



كلية العلوم
السمالية - مراكش
FACULTÉ DES SCIENCES
MARRAKECH - SEMLALIA

UNIVERSITÉ CADI AYYAD FACULTÉ DES SCIENCES SEMLALIA

DÉPARTEMENT D'INFORMATIQUE

SCIENCES MATHÉMATIQUES ET INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ETUDES

Bio generator for Junior Data Scientists

RÉALISER PAR: CHEGRI MOHAMED

Membre de jury:

Dr. Zahir jihad

Dr. MOUSSANIF Hajar

Encadré par:

Dr. Zahir jihad

15 juillet 2021

Dédicace

A ma très chère maman Nezha,

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive. Que Dieu vous protège. A mes chers ami(e)s, En souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble. Veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.

Remerciement

Avant tout remerciement, louange à Dieu. Les mots ne peuvent pas exprimer notre gratitude pour **Dr. Jihad Zahir**, notre honorable encadrant qui nous à accompagner pendant tout le chemin vers l'accomplissement de ce projet de fin d'études, nous ne pouvions pas trouver un plus mieux encadrement, nous sommes très révéres d'avoir la chance de travailler avec un tel encadrant. Nous tenons aussi à adresser nos profondes reconnaissances et nos extrêmes gratitudes au jury de notre soutenance de projet de fin d'études, qui a accepté d'évaluer notre travail.

Également, nous remercions l'ensemble du corps enseignant à la Faculté des Sciences Semlalia Marrakech et particulièrement celui du Département Informatique qui nous a incités pendant notre formation et qui a posé dans nos mains le savoir-faire, qui nous a aidés à bien mener ce travail.

Ainsi, nous adressons nos remerciements à la direction de Faculté des Sciences Semlalia Marrakech.

Enfin, que tous ceux qui ont participé de près ou de loin à l'élaboration de ce travail, trouvent ici le témoignage de nos sincères remerciements.

Avant propos

La formation licence fondamentale mathématiques et informatiques (SMI) du département informatique de la faculté des sciences Semlalia Marrakech est complété par un stage obligatoire d'un mois, du 1er Mai au 30 Juin, afin d'adapter la formation académique aux exigences du marché de emploi, rendre plus pratique les connaissances théoriques acquises durant les années d'études, donner une vision du domaine du travail et préparer l'étudiant à l'insertion au domaine professionnel. J'ai effectué ce stage au sein de département informatique de la faculté des sciences Semlalia Marrakech, la mission était comme suit de crée un application web qui aide les les jeunes Data Scientistes de mieux rédiger leur biographies sur Linkedin utilisant l'une des principales technologies de Machine Learning et d'intelligence artificielle.

Résumé

Dans ce rapport, nous allons proposer une application de génération des biographies pour les jeunes diplômés en se basant sur NLP "Natural Language Processing" afin de généraliser le processus pour les autres domaines professionnels. Notre travail a été divisé en deux grand partie, la première consiste a collecter du data depuis le site Internet LinkedIn, pour ceci on a utilisé la bibliothèque d'automatisation de python Selenium. Après la collecte des données, on faisait du data clining en utilisant des scripts de python, avant d'entraîner nos modèles pour générer des nouveaux texte en utilisant le paquet du python textgenrnn".

La deuxième partie est la représentation du résultat, pour cela on a réalisé une application web qui sera une interface graphique afin d'interagir avec l'utilisateur, pour ceci on a utilisé le micro fremework flask et le déployer sur un contenant de docker composé pour garantir une indépendances entre notre environnement du travail et le host de l'application.

Table des figures

1.1	Architecture d'aplication web.	1
2.1	Architecture de réseau neuronal.	4
2.2	Un exemple de RNN	5
2.3	Génération de nouveau biographie	6
2.4	Modèle conceptuel data base	6
2.5	Présentation de l'application web	7
2.6	Modèle conceptuel data base	8
2.7	Modèle data base	8
2.8	algorithme d'utilisation	8
2.9	biographie exemplaire pour un data scientifique.	9
2.10	Biographie graphie générée automatiquement pour un datas scientifique	9
2.11	Biographie construit pour l'utilisateur alex	10
2.12	Présentation du site	11

Liste d'abréviations

ANN	Artificial Neural Network
NLP	Natural language Processing
ML	Machine Learning
HTML	Hyper Text Markup Language
JSON	JavaScript Object Notation
API	Interface de programmation
XML	Extensible Markup Language
LSTM	long short-term memory
RNN	Recurrent neural network

Table des matières

Dédicace	ii
Remerciement	iii
Avant propos	iv
Résumé	v
Liste d'abréviations	vii
Introduction générale	ix
1 Étude théorique de projet	1
2 Implémentations de projet	3
2.1 Récupération du data	3
2.2 Nettoyage du carpus textuel	3
2.3 Entraîner du dataset	4
2.4 Génération du nouveau biographie	5
2.5 Modélisation de l'application web	6
2.5.1 Création d'une web application	7
2.5.2 Clasification des biographie	8
2.5.3 les biographies exemplaire	8
2.5.4 Les biographies généraix automatiquement	9
2.5.5 Biographie sur demande	10
2.6 Vue globale de l'application Web	10
3 Environnement de travail	12
3.1 Flask	12
3.2 DJANGO	12
3.3 FLASK OR DJANGO	13
3.4 HTML BOOTSTRAP	13
3.5 JINJA	14
3.6 Docker	14

Introduction générale

Après avoir fini les études, les carrières s'apparaissent, le premier problème qui affronte les nouveaux lauréats est la difficulté de trouver un stage pré-embauche ou un emploi. Pour ce faire ils ont besoin d'un réseau qui contient des professionnels de leurs domaines, ce réseau va les aider pour initialiser leurs carrières et avoir une vue globale du marché d'emploi est ses besoins.

LinkedIn par exemple est un réseau qui contient 660 millions de membres dans le monde en 2021, avec 69% des membres active, les utilisateurs déposent leur cv et une description de leurs parcours académique et professionnels, en USA par exemple 87% des recruteurs examineront la candidate en visitant leur profil LinkedIn. Google aussi utilise le réseau LinkedIn pour recruter des nouveaux employés.

Un bon profil LinkedIn commence par un résumé fantastique, qui résume la carrière et le parcours académique de l'utilisateur, ce dernier nécessite des techniques d'écritures pour une bonne rédaction et description du profil.

Avoir un outil qui suggère des exemples des biographies pour que les utilisateurs puissent s'inspirer de la méthode d'écriture des professionnels, et pour créer leurs propres biographies ou avoir un système intelligent qui le rédige pour eux et pourquoi pas on donne les informations personnelles de l'utilisateur, avoir une biographie générée automatiquement et plus spécifique.

Pour répondre à cette problématique, on va créer une application web qui génère d'une façon automatique les biographies pour les jeunes data scientistes en utilisant une technologie de Machine Learning qui s'appelle le deep learning. Pour ce fait, on va faire appel à la bibliothèque open source écrite en python Keras et la plateforme tensorflow afin de déployer le modèle de machine learning via le micro framework flask et docker-compose afin de créer une interface graphique qui permet aux jeunes data scientistes de générer une biographie par un clic.

Chapitre 1

Étude théorique de projet

Au début on doit définir notre environnement de travail ainsi que les bibliothèques et les algorithmes utilisés dans ce projet et l'architecture qu'on doit suivre pour créer notre application .

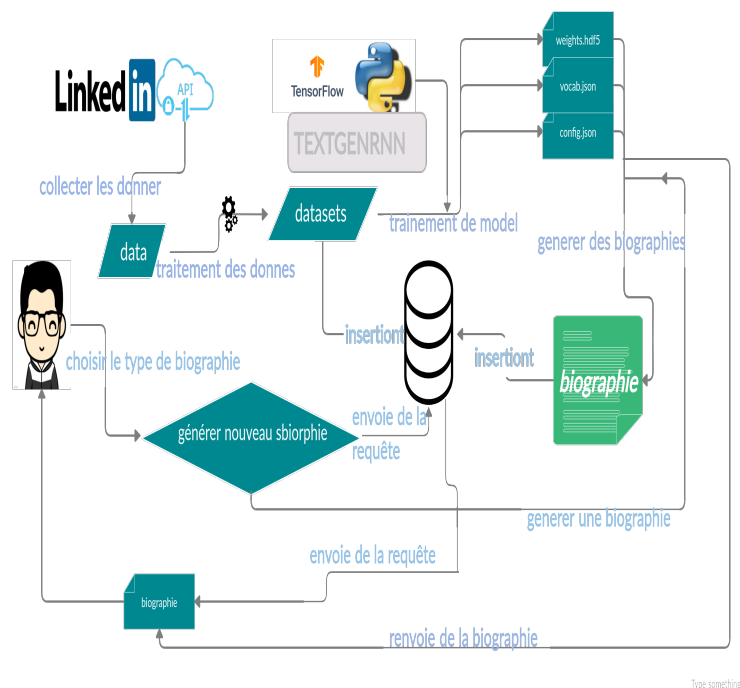


FIGURE 1.1 – Architecture d'application web.

Machine Learning : est une technologie d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Pour apprendre et se développer, les ordinateurs ont tout de même besoin des données à analyser et sur lesquelles s'entraîner. De fait.

Deep learning : Ou l'apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées jeu de données : (en anglais dataset ou data set)

selon wikipedia est un ensemble de valeurs (ou données) où chaque valeur est associée à une variable (ou attribut) et à une observation. Une variable décrit l'ensemble des valeurs décrivant le même attribut et une observation contient l'ensemble des valeurs décrivant les attributs d'une unité.

TensorFlow : TensorFlow est une bibliothèque de logiciels open source pour le calcul numérique utilisant des graphiques de flux de données. Les nœuds du graphique représentent des opérations mathématiques, tandis que les bords du graphique représentent les tableaux de données multidimensionnels (tenseurs) communiqués entre eux. L'architecture flexible vous permet de déployer le calcul sur un ou plusieurs CPU ou GPU sur un ordinateur de bureau, un serveur ou un appareil mobile avec une seule API.

Keras : Est une API Python basée sur un réseau neuronal de haut niveau qui s'exécute sur CPU ou GPU. Il prend en charge les réseaux convolutionnels et récurrents et peut s'exécuter au-dessus de TensorFlow, CNTK ou Theano. L'objectif principal est de permettre aux utilisateurs une expérimentation rapide. Il existe de nombreux exemples d'utilisation de Keras dans le tableau comparatif : classification, génération et résumé de texte, balisage, analyse, traduction automatique, reconnaissance vocale et autres.

RNN : Recurrent neural network, est une classe de réseaux de neurones artificiels où les connexions entre les nœuds forment un graphe orienté tout au long d'une séquence temporelle. Cela lui permet de présenter un comportement dynamique temporel. Les RNN peuvent utiliser leur état interne (mémoire) pour traiter des séquences d'entrées de longueur variable. Cela les rend applicables à des tâches telles que la reconnaissance d'écriture manuscrite ou la reconnaissance vocale .

LSTM Long short-term memory, La mémoire à long court terme est une qui possède une mémoire interne appelée cellule (ou cell). La cellule permet de maintenir un état aussi longtemps que nécessaire. Cette cellule consiste en une valeur numérique que le réseau peut piloter en fonction des situations, Il peut non seulement traiter des points de données uniques (tels que des images), mais également des séquences entières de données (telles que de la parole ou de la vidéo). Par exemple, LSTM est applicable à des tâches telles que la reconnaissance d'écriture manuscrite, la reconnaissance vocale.

Char-rnn : Une implémentation d'un multi-layer Recurrent Neural Network(RNN, LSTM) pour l'apprentissage à partir de modèles de langage au niveau des caractères. En d'autres termes, le modèle prend un fichier texte en entrée et forme un réseau neuronal récurrent qui apprend à prédire le caractère suivant dans une séquence. Le RNN peut ensuite être utilisé pour générer du texte caractère par caractère qui ressemblera aux données d'apprentissage d'origine.

NLP Exactement ce que cela ressemble, les techniques utilisées pour permettre aux ordinateurs de comprendre le langage humain naturel, plutôt que d'avoir à interagir avec les gens à travers des langages de programmation. Le traitement du langage naturel est nécessaire pour des tâches telles que la classification de documents Word ou la création d'un chatbot.

Implémentations de projet

La création de notre application est divisée en cinq phases, la première est la récupération du data en utilisant le web scraping, puis l'exploration du dataset en deuxième phase, puis, l'entraînement du modèle en utilisant le paquet de python textgenrnn pour entraîner notre modèle, la dernière phase, c'est création d'une interface web pour visualiser le résultat de notre training et aussi avoir l'accès aux biographies qui sont écrits par des professionnels.

2.1 Récupération du data

La première étape du traitement des données du texte est de récupérer le texte et le nettoyer afin de pouvoir l'utiliser ultérieurement dans l'apprentissage, Dans mon cas, j'ai scrapé la page LinkedIn, mais malheureusement cette dernière possède une politique qui sécurise les informations des utilisateurs que ce soit publiques comme le nom, prénom, biographie, ou critique comme les numéros de téléphones et les adresses personnelles, mails, en bref LinkedIn interdit tout sort de collecte d'informations, il détecte ce comportement en calculant le nombre de requêtes envoyés par chaque utilisateur, toute personne essaye de collecter les informations, son compte sera bloquée, ou supprimer après le premier avertissement. La solution est de minimiser les requêtes envoyer aux serveurs de linkedin, et de créer de nombreux faux compte "zombies" afin d'éviter la suppression de notre compte, la bibliothèque python selenium nous permet de scraper les pages de recherches pour chaque domaine, on extrait les urls des profils et on construit une table pour chaque profession. On a utilisé aussi ce qu'on appelle Application Programming Interfacesou API, et pour cela on doit avoir une application dans Linkendin, en créant cette dernière, on a pu extraire la moitié des biographies, et la raison pour laquelle on n'a pas passé toutes les biographies c'est que le serveur bloque l'adresse IP du machine qui passe un nombre déterminé de requêtes, pour éviter les attaques de cybersécurité, la sécurité comme DDOS attack, et la moitié de d'autre urls on le passe a la bibliothèque de LinkedIn_scraper.

