Cahier des besoins
Projet Machine Learning 2

CLASSES 4 Data Science

2019/2020

Sommaire

1. Introduction

Ce projet vise à se doter d'un outil de Machine / Deep Learning qui permet de télécharger et agréger des offres d'emploi provenant de sites de recrutement, réseaux sociaux et d'autres sources (notamment d'e-mails, PDF, Word, etc.). Ensuite, l'outil doit pouvoir les formater pour correspondre à son moteur de recherche sémantique.

Pour répondre à ce besoin, le besoin consiste à transformer les données « input » initialement sous format PDF, WORD ou HTML en un vecteur de compétences unidimensionnel. La chaine de valeur peut être découpée en 2 phases :

- **Phase 1** : Récupérer les offres d'emploi des différentes sources, retraiter le texte et repérer les éléments de ce texte qui correspondent aux différents labels (par ex : missions, nom de l'entreprise, années d'expérience, etc.) ;

- **Phase 2** : Retrouver les compétences (qualitatives et quantitatives figurant dans le dictionnaire sémantique implicitement évoquées dans les offres d'emploi.

Dans le cadre de ce projet, vous devez définir le ***panel de compétences (8 à10 compétences)*** et à ***un seul type de métier*** (Informatique, Gestion, Finance..) en milieu bancaire. Par conséquent, l'objectif du projet est non seulement d'allouer le temps de travail à la composante technique uniquement, mais aussi à identifier la cartographie des compétences et la définition des sémantiques.

Dans ce projet, les offres collectées se ressemblent et donc il est normal de retrouver les mêmes compétences sur toutes ces dernières. Cependant, vous pourrez apprécier les subtilités entre les différents types du métier choisi que nous pourrons capter avec un dictionnaire sémantique plus précis.

Par exemple (pour le métier d'un juriste) : distinguer le droit fiscal du droit bancaire ou bien insister sur une norme comptable en particulier et pas sur l'autre (IFRS pour l'Europe et GAAP pour les USA).

2. Inputs

2.1. Les offres d'emploi

Vous devrez collecter ***au minimum 50 offres d'emploi*** dans différents supports : PDF ; Word et HTML. Ces offres sont issues des sites de recrutement (tunisien, français ou autre..) et n'ont donc ni le même format ni la même tournure. Ce choix devrait rendre l'exercice plus complet.

Vous serez donc amenés à structurer le panel d'offres pour constituer les ensembles d'apprentissage selon la logique qui leur paraît la plus plausible.

2.2. Dictionnaire sémantique

Le dictionnaire sémantique regroupe les éléments qui, en plus de l'entraînement manuel de la machine à partir des offres d'emploi, doit renvoyer aux compétences auxquelles ils sont affectés.

Exemple (pour le métier d'un juriste) : Il est défini comme suit :

- Droit français : droit des contrats ; droit commercial ; droit des garanties ; droit fiscal des entreprises ; droit patrimonial ; régime matrimonial ; succession ; Droit bancaire.

- Droit bancaire : FBF ; ISDA ; Normes comptables ; IFRS ; GAAP ; Droit financier ; Code Monétaire et Financier ; CMF.
- Produits bancaires : produits dérivés ; instruments financiers ; produit d'investissement ; produits de placement ; produit de spéculation ; instrument de marchés ; produits structurés ; produits d'épargne ; produits d'assurance ; émissions obligataires ; titres de créance ; Opérations de marché ; Marchés financiers.
- Fiscalité : l'impôt sur le revenu ; IR ; l'impôt sur les sociétés ; IS ; impôts sur les bénéfices ; l'impôt de solidarité sur la fortune ; ISF ; la taxe sur la valeur ajoutée ; TVA ; la taxe intérieure sur les produits pétroliers ; TIPP ; code général des impôts ; CGI ; taxes ; CGS ; CRDS.
- Risque bancaire : risque de marché ; risque opérationnel ; risque de contrepartie ; risque de crédit ; Normes de réglementation ; FRTB ; BALE2 ; BALE 2.5 ; EMIR ; VAR ; LCR ; CVAR ; Expected Shortfall ; ES.
- Outils informatiques : Microsoft Office ; Excel ; Outlook ; Access ; VBA ; Word ; bureautique.
- Organisation : Agenda ; Rigueur ; planifier ; Travail en équipe ; Outlook ;
- Communication : Convaincre ; Relationnel ; collaboration ; réunions ; compte rendu ; Anglais ; Ecrit ; Oral.

NB : Dans les offres, il se peut également que la syntaxe change (par ex : BCE, B.C.E.) ou que l'auteur de l'offre ait commis une erreur d'orthographe. Il est donc recommandé de réfléchir à une solution, dynamique de préférence, à ce genre de situations.

