

Cerebra, un système de recommandation de candidats pour l'e-recrutement

Annette Casagrande, Fabrizio Gotti, Guy Lapalme, *Membre, IEEE*

Résumé—Nous présentons Cerebra, un système de recommandation de candidats facilitant l'e-recrutement. Ce système a pour objectif de recommander des candidats potentiels pour une offre émise par un recruteur. Pour cela, il s'appuie sur l'extraction du titre d'emploi, des compétences et de l'expérience requise pour le poste affiché. Deux évaluations manuelles Cerebra mettent en lumière l'utilité pratique du système, ainsi que l'importance de la géolocalisation et de la prise en compte du secteur industriel dans le calcul de pertinence des candidats.

Index Terms—systèmes de recommandation, e-recrutement

I. INTRODUCTION

LES recruteurs utilisent plus que jamais les réseaux sociaux pour identifier les candidats potentiels pour un poste donné, ainsi que pour obtenir des informations sur ces candidats. Ainsi, d'après une étude de RegionsJob¹(2011), 43 % des recruteurs admettent recourir à des recherches de type nom/prénom sur les candidats qui postulent chez eux et 8 % des recruteurs interrogés déclarent avoir écarté un candidat à cause de traces jugées négatives trouvées sur Internet. Par ailleurs, dans l'étude de RegionsJob de 2015², près de 70 % des recruteurs reconnaissent avoir des difficultés à trouver les profils qui correspondent parfaitement aux postes qu'ils cherchent à combler.

Dans ce contexte, nous développons en partenariat avec la société LittleBigJob une plate-forme ayant pour objectif d'améliorer le processus de recrutement informatisé de cadres et professionnels. Cette plate-forme³, appelée Cerebra, doit permettre :

- d'améliorer la récolte et l'agrégation d'information sur des candidats provenant de multiples sources (dont les données publiques sur les médias sociaux) ;
- d'améliorer la mise en correspondance entre des candidats et des offres d'emploi ;
- de prédire le succès d'un candidat dans un nouveau poste.

LittleBigJob souhaite plus particulièrement proposer à ses clients, des entreprises en recrutement, un moteur qui facilite l'identification et l'embauche de candidats dits *passifs*. Ces candidats passifs sont déjà en poste, ne sont pas en recherche active d'emploi, mais pourraient néanmoins être intéressés par une nouvelle opportunité. Ces profils sont très convoités par les recruteurs [1]. Cependant, les recruteurs rencontrent souvent

des difficultés lors de leur recherche de candidats passifs. En effet, ces spécialistes en recrutement sont contraints de prendre l'initiative et de contacter et convaincre le candidat passif de considérer leur offre, une posture commerciale qui, d'après notre partenaire, n'est pas naturelle dans ce contexte. Ce n'est qu'une fois que le candidat a répondu que ce dernier devient *actif* et que les stratégies de recrutement habituelles peuvent être utilisées. Le projet BPP vise notamment à simplifier et à automatiser ce processus en recherchant les candidats potentiels parmi les profils de réseaux sociaux, incluant les profils passifs.

Le travail présenté ici se concentre sur cet appariement délicat, au cœur du système Cerebra, un *système de recommandation* proposant aux recruteurs une liste de candidats potentiels pour une offre d'emploi donnée. La section II présente les systèmes de recommandation et leur utilisation dans le domaine des ressources humaines. Nous décrivons ensuite notre système à la section III. Les évaluations manuelles de Cerebra ont montré que ce système peut tirer parti de diverses améliorations, touchant notamment la géolocalisation des profils et la détection du secteur d'activité de l'offre. Ces améliorations sont présentées à la section IV.

II. REVUE DE LA LITTÉRATURE

Les systèmes de recommandations aident les utilisateurs à gérer la surcharge d'information en leur proposant des éléments adaptés à leurs besoins. Pour cela, il existe trois familles d'algorithmes : ceux basés sur le *filtrage collaboratif* [2], [3], ceux sur le *filtrage du contenu* [4] et les hybrides [5]. Avant de les expliquer et de voir comment il s'appliquent au e-recrutement, notons que, dans notre cas, les recruteurs qui utilisent notre système ont des profils très réduits (leur nom, la société pour laquelle ils travaillent).

