Projet SmartRecruiting

Rapport technique

Estelle LANDI
Estelle REYMOND
Loïc SCHANEN
Rémi VARENNE

2019-2020

Cahier des charges

Après avoir échangé avec Anthony Geourjon et Gerard Pollier, nos missions étaient tout d'abord de nous occuper du persona responsable de scolarité en premier lieu, pour après s'intéresser au côté Green IT et au persona chargé de recrutement.

Commençons par décrire les demandes concernant le persona gestionnaire de scolarité. La majeure partie des demandes comprenait des statistiques ayant pour but de montrer l'évolution des intérêts des autres acteurs du sujet (étudiants voulant un stage, étudiants voulant une formation et chargé de recrutement).

Suite à des échanges entre Anthony et Nadine Chatti, la responsable relations entreprises de Polytech Grenoble, certaines informations ont été retenues car elles semblaient intéressantes et utiles. En voici la liste :

- affichage du nombre d'offres où la filière / une des filières de la composante arrive première
- Affichage des domaines (ou secteurs) associés aux offres de stages et donc aux entreprises
- Affichage des compétences requises que ce soit pour les formations ou pour les offres
- Affichage de tendances des domaines et compétences les plus recherchés
- Affichage des entreprises les plus demandeuses et dont le nombre d'offres a le plus évolué
- Affichage d'un indice de confiance si l'offre est adéquate à la filière / un des filières de la composante
- …

Le second sprint concernait le côté green IT qui avait pour but d'ajouter en premier lieu des recommandations sur les techniques permettant d'obtenir un code un peu plus éco-responsable. Il faut savoir qu'actuellement, il y a très peu de travaux dont l'objectif est d'avoir un code éco-responsable. De nombreuses recherches ont été faites, nous avons trouvé un nombre important de recommandations mais très peu sont appliquées actuellement. Il est bien dommage que ces recommandations ne soient pas mises en place car la consommation d'énergie due aux services informatiques est encore actuellement sous-évaluée.

C'est un aspect d'autant plus intéressant pour les projets utilisant des méthodes

d’intelligence artificielle. En effet, l’utilisation d’intelligence artificielle est souvent omise dans les consommations énergétiques. Les recherches sont beaucoup moins importantes pour cette partie là de l’informatique. C’est pourtant un domaine extrêmement consommateur même s’il permet en même temps de proposer des solutions ayant pour but de réduire les consommations d’énergie.

Le troisième sprint concernait la gestion des entreprises, notamment l’inscription de chargé de recrutement avec l’inscription de l’entreprise associée en base de données. Il fallait aussi permettre la gestion des offres par l’entreprise, que ce soit l’ajout, la modification ou la suppression d’offres. Il s’agissait de poursuivre le travail du groupe d’étudiants d’ENSIMAG/GEM en travaillant notamment sur la partie Front du code.

Technologies utilisées

Keras

Keras est l'une des principales API de réseaux de neurones de haut niveau. Elle est écrite en Python et prend en charge plusieurs moteurs de calcul de réseaux neuronaux en arrière-plan.

Keras a été créé pour être modulaire, facile à prendre en main. Elle est conçue "pour les êtres humains, pas pour les machines" et "suit les meilleures pratiques pour réduire la charge cognitive".

Les couches neuronales, les fonctions de coût, les optimisateurs, les fonctions d'activation et les fonctions régularisations sont tous des modules autonomes qu'il est possible de combiner pour créer de nouveaux modèles. Les nouveaux modules sont simples à ajouter, comme de nouvelles classes et fonctions. Les modèles sont définis en Python, et non dans des fichiers de configuration de modèle séparés.

Les principales raisons d'utiliser Keras découlent de ses principes directeurs, principalement celui de la convivialité. Au-delà de la facilité d'apprentissage et de construction de modèles, Keras offre les avantages d'une large adoption, de la prise en charge d'un large éventail d'options de déploiement en production, de l'intégration d'au moins 5 moteurs back-end (Tensorflow, CNTK, Theano, MXNet et PlaidML), et d'une forte prise en charge de plusieurs GPU et la formation distribuée.