Le problème de cette partie c'est que nombreux profils ne possède pas de biographie ce qui fait que la plupart des requêtes n'a aucune utilité, ainsi que de nombreux résumés sont écrit en d'autre langue que du l'anglais,

2.2 Nettoyage du corpus textuel

La premier étape après la récupération de corpus et d'analyser et nettoyer en créant des scripts qui supprime toute biographie qui n'utilise pas l'encodage ASCII qui peuvent

mener à des erreurs faciles à éviter est chaque caractère doit être affichable.

En utilisant la bibliothèque textblob du python on va classifier les biographies selon leur langues avant de supprimer tout biographie écrit en autre langue que la langue anglaise, a cause de modèle textgenrnn qui entraîne le dataset utilise le vocabulaire anglais et toute biographie non anglaise peut avoir des résultats non souhaitable.

2.3 Entraîner du dataset

De nombreux ressources de génération de texte utilise les réseaux de neurones, copient simplement une implémentations de char-rnn existante tout en modifiant l'ensemble de données d'entrée (dataset). C'est une approche, mais il existe une opportunité d'amélioration avec les outils modernes d'apprentissage en profondeur. Grâce à des frameworks comme TensorFlow et Keras. Textgenrnn est un package Python qui résume le processus de création et de formation de tels char-rnns à quelques lignes de code , avec de nombreuses améliorations d'architecture de modèle et de formation telles que les intégrations de caractères. Un avantage intéressant de textgenrnn est qu'il peut être facilement utilisé pour entraîner très rapidement des réseaux de neurones sur un GPU.

Textgenrnn traite automatiquement le texte saisi en séquences de caractères prêtes à entraîner le réseau. Après toutes les 2 époques (un passage complet à travers les données), le réseau générera un échantillon de texte à différentes températures, qui représentent la «créativité» du texte (c'est-à-dire qu'il permet au modèle de faire des prédictions de plus en plus optimales. Dans notre modelé on a utiliser Les options de configuration : 256 rnns, 256 passages complets des données , 16 rnn layers des séquences de modèles à entraîner dans une étape.

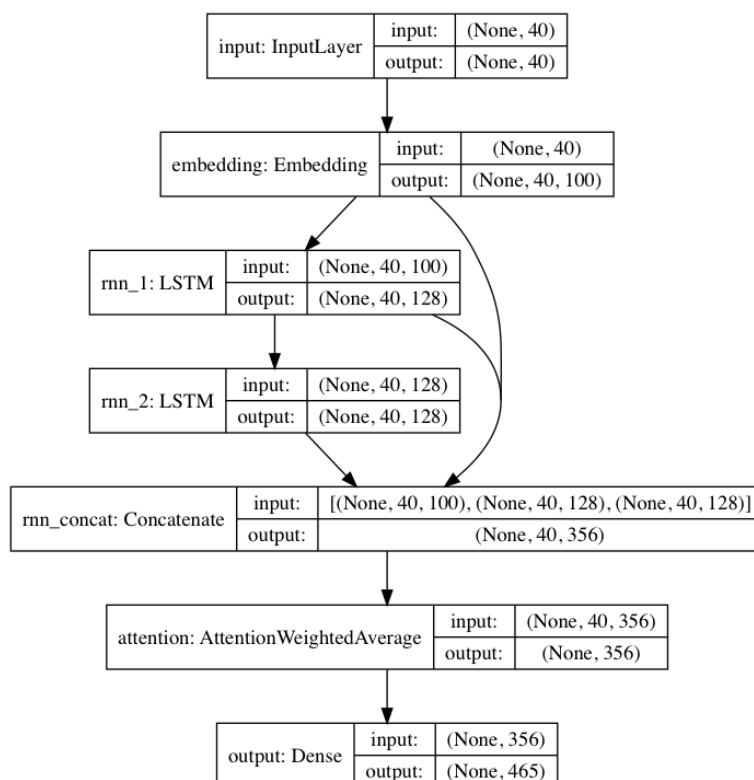


FIGURE 2.1 – Architecture de réseau neuronal.

Le modèle par défaut, textgenrnn prend une entrée de jusqu'à 40 caractères, convertit chaque caractère en un vecteur d'intégration de caractères 100-D et les alimente dans

une couche récurrente de 128 cellules de mémoire à long-court terme (LSTM). Ces sorties sont ensuite introduites dans un autre LSTM à 128 cellules. Les trois couches sont ensuite introduites dans une couche d'attention pour pondérer les caractéristiques temporelles les plus importantes et les moyenner ensemble (et puisque les intégrations + le 1er LSTM sont connectés par saut dans la couche d'attention, les mises à jour du modèle peuvent s'y propager plus facilement et éviter la disparition dégradés). Cette sortie est mappée sur des probabilités pour un maximum de 394 caractères différents qu'il s'agit du caractère suivant de la séquence, y compris les majuscules, les minuscules, la ponctuation et les emoji.

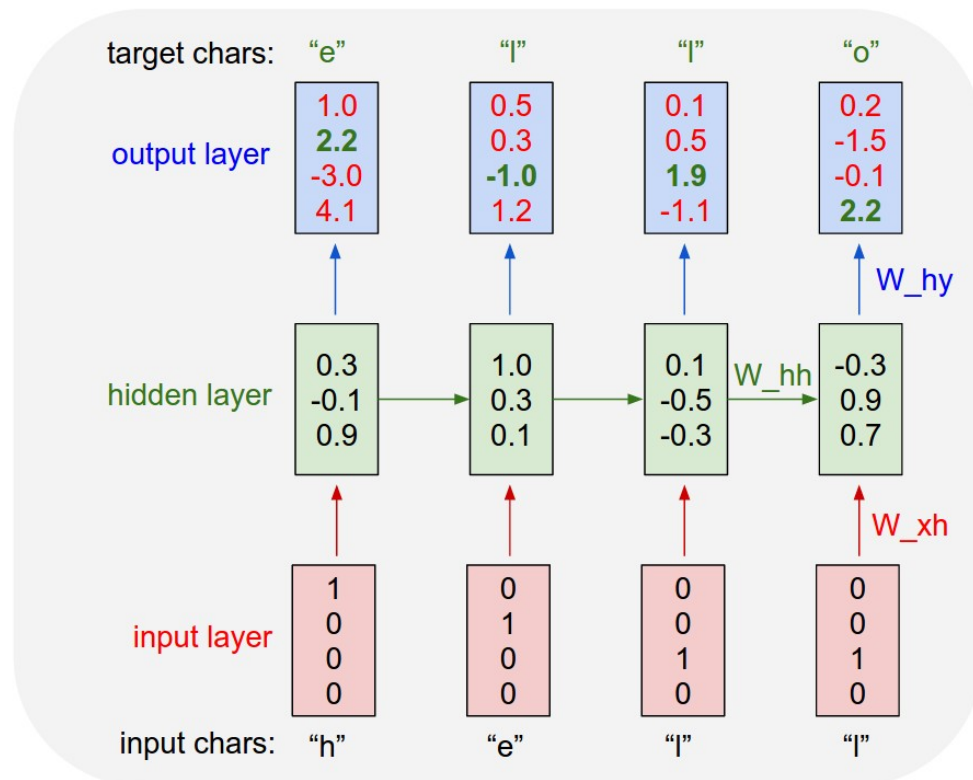


FIGURE 2.2 – Un exemple de RNN

2.4 Génération du nouveau biographie

Une fois la modèle terminée, on télé-chargera trois fichiers : un fichier contient le pois , vocabulaire, configuration du modèle qui sont tous nécessaires pour régénérer les nouveaux biographie, en générant de nouveaux biographie le programme prend plus de 5 secondes, se qui est une anomalie pour l'utilisateur qui peut pas attendre 6 secondes pour générer de nouveau biographie. Pour cette raison on génère plus de 1000 biographie et stocker ces derniers dans une base de données. Pour chaque domaine professionnel. En utilisant un script python qui va générer pour chaque domaine professionnelle 1000 biographie et le stocker dans une base de données afin d'utiliser ce dernier pour interagir avec l'utilisateur

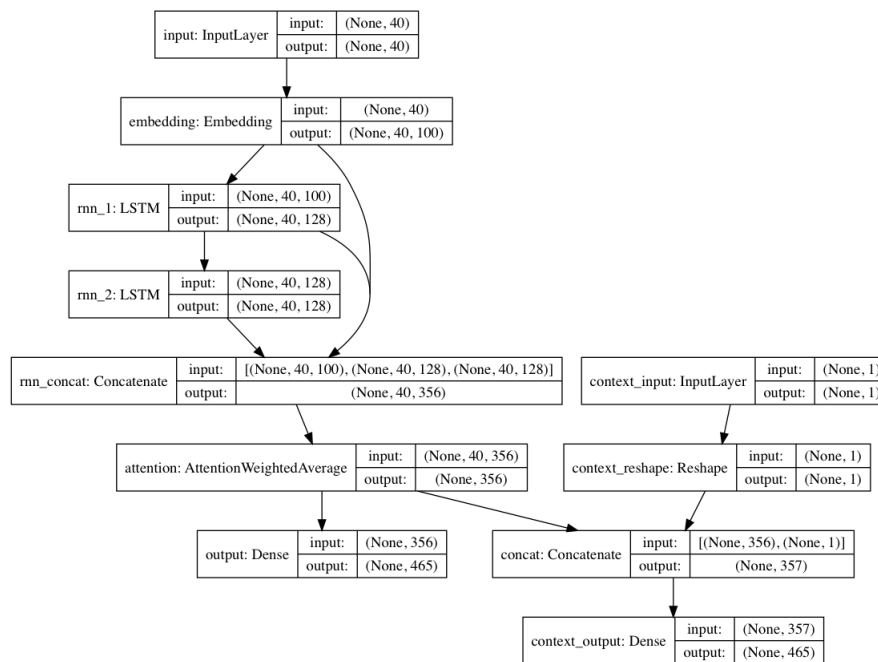


FIGURE 2.3 – Génération de nouveau biographie

2.5 Modélisation de l'application web

une page web pour premierment intragir avec l'utilisateur et de La création de l'appli-
cation web sera à la fois présenter a l'utilisateur des biographie exemplaire crée par des
experts et aussi des biographie générer par notre modelé qui été générer ultérieurement

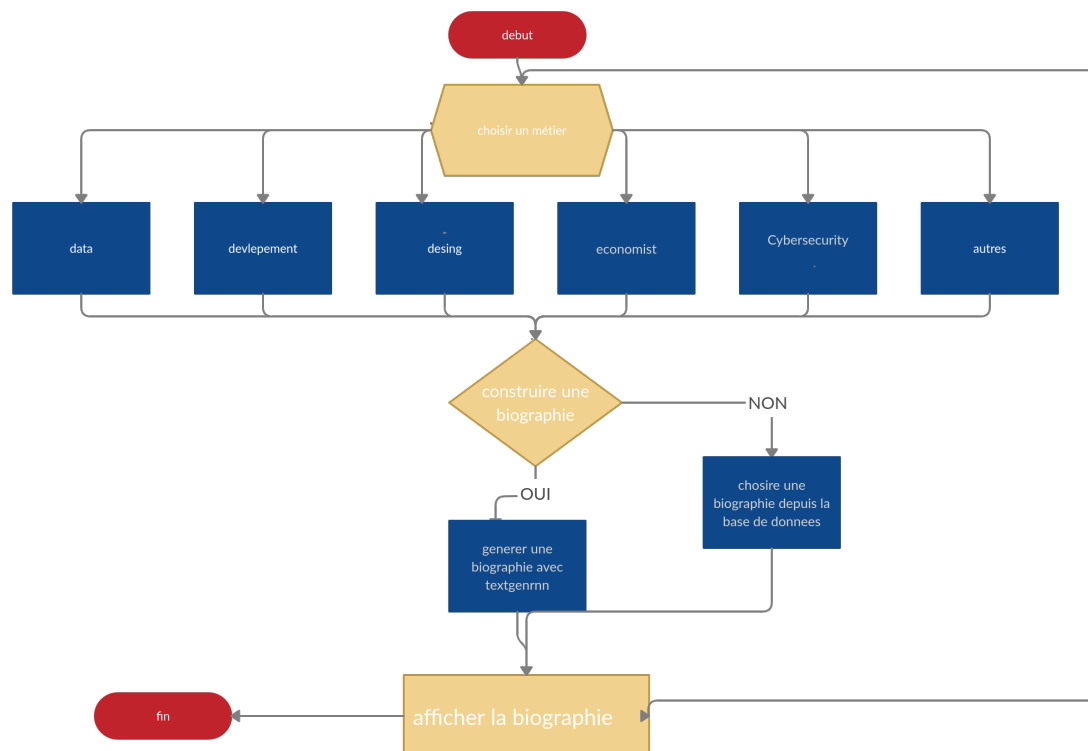


FIGURE 2.4 – Modèle conceptuel data base

2.5.1 Création d'une web application

premièrement on doivent définir notre application web et les raisons pour la quel a été crée , et les motives qui pous l'utilisateur a utiliser ainsi que l'importance de biographies dans le domaine professionnel, en déclarons les service que l'application servit.