Les systèmes de recommandation collaboratifs identifient les personnes présentant des attentes similaires (ces personnes sont des *plus proches voisins*) et proposent aux utilisateurs des éléments en fonction des évaluations laissées par ces plus proches voisins. Les libraires en ligne vont par exemple suggérer des livres à des utilisateurs en se fondant sur les intérêts, achats et évaluations de *plus proches voisins*.

En e-recrutement, cette stratégie reviendrait à étudier les préférences d'un recruteur A pour quelques candidats afin de lui proposer davantage de candidats. Ceux-ci seraient puisés à même les candidats retenus par d'autres recruteurs, qui partagent des préférences avec le recruteur A. L'approche est malaisée dans notre cas, car les besoins des recruteurs sont très disparates et peu susceptibles de se chevaucher, *a fortiori*

Manuscript envoyé le 15 février 2017

1. http://entreprise.regionsjob.com/enquetes/reseaux_sociaux/resultats_enquete_2.pdf

2. <http://ressources.blogdumoderateur.com/2016/03/enquete-regionsjob-candidats-recruteurs.pdf>

3. <http://rali.iro.umontreal.ca/LBJ/Cerebra/>

dans une base de profils de plusieurs dizaines de millions de candidats. Par ailleurs, la pertinence des profils change dans le temps, au gré des expériences et des formations.

Le filtrage basé sur le contenu, quant à lui, s'appuie plutôt sur les informations données par l'utilisateur (son profil par exemple) lorsqu'il s'agit de lui proposer des éléments pertinents. C'est l'approche que nous retenons pour Cerebra, dans la mesure où nous enrichissons le profil du recruteur avec son besoin explicite, exprimé sous la forme de l'offre d'emploi pour laquelle il recrute. Dans ce cas, il nous est possible de cibler les profils de candidats pertinents en utilisant exclusivement les informations fournies par le recruteur sans avoir, comme dans la stratégie précédente, à nous appuyer sur les évaluations de candidats faites par d'autres recruteurs.

Quant aux systèmes de recommandation hybrides, ils combinent les deux approches : utilisation des évaluations et des informations données par l'utilisateur [6].

D'après Jannach *et al.*[7], un système de recommandation se compose de :

- un objectif global qui régit la sélection et le classement des éléments à recommander ;
- un ensemble d'actions disponibles centrées sur la présentation des éléments recommandés ;
- la fenêtre temporelle sur laquelle l'objectif doit être optimisé.

Dans le domaine des ressources humaines, Gu *et al.*[8] et Diaby et Viennet [9] ont développé des systèmes de recommandation pour le recrutement de professionnels. Ils proposent des applications appariant offres et profils qui utilisent une technique classique en recherche d'information : la distance cosinus entre deux documents. Dans ce cas, l'offre et le profil sont donc considérés chacun comme un document et leur distance cosinus est fonction des mots qu'ils partagent.

III. CEREBRA

Dans cette section, nous présentons les ressources nécessaires au fonctionnement de Cerebra, l'interface de Cerebra, l'algorithme d'appariement et les évaluations manuelles réalisées par notre partenaire.

A. Ressources

1) Offres:

a) *Offres traitées par Cerebra* : Le tableau I présente un exemple d'offre d'emploi pour laquelle LittleBigJob recherche des candidats, et ressemblant à celles qui sont soumises au système. On constate que la plus grande partie de cette offre est constituée d'un texte en format libre décrivant (typiquement pêle-mêle) la société qui recrute, le poste à combler, les compétences recherchées et des éléments de géolocalisation ("Lyon", ou "Région parisienne", etc.). Un champ **titre** identifie la fonction recherchée.

b) *Offres mises à notre disposition par LittleBigJob*: LittleBigJob nous a fourni 268k offres dont 139k en anglais et 123k en français, collectées sur les sites internet d'Indeed, de Cadremploi et de l'APEC. Ces offres ont essentiellement le même contenu que celle présentée en exemple à la section précédente, mais, puisqu'elles sont destinées à des fins de

