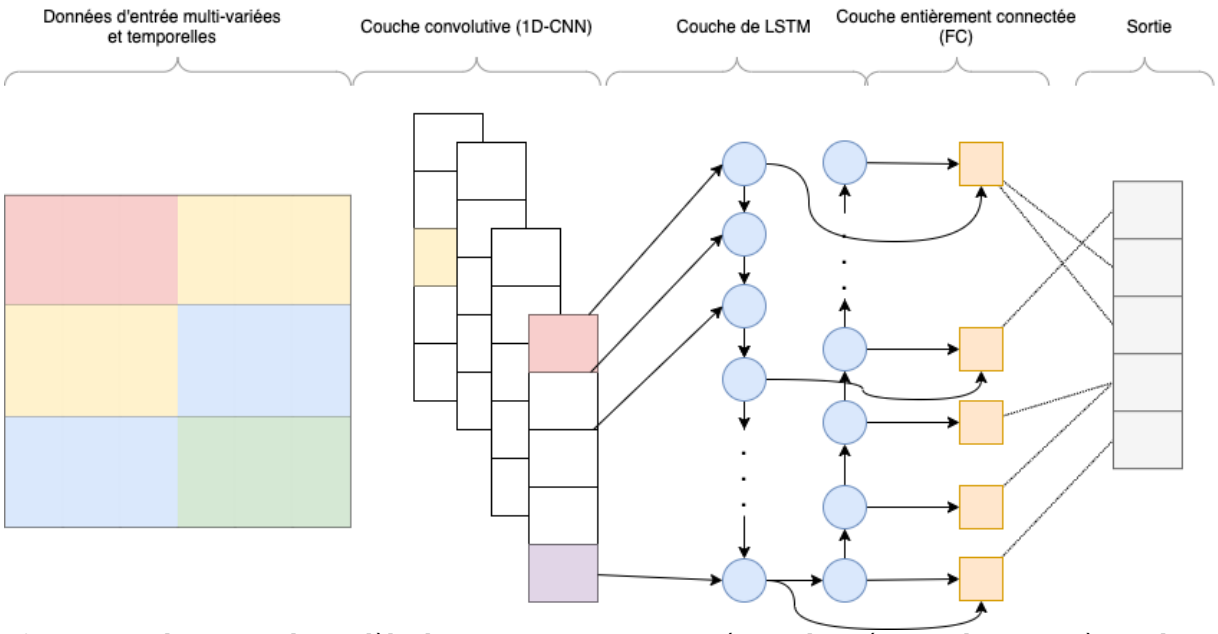


Fig. 4.3. Architecture du modèle d'apprentissage supervisé pour la prévision des paramètres dans le futur

4.3.2 Application du modèle

Étant donné que les paramètres évoluent dans le temps et que l'affectation des contextes de décision à des items dépend de ces paramètres, l'objectif de ce modèle de prévision est de prédire les valeurs futures du paramètre h_{D_i,q_j} pour le contexte D_i et l'item q_j pour les périodes futures. Définissons :

— $T' = \{\hat{T}_T : T \in \{O, ..., g\}\}$, où $\hat{T}_O$ correspond à la période courante (période de 7 jours) $\{\hat{T}_T : T \in \{1, ..., g\}\}$ correspond aux périodes futures.

— $T'' = \{\hat{T} : \hat{T} \in \{-r, ..., -1\}\}$ est l'ensemble des périodes passées.

Le module d'apprentissage pour la classe source CLS_d dont le contexte normalisé est n_{CLS_d} et la série chronologique $h_{n_{CLS_d},q_j}(t)$ pour l'item j retourne :

$$\gamma_{CLS_d,q_j}(t) = \{\hat{\theta}_{CLS_d,q_j}(t) : CLS_d \in S, q_j \in Q, t \in T'\}$$

qui est la série chronologique du paramètre h_{ij} pour la classe source CLS_d et l'item q_j pour les périodes futures appartenant à l'ensemble T'.

Les paramètres prédits permettent de filtrer les canaux qui ne répondront pas aux objectifs du décideur. De ce fait, cet ensemble sera utilisé dans les modules suivants qui ont pour objectif de filtrer et d'appliquer le modèle d'optimisation pour recommander les items répondant au besoin du décideur.

4.4 Module de filtrage

Certains items ont des caractéristiques spécifiques. Par exemple, dans le domaine de la publicité les sites sont spécifiques à certaines catégories d'articles de

publicité. De plus, les décideurs ont des préférences et peuvent donc personnaliser les items en imposant un filtre sur le type d'item. Par conséquent, certaines paires ne sont pas possibles : $z_{ij} = 0$. De ce fait, nous considérons les z_{ij} uniquement pour les items qui sont adéquats au contexte de décision et en fonction des préférences du décideur. Cela permet de réduire considérablement l'espace de recherche des solutions optimales pour répondre aux objectifs des décideurs.

4.4.1 Résolution du problème par filtrage

Afin d'appliquer un filtre sur l'ensemble des items permettant de réduire l'ensemble des solutions optimales d'affectation de contexte de décision à un ensemble d'items pour le décideur, nous créons :

(a) une matrice A_{D_i,l_m} pour modéliser les caractéristiques du contexte de décision D_i qui sont représentées par les séquences normalisées pour chaque étiquette appartenant à l'ensemble $N\varphi_{D_i,l_w}$ avec $L = \{l_w : w \in \{1, ..., E\}\}$ l'ensemble de ces "étiquettes". La dimension de la matrice est de $N * E$.

$$A_{D_i,l_m} = \begin{pmatrix} N\varphi_{D_1,l_1} & N\varphi_{D_1,l_2} & \cdots & N\varphi_{D_1,l_w} & \cdots & N\varphi_{D_1,l_E} \\ N\varphi_{D_2,l_1} & N\varphi_{D_2,l_2} & \cdots & N\varphi_{D_2,l_w} & \cdots & N\varphi_{D_2,l_E} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N\varphi_{D_i,l_1} & N\varphi_{D_i,l_2} & \cdots & N\varphi_{D_i,l_w} & \cdots & N\varphi_{D_i,l_E} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N\varphi_{D_N,l_1} & N\varphi_{D_N,l_2} & \cdots & N\varphi_{D_N,l_w} & \cdots & N\varphi_{D_N,l_E} \end{pmatrix}$$

(b) une matrice I_{q_j,l_m} pour représenter les caractéristiques des items, avec ch_{q_j,l_m} la séquence normalisée associée à la caractéristique de l'item q_j pour l'étiquette l_m avec $q_j \in Q$ et $l_m \in L$. La dimension de la matrice est de $U * N$.

$$I_{q_j,l_m} = \begin{pmatrix} ch_{q_1,l_1} & ch_{q_1,l_2} & \cdots & ch_{q_1,l_w} & \cdots & ch_{q_1,l_N} \\ ch_{q_2,l_1} & ch_{q_2,l_2} & \cdots & ch_{q_2,l_w} & \cdots & ch_{q_2,l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ch_{q_j,l_1} & ch_{q_j,l_2} & \cdots & ch_{q_j,l_w} & \cdots & ch_{q_j,l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ch_{q_U,l_1} & ch_{q_U,l_2} & \cdots & ch_{q_U,l_w} & \cdots & ch_{q_U,l_N} \end{pmatrix}$$

Avant l'application du filtrage, nous avons z_{ij} la variable de décision dont les items q_i appartiennent à l'ensemble des items Q. Étant donné l'application du filtrage pour éliminer les items dont les caractéristiques sont différentes du contexte de décision, l'ensemble des items va réduire. Nous appelons cet ensemble $K = \{q_j : j \in \{1, ..., M\}\}$. La création de cet ensemble est présentée dans l'algorithme 5 et se base sur le calcul de similarité entre chaque élément des matrices A_{D_i,l_m} et I_{q_j,l_m}. Si le score de similarité est inférieur à un seuil de 0.7 alors les items sont éliminés. Ce seuil est choisi afin d'éliminer les items qui ne correspondent pas à au moins 70% du contexte de décision (ce seuil dépend des besoins. Il n'existe pas de seuil conseillé dans la littérature).

Algorithm 5 Filter-channels(D_v,L,Q)

$Q_f \leftarrow \emptyset$ // Initialisation de l'ensemble des items possibles
for each q_j in Q **do**
  for each l_m in L **do**
    $sc_{q_j,l_m} \leftarrow computeSim(ch_{q_j,l_m}, N\varphi_{D_v,l_m})$ // [18]
  end for
  $meanSC_{q_j} = \frac{1}{E}\sum_{w=1}^{E} sc_{q_j,l_w}$
  if $0.7 < meanSC_{q_j}$ **then**
    $Q_f \leftarrow q_j$ // [19]
  end if
end for

return Q_f

A la sortie de ce module nous définissons la nouvelle variable de décision $X = \{x_{ij} : i \in \{1, ..., N\}, j :\in K\}$ avec $K = \{q_j : j \in \{1, ..., M\}\}$

4.5 Module d'optimisation

Ce module d'optimisation a pour objectif de sélectionner l'ensemble final des items qui répond aux objectifs du décideur pour la classe cible (qui est, pour rappel, la classe qui est associée au nouveau contexte de décision pour lequel le décideur souhaite se voir attribuer des recommandations).

Soient :

- D_v le nouveau contexte de décision pour lequel nous souhaitons aider le décideur à prendre une décision.
- n_{D_v} le contexte normalisé de D_v.
- $CLS_{n_{D_v}}$ est la classe source la plus similaire retournée par l'algorithme 4.
- x_{D_v,q_j} avec $j \in \{1, ..., M\}$ et $1 \leq \sum_{j=1}^{M} x_{D_V,q_j}$, est la variable de décision.
- $\gamma_{CLS_{n_{D_v}},q_j}$ est la série chronologique attribuée à la classe cible pour la suite du module pour l'item $j \in K$ qui est l'ensemble des items possibles construits dans le module de filtrage.
- B_{D_v} est la constante de contraintes liées au contexte de décision D_v.
- c_{D_v,q_j} est la constante de contraintes liées au contexte de décision q_j pour le contexte de décision D_v.

Il existe trois méthodes d'optimisation pour l'aide à la décision. Les méthodes de résolution de problèmes d'optimisation multi-objectifs peuvent être répartis en trois familles [70] :

(a) Les méthodes d'optimisation **a priori**. Cette méthode consiste à considérer que le compromis que le décideur souhaite faire entre les différents objectifs est connu par avance ;

(b) Les méthodes d'optimisation **progressives**. Dans ce cas, le décideur intervient dans le processus de recherche de solutions en répondant à différentes questions. Cette méthode prend en compte les préférences du décideur, mais nécessite sa disponibilité pendant toute la phase de recherche de solution optimale ;

(c) Les méthodes d'optimisation **a posteriori**. Dans ce cas, le système cherche à fournir au décideur un ensemble de bonnes solutions réparties correctement. Il peut ensuite au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Cette méthode ne nécessite donc pas la disponibilité du décideur ou de ses préférences.

Le système d'aide à la décision que nous proposons fait partie de la troisième famille de méthodes. Dans ce type de méthodes, il existe deux étapes importantes à suivre [70] :

(a) la phase de recherche des solutions Pareto optimales. Nous allons présenter cette phase dans la suite de ce chapitre,

(b) la phase de choix parmi ces solutions. L'expert retient une solution parmi les solutions Pareto optimales.

Nous proposons de traiter plusieurs problèmes d'optimisation. Pour cela, dans la suite du document nous allons présenter trois modèles d'optimisation :

(a) optimisation mono-période et mono-objectif,

(b) optimisation multi-périodes et mono-objectif,

(c) optimisation mono-période et multi-objectif.

4.5.1 Problème mono-objectif mono-période

4.5.1.1 Formulation

Dans cette section, nous proposons un modèle mono-objectif qui prend en compte les différentes spécificités des contextes et des items pour le paramètre h. Ce premier problème consiste à ne considérer que les données passées de la classe source la plus similaire à notre classe cible qui sont des valeurs exactes pour se rapprocher des données réelles de ce paramètre.

Pour cela, nous introduisons un lissage exponentiel qui est utilisé dans les travaux sur les séries temporelles [89] pour considérer les données passées. La considération de ces dernières permet :

— d'assurer la convergence du modèle,

— d'éviter l'élimination d'un item parmi la liste à recommander au décideur dont le paramètre est faible pour la période courante T=0 et qui a été finalement influencé par un facteur inconnu (crise économique, etc.)

Le principe du lissage exponentiel consiste à donner plus d'importance aux dernières observations d'une série chronologique comme suit :

$$\hat{y}(t) = \alpha y(t) + \alpha(1-\alpha)y(t-1) + \alpha(1-\alpha)^2 y(t-2)...$$

$$0 < \alpha < 1$$

La valeur de α est choisie selon l'objectif souhaité :

— $\alpha = 0$, la prévision est égale à la dernière observation,

— α proche de 0, prévision souple. La prévision est fortement influencée par les observations les plus récentes

— α est proche de 1, prévision rigide. L'influence des observations passées est d'autant plus importante et remonte loin dans le passé

— $\alpha = 1$ les observations passées ont la même importance sur toutes les périodes passées.

Dans les travaux de la littérature, $\alpha \in]0, 1[$ afin d'exclure ces deux cas extrêmes. Nous définissons donc une nouvelle valeur qui permet d'appliquer le lissage exponentiel sur les séries chronologiques de la classe source CLS_d la plus similaire à notre classe cible dont le contexte de décision est D_v.

:

$$\Theta_{D_v,q_j} = \alpha h_{CLS_d,q_j}(t) + \alpha(1-\alpha)h_{CLS_d,q_j}(t-1) + \alpha(1-\alpha)^2 h_{CLS_d,q_j,j}(t-2)...$$

Rappelons que le contexte de décision a une contrainte liée à chaque item. De ce fait le problème d'optimisation est formulé comme suit :

$$\begin{cases} \max f(x)_{D_v} = \sum_{j=1}^{M} \Theta_{D_v,q_j} * x_{D_v,q_j} \\ s.c \\ \sum_{j=1}^{M} c_{D_v,q_j} * x_{D_v,q_j} \leq B_{D_v} \end{cases}$$

4.5.1.2 Résolution

Le système d'aide à la décision proposé fait partie de la famille d'optimisation **a posteriori** entraînant la génération des solutions Pareto. De ce fait, les méthodes à base de populations travaillant avec un ensemble de solutions potentielles, tels que les algorithmes évolutionnaires, semblent adapter pour ce problème.

Un algorithme génétique se base au départ sur une population de solutions candidates appelées individus qui va évoluer de génération en génération jusqu'à trouver celle qui contient les meilleures solutions. Chaque individu peut être sujet à des transformations génétiques (mutation, croisement par exemple). Chaque individu est ensuite évalué suivant l'objectif défini et cette valeur d'aptitude (fitness value) est un critère pour sa survie d'une génération à une autre. Un algorithme génétique suit les étapes suivantes représentées aussi dans la Figure 4.5 [163] :

— La **définition de la population** consiste à définir la taille de la population initiale, N, la probabilité de croisement p_c et la probabilité de mutation p_m. Généralement, la population initiale est générée aléatoirement de manière à répartir les individus uniformément sur l'espace de recherche.

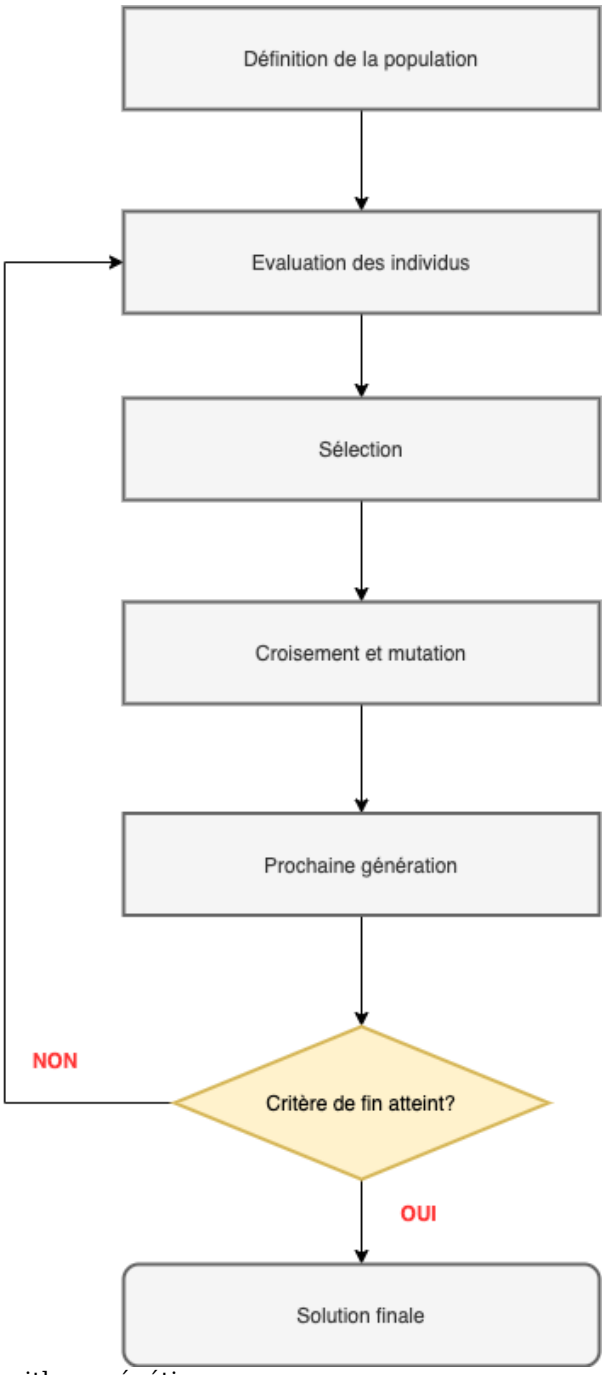


Fig. 4.4. Étapes d'un algorithme génétique

— L'objectif d'un algorithme génétique est d'optimiser une fonction donnée dans un espace de recherche précis. L'étape d'**évaluation** consiste à définir la fonction objectif du problème pour mesurer la performance des individus. Un algorithme génétique a besoin d'une fonction qui permette d'évaluer l'adaptation d'un individu, ce qui offre la possibilité de comparer des individus. Cette fonction est construite à partir du critère que l'on désire optimiser. L'application de cette fonction à un élément de la population donne sa valeur d'aptitude (fitness).

— La **sélection** consiste à définir un nombre d'individus que nous souhaitons avoir par génération. Une fois l'évaluation de la génération réalisée, une sélection est appliquée à partir de la fonction d'évaluation. Seuls les individus passant l'épreuve de sélection peuvent accéder à la génération intermédiaire et s'y reproduire.

— Le **croisement** ou "crossover" est un opérateur de recombinaison qui fournit un couple d'enfants à partir d'un couple de parents de la génération précédente. Il s'agit de l'heuristique prépondérante de l'exploration d'un espace de recherche par un algorithme génétique [12]. Une mutation est ensuite effectuée. Celle-ci joue le rôle de bruit et empêche l'évolution de se figer en introduisant constamment de nouveaux gènes dans la population [47]. Cet opérateur permet d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre de gènes mutés [12]. La **mutation** est appliquée avec une probabilité p_m fixée initialement. Pour chaque chromosome de la population, on tire au hasard un nombre entre 0 et 1. Si le nombre est inférieur à la probabilité de mutation p_c, le chromosome est muté sinon il est recopié dans la nouvelle population.

— Prochaine génération. Cette étape consiste à considérer les contraintes dans l'espace de recherche en évaluant si les individus sélectionnés sont admissibles en répondant aux contraintes définies au départ ou non.

— Pour avoir notre **solution finale**, nous choisissons notre critère de fin de cycle. Ce qui permet d'avoir la génération finale dans laquelle se trouve notre solution finale. Nous définissons comme critère :

  — Le nombre d'itérations qui marquera la fin du cycle.

  — Le temps de calcul CPU. Une limite de temps d'exécution est définie afin de permettre de rapidement proposer au décideur des solutions répondant à son objectif.

  — Stagnation de la valeur du fitness. En effet, après un certain nombre d'itérations, on évalue notre fonction objectif. Si celle-ci ne s'améliore pas, on arrête le cycle. Sinon, on continue avec le nombre d'itérations déjà défini.

  Le résultat présenté au décideur sera l'ensemble des items pour lesquels $x_{D_v,q_j} = 1$ avec la valeur de la fonction objectif associée.

4.5.2 Problème mono-objectif et multi-périodes

4.5.2.1 Formulation

Le problème multi-périodes consiste à utiliser les prévisions du paramètre h (résultat du modèle d'apprentissage supervisé) afin d'anticiper l'évolution dans le temps de ce paramètre. Étant donné que ce paramètre peut évoluer d'une période à une autre, nous définissons une nouvelle variable de décision dépendant de la période $x_{D_v,q_j,\hat{T}} \in \{0,1\}$, avec $x_{D_v,q_j,\hat{T}} = 1$ si l'item q_j est affecté au contexte de décision D_v à la période $\hat{T}$; 0 sinon avec :
$1 \leq \sum_{j=1}^{M} x_{D_v,q_j,\hat{T}}$ et $X_{\hat{T}} = \{x_{D_v,q_j,\hat{T}} : q_j \in Q, \hat{T} \in T'\}$
La fonction objectif est définie comme suit :

$$maxf(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{\Theta}_{d_j}(\hat{T}) * x_{D_v,q_j,\hat{T}}$$

Le problème multi-périodes est ainsi formulé :

$$\begin{cases} maxf(x) = \sum_{\hat{T}=1}^{g} \sum_{j=1}^{M} \hat{\Theta}_{D_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ s.c \\ \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} c_{D_v,q_j} * x_{D_v,q_j\hat{T}} \leq B_{D_v} \end{cases}$$

4.5.2.2 Résolution

La résolution de ce problème d'optimisation se fait de la même façon que pour le problème mono-période mono-objectif étant donné qu'un seul objectif est considéré.

4.5.3 Problème multi-objectifs et multi-périodes

4.5.3.1 Formulation

Le décideur peut vouloir optimiser plusieurs objectifs. Nous définissons pour cela, $F = \{f_o : o \in \{1,...,H\}\}$ l'ensemble des fonctions objectifs. Chacune de ces fonctions objectifs ont un paramètre qui évolue dans le temps (dans les sections précédentes il s'agissait de h). Pour cela, nous définissons $\tilde{P} = \{p_o : o \in \{1,...,H\}\}$ l'ensemble des paramètres liés à chacune des fonctions objectifs. Soit $\hat{p_o}_{d_v,q_j}(t) : j \in K, t \in T'$ la prédiction du paramètre p_o pour le contexte de décision d_v à l'instant $t \in T'$ pour l'item q_j. Notre variable de décision dans ce problème reste la même que précédemment : $x_{D_v,q_j,\hat{T}} \in \{0,1\}$, avec $x_{D_v,q_j,\hat{T}} = 1$ si l'item q_j est affecté au contexte de décision D_v à la période $\hat{T}$;

$$\begin{cases} \max f_1(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p1}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ \max f_2(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p2}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ .... \\ \max f_o(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{p_o}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ .... \\ \max f_H(x) = \sum_{\hat{T}=0}^{g} \sum_{j=1}^{M} \hat{pH}_{d_v,q_j}(\hat{T}) * x_{D_v,q_j,\hat{T}} \\ s.c \\ \sum_{\hat{T}=1}^{g} \sum_{j=0}^{M} c_{D_v,q_j} * x_{D_v,q_j\hat{T}} \leq B_{D_v} \end{cases}$$

4.5.3.2 Résolution

La résolution de ces problèmes repose sur l'identification des solutions optimales, non dominées ou front de Pareto [114]. Le front de Pareto est l'ensemble de tous les points Pareto optimaux : si partant d'un point de cet espace de solutions, la valeur de chaque fonction objectif ne peut être améliorée sans 'altérer' au moins une des autres valeurs [90]. Les solutions résultants appelées front de Pareto seront alors classées en "Pareto optimale". En raison de son efficacité reconnue [90, 165], nous avons porté notre choix sur l'implémentation de l'algorithme génétique NSGA-II qui est représenté dans la Figure 4.5. Cet algorithme est basé sur une classification des individus en plusieurs niveaux :

(a) il utilise une approche élitiste permettant de sauvegarder les meilleures solutions trouvées lors des générations précédentes afin de préserver la diversité,

(b) il ne nécessite aucun réglage de paramètres,

(c) il utilise un opérateur de comparaison qui est basé sur un calcul de la distance de *crowding*.

L'algorithme choisi, suit le schéma classique d'un algorithme génétique avec une sélection et un tri différent [38]. Dans le NSGA-II, les individus sont d'abord sélectionnés de manière frontale. Ce faisant, il y aura une situation où un front devra être divisé parce que tous les individus ne sont pas autorisés à survivre et ne respectent pas les contraintes. Dans ce front divisé, les solutions sont sélectionnées en fonction de la distance de crowding. Celle-ci peut être calculée de différentes façons, en utilisant par exemple la distance euclidienne [45]. Elle permet de mesurer l'inertie qui est la distance habituellement retenue en classification hiérarchique. Le point faible de cette distance est le poids important des points se trouvant à une grande distance de l'origine de la mesure. Cette distance n'isole pas ces points davantage. Une autre distance utilisée fréquemment est la distance de Manhattan. Réservée aux classifications hiérarchiques, il s'agit de la somme des valeurs absolues des distances. Elle ne majore donc pas la pondération des outliers (Valeurs aberrantes et extrêmes). En effet, la distance de Manhattan essaiera de réduire toutes les erreurs de manière égale, car le gradient a une amplitude constante. Cette distance est aussi utilisée lorsque l'espace

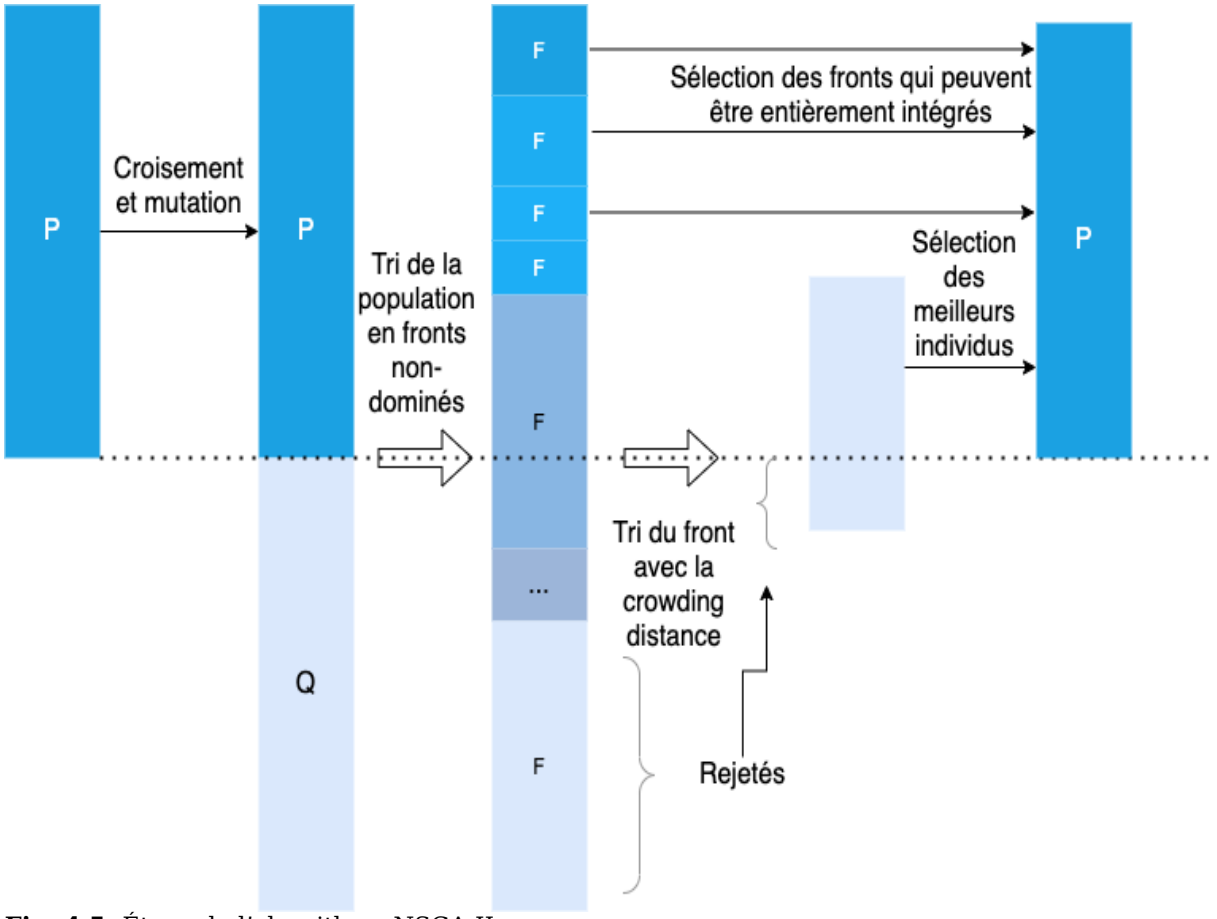


Fig. 4.5. Étape de l'algorithme NSGA-II

de recherche contient beaucoup de dimensions. En effet, cette distance permet de converger plus vite. Dans les travaux de [1], les auteurs montrent qu'avec une dimension de 20, ils obtiennent une meilleure classification avec la distance de Manhattan plutôt qu'avec la distance euclidienne. Le résultat présenté au décideur sera (1) l'ensemble des items pour lesquels les variables de décision $x_{D_v,q_j,\hat{T}}$ sont égales à 1 (2) l'ensemble des valeurs des fonctions objectifs. Selon ce que le décideur souhaitera comme valeur, il pourra choisir les items pour son contexte de décision.

4.6 Module d'apprentissage par renforcement

4.6.1 Description

Les problèmes d'optimisation multi-périodes définis dans le module d'optimisation utilisent les prévisions des paramètres résultant du module d'apprentissage et permettent une première recommandation de solutions. Le décideur à partir de ces recommandations fait un choix parmi les solutions proposées. L'affectation des items au contexte de décision entraîne la récolte de données réelles des paramètres évoluant dans le temps. Ces données réelles sont donc utilisées dans

ce module afin de mettre à jour les recommandations. L'apprentissage par renforcement est basé sur l'interaction avec son environnement. L'objectif de cet apprentissage est de savoir quoi faire dans un contexte pour maximiser les récompenses. Cet apprentissage est basé sur deux conditions :

— s'il reçoit un encouragement, alors l'action choisie est encourageante et satisfait l'objectif du décideur,

— s'il reçoit une punition, alors l'action choisie n'est pas la bonne et doit être changée, car l'action n'a pas répondu à l'objectif du décideur.

4.6.2 Fonction de mise à jour pour l'apprentissage par renforcement

Les caractéristiques de notre système :

— la récompense n'est pas donnée de manière instantanée

— en accumulant de l'expérience, le système apprendra à modifier les recommandations pour maximiser les récompenses (objectifs du décideur satisfaits).

— l'apprentissage par renforcement travaille dans un monde incertain et fait évoluer ses prédictions en fonction des données qu'il reçoit.

Le système d'apprentissage utilise une fonction de mise à jour, ainsi les recommandations sont mises à jour. Notre fonction de mise à jour est la suivante :

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

$$\Leftrightarrow$$

$$estimate_{new} \leftarrow estimate_{old} + \alpha(observation - estimate_{old})$$

Une mise à jour de la fonction objectif et de la contrainte est faite à chaque nouvelle période. Le résultat de ce module est une nouvelle recommandation d'items pour le contexte de décision étant donné la mise à jour des estimations des paramètres. La réitération du module d'optimisation est nécessaire pour réévaluer l'ensemble des solutions optimales.

4.7 Conclusion

Pour répondre aux différents verrous (QR3-C1) à (QR3-C4) nous avons mis en place un système d'aide à la décision qui se compose de plusieurs modules qui permettent de répondre individuellement ou globalement aux verrous. Notre premier module de **traitement de l'information** a pour objectif de traiter le contexte décisionnel textuel **(QR3-C4)** Pour cela, le chapitre de contribution sur l'étiquetage de séquences et la normalisation a été utilisée afin de traiter ce contexte et en extraire les informations nécessaires pour le caractériser le contexte. Ce module de traitement de l'information a aussi pour objectif d'unifier

et préparer les données qui sont variables et évolutives dans le temps **(QR3-C2)**. Afin de traiter ce verrou, une création de classes sources a été définie afin de générer des groupes de données sémantiquement proches et contribuer à l'agrégation de séries chronologiques afin de prédire les paramètres qui influenceront la recommandation d'items pour le décideur. Le module d'**apprentissage supervisé** répond au verrou **(QR3-C2)**. En effet, une prévision de la série temporelle utilisant un CNN-LSTM a été utilisée pour prédire les paramètres et pouvoir utiliser cette prédiction pour la recommandation des items dans les modules suivants. Étant donné la multiplicité des items et leurs caractéristiques, un premier filtre dans le module de **filtrage** a été proposé afin de ne garder que les items ayant des caractéristiques similaires au contexte de décision. Le deuxième objectif de ce module est de limiter l'espace de recherche des items répondant aux objectifs du décideur **(QR3-C3)**. L'avant-dernier module est un module d'**optimisation** qui utilise un algorithme génétique NSGA-II afin de choisir les items répondant aux multiples objectifs du décideur en considérant ses contraintes **(QR3-C3)**. Étant donné l'environnement incertain **(QR3-C1)** notre système d'aide à la décision met à jour ses recommandations à chaque nouvelle donnée réelle de paramètre grâce au module de **renforcement** pour améliorer les recommandations pour les périodes à venir. Dans la Figure 4.6 nous pouvons visualiser ces différentes étapes.

L'approche proposée peut avoir certaines limites puisque chaque module utilise le résultat du module qui le précède entraînant éventuellement une propagation des erreurs au fur et à mesure. De ce fait, il est important d'évaluer chaque module individuellement et caractériser le type d'erreur qui peut se produire et qui peut entraîner une recommandation non fiable des données.

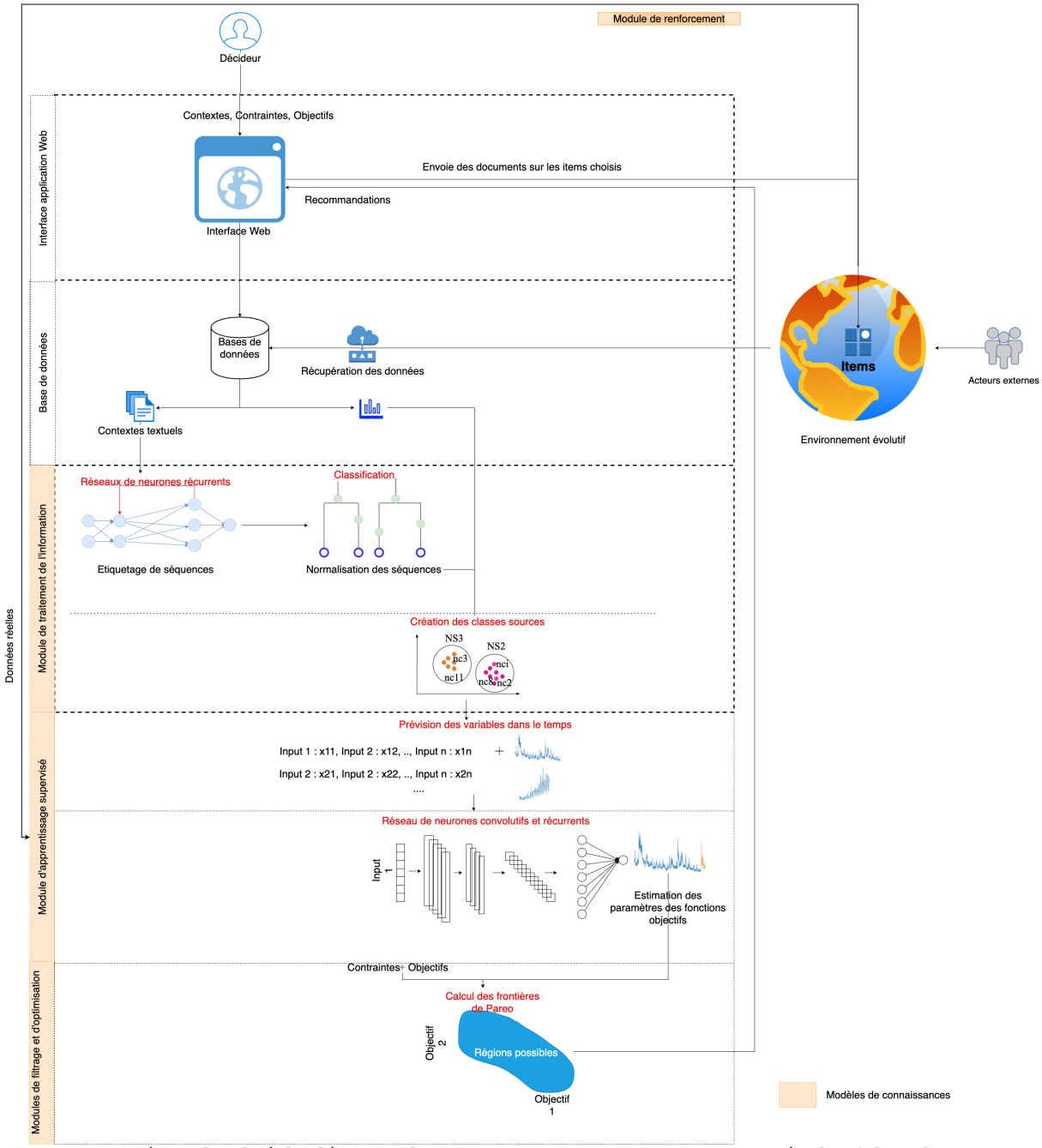


Fig. 4.6. Système d'aide à la décision dans un environnement incertain et évolutif dans le temps

5

Validation expérimentale sur l'e-recrutement

Sommaire

5.1 Contexte industriel et ses objectifs

5.1.1 Besoins industriels

5.1.1.1 Objectifs

L'entreprise partenaire Xtramile souhaite proposer une approche différente du recrutement digital, appelé aujourd'hui e-recrutement. Désormais les recruteurs utilisent les canaux de diffusion comme support pour diffuser leurs offres d'emploi. Cependant, très peu de recruteurs ont la connaissance des caractéristiques de ces canaux, d'autant plus que celles-ci évoluent dans le temps. Par exemple, certains canaux sont spécialisés dans des catégories de métier spécifiques, ou encore des types de contrat spécifique entraînant des profils candidats différents d'un canal à un autre.
Par l'usage des techniques d'analyse de données, Xtramile souhaite utiliser les données récoltées par le biais des diffusions des offres d'emploi sur le Web pour aider le recruteur à prendre une décision sur les canaux de diffusion adéquats pour chaque offre d'emploi. L'objectif de Xtramile est donc de viser à identifier les meilleures sources de diffusion des offres d'emploi pour chaque recruteur, et ce, quel que soit le positionnement géographique, le métier (informatique, ressources humaines, comptabilité, etc.), les années d'expérience demandées, etc. Dès lors l'objectif applicatif est :

— optimiser la recherche de candidats, ce qui dépasse un simple site de recrutement pour diffuser plus largement sur les réseaux sociaux, les blogs, les médias digitaux,

— proposer au recruteur les canaux de diffusion les plus pertinents, qui permettront de répondre à ses objectifs : maximisation de conversions, minimisation des clics, maximisation des CV pertinents, etc.