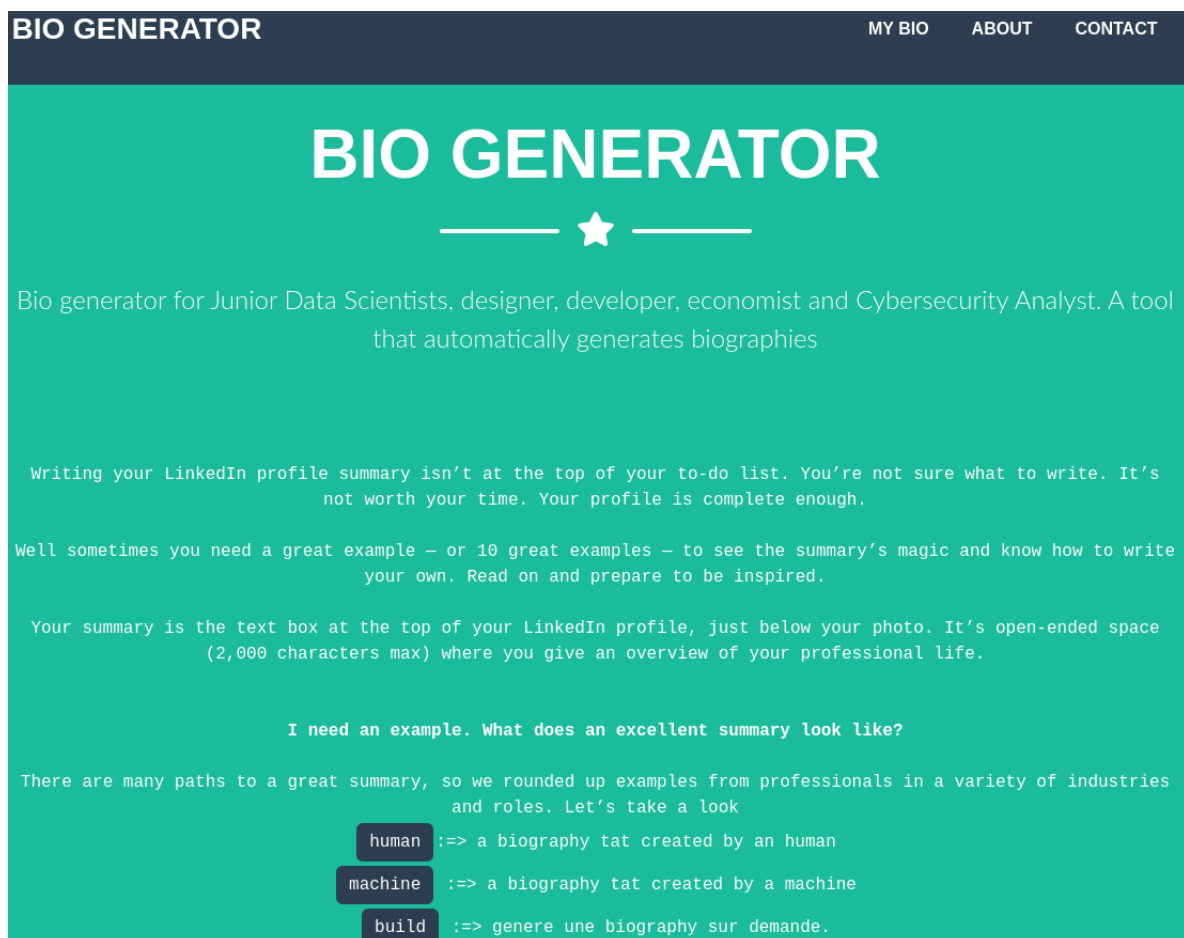


FIGURE 2.5 – Présentation de l'application web

Les biographie serons classer en premier classe par profession la même chose que du classification lors du collecte du data puis pour chaque profession on trouve trois sous classe

Human :les biographies exemplaire crée par des expert afin d'inspirer et remarquer la méthode de création des biographie machine : des biographie génère automatiquement en utilisant le modèle génère ultérieurement

"build" : création des biographie en utilisant les information des utilisateur en paramètre pour génère des biographie pour chaque utilisateur en ce basent sur le nome et le métier ou de quel domaine d'apprentissage s' il est étudiant, et l'entreprise dans la quel travaille ou étudier.

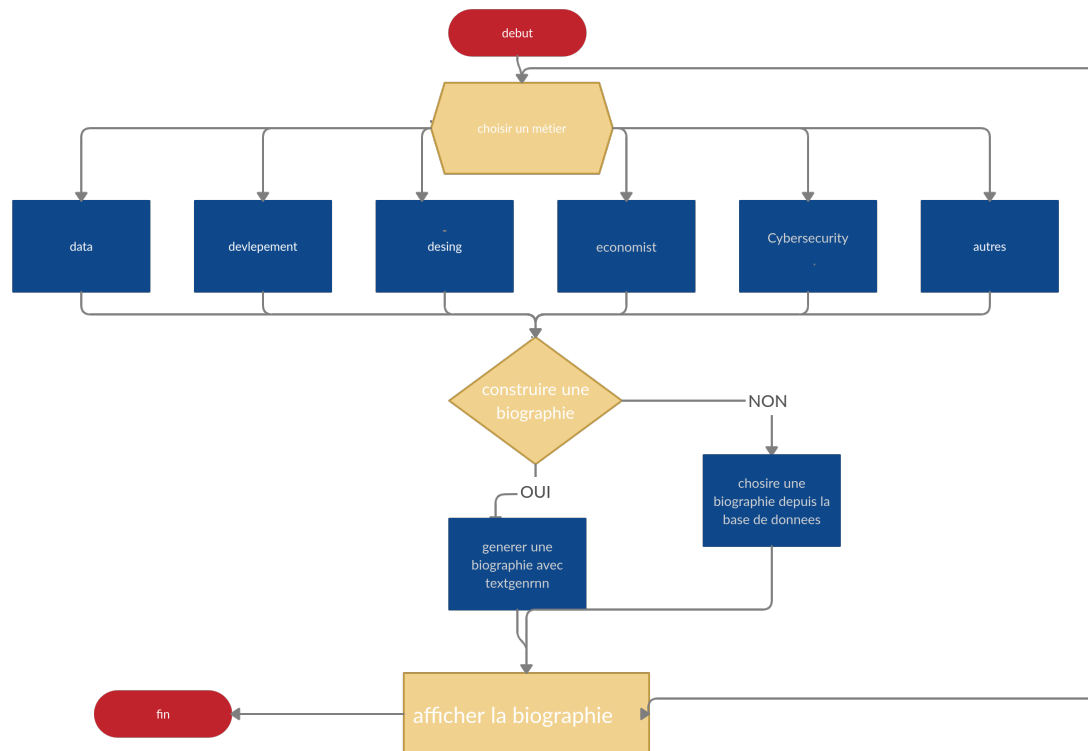


FIGURE 2.6 – Modèle conceptuel data base

2.5.2 Clasification des biographie

les biographies seront classer dans l'ordre dans la quel, la collections du data set a été fais, avec l'ajout d'une nouveau partie qui globalise tous les autre domaine cette dernier a été ajouter pour traîner tout les biographie collecter depuis linkedin sans prendre en considération les profession.

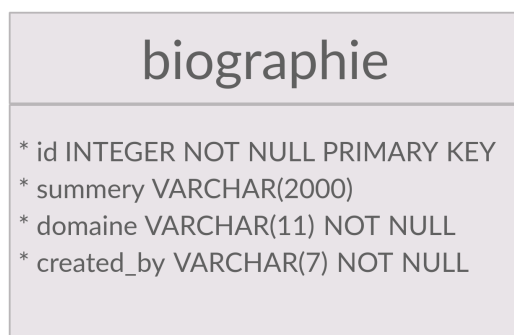


FIGURE 2.7 – Modèle data base

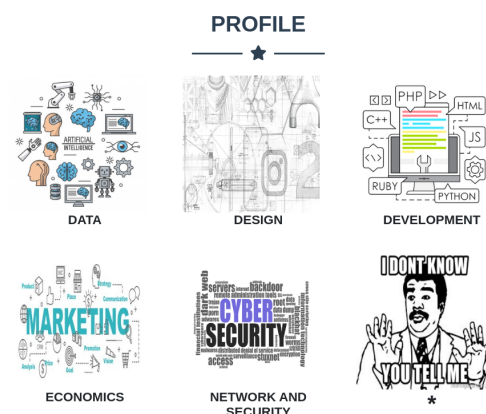


FIGURE 2.8 – algorithme d'utilisation

2.5.3 les biographies exemplaire

Dans cette rubrique on va donner la possibilité aux utilisateur la possibilité de lire les biographie collecter depuis linkedin, et avoir une idée comment les experts écrit leur

biographie et aussi avoir une vue sur d'autre domaine profériions, peut être qui il va trouver un profil qui va intersectée avec sa propre carier.

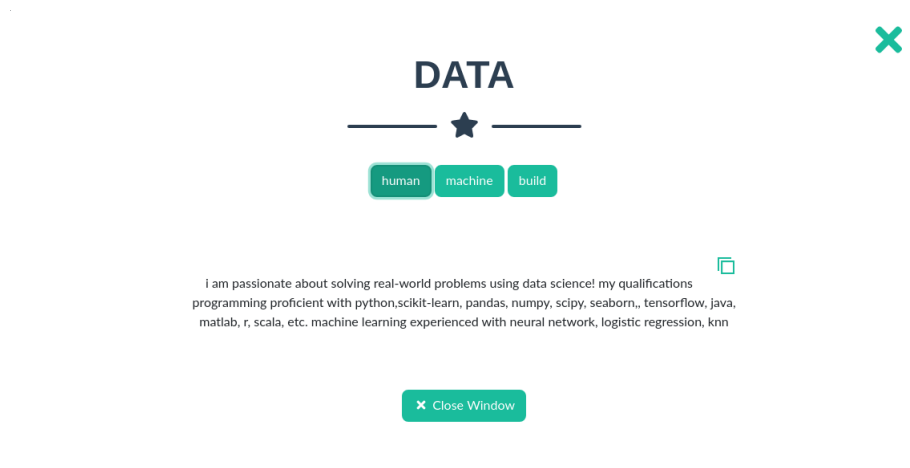


FIGURE 2.9 – biographie exemplaire pour un data scientifique.

2.5.4 Les biographies généraies automatiquement

Dans cette parte l'utilisateur peut choisir des biographies qui sont déjà génère par la machine et stocker dans une basse de donner pour l'utilisateur l'avoir immédiatement par-ce que la génération instantané des biographie prend entre 4 et 12 seconde qui est un très grand ce qui rendre difficile de choisie une bon biographie.

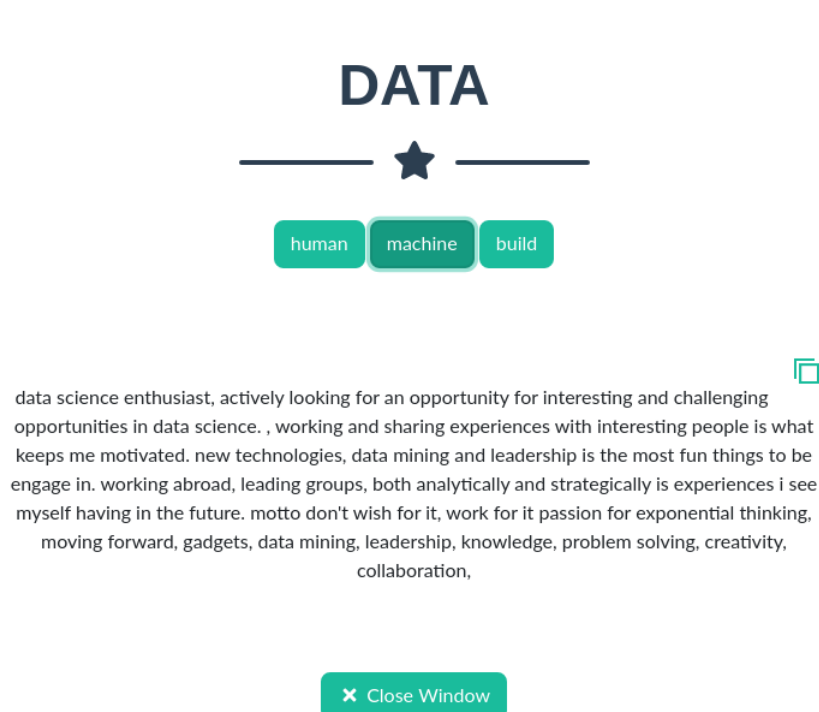


FIGURE 2.10 – Biographie graphie générée automatiquement pour un datas scientifique

2.5.5 Biographie sur demande

la section le plus important et qu'on donne a l'utilisateur de la possibilité de génère une biographie qui comporte son propre nom ainsi sa profession son lieu de travaille aux d'étude.

The screenshot shows a web application window titled "DATA" with a star icon. Below the title are three buttons: "human", "machine", and "build". The "human" button is selected. Below these buttons are three input fields: "Full_name:" with the value "Alex Drysdale", "Job:" with the value "Junior Web Developer", and "compañie:" with the value "Oswald Technologies". A "generate" button is to the right of the "compañie:" field. Below the input fields is a text area containing a generated biography for Alex Drysdale, a Junior Web Developer at Oswald Technologies. The biography describes his background in engineering, technology, and data science. At the bottom of the window is a "Close Window" button.