MySQL

MySQL est un système de gestion de bases de données relationnelles. Il fait partie des logiciels de gestion de bases de données les plus utilisés au monde. C'est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place vers celui de mises à jour fréquentes et fortement sécurisées.

Flask

Flask est un framework web qui fournit des outils, des bibliothèques et des technologies qui vous permettent de construire l'application web. L'application web peut être une page web, un blog, un wiki ou être aussi grande qu'une application de calendrier sur le web.

Il fait partie des catégories des micro-frameworks qui sont normalement des frameworks ayant peu ou pas de dépendances avec des bibliothèques externes. Cela présente aussi bien des avantages que des inconvénients. Les avantages sont que le framework est léger, qu'il y a peu de dépendances à mettre à jour et de bugs de sécurité à surveiller. Les inconvénients sont qu'à un moment donné, il faudra faire plus de travail par soi même ou augmenter le nombre de dépendances en ajoutant des plugins. Dans le cas de Flask, les dépendances actuelles sont :

- Werkzeug, une librairie d'utilitaire WSGI
- jinja2 qui est un moteur d'impact

Angular et Material Design Bootstrap

Angular est un framework développé par Google basé sur TypeScript. Il est grandement orienté autour des composants et des services. Ce framework nous a permis de développer la partie front-end du projet.
Pour bien commencer le projet, nous nous sommes basés sur un template de dashboard de MDBootstrap (Material Design Bootsrap). MDBootstrap est très populaire pour développer des applications web réactives.

Réalisation technique

Vectorisation

Le modèle gensim Word2Vec, qui est un simple réseau neuronal à deux couches, a été pris pour vectoriser le corpus de texte. Word2Vec est une technique d'intégration de mots qui transforme un corpus de texte en des vecteurs, et elle a récemment gagné en popularité dans le domaine du traitement de langage naturel (PNL) grâce à ses performances importantes.

Si l'on essaye d'expliquer le plus simplement possible cette technique, on peut dire que l'on projette les mots un à un dans un espace de dimension importante (cela peut varier de 50 à 200 dimensions en général, sachant que sa taille sera plus ou moins proportionnelle à la taille du corpus). Le modèle va placer les mots similaires proches les uns des autres dans cet espace. Si l'on considère que le résultat, ou la sortie, de Word2Vec est un des vecteurs, les mots compris dans un vecteur seront les mots se ressemblant. Ainsi, plus la dimension sera importante, plus il y aura de groupe de mots.

Le modèle Word2Vec a besoin de plusieurs hyperparamètres pour entraîner le modèle :

- taille : dimension de l'espace d'intégration
- fenêtre : la distance maximale entre un mot cible et des mots autour du mot cible
- min_count : la fréquence minimum de mots à considérer lors de la formation du modèle, les mots dont l'occurrence est inférieure à ce nombre sur le corpus
- sg : l'algorithme de formation des vecteurs. Il en existe deux, CBOW qui utilise le contexte pour prédire un mot cible, et skip-gram qui utilise un mot pour prédire un contexte cible. Généralement, la méthode skip-gram peut avoir de meilleures performances que la méthode CBOW, car elle peut capturer deux sémantiques pour un seul mot.