— aider le recruteur à allouer le budget adéquat pour atteindre ses objectifs.

Ainsi les deux questions clés posées par l'entreprise sont :

— Comment choisir les canaux de diffusion optimaux par rapport à une offre d'emploi ou une catégorie d'offre d'emploi ?

— Comment optimiser le budget alloué sur les canaux pour répondre aux objectifs du recruteur ?

Les objectifs que souhaite atteindre Xtramile en proposant un système capable de répondre à ces questions clés sont les suivants :

— Le gain de temps. Dans le recrutement digital plusieurs tâches sont chronophages et peuvent être automatisées comme (1) la création des fichiers

contenant les offres d'emploi adaptées aux requis de chaque canal (2) l'évaluation de la pertinence des canaux. De plus, certaines tâches nécessitent une expertise et la connaissance de tous les canaux de diffusion. Étant donné la multiplicité des canaux sur le Web, cette tâche reste très difficile à mettre en place par un humain.

— Le gain d'efficacité. Un des objectifs d'un tel système pour le recruteur est de l'aider à être plus efficace dans les tâches qui nécessitent une expertise et ne peuvent pas être automatisées par exemple (1) choisir parmi les CV pertinents qu'il a reçu, les candidats qu'il souhaite voir en entretien (2) faire passer les entretiens.

— Le gain d'argent. Ce système a un objectif économique en allouant de façon optimisée les budgets disponibles de recrutement. De plus, le gain de temps et d'efficacité entraînent un gain d'argent que Xtramile souhaite mettre en avant à travers ce système.

5.1.1.2 Processus de diffusion des offres d'emploi

En général Le processus de diffusion des offres d'emploi en ligne nécessite de structurer l'offre d'emploi c'est-à-dire transformer l'offre d'emploi rédigée en langage naturel en un format XML[20]. Par ailleurs, les champs constituant une offre étant différents d'un canal à un autre, il est nécessaire d'adapter ce fichier pour chaque canal sur lequel le recruteur souhaite diffuser. Cette tâche est généralement faite manuellement.

Nous avons décrit le processus de diffusion dans la Figure 5.1. Comme nous pouvons le voir dans cette figure, le recruteur doit en premier lieu transformer son offre d'emploi en format XML avec des champs adaptés pour chaque canal. Une fois le fichier prêt, il peut l'envoyer par email au canal de diffusion pour vérification de la présence de tous les champs. Une fois le fichier accepté, le recruteur est en mesure de visualiser l'offre d'emploi sur les canaux choisis. Enfin, il peut demander les données liées aux évènements de clics, vues, etc. reçus pour chaque offre sur chaque canal. Ces données sont analysées par le recruteur pour qu'il puisse adapter le choix de ses canaux pour les prochaines campagnes de recrutement.

Comme nous pouvons le constater, ce processus est lent, redondant et fastidieux. Partant de ce fait, Xtramile a souhaité s'introduire dans le marché du recrutement pour aider les recruteurs à mieux gérer ce processus.

Il existe plusieurs types d'intermédiaires du marché de travail [17] pour aider le recruteur à atteindre ses objectifs de recrutement. Ces types d'intermédiaire sont présentés dans la Figure 5.2. Ces intermédiaires proposent différents services : utiliser les connaissances d'expert ressources humaines (RH) pour trouver les candidats, faire rencontrer des recruteurs et des candidats, etc.

20. Langage informatique de description utilisé pour faciliter les échanges d'informations

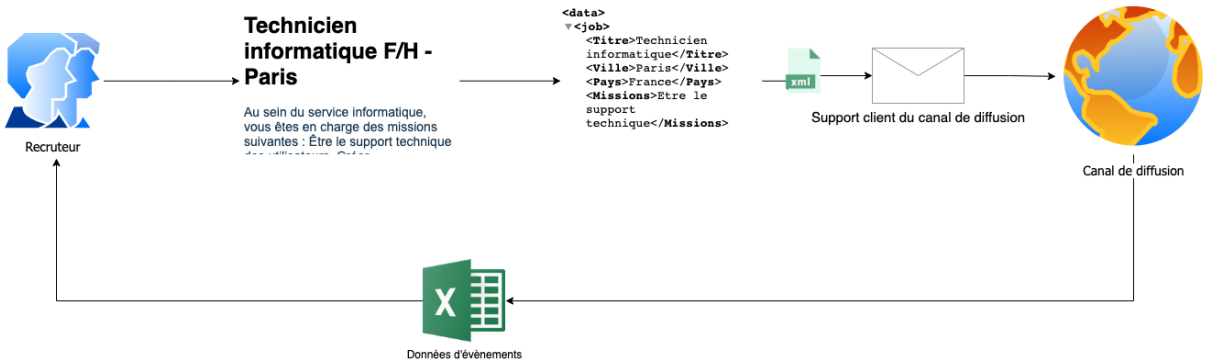


Fig. 5.1. Processus de diffusion d'une offre d'emploi sur un canal de diffusion

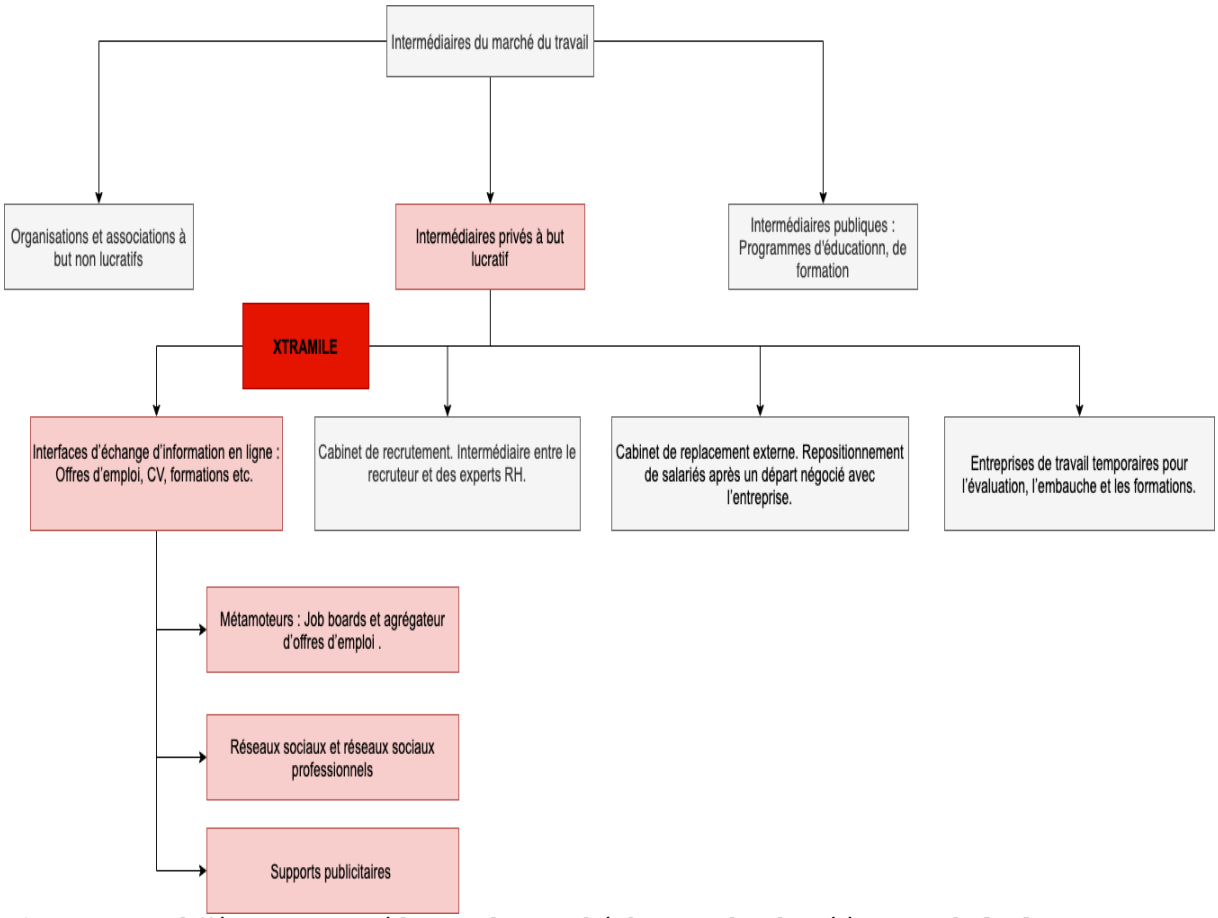


Fig. 5.2. Les différents intermédiaires du marché du travail. Adapté à partir de [17]

Xtramile Xtramile est une entreprise qui existe depuis 2015 et qui se place en tant que nouvel intermédiaire du marché de travail. Xtramile se positionne entre les intermédiaires privés à but lucratif et les interfaces d'échange d'information en ligne. Xtramile souhaite à travers ce positionnement proposer une application Web qui laisse le recruteur décider des canaux qu'il souhaite utiliser tout en l'aidant à automatiser les tâches redondantes et à choisir les canaux qui répondront à ses objectifs.

Pour ce faire, Xtramile propose une application qui s'adresse au recruteur. Le recruteur suit les étapes suivantes sur celle-ci lorsqu'il se connecte sur l'application :

— Étape 1 : Le recruteur ajoute les offres d'emploi en format souhaité (texte, XML, etc.), le budget disponible et les objectifs. Celles-ci sont stockées dans la base de données et envoyées au système Xtramile pour analyse.

— Étape 2 : Le système recommande des canaux de diffusion au recruteur sur la base de l'analyse de l'historique des évènements de clics, CV, etc. et des objectifs du recruteur.

— Étape 3 : Le recruteur choisit les canaux. Ses choix sont transmis au système.

— Étape 4 : Le système transforme les offres d'emploi en données structurées pour alimenter les fichiers XML. Le fichier est adapté pour chaque canal (choisi par le recruteur). Enfin, le système diffuse les offres d'emploi sur les canaux.

Notre contribution dans ce travail de thèse se place dans les étapes 2 et 4 qui nécessitent chacune, une contribution scientifique étant donné les verrous scientifiques que nous avons présenté dans l'introduction (QR1), (QR2) et (QR3). Ces contributions sont présentées dans les chapitres 2 à 4. Avant d'expérimenter ces contributions sur le domaine du e-recrutement nous le modélisons dans le contexte applicatif de l'entreprise Xtramile.

Pour ce faire, nous avons tout d'abord modélisé les acteurs principaux que nous présentons par la suite.

5.1.2 Modélisation du e-recrutement dans le contexte Xtramile

À l'issue des analyses de données de Xtramile et de l'état de l'art, nous souhaitons proposer un modèle conceptuel du domaine du e-recrutement qui regroupe trois principaux acteurs : le recruteur, le candidat et le Web et qui cherche à répondre aux objectifs industriels définis précédemment. La Figure 5.4 modélise en UML les différents concepts de notre modèle. En rose sont présentées les caractéristiques de l'acteur candidat, en orange celles du recruteur et enfin en violet celles du Web. Ces trois acteurs sont présentés plus en détail dans les sections ci-dessous.

5.1.2.1 Le recruteur

Dans l'application Xtramile le recruteur est représenté par une offre d'emploi qui décrit le profil du candidat recherché. Une offre d'emploi est caractérisée par des

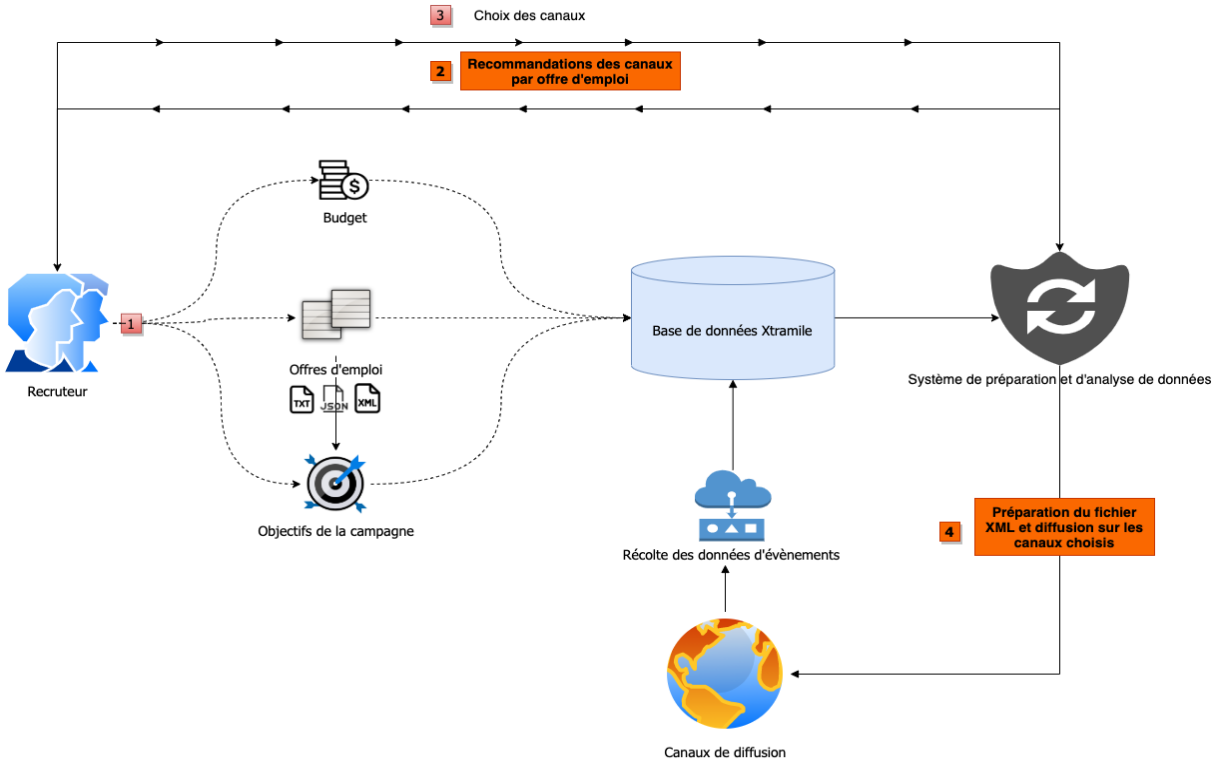


Fig. 5.3. Processus de diffusion d'une offre d'emploi sur un canal de diffusion sur l'application Web Xtramile.

besoins spécifiques sur le métier, les compétences techniques, générales, transversales, les missions, l'expérience, la formation recherchée, etc. Il est important dans le cadre de cette thèse de bien spécifier les informations qu'une offre d'emploi peut contenir et qui servira à l'analyse de la pertinence d'un CV [160] et aussi de la préparation des fichiers XML pour la diffusion des offres d'emploi. Avec l'aide d'un expert en ressources humaines, nous avons identifié, les informations suivantes comme étant importantes pour décrire le profil recherché : "ville ", "type de contrat", "formation", "expérience", "durée de l'expérience", "compétences techniques", "compétences générales", "missions", "métier", "code postal" et "salaire". Certaines de ces informations sont couramment utilisées dans la littérature [24, 75], et d'autres, telles que "missions" et "compétences générales", ont été ajoutées à cet ensemble. Les "missions" ne sont pas souvent incluses dans la littérature, mais cela permet d'identifier les compétences transversales, importantes selon l'expert. En outre, les "compétences générales" (travail en équipe, autonomie, etc.) sont des informations essentielles qui sont couramment utilisées par les recruteurs.

Le recruteur est aussi représenté par des objectifs et des contraintes financières. Les objectifs du recruteur peuvent être multiples par exemple maximiser le nombre de conversions ou maximiser le nombre de CV pertinents. Ces objectifs sont contraints par un budget à ne pas dépasser.

Enfin, le recruteur est également représenté par son entreprise qui a un nom, un lieu géographique, mais aussi :

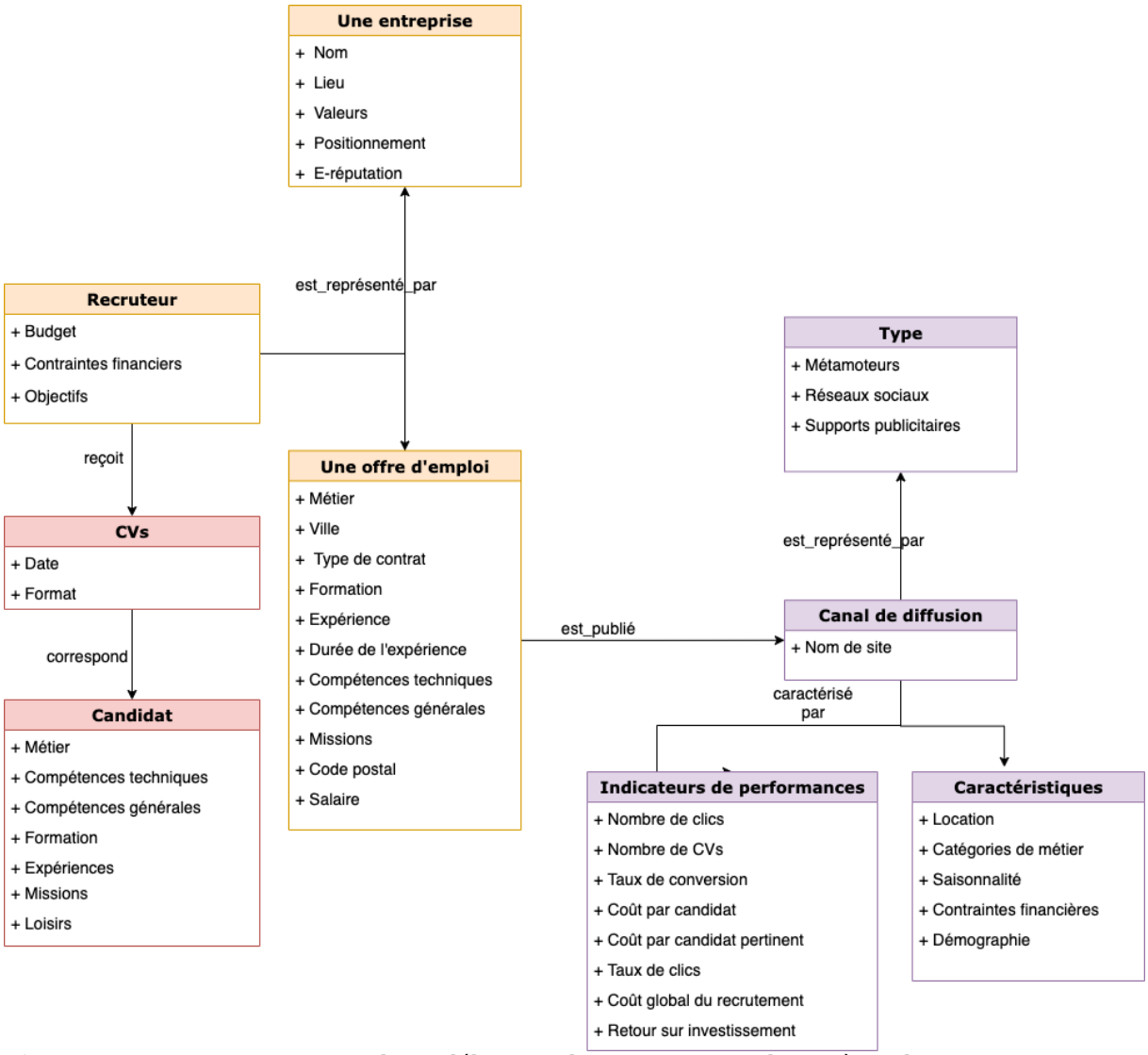


Fig. 5.4. Diagramme UML pour la modélisation des trois acteurs du système de e-recrutement

— Des valeurs qui la définissent. Cette caractéristique constitue un filtre et permet d'assurer que les candidats sont compatibles ou pas avec la culture de l'entreprise.

— Un positionnement par rapport à ses concurrents, qui permet de filtrer les canaux de diffusion où l'entreprise risque d'être pénalisée par un positionnement moins important que ses concurrents.

— La e-réputation sur le Web qui correspond à la réputation de l'entreprise aux yeux des internautes et des anciens et actuels employés. Cette caractéristique permet de filtrer les canaux de diffusion où l'entreprise risque d'être pénalisée par sa e-réputation.

Ces différentes caractéristiques de l'entreprise peuvent influencer les évènements de clic, vues, etc. De ce fait, il est important de les considérer dans le modèle recruteur pour permettre de les considérer si cela est opportun dans le processus de recommandation des canaux.

5.1.2.2 Les candidats

Le candidat est un acteur qui joue un rôle important dans le processus de choix des canaux. En effet, il s'agit d'un acteur important pour le recruteur. Le candidat interagit avec le canal en faisant un ensemble d'actions pour une offre d'emploi. Cette action traduit un intérêt ou non du candidat qui peut être utilisée pour les recommandations des canaux au recruteur. Ses actions se traduisent par un clic, une vue, une réponse à un formulaire, un envoi de CV, etc. Lorsque l'action du candidat est la transmission du CV, ce fichier est exploité afin d'analyser le profil du candidat et d'en tenir compte pour les recommandations. Le CV contient les informations suivantes (présentées aussi dans la Figure 5.4 sous l'entité candidat) : "Métier", "Compétences techniques", "Compétences générales", "Formation", "Expériences", "Missions", "Loisirs". Ces informations ont également été identifiées par l'expert RH. Nous avons pu le confirmer en vérifiant leur représentativité dans 200 CV. Dans le cadre de nos échanges avec les experts RH nous avons identifié que les compétences générales (le travail en équipe, l'autonomie, etc.) sont des informations exploitées par les recruteurs. En effet, les recruteurs sont aujourd'hui à la recherche de profils qui possèdent, en plus de leurs connaissances techniques, des compétences nécessaires pour s'intégrer aux équipes et aux valeurs de l'entreprise. Cette information est une information qui peut être implicitement présente dans les CV. En effet, elle peut être déduite à partir des centres d'intérêts du candidat, des différents métiers qu'il a réalisé, etc.

5.1.2.3 Les canaux de diffusion

Le dernier acteur est l'intermédiaire entre le recruteur et les candidats : le canal de diffusion. Nous définissons un canal de diffusion au travers de plusieurs caractéristiques. Certaines caractéristiques sont issues de la littérature et d'autres ajoutées par les experts RH pour mieux modéliser cet acteur.

Les types de canaux Tout d'abord, un canal de diffusion a plusieurs types [17] présentés dans la Figure 5.2 en dessous de l'intermédiaire d'échange d'information en ligne :

— les métamoteurs, sont des systèmes Web de recherche d'emploi, par exemple Indeed, Glassdoor, etc. Il existe différents types de métamoteurs. Il y a tout d'abord les métamoteurs dit « généralistes ». Ces derniers remplissent la fonction de mise en ligne des offres d'emploi ainsi que l'enregistrement des CV des candidats, toutes catégories d'offres d'emploi et tout profil candidat. D'un autre côté, il y a les métamoteurs dits « spécialisés » qui eux se focalisent sur une catégorie spécifique d'offres d'emploi. Par exemple jobFinance qui se focalise sur la catégorie d'offres d'emploi Finance.

— les réseaux sociaux et les réseaux sociaux professionnels, qui permettent une diffusion de l'offre d'emploi sous un format publicitaire pour attirer l'attention d'actuel ou futur demandeur d'emploi (respectivement candidats actifs et candidats passifs). Par exemple Facebook et Linkedin sont deux réseaux sociaux qui ont des cibles professionnelles différentes. Dans Facebook on y retrouve des étudiants donc les offres d'emploi mis en avant sont généralement des alternances par exemple.

— Les supports publicitaires classiques (support display) qui, de la même façon que les réseaux sociaux diffusent des publicités sur différents sites de tout type. Les supports sont utilisés en complément des métamoteurs pour attirer des candidats (améliorer la marque employeur de l'entreprise du recruteur) et donc maximiser les chances de conversion de candidats sur les métamoteurs et les réseaux sociaux.

Selon les experts RH, les recruteurs souhaitent de plus en plus mobiliser les supports publicitaires et les réseaux sociaux pour anticiper certains besoins et aller vers des candidats qui ne seraient pas venus spontanément vers eux. Ils sont intéressés en particulier par les candidats dits « passifs » qui sont en veille par rapport aux opportunités du marché. Nous notons également la volonté d'avoir une approche plus qualitative, plus relationnelle avec les candidats [36] entraînant la nécessité de réduire le nombre de CV non pertinents reçus pour chaque offre.

Les caractéristiques des canaux Dans cette section, nous présentons les caractéristiques des canaux. Certaines d'entre elles sont issues de la littérature et d'autres ajoutées par les experts RH. Les canaux ont différentes caractéristiques, qui peuvent différer d'un canal à un autre. Par exemple, sur les métamoteurs les candidats sont à la recherche activement d'un emploi. Ils sont appelés candidats actifs [36]. Sur les réseaux sociaux et les supports publicitaires les candidats peuvent être passifs. Ils ne sont pas à la recherche active d'un nouveau poste, mais sont à l'écoute du marché du travail [36].

Les caractéristiques que partagent ces différents types de canaux sont les suivantes :

(a) Une spécificité sur la localisation, pays ou ville. Chaque canal de diffusion cible un ou plusieurs pays en particulier. Par exemple 86% des candidats en France sont actifs sur le métamoteur Glassdoor, contre 1.37% aux États-Unis [21].

(b) Une spécificité sur les catégories de métiers. Il existe différents types de métamoteurs. Il y a tout d'abord les métamoteurs dit « généralistes ». Ceux-ci remplissent la fonction de mise en ligne des offres d'emploi ainsi que l'enregistrement des CV des candidats, toutes catégories de métier et tous profils confondus. D'un autre côté, il y a les métamoteurs dits « spécialisés » qui eux se focalisent sur une catégorie spécifique du marché de l'emploi. Par exemple "LesJeudis" est un métamoteur spécialisé qui ne diffuse que des offres d'emploi en informatique.

(c) Saisonnalité :

— Propres à chaque catégorie de métier [17]. En effet, il existe des fluctuations de l'activité des candidats pour chaque secteur de métier. Certains secteurs deviennent de plus en plus en pénurie et nécessitent l'étude du marché de l'emploi avec précision. Par exemple "en finance et comptabilité, le deuxième trimestre apparaît plus favorable aux embauches". Cette période coïncide avec la remise de diplôme des étudiants.

— Paramètres généraux de l'emploi [17, 51]. Dans ce cas, les fluctuations dépendent de plusieurs facteurs tels que les vacances scolaires, les périodes de délivrance de diplômes, etc. Par exemple, "Prospecter des candidats pendant l'été permet de prévoir le redémarrage de l'activité après les vacances estivales".

(d) Contraintes financières :

— Payant ou gratuit

— Le coût d'une diffusion d'offres d'emploi, qui est représenté par un prix par clic (coût par clics), par un prix toutes les 1000 vues (coût par Mille), etc.

(e) La démographie. Les canaux sont plus ou moins populaires selon certaines caractéristiques de candidats, tels que l'âge, le sexe, etc.

Les indicateurs de performance des canaux Les indicateurs de performance sont des métriques utilisées par les recruteurs pour mesurer la pertinence des canaux. Les indicateurs de performance présentés ci-après sont ceux très souvent utilisés par les recruteurs (données issues des campagnes de recrutement menées chez Xtramile) :

— Le nombre de clics que l'offre d'emploi a reçu, est un indicateur qui permet au recruteur d'évaluer si son offre est visible par les candidats.

— Le nombre de CV que l'offre d'emploi a reçu qui permet au recruteur d'évaluer si les candidats sont intéressés par son offre.

21. Source provenant du site `https://www.similarweb.com/`

— Le taux de conversion qui correspond au rapport entre le nombre de candidats qui ont cliqué et les candidats qui ont envoyé leurs CV. Cet indicateur permet de déduire l'efficacité d'une campagne de recrutement, mais pas de sa rentabilité. Sur les métamoteurs, le taux se situe entre 3 et 7%[22] alors que sur les supports publicitaires classiques, il est plutôt de 2%. En effet, les métamoteurs disposent d'une audience de candidats en recherche active et donc réceptive. Sur les supports publicitaires classiques et réseaux sociaux, on s'adresse à une audience plus large de candidats potentiels et générant des candidatures incrémentales.

— Le coût par candidat correspond au budget dépensé pour recevoir un CV. Celui-ci permet au recruteur d'estimer le prix d'un CV.

— Le coût par candidat pertinent qui correspond au budget dépensé pour recevoir un CV qualifié. Cet indicateur permet de déduire la rentabilité d'une campagne de recrutement.

— le rapport entre le nombre d'utilisateurs qui cliquent sur l'affiche de la publicité et le nombre total d'utilisateurs qui consultent la page de la publicité appelé CTR. Il permet au recruteur de connaître le taux d'intérêt de sa publicité.

— Le coût global du recrutement par rapport aux profils réalisés. Cet indicateur nécessite de suivre le processus global de recrutement, de la diffusion de l'offre à l'intégration du candidat. Ce coût permet d'anticiper le budget pour les futures campagnes de recrutement.

— Le retour sur investissement (ROI) qui permet de mesurer et de comparer le rendement d'un investissement dans un canal de diffusion plutôt que dans un autre canal. Généralement, le retour sur investissement se base sur le calcul du ratio bénéfices de l'investissement / coût de l'investissement. Dans le cas du recrutement, et notamment des canaux de diffusion, le ROI représente le rendement d'un canal plutôt qu'un autre.

Xtramile propose dans son système l'ensemble de ces indicateurs comme paramètres d'objectifs du recruteur.

Pertinence d'un canal La pertinence d'un canal de diffusion dépend des besoins du recruteur. En effet, chaque entreprise a ses propres objectifs. Les facteurs explicatifs de la pertinence d'un canal sont de deux types différents [130] :

1. Facteurs inflexibles qui sont déterminés à travers l'objectif du recruteur et sont donc considérés comme des données. L'objectif du recruteur est formulé sur l'application en choisissant des indicateurs de performance précis par exemple : nombre de CV = 10 ou coût par candidat = 15 euros. L'intérêt de cette approche est que la pertinence d'un canal est définie à travers les indicateurs de performance choisis par le recruteur.
2. Facteurs flexibles qui entrent en jeu lorsque le recruteur n'a pas assez de recul pour avoir des objectifs clairs sur la performance souhaitée lors de sa campagne de recrutement. Dans ce cas, les objectifs sont définis par un manager des campagnes de recrutement qui analyse à travers son historique de données les indicateurs de performance associés au type de profil recherché par le recruteur.

22. Source provenant du site `https://www.goldenbees.fr/`

Si ce type de profil n'a jamais été traité, une comparaison des indicateurs de performance cités plus haut pour chaque canal est effectuée. Il n'existe pas de règles strictes pour cette analyse, la comparaison de tous les indicateurs permet d'identifier les canaux pertinents ou non individuellement ou en combinaison.

Xtramile utilise les deux facteurs de pertinence. Si le recruteur a des objectifs, le système proposé et la recommandation des canaux s'appuiera sur ces objectifs. De l'autre côté, si les objectifs sont inconnus par le système Xtramile, par défaut l'objectif est la maximisation du nombre de conversions (qui est un besoin assez commun entre les recruteurs).

5.1.3 Interactions des différents acteurs modélisés dans l'application Xtramile

Les différents acteurs modélisés ci-dessus interagissent de différentes façons sur l'interface Web Xtramile.

5.1.3.1 Interactions recruteur/interface

Le recruteur a accès à une interface Web sur laquelle il peut créer une campagne de recrutement (Activité A0) qui consiste en plusieurs sous-activités présentées dans le diagramme d'activité présenté 5.5 :

1. Le recruteur ajoute les offres d'emploi sur l'interface.
2. Le recruteur ajoute les contraintes financières liées à chaque offre.
3. Ces informations sont stockées par le système. Celles-ci sont affichées sur l'interface Web.
4. Le système envoie des recommandations de canaux.
5. Le recruteur sélectionne les canaux.
6. Le système génère les fichiers XML contenant les offres d'emploi et les informations requises pour chaque canal choisi.
7. Le recruteur reçoit le fichier XML sur l'interface et débute la campagne.

5.1.3.2 Interactions Recruteur/ Système/ Canaux de diffusion/Candidats

Le recruteur prépare sa campagne à l'activité A0 Figure 5.5 que nous avons décrit dans la section précédente. Une fois le recruteur lance la campagne :

1. Les offres sont diffusées sur les canaux choisis.
2. Les candidats (utilisateurs des canaux) peuvent désormais visualiser les offres d'emploi et faire une action :
   — Cliquer sur l'offre.
   — Voir l'offre.
   — Envoyer son CV.
3. Cette action est enregistrée par le système.