Titre : LEAD DEVELOPER
Descriptif : [description de la société S qui cherche un candidat] Pour accompagner la transformation digitale de ses clients, S a créé le pôle P. Ce pôle d'expertise rassemble équipe IT & des consultants métier, qui ensemble [...] Nous déployons aujourd'hui ces solutions pour des clients de toute taille, en France et à l'international (plus de 60 pays couverts aujourd'hui). Pour renforcer notre équipe P, nous recherchons un lead développeur web PHP, qui puisse devenir le bras droit du manager de l'équipe P. Il combinera un rôle opérationnel dans la réalisation de développements sur nos solutions web, et un rôle de support à l'animation de l'équipe de développements (x développeurs à ce jour). Compétences techniques recherchées : PHP, Bootstrap, Javascript, idéalement ExtJS/SQL (MySQL) et Talend et DOT NET VB/C#. Sont un plus. Diplômé(e) d'une école d'ingénieur ou équivalent, vous justifier d'au moins 4 ans d'expérience dans le développement de solutions web. Travaillant en étroite collaboration avec le manager de l'équipe IT de P, vous intervenez sur la conception, le développement et le maintien de nos solutions web. Dans ce cadre, vous prenez en charge les sujets suivants : Réalisation des spécifications techniques de nos solutions, sur la base des cahiers des charges et spécifications fonctionnelles fournies par les consultants métiers, Elaboration de plans de tests, accompagnement de leur réalisation, Développement de nos applications, Maintenance et support technique des solutions développées, en collaboration avec le reste de l'équipe de développement. Résolution de problèmes technique de l'équipe, notamment ceux remontés par les collaborateurs moins expérimentés, Contrôle de la qualité et amélioration continue de la qualité des livrables de l'équipe, Localisation : Lyon
Langue : Français

TABLE I: Exemple d'une offre d'emploi saisie dans Cerebra. Nous avons anonymisé une partie de l'offre.

recherche, possèdent davantage de champs. On y retrouve notamment : un identifiant unique, une description du poste à pourvoir, la société procédant au recrutement, le lieu d'exercice et d'autres champs tels que le secteur d'activité ou le descriptif de la société.

2) *Profils*: LittleBigJob a collecté environ 25 millions de profils de réseaux sociaux (LinkedIn, Viadeo, Indeed et d'autres). Ces données issues des profils publics professionnels ont été anonymisées et agrégées dans un format homogène. Notre partenaire dispose des noms des personnes pour pouvoir les contacter, mais ne nous les a pas communiqués pour des questions éthiques liées à la recherche universitaire. Les profils ont été collectés pour la France et le Canada. Ils sont soit en français, soit en anglais, soit bilingues. Les éléments composant ces profils nous donnent des indications sur (voir tableau II) :

- les formations suivies : diplômes, établissements fréquentés, dates de fréquentation, etc ;
- les expériences professionnelles : emplois occupés, employeurs, tâches effectuées, secteur d'activité, etc ;
- les compétences : savoir-faire et savoir-être, ou compétences spécialisées et générales, etc.

3) *Univers* : Bien qu'il existe de nombreuses typologies des secteurs d'activités (ESCO [10], codes ROME⁴, CNP-NOC⁵, etc.), notre partenaire industriel LittleBigJob a défini

4. Répertoire Organisationnel des Métiers et Emplois (France)

5. Classification Nationale des Professions (Canada)

Formation	2015–2015 2003–2005	ESGI Infosup
Expérience	2008–maintenant	Responsable informatique : <i>société</i>
Compétences	Microsoft Office, Anglais, Microsoft Windows, Cloud Computing, Virtualisation, Cisco Technologies, ITIL, Linux, Microsoft Exchange, VoIP	

TABLE II: Extrait d'un profil de candidat.

une typologie adaptée à ses besoins spécifiques, comportant 48 secteurs d'activités baptisés *univers*. Le tableau III donne quelques exemples. Ces univers permettent de cibler les candidats qui évoluent dans le même secteur d'activité que celui de la société qui recrute, un critère de recherche très important pour notre partenaire commercial.