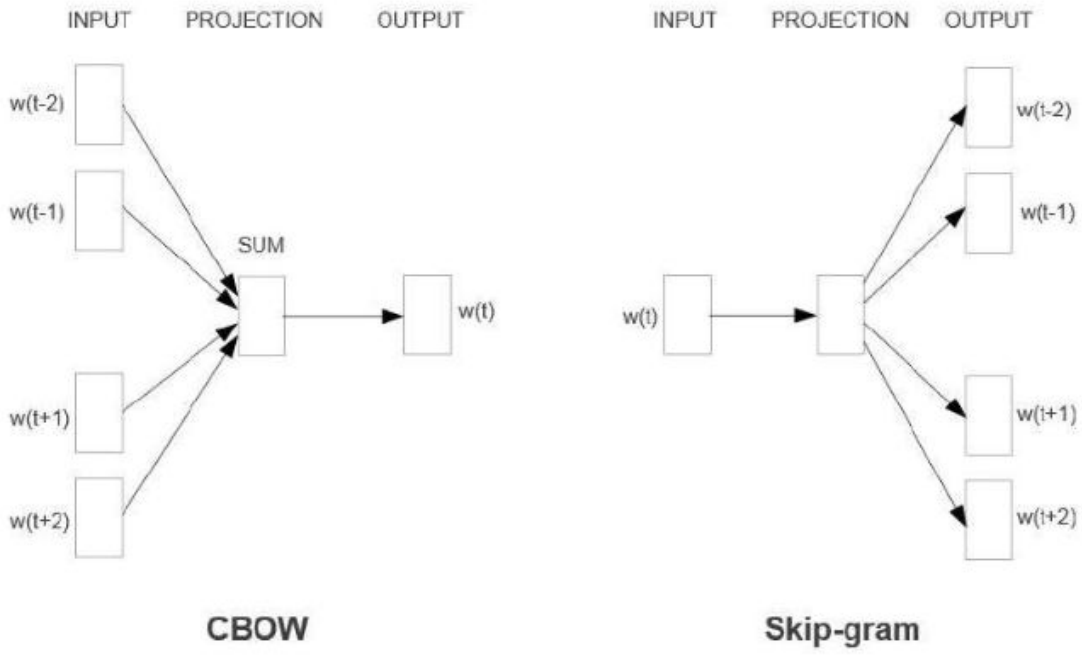

Figure 1 : Représentation des algorithmes de word2Vec

Réseau de neurones

Le modèle de vectorisation qui a été créé juste avant est ajouté comme couche au modèle de réseau neuronal créé avec Keras. Toutes les dimensions du modèle Word2Vec sont ajoutées comme neurones au réseau neuronal convolutionnel. Les poids associés à ces neurones sont les valeurs des tokens associés de cette dimension. Les couches cachées du modèle de Keras sont formées d'une couche convolutionnelle en une dimension comme dit auparavant et d'une couche de recouvrement qui permet de former correctement les sorties du modèle afin de pouvoir les traiter ensuite et de les passer à la partie front-end. Même si la taille des données d'entrée peuvent sembler importantes, elles sont en réalité plutôt faible pour des modèles de deep-learning. C'est pourquoi on a préféré la structure simple qui avait été prise auparavant pour la couche cachée. En effet, la couche cachée est en lien direct avec la couche de sortie et le nombre de neurones, ou d'éléments, sera égal au nombre de classes, ou de catégories, qui sont possible d'être retourné par le modèle.

Dans ce réseau de neurones, la fonction "softmax" a été choisie pour généraliser la régression logistique que fait le modèle dans le cas où nous voulons traiter plusieurs classes. Cette fonction est particulièrement utile pour les réseaux de neurones où l'on souhaite appliquer une classification non binaire, ce qui est notre cas pour ce modèle.

Un autre élément qui est important dans ce modèle est le “shuffle” qui permet de mélanger les données. Grâce à cela, il est possible de lancer plusieurs fois le modèle, les données de test ne seront jamais les mêmes et nous pourrons correctement évalués le modèle.

Tf-idf

Le “term frequency-inverse document frequency” (tf-idf) est l’une des techniques les plus utilisées dans le domaine de l’exploration d’information textuelle. Elle permet d’identifier dans quelles proportions certains mots du corpus peuvent être évalués par rapport au reste du corpus. Cette technique de référencement est très utile pour déterminer les mots-clés qui pourront être idéalement utilisés pour l’entrainement du modèle.

Revenons sur la technique qui évolue les informations portées par chaque token dans un document. Cette technique utilise deux termes, la “term-frequency” (tf) et l’inverse document frequency”.

La tf détermine la fréquence relative d’un mot ou d’une combinaison de mots dans une offre de stage. Cette fréquence du terme sera comparée à l’apparition de tous les autres termes du corpus. Le fait que cette fonction soit une fonction logarithmique permet de prendre en compte la proportion de chaque mot utilisé dans une offre de stage au lieu de ne prendre que la distribution d’un seul mot, ce qui serait le cas pour le calcul de la densité d’un mot.