FIGURE 2.11 – Biographie construit pour l'utilisateur alex

2.6 Vue globale de l'application Web

Après avoir termine la dernière partie du backend, on passe a la partie de devops. Cette partie sera consacrer au containeriser l'application web pour assure l'indépendance entre les différentes facteurs et aussi de faire une l'application portable afin d' tout prochaine installation quelque soit la machine host et leur système d'exploitation serveur il suffit d'avoir un le programme docker seulement, cette solution a été utilise pour plusieurs raison, la première vu les erreurs de la bibliothèque textgenrnn qui génère des erreurs qui doivent être parser pour assurer le bon lancement du programme, les erreurs viennent a la fois de la compatibilité entre kerase et tensorflow et entre le programme lui mémé et le matériel. La dernière raison c'est d'installer nginx et le configurer nos propre nom domaine en local.

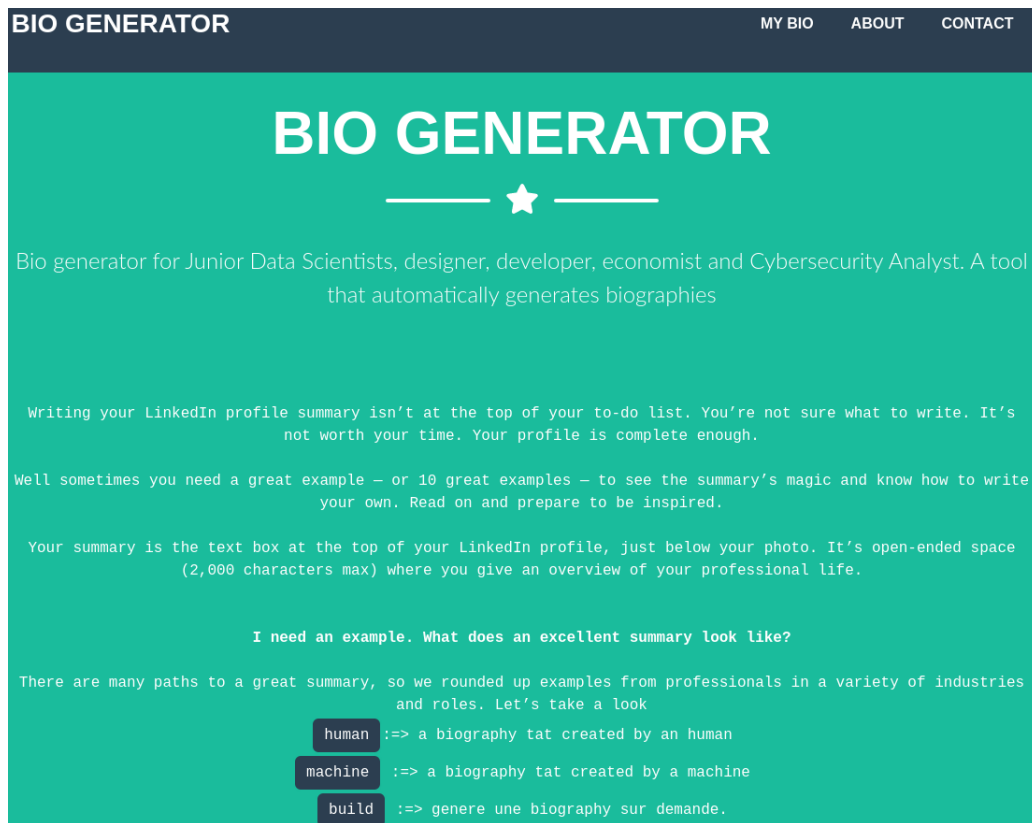


FIGURE 2.12 – Présentation du site

Environnement de travail

Colaboratory, souvent raccourci en "Colab", nous permet d'écrire et d'exécuter du code Python dans notre navigateur. Il ne nécessite aucune configuration requise plus l'accès gratuit aux GPU mais avec des des limite de taux d'apprentissage.

Vscode : l'éditeur de texte le plus utiliser dan le domaine de coding car il s'adapte avec tout les environnement de travail, web, mobile, réseau. Et aussi possède d'une très grand bibliothèque des extention qui facilite la tache aux programeur ainsi une paquet intègre de github qui facilite la gestion d'historique de code

3.1 Flask

Flask est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de La programmation Python, associé à un système de templates. Il est distribué sous licence BSD Flask a été créé initialement par Armin Ronacher comme étant un poisson d'avril, Flask se base sur deux modules werkzeug et jinja2 pour proposer plusieurs des fonctionnalités suivantes : Serveur de développement⁶ et debugger Simplifie l'écriture de tests unitaires⁷ Moteur de template pour le rendu HTML⁸ Supporte les cookies sécurisés (session) etc.

3.2 DJANGO

Django est un Framework Python full-stack et développé, ce qui facilite à ses utilisateurs certaines taches communes de développement, comme l'authentification, le routage d'URL, ou encore la migration des schémas de base de données. Django accélère également le développement en fournissant des Template, un système ORM, et un outil de bootstrapping. Composé de trois parties distinctes : 1. Un langage de gabarits flexible qui permet de générer du HTML, XML ou tout autre format texte ; 2. Un contrôleur fourni sous la forme d'un « remapping » d'URL à base d'expressions rationnelles ; 3. Une API d'accès aux données est automatiquement générée par le cadre compatible CRUD. Inutile d'écrire des requêtes SQL associées à des formulaires, elles sont générées automatiquement par l'ORM.

3.3 FLASK OR DJANGO

Avant d'insister sur leurs différences, soulignons tout de même que Flask et Django restent tous les deux des Framework utilisables pour Python et sont dans l'ensemble assez similaires. Pour les deux, on retrouve la logique de la structure MVC. Il est donc assez simple de passer d'un Framework à l'autre sans perdre ses repères. Flask, quant à lui, est un Framework simple, léger, minimaliste. Bien que Flask offre moins de fonctionnalités que Django, celui-ci facilite le développement et garantit une meilleure souplesse. Django facilite la prise en main des tâches classiques d'administration grâce à un Framework admin prêt à l'emploi. Il génère le module admin automatiquement en fonction des modèles du projet. Les développeurs peuvent même customiser l'interface admin afin que celles-ci s'adaptent à des exigences business spécifiques. De ce point de vue, Django offre donc plus de possibilités que Flask, qui ne propose pas ces options. Flask est développé sur la base du template Jinja2. Ce Template intègre un environnement d'exécution Sand Box et de nombreux modèles permettant aux développeurs de gagner du temps sur le développement d'applications web dynamiques. Django propose un système de templates entièrement intégré permettant aux développeurs de définir un modèle et de customiser l'interface utilisateur en écrivant de nouveaux templates dans le langage Django (DTL). Flask n'intègre pas d'outil de Bootstrapping, contrairement à Django, qui propose l'outil `django-admin`. Ce dernier permet de commencer à développer des applications sans aucun input externe. Les développeurs utilisant Django peuvent diviser un seul et même projet en différentes applications, et créer de nouvelles applications au sein d'un projet. Le framework offre la possibilité d'utiliser ces applications pour ajouter des fonctionnalités segmentées à ces différentes applications en fonction des besoins spécifiques de chaque business.

Django propose une structure de fichiers/dossiers pour donner à chaque projet une structure claire ; : un fichier de configuration central, avec un dossier par application contenant chacun ainsi que des templates. Flask offre plus de libertés pour permettre à chaque développeur de s'organiser comme il le souhaite. Cela étant, il n'est pas toujours simple de déterminer à l'avance quelle sera la meilleure structure pour notre projet, en particulier lorsque celui-ci se complexifie mois après mois. Nombreux sont les développeurs qui finalement, choisissent d'imiter la structure proposée par Django même en utilisant Flask, pour s'imposer un cadre plus strict et s'éviter un contexte de développement trop désorganisé. Django reste cependant plus populaire, car celui-ci propose des fonctionnalités plus robustes et davantage adaptées au déploiement d'applications complexes. Pour autant, Flask reste avantageux pour tous les développeurs souhaitant pouvoir customiser la structure de leur projet pour répondre à des besoins précis. Flask est donc plus adapté aux débutants souhaitant acquérir de l'expérience et progresser en Python en ayant la liberté de contrôler les composants à leur guise. Django est davantage adapté aux projets de grande envergure, car sa structure permet d'avoir une organisation rigoureusement disciplinée, ce qui est indispensable pour pouvoir gérer au mieux la complexité de chaque projet au fil des mois.

3.4 HTML BOOTSTRAP

Bootstrap est un Framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation... Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées. Les navigateurs sont pleins de fantaisies et ont

des comportements très différents malgré leur lente convergence vers les standards. Les Framework sont cross-browser, c'est à dire que la présentation est similaire quel que soit le navigateur utilisé et d'une parfaite compatibilité.

- Les Framework CSS font gagner du temps de développement parce qu'ils nous proposent les fondations de la présentation.
- Les Framework CSS normalisent la présentation en proposant un ensemble homogène de styles.
- Les Framework CSS proposent en général une grille pour faciliter le positionnement des éléments.
- Les Framework CSS offrent souvent des éléments complémentaires : boutons esthétiques, barres de navigation, etc.
- La grande diffusion de nouveaux moyens de visualisation du web (Smartphones, tablettes...) impose désormais la prise en compte de tailles d'écran très variées ; les Framework CSS prennent généralement en compte cette contrainte.

3.5 JINJA

Dans notre première implémentation, nous avons créé le code HTML de notre site directement dans notre code Python. En pratique, le fait de mixer du code HTML avec du code Python peut rapidement devenir fastidieux. Pour éviter ce problème, il est préférable de sous-traiter la conception des pages à un moteur de template comme Jinja2 (le plus populaire en Python). A chacun son rôle, Flask pour le traitement des informations et Jinja2 pour la création des pages. Le moteur de Template Jinja2 permet, à partir de plusieurs informations passées en entrée, de construire un document ayant une structure bien spécifique. Concrètement, la construction d'un document HTML s'obtient à partir de deux fichiers.

3.6 Docker

Avec la technologie Docker, nous pouvons traiter les conteneurs comme des machines virtuelles très légères et modulaires. En outre, ces conteneurs nous offrent une grande flexibilité : nous pouvons les créer, déployer, copier et déplacer d'un environnement à un autre, ce qui vous permet d'optimiser vos applications pour le cloud. la raison pour utilisant cette technologie est a- la bibliothèque textgenrnn contienne des erreurs a cause de le non compatibilité entre kerase et tanserflow et pour les éviter chaque fois se changer les indépendance j' ai utiliser cette technologie pour garantir l'automatisation de lance le site web b- docker peut aussi nous aidais lord déploiement de site dans un serveur car il installe tout les indépendance dans un noyau de kernel dans notre car ce ubuntu

Conclusion

Durant Notre parcours à la filière informatique au Faculté de Science Semlalia on a pu mettre en pratique de nombreuses compétences acquises durant notre formation, que ce soit au niveau technique, conceptuel ou encore organisationnel, ce qui nous a permis de mettre en pratique le processus de conduite ce projet.

Le rapport contient trois chapitres, dans lesquels nous avons décrit les étapes principales de la réalisation du projet et qui nous a menés à un résultat satisfaisant, et simple à utiliser par n'importe quel utilisateur. Le premier chapitre est comme une introduction à ce projet, on a traité l'historique de l'embauche et le réseaux professionnel et comment ça évolué au temps et les déferente manier d'écriture des biographie etc. après on a mentionné les objectifs principaux de ce projet, ensuite on a situé quelques problèmes rencontrés au travail et quelque solution concernant ces problèmes-là

Pour le deuxième chapitre, à l'objectif pour présenter le projet de façon simple, d'abord on a commencé par web scraping en prennent l'automatisation en utilisant les scriptes python, en plus l'API est comment une application web peux communiquer avec un le serveur d'un autre notre site, ensuite on a introduit la NLP et comment va nous aide pour réaliser ce projet, après cela, nous avons abordé la notion de Machine learning « ML » pour l'apprentissage automatique et on a fini par les bibliothèques qui supportent ce dernier.

Concernant le troisième chapitre on a présenté le modèle textgenrnn avec qui nous travaillons pour builder notre apprentissage , et la façon pour traîner ce modèle et les résultats générés après. Ensuite le quatrième chapitre c'est pour le but de souligner toutes les techniques lesquelles on a utilisé dans ce projet soit dans la partie du code ou bien l'interface graphique Le troisième chapitre nous représentons l'application « Interface utilisateur » dont nous sommes fières en souhaitant l'évoluer encore plus, malgré tous les obstacles techniques que nous avons rencontrés, mais que nous avons pu surmonter pour en arriver jusque-là. Nous souhaitons, enfin que ce modeste travail soit en mesure de satisfaire ses lecteurs et récolter ainsi leurs félicitationss

Bibliographie

- [1] <https://stackoverflow.com/>
- [2] <https://www.w3schools.com/python/default.asp>
- [3] <https://www.coursera.org/specializations/natural-language-processing>
- [4] <https://www.digitalocean.com/community/>
- [5] <https://www.linkedin.com/developers/>
- [6] <https://www.analyticsvidhya.com/blog/2020/08/web-scraping-selenium-with-python/>
- [7] https://github.com/joeyism/linkedin_scraper
- [8] <https://github.com/minimaxir/textgenrnn>
- [9] <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- [10] <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- [11] <https://github.com/karpathy/char-rnn>
- [12] <https://www.codecademy.com/catalog/language/python>
- [13] <https://openclassrooms.com/fr/courses/5801891-initiez-vous-au-deep-learning>



كلية العلوم
السمالية - مراكش
FACULTÉ DES SCIENCES
MARRAKECH - SEMLALIA

UNIVERSITÉ CADI AYYAD FACULTÉ DES SCIENCES SEMLALIA

DÉPARTEMENT D'INFORMATIQUE

SCIENCES MATHÉMATIQUES ET INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ETUDES

Bio generator for Junior Data Scientists

RÉALISER PAR: CHEGRI MOHAMED

Membre de jury:

Dr. Zahir jihad

Dr. MOUSSANIF Hajar

Encadré par:

Dr. Zahir jihad

15 juillet 2021

Dédicace

A ma très chère maman Nezha,

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive. Que Dieu vous protège. A mes chers ami(e)s, En souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble. Veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.