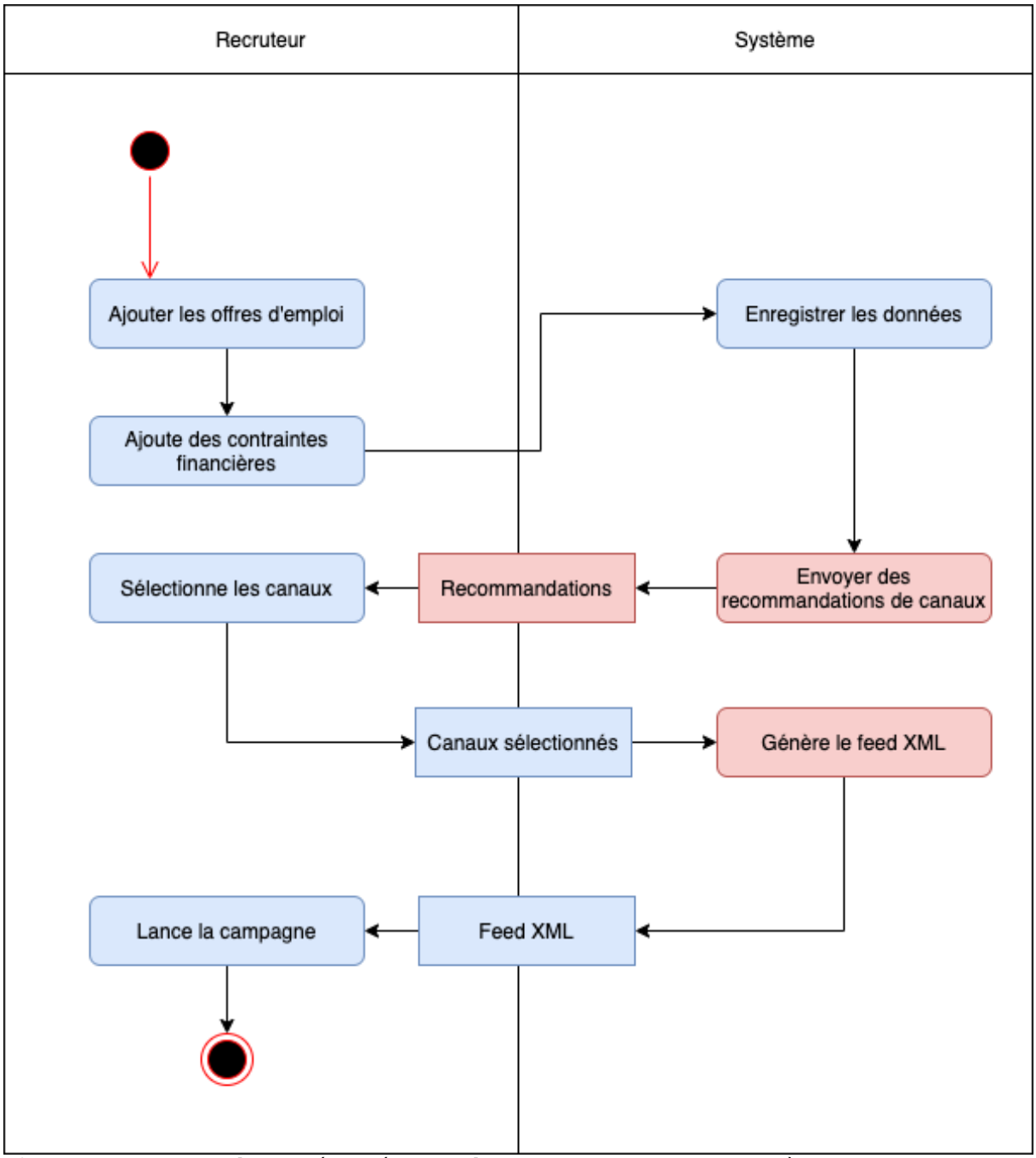


Fig. 5.5. Diagramme d'activité représentant les interactions recruteur/système

4. Le système calcule les indicateurs de performance et les affiche au recruteur sur l'interface.
5. Le système met à jour les recommandations après une semaine de campagne (période habituelle pour vérifier les évènements et les résultats des choix des canaux par le recruteur) après avoir évalué les premiers retours sur les indicateurs de chaque canal.
6. Le recruteur peut visualiser et mettre à jour le choix des canaux.

Ces actions sont représentées dans le diagramme d'activité Figure 5.6

Les contributions de nos travaux de recherche sont colorées en rose dans les diagrammes d'activités présentés Figures 5.5 et 5.6. Dans la suite du document, nous allons expérimenter nos contributions scientifiques présentées dans les chapitres 2, 3 et 4 dans le cadre contextuel présenté dans ce chapitre.

5.2 DEEP : Une méthodologie pour l'extraction d'entités en se basant sur le schéma organisationnel à partir de textes rédigés en langage naturel

Nous proposons d'évaluer DEEP dans le domaine des offres d'emploi. Une offre d'emploi est un texte simple avec trois caractéristiques principales :

— Elle est rédigée librement dans un langage naturel. Les informations et le vocabulaire utilisés dans une offre d'emploi peuvent donc différer d'une offre à l'autre.

— Elle présente un schéma organisationnel. Une offre d'emploi se compose d'une série de sections, chacune contenant un type d'information spécifiques. Par exemple, la section "description de l'entreprise" contient son nom, ses valeurs, son type de structure, etc. Même si un ordre type entre ces sections est communément adopté, il peut varier d'un emploi à l'autre.

— Le vocabulaire évolue. Depuis quelques années, on assiste à l'émergence de nouveaux métiers, de nouvelles compétences et le vocabulaire utilisé dans les offres d'emploi a donc tendance à évoluer. En outre, certaines informations partagent un vocabulaire commun (par exemple, compétence et expérience). Par conséquent, cela peut conduire à une ambiguïté lors de l'étiquetage de l'offre d'emploi.

Pour l'évaluation, nous avons exploité un corpus réel d'offres d'emploi que nous avons constitué, composé de 3 335 offres d'emploi françaises (1 094 562 mots), extraites de plusieurs sites d'offres d'emploi français (Indeed[23], Leboncoin[24], ...) entre 2017 et 2019. Les offres d'emploi étaient réparties équitablement entre 25 secteurs d'activités (ressources humaines, informatique, éducation, ...) et contenaient chacune une moyenne de 328 mots avec un écart-type de 20,4. Il y avait 21 790 mots différents dans l'ensemble du corpus.

23. Moteur de recherche d'emploi
24. Site d'annonces commerciales

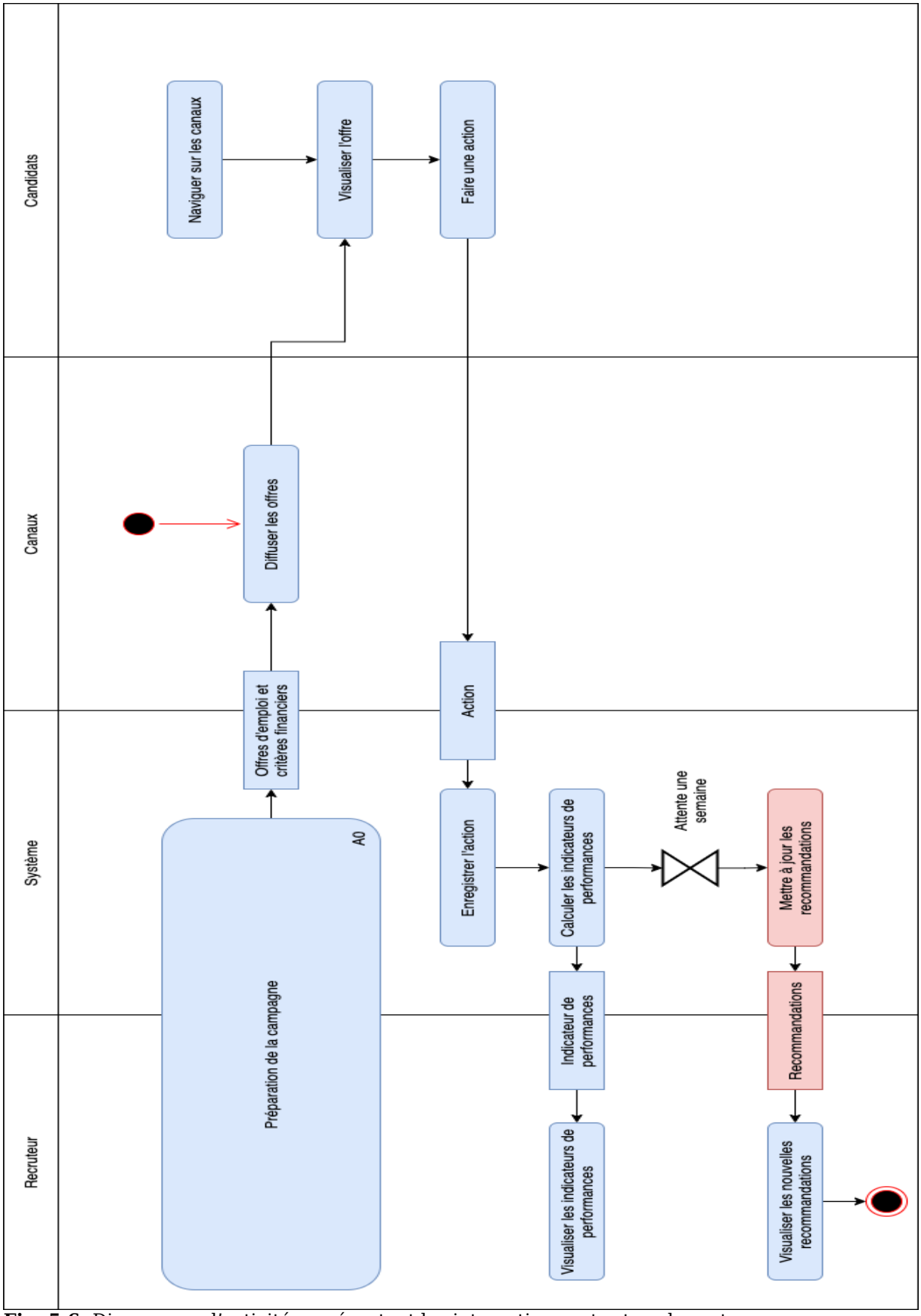


Fig. 5.6. Diagramme d'activité représentant les interactions entre tous les acteurs

L'ensemble de données peut sembler petit. Cependant, l'objectif est de montrer qu'une grande précision peut être obtenue même avec un petit nombre de documents lorsque le corpus d'apprentissage contient peu d'ambiguïtés et que l'algorithme choisi est adapté.

Les participants à cette évaluation sont : un recruteur dans le rôle de l'expert, un employé des ressources humaines ayant une expérience d'un an dans le rôle du master et trois étudiants en licence ou master chacun dans un domaine différent dans le rôle des annotateurs.

Nous décrivons ci-après chaque étape de la méthodologie.

5.2.1 Création du corpus d'apprentissage

5.2.1.1 Création du guide d'instruction pour les annotateurs (A1)

Rappelons que DEEP repose sur la création d'un corpus d'apprentissage. En pratique, deux corpus sont créés. Le corpus de validation est utilisé pour valider le guide d'instruction et contient 335 offres d'emploi (10% du corpus). Le corpus final contient 3 335 offres d'emploi.

1. A11. Définition des schémas organisationnels des textes. L'expert a étudié et analysé les schémas organisationnels des offres d'emploi du corpus de validation, pour aboutir à un tableau comprenant toutes les informations permettant de décrire un schéma organisationnel, présenté ci-après.

   — **L'ordre le plus courant des informations.** : la première section présente généralement l'entreprise à travers ses activités, sa taille, ses valeurs, etc. La deuxième section est généralement consacrée à la description du poste à travers les principales tâches, le type de contrat, le salaire, etc. La dernière section concerne le profil recherché à travers les compétences requises, l'expérience, la formation, etc. La section suivante concerne les conditions (date de début, etc.) et enfin les informations complémentaires telles que le contact e-mail, etc. Cet ordre confirme que les recruteurs suivent généralement un schéma organisationnel pour une offre d'emploi.

   — **Les différentes sections** dans les offres d'emploi sont : la description de l'entreprise, la description du poste, le profil souhaité, les conditions, les informations complémentaires.

   — **La description**. Une description de chaque section et du type de schéma organisationnel a été ajoutée au guide d'instruction. Par exemple, la section de description du profil contient la durée de l'expérience, suivie d'une compétence, d'un métier ou d'une spécialisation.

   — **Les indicateurs** choisis pour l'expérience requise sont les suivants : expérience, durée, année, domaine, etc. Le recruteur a fourni des mots indicateurs pour chaque section et étiquette.

   — **Les exemples de séquences** Par exemple, pour les missions, "La mission principale du poste est de réaliser des simulations avec le modèle de biochimie", est

une séquence exemple pour l'étiquette "Missions". Les séquences associées à l'étiquette "Missions" utilisent généralement des sujets qui décrivent une action.

2. A12. Identification de l'ensemble des étiquettes. L'ensemble des étiquettes est choisi pour répondre à un besoin. Dans le cas de cette expérimentation, il y a deux besoins :
   - la diffusion des offres d'emploi sur les sites d'emploi (car l'entrée pour cette tâche est un document en format XML. Le format XML est approprié pour représenter les informations hiérarchiques, car c'est un fichier contenant des balises pour catégoriser les informations [147]).
   - Utiliser l'entité et les séquences associées pour calculer les similarités avec le CV. Pour ce faire, le master a identifié l'ensemble suivant de 11 étiquettes "ville ", "type de contrat", "formation", "expérience", "durée de l'expérience", "compétences techniques", "compétences générales", "missions", "métier", "code postal" et "salaire" présentées dans la modélisation du e-recrutement.
3. **A13. Création des règles d'annotation spécifiques au domaine.**
   L'expert a créé 6 règles liées au domaine des ressources humaines. Par exemple, une règle spécifique pour distinguer les "compétences techniques" des "compétences générales" : "si une compétence peut être transversale (peut être appliquée dans différents métiers), il s'agit d'une compétence non technique. Sinon, il s'agit d'une compétence technique (spécifique à une profession)". La Table E.1 de l'annexe E contient ces règles.
4. **A14. Création de règles d'annotation générales.**
   Le master a créé 15 règles relatives à l'annotation générale. Par exemple, une règle spécifique à la ponctuation, qui prend en compte la ligne sautée qui sépare différentes séquences liées à la même étiquette. La Table E.2 de l'annexe E contient ces règles.
5. **A15. Composer le guide d'instruction.**
   Le master a créé le guide d'instruction contenant l'objectif, l'entrée et la sortie de chaque sous-activité.
6. **A16. Valider le guide d'instruction.**
   L'expert valide le guide d'instruction

5.2.1.2 Étiquetage manuel du corpus (A2)

1. **A21. Choix de l'outil d'annotation.**
   Le master a comparé différents outils d'annotation afin de choisir celui qui convient à cette tâche sur la base de différentes caractéristiques présentées dans la Table 5.1. L'outil choisi par le master est Dataturks car c'est celui qui répond à toutes les caractéristiques. Le master a ensuite créé trois projets d'annotation différents qui contiennent le corpus de validation des offres d'emploi pour chaque annotateur.

Table 5.1
Comparaison des outils d'annotation

Caractéristiques	Dataturks	Brat	Amazon Mechanical Turk	LabelBox
Open source	✓	✓	✓	✓
Web/Standalone	Web	Web/Standalone	Web	Web/Standalone
Annotation collaborative	✓	✗	✓	✓
Export en Json/XML	✓	✓	✓	✓
Filtres	✓	✗	✓	✓
Rapports sur les annotations	✓	✗	✗	✓
Crowdsourcing	✓	✗	✓	✓
Simplicité d'utilisation	⋆⋆	⋆	⋆	⋆⋆⋆
Étiquetage facile et intuitif	⋆⋆⋆	⋆	⋆	⋆⋆
Accès et création de projets intuitives	⋆⋆	⋆	⋆⋆	⋆⋆
Coût	Gratuit	Gratuit	Gratuit	Payant

2. **A22. Pré-validation du guide d'instruction.**

   (a) **A221. Étiquetage manuel du corpus de validation par les annotateurs.** Les annotateurs ont pris comme entrée le corpus de validation de l'outil et le guide d'instruction (voir Annexe A) pour étiqueter les documents manuellement. L'annexe Figure B montre un exemple d'offre d'emploi étiquetée manuellement par un annotateur dans Dataturks. Dans cet exemple, la séquence "développeur Web (HTML/CSS)" est étiquetée "métier".

   (b) **A222. Validation du guide d'instruction.**
   Les mesures IAA et GAA sont examinées après que tous les annotateurs aient fini d'annoter le corpus.

   — L'accord inter-annotateurs (IAA) est de 88%.

   — L'accord gold-annotateur (GAA) est de 75%.

   En raison de la valeur élevée de l'IAA, nous pouvons valider partiellement le guide d'instruction et les règles générales. Le GAA est inférieur au seuil (79% qui est considéré comme une valeur élevée [16]). Nous pouvons conclure que les règles relatives au domaine du recrutement ne sont pas claires. Les séquences annotées qui ne correspondaient pas à celles provenant du corpus de validation ont été identifiées et extraites. Après avoir étudié ces séquences, le master a ajouté 4 règles qui ont été validées par l'expert pour

ces séquences spécifiques. La deuxième itération de la validation a donné un IAA égal à 92% et un GAA égal à 88%. Comme prévu, la deuxième itération a pris moins de temps que la première comme nous pouvons le voir dans la Table 5.2.

5.2.1.3 Annotation manuelle du jeu de données final (A3)

La dernière activité de création de corpus a été l'annotation manuelle du corpus final, qui a été effectuée par trois annotateurs qui ont annoté chacun 1/3 du corpus. Le corpus étiqueté créé après l'annotation manuelle était composé de 317 693 séquences étiquetées, représentant 66% des séquences. 34% des séquences ont été annotées comme "autres" (non associées à une étiquette) et donc non essentielles (elles ne fournissent pas d'informations importantes). La création du corpus a nécessité 300 heures de travail pour l'ensemble des participants : 10% pour l'expert, 30% pour le master, et enfin 60% pour les annotateurs (l'objectif étant de réduire le temps des experts). L'expert et le master n'ont pas été impliqués davantage après la validation des directives.

Table 5.2
Rapport du temps passé pour chaque annotateur (secondes/offre)

Annotateur	Première itération (A221)	Seconde itération (A222)	Annotation finale (A3)
Nombre d'offres d'emploi	**335**	**125**	**3,335**
1	240	75	150
2	230	80	165
3	200	75	140

5.2.2 Enrichissement du corpus d'apprentissage

Le pré-traitement des données a été effectué à l'aide d'une analyse morphosyntaxique. Nous avons également ajouté au corpus la lemmatisation des mots, la position de la séquence dans le document et les mots qui la composent. La Figure 5.7 montre la répartition des différentes étiquettes dans le corpus final. On peut voir que "compétence technique", "missions" et "métier" représentent les étiquettes les plus fréquentes dans les offres. En outre, les "compétences générales" représentent également une partie essentielle des étiquettes. Cette information confirme qu'il s'agit d'une information importante pour le recruteur. Enfin, on note que le "salaire" est l'étiquette la moins utilisée, ce qui signifie qu'il ne s'agit pas d'une information obligatoire dans les offres. Ces valeurs serviront de base à l'interprétation des résultats du modèle d'extraction automatique d'entités.

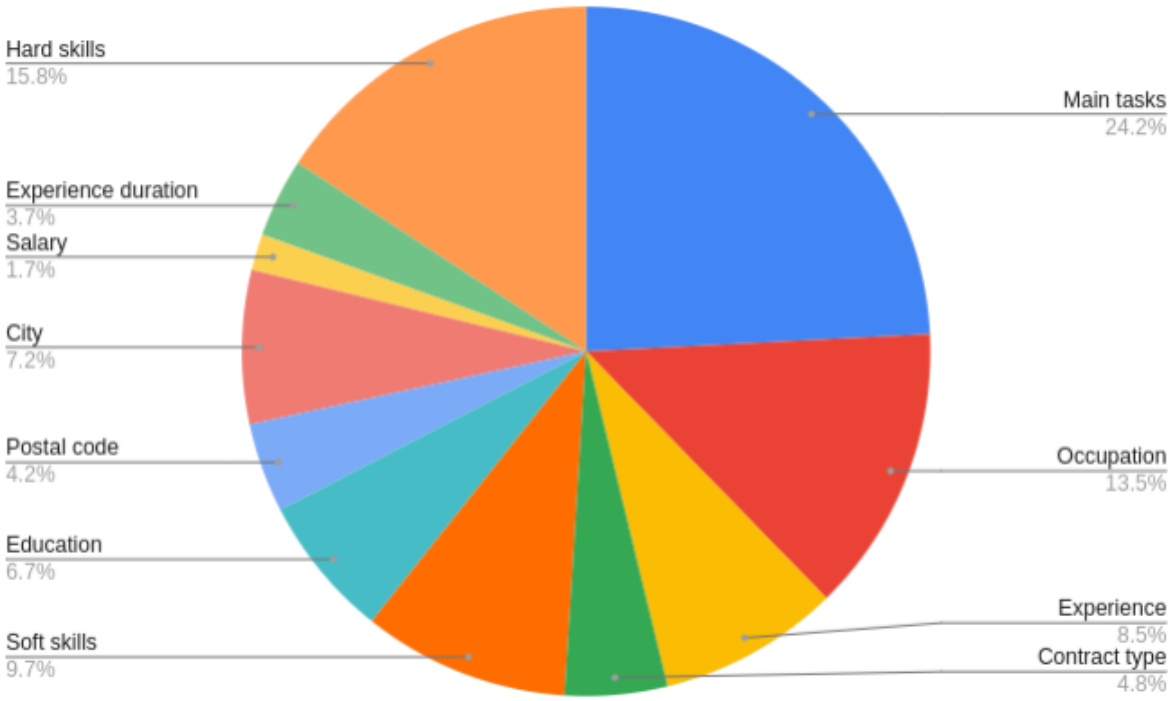


Fig. 5.7. Répartition des étiquettes dans les offres d'emploi

5.2.3 Application de l'algorithme d'apprentissage

Nous avons choisi de nous concentrer sur les algorithmes les plus populaires dans la littérature, à savoir CRF, LSTM CRF et bi-LSTM CRF. L'entrée de ces modèles d'apprentissage est le corpus enrichi. La sortie de ces modèles est la paire étiquette/séquence.

La première expérience a utilisé le modèle LSTM avec les mêmes paramètres que ceux utilisés dans [40], à l'exception de certains d'entre eux que nous avons modifiés pour adapter le modèle à la taille du jeu de données, au nombre moyen de mots par texte brut et par étiquette : la représentation vectorielle utilisée a une taille de 150. Les couches cachées du réseau neuronal récurrent avaient une taille de 512. Le Dropout [40], qui est une méthode de régularisation visant à empêcher le surapprentissage, était de 0,2. Pour les modèles CRF LSTM et CRF bi-LSTM, nous avons choisi les mêmes paramètres que les LSTM, auxquels nous avons ajouté une couche de sortie CRF. La représentation vectorielle utilise le modèle CamemBERT, qui est un modèle de langage pré-entraîné sur un jeu de données francophone. Ce modèle est une version de BERT pour laquelle, certains hyperparamètres du pré-entraînement ont été modifiés.

Nous nous intéressons également à l'algorithme d'apprentissage SVM, qui est largement utilisé dans la littérature et qui a montré des performances exceptionnelles. Cependant, le SVM ne prend pas en compte l'enchaînement des séquences, c'est-à-dire les schémas organisationnels. La classification SVM a été choisie pour valider l'impact de la prise en compte des schémas organisationnels et, par conséquent, de

l'évolution du vocabulaire, de la minimisation de l'ambiguïté associée au vocabulaire partagé entre les étiquettes, et du choix de l'approche d'étiquetage des séquences. La technique d'évaluation choisie a été la validation croisée 10-fold appliquée sur le corpus d'apprentissage étiqueté enrichi, comme nous pouvons le voir sur la Figure 5.8.

Les métriques d'évaluation que nous avons choisi sont les métriques traditionnelles : précision (P), rappel (R) et score F1 (F1). Le jeu de données utilisé est le corpus entier étiqueté (3 335 offres d'emploi).

5.2.4 Résultats

5.2.4.1 Performance des algorithmes choisis

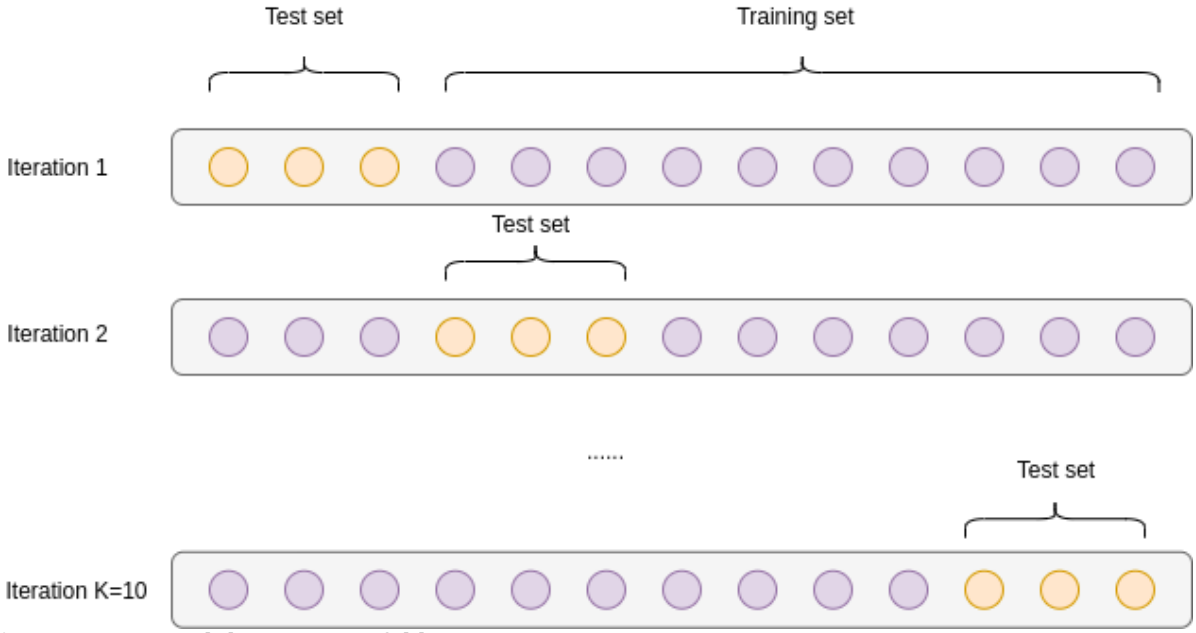


Fig. 5.8. Cross-validation en 10-fold

5.2.5 Résultats

5.2.5.1 Performance des algorithmes choisis

La Table 5.3 montre la prédiction, le rappel et la F1 des modèles CRF, LSTM CRF, bi-LSTM CRF et SVM pour chaque étiquette. Nous avons tout d'abord observé que la valeur moyenne de F1 pour CRF, LSTM CRF, bi-LSTM CRF était très élevée : supérieure à 0.89. Nous pouvons en déduire que l'approche d'étiquetage des séquences est adaptée à l'extraction d'entités.

Nous pouvons également observer que la valeur F1 associée à chaque étiquette pour tous les modèles est très élevée (au moins 0.76). Nous constatons que le modèle CRF

Table 5.3
Précision, Rappel et F1 de l'apprentissage supervisé sur les quatre modèles

	CRF			LSTM CRF			bi-LSTM CRF			SVM		
Métriques	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1
Etiquettes												
Ville	0.94	0.90	**0.92**	0.88	0.84	0.86	0.95	0.85	0.90	0.67	0.50	0.57
Type de contrat	0.92	0.85	0.89	0.90	0.91	**0.91**	0.96	0.87	**0.91**	0.90	0.46	0.63
Formation	0.92	0.93	0.91	0.94	0.92	**0.95**	0.94	0.94	0.93	0.89	0.64	0.74
Expérience	0.77	0.81	**0.91**	0.78	0.84	0.87	0.83	0.85	0.86	0.84	0.70	0.76
Durée de l'expérience	0.97	0.96	**0.97**	0.96	0.96	0.96	0.99	0.99	**0.97**	0.60	0.36	0.45
Compétences techniques	0.84	0.75	0.80	0.77	0.76	0.76	0.76	0.86	**0.81**	0.69	0.90	0.78
Missions	0.86	0.90	0.88	0.91	0.87	0.89	0.93	0.91	**0.91**	0.82	0.64	0.72
Métier	0.96	0.93	**0.95**	0.95	0.92	0.93	0.99	0.88	0.93	0.86	0.38	0.52
Code postal	0.99	0.99	**0.99**	1.00	0.95	0.97	1.00	0.97	**0.99**	0.83	0.69	0.75
Salaires	0.93	0.91	**0.92**	0.93	0.90	0.91	0.91	0.93	**0.92**	0.88	0.80	0.83
Compétences générales	0.91	0.87	0.89	0.88	0.79	0.89	0.92	0.90	**0.91**	0.77	0.60	0.68
Moyenne	0.91	0.87	0.89	0.91	0.87	0.89	0.92	0.90	**0.91**	0.81	0.58	0.67

bi-LSTM affiche la meilleure performance en moyenne. De plus, les étiquettes "salaire", "code postal", "éducation" et "durée d'expérience" ont été identifiées avec précision par les trois modèles avec un minimum de 0.90, alors que la quantité de données d'entraînement sur ces étiquettes était relativement faible. Les séquences connexes étaient faciles à identifier grâce à leur position courante dans l'offre d'emploi. De plus, en ce qui concerne l'étiquette "compétences générales" proposée par le master, les performances étaient particulièrement prometteuses, notamment pour le CRF bi-LSTM. Il s'agit du sixième meilleur label avec un score de précision et de rappel de 0,92 et 0,90, respectivement. Nous notons également que pour les étiquettes "ville" et "métier", le modèle bi-LSTM a obtenu des performances inférieures en termes de rappel par rapport aux CRF et LSTM CRF. Cela peut s'expliquer par le fait qu'il nécessite plus de données en raison de sa caractéristique bidirectionnelle.

Il convient également de noter que lors de l'étiquetage des modèles CRF et CRF LSTM, les étiquettes "expérience" et "compétences techniques" ont été mélangées dans certaines offres d'emploi, comme pour la séquence "vous avez de l'expérience

dans le langage Python". "Le langage Python" peut être soit une expérience, soit une compétence. Ces deux modèles n'ont pas fait la distinction entre ces deux étiquettes, contrairement à la CRF bi-LSTM, qui avait un bon rappel sur ces étiquettes, mais une précision plus faible. La bi-directionnalité du bi-LSTM améliore la prise en compte des schémas organisationnels d'un texte brut et favorise donc les performances de ce modèle.

Nous pouvons conclure que la méthodologie proposée pour créer le corpus permet une analyse syntaxique des offres d'emploi de haute qualité. De plus, le modèle qui a fourni les meilleures performances sur les offres d'emploi est le CRF bi-LSTM.

Rappelons que notre méthodologie traite de l'hypothèse suivante : même pour une forme de structure incertaine, la qualité de l'extraction d'entités peut être améliorée en utilisant un étiquetage de séquences capable de prendre en compte la structure, ce qui était notre hypothèse (H). En plus de (H), nous supposons que DEEP est capable de gérer les défis spécifiques : (C1) la variabilité et l'incertitude de l'étiquetage manuel des séquences, (C2) l'ambiguïté entre les paires étiquette/séquence et (C3) l'évolution du vocabulaire. Les sections ci-après sont consacrées à la validation de ces hypothèses.

5.2.5.2 Validation de (H) : L'utilisation des schémas organisationnels améliore la qualité de l'extraction des entités

Notre travail part du principe que la prise en compte des schémas organisationnels pourrait améliorer l'extraction d'entités. Pour confirmer cette hypothèse, nous avons comparé un modèle d'apprentissage bi-LSTM qui est supposé prendre en compte les schémas organisationnels (en considérant le contexte de chaque séquence étiquetée), à un modèle SVM.

Le SVM prend en entrée les séquences et en sortie les étiquettes. Il ne considère pas les séquences étiquetées précédentes et suivantes pour capturer le contexte. Chacune des séquences et les étiquettes correspondantes sont considérées comme indépendantes les unes des autres.

La Table 5.3 présente la précision, le rappel et le F1 obtenus avec le SVM. Nous notons une différence significative dans les performances du SVM et du bi-LSTM. Le SVM a une précision moyenne de 0.81, ce qui est significativement plus faible que celle du CRF bi-LSTM (0.92) et des autres algorithmes (0.91). Nous pouvons également noter que pour les "compétences techniques", le SVM a généré moins d'incertitudes que pour l'étiquette "ville" qui peut prêter à confusion, puisque nous pouvons trouver la ville de la description du poste et la ville du siège de l'entreprise. Le SVM différencie à peine ces deux villes, à cause du contexte inconnu ; contrairement au bi-LSTM qui est capable de capturer le contexte et donc la structure. La performance significative du bi-LSTM confirme que DEEP et les modèles d'apprentissage sont capables de capturer les modèles organisationnels d'un texte simple.

5.2.5.3 Validation de C1 : Variabilité et incertitude de l'étiquetage manuel des séquences

Le premier défi de ce travail est la variabilité de l'annotation manuelle. Nous considérons que DEEP est capable de gérer ce défi. Notre expérimentation a mis en évidence le fait que **"A13. Création des règles d'annotation spécifiques au domaine"**, **"A14. Création de règles d'annotation générales"** et **"A2. Étiquetage manuel du corpus"** permet aux annotateurs d'associer certaines étiquettes aux séquences, même pour le vocabulaire spécifique au domaine. À cette fin, nous avons comparé une annotation manuelle basée sur toutes les activités proposées et une annotation manuelle sans les sous-activités **A13** et **A14**. Nous avons remarqué que l'annotation d'une offre d'emploi prenait entre 4 et 5 minutes lorsque l'annotateur avait à sa disposition la sortie des modèles organisationnels de **A11**, contre 6 à 7 minutes dans le cas contraire, comme nous pouvons le voir dans la Table 5.4. Ces activités sont essentielles pour éviter les incertitudes lors de l'annotation. Nous avons ensuite comparé une annotation manuelle prenant en compte toutes les activités proposées et une annotation manuelle sans validation de celles-ci. Nous avons noté des divergences entre les annotateurs, ce qui a donné lieu à un modèle d'apprentissage avec un score plus faible (F1= 0,54 avec le CRF bi-LSTM) que dans le DEEP complet. Notre validation finale a également obtenu le score le plus élevé de nos modèles, en particulier sur le score de rappel, confirmant que le modèle présente très peu d'ambiguïtés en raison des faibles incertitudes lors de l'annotation manuelle.

Table 5.4
Rapport du temps passé par chaque annotateur sans exécuter les activités A13 et A14 (en secondes)

	Annotation sans les sorties des activités A13 et A14	**Annotation sans les sorties des activités A13 et A14**
Nombre de documents/ Annotateurs	**30**	**30**
1	380	295
2	365	248
3	370	255

5.2.5.4 Validation de C2 : Amélioration de l'ambiguïté entre certaines paires étiquette/séquence

Comme mentionné dans la section 4, certaines offres d'emploi partagent un vocabulaire commun (par exemple, pour les étiquettes "missions" et "compétences techniques", les séquences associées peuvent partager un vocabulaire commun). Cela peut

donc conduire à des incertitudes lors de l'annotation et donc à des ambiguïtés dans le modèle d'apprentissage. Pour relever ce défi, nous avons d'abord introduit le **"A11. Définition des schémas organisationnels des textes"** dans la méthodologie afin de s'assurer que les annotateurs puissent facilement associer certaines étiquettes à une séquence grâce aux schémas organisationnels. Nous avons remarqué que l'annotation d'une offre d'emploi prenait 4 minutes lorsque l'annotateur disposait de la sortie de **A11**, contre 7 minutes dans le cas contraire. De plus, en incluant l'activité **A13.Création des règles d'annotation spécifiques au domaine"** nous avons noté une amélioration de la vitesse d'annotation et du score de notre modèle.

La deuxième validation a consisté à créer deux dictionnaires complets des phrases les plus courantes associées aux étiquettes "missions" et "compétences techniques". Nous avons reproduit l'approche "Dictionnaire de mots" et l'avons comparée au modèle d'indexation CRF bi-LSTM. Les scores présentés dans la Table 5.5 montrent une différence significative entre les deux approches et la validation de la nôtre. Il y a une différence de plus de 40% entre les deux approches. Nous pouvons conclure que DEEP et le choix de l'algorithme ont minimisé les incertitudes dans l'annotation des séquences qui partagent le même vocabulaire et n'ont pas conduit à une ambiguïté dans notre modèle.

5.2.5.5 Validation de C3 : prise en compte de l'évolution du vocabulaire

Nous supposons dans ce travail que la prise en compte des schémas organisationnels aidera à considérer l'évolution du vocabulaire. Pour évaluer la capacité de DEEP à prendre en compte l'évolution du vocabulaire, nous avons testé le modèle bi-LSTM sur une offre d'emploi datant de 19 ans. Cette offre contenait quatre séquences qui n'étaient pas présentes dans le corpus d'apprentissage supervisé. Dans cette offre, F1 = 0,82 et les séquences inconnues du système ont été étiquetées correctement. Par exemple, la séquence "fort esprit d'entreprise" a été correctement étiquetée comme "compétences générales". La même chose a été faite avec le modèle SVM. La séquence n'avait pas été associée à l'étiquette "compétences générales". Pour rendre l'expérience plus précise, nous avons effectué des évaluations supplémentaires en remplaçant une partie du vocabulaire connu (20 mots) de l'ensemble de test par un vocabulaire inconnu. Nous avons remarqué que les étiquettes associées aux séquences restaient les mêmes, même si le vocabulaire était inconnu. Nous avons effectué des expériences supplémentaires sur 60 offres d'emploi dans trois secteurs d'emploi inconnus par le modèle : - Travaux publics -Métiers des forces armées -Producteurs d'animaux. Le modèle a obtenu un score de $F1 = 0,89$ pour ces nouveaux secteurs. Ces validations ont confirmé que DEEP prend en compte l'évolution du vocabulaire.

5.2.6 Normalisation des séquences

5.2.6.1 Normalisation de type primitif

La normalisation de type primitif a été appliquée sur les séquences associées aux étiquettes suivantes : "Ville", "Formation", "Durée de l'expérience", "Code postal" et

Table 5.5
Comparaison des performances entre l'approche basée sur les règles utilisant le dictionnaire et le CRF bi-LSTM.