L’idf complète l’analyse de l’évaluation du mot. Il permet de corriger les erreurs liées à la tf. L’idf ajoute dans le calcul de la tf-idf la fréquence des offres de stages pour un mot précis. C’est-à-dire, l’idf compare le chiffre correspondant à tous les documents connus avec le nombre d’offres de stage contenant le mot en question. L’idf permet d’identifier la pertinence d’une offre de stage en considérant un mot-clé précis. Afin d’obtenir des résultats utiles, la formule a besoin d’être appliquée à tout mot-clé significatif dans une offre de stage et plus le corpus aura un nombre d’offres de stage important et plus les résultats seront précis.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Formule 1 : Équation de calcul du tf-idf d'un mot

Allocation de Dirichlet Latente

La Latent Dirichlet Allocation (LDA) est un modèle statistique qui est largement appliqué dans le domaine du traitement du langage naturel, ou natural language processing (NLP).

Imaginons un ensemble fixe de thèmes. Chaque thème est associé à un ensemble de mots. L'objectif de la LDA est d'associer tous les documents aux thèmes de manière à ce que les mots de chaque document soient principalement représentés par ces thèmes imaginaires.

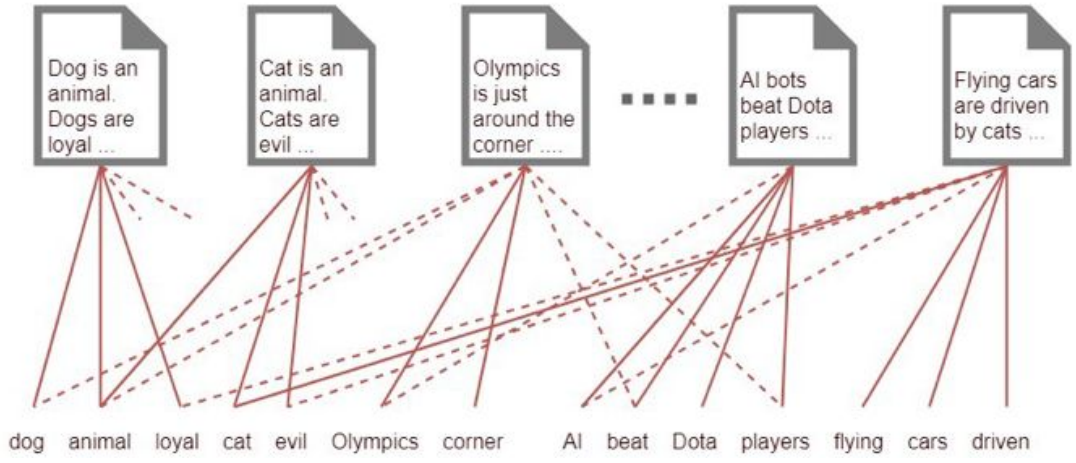

Figure 2 : Représentation du fonctionnement du modèle statistique LDA

Le score de cohérence permet d'évaluer la qualité des thèmes appris. Il calcule principalement la co-occurrence globale en prenant les mots deux-à-deux dans un même thème. Plus le score de cohérence est élevé, plus il y a de chance que les mots d'un même thème se suivent.

La libraire pyLDAvis (Sievert et al, 2004) a également été utilisée pour inspecter les performances du modèle LDA en termes de cohérence et de pertinence. Pour être un peu plus clair, un bon sujet aura tendance à avoir un regroupement d'un grand nombre de mots et sa composition sera cohérente, il y aura une bonne corrélation entre un grand nombre de mots. Tous les thèmes doivent être bien couverts, c'est à dire qu'il faut qu'il soit composé d'un nombre important de mots, et que les mots se recoupent le moins possible (il faut pas qu'on trouve le moins de mots possible dans plusieurs

regroupements différents).

Bien que cette technique soit promettante, il existe quelques inconvénients. En effet, ce n'est pas une méthode simple à mettre en oeuvre. Il est nécessaire de calibrer le modèle afin de s'approcher des performances maximales du modèle. Parmi les principales problématiques, on retrouve les performances du modèle qui sont grandement affectées par les hyperparamètres alpha et bêta. Cela implique de suivre un protocole de calibration dédié. Par ailleurs, le score de cohérence est très difficile à interpréter tout comme les résultats donnés par l'outil de gensim, pyLDAvis. En effet, les mots auxquels on a appliqué le stemmer (fonction permettant de récupérer que la racine d'un mot) ne sont pas très claires, il est nécessaire d'inspecter manuellement chaque résultat car ils est très dur d'en tirer des modélisations.