Remerciement

Avant tout remerciement, louange à Dieu. Les mots ne peuvent pas exprimer notre gratitude pour **Dr. Jihad Zahir**, notre honorable encadrant qui nous à accompagner pendant tout le chemin vers l'accomplissement de ce projet de fin d'études, nous ne pouvions pas trouver un plus mieux encadrement, nous sommes très révéres d'avoir la chance de travailler avec un tel encadrant. Nous tenons aussi à adresser nos profondes reconnaissances et nos extrêmes gratitudes au jury de notre soutenance de projet de fin d'études, qui a accepté d'évaluer notre travail.

Également, nous remercions l'ensemble du corps enseignant à la Faculté des Sciences Semlalia Marrakech et particulièrement celui du Département Informatique qui nous a incités pendant notre formation et qui a posé dans nos mains le savoir-faire, qui nous a aidés à bien mener ce travail.

Ainsi, nous adressons nos remerciements à la direction de Faculté des Sciences Semlalia Marrakech.

Enfin, que tous ceux qui ont participé de près ou de loin à l'élaboration de ce travail, trouvent ici le témoignage de nos sincères remerciements.

Avant propos

La formation licence fondamentale mathématiques et informatiques (SMI) du département informatique de la faculté des sciences Semlalia Marrakech est complété par un stage obligatoire d'un mois, du 1er Mai au 30 Juin, afin d'adapter la formation académique aux exigences du marché de emploi, rendre plus pratique les connaissances théoriques acquises durant les années d'études, donner une vision du domaine du travail et préparer l'étudiant à l'insertion au domaine professionnel. J'ai effectué ce stage au sein de département informatique de la faculté des sciences Semlalia Marrakech, la mission était comme suit de crée un application web qui aide les jeunes Data Scientistes de mieux rédiger leur biographies sur Linkedin utilisant l'une des principales technologies de Machine Learning et d'intelligence artificielle.

Résumé

Dans ce rapport, nous allons proposer une application de génération des biographies pour les jeunes diplômés en se basant sur NLP "Natural Language Processing" afin de généraliser le processus pour les autres domaines professionnels. Notre travail a été divisé en deux grand partie, la première consiste a collecter du data depuis le site Internet LinkedIn, pour ceci on a utilisé la bibliothèque d'automatisation de python Selenium. Après la collecte des données, on faisait du data clining en utilisant des scripts de python, avant d'entraîner nos modèles pour générer des nouveaux texte en utilisant le paquet du python textgenrnn".

La deuxième partie est la représentation du résultat, pour cela on a réalisé une application web qui sera une interface graphique afin d'interagir avec l'utilisateur, pour ceci on a utilisé le micro fremework flask et le déployer sur un contenant de docker composé pour garantir une indépendances entre notre environnement du travail et le host de l'application.

Table des figures

1.1	Architecture d'aplication web.	1
2.1	Architecture de réseau neuronal.	4
2.2	Un exemple de RNN	5
2.3	Génération de nouveau biographie	6
2.4	Modèle conceptuel data base	6
2.5	Présentation de l'application web	7
2.6	Modèle conceptuel data base	8
2.7	Modèle data base	8
2.8	algorithme d'utilisation	8
2.9	biographie exemplaire pour un data scientifique.	9
2.10	Biographie graphie générée automatiquement pour un datas scientifique	9
2.11	Biographie construit pour l'utilisateur alex	10
2.12	Présentation du site	11

Liste d'abréviations

ANN	Artificial Neural Network
NLP	Natural language Processing
ML	Machine Learning
HTML	Hyper Text Markup Language
JSON	JavaScript Object Notation
API	Interface de programmation
XML	Extensible Markup Language
LSTM	long short-term memory
RNN	Recurrent neural network

Table des matières

Dédicace	ii
Remerciement	iii
Avant propos	iv
Résumé	v
Liste d'abréviations	vii
Introduction générale	ix
1 Étude théorique de projet	1
2 Implémentations de projet	3
2.1 Récupération du data	3
2.2 Nettoyage du carpus textuel	3
2.3 Entraîner du dataset	4
2.4 Génération du nouveau biographie	5
2.5 Modélisation de l'application web	6
2.5.1 Création d'une web application	7
2.5.2 Clasification des biographie	8
2.5.3 les biographies exemplaire	8
2.5.4 Les biographies généraix automatiquement	9
2.5.5 Biographie sur demande	10
2.6 Vue globale de l'application Web	10
3 Environnement de travail	12
3.1 Flask	12
3.2 DJANGO	12
3.3 FLASK OR DJANGO	13
3.4 HTML BOOTSTRAP	13
3.5 JINJA	14
3.6 Docker	14

Introduction générale

Après avoir fini les études, les carrières s'apparaissent, le premier problème qui affronte les nouveaux lauréats est la difficulté de trouver un stage pré-embauche ou un emploi. Pour ce faire ils ont besoin d'un réseau qui contient des professionnels de leurs domaines, ce réseau va les aider pour initialiser leurs carrières et avoir une vue globale du marché d'emploi est ses besoins.

LinkedIn par exemple est un réseau qui contient 660 millions de membres dans le monde en 2021, avec 69% des membres active, les utilisateurs déposent leur cv et une description de leurs parcours académique et professionnels, en USA par exemple 87% des recruteurs examineront la candidate en visitant leur profil LinkedIn. Google aussi utilise le réseau LinkedIn pour recruter des nouveaux employés.

Un bon profil LinkedIn commence par un résumé fantastique, qui résume la carrière et le parcours académique de l'utilisateur, ce dernier nécessite des techniques d'écritures pour une bonne rédaction et description du profil.

Avoir un outil qui suggère des exemples des biographies pour que les utilisateurs puissent s'inspirer de la méthode d'écriture des professionnels, et pour créer leurs propres biographies ou avoir un système intelligent qui le rédige pour eux et pourquoi pas on donne les informations personnelles de l'utilisateur, avoir une biographie générée automatiquement et plus spécifique.

Pour répondre à cette problématique, on va créer une application web qui génère d'une façon automatique les biographies pour les jeunes data scientistes en utilisant une technologie de Machine Learning qui s'appelle le deep learning. Pour ce fait, on va faire appel à la bibliothèque open source écrite en python Keras et la plateforme tensorflow afin de déployer le modèle de machine learning via le micro framework flask et docker-compose afin de créer une interface graphique qui permet aux jeunes data scientistes de générer une biographie par un clic.

Chapitre 1

Étude théorique de projet

Au début on doit définir notre environnement de travail ainsi que les bibliothèques et les algorithmes utilisés dans ce projet et l'architecture qu'on doit suivre pour créer notre application .

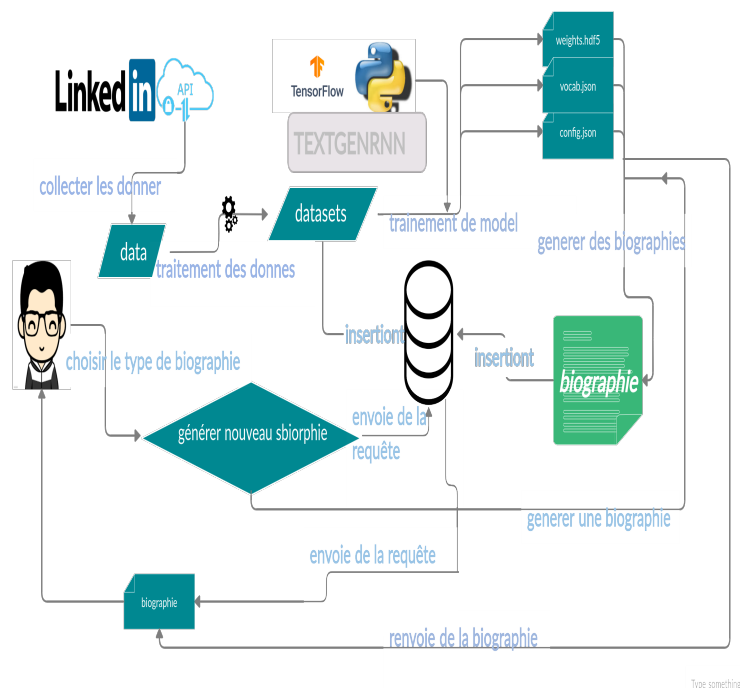


FIGURE 1.1 – Architecture d'application web.

Machine Learning : est une technologie d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Pour apprendre et se développer, les ordinateurs ont tout de même besoin des données à analyser et sur lesquelles s'entraîner. De fait.

Deep learning : Ou l'apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées jeu de données : (en anglais dataset ou data set)

selon wikipedia est un ensemble de valeurs (ou données) où chaque valeur est associée à une variable (ou attribut) et à une observation. Une variable décrit l'ensemble des valeurs décrivant le même attribut et une observation contient l'ensemble des valeurs décrivant les attributs d'une unité.

TensorFlow : TensorFlow est une bibliothèque de logiciels open source pour le calcul numérique utilisant des graphiques de flux de données. Les nœuds du graphique représentent des opérations mathématiques, tandis que les bords du graphique représentent les tableaux de données multidimensionnels (tenseurs) communiqués entre eux. L'architecture flexible vous permet de déployer le calcul sur un ou plusieurs CPU ou GPU sur un ordinateur de bureau, un serveur ou un appareil mobile avec une seule API.

Keras : Est une API Python basée sur un réseau neuronal de haut niveau qui s'exécute sur CPU ou GPU. Il prend en charge les réseaux convolutionnels et récurrents et peut s'exécuter au-dessus de TensorFlow, CNTK ou Theano. L'objectif principal est de permettre aux utilisateurs une expérimentation rapide. Il existe de nombreux exemples d'utilisation de Keras dans le tableau comparatif : classification, génération et résumé de texte, balisage, analyse, traduction automatique, reconnaissance vocale et autres.

RNN : Recurrent neural network, est une classe de réseaux de neurones artificiels où les connexions entre les nœuds forment un graphe orienté tout au long d'une séquence temporelle. Cela lui permet de présenter un comportement dynamique temporel. Les RNN peuvent utiliser leur état interne (mémoire) pour traiter des séquences d'entrées de longueur variable. Cela les rend applicables à des tâches telles que la reconnaissance d'écriture manuscrite ou la reconnaissance vocale .

LSTM Long short-term memory, La mémoire à long court terme est une qui possède une mémoire interne appelée cellule (ou cell). La cellule permet de maintenir un état aussi longtemps que nécessaire. Cette cellule consiste en une valeur numérique que le réseau peut piloter en fonction des situations, Il peut non seulement traiter des points de données uniques (tels que des images), mais également des séquences entières de données (telles que de la parole ou de la vidéo). Par exemple, LSTM est applicable à des tâches telles que la reconnaissance d'écriture manuscrite, la reconnaissance vocale.

Char-rnn : Une implémentation d'un multi-layer Recurrent Neural Network(RNN, LSTM) pour l'apprentissage à partir de modèles de langage au niveau des caractères. En d'autres termes, le modèle prend un fichier texte en entrée et forme un réseau neuronal récurrent qui apprend à prédire le caractère suivant dans une séquence. Le RNN peut ensuite être utilisé pour générer du texte caractère par caractère qui ressemblera aux données d'apprentissage d'origine.

NLP Exactement ce que cela ressemble, les techniques utilisées pour permettre aux ordinateurs de comprendre le langage humain naturel, plutôt que d'avoir à interagir avec les gens à travers des langages de programmation. Le traitement du langage naturel est nécessaire pour des tâches telles que la classification de documents Word ou la création d'un chatbot.

Implémentations de projet

La création de notre application est divisée en cinq phases, la première est la récupération du data en utilisant le web scraping, puis l'exploration du dataset en deuxième phase, puis, l'entraînement du modèle en utilisant le paquet de python textgenrnn pour entraîner notre modèle, la dernière phase, c'est création d'une interface web pour visualiser le résultat de notre training et aussi avoir l'accès aux biographies qui sont écrits par des professionnels.

2.1 Récupération du data

La première étape du traitement des données du texte est de récupérer le texte et le nettoyer afin de pouvoir l'utiliser ultérieurement dans l'apprentissage, Dans mon cas, j'ai scrapé la page Linkedin, mais malheureusement cette dernière possède une politique qui sécurise les informations des utilisateurs que ce soit publiques comme le nom, prénom, biographie, ou critique comme les numéros de téléphones et les adresses personnelles, mails, en bref Linkedin interdit tout sort de collecte d'informations, il détecte ce comportement en calculant le nombre de requêtes envoyés par chaque utilisateur, toute personne essaye de collecter les informations, son compte sera bloquée, ou supprimer après le premier avertissement. La solution est de minimiser les requêtes envoyer aux serveurs de linkedin, et de créer de nombreux faux compte "zombies" afin d'éviter la suppression de notre compte, la bibliothèque python selenium nous permet de scraper les pages de recherches pour chaque domaine, on extrait les urls des profils et on construit une table pour chaque profession. On a utilisé aussi ce qu'on appelle Application Programming Interfacesou API, et pour cela on doit avoir une application dans Linkendin, en créant cette dernière, on a pu extraire la moitié des biographies, et la raison pour laquelle on n'a pas passé toutes les biographies c'est que le serveur bloque l'adresse IP du machine qui passe un nombre déterminé de requêtes, pour éviter les attaques de cybersécurité, la sécurité comme DDOS attack, et la moitié de d'autre urls on le passe a la bibliothèque de Linkedin_scraper.