	Dictionary			**bi-LSTM CRF**		
Métriques	P.	R.	F1	P.	R.	F1
Étiquettes						
Compétences techniques	0.57	0.26	0.36	0.76	0.86	**0.81**
Missions	0.62	0.47	0.54	0.93	0.91	**0.91**

"Salaire". L'approche utilisée est à base de règles : expressions régulières et dictionnaires. Chaque étiquette s'est vu attribuer une norme commune. Dans la Table 5.6 nous pouvons voir le référentiel choisi pour chaque étiquette.

Table 5.6
Référentiel de type primitif utilisé pour chaque étiquette

Étiquettes	**Référence**
Ville	Règles typographiques de base [25]
Formation	Nomenclature du ministère de l'enseignement supérieur, de la recherche et de l'innovation
Durée de l'expérience	Interne (minimum, maximum, unité)
Code postal	Norme Postale AFNOR NF Z 10-011 [26]
Salaire	Interne [(minimum, maximum, unité), devise]

Des exemples de normalisation de certaines séquences pour chacune de ces étiquettes sont présentés dans la Table 5.7

5.2.6.2 Normalisation de type référence

La normalisation de type référence a été appliquée sur les séquences associées aux étiquettes suivantes : "Ville", "Formation", "Durée de l'expérience", "Type de contrat", et "Métier". L'approche utilisée est à base de règles : expressions régulières et dictionnaires pour les étiquettes "Ville", "Formation", "Durée de l'expérience", "Type de

Table 5.7
Exemples de séquences et de leur normalisation pour chaque étiquette

Étiquettes	Séquence	Séquence normalisée
Ville	paris	Paris
Formation	bac 5	Bac + 5
Durée de l'expérience	2 à 5 ans d'expérience 5 mois d'expérience minimum	(2, 5, ans) (5, 5, mois)
Code postal	54	54000
Salaire	35K euros Entre 10 et 11 euros l'heure	[(35 000, 35 000 , an), EUR] (10, 11 , heure), EUR

contrat". Chaque étiquette s'est vu attribuer un référentiel choisi par un expert lié au besoin de diffusion des offres d'emploi sur les canaux. Dans la Table 5.8, nous pouvons voir le référentiel choisi pour chaque étiquette. nous avons validé cette approche en utilisant un corpus normalisé de 200 offres d'emploi. Le taux de précisions des séquences normalisées est de 89%.

Dans la Table 5.9 sont présentés des exemples de séquences normalisées en référence.

Nous intégrons aussi une nouvelle étiquette qui est déduite à partir de l'offre d'emploi entière, qui est le secteur d'activités.

Le secteur d'activité est une information très importante pour décrire l'entreprise. Cette information est souvent implicite dans l'offre d'emploi et peut être déduite à partir de ce texte. De ce fait, l'approche nécessaire pour déduire le secteur est la normalisation de type référence. Nous allons donc attribuer à chaque offre d'emploi un secteur d'activités. Le référentiel qui a été choisi pour le secteur d'activité est celui du canal de diffusion

Pour l'étiquette "Métier" deux référentiels ont été choisi :

— Métier. Ce référentiel contient 153 classes de métiers

— Secteur. Ce référentiel contient 32 classes de secteurs de métiers.

L'approche à base de règles n'est pas possible pour cette étiquette puisque les séquences à attribuer à chaque classe ne sont pas d'ordre fini. Nous ne pouvons pas créer un dictionnaire ni utiliser des expressions régulières. De ce fait, une classification pour cette étiquette est nécessaire.

Table 5.8
Référentiel de type référence utilisé pour chaque étiquette

Étiquettes	Référentiel
Ville	- (Correction orthographique)
Type de contrat	CDI CDD Interim Stage Alternance Indépendant Intermittent Saisonnier Fonctionnariat Bénévolat
Formation	Sans diplôme CAP, BEP BAC Professionnel, Technique, Brevet de Technicien BTS, DUT, DEUG, DEUST Licence Maîtrise Master 2 Bac+6 et plus Doctorant Bac+9 et plus
Durée de l'expérience	Stage 0 à 1 an ← Débutant 2 à 4 ans ← Junior 5 à 9 ans ← Confirmé 10 ans et plus ← Senior

Approche de classification Les données utilisées pour entraîner le modèle sont différentes offres d'emploi extraites de canaux de diffusion. Nous avons entraîné le modèle sur 67 759 offres d'emploi classées. Nous avons utilisé une cross-validation 10-fold. L'algorithme utilisé est le bi-LSTM et la représentation vectorielle s'est faite à l'aide du modèle pré-entraîné CamemBERT.

Table 5.9
Référentiel utilisé pour chaque étiquette

Étiquettes	Séquence	Séquence normalisée
Ville	Paris	Ile de France
Formation	Bac + 5	Maîtrise Master 2
Durée de l'expérience	5 ans d'expériences 2 à 5 mois	Confirmé Débutant
Type de contrat	Contrat à durée indéterminée	CDI

Table 5.10
Exemple de séquences de métiers normalisées à l'aide du référentiel choisi

Séquence	Séquence normalisée
Technicien Maintenance Électronique	Technicien des méthodes
Comptable Fournisseurs Confirmé	Comptable
Technicien SAV itinérant	Technicien des méthodes
CONSEILLER DE VENTE	Chargé de clientèle
Responsable communication externe et marque	Chargé de communication

5.2.6.3 Résultats

Dans la Figure 5.9, nous pouvons visualiser le score F1 pour certaines classes du référentiel métiers choisi. Nous notons un score très différent selon les classes. Certaines classes dépassent les 90% de score F1. En revanche, certaines classes ne dépassent pas les 80%. Ceci est dû à la sous-représentativité de certaines classes dans les données d'apprentissage. En effet, pour certaines classes, les offres d'emploi les

représentant sont en nombre très faible, entraînant un taux de précision entre 50% et 80% et une moyenne de 75% pouvant être amélioré grâce à l'ajout de données. En revanche, certaines classes comme "Distribution Commerce de détail" sont des classes bien représentées dans les données d'apprentissage. Néanmoins, il existe 30% de faux positifs (offre d'emploi attribuée à cette classe, mais qui n'appartient pas à cette classe) dus au fut que la classe est très large pouvant regrouper plusieurs séquences de métiers sémantiquement différents. Pour ce type de classe, il est nécessaire de revoir le référentiel et créer de nouvelles classes afin de minimiser les ambiguïtés. Le score F1 moyen de l'ensemble des catégories pour le référentiel métiers est de 62 sur l'ensemble des classes.

Concernant le référentiel de secteur d'activités, de même que pour les métiers, certaines classes sont sous-représentées. Cependant, étant donné un plus petit nombre de classes, le score F1 moyen de l'ensemble des catégories pour le référentiel secteurs est de 89 sur l'ensemble des classes, ce qui reste plus élevé que pour le référentiel métiers. De plus, les données d'entraînement contiennent beaucoup plus de contexte sur le métier, les missions et les secteurs, ce qui entraîne une meilleure classification que pour les séquences de métier ayant en moyenne 56 Token (mots) maximum.

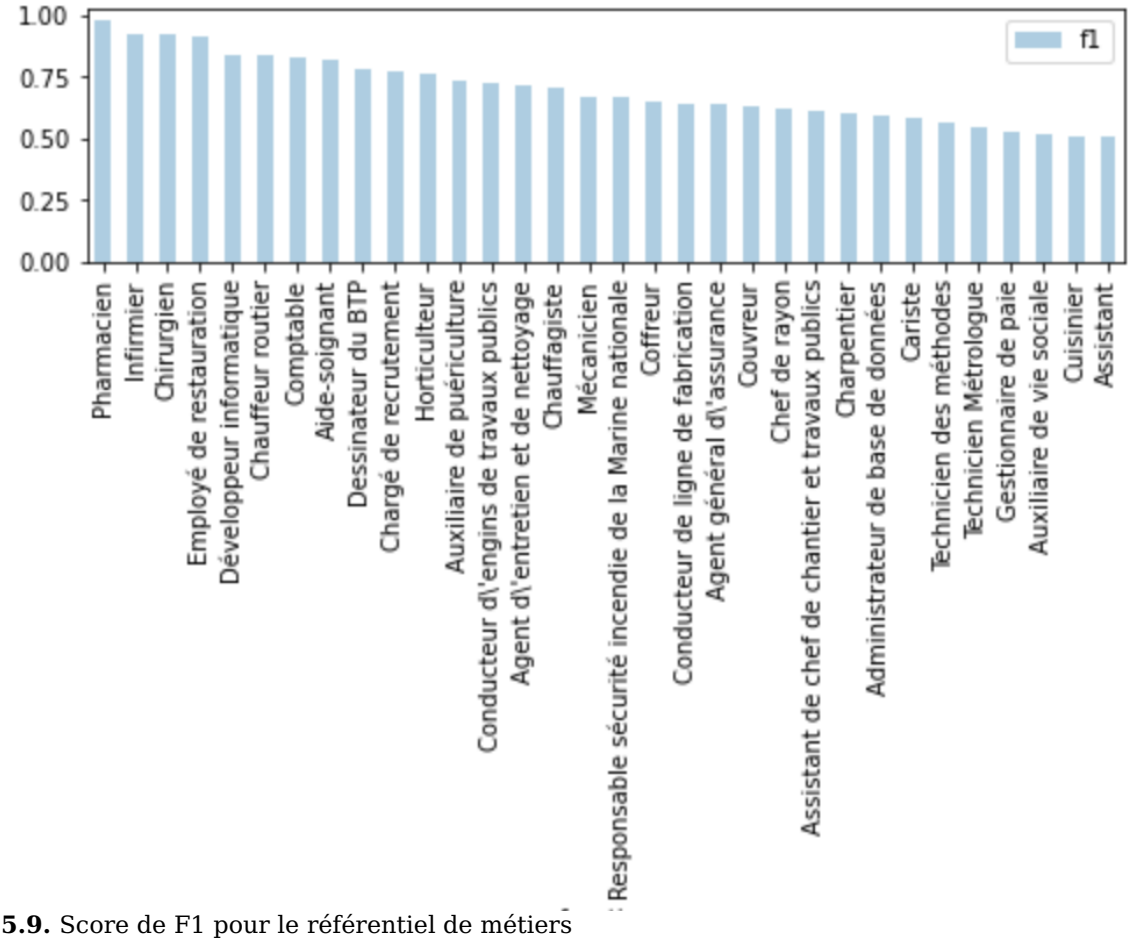


Fig. 5.9. Score de F1 pour le référentiel de métiers

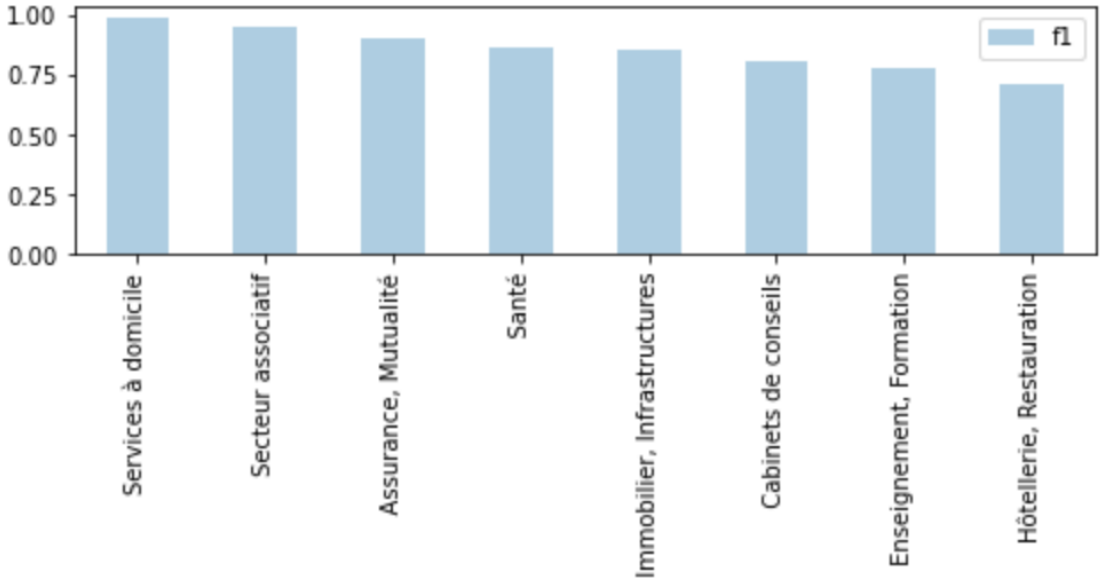


Fig. 5.10. Score de F1 pour le référentiel de secteurs

5.2.7 Conclusion

DEEP a été validée sur un corpus réel d'offres d'emploi. Sa mise en œuvre dans le domaine d'application des offres d'emploi, avec un expert, un master et trois annotateurs, a pris environ 300 heures. Comparé au temps d'implémentation requis par d'autres approches automatiques, DEEP peut sembler plus long. En revanche, cette approche permet d'aborder les différentes spécificités : (C1) variabilité et incertitude de l'étiquetage manuel, (C2) amélioration de l'ambiguïté entre certaines paires d'étiquettes/séquences et (C3) prise en compte de l'évolution du vocabulaire. Les différentes expériences ont montré que les étapes **"A13. Création de règles d'annotation spécifiques au domaine"**, **"A14. Création de règles d'annotation générales"** et **"A2. Étiquetage manuel du corpus"** sont essentielles pour éviter les incertitudes lors de l'annotation et pour relever le défi F1. Pour le défi F2, la sous-activité **"A13. Création des règles d'annotation spécifiques au domaine"** et **"A11. Définition des schémas organisationnels"** permettent à l'annotateur d'avoir moins d'incertitudes entre les étiquettes qui partagent un vocabulaire commun. Enfin, les expériences montrent que la prise en compte de la structure peut aider à associer le vocabulaire inconnu du modèle aux bonnes étiquettes.

Par conséquent, DEEP ne nécessite pas une mise à jour continue du vocabulaire, contrairement aux autres approches, ce qui permet de gagner du temps. Remarquons qu'un corpus plus petit peut être étiqueté manuellement, puis une annotation semi-automatique peut être effectuée, ce qui entraîne une diminution du temps requis. Dans la suite, nous chercherons à améliorer l'extraction d'entités en incluant la relation d'entité et l'apprentissage actif pour améliorer notre modèle.

Nous avons proposé de compléter cette contribution en normalisant les séquences étiquetées. Pour cela, nous avons utilisé plusieurs approches : à base de règles et

classification. Les approches à base de règles ont un taux de précision de 89% quant aux approches par classification, elles nécessitent plus de données pour améliorer le score. En effet, les données ne sont pas distribuées uniformément entraînant un faible score pour certaines classes. Dans la suite de nos travaux, nous souhaitons ajouter des données pour les classes sous-représentées et compléter cette approche à l'aide d'un calcul de distance sémantique. Nous souhaitons aussi évaluer d'autres approches de classification telle que la classification hiérarchique qui utilise différents niveaux d'informations. D'autant plus que les référentiels de métiers et compétences sont des informations hiérarchisées, il est important d'évaluer des approches de classification qui utilisent ce type d'information.

5.3 Appariement de deux textes rédigés en langage naturel

Étant donné les limites des travaux de la littérature sur l'appariement de deux textes comme (1) l'utilisation du contenu textuel entier pour effectuer la similarité, (2) l'utilisation d'approches lexicales mettant de côté la sémantique du contenu, (3) l'utilisation d'approches sémantiques nécessitant du temps ou entraînant une opacité du système, notre contribution s'est basée sur l'hypothèse que l'appariement par type d'information pourrait améliorer la qualité du système. Notre contribution s'appuie sur trois modules "Extraction d'entités et normalisation", "Représentation vectorielle" et "Calcul de la distance sémantique". Dans la suite de ce chapitre, nous allons présenter les démarches pour l'expérimentation de l'approche proposée. Dans un premier temps, nous allons évaluer l'approche de représentation vectorielle appliquée au domaine du e-recrutement et dans un second temps, nous allons évaluer l'approche générale sur des paires d'offres d'emploi et CV annotés positivement ou négativement par des experts.

5.3.1 Extraction d'entités et normalisation

Notre contribution utilise l'extraction d'entités et la normalisation pour identifier, extraire et normaliser les séquences. De plus, nous avons proposé une approche de représentation vectorielle basée sur un modèle pré-entraîné auquel nous avons ajouté un corpus de paires similaires et non similaires sémantiquement d'un domaine spécifique. Dans la suite de ce chapitre, nous allons expérimenter notre approche sur le domaine du recrutement et plus précisément sur les offres d'emploi et les CV qui sont des documents textuels rédigés en langage naturel.

Notre contribution base son hypothèse sur le fait que l'identification et l'extraction d'informations de l'offre et du CV permettraient d'améliorer les résultats d'appariement. Pour cela, nous avons utilisé le modèle présenté dans le chapitre précédent. Cette extraction nous a permis d'appliquer une normalisation sur les séquences identifiées et extraites. Dans l'annexe C et D nous pouvons visualiser les résultats de cette étape pour une offre d'emploi et pour un CV.

Tout d'abord les offres d'emploi et les CV sont reçus dans différents formats. Des traitements différents sont appliqués selon le type de format :

— Pdf, docx, doc sont des fichiers pour lesquels des librairies en langage python sont utilisées pour en extraire le texte.

— Png, jpeg, etc. sont des formats images. Ce type de document est plus difficile à traiter puisque le texte n'est pas encodé. Une technique très connue est utilisée l'océrisation qui dérive de l'abréviation OCR : Optical Character Recognition. Il s'agit de la reconnaissance optique des caractères. Techniquement, il s'agit du traitement d'une image sur laquelle un modèle de reconnaissance de caractères est utilisé pour en extraire le texte.

Cette transformation permet d'avoir le texte brut du fichier pour pouvoir appliquer le module d'extraction d'entités et de normalisation dans le module suivant. Ce module permet aussi d'éviter toute discrimination liée au sexe, à la nationalité, aux écoles etc. En effet, l'identification de l'information permet ainsi d'anonymiser les CV en permettant l'identification et l'extraction de l'identité, entraînant ainsi la considération dans l'appariement uniquement des informations non discriminante :

— le métier,

— l'expérience,

— la formation,

— les compétences.

La normalisation quant à elle permet d'éviter la discrimination liée aux lieux de scolarité. Par exemple un diplômé de l'école ENSGSI voit son diplômé être substitué par « Ecole d'ingénieur Bac + 5 ». La normalisation permet aussi de ne pas utiliser le lieu d'habitation, mais une géolocalisation permettant ainsi au recruteur de filtrer selon une distance maximale.

5.3.2 Représentation vectorielle

5.3.2.1 Choix du modèle pré-entraîné

Dans notre contribution, nous avons proposé une approche qui enrichit les modèles pré-entraînés pour les domaines applicatifs dont le vocabulaire est spécifique. Le recrutement est un domaine qui évolue et dont le vocabulaire évolue aussi. L'alimentation du modèle pré-entraîné nécessite tout d'abord le choix de ce dernier. Ce choix s'est fait sur la base des critères suivants proposés dans le chapitre de contribution 3 :

1. Un corpus d'entraînement français ou multi-lingue. Pour cela plusieurs choix se sont offerts à nous. Nous les avons présentés dans le chapitre de contribution :

   — distiluse-base-multilingual-cased-v2,

   — stsb-bert-base,

   — paraphrase-multilingual-MiniLM-L12-v2.

   Ces trois modèles sont entraînés sur des corpus multi-lingues de tailles très larges. Le choix du modèle se fera donc à l'étape suivante.

2. Un modèle ayant la meilleure précision sur un vocabulaire d'un domaine spécifique. Pour cela nous avons testé les trois modèles avec un vocabulaire issu d'offres d'emploi et de CV. Quelques exemples de paires de séquences et leur score de distance sémantique pour chaque modèle est présenté dans la Table 5.11 pour les métiers et dans la Table 5.15 pour les compétences.

Table 5.11
Paires de séquences de métiers et leurs distances sémantiques par modèle

Séquence A	Séquence B	Distance sémantique	Modèle
Chauffeur	Automobiliste	0.8226	distiluse-base-multilingual-cased-v2
Chauffeur	Automobiliste	0.2680	stsb-bert-base
Chauffeur	Automobiliste	0.8301	paraphrase-multilingual-MiniLM-L12-v2
Magasinier	Cariste	0.5955	distiluse-base-multilingual-cased-v2
Magasinier	Cariste	0.4428	stsb-bert-base
Magasinier	Cariste	0.6826	paraphrase-multilingual-MiniLM-L12-v2
Spectacle	Décoriste	0.5860	distiluse-base-multilingual-cased-v2
Spectacle	Décoriste	0.4286	stsb-bert-base
Spectacle	Décoriste	0.5391	paraphrase-multilingual-MiniLM-L12-v2

Les résultats de cette première évaluation des modèles ont montré que le modèle qui a les résultats les plus satisfaisants (se rapprochant de notre corpus de validation (le corpus Gold)) est le **paraphrase-multilingual-MiniLM-L12-v2**.

5.3.2.2 Pré-entraînement sur un vocabulaire spécifique

Nous avons proposé une automatisation de la création du corpus de données qui est constitué de paires de séquences et leurs distances sémantiques afin d'alimenter le modèle pré-entraîné choisi à l'étape précédente **paraphrase-multilingual-MiniLM-**

Table 5.12
Paires de séquences de compétences et leurs distances sémantiques par modèle

Séquence A	Séquence B	Distance sémantique	Modèle
Back-end et front-end	Fullstack	0.2866	distiluse-base-multilingual-cased-v2
Back-end et front-end	Fullstack	0.3002	stsb-bert-base
Back-end et front-end	Fullstack	0.4314	paraphrase-multilingual-MiniLM-L12-v2
En charge d'organiser les réunions et les animer	Aptitude à l'animation d'une équipe pendant les réunions	0.5147	distiluse-base-multilingual-cased-v2
En charge d'organiser les réunions et les animer	Aptitude à l'animation d'une équipe pendant les réunions	0.6192	stsb-bert-base
En charge d'organiser les réunions et les animer	Aptitude à l'animation d'une équipe pendant les réunions	0.6869	paraphrase-multilingual-MiniLM-L12-v2

L12-v2. La préparation du corpus s'est faite en deux étapes. Tout d'abord, la **sélection des séquences**, sur lesquelles nous souhaitons effectuer l'entraînement. Pour cela, nous avons choisi plusieurs sources de données dans le domaine du recrutement pour construire les paires de séquences et leurs distances sémantiques :

1. Les synonymes des métiers issus du référentiel de Météojob (Canal de diffusion des offres d'emploi). Ce canal de diffusion a l'avantage de diffuser les offres d'emploi et stocker les synonymes de chaque métier. Un exemple des données utilisées à partir de cette source de données est présenté dans la Table 5.13. Ce canal de diffusion a été choisi pour la multiplicité des catégories de métiers qu'il propose et la structuration des données. À chaque catégorie de métiers lui est associé des métiers et chaque métier a ses compétences et ses synonymes. De ce fait pour les métiers issus de catégorie différentes la distance sémantique sera

nulle et pour les métiers issus de la même catégorie la distance sera de 1. De même pour les compétences. Nous pouvons voir un exemple dans la Figure 5.11

Table 5.13
Exemple de données issues de Meteojob pour créer les paires de séquences

Catégorie de métiers	**Métier**	**Synonymes**	**Compétences**
Santé	Médecin	Gastro-entérologue ; Rhumatologue ; Ophtalmologue etc.	- Recevoir et soigner les patients - Faire des consultations quotidiennes - Suivre la santé de ses patients
SSI, Editeurs de logiciel, services informatiques	Développeur informatique	Analyste-programmeur ; Réalisateur en informatique ; Analyste fonctionnel ; Analyste réalisateur ; Ingénieur logiciel ; Ingénieur d'études logiciel ; Développeur, etc.	- Étude des besoins client en imaginant la meilleure solution technique et détermine les étapes de fonctionnement du programme - Développement en produisant des lignes de code en suivant le cahier des charges - Effectuer les essais pour vérifier la bonne marche des fonctionnalités du logiciel. Il détecte les éventuelles erreurs.

2. Le référentiel métiers FigaroClassified [27] construit à partir de 2 principales arborescences :
   (a) Les catégories de métier sont réparties en trois niveaux allant du métier le plus général au plus précis.

27. Leader français du marché des annonces classées sur internet

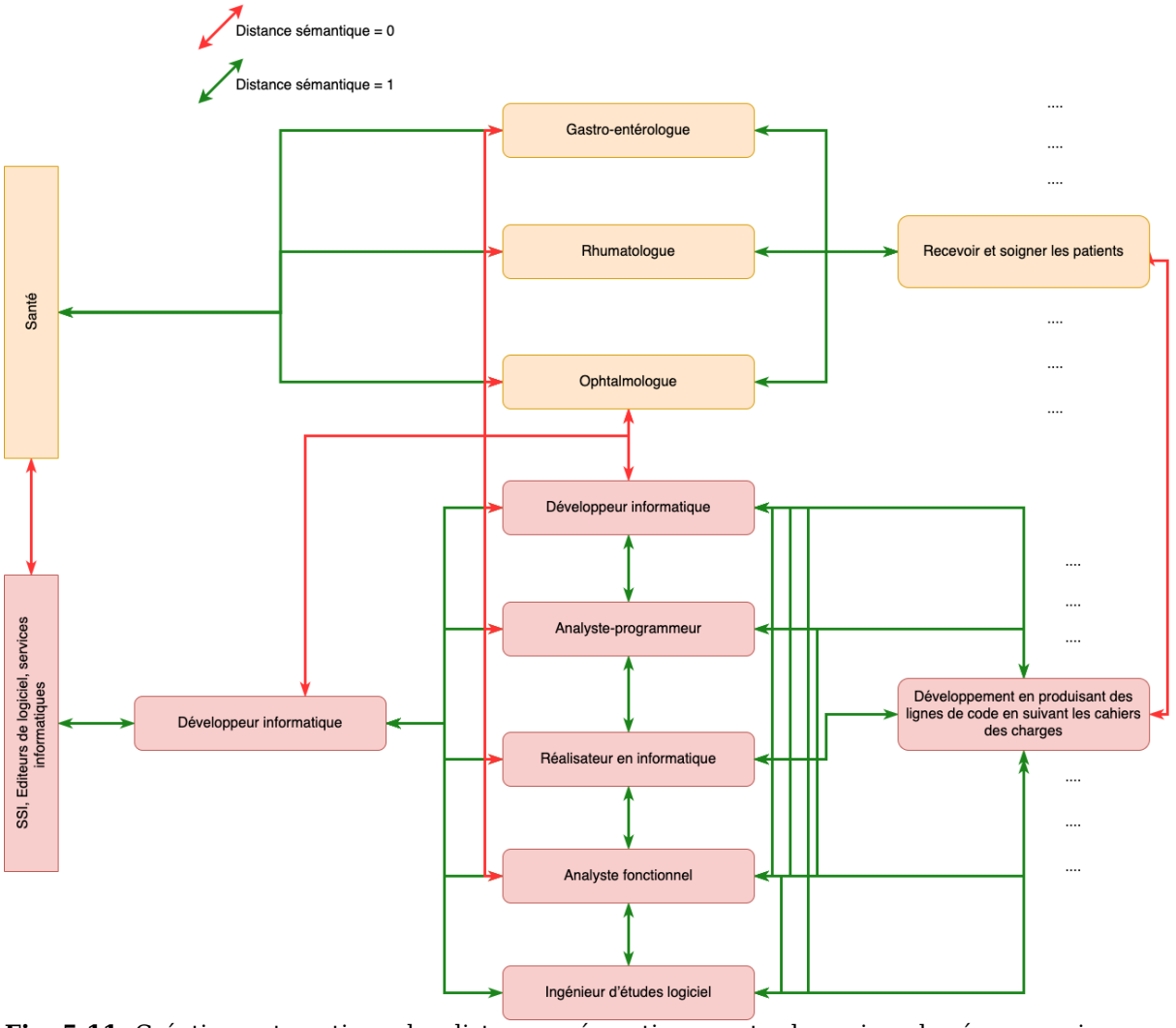


Fig. 5.11. Création automatique des distances sémantiques entre les paires de séquences issues du référentiel métiers

(b) Les secteurs d'activité sont répartis en deux niveaux allant du secteur plus général au plus précis.

Ces deux arborescences ont été utilisées pour construire les paires de séquences et leurs distances sémantiques.

Les distances sémantiques ont été attribuées comme expliqué dans le chapitre 3. Une distance sémantique de 1 a été attribuée :

— à chaque appellation de métier et ses synonymes ou autres appellations.

— à chaque métier et ses compétences

Une distance sémantique de 0.5 a été attribuée :

— aux métiers appartenant à la même catégorie par exemple kinésithérapeute ostéopathe et médecin.

Une distance sémantique de 0 a été attribuée :

— aux métiers n'appartenant pas à la même catégorie de métiers par exemple kinésithérapeute ostéopathe ou médecin qui font partie de la catégorie santé et développeur informatique, programmeur, etc. qui font partie de la catégorie SSI.

— aux compétences techniques des métiers appartenant à des catégories de métiers différentes.

Cette automatisation nous a permis d'avoir 150 000 paires de séquences ayant une distance sémantique positive, 170 000 paires de séquences ayant une distance sémantique nulle et 30 000 paires de séquences ayant une distance sémantique de 0.5. Dans le cadre de ces expérimentations, nous avons souhaité nous concentrer sur deux référentiels spécifiques à des métiers dans un premier temps pour évaluer si les distances attribuées automatiquement sont cohérentes sur ce vocabulaire après l'entraînement d'un modèle.

Le corpus de données après création a été évalué sur 10% du corpus de données de départ. Le modèle a été entraîné avec les paramètres suivants :

— Dimension du vecteur : 128

— Longueur maximale de la séquence : 256

— Fonction d'activation : Relu ; Fonction qui permet d'effectuer un filtre sur les données d'entrée en ne laissant passer que les valeurs positives dans les couches suivantes du réseau de neurones.

Le modèle a été entraîné sur 90% du corpus de données et évalué sur 10%.

Le modèle a été évalué en calculant la corrélation de Spearman et de Pearson par rapport aux corpus de test (présentés dans le chapitre 3). Les résultats du nouveau modèle entraîné sur le corpus de données préparées sont présentés dans la Figure 5.12. Comme nous pouvons le voir dans la figure, la corrélation augmente au fur et à mesure de l'entraînement, entraînant un score de 0.92 pour la corrélation de Pearson, et plus de 0.81 pour celle de Spearman.

Rappelons que la corrélation de Spearman est simplement celle de Pearson utilisant les rangs (statistiques d'ordre) au lieu des valeurs numériques réelles. Dans notre cas, la relation entre les distances prédites et réelles est linéaire, mais peu monotone. De

ce fait, la corrélation de Spearman est moins élevée que la corrélation de Pearson. Ceci n'est pas problématique puisque la corrélation de Pearson est plus importante puisqu'elle représente plus les valeurs numériques réelles. Nous pouvons en déduire la qualité du modèle sur le vocabulaire issu des deux référentiels et la cohérence des distances attribuées automatiquement.

Pour compléter notre évaluation, nous avons comparé la distance entre des séquences lexicalement opposées, mais sémantiquement proches dans les Tables 5.18 et 5.15. Ces tables montrent que notre modèle est capable de mieux représenter les séquences de façon à mieux considérer la sémantique par rapport aux modèles de départ entraînés à partir de données non spécifiques au domaine du recrutement.

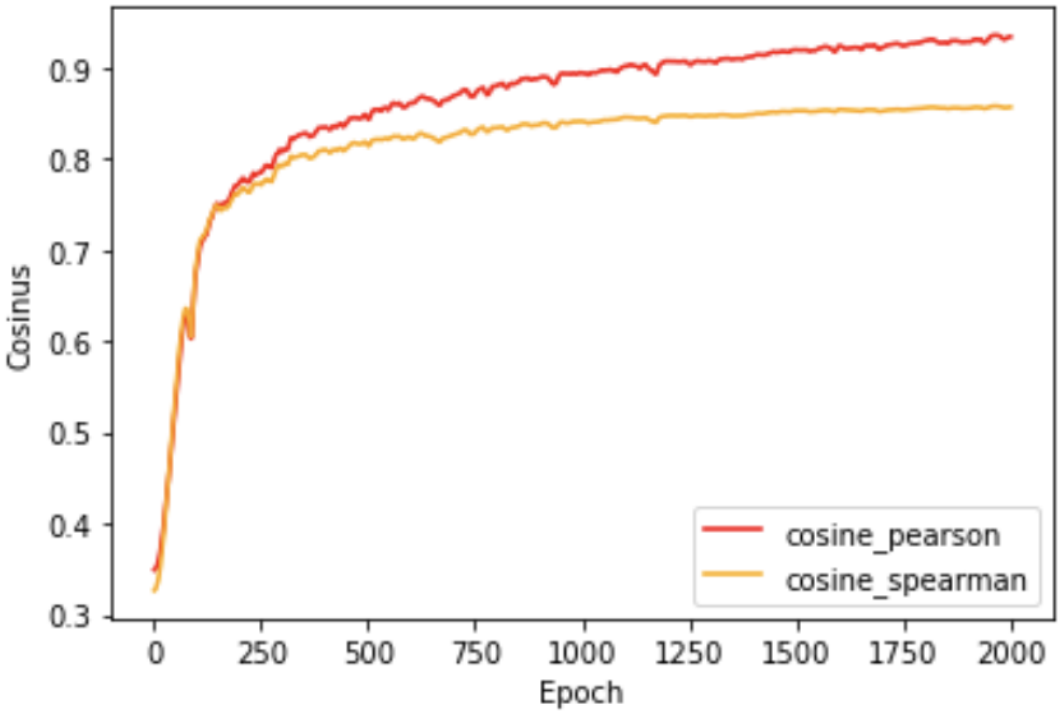


Fig. 5.12. Évolution de la corrélation entre la distance réelle entre les séquences et la distance prédite par le modèle pendant l'entraînement

5.3.3 Méthode d'évaluation de l'approche d'appariement

Afin d'évaluer l'approche d'appariement proposée basée sur la distance sémantique par type d'information, nous avons utilisé un corpus annoté de CV et offre d'emploi. Un expert en ressources humaines a attribué un score de 1 si le CV répond aux attentes du recruteur par rapport à une offre d'emploi et -1 sinon. Ce corpus d'offres et CV est issu d'une vraie campagne de recrutement pendant laquelle le recruteur a diffusé des offres d'emploi en intérim et a reçu des CV. Ce corpus contient :

Table 5.14
Paires de séquences de métiers et leurs distances sémantiques par modèle

Séquence A	Séquence B	Distance sémantique - paraphrase-multilingual-MiniLM-L12-v2	Distance sémantique - Nouveau modèle	Distance lexicale - TF-IDF
Chauffeur	Automobiliste	0.8301	1.00	0
Magasinier	Cariste	0.6826	0.99	0
Spectacle	Décoriste	0.5391	0.5735	0

Table 5.15
Paires de séquences de compétences et leurs distances sémantiques par modèle

Séquence A	Séquence B	Distance sémantique - paraphrase-multilingual-MiniLM-L12-v2	Distance sémantique - Nouveau modèle	Distance lexicale - TF-IDF
Back-end et front-end	Fullstack	0.4314	0.6080	0
En charge d'organiser les réunions et les animer	Aptitude à l'animation d'une équipe pendant les réunions	0.6869	0.9521	0

— 1099 paires d'offres d'emploi et CV ; Pour chaque CV et offre, nous avons un score de 1 si le CV est pertinent pour l'offre et -1 sinon,

— 456 offres,

— 771 CV,

— 588 paires, c'est-à-dire, 588 CV sont pertinents pour les offres d'emploi correspondantes,

— 511 paires sont négatives, c'est-à-dire, 511 CV ne sont pas pertinents pour les offres d'emploi correspondantes.

Nous considérons qu'une distance sémantique entre l'offre et le CV supérieure à 0.7

représente une paire offre/CV positive, elle est négative dans le cas contraire. Ce seuil a été choisi par défaut avec l'aide d'un recruteur (il peut être modifié selon le besoin du recruteur). Les résultats présentés dans la Table 5.16 montre que l'approche a de meilleurs résultats pour les paires négatives que pour les paires positives. Après discussion avec les recruteurs, il est plus important d'éliminer les candidats n'ayant aucune correspondance avec l'offre, que remonter des candidats potentiellement bons. Donc, l'approche proposée permet de mettre de côté les CV qui ne correspondent pas à l'offre, permettant ainsi de laisser la chance aux candidats potentiellement bons pour une offre. Ceci, permet de faire gagner du temps au recruteur, lui laissant ainsi plus de temps pour les entretiens.