Méthodes de classification de type "boîte blanche"

En plus du réseau neuronal convolutionnel, il est possible d'utiliser d'autres modes de classification des offres de stage de type "boîte blanche". Ce sont des algorithmes de classification qui rendent les résultats explicables contrairement aux méthodes de type "boîte noir" où les résultats sont beaucoup plus obscures comme le deep-learning.

La première méthode utilisée est le classifieur de type Naive Bayes Multinomial. Cette méthode est souvent utilisée dans l'exploration de textes même si une des hypothèses de base de la méthode Bayésienne naïve soit qu'il y ait une indépendance entre les différentes fonctionnalités d'entrée. Hors ce n'est pas le cas en classification de textes.

Une autre méthode est un algorithme de type arbre de classification, aussi appelé random forest. On combine cet algorithme avec la classification des tf-idf pour avoir une plus grande précision.

Nouvelle base de données

Une fois les nouveaux objectifs déterminés, nous avons dû revoir la base de données. En effet, au départ la base de données était seulement composée des tables nécessaires pour faire fonctionner le deep learning.

Notre Product Owner a rencontré une gestionnaire de scolarité pour définir les nouvelles données nécessaires pour le persona Gestionnaire, et nous nous sommes mis d'accord sur les données liées au persona Étudiant.

Nous avons donc rajouté plusieurs tables pour nos besoins. Nous avons, dans un premier temps, créé les tables pour les différents utilisateurs puis les tables pour les écoles, les formations et les entreprises.

Voici la nouvelle base de données:

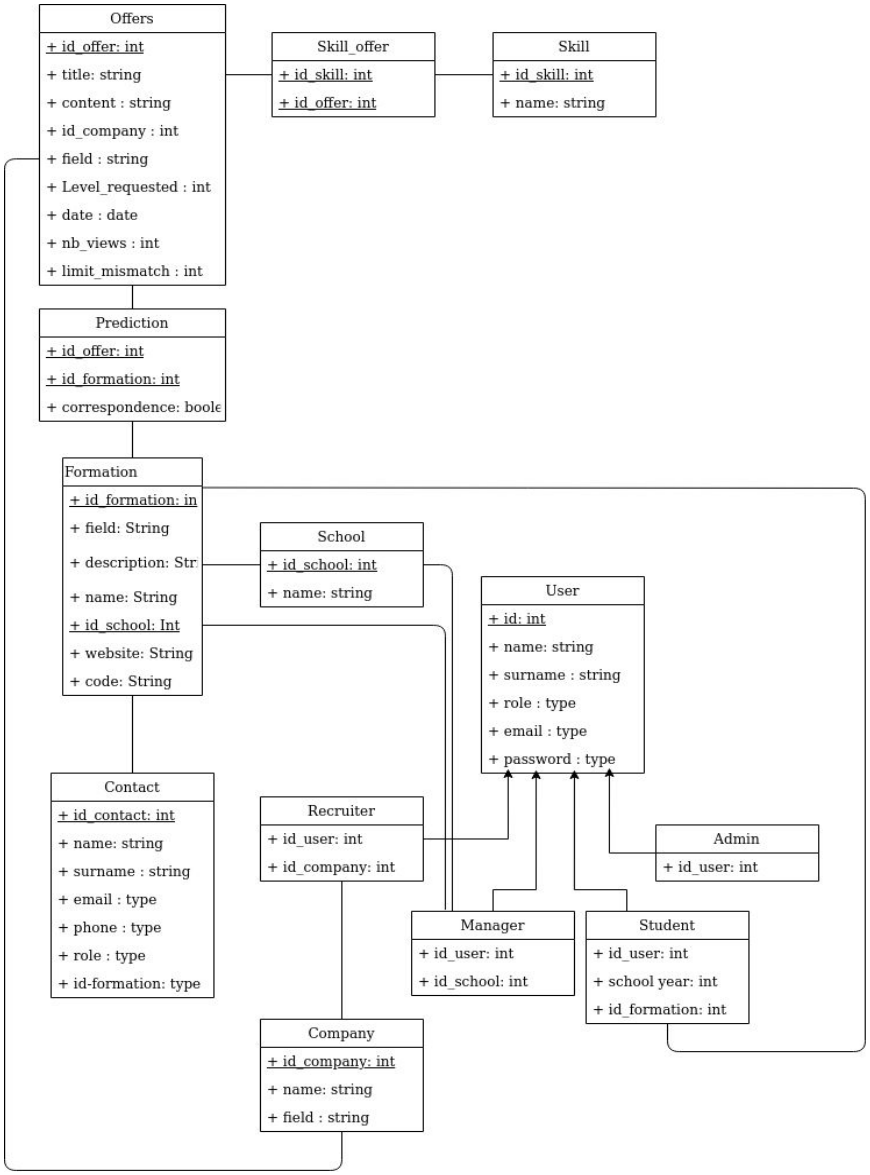

Figure 3 : Architecture de la base de données

Nous avons donc également revu comment remplir les tables, cependant nous avons rencontré un problème pour remplir entièrement l'ensemble des champs des tables de Offers, Formations, Contact etc. En effet, il nous manquait des données. Nous avons donc rempli nous-même certaines bases pour avoir le plus d'informations possible pour pouvoir faire les statistiques demandées pour le persona Gestionnaire.

Nous avons essayé d'anticiper pour le futur, c'est-à-dire rajouter dans la base, les données pour que le persona Étudiant puisse être réalisé facilement sans à avoir à reconstruire la base de données.

Diagnostic sécurité

Étant donné que le but de ce projet était d'avoir une première version livrable, l'aspect sécurité de l'application n'était pas une priorité.

Néanmoins, nous avons tout de même pris certaines dispositions simples en vue d'avoir une application ne possédant pas de failles évidentes. Cela s'est traduit d'une part par un respect de certaines règles dans l'écriture du code, comme par exemple ne pas avoir de code "mort" ou de variables ou fonctions non utilisées. Mais cela s'est également traduit par un restriction des accès API aux utilisateurs connectés. Pour la connexion, nous avons également fait attention à la sécuriser un minimum avec des tokens JWT pour l'authentification ainsi que des gardes de sécurités. Néanmoins, par manque de temps nous n'avons pas pu vérifier soigneusement si le système d'authentification mis en place est bien sécurisé ou non, ce serait donc une première étape pour une reprise du projet.

Il subsiste évidemment de nombreux aspects où la sécurité de l'application n'est pas optimale. En effet, nous n'avons tout d'abord pas d'administrateur général de l'application pouvant gérer les différents users connectés. Ainsi n'importe qui peut s'inscrire en se déclarant par exemple gestionnaire de scolarité de Polytech sans qu'il n'y ait de validation derrière. C'est un aspect auquel nous avons bien évidemment pensé néanmoins nous n'avons pas eu le temps nécessaire pour le gérer. De la même manière, il n'y a pas de vérifications lors de la création d'une entreprise. Avoir une gestion des mails afin de gérer la fonctionnalité de mot de passe oublié aurait aussi été une bonne chose.

Pour finir, l'application que nous avons développé met en forme des données qui peuvent être sensibles, il serait donc essentiel dans le future de s'assurer que l'accès à ses données soit bien protégé.

Bibliographie

https://disrupt-campus.univ-grenoble-alpes.fr

https://towardsdatascience.com/a-beginners-guide-to-word-embedding-with-gensim-word2vec-model-5970fa56cc92

https://fr.ryte.com/wiki/TF*IDF

https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158

https://fr.wikipedia.org/wiki/MySQL

https://www.infoworld.com/article/3336192/what-is-keras-the-deep-neural-network-api-explained.html

https://pymbook.readthedocs.io/en/latest/flask.html

https://agiliste.fr/introduction-methodes-agiles/

https://mdbootstrap.com/freebies/react/admin-dashboard/

https://jasonwatmore.com/fr/post/2019/06/10/angular-8-tutoriel-et-exemple-sur-lenregistrement-et-lauthentification-des-utilisateurs

https://stackblitz.com/edit/angular-packed-bubble-chart