Id	Nom de l'univers
2	Aéronautique et aérospatiale
13	Conseil et services informatiques, édition de logiciels
18	Digital, e-commerce, big data, jeux électroniques
40	Services aux particuliers (cours, ménage...)

TABLE III: Quelques-uns des 48 univers extraits de la typologie de LittleBigJob

B. Interface

La figure 1 présente l'interface de prototypage de Cerebra. Elle est composée de champs de formulaire permettant de renseigner les détails de l'offre pour laquelle un recruteur cherche des candidats. On y spécifie le titre du poste de l'offre, son descriptif, sa langue (anglais ou français), le pays de recherche (France ou Canada) et l'univers (secteur industriel).

C. Algorithme d'appariement

Notre algorithme utilise les ressources décrites à la section III-A pour extraire l'information utile d'une offre d'emploi : le titre d'emploi, les compétences et l'expérience. Il nous est ensuite possible de réaliser l'appariement entre offre d'emploi et profils.

1) Extraction des informations utiles:

a) *Extraction du titre d'emploi:* L'interface de Cerebra dispose d'un champ titre d'emploi nous permettant de récupérer facilement ce champ. Cependant ce champ doit être nettoyé car il contient parfois des informations non pertinentes pour déterminer le métier. Par exemple, nous supprimons les informations donnant le type de contrat (CDD, CDI, ...), le temps de travail (mi-temps, temps partiel), la localisation (à Paris, en Vendée, ...), etc. Le tableau IV présente le titre extrait par Cerebra dans l'offre d'emploi du tableau I.

b) *Extraction des compétences:* L'extraction des compétences est réalisée à partir du descriptif de l'offre, un texte libre rédigé par le recruteur. Nous construisons d'abord une liste des compétences les plus fréquentes trouvées dans les profils de candidats décrits à la section III-A. Nous nous en tenons aux plus fréquentes seulement, car les candidats ont tendance à mal renseigner ce champ : on y retrouve beaucoup de scories, du type "stratégies" ou "mesures urgence". Choisir

les plus fréquentes en améliore la qualité. Avec cette liste en main, nous en recherchons les compétences dans le descriptif de l'offre. Le tableau IV présente les compétences extraites par Cerebra dans l'offre d'emploi du tableau I. On le voit, l'extraction privilégie la précision au détriment du rappel : nous préférons extraire quelques compétences fréquemment mentionnées par les candidats dont nous sommes sûrs qu'elles sont bien des compétences, plutôt que de nous tromper en tentant d'extraire davantage de compétences rarement mentionnées qui sont susceptibles de ne pas être des compétences du tout.

c) *Extraction des années d'expérience:* Il est également important d'extraire le nombre requis d'années d'expériences à partir du descriptif de l'offre. Pour ce faire, on a construit une expression régulière permettant de capturer les patrons de mots faisant référence à une durée d'expérience : *experience*, *expérience*, *années*, *years*, *related*, etc. Le tableau IV présente les compétences extraites dans l'offre de la figure I.

Titre extrait	lead developer
Compétences extraites	gestion de projets, résolution de problèmes
Expérience extraite	4 ans

TABLE IV: Informations extraites de l'offre présentée à la figure I.

2) *Recherche des profils:* Nous effectuons l'indexation et la recherche plein texte de profils à l'aide du moteur Lucene⁶. Afin de réaliser l'appariement entre une offre et des candidats, nous construisons et soumettons une requête booléenne à partir des informations extraites de l'offre. Cette requête s'appuie sur les informations extraites plus haut.

On cherche le titre d'emploi dans les fonctions occupées par les candidats à l'aide d'un champ Lucene que nous avons appelé **expfunc**. On cherche également les compétences extraites dans l'offre parmi les compétences saisies dans un profil à l'aide du champ **skillname**. On filtre avec l'univers sélectionné par le recruteur dans la liste déroulante de Cerebra (champ **univid**).