3. Phase 1 : repérage des rubriques et des informations clés

3.1. Lecture de supports texte

La première étape consiste à lire les fichiers d'offres d'emploi et en extraire le texte brut. Il est donc ici question d'enlever les éléments non textuels (mise en forme, balises, images, header, footer et éventuellement pub dans le sidenav en HTML, etc.). Cette étape, permettra d'obtenir une suite de phrases sensées en lien direct avec l'offre d'emploi.

3.2. Traitement de texte

Cette étape consiste en la transformation du texte en vecteurs ou matrices selon le choix de la méthode de travail (Word2Vec, TFIDF, Bag of words, etc.). Nous recommandons une approche qui permet de découper le texte en ensembles compacts isolés portant soit sur un sens soit sur une logique lexicale. La représentation du texte doit permettre la classification. Par la suite, le training de l'algorithme reposera sur le « training set » composé d'offres d'emploi ainsi que sur des API ou mini-dictionnaires comme décrit dans la suite.

3.3. Classification et labels

Lors de cette première phase, l'algorithme est censé retrouver des bouts de textes appartenant à chacun de ces labels et en effectuant son apprentissage soit sur les offres d'emploi en input. Les différents labels du projet sont les suivants :

- **Formation :** Indiquer le niveau d'éducation minimum exigé allant de niveau BAC à Doctorat ;
- **Intitulé du poste :** Reconnaitre les variantes du métier de juriste ;
- **Type de contrat** : Indiquer la nature du contrat pouvant être un stage, Alternance, Intérim, CDD ou CDI ;
- **Nom de l'entreprise :** Extraire de l'offre le nom de l'employeur ou client dans le cas d'une prestation de service;
- **Années d'expérience :** Indiquer, si mentionné, le niveau d'expérience exigé pour le poste en question
- **Missions / Compétences :** Ressortir toutes les tâches que devra effectuer le nouvel arrivant ainsi que les compétences explicitées dans l'offre ;
- **Ville :** Reconnaitre la ville où se trouve l'emploi. Piste/Indication : Utiliser une API Géo qui couvre les codes postaux, villes et pays.

3.4. Output intermédiaire

A l'issue de cette phase, chaque offre d'emploi en entrée texte doit être découpée en morceaux (phrases, suites de mots ou mots) selon les 7 labels suscités. Le choix du format de l'output n'import que dans la mesure où le label « missions/compétences » doit être facilement exploitable par la phase 2 puisqu'il en est l'input.

Au terme de la phase 1, vous devez récupérer les phrases présentes dans les classes Missions / Tâches afin d'attribuer à chacune la (les) compétence(s) implicites. Un format Excel ou Csv est préférable:
- En ROWS: les phrases
- En COLUMNS: les compétences implicites

Exemple: (pour le métier juriste)

Phrase	Comptabilité	Consolidation	Trading	C++	JAVA
Le comptable devra mettre à jour le bilan comptable consolidé de la banque	X	X			

4. Phase 2 : Implication des compétences

4.1. Entraînement de l'algorithme

A la suite de la phase 1, l'ensemble des éléments du label « missions / compétences » devra être découpé en phrases et regroupé dans un document consolidé pour procéder à la phase d'apprentissage sous format Excel ou CSV.
Ensuite, vous devrez se charger de labéliser chaque élément du label Missions/ compétences selon les 8 compétences du dictionnaire sémantique. D'ailleurs, l'usage du champ sémantique et la manière de le faire devront être définis par les étudiants.

Par ailleurs, si une compétence est retrouvée une fois, l'algorithme doit pouvoir la retrouver une seconde fois si elle est évoquée plusieurs fois. Cela permettra de mesurer le caractère exact et complet de l'algorithme qui sera mesuré par les scores F1 (1 score par compétence). Cependant, dans l'output final il est inutile de renseigner le nombre d'occurrence.

4.2. Output

L'algorithme doit être en mesure de fournir pour chaque offre d'emploi (missions/compétences pour être plus précis) un vecteur booléen qui indique si oui ou non la compétence i est évoquée dans cette offre d'emploi.

5. Mesure de la performance

5.1. Score F1

Pour mesurer l'efficacité du problème de classification, il est possible d'utiliser une variante du score F1. En effet, nous pourrions, pour chaque compétence, voir si l'algorithme retrouve bien la compétence là où elle est évoquée.

Pour pouvoir calculer ce facteur il est important de découper convenablement la partie « missions / compétences ».

5.2. Spécificités de l'outil

A ce stade, les principaux points auxquels nous aimerions attirer votre attention sont les suivants :

- Gestion des erreurs de frappe (typos) ;
- Le nombre d'offres en input et le nombre de compétences pouvant varier, le code doit être en mesure de compiler en changeant ces paramètres ;