Table 5.16
Paires de séquences de compétences et leurs distances sémantiques par modèle

	Paires positives	**Paires négatives**
Corpus d'évaluation	588	511
Approche par étiquetage de séquences et normalisation	390	480
Précision	65 %	94%

5.3.4 Conclusion

L'objectif de nos travaux était, dans un premier temps, de profiter de la richesse des informations contenues dans un texte afin d'effectuer un appariement appliqué sur les informations les plus utiles d'un texte. Et ensuite, être en mesure d'expliquer sur quel ensemble d'informations s'est basé le score de similarité. De ce fait, notre contribution repose sur l'hypothèse (QR2-H) que (1) l'appariement entre étiquettes et séquences extraites des textes permettrait d'éviter l'opacité du système et (2) la représentation vectorielle en utilisant BERT-sentence permettrait d'améliorer la considération de la sémantique pour calculer la distance entre les séquences. Nous avons proposé une approche de création de corpus de données automatique pour l'entraînement d'un modèle capable de calculer la distance sémantique entre les séquences dans le domaine du recrutement. Nous avons créé le corpus en nous basant sur des référentiels métiers et de compétences existantes. L'évaluation nous a permis de valider l'hypothèse que la représentation vectorielle en utilisant BERT-sentence permet d'améliorer la considération de la sémantique pour calculer la distance entre les séquences. De plus, nous avons évalué notre approche d'appariement basée sur l'appariement par type d'information. Nous avons constaté la facilité d'attribuer un score par type d'in-

formation aidant ainsi le recruteur à attribuer un degré d'importance pour chaque type d'information et aussi pouvoir expliquer le score final.

En revanche, dans les travaux qui ont été faits, nous avons remarqué la nécessité de ré-entraîner un modèle pour inclure les nouveaux vocabulaires étant donné l'évolution. De ce fait, nous souhaitons dans nos futurs travaux pouvoir intégrer une automatisation de cette tâche.

5.4 ADE^2 : un système d'Aide à la Décision dans un Environnement Évolutif

5.4.1 Représentation et formulation du problème

Rappelons que les questions industrielles du partenaire entreprise sont les suivantes :

— (QI-1) Comment choisir les canaux de diffusion optimaux par rapport à une offre ou une catégorie d'offres ?

— (QI-2) Comment optimiser le budget alloué sur les canaux pour répondre aux objectifs du recruteur ?

Nous proposons de répondre à ces questions industrielles en appliquant ADE^2 le système d'aide à la décision présenté dans le chapitre 4. Les caractéristiques du système de e-recrutement sont les suivantes :

— le décideur est le recruteur,

— la décision du décideur est le choix d'un ensemble de canaux de diffusion,

— le recruteur souhaite atteindre plusieurs objectifs en choisissant les canaux de diffusion, par exemple, maximiser le nombre de conversions et minimiser le nombre de clics,

— le recruteur prend une décision en fonction d'un profil souhaité décrit dans l'offre d'emploi,

— le contexte de décision est l'offre d'emploi qui est caractérisée par un contenu textuel rédigé en langage naturel,

— les paramètres sont des indicateurs de performance qui évoluent dans le temps et dans un environnement incertain,

— le recruteur peut avoir un ou plusieurs objectifs.

Les caractéristiques de ce système entraînent les verrous (définis dans l'état de l'art) : (QR3-C1) L'environnement est incertain ; (QR3-C2) L'environnement est évolutif entraînant l'évolution dans le temps des paramètres et donc un problème non stationnaire ; QR3-C3) Les objectifs du recruteur à atteindre sont multiples ; (QR3-C4) L'offre d'emploi rédigée en langage naturel ;

Nous définissons dans le contexte d'expérimentation pour la suite du document :

— Un item q_j comme un canal de diffusion, par exemple Indeed.

— Un contexte de décision D_i comme l'offre d'emploi qui est un document rédigé en langage naturel représentant le profil recherché par le recruteur.

— La variable de décision z_{D_i,q_j} qui est égale à 1 si le canal q_j pour l'offre d'emploi D_i est pertinent ; 0 sinon.

— Un ensemble d'objectifs comme la maximisation du nombre de conversions et la minimisation du nombre de clics.

— Une contrainte, comme un budget maximal de 800 euros à ne pas dépasser pour la diffusion des offres d'emploi (ou de 200 euros par période d'une semaine sur quatre périodes au total).

— Les paramètres, comme les indicateurs de performance des canaux de diffusion.

Dans le cadre de la validation des objectifs des recruteurs (gain de temps et d'argents), nous avons choisi de comparer le choix des canaux de diffusion par un manager des campagnes pour une offre d'emploi d'"Aide à domicile" et les recommandations à la sortie d'ADE^2.

5.4.2 Module de traitement de l'information

5.4.2.1 Extraction d'entités et normalisation des données textuelles

Le recruteur a à sa disposition des offres d'emploi qui représentent son contexte de décision. Ces offres d'emploi contiennent un ensemble d'informations qu'il est important d'identifier et extraire pour interpréter le contexte de décision. Pour cela, la méthodologie DEEP d'extraction et de normalisation permet de répondre au verrou du contexte de décision rédigé en langage naturel. Pour cela, nous avons utilisé le module d'extraction et de normalisation présenté dans le chapitre 2 et expérimenté dans le chapitre 5.2.

5.4.2.2 Uniformisation des données

Cette étape consiste à uniformiser les données issues de différentes sources. Après chaque diffusion, les évènements (clics, envois de CV, vues, etc.) sont stockés dans le système (à travers un système de tracker[28]). En revanche, chaque canal est une source de données différente. Ces évènements sont stockés différemment par chaque canal, entraînant la nécessité d'uniformiser les données afin de les utiliser dans les étapes suivantes. Pour cela :

1. un modèle de données pour stocker les évènements a été créé. Ce modèle contient les caractéristiques de l'offre d'emploi (issu du module d'extraction d'entités et normalisation des données textuelles), le canal de diffusion, les dates et les indicateurs de performance à cette date pour l'offre d'emploi sur le canal de diffusion (sur lequel a eu lieu la diffusion). Le modèle de données contient plus précisément les colonnes suivantes : "Nom de l'entreprise", "Métier", "Durée de l'expérience", "Type de contrat", "Niveau de formation", "Ville", "Département", "Secteur", "Canal", "Date", "Action"[29] (Clics, Vues, Candidature, etc.), "Valeur de l'action"[30], "Budget Total", "Devise"[31].

28. Système qui récupère en temps réel les évènements qui ont lieu sur une publicité ou une offre d'emploi
29. l'action est un clic, une vue, un envoi de CV, etc.
30. La valeur de l'action est le nombre associé à l'action
31. La devise utilisée par le canal : euros, dollar, etc.

2. une table de correspondance entre les modèles des différentes sources de données et le modèle de donnée final a été créée.
3. les données de différentes sources ont été transformées, selon le modèle de données proposé.

5.4.2.3 Création des classes sources

La création des classes sources utilise les données uniformisées résultant du module de traitement d'informations. Rappelons que les classes sources sont fondée à partir du regroupement des offres d'emploi sémantiquement proches. Ce module a pour intérêt de :

1. considérer la sémantique des offres d'emploi,
2. pallier le manque de données,
3. créer des classes qui faciliteront la considération de nouvelles offres d'emploi.

Pour ce faire, l'approche proposée est composée des étapes suivantes :

— regroupement des contextes similaires. Dans le cas du recrutement, il s'agit du regroupement des offres d'emploi sémantiquement proches,

— agrégation des paramètres évoluant dans le temps. Il s'agit des indicateurs de performance. Les séries chronologiques de chaque offre d'emploi sur chaque canal sont agrégées lorsque les offres d'emploi sont similaires.

Pour cette étape, nous avons appliqué l'algorithme 3 présenté dans la contribution du chapitre 4. Une figure pour schématiser ce sous-module est présentée dans la Figure 5.13.

Cette étape nous a permis d'être en mesure (1) de considérer les offres d'emploi lexicalement différentes, mais sémantiquement proches, (2) d'avoir plus de données pour être capable de capturer dans le module d'apprentissage, les tendances. Nous pouvons par exemple, voir dans la Table 5.17 que pour trois offres sémantiquement proches, leur regroupement dans une même classe permet de générer une série chronologique plus représentative des fluctuations et des tendances sur un canal pour des métiers appartenant à la même classe.

5.4.2.4 Préparation de la classe cible

La classe cible dans notre contexte est la classe qui est associée à l'offre d'emploi pour laquelle le recruteur souhaite se voir attribuer un choix de canaux de diffusion qui répond à ses besoins. Le recruteur rentre dans le système les trois informations suivantes :

— L'offre d'emploi (ou un ensemble d'offres d'emploi)

— Le budget, qui est un paramètre de la fonction de contrainte.

— Les objectifs à atteindre.

Table 5.17
Création d'une classe source regroupant deux offres d'emploi
Offre d'emploi diffusée séparément

Offre d'emploi (contexte de décision)	Canal (item)	Semaine de l'année	Indicateurs de performance
AUXILIAIRE DE VIE SOCIALE F/H Paris 17 pour un CDI	Adzuna	25 26 27 28 29 30 31	✓ ✓ ✓ ✓ ✓ ✗ ✗
Aide à domicile F/H Paris pour un Contrat à durée indéterminée	Adzuna	25 26 27 28 29 30 31	✗ ✗ ✗ ✗ ✗ ✗ ✗
Assistant de vie F/H Paris	Adzuna	25 26 27 28 29 30 31	✗ ✗ ✗ ✓ ✓ ✓ ✓

Classe source regroupant les offres d'emploi ci-dessus

Classe source (contexte de décision normalisé)	Canal	Semaine de l'année	Indicateurs de performance
Service à la personne - CDI - Île-de-France - Toute expérience	Adzuna	25 26 27 28 29 30 31	✓ ✓ ✓ ✓ ✓ ✓ ✓

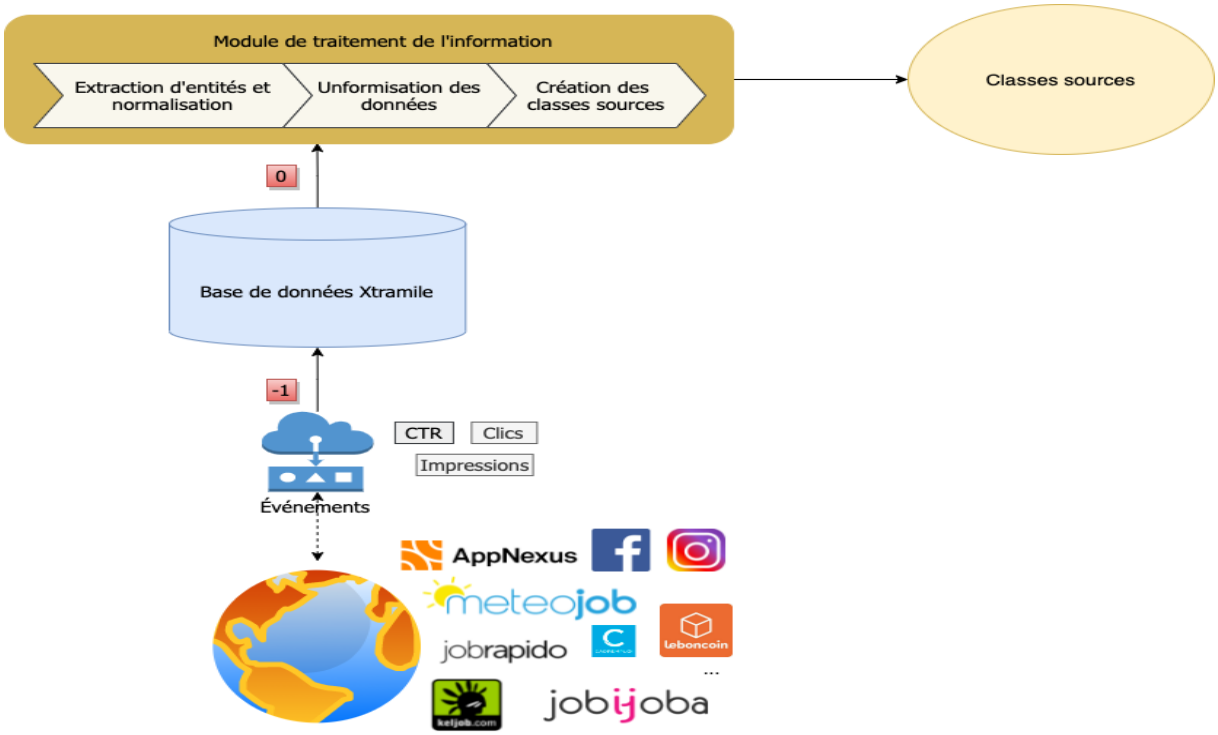


Fig. 5.13. Module de création des classes sources

Ces informations sont stockées dans la base de données et ensuite envoyées au sous-module d'extraction d'entités et normalisation des données textuelles comme nous pouvons le voir dans la Figure 5.14. Il en résulte une offre d'emploi structurée et normalisée, ce qui permet d'appliquer l'algorithme de recherche de la classe source la plus similaire présenté dans le chapitre 4. Pour rappel, cet algorithme utilise la fonction d'appariement entre deux textes rédigés en langage naturel présentée dans le chapitre 3 et expérimentée dans le chapitre 5.3.

Ce module permet d'attribuer une classe source à la classe cible. Pour notre étude, le recruteur souhaite des recommandations pour une offre d'emploi d'"Aide à domicile". Le système Xtramile n'a jamais traité auparavant ce type d'offre d'emploi d'"Aide à domicile" . En revanche, étant donné le module d'extraction et de normalisation, cette offre d'emploi a été associée à la classe source de Service d'aide à la personne et s'est vue attribuer la série chronologique liée à celle-ci. Celle-ci est utilisée dans le module d'apprentissage ci-dessous pour faire une prévision des indicateurs de performance : clics et conversions des fonctions objectifs du recruteur.

5.4.3 Module d'apprentissage

5.4.3.1 Préparation du corpus d'apprentissage

Le module d'apprentissage a pour objectif de faire une prévision des séries chronologiques pour les indicateurs de performance (paramètres des objectifs du recruteur). Dans ce chapitre d'expérimentation, nous allons appliquer ce module d'apprentissage sur le nombre de clics et le nombre de conversions (CV, inscription, etc.). L'étape de

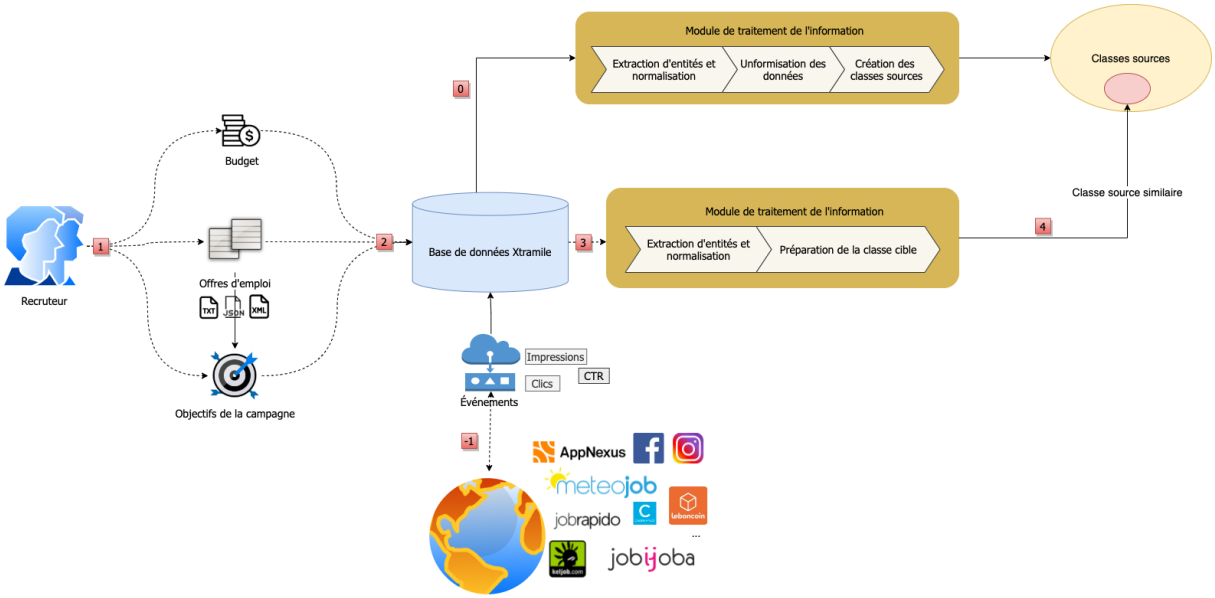


Fig. 5.14. Module de préparation de la classe cible

préparation du corpus d'apprentissage consiste à d'abord encoder les données catégoriques comme le type de contrat, le métier, etc. en valeurs numériques. Par la suite, nous avons transformé les données résultant du module précédent en données d'apprentissage supervisé. La fonction est décrite dans le chapitre 4 et l'algorithme en python est présenté dans l'Annexe F.

5.4.3.2 Application de l'algorithme d'apprentissage

L'architecture du module d'apprentissage est similaire à celle présentée dans le chapitre 4. Voici les paramètres de chaque couche de l'algorithme CNN-LSTM :

- couche d'enveloppement temporel (Time Distributed Layer) contient :
  - une couche de convolution d'une dimension (1D-CNN), avec comme activation Relu,
  - une couche de Pooling de dimension 1 et de taille 2,
- une couche de réseaux de neurones récurrents LSTM, avec comme activation Relu,
- un dropout de 0.5 qui est pour rappel un paramètre de régularisation visant à empêcher le sur-apprentissage.

Le corpus de données des séries chronologiques considère 8 périodes de temps en entrée et prédit pour 4 périodes de temps. La taille du corpus est de 15 000. Chaque entrée contient les données du contexte de décision (étiquettes de l'offre) et les valeurs de l'indicateur de performance pour 8 périodes de temps. Chaque sortie contient les valeurs de l'indicateur de performance pour 4 périodes de temps. Ce corpus a été créé pour deux indicateurs de performance : conversions et clics. Ce corpus considère les évènements de 15 canaux de diffusion, et 10 classes sources. Les premiers

90% du corpus sont donnés au corpus d'entraînement et les derniers 10% au corpus de test. Rappelons que l'architecture de l'algorithme et du corpus d'entrée de l'algorithme d'apprentissage ont été choisis afin de capturer les tendances générales sur les canaux de diffusion quelles que soient les classes sources. En effet, grâce à ces architectures, si pour une classe source nous n'avons pas de données récentes sur certains canaux, la considération de tous les canaux et toutes les classes sources dans un même modèle d'apprentissage permet d'être en mesure d'avoir quand même une estimation des indicateurs de performance. Cela apporte une réponse à un problème de cold-start, c'est-à-dire lorsque certaines données d'entrée ne sont pas connues par le système et pour lesquelles nous souhaitons faire des recommandations (en l'occurrence ici les données caractérisant l'offre d'emploi).

Comme mesure d'évaluations, nous avons utilisé l'erreur quadratique moyenne (MSE) présentée dans le chapitre 4.

5.4.3.3 Résultats

Dans cette section de résultats, nous allons comparer trois modèles entraînés pour d'abord l'indicateur de performance des clics et ensuite des conversions :

A) Un modèle qui a été entraîné sans utiliser les classes sources.

B) Un modèle qui utilise les étiquettes "Secteur", "Département", "Expérience", pour créer les classes sources. Ces étiquettes ont été choisies après une analyse statistique de corrélation entre le nombre de clics et les étiquettes caractérisant l'offre d'emploi. Le coefficient de corrélation est la mesure qui quantifie la force de la relation linéaire entre deux variables. La valeur p a été utilisée pour tester l'hypothèse alternative que la corrélation mesurée est présente dans nos données $p \leq 0.05$.

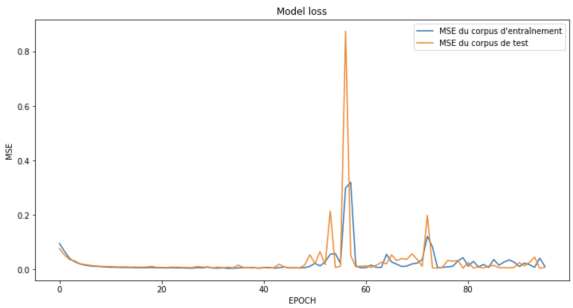
C) Un modèle qui utilise l'ensemble des étiquettes pour créer les classes sources.

Nombre de clics Dans la Figure 5.15, nous pouvons visualiser l'évolution de la MSE au fur et à mesure des entraînements pour le corpus d'entraînement et le corpus de test pour les modèles A, B et C. Nous pouvons voir que parmi ces trois modèles celui qui est le plus stable est le modèle B (Figure 5.15b). En effet, la MSE de test (du corpus de test) diminue à mesure que l'entraînement évolue. Ainsi, nous pouvons en déduire que pour ce modèle, les tendances des séries chronologiques pour les classes sources sont capturées au fur et à mesure de l'entraînement. Contrairement au modèle B, la MSE du modèle A (Figure 5.15a) n'arrive pas à se stabiliser à mesure que l'entraînement avance. Ceci est dû au fait que les contextes de décision, c'est-à-dire les offres d'emploi ne sont pas suffisamment regroupées entraînant un plus grand nombre de classes sources pour lesquelles le modèle doit capturer les tendances. De plus, pour certaines classes, les séries chronologiques associées sont très courtes. Les tendances sont donc plus difficiles à capturer. Enfin, le modèle C (Figure 5.15c) considère, en entrée de l'apprentissage supervisé, toutes les étiquettes de l'offre pour regrouper les offres d'emploi. La création des classes sources se base aussi sur l'ensemble de ces étiquettes. Nous pouvons remarquer que pour ce modèle, la MSE s'est

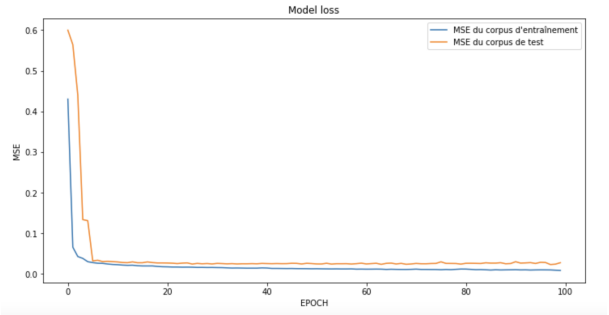
stabilisée entraînant une valeur plus faible que pour le modèle A (Table 5.18). En revanche, ce modèle reste moins stable. Ceci peut être dû au fait que la considération de plusieurs étiquettes entraîne de même que pour le modèle A des séquences de séries chronologiques plus courtes pour capturer des tendances. Ces déductions se confirment à travers les scores de la MSE maximale, minimale et moyenne présentés dans la Table 5.18. Nous pouvons y voir que le modèle B est celui qui produit les meilleurs résultats, c'est-dire-dire qui obtient la MSE la plus faible.

Concernant le modèle de prédiction pour la classe cible, nous avons décidé d'utiliser le modèle B qui a été entraîné sur des données dont les classes sources ont été regroupées selon les étiquettes "Métier", "Secteur", "Département", "Expérience". En effet, c'est celui qui a permis d'avoir la plus petite MSE et à capturer les tendances sur le corpus de test comme nous pouvons le voir sur la Figure 5.16. Nous l'avons donc appliqué pour prédire les valeurs futures du nombre de clics 5.22. Après avoir analysé les résultats concernant la prédiction du nombre de clics, nous allons dans la suite analyser les résultats de la prédiction du nombre de conversions.

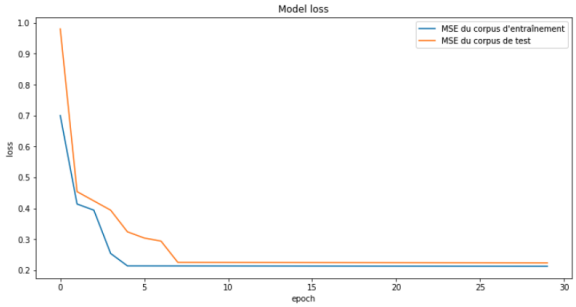
Nombre de conversions Rappelons qu'une conversion représente une personne qui a envoyé sa candidature. Pour la prédiction des valeurs futures du nombre de conversions, l'architecture de l'algorithme est la même que dans le problème de prédiction des clics. Les résultats de la MSE sont affichés dans la Figure 5.15. Tout d'abord, nous pouvons remarquer que le modèle A qui considère les offres d'emploi individuellement a les mêmes tendances que dans le cas du nombre de clics. En effet, l'évolution de la MSE n'est pas stable et les modèles d'entraînement et de test n'arrivent pas à se stabiliser. Ceci est dû au fait que les séries chronologiques associées sont courtes et les tendances ne sont pas capturées. Dans un second temps, contrairement aux résultats du nombre de clics, les tendances des modèles B 5.15e et C 5.15f s'inversent. En effet, le modèle C entraîné sur l'ensemble des étiquettes est beaucoup plus stable lors de l'entraînement et de l'évaluation et la MSE est plus faible que pour le modèle B. C'est très intéressant de voir ce phénomène, puisque nous pouvons déduire que pour prédire le nombre de conversions pour une offre d'emploi, il est important de considérer dans les données d'entrée toutes les étiquettes qui la décrivent, contrairement au modèle de prévisions des clics qui lui est plus performant avec uniquement trois étiquettes. Nous pouvons déduire que les étiquettes choisies par l'expert représentent bien la description d'un profil recherché puisqu'elles sont fortement corrélées aux nombres de conversions sur les canaux. De plus, les clics sont représentés par des profils de candidats assez variés qui ne vont pas jusqu'à l'envoie de leur candidature, contrairement aux candidats qui postulent (entraînant une conversion) qui eux nécessitent une plus précise représentation des caractéristiques du profil recherché. Ces déductions se confirment à travers les scores de la MSE maximale, minimale et moyenne présentés dans la Table 5.18. Nous pouvons y voir que le modèle C est celui qui obtient les meilleurs résultats. Dans la Figure 5.17 nous pouvons visualiser les conversions prédites et réelles sur le corpus de test et voir que le modèle capture les tendances. Nous pouvons remarquer aussi que la MSE est plus faible que pour les clics. Ceci est normal, puisque le nombre de conversions n'est pas aussi variable que le nombre de clics. Très souvent le modèle doit prédire entre



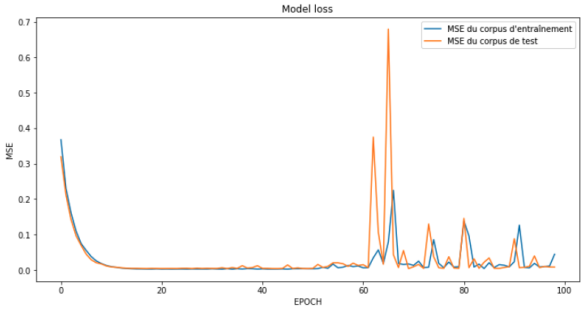
(a) Erreur quadratique moyenne (MSE) du nombre de clics pour le modèle A :
sans utiliser les classes sources



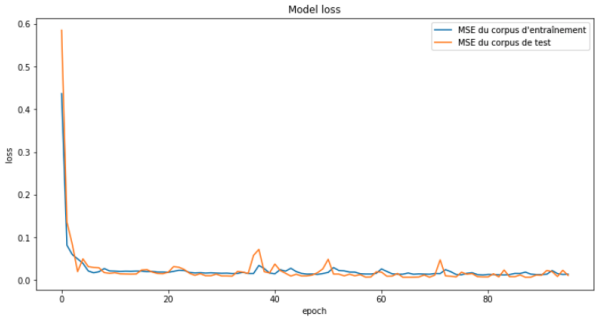
(b) Erreur quadratique moyenne (MSE) du nombre de clics pour le modèle B : en utilisant les étiquettes "Secteur", "Département", "Expérience" pour créer les classes sources



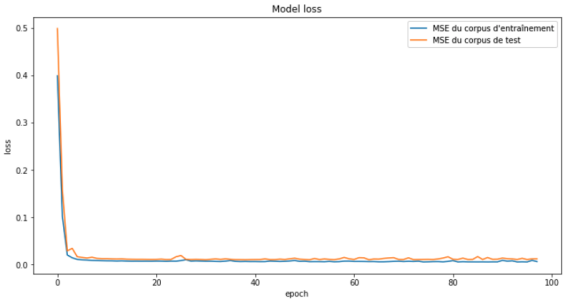
(c) Erreur quadratique moyenne (MSE) du nombre de clics pour le modèle C : en utilisant l'ensemble des étiquettes pour créer les classes sources



(d) Erreur quadratique moyenne (MSE) du nombre de conversions pour le modèle A :
sans utiliser les classes sources



(e) Erreur quadratique moyenne (MSE) du nombre de conversions pour le modèle B : en utilisant les étiquettes suivantes pour créer les classes sources : "Secteur", "Département", "Expérience"



(f) Erreur quadratique moyenne (MSE) du nombre de conversions pour le modèle C : en utilisant l'ensemble des étiquettes pour créer les classes sources

Fig. 5.15. Évolution de l'erreur quadratique lors de l'entraînement de la prédiction des valeurs futures pour les clics et les conversions en utilisant les modèles A,B et C

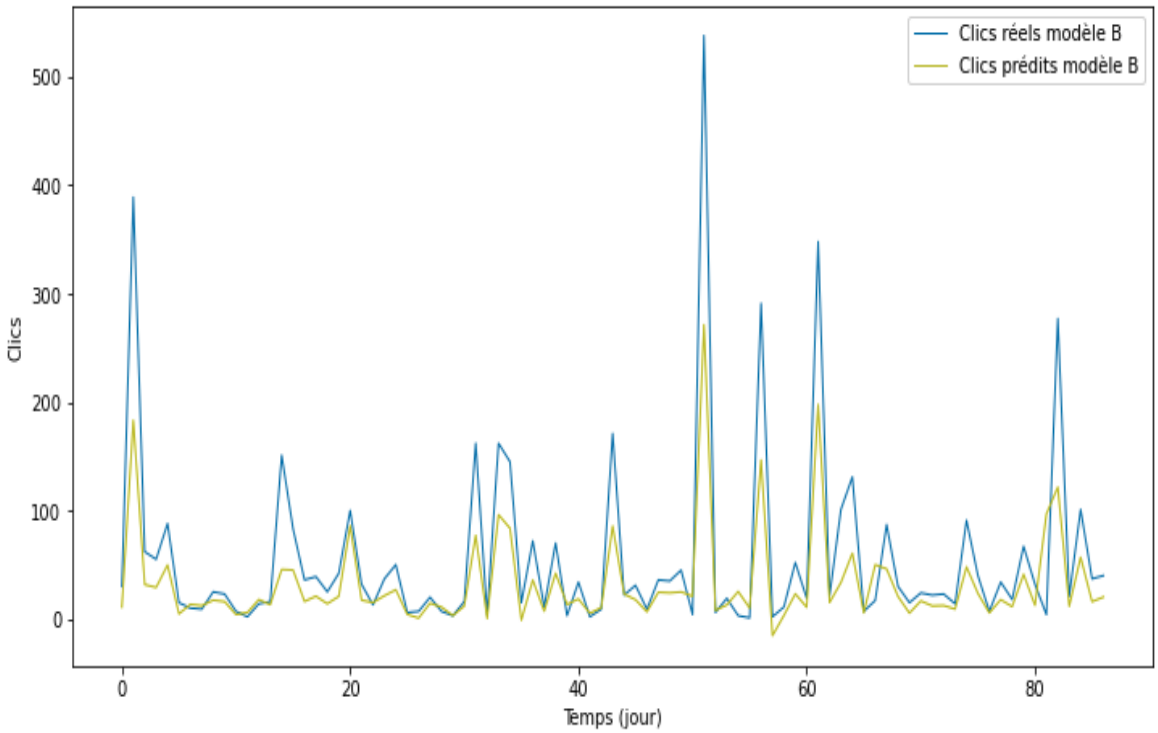


Fig. 5.16. Clics réels et prédits avec le modèle B

des valeurs entre 0 et 10.

Table 5.18
Erreur moyenne quadratique MSE pour la prédiction des indicateurs de performance clics et conversions futurs

Métrique	Indicateurs de performance	Modèle A	Modèle B	Modèle C
MSE maximum	Clics Conversions	0.85 0.68	**0.59** 0.6	0.98 **0.49**
MSE minimum	Clics Conversions	**0.03** 0.090	**0.03** 0.056	0.2 **0.0019**
MSE moyen	Clics Conversions	0.16 0.20	**0.05** 0.03	0.29 **0.0090**

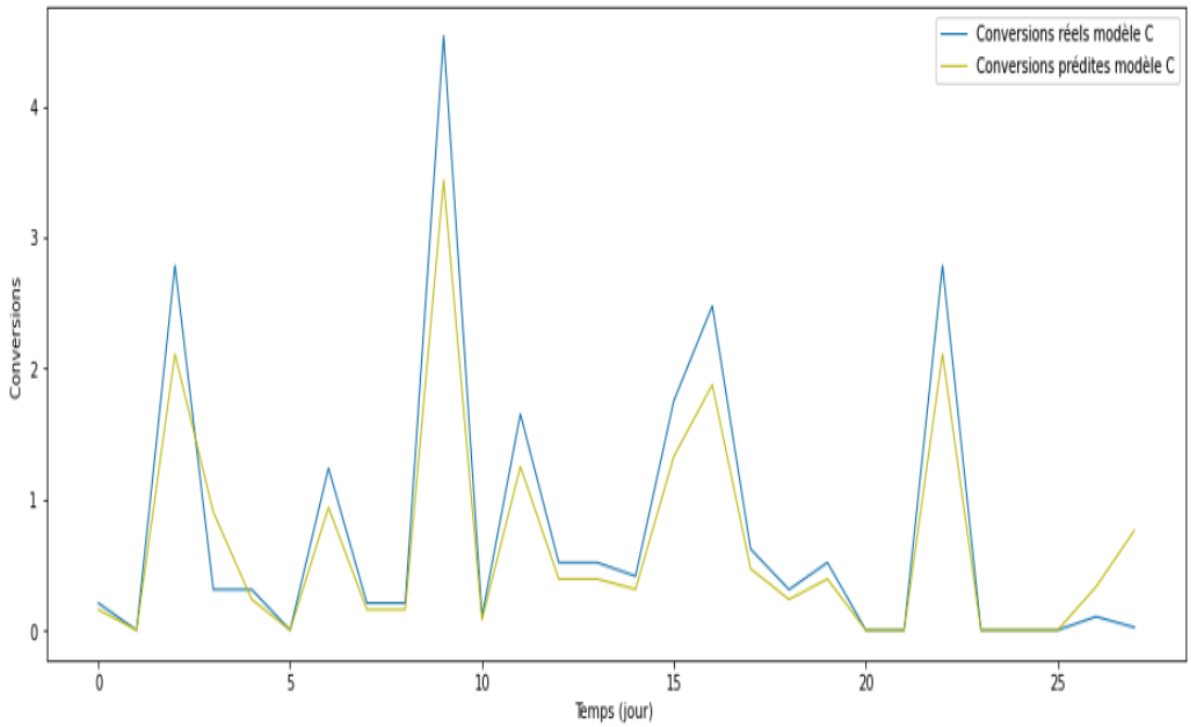


Fig. 5.17. Conversions réelles et prédites avec le modèle B

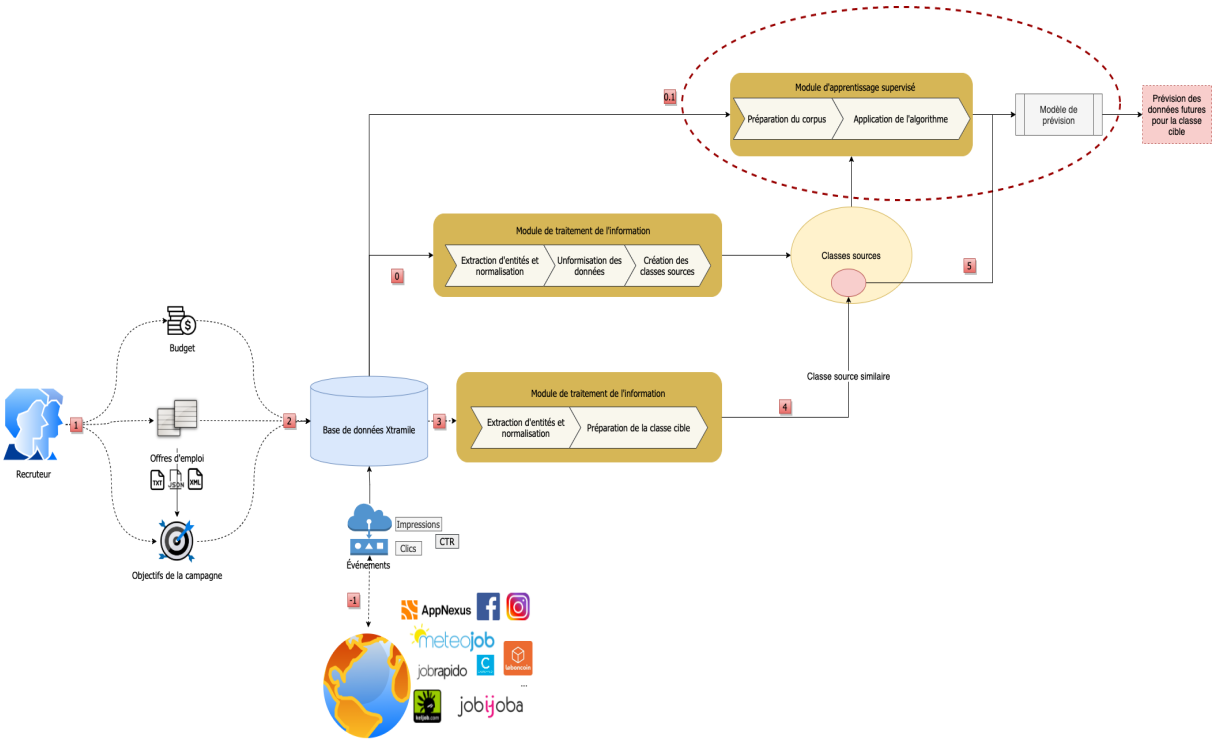


Fig. 5.18. Application du modèle de prévision des clics et des conversions sur la classe cible

5.4.4 Module de filtrage

Les canaux de diffusion étant multiples, il est difficile de les considérer tous dans les problèmes d'optimisation. En effet, cela entraîne la recherche de solutions optimales dans un espace de recherche très grand qui potentiellement entraînera une convergence plus lente vers les solutions optimales. Rappelons que le module de filtrage consiste à utiliser le profil normalisé du contexte pour filtrer les items à recommander au décideur. Ce module nécessite la caractérisation des canaux de diffusion afin de recommander uniquement ceux ayant des caractéristiques similaires au profil recherché par le recruteur. Pour caractériser les canaux, nous nous sommes appuyés sur des connaissances issues de sources externes et internes.