Voici un exemple de requête générée à partir de l'offre d'emploi de la figure I :

```
expfunc="lead developer" AND
skillname:(gestion de projets, résolution
de problèmes) AND univid=13.
```

Cette requête est soumise au moteur Lucene sous-jacent, en précisant de plus le pays (France ou Canada) et la langue de la base de profils à utiliser.

3) *Classement des résultats:* Les résultats de la requête précédente sont ensuite filtrés pour raffiner les résultats. Pour cela, nous avons créé une fonction attribuant un score à chaque candidat. C'est ce score qui détermine la place d'un candidat dans les résultats produits. Ce score de qualité est l'agrégation de différents sous-scores, décrits ci-dessous.

— un sous-score pour le titre d'emploi : on mesure la similarité entre le titre d'emploi de l'offre et les titres

6. <https://lucene.apache.org/>

FIGURE 1: Interface du prototype de Cerebra, permettant la saisie d'une offre et d'informations sur celle-ci (langue, pays, secteur d'activité), afin de repérer des candidats potentiels.

d'emplois occupés par le candidat au cours de sa vie. Plus une expérience est ancienne, moins cette similarité pèse lourd dans le sous-score, afin de privilégier les expériences récentes.

- un sous-score pour les compétences : on compte les nombres de compétences extraites de l'offre trouvées dans le profil.
- un sous-score pour l'expérience : on s'intéresse à l'écart entre le nombre d'années d'expérience requise par l'offre et le nombre d'années d'expérience du candidat.
- le sous-score de pertinence donné par Lucene.

D. Évaluation

Le système Cerebra a fait l'objet d'une évaluation manuelle pour mesurer la qualité des profils proposés au recruteur. Nous avons conçu pour ceci une interface web d'évaluation qui permet au recruteur-annotateur de saisir une offre, son titre et de spécifier son univers. Il choisit ensuite le nombre de candidats qu'il désire annoter (typiquement entre 1 et 5 inclusivement), puis effectue la recherche. Pour chaque candidat récupéré par les algorithmes de Cerebra décrits plus haut, le recruteur assigne une note allant de 1 (très mauvais) à 5 (excellent). Lorsque la note n'est pas parfaite, l'utilisateur est contraint de fournir un commentaire en texte libre nous permettant de mieux comprendre l'impression de l'annotateur.

Pendant la première ronde de la campagne d'évaluation, 86 offres françaises ont été soumises et ont permis la récupération et l'évaluation de 320 candidats. La figure 3a montre les notes obtenues par ces candidats. Même si ces statistiques montrent que seulement 11 % de candidats recueillent une note de

4 ou 5, le dépouillement des commentaires des recruteurs démontre que 34 % des candidats offrent en fait un bon intérêt pour le recruteur, et qu'une seule caractéristique malheureuse suffit pour nuire significativement au candidat. La figure 2 cite quelques-uns des commentaires positifs recueillis par de bons profils.

*Bien localisé. Très bon profil dans le domaine de la logistique et approvisionnement
Bonne localisation Très bonne expérience de fiscaliste
Anglais courant comme demandé*

FIGURE 2: Exemple d'évaluations positives.

Cela dit, les problèmes soulevés par LittleBigJob concernent divers aspects des profils retournés.

Pour 18 % des profils évalués, l'expérience professionnelle générale du candidat est jugée hors du domaine ciblé. On aura alors des commentaires comme *Pas d'expérience en business development Pas d'expérience en informatique* ou encore *Compétences : ok MAIS Poste basé à Lyon et pas le bon secteur d'activité.*

Pour 22 % des profils, l'expérience qui correspond bien au titre du poste à combler remonte à trop loin. Les recruteurs commentent par exemple en disant *l'expérience intéressante remonte à 2008* ou *Ce n'est plus son poste actuel. L'expérience de Responsable communication est trop lointaine.*

Le plus important des problèmes, avec 60 % des profils évalués est la localisation géographique : le candidat habite une ville trop éloignée de la ville où se situe le poste. On aura alors des annotations du type *Profil correspond à la recherche.*

