ENSIIE

et

L'équipe ILES du LISN

Stage Ingénieur 1A : Prototypage de Système de Dialogue et de Question-Réponse basé sur le Deep Learning

Rapport de stage :

Arthur Babin

Encadré par :
Christophe Servan

13 juin - 26 août 2022

Table des matières

1 Préambule

1.1 Synthèse

L'objectif de ce stage est d'élaborer une architecture client-serveur pour mettre en place un Système de Question-Réponse (SQR) dans le cadre d'un partenariat entre le Laboratoire Interdisciplinaire des Sciences du Numérique (LISN) et la Direction des Bibliothèques, de l'Information et de la Science Ouverte (DiBISO).

En d'autres termes il s'agit de déployer un agent conversationnel capable de répondre intelligemment aux questions que l'on pose généralement à la DiBISO :

— Quand ouvre la bu d'orsay ?
— Quel est le prix d'une impression en noir et blanc ?
— Je suis étudiant à l'université Paris Saclay, combien de livres puis-je emprunter ?
— ...

1.2 Remerciements

Mes remerciements vont tout d'abord à Christophe Servan qui m'a suivi et guidé et dont les conseils m'ont beaucoup apporté durant ce stage. Qu'il soit aussi remercié pour avoir su m'inspirer un très vif intérêt pour ce domaine.

Je tiens également à remercier Oralie Cattan, Sahar Ghannay et Sophie Rosset pour m'avoir fait par de leur expertise avec pédagogie et Anne-Laure Ligozat pour m'avoir orienté vers ce stage.

De façon plus générale, je souhaiterais dire merci à tous les membres du LISN que j'ai eu la chance de rencontrer durant ce stage pour leur accueil et leur bienveillance.

Merci enfin à mes réviseurs et en particulier à Clémence pour son soutien précieux.

2 Introduction

2.1 Objectif et sujet du stage

Le stage consistait à réaliser un prototype grâce aux outils du Traitement Automatique des Langues (TAL) afin de faciliter au mieux le travail de l'équipe Information Langue Écrite et Signée (ILES) du LISN dans le cadre de ce partenariat avec la DiBISO.

Je me suis donc familiarisé avec les techniques utilisées pour ce type de système et avec les technologies d'implémentation qui gravitent autour afin de produire une architecture cohérente et fonctionnelle capable de prendre en entrée une question et de retourner une réponse.

À la fin, on souhaite que le système soit en mesure de répondre aux questions en tout genre de l'utilisateur final concernant les bibliothèques sous la tutelle de la DiBISO pour y améliorer l'efficacité de la recherche d'informations.

Une fois le système déployé, il serait également intéressant de pouvoir ajouter sans friction du contenu dans lequel le système viendra extraire les réponses aux questions des utilisateurs.

2.2 Démarche et environnement

Au sein de l'équipe ILES, j'ai pu dans un premier temps me former grâce aux ressources que l'on m'a fournies et ainsi mieux comprendre les enjeux de ce stage.

Sous les directives de Christophe Servan et sur la machine virtuelle (VM) mise en place par le service informatique j'ai ensuite deployé une architecture simple que j'ai améliorée par étape en y ajoutant au fur et à mesure des composantes. Il s'agissait surtout d'utiliser la conteneurisation informatique afin que cette architecture puisse être déployée sur une autre machine facilement et sans soucis de compatibilité.

Ce rapport dressera donc un bref état de l'art des approches du TAL basées sur le Deep Learning ou Apprentissage Profond en français (DL). En effet ces approches sont beaucoup utilisées dans les SQR qui effectuent la tâche de réponse automatique à une question posée en langage naturel car elles ont pour but d'imiter l'intelligence humaine et la façon dont les neurones de notre cerveau s'organisent pour acquérir certaines compétences.

Ensuite ce rapport présentera la construction de l'architecture ainsi que les modules complémentaires développés pour automatiser la maintenance et l'amélioration d'un tel système.