Le problème de cette partie c'est que nombreux profils ne possède pas de biographie ce qui fait que la plupart des requêtes n'a aucune utilité, ainsi que de nombreux résumés sont écrit en d'autre langue que du l'anglais,

2.2 Nettoyage du corpus textuel

La premier étape après la récupération de corpus et d'analyser et nettoyer en créant des scripts qui supprime toute biographie qui n'utilise pas l'encodage ASCII qui peuvent

mener à des erreurs faciles à éviter est chaque caractère doit être affichable.

En utilisant la bibliothèque textblob du python on va classer les biographies selon leur langues avant de supprimer tout biographie écrit en autre langue que la langue anglaise, a cause de modèle textgenrnn qui entraîne le dataset utilise le vocabulaire anglais et toute biographie non anglaise peut avoir des résultats non souhaitable.

2.3 Entraîner du dataset

De nombreux ressources de génération de texte utilise les réseaux de neurones, copient simplement une implémentations de char-rnn existante tout en modifiant l'ensemble de données d'entrée (dataset). C'est une approche, mais il existe une opportunité d'amélioration avec les outils modernes d'apprentissage en profondeur. Grâce à des frameworks comme TensorFlow et Keras. Textgenrnn est un package Python qui résume le processus de création et de formation de tels char-rnns à quelques lignes de code , avec de nombreuses améliorations d'architecture de modèle et de formation telles que les intégrations de caractères. Un avantage intéressant de textgenrnn est qu'il peut être facilement utilisé pour entraîner très rapidement des réseaux de neurones sur un GPU.

Textgenrnn traite automatiquement le texte saisi en séquences de caractères prêtes à entraîner le réseau. Après toutes les 2 époques (un passage complet à travers les données), le réseau générera un échantillon de texte à différentes températures, qui représentent la «créativité» du texte (c'est-à-dire qu'il permet au modèle de faire des prédictions de plus en plus optimales. Dans notre modelé on a utiliser Les options de configuration : 256 rnns, 256 passages complets des données , 16 rnn layers des séquences de modèles à entraîner dans une étape.

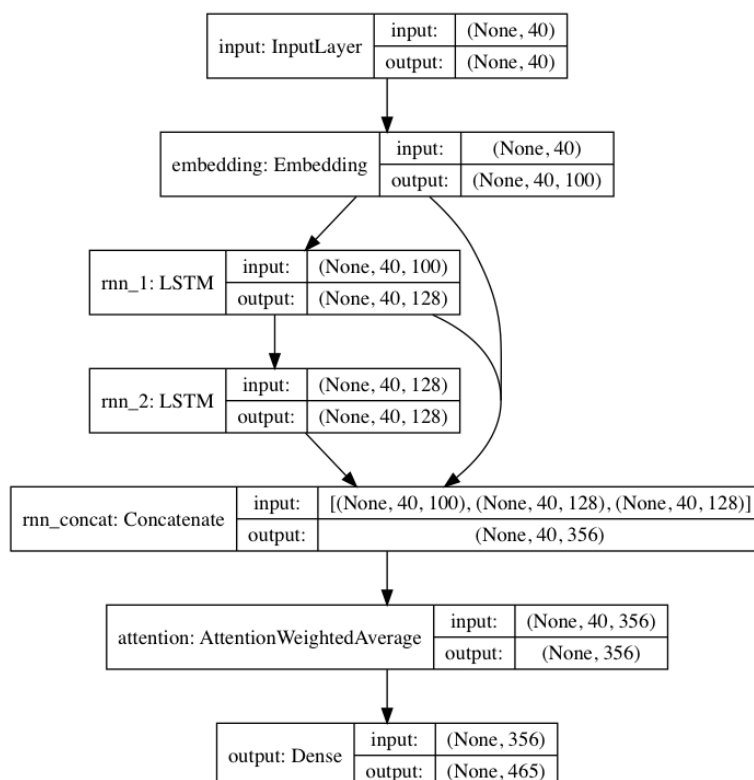


FIGURE 2.1 – Architecture de réseau neuronal.

Le modèle par défaut, textgenrnn prend une entrée de jusqu'à 40 caractères, convertit chaque caractère en un vecteur d'intégration de caractères 100-D et les alimente dans

une couche récurrente de 128 cellules de mémoire à long-court terme (LSTM). Ces sorties sont ensuite introduites dans un autre LSTM à 128 cellules. Les trois couches sont ensuite introduites dans une couche d'attention pour pondérer les caractéristiques temporelles les plus importantes et les moyenner ensemble (et puisque les intégrations + le 1er LSTM sont connectés par saut dans la couche d'attention, les mises à jour du modèle peuvent s'y propager plus facilement et éviter la disparition dégradés). Cette sortie est mappée sur des probabilités pour un maximum de 394 caractères différents qu'il s'agit du caractère suivant de la séquence, y compris les majuscules, les minuscules, la ponctuation et les emoji.

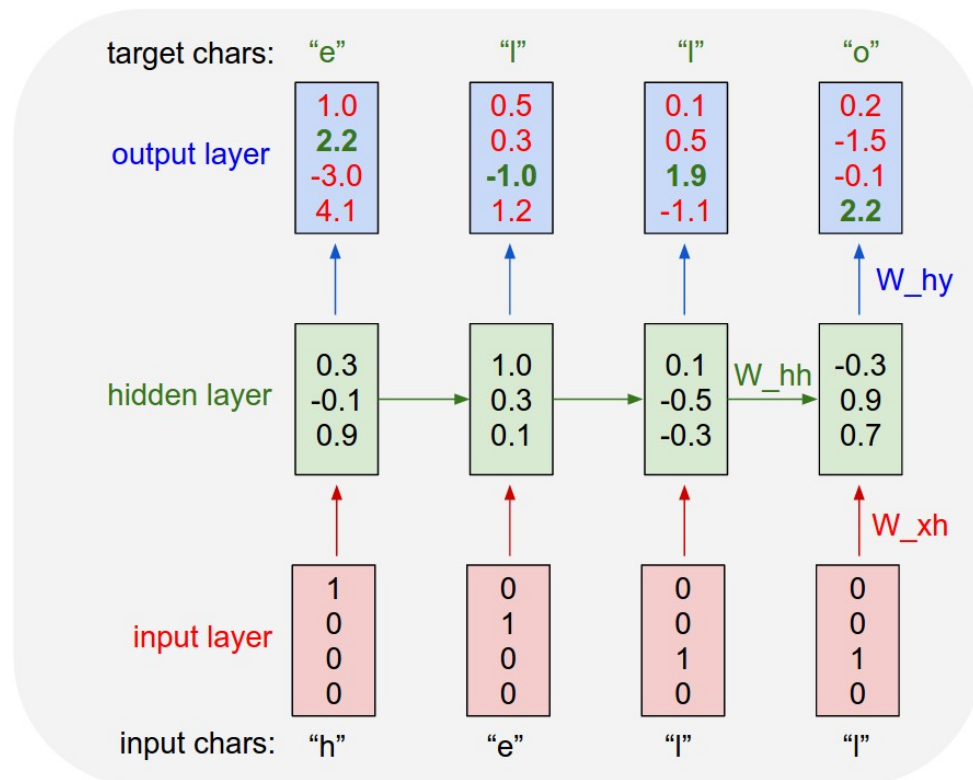


FIGURE 2.2 – Un exemple de RNN

2.4 Génération du nouveau biographie

Une fois la modèle terminée, on télé-chargera trois fichiers : un fichier contient le pois , vocabulaire, configuration du modèle qui sont tous nécessaires pour régénérer les nouveaux biographie, en générant de nouveaux biographie le programme prend plus de 5 secondes, se qui est une anomalie pour l'utilisateur qui peut pas attendre 6 secondes pour générer de nouveau biographie. Pour cette raison on génère plus de 1000 biographie et stocker ces derniers dans une base de données. Pour chaque domaine professionnel. En utilisant un script python qui va générer pour chaque domaine professionnelle 1000 biographie et le stocker dans une base de données afin d'utiliser ce dernier pour interagir avec l'utilisateur

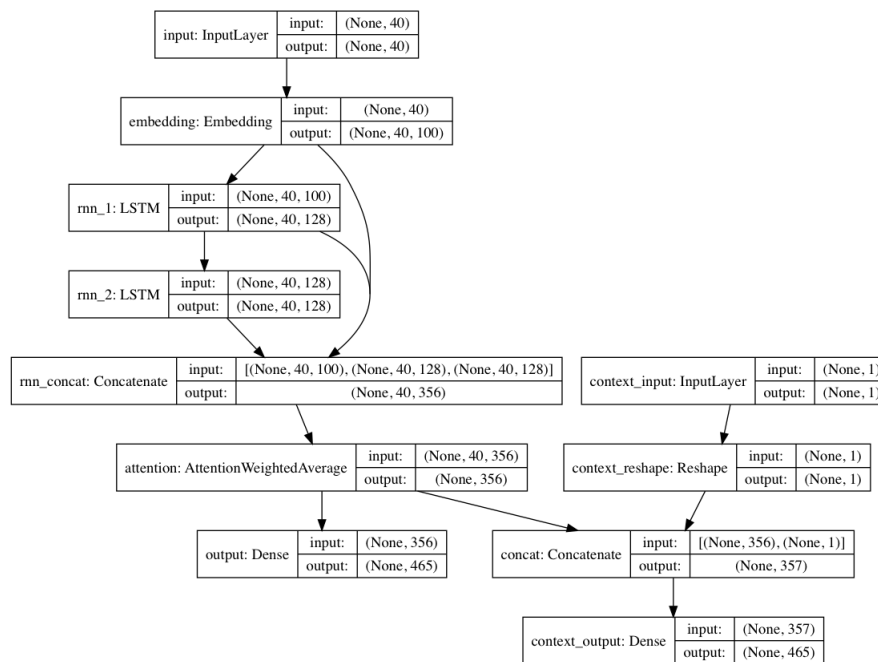


FIGURE 2.3 – Génération de nouveau biographie

2.5 Modélisation de l'application web

une page web pour premierment intragir avec l'utilisateur et de La création de l'appli-
cation web sera à la fois présenter a l'utilisateur des biographie exemplaire crée par des
experts et aussi des biographie générer par notre modelé qui été générer ultérieurement

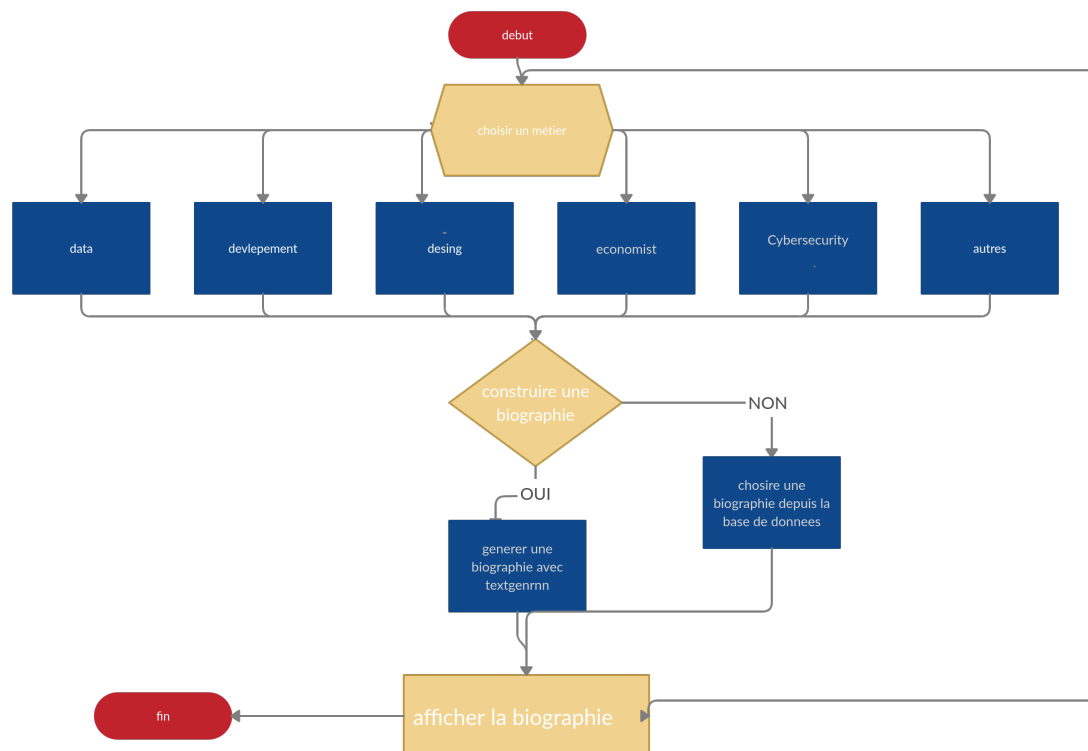


FIGURE 2.4 – Modèle conceptuel data base

2.5.1 Création d'une web application

premièrement on doivent définir notre application web et les raisons pour la quel a été crée , et les motives qui pous l'utilisateur a utiliser ainsi que l'importance de biographies dans le domaine professionnel, en déclarons les service que l'application servit.