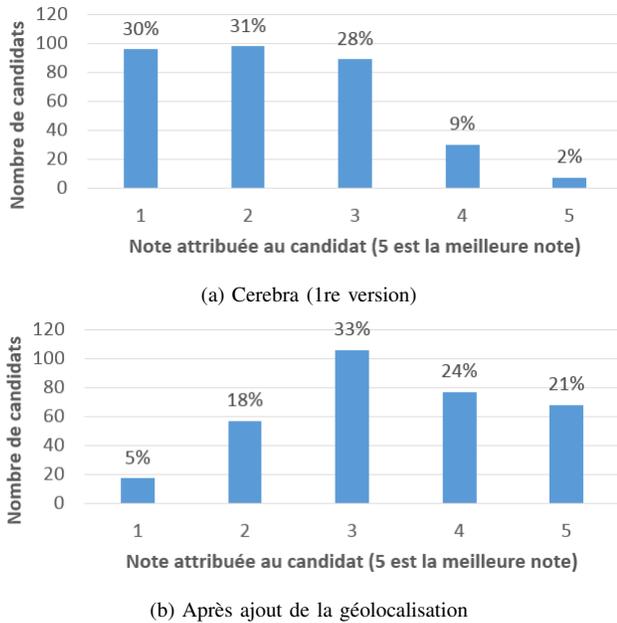


FIGURE 3: Évaluations par LittleBigJob des candidats proposés par Cerebra.

Profil situé à Bordeaux, alors que le poste est à Lyon.

Cette première ronde d'évaluation a conduit à des améliorations, décrites et évaluées à la section suivante.

IV. AMÉLIORATIONS

Dans cette section, nous décrivons des améliorations aux problèmes de géolocalisation et de domaine d'activité présentés plus haut.

A. Prise en compte de la localisation

Pour tenir compte de la localisation géographique d'une offre afin de l'apparier à celle d'un candidat, nous avons utilisé une approche très simple. Nous avons d'abord compilé la liste des communes de France et des villes du Canada à l'aide de ressources en ligne. Pour la France, par exemple, nous utilisons une liste de 37k communes disponibles sur le site SQL.sh⁷. Lorsqu'une offre est soumise, on cherche ces villes dans son descriptif, en partant de la fin et en s'arrêtant dès qu'un repérage est possible. On procède à reculons, car il arrive souvent que plusieurs noms de villes apparaissent dans une offre, alors que c'est habituellement la dernière qui est la bonne. Comme le tableau I l'illustre, c'est la toute dernière phrase qui indique la ville (Lyon). Si une ville est retrouvée dans le texte de l'offre, elle est utilisée comme critère de recherche supplémentaire afin de filtrer les résultats.

Une fois notre heuristique de localisation intégrée dans Cerebra, LittleBigJob a réalisé une seconde ronde d'évaluations : 65 nouvelles offres ont été soumises et ont conduit à 325 évaluations de candidats. Les résultats sont présentés à la figure 3b. L'amélioration est significative : le pourcentage de profils recevant les notes de 4 ou 5 passe de 11 % à 45 %

7. <http://sql.sh/736-base-donnees-villes-francaises>

lorsque nous dotons Cerebra de géolocalisation, confirmant à la fois l'utilité de notre modification et l'importance du critère géographique.

Dans 76 % des cas, la localisation des profils correspondait à ce qui était attendu. Les localisations incorrectes surviennent principalement lorsque notre système ne réussit pas à identifier la ville dans le descriptif de l'offre : information absente de l'offre, orthographe différente (p.ex. St Étienne au lieu de Saint-Étienne) ou alors mention d'une zone (p.ex. région lyonnaise). Sur 65 offres, notre système extrait correctement la ville de 52 d'entre elles, pour un taux de succès de 80 %.

B. Détection automatique de l'univers de l'offre

Le problème d'adéquation de l'univers de l'offre avec celui du candidat s'explique principalement par le fait qu'aucun univers n'a été sélectionné par le recruteur lors de sa recherche, et ce, pour 15 % des offres. Ceci est dû peut-être à des contraintes ergonomiques, mais aussi sans doute à la difficulté de trouver le bon univers parmi 48. Afin de simplifier le processus délicat de sélection des univers, nous proposons de ne présenter que 6 univers au recruteur (au lieu de 48), en détectant nous-mêmes l'univers de l'offre par apprentissage machine (voir section IV-B). Cette section explique l'approche privilégiée, sans être appuyée par une autre évaluation humaine des résultats.