5.4.4.1 Sources externes de connaissance

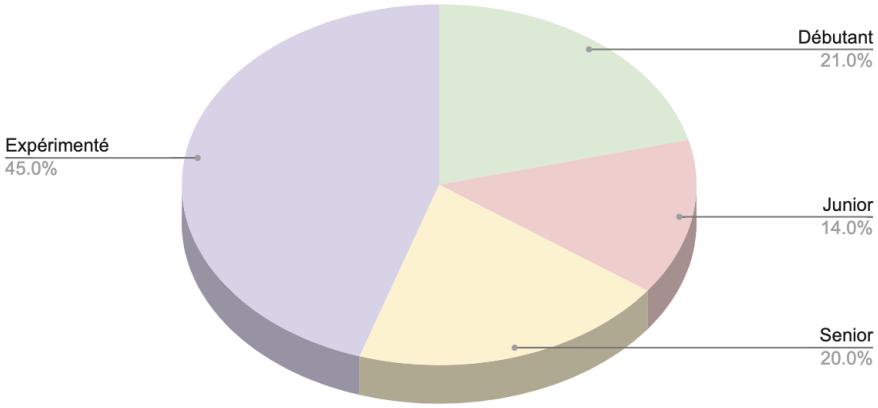
Les sources externes de connaissance utilisées s'appuient sur l'analyse des différents canaux à partir d'outil permettant de récolter leurs informations. Ces sources externes de connaissance regroupent les éléments suivants pouvant caractériser un canal : date de création, nombre d'employés, pays, type de nation (nationale ou internationale), satisfaction utilisateur, description du canal, stratégie de marketing, temps moyen passé sur le site par utilisateur, nombre de visites par mois, catégories de métier, nombre d'années d'expérience pour les offres diffusées, formation requise pour les offres diffusées, provenance des utilisateurs (direct, publicité, etc.). L'utilisation de sources externes de connaissance nécessite l'analyse de leurs données, leur structuration et leur traduction. Par exemple, la source de connaissance externe 'JobboardFinder' met à disposition les profils d'expérience les plus présents sur les canaux de diffusion (Figure 5.19a) ou encore les catégories de métiers (Figure 5.19b). Dans cette base de source externe, il existe 195 canaux de diffusion caractérisés.

5.4.4.2 Connaissances internes

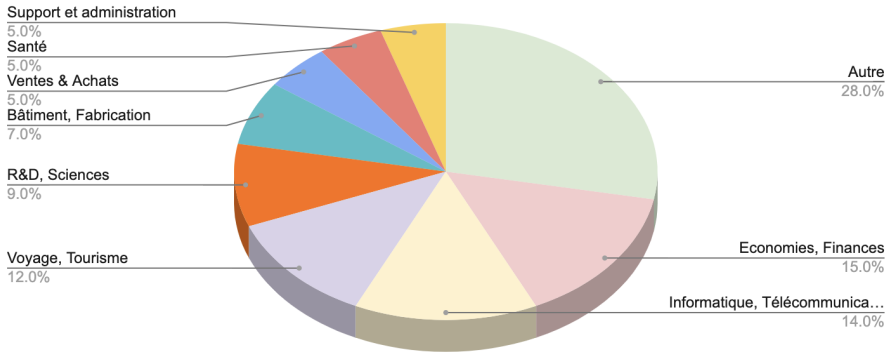
Les sources internes de connaissance utilisées s'appuient sur l'analyse des différentes données récoltées en interne lors des campagnes de recrutement afin de caractériser les canaux. Ces sources internes de connaissance regroupent les éléments suivants pouvant caractériser un canal : le type de contrat, les villes et régions, les formations.

Ces données issues des sources externes et internes sont stockées dans une table de données faisant office de matrice représentant les caractéristiques des canaux de diffusion. Cette matrice est utilisée dans l'algorithme 5 présenté dans le chapitre 4. Cet algorithme permet d'éliminer les canaux ne permettant pas de retrouver les profils souhaités dans l'offre étant données leurs caractéristiques, entraînant ainsi un espace de recherche des canaux plus petit répondant aux objectifs du recruteur.

Les canaux de diffusion intégrés chez Xtramile au moment de ces expérimentations sont de 21 au total et de différents types : réseaux sociaux, métamoteurs et sites publicitaires. Le module de filtrage appliqué à l'offre d'emploi d'"Aide à domicile" a permis de garder 9 canaux sur 21. Les canaux éliminés sont principalement des canaux spécialisés dans des métiers spécifiques comme l'informatique pour "LesJeudis" ou



(a) Profils d'expérience



(b) Profils de métier

Fig. 5.19. Exemple de caractéristiques du canal Adzuna d'après la source 'JobboardFinder'

encore spécialisés dans des profils de cadre comme "cadreEmploi", etc. Les 9 canaux de diffusion restants sont utilisés dans le module suivant d'optimisation.

5.4.5 Module d'optimisation

Le module d'optimisation a pour objectif de répondre aux objectifs du recruteur tout en respectant ses contraintes budgétaires. Dans le chapitre de contribution, nous avons défini 3 problèmes d'optimisation :

— Problème mono-objectif et mono-période (OP1)

— Problème multi-objectifs et mono-période (OP2)

— Problème multi-objectifs et multi-périodes (OP3)

Dans la suite de cette section, nous allons appliquer les algorithmes définis dans la contribution pour chacun de ces problèmes et analyser les résultats.

Pour ce faire, nous nous sommes intéressés dans un premier temps aux différents

outils pour appliquer nos algorithmes à nos données. Dans la Table 5.19, nous présentons une comparaison de ces différents outils sur différents critères :

— Possibilité de considération des problèmes multi-objectifs

— Possibilité de choisir les algorithmes et leurs paramètres

— Aide à la prise de décision

— Accès gratuit

Nous pouvons constater à travers notre comparaison des différents outils d'optimisation, que pymoo est celui qui permet de répondre à toutes nos caractéristiques. De ce fait, dans la suite de ce module, l'ensemble des problèmes d'optimisation utilisent l'outil pymoo pour leur résolution. Ce module reçoit des données présentées dans l'annexe G contenant :

— Les séries chronologiques sur chaque canal possible (après le module de filtrage) prédites, liées à l'offre

— L(es) indicateur(s) de performance concerné(s) par les séries chronologiques

— l'objectif à atteindre (maximisation ou minimisation) pour chaque indicateur

— l(es) contrainte(s) liée(s) à chaque indicateur

Table 5.19
Comparaison des outils pour la résolution des problèmes d'optimisation

Caractéristiques	**PyGMO**	**Platypus**	**pymoo**	**GAMS solver**
Possibilité de considération des problèmes multi-objectifs	✓	✓	✓	✓
Possibilité de choisir les algorithmes et leurs paramètres	✓	✓	✓	✓
Aide à la prise de décisions	✗	✗	✓	✓
Accès gratuit	✓	✓	✓	✗ (selon les algorithmes)

Pour nos expérimentations du module d'optimisation, nous rappelons que nous souhaitons recommander pour l'offre d'emploi d'"Aide à domicile" D_i un ensemble de canaux optimal. Ceci revient à sélectionner les variables de décision x_{D_i,q_j} qui ont une valeur de 1 qui répondent aux objectifs du recruteur et d'éliminer x_{D_i,q_j} d'une valeur de 0. La liste de départ des canaux est la liste suivante Adzuna, Appnexus, Facebook / Instagram, Indeed, Jobijoba, Jobrapido, Jobtome, Jooble et Talent.com. Nous allons donc dans la suite de ce chapitre appliquer les différentes optimisations (OP1), (OP2),

(OP3) et discuter des résultats qui en découlent. À l'offre d'emploi "Aide à domicile" a été attribuée la classe source présentée dans la Table 5.17. De ce fait, elle s'est vue attribuée la série chronologique qui représente les données passées et futures de cette classe source.

5.4.5.1 Problème mono-période et mono-objectif

Nous avons considéré dans un premier temps un unique objectif du recruteur sur une seule période. L'objectif du recruteur est extrait des données d'entrée issues du système. Un exemple est présenté dans l'Annexe G. Dans ces données, nous pouvons voir que l'objectif du recruteur est de maximiser le nombre de conversions sur l'offre d'emploi tout en minimisant le nombre de clics et en respectant le budget de 200 euros. Dans ce premier problème, nous allons nous focaliser sur un seul objectif : maximiser les conversions.

La résolution de ce problème s'est basée sur l'algorithme génétique présenté dans le chapitre de contribution dont les paramètres sont les suivants :

— La taille de la population est fixée comme constante d'une valeur de 100. Une taille de 100 ou 150 individus s'avère souvent amplement suffisante, tant pour la qualité des solutions trouvées que pour le temps d'exécution de l'algorithme.

— L'initialisation de cette population est paramétrée à travers la méthode d'échantillonnage "sampling" qui est aléatoire. Elle génère une distribution presque uniforme des points dans l'espace de recherche. Par défaut, la méthode sélectionnée dans les travaux de la littérature et dans les outils de développement est cette méthode d'initialisation [82, 116]. Les valeurs non faisables sont écartées et un nouveau tirage est effectué jusqu'à obtenir 100 individus.

— Le taux de croisement est de 0.9 et le taux de mutation est de 0.03. Ces valeurs sont très couramment utilisées dans la littérature et sont fortement recommandées [26, 67, 58].

Résultats

— **Les meilleures solutions trouvées** : $[110001011]$. Ces valeurs correspondent à la variable de décision. Cette valeur est de 1 lorsque l'offre d'emploi d'aide à domicile est affectée au canal.

— **Valeurs de la fonction objectif** $[-63]$. Cette valeur signifie qu'en suivant la meilleure solution proposée le nombre de conversions espéré est de 63.
Note : cette valeur est négative comme résultat de l'algorithme, car nous souhaitons maximiser et l'algorithme n'accepte que des minimisations. Nous avons donc transformé le problème de maximisation en problème de minimisation

— **Violation des contraintes** : $[0]$ est la valeur confirmant que les contraintes financières sont respectées.

— **Valeur de la fonction de contrainte** $[-8]$. 8 euros est le budget restant en choisissant cette solution.

— **Temps d'exécution** : 1.68 secondes

Interprétation Les canaux choisis comme solution optimale par l'algorithme génétique sont :

— Facebook avec un total de conversions estimé à 4 pour un budget dépensé estimé d'environ 39.3 euros.

— Adzuna avec un total de conversions estimé à 4 pour un budget dépensé estimé d'environ 42.9 euros.

— Jobtome avec un total de conversions estimé à 4 pour un budget dépensé estimé d'environ 38.4 euros.

— Indeed avec un total de conversions estimé à 50 pour un budget dépensé estimé d'environ 63 euros.

— Jobrapido avec un total de conversions estimé à 1 pour un budget dépensé estimé d'environ 8.4 euros.

La solution optimale respecte la contrainte financière de 200 euros pour la première période en proposant une solution qui implique une dépense de 192 euros pour 63 conversions au total.

5.4.5.2 Problème mono-période et multi-objectif

Nous avons considéré dans un second temps deux objectifs du recruteur sur une seule période. En effet, nous avons remarqué que certains canaux, bien qu'ils génèrent beaucoup de conversions, ont tendance à nécessiter beaucoup de clics, entraînant une consommation excessive du budget. De ce fait, nous allons comparer les résultats d'une optimisation bi-objectifs, où les objectifs sont parfois conflictuels et l'optimisation à un seul objectif expérimentée ci-dessus.
La résolution de ce problème s'est basée sur l'algorithme NSGA-II présenté dans le chapitre de contribution avec les mêmes paramètres choisis pour le problème d'optimisation mono-objectif.

Résultats L'algorithme propose un front de Pareto composé de 12 solutions présentées dans la Figure 5.20 ci-dessous trois solutions du front de Pareto :

1. **Temps d'exécution** : 0.86 secondes
2. Solution 1 :
   — **Les meilleures solutions trouvées** : $[100101011]$. Ces valeurs correspondent à la variable de décision. Cette valeur est de 1 lorsque l'offre d'emploi d'"Aide à domicile" est affectée au canal.
   — **Valeurs des fonctions objectifs** : $[-62, 605]$. Ces valeurs correspondent respectivement aux nombres de conversions (62) et aux nombres de clics (605) espérés en choisissant les affectations au-dessus.
   — **Violation des contraintes** : $[0]$ est la valeur confirmant que les contraintes financières sont respectées.
   — **Valeur de la fonction de contrainte** : $[-18.5]$ signifie qu'il reste un budget de 18.5 euros.

3. Solution 2 :
   - **Les meilleures solutions trouvées** : $[000000110]$. Ces valeurs correspondent à la variable de décision. Cette valeur est de 1 lorsque l'offre d'emploi d'"Aide à domicile" est affectée au canal.
   - **Valeurs des fonctions objectifs** $[-53, 335]$. Ces valeurs correspondent respectivement aux nombres de conversions (53) et aux nombres de clics (335) espérés en choisissant les affectations au-dessus.
   - **Violation des contraintes** : $[0]$ est la valeur confirmant que les contraintes financières sont respectées.
   - **Valeur de la fonction de contrainte** : $[-99.5]$ signifie qu'il reste un budget de 99.5 euros.
4. Solution 3 :
   - **Les meilleures solutions trouvées** : $[000010111]$. Ces valeurs correspondent à la variable de décision. Cette valeur est de 1 lorsque l'offre d'emploi d'"Aide à domicile" est affectée au canal.
   - **Valeurs des fonctions objectifs** : $[-56, 461]$. Ces valeurs correspondent respectivement aux nombres de conversions (56) et aux nombres de clics (461) espérés en choisissant les affectations au-dessus.
   - **Violation des contraintes** : $[[0]]$ est la valeur confirmant que les contraintes financières sont respectées.
   - **Valeur de la fonction de contrainte** : $[-61.7]$ signifie qu'il reste un budget de 61.7 euros.

Toute la population résultant est faisable (ne viole pas les contraintes financières) à partir de la génération 0 comme nous pouvons le voir dans la Figure de l'Annexe H.1b. Contrairement au problème d'optimisation mono-période qui lui a permis d'avoir une non-violation des contraintes qu'à partir de la cinquième génération de population (voir Annexe H.1a). Cette remarque est logique puisque les premières semaines de campagne de recrutement génèrent plus de clics et de conversions entraînant la nécessite d'un budget plus conséquent. De ce fait, il nous paraît intéressant dans nos futures expérimentations de prévoir un budget plus haut les premières semaines plutôt que de le distribuer uniformément sur toutes les périodes.

Interprétation Dans la Figure 5.20, nous pouvons visualiser le front de Pareto suite à l'application de l'algorithme NSGA-II sur notre problème d'optimisation multi-objectifs et multi-périodes. Les solutions du front de Pareto avec les valeurs En comparaison à l'optimisation (OP1), nous pouvons remarquer qu'en incluant un objectif de minimisation de clics les solutions diffèrent de celles proposées à OP1. En effet, dans les trois premières solutions du front de Pareto, le nombre de conversions est légèrement plus petit que pour la solution proposée à OP1. En revanche, le budget restant est beaucoup plus grand permettant ainsi de distribuer ce budget sur d'autres canaux et d'augmenter d'autant plus le nombre de conversions aux prochaines périodes.

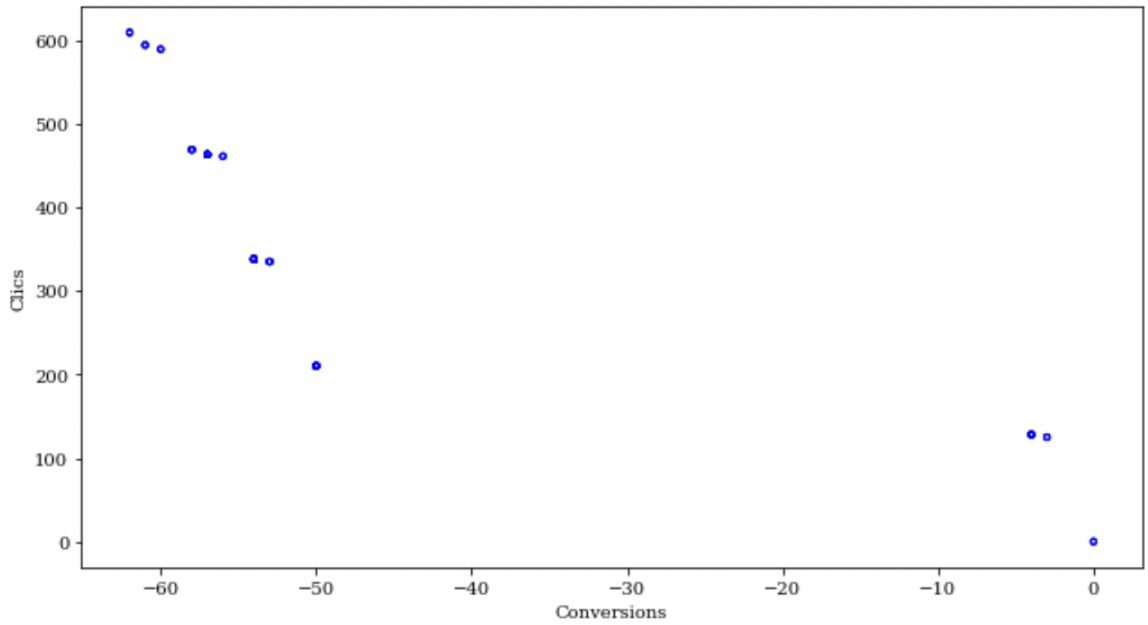


Fig. 5.20. Résultats du front de Pareto après l'application de l'algorithme NSGA-II pour le problème mono-période et multi-objectifs

5.4.5.3 Problème multi-périodes et multi-objectifs

Au début de la campagne, et malgré son expertise dans le e-recrutement, il peut s'avérer très difficile pour le recruteur d'anticiper l'évolution des indicateurs de performance sur les canaux.

De ce fait, ce problème d'optimisation multi-périodes est important puisqu'il utilise les prévisions du module précédent pour aider le recruteur à avoir une vision globale sur les canaux pour l'ensemble des périodes futures. Pour commencer, nous avons considéré quatre périodes de 7 jours (c'est généralement la durée minimale d'une campagne de recrutement). Cette optimisation multi-périodes entraîne la recommandation des canaux sur les quatre périodes, entraînant la multiplication du nombre de variables de décision par quatre.

Résultats Les résultats de l'algorithme sont le front de Pareto avec toutes les solutions optimales répondant aux objectifs (voir Figure 5.21) :

1. **Temps d'exécution** : 1.33 secondes
2. Solution 1 :
   - **Les meilleures solutions trouvées** : $[11100111011111010001111111011101011]$. Ces valeurs correspondent à la variable de décision pour chaque période T=0 à T=3 pour chaque canal. Par exemple, il est recommandé d'affecter l'offre d'"Aide à domicile" au premier canal pour la période T=0, T=1 et T=2. Pour T=3 la variable de décision est nulle donc non recommandée.
   - **Valeurs des fonctions objectifs** : $[-132, 2191]$. Ces valeurs correspondent respectivement aux nombres de conversions (132) et aux nombres de clics

(2191) espérés en choisissant les affectations au-dessus pour les quatre périodes de temps T=0 à T=3.

- **Violation des contraintes** : $[0]$ signifie que les contraintes financières sont respectées.
- **Valeur de la fonction de contrainte** : $[-142.7]$ signifie qu'il reste un budget de 142.7 euros.

3. Solution 2 :
   - **Les meilleures solutions trouvées** : $[00000110000101000011010000010001111]$. Ces valeurs correspondent à la variable de décision pour chaque période T=0 à T=3 pour chaque canal. Par exemple, il est recommandé de ne pas affecter l'offre d'"Aide à domicile" au premier canal pour les quatre périodes.
   - **Valeurs des fonctions objectifs** : $[-41.452.]$. Ces valeurs correspondent respectivement aux nombres de conversions (41) et aux nombres de clics (452) espérés en choisissant les affectations au-dessus pour les quatre périodes de temps T=0 à T=3.
   - **Violation des contraintes** : $[0]$ signifie que les contraintes financières sont respectées.
   - **Valeur de la fonction de contrainte** $[-664.4]$ signifie qu'il reste un budget de 664.4 euros.
4. Solution 3 :
   - **Les meilleures solutions trouvées** : $[01100011000101000001010111001011101 1]$. Ces valeurs correspondent à la variable de décision pour chaque période T=0 à T=3 pour chaque canal. Par exemple pour le premier canal, il est recommandé d'affecter ce canal à l'offre d'"Aide à domicile" pour la période T=1 et T=2. Pour T=0 et T=3 la variable de décision est nulle donc non recommandée.
   - **Valeurs des fonctions objectifs** : $[-93.1158.]$. Ces valeurs correspondent respectivement aux nombres de conversions (93) et aux nombres de clics (1158) espérés en choisissant les affectations au-dessus pour les quatre périodes de temps T=0 à T=3.
   - **Violation des contraintes** : $[0]$ signifie que les contraintes financières sont respectées.
   - **Valeur de la fonction de contrainte** $[-452.6]$ signifie qu'il reste un budget de 664.4 euros.

Interprétation Dans la Figure 5.21, nous pouvons visualiser le front de Pareto suite à l'application de l'algorithme NSGA-II sur notre problème d'optimisation multi-objectifs et multi-périodes. En comparaison à l'optimisation (OP2), nous pouvons remarquer qu'en incluant des variables de décision sur la période de diffusion, différentes solutions avec différents compromis sont proposées au décideur. En effet, chacune de ces solutions entraîne une consommation de budget différente pouvant être visualisée à

travers la valeur de la fonction de contrainte. Si le décideur n'est pas satisfait des valeurs des fonctions objectifs pour les solutions proposées, il peut s'orienter vers celle qui consomme le moins de budget. L'avantage de cette optimisation multi-périodes est de laisser le recruteur décider des périodes qu'il préfère pour commencer ou continuer une campagne de recrutement puisqu'il a une meilleure visibilité des tendances des indicateurs de performance pour son offre d'emploi sur les canaux de diffusion sur différentes périodes futures.

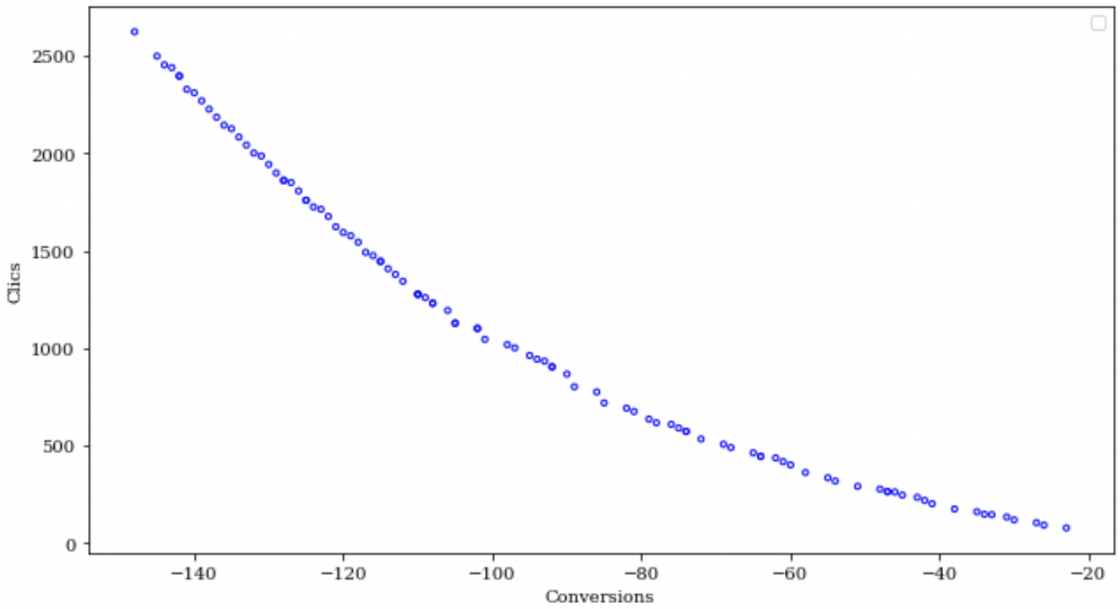


Fig. 5.21. Résultats du front de Pareto après l'application de l'algorithme NSGA-II pour le problème multi-périodes et multi-objectifs

5.4.6 Module d'apprentissage par renforcement

Le module d'apprentissage par renforcement est appliqué après le choix du décideur et la diffusion des offres d'emploi sur les canaux choisis. Le tracker reçoit les données réelles des indicateurs de performance sur chaque canal pour chaque offre d'emploi. Le système d'aide à la décision met à jour les séries chronologiques et se ré-entraîne. En parallèle, le module d'apprentissage par renforcement utilise la fonction de mise à jour présentée dans la contribution du chapitre 4 afin de recalculer le problème d'optimisation pour les futures périodes avec les nouvelles valeurs de prédiction. La fonction de mise à jour étant :

$$Q_{n+1} \leftarrow Q_n + \alpha(R_n - Q_n)$$

$$\Leftrightarrow$$

$$estimate_{new} \leftarrow estimate_{old} + \alpha(observation - estimate_{old})$$

Afin d'évaluer l'apport de ce module dans le système d'aide à la décision proposé, nous avons mis à jour le modèle d'apprentissage supervisé en utilisant les données

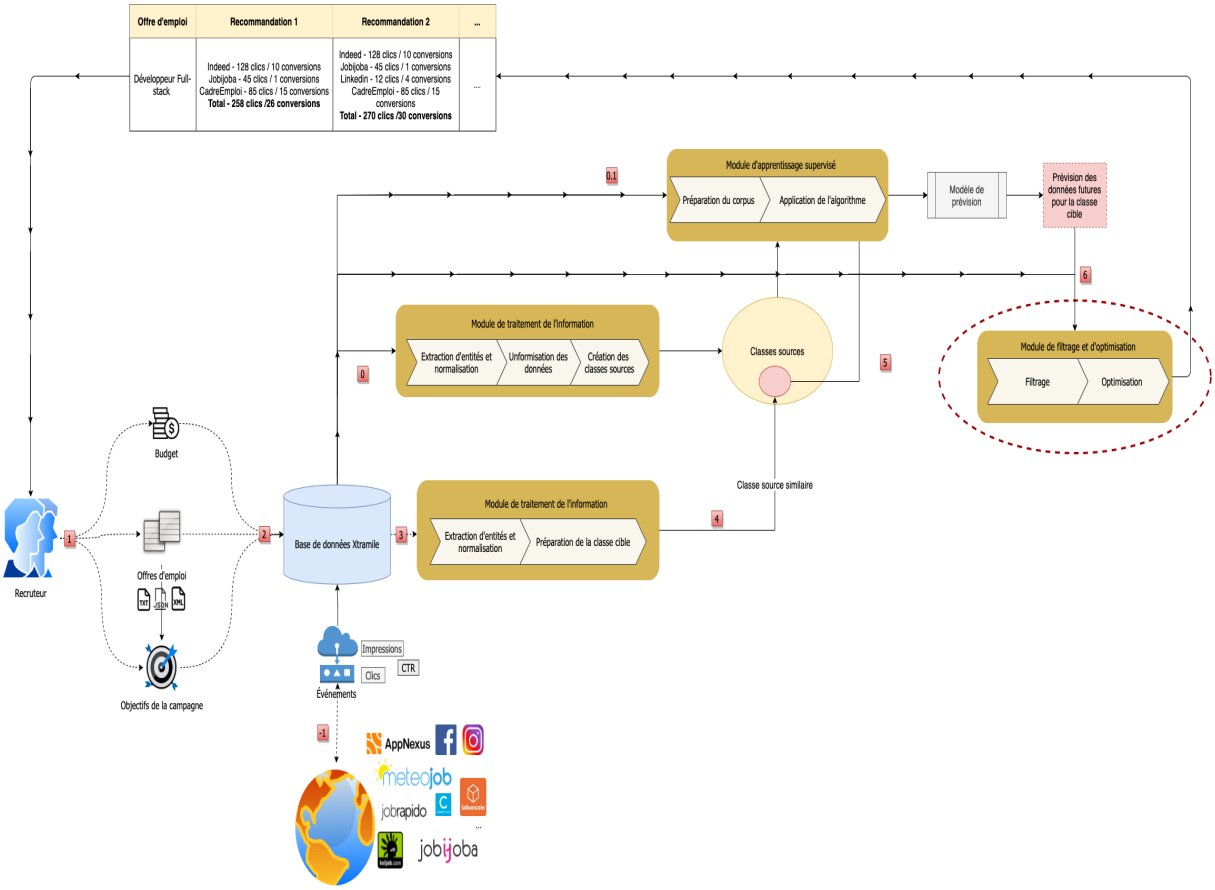


Fig. 5.22. Application du module de filtrage et d'optimisation pour recommander les canaux au décideur

de la première période suite au choix du décideur et nous avons ré-entraîné le modèle. Nous avons noté l'amélioration de l'estimation des indicateurs de performance clics et conversions. En revanche, les nouvelles prédictions du nombre de clics et de conversions pour les périodes suivantes n'ayant pas évolué, les résultats des algorithmes d'optimisation n'ont pas changé. De ce fait, dans nos prochains travaux, nous souhaitons évaluer ce module sur un contexte de décision différent pour noter ses avantages.

5.4.6.1 Comparaison avec et sans recommandation

Nous avons comparé les résultats des décisions du recruteur sur le choix des canaux sans les recommandations d'ADE^2 et avec les recommandations. Pour cela, nous avons laissé le manager de campagnes choisir les canaux selon son expertise et diffuser les offres d'emploi. À la fin des quatre périodes de campagne, nous avons comparé nos résultats prédictifs et les résultats obtenus suite à la décision des choix des canaux du manager de campagnes. Les recommandations d'ADE^2 sont les suivantes :

1. Période 1 : $x_{D_i,Facebook} = 1$, $x_{D_i,Adzuna} = 0$, $x_{D_i,Jobijoba} = 1$, $x_{D_i,Appnexus} = 1$, $x_{D_i,Jooble} = 0$, $x_{D_i,Jobtome} = 1$, $x_{D_i,Talent.com} = 1$, $x_{D_i,Indeed} = 1$, $x_{D_i,Jobrapido} = 1$
2. Période 2 : $x_{D_i,Facebook} = 1$, $x_{D_i,Adzuna} = 1$, $x_{D_i,Jobijoba} = 1$, $x_{D_i,Appnexus} = 1$, $x_{D_i,Jooble} = 0$, $x_{D_i,Jobtome} = 1$, $x_{D_i,Talent.com} = 1$, $x_{D_i,Indeed} = 1$, $x_{D_i,Jobrapido} = 0$
3. Période 3 : $x_{D_i,Facebook} = 1$, $x_{D_i,Adzuna} = 1$, $x_{D_i,Jobijoba} = 0$, $x_{D_i,Appnexus} = 0$, $x_{D_i,Jooble} = 0$, $x_{D_i,Jobtome} = 1$, $x_{D_i,Talent.com} = 1$, $x_{D_i,Indeed} = 1$, $x_{D_i,Jobrapido} = 1$
4. Période 4 : $x_{D_i,Facebook} = 0$, $x_{D_i,Adzuna} = 1$, $x_{D_i,Jobijoba} = 1$, $x_{D_i,Appnexus} = 1$, $x_{D_i,Jooble} = 1$, $x_{D_i,Jobtome} = 1$, $x_{D_i,Talent.com} = 0$, $x_{D_i,Indeed} = 0$, $x_{D_i,Jobrapido} = 1$

En comparaison avec le manager des campagnes qui a décidé de diffuser sur l'ensemble de ces canaux, nous avons pu remarquer :

1. en choisissant de diffuser sur l'ensemble de ces canaux sur les quatre périodes, il y a eu un gain de temps au niveau de la mise hors ligne et en ligne des offres d'emploi qui est de l'ordre de 30 minutes si le recruteur avait suivi nos recommandations.
2. les canaux tels que Talent.com et Indeed ont généré à la dernière période plus de clics que de conversions entraînant une perte d'argent de 200 euros pour le recruteur.
3. Le canal Jooble a été choisi par le manager des campagnes puisque "ce canal marche bien sur des profils d'assistant". En revanche, après analyse de ce canal, nous avons remarqué que celui-ci ne marche pas sur tous les secteurs d'activité. De ce fait, il est important de créer les classes sources sur la base de plusieurs étiquettes et donc de plusieurs types d'information décrivant l'offre.

De ce fait, ces recommandations auraient permis un gain d'argent de 500 euros pour un nombre de conversions plus faible de 0.2 %.

5.4.7 Conclusion

Dans ce chapitre notre objectif de recherche est de répondre à la problématique suivante :

QR3

Comment concevoir un système d'aide à la décision qui s'adapte à un environnement incertain, évolutif et répondant à des objectifs multiples dont les paramètres sont variables ?

Cette question de recherche entraîne les verrous suivants que nous avons définis auparavant :

1. (QR3-C1) L'environnement est incertain ;
2. (QR3-C2) L'environnement est évolutif entraînant l'évolution dans le temps des variables du système et donc un problème non stationnaire.
3. (QR3-C3) Les objectifs du décideur à atteindre sont multiples ;
4. (QR3-C4) Le contexte de décision rédigé en langage naturel ;

L'objectif d'ADE^2 est de recommander au recruteur des canaux qui évoluent dans un environnement générant des indicateurs de performance évoluant dans le temps. Le recruteur a un contexte de décision contenu dans l'offre d'emploi qui est rédigé en langage naturel. Le recruteur a aussi des objectifs et des contraintes financières.

Pour répondre aux objectifs industriels, nous avons instancié le SAD générique proposé défini dans le Chapitre de contribution 4 dans le cas de l'affectation des offres d'emploi (contexte de décision) à des canaux (items) à afin de démontrer la pertinence de cette proposition. Nous évaluons la qualité des résultats obtenus pour vérifier l'intérêt du SAD et des différents modules.

Pour cela, nous avons mis en place d'ADE^2 un système d'aide à la décision dans un environnement évolutif qui se compose de plusieurs modules permettant de répondre individuellement ou globalement aux verrous. Notre premier module de **traitement de l'information** a pour objectif de traiter l'offre d'emploi qui est rédigée en langage naturel **(QR3-C4)**. Pour répondre à cette problématique, le chapitre de contribution sur l'étiquetage de séquences et la normalisation a été utilisé afin de simplifier le traitement de cette offre d'emploi. L'étiquetage automatique des séquences et la normalisation ont permis de transformer les offres d'emploi de façon à simplifier leur interprétation dans les modules suivants. Elles sont également utilisées pour la transformation de l'offre d'emploi en fichier XML pour la diffusion des offres d'emploi sur les canaux. Le module de traitement de l'information a également pour objectif d'unifier et préparer les indicateurs de performance qui sont variables et évolutifs dans le temps **(QR3-C2)**. Afin de traiter ce verrou, une création de classes sources a été définie afin de générer des groupes de données sémantiquement proches et contribuer à l'agrégation de séries chronologiques afin d'anticiper leur prévision. Ce module est important pour l'apprentissage supervisé puisqu'il a permis de remarquer que la prédiction des valeurs futures des clics et des CV dépendaient fortement de certaines étiquettes décrivant l'offre d'emploi en utilisant l'algorithme CNN-LSTM. L'utilisation des étiquettes "Secteur", "Département", "Expérience" a permis d'avoir une MSE moyen de 0.090 qui est très bon et le modèle d'entraînement se stabilise très rapidement. Nous avons pu déduire à travers nos résultats que pour prédire les clics, l'utilisation de toutes les étiquettes pour décrire les classes sources ne permet

pas d'avoir de bons résultats. Les clics sont représentés par des profils de candidats assez variés contrairement aux conversions qui elles nécessitent une plus précise représentation des offres d'emploi et donc de toutes les étiquettes pour construire les classes sources. Le module d'**apprentissage supervisé** répond au verrou **(QR3-C2)**.