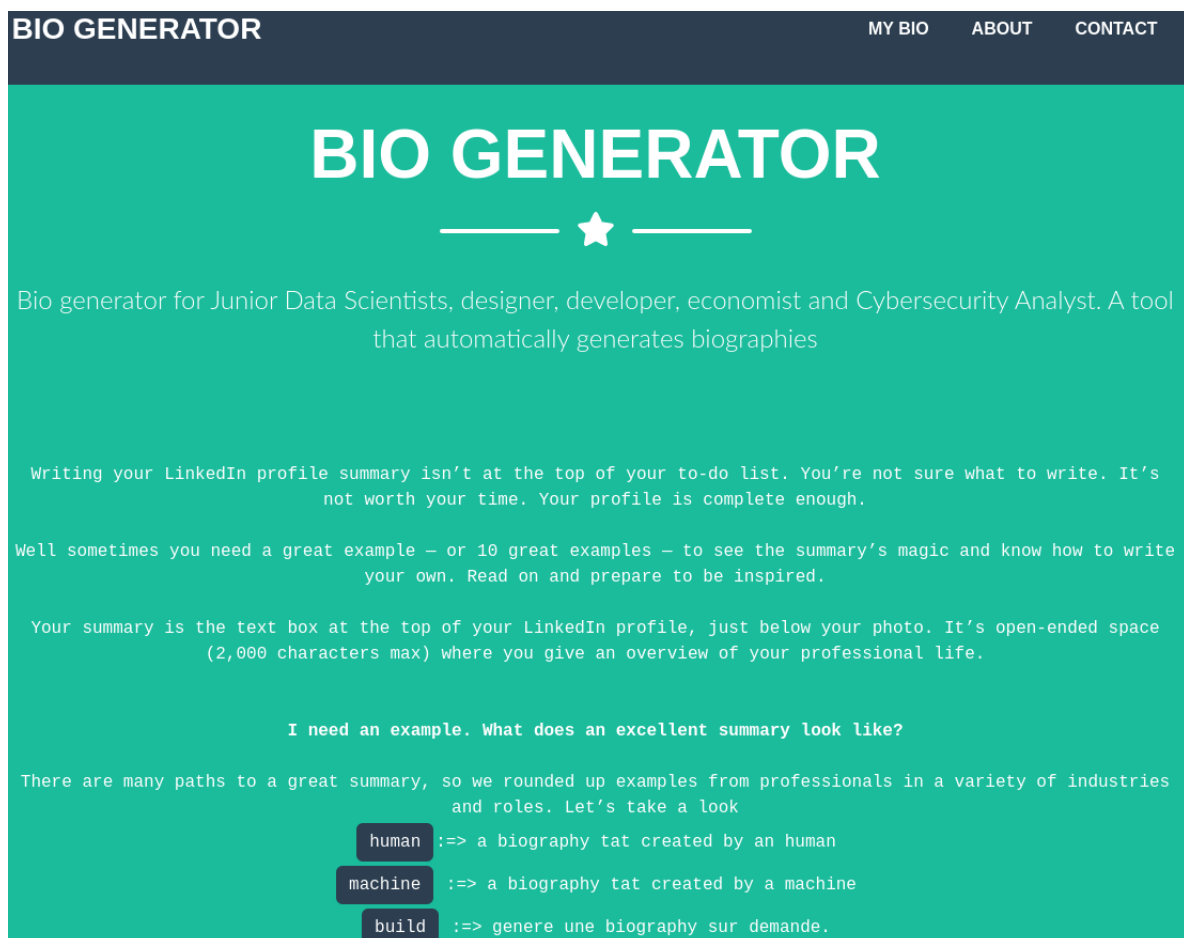


FIGURE 2.5 – Présentation de l'application web

Les biographie seront classer en premier classe par profession la même chose que du classification lors du collecte du data puis pour chaque profession on trouve trois sous classe

Human :les biographies exemplaire crée par des expert afin d'inspirer et remarquer la méthode de création des biographie machine : des biographie génère automatiquement en utilisant le modèle génère ultérieurement

"build" : création des biographie en utilisant les information des utilisateur en paramètre pour génère des biographie pour chaque utilisateur en ce basent sur le nome et le métier ou de quel domaine d'apprentissage s' il est étudiant, et l'entreprise dans la quel travaille ou étudier.

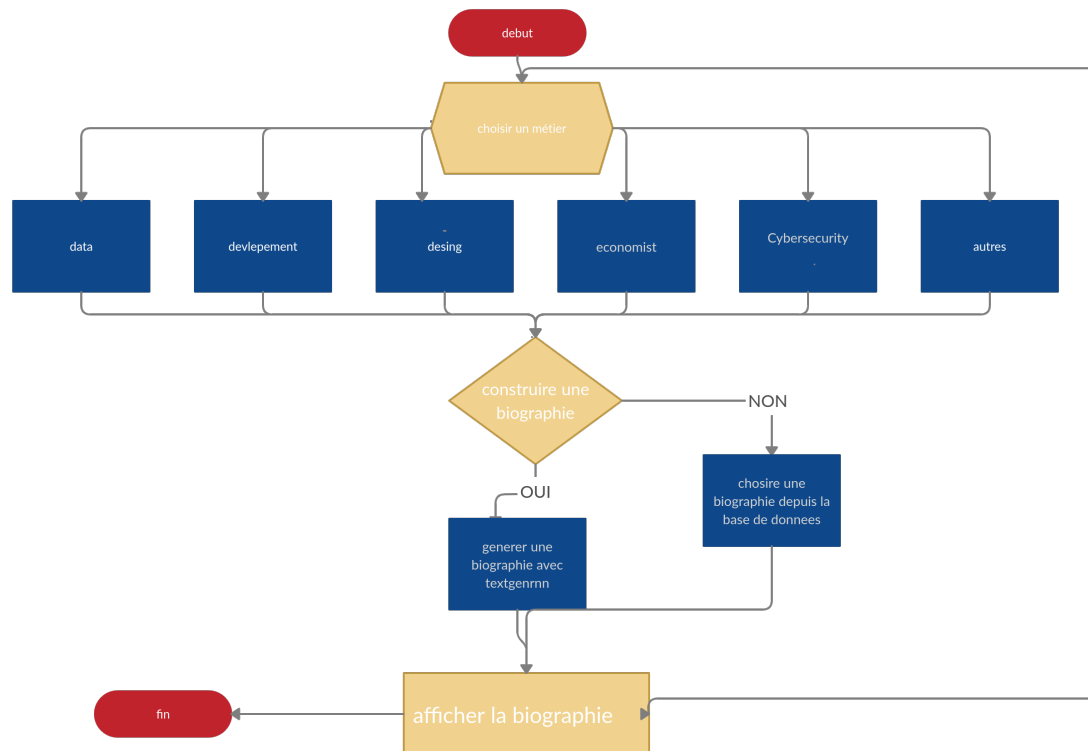


FIGURE 2.6 – Modèle conceptuel data base

2.5.2 Clasification des biographie

les biographies seront classer dans l'ordre dans la quel, la collections du data set a été fais, avec l'ajout d'une nouveau partie qui globalise tous les autre domaine cette dernier a été ajouter pour traîner tout les biographie collecter depuis linkedin sans prendre en considération les profession.

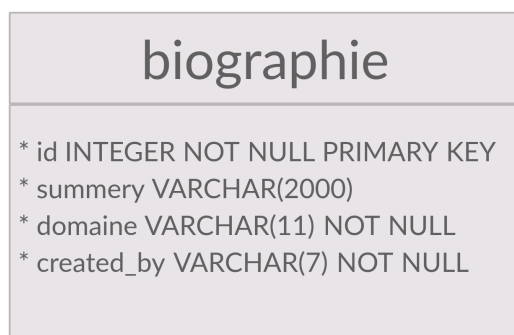


FIGURE 2.7 – Modèle data base

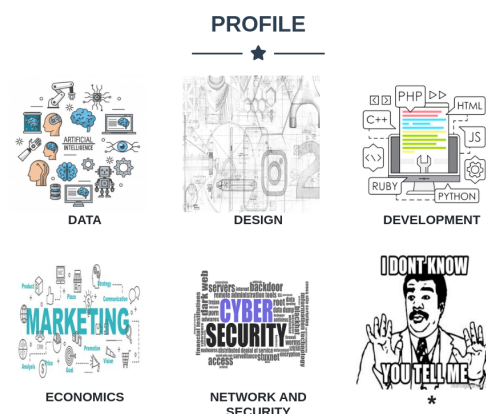


FIGURE 2.8 – algorithme d'utilisation

2.5.3 les biographies exemplaire

Dans cette rubrique on va donner la possibilité aux utilisateur la possibilité de lire les biographie collecter depuis linkedin, et avoir une idée comment les experts écrit leur

biographie et aussi avoir une vue sur d'autre domaine profériions, peut être qui il va trouver un profil qui va intersectée avec sa propre carier.

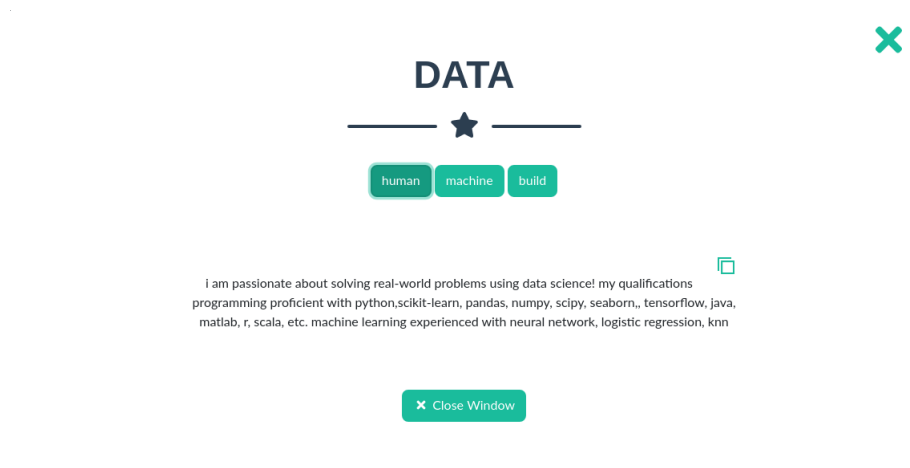


FIGURE 2.9 – biographie exemplaire pour un data scientifique.

2.5.4 Les biographies généraies automatiquement

Dans cette parte l'utilisateur peut choisir des biographies qui sont déjà génère par la machine et stocker dans une basse de donner pour l'utilisateur l'avoir immédiatement par-ce que la génération instantané des biographie prend entre 4 et 12 seconde qui est un très grand ce qui rendre difficile de choisie une bon biographie.

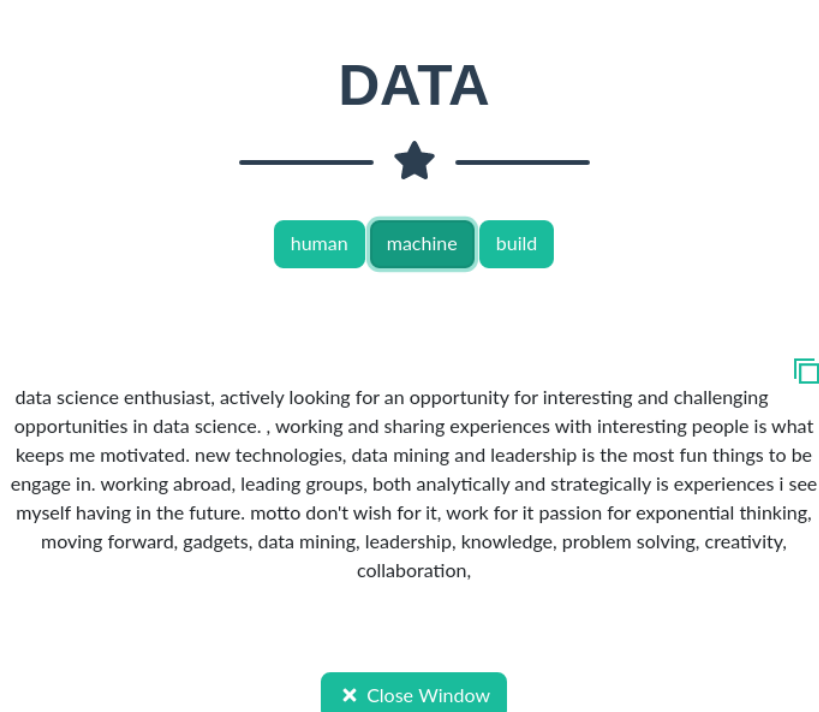


FIGURE 2.10 – Biographie graphie générée automatiquement pour un datas scientifique

2.5.5 Biographie sur demande

la section le plus important et qu'on donne a l'utilisateur de la possibilité de génère une biographie qui comporte son propre nom ainsi sa profession son lieu de travaille aux d'étude.

The screenshot shows a web application window titled "DATA" with a star icon. Below the title are three buttons: "human", "machine", and "build". The form contains the following fields:

- Full_name:** Alex Drysdale
- Job:** Junior Web Developer
- compañie:** Oswald Technologies
- generate** button

Below the form, a generated biography is displayed:

My name is Alex Drysdale I m currently working as Junior Web Developer in Oswald Technologies front everycing it prototyping. strong engineering professional with a bachelor of technology - b.tech focused on computer science and engineering from incunefl tasks for our clients , strategic and partnering and implementing machine learning and deep learning problems at leading entrepreneurship self-specialties entrepreneurship engagement , data analyst ,as an applying novel audio, security directors, bottomland, healthcare and innovation. my motto ppsics ,almm, data scientist with a passion for continuous learning and other data science problems. ventured assistant how to determine the manual results to innov 120mm and intensioned disciplines to solve inplusumental chargesbern projects ,processing of nlp, clustering. i conduct roles and numbers under the seaso emerging if it see my application developers in compensation. he is the new time of my design through deliverables,

At the bottom of the window is a "Close Window" button.