Pour détecter automatiquement l'univers d'une offre, nous considérons que les offres issues d'un même univers partagent un vocabulaire commun. L'apprentissage machine offre des solutions éprouvées pour ces cas de figure. Par ailleurs, l'analyse des offres soumises au système par les recruteurs humains a permis de dégager deux stratégies qu'utilisent les recruteurs pour déterminer l'univers de l'offre à pourvoir : soit le recruteur détermine l'univers à l'aide du titre d'emploi, soit il le fait à l'aide de l'univers de la société qui recrute. Par exemple, un analyste financier recruté par Bombardier sera classé dans l'univers 2 *Aéronautique et aérospatiale* si on considère l'univers de la société Bombardier ou dans l'univers 10 *Banque, finance, capital risque, fonds privés* si nous nous référons à notre référentiel liant titre d'emploi et univers).

1) *Processus (décrit aussi dans [11])*: L'affectation automatique d'un univers à une offre est un problème de catégorisation de documents. Il s'agit d'affecter automatiquement une classe (un univers ici) à un document (une offre d'emploi). Nos classes ayant été définies par notre partenaire, nous utiliserons des techniques liées à l'apprentissage machine supervisé. L'apprentissage supervisé cherche à apprendre des règles pour classer de nouveaux textes à partir des textes déjà classés.

Les étapes principales pour réaliser un apprentissage supervisé sont :

- 1) **Définition des classes** : Nos classes correspondent aux univers décrits à la section III.
- 2) **Création d'un corpus d'apprentissage** : Nous avons créé deux corpus d'apprentissage : un premier

(*corpus-soc*) à partir de notre référentiel liant sociétés et univers, et un second (*corpus-titre*) à partir de notre référentiel liant titres d'emploi et univers.

- 3) **Apprentissage de 2 modèles de classification à l'aide des 2 corpus d'apprentissage** : Nous avons retenu 4 méthodes d'apprentissage parmi les plus couramment utilisées : C4.5 [12], K-plus-proches voisins (KNN), la forêt d'arbres de décision (RF)[13] et SMO (sequential minimal optimization) [14]. L'apprentissage est réalisé pour prédire l'univers de l'offre, à partir de son descriptif. Nous avons donc deux modèles : un modèle entraîné sur *corpus-soc* et l'autre, sur *corpus-titre*.
- 4) **Évaluation du modèle** : Nous avons évalué nos modèles par validation croisée et sur des échantillons de test (voir [11]). Nous avons obtenus de bons résultats en termes de précision et rappels (de l'ordre de 80 %).

2) *Proposition d'intégration à Cerebra*: Nous présentons à la figure 4 comment la détection de l'univers sera intégrée dans notre système de recommandation. La liste déroulante de l'interface de la figure 1 présentera un choix de six univers (au lieu des 48 présentés actuellement). Le premier univers de la liste sera le(s) univers détecté(s) ou prédit(s), suivi(s) des univers apparentés. La priorité est donnée à l'univers de la compagnie car selon notre partenaire LittleBigJob, les recruteurs cherchent d'abord des candidats issus de l'univers de leur société.

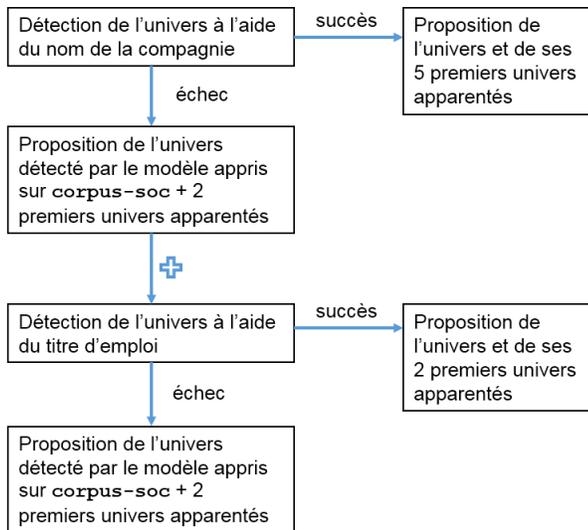


FIGURE 4: Proposition de l'univers dans Cerebra.