Étant donné la multiplicité des canaux et leurs caractéristiques, un premier filtre dans le module de **filtrage** a été proposé afin de ne garder que les canaux ayant des caractéristiques semblables à l'offre d'emploi. Pour caractériser les canaux, nous avons utilisé des sources internes et externes. Le deuxième objectif de ce module est de limiter l'espace de recherche des canaux répondant aux objectifs du décideur **(QR3-C3)**. Nous avons remarqué qu'à l'issue de ce module, l'espace de recherche des canaux optimaux a diminué permettant de baisser leur nombre de 26 à 10 et d'éliminer ceux qui ne répondent pas aux profils attendus par le recruteur. L'avant-dernier module d'**optimisation** utilise l'algorithme génétique NSGA-II et l'outil pymoo afin de choisir les canaux répondant aux multiples objectifs du recruteur en considérant ses contraintes **(QR3-C3)**. Étant donné l'environnement évolutif **(QR3-C1)**, ADE^2 met à jour ses recommandations à chaque période avec la prise en compte des valeurs des clics et des conversions, obtenus lors de la dernière période grâce au module de **renforcement** pour améliorer les recommandations pour les périodes à venir.

Les objectifs d'ADE^2 sont le gain de temps, d'efficacité et d'argent. Le gain de temps a été noté grâce à l'étape d'étiquetage de séquences qui permet non seulement d'interpréter le profil recherché par le recruteur, mais aussi de structurer l'offre d'emploi afin de générer un fichier XML pour une offre d'emploi en 6 secondes à comparer à 8 minutes lorsque c'est fait manuellement. De plus, l'analyse des données et les prédictions permettent au recruteur de prendre plus rapidement une décision lors du choix des canaux. Concernant le gain d'argent nous avons remarqué lors de nos expérimentations que le recruteur se fie beaucoup aux expériences passées qu'il a eues. Néanmoins, étant donné l'évolution des canaux dans le temps, il peut se tromper entraînant une perte d'argent. Le modèle d'optimisation multi-périodes a permis d'optimiser son budget. Concernant le gain d'efficacité, nous n'avons pas pu évaluer cette métrique par une expérimentation concernant un recruteur souhaitant optimiser le nombre de CV pertinents.

Conclusion et perspectives

1 Contexte et questions de recherche

Les problématiques du recruteur ont évolué avec la révolution digitale. En effet, la principale préoccupation du recruteur avant l'émergence des canaux de diffusion était la visibilité de son offre d'emploi pour recruter. Avec la révolution digitale, le nombre de canaux s'est multiplié entraînant de nouvelles difficultés pour le recruteur. Nous sommes donc face aux questions industrielles ci-dessous :

- **(QI-1)** Comment choisir les canaux de diffusion optimaux par rapport à une offre ou une catégorie d'offres ?
- **(QI-2)** Comment optimiser le budget alloué sur les canaux pour répondre à des objectifs de recrutement ?

La littérature s'est intéressée à la problématique du e-recrutement pour répondre à ces questions en proposant des systèmes de recommandation de canaux de diffusion pour le recrutement. Cependant, nous avons noté plusieurs limites à ces travaux. Tout d'abord, l'absence de l'identification du profil recherché à partir de l'offre d'emploi pour la recommandation des canaux (L1). Ces systèmes exploitent des indicateurs de performance reposant uniquement sur le nombre de clics et le taux de conversions (L2). De plus, dans ce système un unique objectif est considéré (L3). Enfin, l'environnement incertain et évolutif dans le temps du e-recrutement n'est pas pris en compte (L4).
L'objectif de cette thèse est de pallier ces limites en proposant un système d'aide à la décision pour répondre aux objectifs multiples du recruteur et à l'environnement évolutif du recrutement. Nous avons décomposé cet objectif en trois sous-objectifs. Le premier consiste à analyser et identifier le profil recherché à partir de l'offre d'emploi, qui répond à la limite (L1) et entraîne la question de recherche :

- **(QR1)** - Comment identifier et extraire de l'information à partir de textes rédigés en langage naturel en utilisant leur schéma organisationnel tout en étant robuste à l'évolution du vocabulaire ?

Le second sous-objectif vise à analyser la pertinence d'un CV pour pouvoir introduire dans le système d'aide à la décision l'indicateur de performance basé sur la pertinence des CV, il répond à la limite (L2) et renvoie à la question de recherche :

- **(QR2)** - Comment apparier deux textes rédigés en langage naturel ?

Enfin, le dernier sous-objectif est de considérer les objectifs multiples du recruteur et l'environnement incertain du e-recrutement afin d'améliorer la prise de décision du recruteur. Ce dernier objectif répond aux limites (L3) et (L4) et amène à la question de recherche :

- **(QR3)** - Comment concevoir un système d'aide à la décision qui s'adapte à un environnement incertain, évolutif et répondant à des objectifs multiples dont les paramètres sont variables dans le temps ?

Nous résumons maintenant les contributions de cette thèse, associées aux questions de recherche.

2 Contributions

DEEP : Une méthodologie pour l'extraction d'entités en se basant sur le schéma organisationnel à partir de textes rédigés en langage naturel

Pour répondre à la question de recherche **(QR1)** nous avons choisi l'approche par étiquetage de séquences puisque nous avons fait l'hypothèse que cette approche permet d'améliorer l'extraction d'entités à partir de textes rédigés en langage naturel (voir Chapitre 2). En revanche, cette approche nécessite la mise en place d'un corpus annoté de qualité pour l'apprentissage. Par conséquent, nous avons intégré le schéma organisationnel qui a pour avantage de simplifier la compréhension d'un texte et d'aider à considérer le contexte d'une séquence ou d'un ensemble de séquences. Nous avons ainsi proposé une méthodologie d'étiquetage de séquences qui permet de considérer le schéma organisationnel tout en répondant aux différentes caractéristiques du langage naturel : ambiguïté et évolution du vocabulaire. Ensuite, afin de traiter facilement ces informations identifiées et extraites à partir du texte, nous avons proposé deux types de normalisations des séquences : type primitif et type référence. Ces deux normalisations se font de deux façons différentes, respectivement par approche par règles et par classification.

DEEP a été expérimentée sur un corpus d'offres d'emploi (voir Section 5.2). Sa mise en œuvre dans le domaine d'application des offres d'emploi, avec un expert, un master et trois annotateurs, a pris environ 300 heures. Nous avons également proposé d'impliquer des annotateurs qui n'étaient pas familiers avec le domaine afin de faciliter la mise en œuvre du corpus et d'améliorer la vitesse. Ceci étant dit, il est également possible d'avoir une seule personne qui réalise l'ensemble des activités, mais le temps passé sera plus long. Comparé au temps d'implémentation requis par d'autres approches automatiques, DEEP peut sembler plus long. En revanche, cette approche permet d'aborder les différentes spécificités : (C1) variabilité et incertitude de l'étiquetage manuel, (C2) amélioration de l'ambiguïté entre certaines paires d'étiquettes/séquences et (C3) prise en compte de l'évolution du vocabulaire. Les résultats ont montré que notre approche valide l'hypothèse pour laquelle l'étiquetage de séquences et la considération du schéma organisationnel permettent de répondre à

la question d'extraction d'information. Les approches à base de règles pour la normalisation de type primitif ont permis d'avoir un taux de précision de 89%. Quant à la normalisation de type référence qui utilise une approche par classification, le taux de précisions oscille entre 50% et 91 % selon les catégories de métiers.

Cette contribution a des implications scientifiques et applicatives. D'un point de vue scientifique, DEEP contribue au développement du domaine de l'extraction d'entités :

— en proposant une méthodologie générique et multi-acteurs conçue pour l'extraction d'entités,

— en concevant la création d'un guide d'annotation qui n'implique que des annotateurs non experts, avec le double objectif suivant : gain de temps et garantie d'une annotation manuelle de qualité,

— en gérant les modèles organisationnels qui améliorent la qualité de l'extraction automatique des entités,

— en proposant l'étiquetage des séquences comme approche pour l'extraction des entités qui gère l'évolution du vocabulaire, l'ambiguïté du vocabulaire et les fautes d'orthographe, qui sont trois caractéristiques spécifiques des textes en langage naturel.

D'un point de vue applicatif, cette contribution propose une méthodologie générique. Elle peut donc être appliquée à de nombreux domaines : médecine, administration, etc. De plus, une fois le guide d'instruction mis en place, elle peut être utilisée par tout annotateur non expert. Dans le domaine du e-recrutement, DEEP aide le recruteur à gagner du temps en identifiant et normalisant les informations obligatoires pour la diffusion des offres d'emploi et en structurant celles-ci pour préparer les documents XML nécessaire à leurs diffusions.

Approche d'appariement par type d'informations entre deux textes rédigés en langage naturel

Pour répondre à la question **(QR2)** nous avons fait l'hypothèse que la similarité par type d'informations peut améliorer la qualité de l'appariement tout en permettant une transparence pour l'explication des résultats (voir Section 3). Pour cela, nous avons proposé une approche qui utilise DEEP pour identifier, extraire et normaliser les séquences des textes. La distance entre deux textes pour apparier deux textes est transformée en distance entre chaque séquence associée à chaque étiquette pour chaque texte. Notre travail s'est aussi basé sur l'entraînement d'un modèle de similarité sémantique en ayant comme base des modèles déjà pré-entraînés utilisant le modèle de langage BERT. Ces modèles pré-entraînés le sont sur des données non spécifiques à des domaines, entraînant une similarité moins qualitative. Par conséquent, nous avons proposé une approche pour construire automatiquement des paires de séquences similaires et non similaires pour compléter ce modèle d'entraînement. Cette approche se base sur une approche automatique qui utilise des référentiels existants dans le domaine applicatif souhaité.

Cette approche d'appariement a été validée sur un corpus d'offres d'emploi et de CV et sur des séquences de métiers et compétences (voir Section 5.3). Nous avons noté

que grâce à la représentation vectorielle entraînée sur un corpus de données spécifique au recrutement, la distance sémantique entre les séquences d'offres d'emploi ou de CV se rapproche plus des distances réelles attribuées par le recruteur. De plus, cette approche permet au recruteur d'attribuer un poids plus important aux caractéristiques du profil candidat qu'il recherche. Nous avons aussi noté que l'attribution d'un score par type d'information permet au recruteur de mieux comprendre l'appariement.

Cette contribution a des implications scientifiques et applicatives. D'un point de vue scientifique, l'approche contribue au développement du domaine de l'appariement de textes :

— en proposant une méthodologie basée sur une similarité par type d'informations,

— en concevant une méthode de représentation vectorielle sémantique qui se base sur une création de corpus automatique,

— en gérant l'importante des séquences dans le calcul de la distance entre les textes.

D'un point de vue applicatif, cette contribution propose une méthodologie générique. Elle peut donc être appliquée à de nombreux domaines : médecine, administration, etc. De plus, la création du corpus pour la représentation vectorielle sémantique peut être utilisée dans différents domaines à condition d'avoir un référentiel qui représente le vocabulaire du domaine d'application. Dans le domaine du e-recrutement, cette approche aide le recruteur à gagner du temps en éliminant les CV qui ne répondent pas du tout au profil recherché par le recruteur. De plus, elle permet aussi d'inclure, comme indicateur de performance d'un canal, le nombre de CV pertinents.

ADE^2 : un système d'Aide à la Décision dans un Environnement Évolutif

Pour répondre à la question **(QR3)** nous avons conçu ADE^2 un système d'aide à la décision qui s'adapte à un environnement évolutif (voir Chapitre 4). Notre approche a pour objectif de répondre à des caractéristiques de problème d'aide à la décision dans un environnement évolutif et incertain. Pour cela, nous avons proposé un système d'aide à la décision qui s'appuie sur différents modules. Notre premier module de **traitement de l'information** a pour objectif d'exploiter les informations contenues dans le contexte décisionnel textuel **(QR3-C4)** (voir Section 4.2). Pour cela, DEEP est utilisée afin de simplifier le traitement de ce contexte et en extraire les informations nécessaires pour caractériser le contexte. Pour la caractéristique **(QR3-C2)** une création de classes sources a été définie afin de générer des groupes de données sémantiquement proches et contribuer à l'agrégation de séries chronologiques afin d'anticiper leur prévision et prédire les paramètres qui influenceront la recommandation d'items pour le décideur dans le module d'**apprentissage supervisé** (voir Section 4.3). Le module de **filtrage** a été proposé (voir Section 4.4) afin de ne garder que les items ayant des caractéristiques semblables au contexte de décision et de limiter l'espace de recherche des items répondant aux objectifs du décideur **(QR3-C3)** dans le module d'**optimisation** (voir Section 4.5). Étant donné l'environnement incertain et évolutif **(QR3-C1)** ADE^2 met à jour ses recommandations à chaque nouvelle donnée réelle de paramètre grâce au module de **renforcement** pour améliorer les recommandations pour les périodes à venir (voir Section 4.6).

Nous avons expérimenté ADE^2 dans le cadre du e-recrutement (voir Section 5.4). Pour vérifier que le système d'aide à la décision répond aux questions industrielles **(QI-1)** et **(QI-2)**, nous avons comparé les résultats obtenus suite à la mise en place d'une campagne de recrutement par un manager des campagnes et les résultats suite à la recommandation des canaux par le système d'aide à la décision. Nos expérimentations ont montré que le système d'aide à la décision permet un gain de temps au recruteur sur (1) la préparation des données pour la diffusion des offres d'emploi sur les canaux, (2) l'analyse des données anciennes, (3) l'analyse des données actuelles, (4) la prise de décision. Ce système permet aussi un gain d'argent, puisque la prédiction des paramètres à l'aide de l'apprentissage supervisé et le système de renforcement qui repose sur une correction permanente des données permettent d'économiser de l'argent sur les périodes où les objectifs du recruteur ne peuvent pas être atteint.

Cette contribution a des implications scientifiques et applicatives. D'un point de vue scientifique, l'approche contribue au développement du domaine des systèmes d'aide à la décision dans un environnement évolutif :

— en proposant une approche qui considère aussi des contextes de décision textuels rédigés en langage naturel,

— en proposant une approche qui crée des classes sources de contexte sémantiquement similaires pour pallier le manque de données des évènements historiques qui influencent la décision du décideur,

— en utilisant une architecture hybride de réseaux de neurones convolutifs et récurrents permettant de considérer dans un même modèle l'ensemble des items et des contextes de décision afin d'être aussi en mesure de capturer les tendances et les corrélations implicites entre les items,

— en proposant une approche qui se base sur une optimisation multi-objectifs dont les paramètres sont évolutifs.

D'un point de vue applicatif, cette contribution propose une méthodologie générique. Elle peut donc être appliquée à de nombreux domaines : bourse, publicité, etc. Dans le domaine du e-recrutement, cette approche aide le recruteur à avoir une vision précise des coûts de recrutement par profil recherché. Elle permet aussi un gain de temps et d'argent puisqu'elle recommande les items les plus adéquats à ses objectifs, contraintes et contextes de décision.

3 Perspectives de recherche

3.1 Perspectives à court terme

DEEP : Une méthodologie pour l'extraction d'entités en se basant sur le schéma organisationnel à partir de textes rédigés en langage naturel

Une des limites de la méthodologie DEEP est la création du corpus qui est manuelle pour assurer sa qualité. En revanche, nous nous demandons si une pré-annotation automatique permettrait de gagner du temps. Pour cela, dans nos futurs travaux, nous souhaitons vérifier cette hypothèse.

Notre second axe d'amélioration de DEEP concerne la qualité de la normalisation de type référence. L'approche par classification proposée nécessite une masse de données lorsque le nombre de classes est très grand. Pourtant, très souvent les référentiels sont organisés de façon hiérarchique : sur plusieurs niveaux. De ce fait, nous visons à expérimenter une classification hiérarchique qui utilise les différents niveaux d'informations pour attribuer une classe à un texte. Cette normalisation est importante puisque la création des classes sources pour la prévision des indicateurs de performance des canaux de diffusions pour les offres d'emploi repose sur les séquences normalisées.

Approche d'appariement par type d'informations entre deux textes rédigés en langage naturel

Nous avons noté deux axes d'amélioration pour cette contribution. Tout d'abord, lors de la validation de notre approche d'appariement, nous avons noté une limite liée à l'utilisation de séquences normalisées pour le calcul de la distance entre les textes. La normalisation des séquences utilise des approches de classifications qui peuvent entraîner des erreurs lors de la prédiction de la classe normalisée associée à une séquence. Partant de ce fait, nous souhaitons détecter les erreurs de classification pour les considérer dans le calcul de la distance entre les séquences normalisées.

D'un point de vue applicatif, nous souhaitons agrandir notre corpus de données pour le modèle de représentation vectorielle en incluant d'autres référentiels métiers dans le corpus de données tels que ESCO ou ROME. Nous visons aussi à avoir une validation des paires de séquences et leurs distances sémantiques par un expert et valider notre modèle sur un corpus plus grand.

3.2 Perspectives à long terme

ADE^2 : un système d'Aide à la Décision dans un Environnement Évolutif

D'un point de vue scientifique, nous visons à exploiter le choix du décideur quant aux items recommandés comme feedback implicite. Ce feedback peut être utilisé afin d'améliorer les recommandations et inclure dans le système d'aide à la décision les préférences implicites du décideur.

Dans un second temps, comme pour l'approche d'appariement, nous souhaitons détecter les erreurs de prévisions des modèles de prédictions afin d'améliorer les recommandations.

Enfin, nous souhaitons aller plus loin que la considération des objectifs de maximisation et minimisation. Pour cela, nous aspirons à aller vers un système d'aide à la décision interactif qui aide le décideur à faire sa sélection en fonction de plusieurs critères. De ce fait, nous envisageons d'introduire un module d'analyse multi-critères. Par leur manière d'intégrer tout type de critères, cette approche semble mieux permettre de se diriger vers un judicieux compromis plutôt qu'un optimum. L'ajout de ce module entraîne plusieurs défis scientifiques. Tout d'abord il sera important de

construire des bases de connaissance qui intègre les différents critères du décideur. Une des approches les plus connues aujourd'hui est l'approche de graphes de connaissance. Dans un second temps, nous supposons que l'intégration de pondération dans le système de filtrage et d'apprentissage supervisé sera nécessaire.

D'un point de vue applicatif, nous visons à utiliser le modèle d'appariement de textes afin d'inclure, comme indicateur de performance des canaux, le nombre de CV pertinents. Nous souhaitons aussi ajouter une fonctionnalité au système d'aide à la décision qui consiste à suggérer les mots clés sur la base du profil identifié à partir de l'offre d'emploi. Ces mots clés peuvent être insérés par le recruteur dans l'offre d'emploi afin d'accroître la visibilité de celle-ci sur les canaux.

Bibliographie

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, page 420–434, Berlin, Heidelberg, 2001. Springer-Verlag.

[2] Maroi Agrebi. *Méthodes d'aide à la décision multi-attribut et multi-acteur pour résoudre le problème de sélection dans un environnement certain/incertain : cas de la localisation des centres de distribution*. Theses, Université de Valenciennes et du Hainaut-Cambresis ; Université de Sfax (Tunisie), April 2018.

[3] Fatima Al-Aswadi, Huah Chan, and Keng Hoon Gan. Automatic ontology construction from text : a review from shallow to deep learning trend. *Artificial Intelligence Review*, pages 1–28, 11 2019.

[4] Ali Amer and Hassan Abdalla. A set theory based similarity measure for text clustering and classification. *Journal of Big Data*, 7, 09 2020.

[5] G.Ch. Nabibekova and. Electronic demography decision making system. *PROBLEMS IN PROGRAMMING*, (2-3) :228–236, sep 2020.

[6] Pierre-Emmanuel Arduin. *Forum Jeunes Chercheuses Jeunes Chercheurs : Actes de la 10e edition*. 2020.

[7] Ron Artstein. Inter-annotator agreement. In *Handbook of linguistic annotation*, pages 297–313. Springer, 2017.

[8] Otmane Azeroual, Gunter Saake, Mohammad Abuosba, and Joachim Schöpfel. Text data mining and data quality management for research information systems in the context of open data and open science. In *ICOA 2018 3e colloque international sur le libre accès*, Actes du 3e colloque international sur le libre accès. Le libre accès à la science : fondements, enjeux et dynamiques, Rabat, Morocco, November 2018. ESI Rabat.

[9] Luis B and Julián L. From pearson to spearman. *Revista Colombiana de Ciencias Pecuarias*, 20 :183–192, 06 2007.

[10] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999. 89, 138.

[11] Sophie Baillargeon, Simon Hallé, and Christian Gagné. Stream clustering of tweets. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1256–1261, 2016.

[12] Nizar Bel Hadj Ali. *Etude de la conception globale des structures en Construction Métallique - Optimisation par les Algorithmes Génétiques*. PhD thesis, Université de Savoie, 10 2003.

[13] Sidahmed Benabderrahmane, Nedra Mellouli, Myriam Lamolle, and Nada Mimouni. When deep neural networks meet job offers recommendation. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 223–230, 2017.

[14] Vedant Bhatia, Prateek Rawat, Ajit Kumar, and Rajiv Ratn Shah. End-to-end resume parsing and finding candidates for a job description using bert. *ArXiv*, abs/1910.03089, 2019.

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(null) :993–1022, March 2003.

[16] Victoria Bobicev and Marina Sokolova. Inter-annotator agreement in sentiment analysis : Machine learning perspective. In *RANLP*, pages 97–102, 2017.

[17] Emmanuelle Boch. Le marketing RH et le recrutement des cadres : comment valoriser le service d'un cabinet de recrutement ? Master's thesis, Inopia Finance, 37 rue de la Cousinerie, 59491 Villeneuve-d'Ascq, 2015.

[18] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3) :307–327, 1986.

[19] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[20] Jane Bromley, Isabelle Guyon, Yann Lecun, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *International Journal of Pattern Recognition and Artificial Intelligence - IJPRAI*, volume 7, pages 737–744, 01 1993.

[21] Duy Bui and Qing Zeng-Treitler. Learning regular expressions for clinical text classification. *Journal of the American Medical Informatics Association*, 21(5) :850–857, 02 2014.

[22] Luis Adrián Cabrera-Diego, Barthélémy Durette, Juan-Manuel Torres-Moreno, and Marc El-Bèze. How can we measure the similarity between résumés of selected candidates for a job ? In *DMIN 2015 International Conference on Data Mining, Las Vegas, Nevada USA*, 07 2015.

[23] jean-pierre Cambus, J. Graeve, E. Rogari, and P. Valdiguié. Systèmes d'aide à la décision et qualité. *Revue Française des Laboratoires*, N° 284 :p. 68–70, 1996.

[24] Annette Casagrande, Fabrizio Gotti, and Guy Lapalme. Cerebra, un système de recommandation de candidats pour l'e-recrutement. In *AISR2017*, Paris, France, May 2017.

[25] Emilia Castańo, Isabel Verdaguer, and Joseph Hilferty. Using metaphor to explore the organizational patterns of expository writing. *Cuadernos de Investigación Filológica*, 46, 03 2019.

[26] N. Chaiyaratana and A.M.S. Zalzala. Hybridisation of neural networks and genetic algorithms for time-optimal control. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 1, pages 389–396 Vol. 1, 1999.

[27] Dhivya Chandrasekaran and Vijay Mago. Comparative analysis of word embeddings in assessing semantic similarity of complex sentences. *Preprint submitted to IEEE Access*, 2021.

[28] Dhivya Chandrasekaran and Vijay Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys*, 54(2) :1–37, Apr 2021.

[29] Surajit Chattopadhyay, Deepak Jhajharia, and Goutami Chattopadhyay. Trend estimation and univariate forecast of the sunspot numbers : Development and comparison of arma, arima and autoregressive neural network models. *Comptes Rendus Geoscience*, 343(7) :433–442, 2011.

[30] Qiaojun Chen and Dong Li. Improved ctr prediction algorithm based on lstm and attention. In *Proceedings of the 5th International Conference on Control Engineering and Artificial Intelligence*, CCEAI 2021, page 122–125, New York, NY, USA, 2021. Association for Computing Machinery.

[31] Weiling Chen, C. Yeo, C. Lau, and Bu-Sung Lee. Leveraging social media news to predict stock index movement using rnn-boost. *Data Knowledge Engineering*, 118 :14–24, 2018.

[32] Yihua Chen, Eric K. Garcia, Maya R. Gupta, A. Rahimi, and L. Cazzanti. Similarity-based classification : Concepts and algorithms. *Journal of Machine Learning Research*, 10 :747–776, 2009.

[33] Emil St Chifu, Viorica Rozina Chifu, Iulia Popa, and Ioan Salomie. A system for detecting professional skills from resumes written in natural language. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 189–196. IEEE, 2017.

[34] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[35] Yann Collette and Patrick Siarry. *Optimisation multiobjectif : Algorithmes*. 2011.

[36] Alain Cucchi. Revue de thèse : “ L’intégration des médias sociaux dans les stratégies d’e-GRH : le cas du recrutement ”, par Aurélie Girard, October 2013.

[37] Punitavathi D, Shinu V, Siva S, and Vidhya P. Online job and candidate recommendation system. *International Research Journal of Multidisciplinary Technovation*, 1 :84–89, 03 2019.

[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm : Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2) :182–197, 2002.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[40] Marco Dinarelli and Isabelle Tellier. New recurrent neural network variants for sequence labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 155–173. Springer, 2016.

[41] Sonal Dixit and Nishchal Verma. Intelligent condition based monitoring of rotary machines with few samples. *IEEE Sensors Journal*, 20 :14337–14346, 07 2020.

[42] Shaokang Dong, Zijian Lei, Pan Zhou, Kaigui Bian, and Guanghui Liu. Job and candidate recommendation with big data support : A contextual online learning approach. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–7, 2017.

[43] Bruno Samways dos Santos, Maria Teresinha Arns Steiner, Amanda Trojan Fenerich, and Rafael Henrique Palma Lima. Data mining and machine learning techniques applied to public health problems : A bibliometric analysis from 2009 to 2018. *Computers and Industrial Engineering*, 138 :106120, 2019.

[44] Khadim Dramé, Gayo Diallo, Fleur Delva, Jean François Dartigues, Evelyne Mouillet, Roger Salamon, and Fleur Mougin. Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology : An application to alzheimer's disease. *Journal of Biomedical Informatics*, 48 :171 – 182, 2014.

[45] Dmitriy Drusvyatskiy, Nathan Krislock, Yuen-Lam Voronin, and Henry Wolkowicz. Noisy euclidean distance realization : Robust facial reduction and the pareto frontier. *SIAM J. Optim.*, 27 :2301–2331, 2017.

[46] Gabriel Dulac-Arnold, Daniel J. Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *ArXiv*, abs/1904.12901, 2019.

[47] Nicolas Durand. *Algorithmes Génétiques et autres méthodes d'optimisation appliqués à la gestion de trafic aérien*. Habilitation à diriger des recherches, INPT, November 2004.

[48] Marc Ehrig and York Sure. Ontology mapping – an integrated approach. In Christoph J. Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *The Semantic Web : Research and Applications*, pages 76–91, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[49] Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4) :987–1007, 1982.

[50] Jhonathan Espino and Rosa-María González. Text mining and its techniques, applications : An overview. *IJARCCE*, 10 :49–53, 05 2021.

[51] Yannick Fondeur. Systèmes d'emploi et pratiques de recrutement. *La Revue de l'IRES*, pages 31–43, March 2013.

[52] Luisa Franchina and Federico Sergiani. High quality dataset for machine learning in the business intelligence domain. In *Proceedings of SAI Intelligent Systems Conference*, pages 391–401. Springer, 2019.

[53] Hongchang Gao, Deguang Kong, Miao Lu, Xiao Bai, and Jian Yang. Attention convolutional neural network for advertiser-level click-through rate forecasting. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1855–1864, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[54] Adem Golec and Esra Kahya. A fuzzy model for competency-based employee evaluation and selection. *Computers Industrial Engineering*, 52(1) :143–161, 2007.

[55] J.C. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67 :81–97, 1985.

[56] Bill Grabe. Using discourse patterns to improve reading comprehension. In *JALT2002 AT SHIZUOKA*, 2003.

[57] Zaven Hakopov, Dmitry Mironov, Dobrica Savic, and Yulia Svetashova. Automated kos-based subject indexing in INIS. In Philipp Mayr, Douglas Tudhope, Joseph A. Busch, Koraljka Golub, Marjorie M. K. Hlava, and Marcia Zeng, editors, *Proceedings of the 18th European Networked Knowledge Organization Systems (NKOS) Workshop co-located with the 22nd International Conference on Theory and Practice of Digital Libraries 2018 (TPDL 2018), Porto, Portugal, September 13, 2018*, volume 2200 of *CEUR Workshop Proceedings*, pages 17–28. CEUR-WS.org, 2018.

[58] Mohammad Hamdan. A heterogeneous framework for the global parallelisation of genetic algorithms. *International Arab Journal of Information Technology*, 5 :192–199, 04 2008.

[59] James D. Hamilton. *Time Series Analysis*. Princeton University Press, 1 edition, January 1994.

[60] Hélène Hamisultane. ECONOMETRIE DES SERIES TEMPORELLES. Licence. France. 2002. September 2002.

[61] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8) :1735–1780, 11 1997.

[62] Ann R. Horowitz. Loss functions and public policy. *Journal of Macroeconomics*, 9(4) :489–504, 1987.

[63] Baotian Hu, Zhengdong Lu, Hang Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, 2014.

[64] Zexin Hu, Yiqi Zhao, and Matloob Khushi. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1), 2021.

[65] Rajni Jain. *Decision Support Systems : an Overview*, pages 42–50. 01 2016.

[66] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *Deep Learning*, pages 403–460. Springer US, New York, NY, 2021.

[67] C. Jareanpon, Wanida Kanarkard, R.J. Frank, and N. Davey. An adaptive rbf network optimised using a genetic algorithm applied to rainfall forecasting. volume 2, pages 1005 – 1010 vol.2, 11 2004.

[68] Bin Ji, Rui Liu, Shasha Li, Jie Yu, Qingbo Wu, Yusong Tan, and Jiaju Wu. A hybrid approach for named entity recognition in chinese electronic medical record. *BMC Medical Informatics and Decision Making*, 19, 04 2019.

[69] Bin-Bin Jia and Min-Ling Zhang. Multi-dimensional classification via decomposed label encoding. *IEEE Transactions on Knowledge and Data Engineering*, page 1, 2021.

[70] Nicolas Jozefowiez. *Optimisation combinatoire multi-objectif : des méthodes aux problèmes, de la Terre à (presque) la Lune*. Habilitation à diriger des recherches, Institut National Polytechnique de Toulouse (INP Toulouse), December 2013.

[71] Janyl Jumadinova, Oliver Bonham-carter, Hanzhong Zheng, Michael Camara, and Dejie Shi. A novel framework for biomedical text mining. *Journal on Big Data*, 2 :145–155, 01 2020.

[72] Ali Karimnezhad and Fahimeh Moradi. Bayes, e-bayes and robust bayes prediction of a future observation under precautionary prediction loss functions with applications. *Applied Mathematical Modelling*, 40(15) :7051–7061, 2016.

[73] Takuma Kato, Kaori Abe, Hiroki Ouchi, Shumpei Miyawaki, Jun Suzuki, and Kentaro Inui. Embeddings of label components for sequence labeling : A case study of fine-grained named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : Student Research Workshop*, pages 222–229, Online, July 2020.

[74] Tobias Keim. Extending the applicability of recommender systems : A multilayer framework for matching human resources. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 169–169, 2007.

[75] Rémy Kesler, Juan-Manuel Torres-Moreno, and Marc El-Bèze. E-gen : traitement automatique d'informations de ressources humaines. *Document numérique*, 13 :95–119, 12 2010.

[76] Rémy Kessler, Nicolas Béchet, Mathieu Roche, Juan-Manuel Torres-Moreno, and Marc El-Bèze. A hybrid approach to managing job offers and candidates. *Information Processing and Management*, 48(6) :1124 – 1135, 2012.

[77] Elena Knyazeva, Guillaume Wisniewski, and François Yvon. Apprentissage par imitation pour l'étiquetage de séquences : vers une formalisation des méthodes d'étiquetage "easy-first". In *Conference TALN*, page (12), Caen, France, 2015.

[78] Sunil Kumar Kopparapu. Automatic extraction of usable information from unstructured resumes to aid search. In *2010 IEEE International Conference on Progress in Informatics and Computing*, volume 1, pages 99–103, 2010.

[79] Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings of the 48th Association for Computational Linguistics*, pages 504–513, Uppsala, Sweden, 2010.

[80] Francesco Lazzeri, Gianmarco Bruno, Jeroen Nijhof, Alessio Giorgetti, and Piero Castoldi. Efficient label encoding in segment-routing enabled optical networks. In *2015 International Conference on Optical Network Design and Modeling (ONDM)*, pages 34–38, 2015.

[81] Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert : Unsupervised language model pre-training for french, 2020.

[82] Matthieu Le Berre. *Optimisation de déploiement et localisation de cible dans les réseaux de capteurs*. Theses, Université de Technologie de Troyes, June 2014.

[83] Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer-Verlag, New York, NY, USA, second edition, 1998.

[84] Chao Li, Zhongtian Bao, Linhao Li, and Ziping Zhao. Exploring temporal representations by leveraging attention-based bidirectional LSTM-RNNs for multimodal emotion recognition. *Information Processing and Management*, 57(3), 2020.

[85] Yuming Li, Pin Ni, and Victor Chang. An empirical research on the investment strategy of stock market based on deep reinforcement learning model. In *COMPLEXIS 2019 - Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk*, pages 52–58. SciTePress, January 2019.

[86] Zhiheng Li, Zhihao Yang, Yang Xiang, Ling Luo, YuanYuan Sun, and Hongfei Lin. Exploiting sequence labeling framework to extract document-level relations from biomedical texts. *BMC Bioinformatics*, 21, 12 2020.

[87] Jerry Chun-Wei Lin, Yinan Shao, Youcef Djenouri, and Unil Yun. Asrnn : A recurrent neural network with an attention model for sequence labeling. *Knowledge-Based Systems*, 212 :106548, 2021.

[88] Jerry Chun-Wei Lin, Yinan Shao, Ji Zhang, and Unil Yun. Enhanced sequence labeling based on latent variable conditional random fields. *Neurocomputing*, 403 :431 – 440, 2020.

[89] Jingyuan Liu, Yain-Whar Si, Defu Zhang, and Ligang Zhou. Trend following in financial time series with multi-objective optimization. *Applied Soft Computing*, 66 :149–167, 2018.

[90] Manel Maamar. *Modélisation et optimisation bi-objectif et multi-période avec anticipation d'une place de marché de prospects Internet : adéquation offre/demande*. Theses, Université Paris Saclay (COmUE), December 2015.

[91] Leandro Maciel and Rosangela Ballini. Neural networks applied to stock market forecasting : An empirical analysis. *Learning and Nonlinear Models*, 8 :3–22, 01 2010.

[92] Sumit Maheshwari, Abhishek Sainani, and P. Krishna Reddy. An approach to extract special skills to improve the performance of resume selection. In *Databases in Networked Information Systems*, pages 256–273, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[93] Saket Maheshwary and Hemant Misra. Matching resumes to jobs via deep siamese network. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 87–88, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[94] Jochen Malinowski, Tim Weitzel, and Tobias Keim. Decision support for team staffing : An automated relational recommendation approach. *Decision Support Systems*, 45(3) :429–447, 2008.

[95] Mohamed Mammeri. *Une approche d'aide multicritère à la décision pour l'évaluation du confort dans les trains : construction d'un modèle d'évaluation*. Theses, Université Paris Dauphine - Paris IX, September 2013.

[96] . Mansour, J. Mohammad, and Y. Kravchenko. Text vectorization using data mining methods. *Izvestiya SFedU Engineering sciences*, pages 154–167, 07 2021.

[97] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert : a tasty french language model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[98] Colleen McCue. Chapter 12 - risk and threat assessment. In *Data Mining and Predictive Analysis (Second Edition)*, pages 257 – 282. Butterworth-Heinemann, Boston, second edition edition, 2015.