FIGURE 2.11 – Biographie construit pour l'utilisateur alex

2.6 Vue globale de l'application Web

Après avoir termine la dernière partie du backend, on passe a la partie de devops. Cette partie sera consacrer au containeriser l'application web pour assure l'indépendance entre les différentes facteurs et aussi de faire une l'application portable afin d' tout prochaine installation quelque soit la machine host et leur système d'exploitation serveur il suffit d'avoir un le programme docker seulement, cette solution a été utilise pour plusieurs raison, la première vu les erreurs de la bibliothèque textgenrnn qui génère des erreurs qui doivent être parser pour assurer le bon lancement du programme, les erreurs viennent a la fois de la compatibilité entre kerase et tensorflow et entre le programme lui mémé et le matériel. La dernière raison c'est d'installer nginx et le configurer nos propre nom domaine en local.

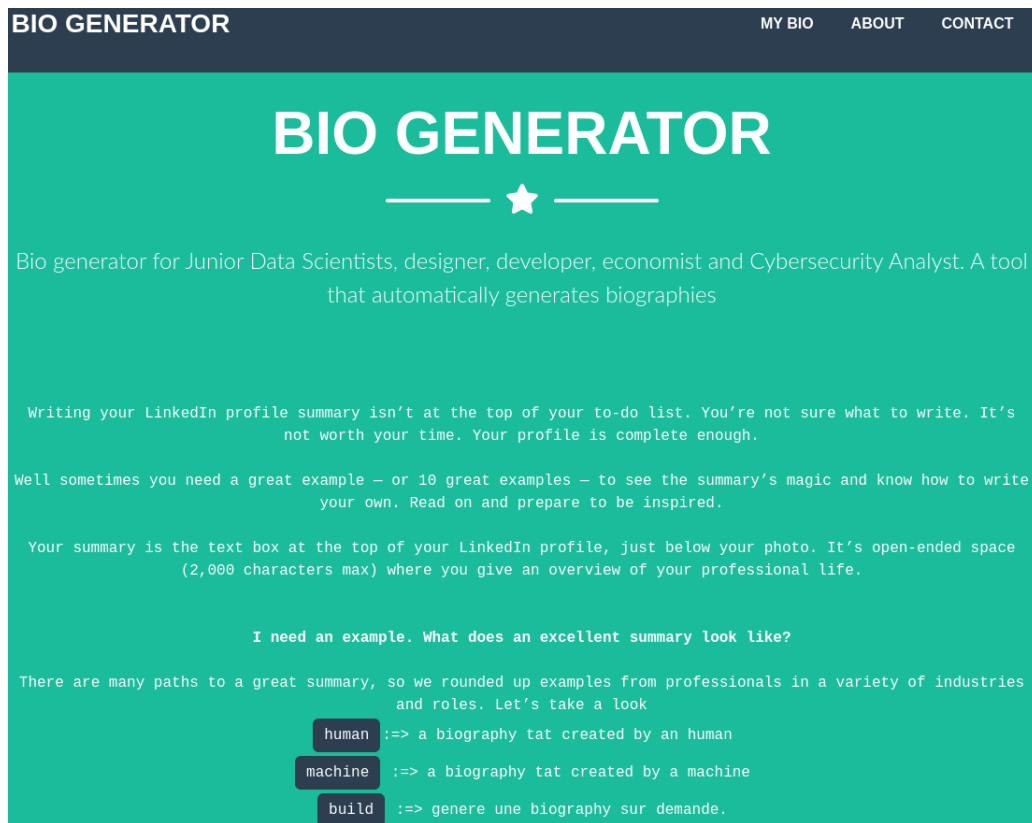


FIGURE 2.12 – Présentation du site

Environnement de travail

Colaboratory, souvent raccourci en "Colab", nous permet d'écrire et d'exécuter du code Python dans notre navigateur. Il ne nécessite aucune configuration requise plus l'accès gratuit aux GPU mais avec des des limite de taux d'apprentissage.

Vscode : l'éditeur de texte le plus utiliser dan le domaine de coding car il s'adapte avec tout les environnement de travail, web, mobile, réseau. Et aussi possède d'une très grand bibliothèque des extention qui facilite la tache aux programmeur ainsi une paquet intègre de github qui facilite la gestion d'historique de code

3.1 Flask

Flask est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de La programmation Python, associé à un système de templates. Il est distribué sous licence BSD Flask a été créé initialement par Armin Ronacher comme étant un poisson d'avril, Flask se base sur deux modules werkzeug et jinja2 pour proposer plusieurs des fonctionnalités suivantes : Serveur de développement⁶ et debugger Simplifie l'écriture de tests unitaires⁷ Moteur de template pour le rendu HTML⁸ Supporte les cookies sécurisés (session) etc.

3.2 DJANGO

Django est un Framework Python full-stack et développé, ce qui facilite à ses utilisateurs certaines taches communes de développement, comme l'authentification, le routage d'URL, ou encore la migration des schémas de base de données. Django accélère également le développement en fournissant des Template, un système ORM, et un outil de bootstrapping. Composé de trois parties distinctes : 1. Un langage de gabarits flexible qui permet de générer du HTML, XML ou tout autre format texte ; 2. Un contrôleur fourni sous la forme d'un « remapping » d'URL à base d'expressions rationnelles ; 3. Une API d'accès aux données est automatiquement générée par le cadre compatible CRUD. Inutile d'écrire des requêtes SQL associées à des formulaires, elles sont générées automatiquement par l'ORM.

3.3 FLASK OR DJANGO

Avant d'insister sur leurs différences, soulignons tout de même que Flask et Django restent tous les deux des Framework utilisables pour Python et sont dans l'ensemble assez similaires. Pour les deux, on retrouve la logique de la structure MVC. Il est donc assez simple de passer d'un Framework à l'autre sans perdre ses repères. Flask, quant à lui, est un Framework simple, léger, minimaliste. Bien que Flask offre moins de fonctionnalités que Django, celui-ci facilite le développement et garantit une meilleure souplesse. Django facilite la prise en main des tâches classiques d'administration grâce à un Framework admin prêt à l'emploi. Il génère le module admin automatiquement en fonction des modèles du projet. Les développeurs peuvent même customiser l'interface admin afin que celles-ci s'adaptent à des exigences business spécifiques. De ce point de vue, Django offre donc plus de possibilités que Flask, qui ne propose pas ces options. Flask est développé sur la base du template Jinja2. Ce Template intègre un environnement d'exécution Sand Box et de nombreux modèles permettant aux développeurs de gagner du temps sur le développement d'applications web dynamiques. Django propose un système de templates entièrement intégré permettant aux développeurs de définir un modèle et de customiser l'interface utilisateur en écrivant de nouveaux templates dans le langage Django (DTL). Flask n'intègre pas d'outil de Bootstrapping, contrairement à Django, qui propose l'outil `django-admin`. Ce dernier permet de commencer à développer des applications sans aucun input externe. Les développeurs utilisant Django peuvent diviser un seul et même projet en différentes applications, et créer de nouvelles applications au sein d'un projet. Le framework offre la possibilité d'utiliser ces applications pour ajouter des fonctionnalités segmentées à ces différentes applications en fonction des besoins spécifiques de chaque business.

Django propose une structure de fichiers/dossiers pour donner à chaque projet une structure claire ; : un fichier de configuration central, avec un dossier par application contenant chacun ainsi que des templates. Flask offre plus de libertés pour permettre à chaque développeur de s'organiser comme il le souhaite. Cela étant, il n'est pas toujours simple de déterminer à l'avance quelle sera la meilleure structure pour notre projet, en particulier lorsque celui-ci se complexifie mois après mois. Nombreux sont les développeurs qui finalement, choisissent d'imiter la structure proposée par Django même en utilisant Flask, pour s'imposer un cadre plus strict et s'éviter un contexte de développement trop désorganisé. Django reste cependant plus populaire, car celui-ci propose des fonctionnalités plus robustes et davantage adaptées au déploiement d'applications complexes. Pour autant, Flask reste avantageux pour tous les développeurs souhaitant pouvoir customiser la structure de leur projet pour répondre à des besoins précis. Flask est donc plus adapté aux débutants souhaitant acquérir de l'expérience et progresser en Python en ayant la liberté de contrôler les composants à leur guise. Django est davantage adapté aux projets de grande envergure, car sa structure permet d'avoir une organisation rigoureusement disciplinée, ce qui est indispensable pour pouvoir gérer au mieux la complexité de chaque projet au fil des mois.

3.4 HTML BOOTSTRAP

Bootstrap est un Framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation... Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées. Les navigateurs sont pleins de fantaisies et ont

des comportements très différents malgré leur lente convergence vers les standards. Les Framework sont cross-browser, c'est à dire que la présentation est similaire quel que soit le navigateur utilisé et d'une parfaite compatibilité.

- Les Framework CSS font gagner du temps de développement parce qu'ils nous proposent les fondations de la présentation.
- Les Framework CSS normalisent la présentation en proposant un ensemble homogène de styles.
- Les Framework CSS proposent en général une grille pour faciliter le positionnement des éléments.
- Les Framework CSS offrent souvent des éléments complémentaires : boutons esthétiques, barres de navigation, etc.
- La grande diffusion de nouveaux moyens de visualisation du web (Smartphones, tablettes...) impose désormais la prise en compte de tailles d'écran très variées ; les Framework CSS prennent généralement en compte cette contrainte.

3.5 JINJA

Dans notre première implémentation, nous avons créé le code HTML de notre site directement dans notre code Python. En pratique, le fait de mixer du code HTML avec du code Python peut rapidement devenir fastidieux. Pour éviter ce problème, il est préférable de sous-traiter la conception des pages à un moteur de template comme Jinja2 (le plus populaire en Python). A chacun son rôle, Flask pour le traitement des informations et Jinja2 pour la création des pages. Le moteur de Template Jinja2 permet, à partir de plusieurs informations passées en entrée, de construire un document ayant une structure bien spécifique. Concrètement, la construction d'un document HTML s'obtient à partir de deux fichiers.

3.6 Docker

Avec la technologie Docker, nous pouvons traiter les conteneurs comme des machines virtuelles très légères et modulaires. En outre, ces conteneurs nous offrent une grande flexibilité : nous pouvons les créer, déployer, copier et déplacer d'un environnement à un autre, ce qui vous permet d'optimiser vos applications pour le cloud. la raison pour utilisant cette technologie est a- la bibliothèque textgenrnn contienne des erreurs a cause de le non compatibilité entre kerase et tanserflow et pour les éviter chaque fois se changer les indépendance j' ai utiliser cette technologie pour garantir l'automatisation de lance le site web b- docker peut aussi nous aidais lord déploiement de site dans un serveur car il installe tout les indépendance dans un noyau de kernel dans notre car ce ubuntu

Conclusion

Durant Notre parcours à la filière informatique au Faculté de Science Semlalia on a pu mettre en pratique de nombreuses compétences acquises durant notre formation, que ce soit au niveau technique, conceptuel ou encore organisationnel, ce qui nous a permis de mettre en pratique le processus de conduite ce projet.

Le rapport contient trois chapitres, dans lesquels nous avons décrit les étapes principales de la réalisation du projet et qui nous a menés à un résultat satisfaisant, et simple à utiliser par n'importe quel utilisateur. Le premier chapitre est comme une introduction à ce projet, on a traité l'historique de l'embauche et le réseaux professionnel et comment ça évolué au temps et les déferente manier d'écriture des biographie etc. après on a mentionné les objectifs principaux de ce projet, ensuite on a situé quelques problèmes rencontrés au travail et quelque solution concernant ces problèmes-là

Pour le deuxième chapitre, à l'objectif pour présenter le projet de façon simple, d'abord on a commencé par web scraping en prennent l'automatisation en utilisant les scriptes python, en plus l'API est comment une application web peux communiquer avec un le serveur d'un autre notre site, ensuite on a introduit la NLP et comment va nous aide pour réaliser ce projet, après cela, nous avons abordé la notion de Machine learning « ML » pour l'apprentissage automatique et on a fini par les bibliothèques qui supportent ce dernier.

Concernant le troisième chapitre on a présenté le modèle textgenrnn avec qui nous travaillons pour builder notre apprentissage , et la façon pour traîner ce modèle et les résultats générés après. Ensuite le quatrième chapitre c'est pour le but de souligner toutes les techniques lesquelles on a utilisé dans ce projet soit dans la partie du code ou bien l'interface graphique Le troisième chapitre nous représentons l'application « Interface utilisateur » dont nous sommes fières en souhaitant l'évoluer encore plus, malgré tous les obstacles techniques que nous avons rencontrés, mais que nous avons pu surmonter pour en arriver jusque-là. Nous souhaitons, enfin que ce modeste travail soit en mesure de satisfaire ses lecteurs et récolter ainsi leurs félicitationss

Bibliographie

- [1] <https://stackoverflow.com/>
- [2] <https://www.w3schools.com/python/default.asp>
- [3] <https://www.coursera.org/specializations/natural-language-processing>
- [4] <https://www.digitalocean.com/community/>
- [5] <https://www.linkedin.com/developers/>
- [6] <https://www.analyticsvidhya.com/blog/2020/08/web-scraping-selenium-with-python/>
- [7] https://github.com/joeyism/linkedin_scraper
- [8] <https://github.com/minimaxir/textgenrnn>
- [9] <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- [10] <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- [11] <https://github.com/karpathy/char-rnn>
- [12] <https://www.codecademy.com/catalog/language/python>
- [13] <https://openclassrooms.com/fr/courses/5801891-initiez-vous-au-deep-learning>