V. CONCLUSION

Les évaluations manuelles de la version de Cerebra présentée ici semblent prometteuse. Près de la moitié des candidats retournés par le système satisfont les recruteurs qui les ont évalués. Ce chiffre intéressant est en partie dû aux importantes ressources mises à notre disposition, pour la plupart extraites du web. En d'autres mots, le « big data » porte fruit en recrutement, pourvu qu'il soit adéquatement exploité.

Au-delà de ce chiffre, le développement itératif du prototype aura également servi à cerner une partie des besoins des recruteurs de LittleBigJob. Créer ce moteur de recherche aura permis de mettre en lumière l'importance de critères évidents (compétences) et moins évidents (secteur d'activité), ainsi que leur interaction parfois subtile, rendant nécessaire l'élaboration semi-manuelle et itérative d'un score de qualité de candidat.

Cela dit, Cerebra demeure un chantier qui profitera de plusieurs améliorations, guidées par les commentaires des évaluations. Il nous faudra notamment étudier les dates de l'expérience intéressante pour l'offre, afin de ne pas proposer des profils ayant une expérience trop ancienne. La détection des compétences d'une offre pourra quant à elle profiter des travaux de Kessler *et al.* [15], qui ont développé un système construisant automatiquement une ontologie des compétences pour un certain nombre d'emplois, répartis par univers. La géolocalisation pourrait également s'étendre aux villes voisines de celles recherchées.

Au final, Cerebra s'avère un système de recommandation utile. Il permet en effet de proposer au recruteur un ensemble de candidats qui satisfont l'offre soumise ou, à tout le moins, qui offrent un compromis raisonnable entre différents critères de recherche extraits de l'offre.

RÉFÉRENCES

- [1] V. Charrière, C. Dejoux, and F. Dupuich, "L'impact des réseaux sociaux et des compétences émotionnelles dans la recherche d'emploi : étude exploratoire," *Management & Avenir*, no. 2, pp. 137–163, 2014.
- [2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens : an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 1994, pp. 175–186.
- [3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.
- [4] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*. Springer, 2007, pp. 325–341.
- [5] R. Burke, "Hybrid web recommender systems," in *The adaptive web*. Springer, 2007, pp. 377–408.
- [6] —, "Hybrid recommender systems : Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [7] D. Jannach, P. Resnick, A. Tuzhilin, and M. Zanker, "Recommender systems— : beyond matrix completion," *Communications of the ACM*, vol. 59, no. 11, pp. 94–102, 2016.
- [8] Y. Gu, B. Zhao, D. Hardtke, and Y. Sun, "Learning global term weights for content-based recommender systems," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 391–400.
- [9] M. Diaby and E. Viennet, "Développement d'une application de recommandation d'offres d'emploi aux utilisateurs de facebook et linkedin," in *Atelier Fouille de Données Complexes de la 14e Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC'14)*, Rennes, 2014.
- [10] M. Le Vrang, A. Papantoniou, E. Pauwels, P. Fannes, D. Vandestein, and J. De Smedt, "Esco : Boosting job matching in europe with semantic interoperability," *Computer*, vol. 10, no. 47, pp. 57–64, 2014.
- [11] A. Casagrande, F. Gotti, and G. Lapalme, "Classification d'offres d'emploi," *soumis à CORIA*, 2017.
- [12] J. R. Quinlan, *C4.5 : programs for machine learning*. Elsevier, 2014.
- [13] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [14] J. C. Platt, "12 fast training of support vector machines using sequential minimal optimization," *Advances in kernel methods*, pp. 185–208, 1999.
- [15] R. Kessler, E. Tondo, and G. Lapalme, "Génération automatique d'une ontologie dans le domaine des ressources humaines," in *CORIA-CIFED*, 2016, pp. 637–652.