[99] Vijay Krishna Menon, Nithin Chekravarthi Vasireddy, Sai Aswin Jami, Viswa Teja Naveen Pedamallu, Varsha Sureshkumar, and K. P. Soman. Bulk price forecasting using spark over nse data set. In *Data Mining and Big Data*, pages 137–146, Cham, 2016. Springer International Publishing.

[100] BONNIE J.F. MEYER and G. ELIZABETH RICE. The interaction of reader strategies and the organization of text. *Text - Interdisciplinary Journal for the Study of Discourse*, 2(1-3) :155–192, 1982.

[101] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. *Levenshtein Distance : Information Theory, Computer Science, String (Computer Science), String Metric, Damerau ?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press, 2009.

[102] Maria Mitrofan, Verginica Barbu Mititelu, and Grigorina Mitrofan. Towards the construction of a gold standard biomedical corpus for the romanian language. *Data*, 3(4) :53, 2018.

[103] Ian I. Mitroff and Frederick Betz. Dialectical decision theory : A meta-theory of decision-making. *Management Science*, 19(1) :11–24, 1972.

[104] Vrinda Mittal, Priyanshu Mehta, Devanjali Relan, and Goldie Gabrani. Methodology for resume parsing and job domain prediction. *Journal of Statistics and Management Systems*, 23, 07 2020.

[105] Minoo Modaresnezhad, Lakshmi Iyer, Prashant Palvia, and Vasyl Taras. Information technology (it) enabled crowdsourcing : A conceptual framework. *Information Processing and Management*, 57(2) :102135, 2020.

[106] Frederick Mosteller. Stochastic models for the learning process. *Proceedings of the American Philosophical Society*, 102(1) :53–59, 2021/07/17/ 1958.

[107] Kumpati Narendra and M.A.L. Thathachar. Learning automata-a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-4, 07 1974.

[108] Friska Natalia, Dea Cheria, and Santi Surya. An ontology-based approach to diagnosis and classification for an expert system in health and food. In *Ontology in Information Science*. IntechOpen, 2019.

[109] Roberto Navigli. Natural language understanding : Instructions for (present and future) use. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5697–5702. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[110] Anastasios Nentidis, Anastasia Krithara, Grigorios Tsoumakas, and Georgios Paliouras. Beyond mesh : Fine-grained semantic indexing of biomedical literature based on weak supervision. *Information Processing and Management*, 57(5) :102282, 2020.

[111] S. Nersisyan, Vera V. Pankratieva, V. Staroverov, and V. E. Podolskii. A greedy clustering algorithm based on interval pattern concepts and the problem of optimal box positioning. *Journal of Applied Mathematics*, 2017 :4323590 :1–4323590 :9, 2017.

[112] Lina Ni, Yujie Li, Xiao Wang, Jinquan Zhang, Jiguo Yu, and Chengming Qi. Forecasting of forex time series data based on deep learning. *Procedia Computer Science*, 147 :647–652, 2019.

[113] Nobal Niraula, Rajendra Banjade, Dan Ştefănescu, and Vasile Rus. Experiments with semantic similarity measures based on lda and lsa. In *Statistical Language and Speech Processing*, pages 188–199, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[114] A. Osyczka and S. Kundu. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural optimization*, 10(2) :94–99, Oct 1995.

[115] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Li Li, Zhaojie Liu, and Yanlong Du. Click-through rate prediction with the user memory network. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, DLP-KDD '19, New York, NY, USA, 2019. Association for Computing Machinery.

[116] Weifeng Pan, Kangshun Li, Muchou Wang, Jing Wang, and Bo Jiang. Adaptive randomness : A new population initialization method. *Mathematical Problems in Engineering*, pages 1–14, 2014.

[117] Kwangil Park, June Seok Hong, and Wooju Kim. A methodology combining cosine similarity with classifier for text classification. *Applied Artificial Intelligence*, 34(5) :396–411, 2020.

[118] Fei Qian and X. Chen. Stock prediction based on lstm under different stability. *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 483–486, 2019.

[119] H Ramdani, A Brun, E Bonjour, and D Monticolo. Définition d'une méthodologie d'indexation de documents textuels par étiquetage de séquences : application aux offres d'emploi. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle - APIA 2020*, Angers, France, June 2020.

[120] Halima Ramdani, Davy Monticolo, Armelle Brun, and Eric Bonjour. Decision support system for online recruitment. In Inès Saad, Camille Rosenthal-Sabroux, Faiez Gargouri, and Pierre-Emmanuel Arduin, editors, *Information and Knowledge Systems. Digital Technologies, Artificial Intelligence and Decision Making*, pages 43–51, Cham, 2021. Springer International Publishing.

[121] Alan Ramponi, Rob van der Goot, Rosario Lombardo, and Barbara Plank. Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5357–5367, 2020.

[122] Nils Reimers and Iryna Gurevych. Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[123] Michael Reusens, Wilfried Lemahieu, Bart Baesens, and Luc Sels. A note on explicit versus implicit information for job recommendation. *Decision Support Systems*, 98 :26–35, 2017.

[124] Stephen Robertson. Understanding inverse document frequency : On theoretical arguments for idf. *Journal of Documentation - J DOC*, 60 :503–520, 10 2004.

[125] Bernard Roy and Denis Bouyssou. *Aide Multicritère à la Décision : Méthodes et Cas*. 09 1993.

[126] Herbert Rubenstein and John Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8 :627–633, 10 1965.

[127] Thomas L Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3) :234–281, 1977.

[128] Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. Keyphrase extraction as sequence labeling using contextualized embeddings. In *Advances in Information Retrieval*, pages 328–335, Cham, 2020. Springer International Publishing.

[129] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11) :2673–2681, 1997.

[130] Julie Séguéla and Gilbert Saporta. A hybrid recommender system to predict online job offer performance. *Revue des Nouvelles Technologies de l'Information*, RNTI -E-25 :177–197, 2013.

[131] Sreelekshmy Selvin, R. Vinayakumar, E. Gopalakrishnan, V. Menon, and K. Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647, 2017.

[132] Zahra Ghorbani Shemshadsara, Touran Ahour, and Nasrin Hadidi Tamjid. Raising text structure awareness : A strategy of improving efl undergraduate students' reading comprehension ability. *Cogent Education*, 6, 2019.

[133] Ze Shen, Qing Wan, and David Leatham. Bitcoin return volatility forecasting : A comparative study between garch and rnn. *Journal of Risk and Financial Management*, 14 :337, 07 2021.

[134] Hong-Gi Shin, Ilkyeun Ra, and Yong-Hoon Choi. A deep multimodal reinforcement learning system combined with cnn and lstm for stock trading. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 7–11, 2019.

[135] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018.

[136] B. Sidahmed, N. Mellouli, and M. Lamolle. On the predictive analysis of behavioral massive job data using embedded clustering and deep recurrent neural networks. *Knowledge-Based Systems*, 151 :95–113, 2018.

[137] B. Sidahmed, N. Mellouli, M. Lamolle, and P. Paroubek. Smart4Job : A Big Data Framework for Intelligent Job Offers Broadcasting Using Time Series Forecasting and Semantic Classification. *Big Data Research*, 7 :16–30, mar 2017.

[138] Sahbi Sidhom. *Morpho-syntactic Analysis Platform for Automatic indexing and Information retrieval :from text-writing to knowledge management*. PhD Thesis, Université Claude Bernard - Lyon I, France, March 2002.

[139] Herbert A. (Herbert Alexander) Simon. *The new science of management decision*. New York : Harper, [1st ed.] edition, 1960.

[140] Amit Singh, Catherine Rose, Karthik Visweswariah, Vijil Chenthamarakshan, and Nandakishore Kambhatla. Prospect : A system for screening candidates for recruitment. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 659–668, New York, NY, USA, 2010. Association for Computing Machinery.

[141] Y. Song and J. Lee. A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. *Applied Intelligence*, 49 :897–911, 2018.

[142] Ralph H. Sprague and Eric D. Carlson. Prentice Hall Professional Technical Reference, 1982.

[143] S. Sulova, L. Todoranova, B. Penchev, and R. Nacheva. USING TEXT MINING TO CLASSIFY RESEARCH PAPERS. In *17th International Multidisciplinary Scientific GeoConference SGEM 2017*, volume 17 of *International Multidisciplinary Scientific GeoConference-SGEM*, pages 647–654. STEF92 Technology, 29 June - 5 July, 2017 2017.

[144] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[145] Oleg Sychev and N Penskoy. Method of lemmatizer selections in multiplexing lemmatization. *IOP Conference Series : Materials Science and Engineering*, 483, 03 2019.

[146] Wen Tai, H. T. Kung, Xin Dong, Marcus Comiter, and Chang-Fu Kuo. exBERT : Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Findings of the Association for Computational Linguistics : EMNLP 2020*, pages 1433–1439, Online, November 2020. Association for Computational Linguistics.

[147] Ebtesam Taktek and Dhavalkumar Thakker. Pentagonal scheme for dynamic xml prefix labelling. *Knowledge-Based Systems*, 209 :106446, 2020.

[148] Sarthak Tiwari, Bharat Goel, and Srividya Bansal. Mold - a framework for entity extraction and summarization. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 445–450, 2020.

[149] Yaşar Tonta and Hamid Darvish. Diffusion of latent semantic analysis as a research tool : A social network analysis approach. *Journal of Informetrics*, 4 :166–174, 04 2010.

[150] Alexis Tsoukiàs. From decision theory to decision aiding methodology. *European Journal of Operational Research*, 187(1) :138–161, 2008.

[151] Wendi Usino, Anton Satria Prabuwono, Khalid Allehaibi, Arif Bramantoro, A. Hasniaty, and Wahyu Amaldi. Document similarity detection using k-means and cosine distance. *International Journal of Advanced Computer Science and Applications*, 10, 01 2019.

[152] Senthil kumaran V and Sankar Annamalai. Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping expert. *International Journal of Metadata, Semantics and Ontologies*, 8 :56–64, 05 2013.

[153] D. Vallet, M. Fernández, and P. Castells. An Ontology-Based Information Retrieval Model. In *The Semantic Web : Research and Applications*, pages 455–470, Berlin, Heidelberg, 2005.

[154] Ronny Velastegui, Luis Zhinin-Vera, Gissela E. Pilliza, and Oscar Chang. Time series prediction by using convolutional neural networks. In *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1*, pages 499–511, Cham, 2021. Springer International Publishing.

[155] Chien-Chih Wang, Chun-Hua Chien, and Amy J. C. Trappey. On the application of arima and lstm to predict order demand based on short lead time and on-time delivery requirements. *Processes*, 9(7), 2021.

[156] Kang Wang, Kenli Li, Liqian Zhou, Yikun Hu, Zhongyao Cheng, Jing Liu, and Cen Chen. Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing*, 360 :107–119, 2019.

[157] Qianqian Wang, Fang'ai Liu, Shuning Xing, Xiaohui Zhao, and Tianlai Li. Research on ctr prediction based on deep learning. *IEEE Access*, PP :1–1, 12 2018.

[158] Christopher Watkins and Peter Dayan. Technical note : Q-learning. *Machine Learning*, 8 :279–292, 05 1992.

[159] Xiaojing Wu, Xingsi Xue, and Wenyu Hu. Argumentation based ontology alignment extraction. In *Advanced Machine Learning Technologies and Applications*, pages 1028–1037, Cham, 2021. Springer International Publishing.

[160] Leila Yahiaoui, Zizette Boufaida, and Yannick Prié. Automatisation du e-recrutement dans le cadre du Web sémantique. In *IC - 17èmes Journées francophones d'Ingénierie des Connaissances*, pages 51–60, Nantes, France, June 2006.

[161] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, pages 87–94, Online, July 2020. Association for Computational Linguistics.

[162] Kyongmin Yeo, Igor Melnyk, Nam Nguyen, and Eun Kyung Lee. De-rnn : Forecasting the probability density function of nonlinear time series. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 697–706, 2018.

[163] Murat Yildizoglu and Thomas Vallée. Présentation des algorithmes génétiques et de leurs applications en économie. *Revue D Economie Politique*, 114 :711–745, 2004.

[164] Pengfei Yu and Xuesong Yan. Stock price prediction based on deep neural networks. *Neural Computing and Applications*, 32, 03 2020.

[165] Yusliza Yusoff, Mohd Salihin Ngadiman, and Azlan Mohd Zain. Overview of nsga-ii for optimizing machining process parameters. *Procedia Engineering*, 15 :3978–3983, 2011. CEIS 2011.

[166] Zhiwen Zeng and Matloob Khushi. Wavelet denoising and attention-based rnn-arima model to predict forex price. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2020.

[167] Sami Zghal, S. Ben Yahia, Engelbert Mephu Nguifo, and Y. Slimani. SODA : Une approche structurelle pour l'alignement d'ontologies OWL-DL. In *1ères Journées Francophones sur les Ontologies (JFO'07)*, Sousse, Tunisia.

[168] Pu Zhao, Chuan Luo, Cheng Zhou, Bo Qiao, Jiale He, Liangjie Zhang, and Qingwei Lin. Rlnf : Reinforcement learning based noise filtering for click-through rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2268–2272, New York, NY, USA, 2021. Association for Computing Machinery.

A

Exemple de guide d’instruction

Changelogs

Actor name	Actor role	Date of the changes	Description of the changes
X	*The expert*	*dd-mm-yyyy*	*Add the organizational patterns of the documents*
Y	*The master*	*dd-mm-yyyy*	*Change the general annotation rule id = 1, after the first guideline validation*

1. The corpus of documents used

Description: This section contains the description of the corpus (how many documents it contains, the source)
Type of documents: *Research article*
Manager: **The master**
Corpus of documents:

Corpus	Size (documents)	Source	Location folder
Validation	*50*	*Knowledge-Based Systems*	*/entity_extractions/validation_documents*
Final	*1000*	*Knowledge-Based Systems*	*/entity_extraction/final_documents*

2. Definition of the organizational patterns

Description: This section contains the description of the activity objective
Manager: **The expert**
Input: The validation corpus
Output:

Common order	Section	Description	Uncertainties
1	*Header*	*The header is **always** the first section of a research paper. It contains the writer's names, address etc.*	*None*
2	*Abstract*	*The abstract is **commonly** the second section in the document*	*Some documents do not contain the abstract. In consequence, the introduction is the first section*
3	*Keywords*	*The keywords is **commonly** the third section in the document*	*Some documents do not contain the keywords.*

Section	Signal words	Sentence topics
Header	*Name, "@", postal code*	*54000, Nancy, France*
Abstract	*Context, research problem, limitations, literature, objective, contribution*	*"Our main contribution", "Our work aims to"*

3. Identification of the set of labels

Description: This section contains the description of the activity objective
Manager: **The master**
Input: The validation corpus and table 1
Output:

Label name	Description	Common position on the typical-structure	Sequence example
Name	*Name of the writer*	*Section 1: Always included in the header*	*"X.Y"*
Address	*Address of the writer*	*Section 1: Always included in the header*	*"xy@gmail.com"*
Context	*The topic of the paper by providing relevant background information*	*Section 2: Commonly included in the introduction*	*"The traditional methods of recruitment require far too much paperwork. Therefore, e-recruitment could be the solution to streamline the process."*

4. Design of the annotation rules

a. Domain specific annotation rules

Description: This section contains the description of the activity objective
Manager: **The expert**
Input: The validation corpus, table 1 and table 2
Output:

Id	Rule	Label(s)	Example
S1	*Include the footnote in the sequence for the label address*	*Address*	*"xy@gmail.com* [1]*"*

b. General annotation rules

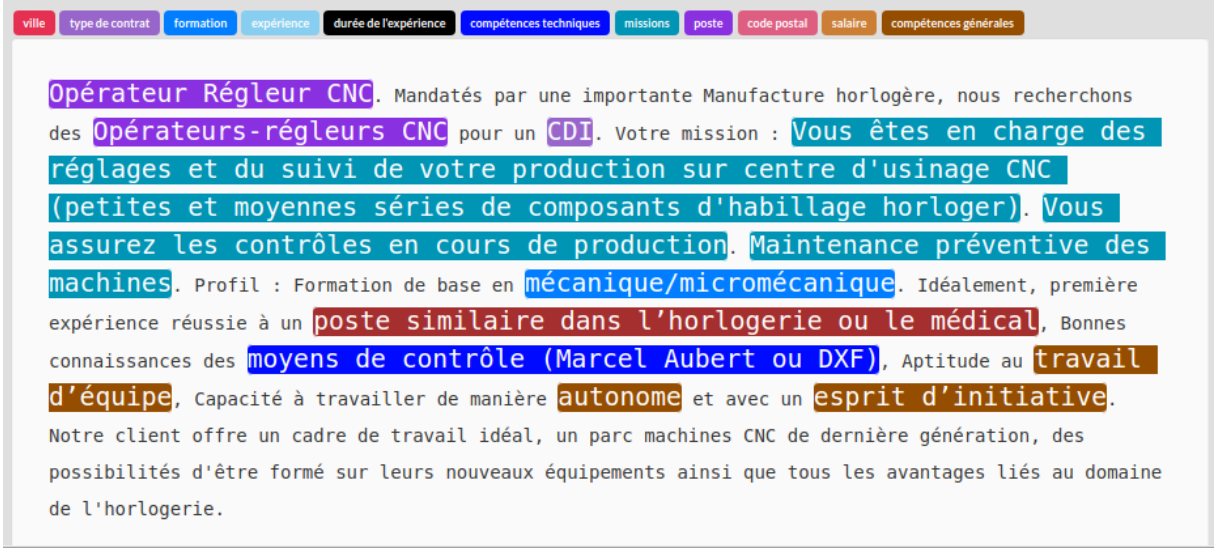
Description: This section contains the description of the activity objective
Manager: **The master**
Input: The validation corpus, table 1 and table 2
Output:

Category	Id	Rule	Example
Punctuation	*P1*	*Include the punctuations "," and "." on the sequence when it is associated to label.*	*Sequence : "Therefore, e-recruitment could be the solution to streamline the process."* *Label: Context*

Fig. A.1. Guide d'instruction pour les annotateurs

B

Exemple d’une offre d’emploi annotée manuellement à l’aide du logiciel Dataturks



C

Exemple d’extractions et de normalisation d’une offre d’emploi


```
[
  {
    "content": "Assistant comptable - pacé (h/f)\n\n\nLe poste\nEn
       Bref: Assistant Comptable (H/F) - CDI - Dynamisme - esprit
       d’équipe - autonomie - Formation - Evolution\n\nVos
       missions:\n\nAu sein du p l e expertise comptable, vous  tes
       sous la responsabilité du chef de mission et intervenez sur un
       portefeuille principalement composé de BIC et sur des
       groupements de supermarchés.\n\nVous assurez de la saisie à la
       révision de vos dossiers.\n\nLe profil recherché\nVotre
       profil:\n\nDe formation bac +2 en comptabilité, vous bénéficiez
       d’une première expérience en tant que comptable sur un poste
       similaire.\n\nVous valorisez l’esprit d’équipe, l’entraide et
       l’honn teté.\n\nDoté(-e) d’un sens client aigu, vous souhaitez
       monter en compétence et évoluer au sein d’une structure
       bienveillante et dynamique.\n\nCe qu’on vous propose:\n\n- Une
       structure dynamique et orienté vers l’humain\n\n- Une formation
       et accompagnement pouvant vous aider dans votre montée en
       compétence et dans votre perfectionnement\n\n- Un manager
       visionnaire et à l’écoute\n\nBien entendu, votre candidature
       sera étudiée avec soin et de manière confidentielle.\n\nCette
       offre vous intéresse vous pouvez postuler directement ou me
       contacter pour plus d’informations.\n\nCamille Favreau -
       Consultante en recrutement Cabinet
       ADSEARCH\n\nL’entreprise\nAdsearch Bretagne recrute pour le
       compte de son partenaire, un cabinet d’expertise et d’audit, un
       Assistant Comptable H/F. Cette opportunité en CDI est à
       pourvoir à Pacé dès que possible.\n\nCette opportunité est pour
```

```
      vous l’occasion de rejoindre une structure chaleureuse, un
      cabinet à taille humaine en pleine croissance.\n\nFort de plus
      de 600 clients, ce cabinet d’expertise comptable intervient
      dans la comptabilité, la gestion d’entreprises de tous les
      secteurs d’activités, TPE, PME, PMI, professions libérales,
      commerce, industrie, associations, BTP, services ",
  "lang": "fr",
  "created_at": "2021-09-27T14:10:51.148202",
  "profile": [
    {
      "education": [
        {
          "level": [
            {
              "plaintext": "bac +2",
              "ref": {
                "id": "4",
                "source": "xtramile_education_levels",
                "value": "bac+2",
                "score": 1
              }
            }
          ],
          "specialization": [
            {
              "plaintext": "comptabilité",
              "ref": {}
            }
          ],
          "status": [],
          "plaintext": "formation bac +2 en comptabilité"
        }
      ],
      "experience": [
        {
          "domain": [],
          "duration": [
            {
              "plaintext": "première expérience",
              "ref": {
                "id": "(0, ’year’, 0, ’year’)",
                "source": "xtramile_experience_levels",
                "value": "(0, ’year’, 0, ’year’)",
                "score": 1
              }
```

```
            }
          ],
          "hard_skills": [],
          "soft_skills": [],
          "occupations": [
            {
              "plaintext": "comptable",
              "ref": {},
              "type": [],
              "esco": "",
              "rome": ""
            },
            {
              "plaintext": "poste similaire",
              "ref": {},
              "type": [],
              "esco": "",
              "rome": ""
            }
          ],
          "plaintext": "une première expérience en tant que comptable
             sur un poste similaire"
        }
      ],
      "skills": [
        {
          "hard_skills": [],
          "soft_skills": [],
          "languages": [],
          "plaintext": "esprit d ’ équipe"
        },
        {
          "hard_skills": [],
          "soft_skills": [],
          "languages": [],
          "plaintext": "entraide et l ’ honn teté"
        },
        {
          "hard_skills": [],
          "soft_skills": [],
          "languages": [],
          "plaintext": "sens client aigu"
        },
        {
          "hard_skills": [],
```

```
            "soft_skills": [],
            "languages": [],
            "plaintext": "monter en compétence et évoluer au sein d ’
               une structure bienveillante et dynamique"
          }
        ],
        "plaintext": "Le profil recherché \n Votre profil : \n\n De
           formation bac +2 en comptabilité , vous bénéficiez d’ une
           première expérience en tant que comptable sur un poste
           similaire . \n\n Vous valorisez l’ esprit d’ équipe , l’
           entraide et l’ honn teté . \n\n Doté(-e ) d’ un sens
           client aigu , vous souhaitez monter en compétence et é
           voluer au sein d’ une structure bienveillante et dynamique
           ."
      }
    ],
    "description": [
      {
        "city": [],
        "country": [],
        "state": [],
        "contract_duration": [],
        "contract_duration_ref": {
          "plaintext": "",
          "ref": {}
        },
        "contract_type": [
          {
            "plaintext": "CDI",
            "ref": {
              "id": "0",
              "source": "xtramile_contract_type",
              "value": "CDI",
              "score": 1
            }
          }
        ],
        "date_start": [],
        "date_end": [],
        "driving_licence": [],
        "education_level": [],
        "experience_level": [],
        "function": [],
        "missions": [
```

```
          "intervenez sur un portefeuille principalement composé de BIC
             et sur des groupements de supermarchés",
          "assurez de la saisie à la révision de vos dossiers"
        ],
        "occupations": [
          {
            "plaintext": "Assistant comptable - pacé",
            "ref": {
              "id": "M1203",
              "source": "rome4_525",
              "value": "Comptabilité",
              "score": 0.98
            },
            "type": [
              {
                "id": "3313",
                "source": "esco4_404",
                "value": "Professions intermédiaires de la
                   comptabilité",
                "score": 0.74
              },
              {
                "id": "M1203",
                "source": "rome4_525",
                "value": "Comptabilité",
                "score": 0.98
              },
              {
                "id": "M12",
                "source": "rome2",
                "value": "Comptabilité et gestion",
                "score": 0.98
              },
              {
                "id": "M",
                "source": "rome1",
                "value": "Support à l’entreprise",
                "score": 0.98
              }
            ],
            "esco": "",
            "rome": ""
          },
          {
            "plaintext": "Assistant Comptable",
```

```
            "ref": {
              "id": "M1203",
              "source": "rome4_525",
              "value": "Comptabilité",
              "score": 0.96
            },
            "type": [
              {
                "id": "3313",
                "source": "esco4_404",
                "value": "Professions intermédiaires de la
                   comptabilité",
                "score": 0.47
              },
              {
                "id": "M1203",
                "source": "rome4_525",
                "value": "Comptabilité",
                "score": 0.96
              },
              {
                "id": "M12",
                "source": "rome2",
                "value": "Comptabilité et gestion",
                "score": 0.96
              },
              {
                "id": "M",
                "source": "rome1",
                "value": "Support à l’entreprise",
                "score": 0.96
              }
            ],
            "esco": "",
            "rome": ""
          }
        ],
        "plaintext": "Assistant comptable - pacé ( h / f ) \n\n\n Le
           poste \n En Bref : Assistant Comptable ( H / F ) - CDI -
           Dynamisme - esprit d’ équipe - autonomie - Formation -
           Evolution \n\n Vos missions : \n\n Au sein du p l e
           expertise comptable , vous  tes  sous la responsabilité du
           chef de mission et intervenez sur un portefeuille
           principalement composé de BIC et sur des groupements de
           supermarchés . \n\n Vous assurez de la saisie à la révision
```

```
          de vos dossiers .",
        "postal_code": [],
        "schedule": [],
        "salary": [],
        "sector": []
      },
      {
        "city": [],
        "country": [],
        "state": [],
        "contract_duration": [],
        "contract_duration_ref": {
          "plaintext": "",
          "ref": {}
        },
        "contract_type": [],
        "date_start": [],
        "date_end": [],
        "driving_licence": [],
        "education_level": [],
        "experience_level": [],
        "function": [],
        "missions": [],
        "occupations": [],
        "plaintext": "Ce qu’ on vous propose : \n\n - Une structure
          dynamique et orienté vers l’ humain \n\n - Une formation et
          accompagnement pouvant vous aider dans votre montée en
          compétence et dans votre perfectionnement \n\n - Un manager
          visionnaire et à l’ écoute \n\n Bien entendu , votre
          candidature sera étudiée avec soin et de manière
          confidentielle .",
        "postal_code": [],
        "schedule": [],
        "salary": [],
        "sector": []
      }
    ],
    "company": [
      {
        "plaintext": "L’ entreprise \n Adsearch Bretagne recrute pour
          le compte de son partenaire , un cabinet d’ expertise et d’
          audit , un Assistant Comptable H / F. Cette opportunité en
          CDI est à pourvoir à Pacé dès que possible .",
        "sectors": []
      },
```

```
      {
        "plaintext": "Cette opportunité est pour vous l’ occasion de
           rejoindre une structure chaleureuse , un cabinet à taille
           humaine en pleine croissance . \n\n Fort de plus de 600
           clients , ce cabinet d’ expertise comptable intervient dans
           la comptabilité , la gestion d’ entreprises de tous les
           secteurs d’ activités , TPE , PME , PMI , professions
           libérales , commerce , industrie , associations , BTP ,
           services    ",
        "sectors": []
      }
    ],
    "sector": {
      "plaintext": "",
      "ref": {
        "id": "3",
        "source": "xt_sector",
        "value": "Banque, Finance _ Banking, Finance",
        "score": 0.43
      },
      "type": [
        {
          "id": "3",
          "source": "xt_sector",
          "value": "Banque, Finance _ Banking, Finance",
          "score": 0.43
        }
      ]
    },
    "form": {},
    "title": {
      "plaintext": "Assistant comptable - pacé",
      "ref": {
        "id": "M1203",
        "source": "rome4_525",
        "value": "Comptabilité",
        "score": 0.98
      },
      "type": [
        {
          "id": "3313",
          "source": "esco4_404",
          "value": "Professions intermédiaires de la comptabilité",
          "score": 0.74
        },
```

```
        {
          "id": "M1203",
          "source": "rome4_525",
          "value": "Comptabilité",
          "score": 0.98
        },
        {
          "id": "M12",
          "source": "rome2",
          "value": "Comptabilité et gestion",
          "score": 0.98
        },
        {
          "id": "M",
          "source": "rome1",
          "value": "Support à l’entreprise",
          "score": 0.98
        }
      ],
      "esco": "",
      "rome": ""
    },
    "source": {
      "id": "",
      "name": "default",
      "url": ""
    }
  }
]
```

D

Exemple d’extractions et de normalisation d’un CV

E

Règles d’annotation

Table E.1
Règles d’annotation spécifiques au domaine

Catégorie de règle	Étiquettes concernées	Les règles
Spécifiques au domaine	“Compétences techniques”, “Compétences générales’, “Missions”	Annotez-les séparément, même lorsqu’ils apparaissent dans la même phrase, en conservant le sens sémantique.
Spécifiques au domaine	“Expérience”, “Compétences techniques”, “Compétences générales’, "Formation"	Inclure le statut de l’étiquette dans la séquence (obligatoire ou facultatif).
Spécifiques au domaine	“Missions”	Les principales tâches sont incluses dans la description du profil. Dans cette section, il n’y a pas de compétences, d’expérience ou de formation. Elles sont généralement décrites comme une action.
Spécifiques au domaine	“Métier”, “Type de contrat”	L’en-tête contient généralement le métier suivi de (H/F) ; ne l’annotez pas. L’en-tête contient également la ville, le type de contrat et le code postal. Annotez-les.
Spécifiques au domaine	"Compétences techniques", "Compétences générales"	Si les séquences associées contiennent une compétence dure ou souple, ne l’annotez pas comme une compétence mais comme l’étiquette à laquelle elle est liée. La compétence sera considérée comme une sous-étiquette pour une deuxième annotation.
Spécifiques au domaine	"Salaire"	Annoter le mois, le jour, l’heure suivant le salaire, mais pas le mot "par" ni les autres mots connecteurs.

Table E.2
Règles d'annotation générales

Catégorie de règle	Étiquettes concernées	Les règles
Langage naturel	"Date de début", "Date de fin"	Considérez dans la séquence associée à ces étiquettes toutes les informations de l'année, du mois, du jour.
Langage naturel	"Date de début", "Date de fin"	Si la Date de début est "dès que possible" ou un synonyme, annotez cette séquence comme date de début.
Langage naturel	"Durée de l'expérience"	Inclure dans l'annotation de la séquence le numéro et l'unité de temps.
Langage naturel	"Salaire"	Annotez le montant et l'unité.
Langage naturel	"Salaire"	N'annotez pas s'il n'est pas crypté (par exemple, "Salaire attractif" n'annotez pas).
Langage naturel	"Salaire"	Annoter le mois, le jour, l'heure à la suite du salaire mais pas le mot "par" ou d'autres mots connecteurs.
Langage naturel	"Salaire"	S'il y a plus d'un salaire selon la période de stage, annotez celui qui suit la période de stage.
Langage naturel	"Salaire"	Inclure dans la séquence associée au "Salaire" les mots "brut" et "net".
Langage naturel	Toutes les étiquettes	Ne pas considérer dans l'annotation les titres des sections.
Langage naturel	Toutes les étiquettes	Si une étiquette/séquence a déjà été annotée et apparaît une autre fois dans le document, annotez-la.
Ponctuations	Toutes les étiquettes	Considérer dans l'annotation de la séquence les informations entre parenthèses.
Ponctuations	Toutes les étiquettes	Considérer dans l'annotation de la séquence la ligne sautée pour garder le contexte.
Connecteurs de séquences	Toutes les étiquettes	S'il existe un connecteur de phrases dans une séquence qui représente la même étiquette, il est pris en compte dans l'annotation.
Connecteurs de séquences	Toutes les étiquettes	Si un connecteur de phrases est lié à deux étiquettes différentes, incluez-le dans la deuxième étiquette.

F

Fonction pour transformer les données en corpus d'apprentissage supervisé

```
#Data: étant les données à transformer.

#Window: étant la fenêtre glissante. Cela revient à faire
#glisser une fenêtre sur les observations antérieures
#qui sont utilisées comme entrées du
#modèle afin de prédire la prochaine valeur de la série.

#Lag: étant une période de combien de jour considérer.
#Si lag = 1 alors on considère un pas d'un jour
#Si lag = 7 alors on considère un pas de 7 jours

def series_to_supervised(data, window=1, lag=1, dropnan=True):

    cols, names = list(), list()
    for i in range(window, 0, -1):
        cols.append(data.shift(i))
        names += [('%s(t-%d)' % (col, i)) for col in data.columns]
    cols.append(data)
    names += [('%s(t)' % (col)) for col in data.columns]
    cols.append(data.shift(-lag))
    names += [('%s(t+%d)' % (col, lag)) for col in data.columns]
    agg = pd.concat(cols, axis=1)
    agg.columns = names
    if dropnan:
        agg.dropna(inplace=True)
    return agg
```

G

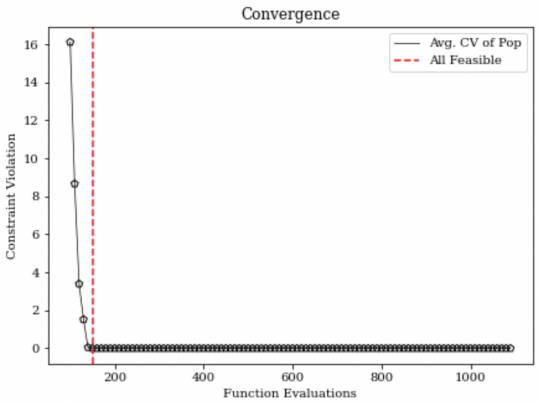
Exemple de json d’entrée pour le module d’optimisation

```
{"optimizations" : [{
                "stats": [
                       {
                           "channel" : "Adzuna",
                           "indicator" : "clics",
                           "stats_vector" :
                              [-1,-1,-1,24,468,27,115,199,338,-1]
                       },
                       {
                           "channel" : "Appnexus",
                           "indicator" : "clics",
                           "stats_vector" : [11,7,15,3,1,1,1,1, 11,0]
                       },
                       {
                           "channel" : "Facebook / Instagram",
                           "indicator" : "clics",
                           "stats_vector" : [-1,
                              479,71,52,44,39,39,26,41,-1]
                       },
                       {
                           "channel" : "indeed",
                           "indicator" : "clics",
                           "stats_vector" : [-1,-1,-1,8,6,6,8,4,1,-1]
                       },
                       {
                           "channel" : "Jobijoba",
                           "indicator" : "clics",
                           "stats_vector" :
                              [-1,-1,-1,-1,-1,-1,159,47,230,-1]
                       },
```

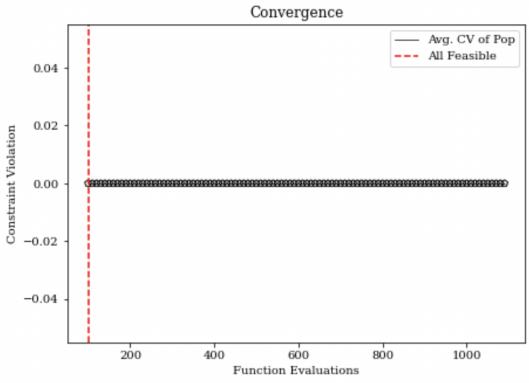
```
                    {
                        "channel" : "Joblift",
                        "indicator" : "clics",
                        "stats_vector" :
                           [-1,-1,-1,-1,-1,-1,-1,0,0,-1]
                    },
                    {
                        "channel" : "Jobrapido",
                        "indicator" : "clics",
                        "stats_vector" :
                           [-1,-1,155,387,333,500,-1,-1]
                    },
                    {
                        "channel" : "Jobtome",
                        "indicator" : "clics",
                        "stats_vector" :
                           [-1,-1,33,102,49,47,72,24,17,9]
                    },
                    {
                        "channel" : "Jooble",
                        "indicator" : "clics",
                        "stats_vector" :
                           [0,0,0,0,257,141,58,114,45,90]
                    },
                    {
                        "channel" : "Talent.com",
                        "indicator" : "clics",
                        "stats_vector" : [-1,-1,1,1,1,1,1,1,1,1]
                    }],
                "objective":
                    {
                    "category" : "max",
                    "indicator" : "clics"
                    }
                ,
                "constraints":[
                    {
                    "indicator":"budget",
                    "Category":"min",
                     "value": "34415"
                     }
                 ]
              }
             ]
        }
```

H

Optimisation multi-périodes et mono-période



(a) Solutions faisables de l'optimisation à mesure de la génération des populations pour le problème d'optimisation mono-période



(b) Solutions faisables de l'optimisation à mesure de la génération des populations pour le problème d'optimisation multi-périodes

Fig. H.1. Solutions faisables de l'optimisation à mesure de la génération des populations pour le problème d'optimisation mono-période et multi-périodes