Échos du Cœur

Poèmes en partage

Collection de poésie française

printemps 2025

Semaine 1

Choisi par Jeanne

Au bord du quai

Emile Verhaeren, Les visages de la vie

Et qu'importe d'où sont venus ceux qui s'en vont,
S'ils entendent toujours un cri profond
Au carrefour des doutes !

Mon corps est lourd, mon corps est las,
Je veux rester, je ne peux pas ;
L'âpre univers est un tissu de routes
Tramé de vent et de lumière ;

Mieux vaut partir, sans aboutir,
Que de s'asseoir, même vainqueur, le soir,
Devant son oeuvre coutumière,
Avec, en son coeur morne, une vie
Qui cesse de bondir au-delà de la vie.

Choisi par Luc

PARIS

Louis Aragon, 1944

Où fait-il bon même au coeur de l'orage
Où fait-il clair même au coeur de la nuit
L'air est alcool et le malheur courage
Carreaux cassés l'espoir encore y luit
Et les chansons montent des murs détruits

Jamais éteint renaissant de la braise
Perpétuel brûlot de la patrie
Du Point-du-Jour jusqu'au Père-Lachaise
Ce doux rosier au mois d'août refleuri
Gens de partout c'est le sang de Paris

Rien n'a l'éclat de Paris dans la poudre
Rien n'est si pur que son front d'insurgé
Rien n'est ni fort ni le feu ni la foudre
Que mon Paris défiant les dangers
Rien n'est si beau que ce Paris que j'ai

Rien ne m'a fait jamais battre le coeur
Rien ne m'a fait ainsi rire et pleurer
Comme ce cri de mon peuple vainqueur
Rien n'est si grand qu'un linceul déchiré
Paris Paris soi-même libéré

Semaine 2

Bonus choisis par Jeanne et Luc

El Desdichado

Gérard de Nerval

Je suis le Ténébreux, – le Veuf, – l'Inconsolé,
Le prince d'Aquitaine à la tour abolie :
Ma seule étoile est morte, – et mon luth constellé
Porte le Soleil noir de la Mélancolie.

Dans la nuit du tombeau, toi qui m'as consolé,
Rends-moi le Pausilippe et la mer d'Italie,
La fleur qui plaisait tant à mon cœur désolé,
Et la treille où le pampre à la rose s'allie.

Suis-je Amour ou Phébus ?... Lusignan ou Biron ?
Mon front est rouge encor du baiser de la reine ;
J'ai rêvé dans la grotte où nage la syrène...

Et j'ai deux fois vainqueur traversé l'Achéron :
Modulant tour à tour sur la lyre d'Orphée
Les soupirs de la sainte et les cris de la fée.

Le Pont Mirabeau

Guillaume Apollinaire, Alcools, 1913

Sous le pont Mirabeau coule la Seine
Et nos amours
Faut-il qu'il m'en souvienne
La joie venait toujours après la peine

Vienne la nuit sonne l'heure
Les jours s'en vont je demeure

Les mains dans les mains restons face à face
Tandis que sous
Le pont de nos bras passe
Des éternels regards l'onde si lasse

Vienne la nuit sonne l'heure

Le ciel est triste et beau comme un grand reposoir.

Le violon frémit comme un cœur qu'on afflige,
Un cœur tendre, qui hait le néant vaste et noir !
Le ciel est triste et beau comme un grand reposoir ;
Le soleil s'est noyé dans son sang qui se fige.

Un cœur tendre, qui hait le néant vaste et noir,
Du passé lumineux recueille tout vestige !
Le soleil s'est noyé dans son sang qui se fige ...
Ton souvenir en moi luit comme un ostensoir !

Semaine 3

Choisi par Jeanne

Dans Paris

Paul Éluard

Dans Paris, il y a une rue
Dans cette rue, il y a une maison
Dans cette maison, il y a un escalier
Dans cet escalier, il y a une chambre
Dans cette chambre, il y a une table
Sur cette table, il y a un tapis
Sur ce tapis, il y a une cage

Dans cette cage, il y a un nid
Dans ce nid, il y a un œuf
Dans cet œuf, il y a un oiseau.

L'oiseau renversa l'œuf
L'œuf renversa le nid
Le nid renversa la cage
La cage renversa le tapis
Le tapis renversa la table
La table renversa la chambre
La chambre renversa l'escalier
L'escalier renversa la maison
La maison renversa la rue
La rue renversa la ville de Paris.

Harmonie du soir

Charles Baudelaire, Les Fleurs du Mal

Voici venir les temps où vibrant sur sa tige
Chaque fleur s'évapore ainsi qu'un encensoir ;
Les sons et les parfums tournent dans l'air du soir ;
Valse mélancolique et langoureux vertige !

Chaque fleur s'évapore ainsi qu'un encensoir ;
Le violon frémit comme un cœur qu'on afflige ;
Valse mélancolique et langoureux vertige !

Les jours s'en vont je demeure

L'amour s'en va comme cette eau courante
L'amour s'en va
Comme la vie est lente
Et comme l'Espérance est violente

Vienne la nuit sonne l'heure
Les jours s'en vont je demeure

Passent les jours et passent les semaines
Ni temps passé
Ni les amours reviennent
Sous le pont Mirabeau coule la Seine

Vienne la nuit sonne l'heure
Les jours s'en vont je demeure

Choisi par Jeanne

Ariettes oubliées (III)

Paul Verlaine

Il pleure dans mon cœur
Comme il pleut sur la ville,
Quelle est cette langueur
Qui pénètre mon cœur ?

O bruit doux de la pluie
Par terre et sur les toits !
Pour un cœur qui s'ennuie
O le chant de la pluie !

Il pleure sans raison
Dans ce cœur qui s'ècœure.
Quoi ! nulle trahison ?
Ce deuil est sans raison.

C'est bien la pire peine
De ne savoir pourquoi,
Sans amour et sans haine,
Mon cœur a tant de peine !

Choisi par Luc

Tout noble cœur...

Anna de Noailles, Les Forces éternelles (1920)

Tout noble cœur souhaite et veut être constant,
Mais vous, bohémienne, ô folle Destinée,
Jouant d'un violon discordant et strident,
Vous traînez sur le temps vos dansantes nuées.

Quel que soit le pas ferme et droit de la Raison,
Le Sort vient sur sa route, et la gêne, et divague ;

Jamais un jour pareil dans la même saison,

Toujours le renflement ou le creux de la vague !

Et le désir humain, cherchant la fixité,
Et ne trouvant sa paix qu'aux choses éternelles,
N'aime enfin plus que vous, immenses jours d'été,
Qui nous donnez, avec votre clarté fidèle,

Et vos airs de bonté et de tranquillité,
Ce trésor d'infini, que l'âme sensuelle
N'a connu qu'en jetant des sanglots irrités,
Dans l'austère, incisive et brûlante querelle
Que s'infligent deux cœurs pendant la volupté...

3 Développement

3.1 Environnement

Ce stage s'inscrit au sein d'une mission qui fait le pont entre :
— les forces de recherche du LISN et plus particulièrement le savoir-faire de l'équipe ILES notamment spécialisée dans le TAL et qui a l'habitude de travailler sur ce type de systèmes
— et la mission d'information de la DiBISO qui s'occupe de coordonner le réseau des bibliothèques avec l'ensemble des services et centres de documentation de l'université Paris-Saclay.

Cependant je n'ai pas vraiment interagi avec le personnel de la DiBISO. J'ai surtout été orienté grâce aux réunions régulières avec Sahar Ghannay, Sophie Rosset et Christophe Servan et à leurs indications.

De plus, au sein du LISN a été mis en place une messagerie instantanée qui s'est révélée très pratique pour demander de l'aide, planifier les réunions et communiquer avec les personnes qui travaillent en distanciel ou depuis d'autres sites.

3.2 Problématique générale

La DiBISO aimerait pouvoir faire appel à une Interface de Programmation d'Application (API) pour mettre en place un service qui s'apparente aux agents conversationnels que l'on retrouve sur de nombreux sites. L'API joue donc le rôle d'intermédiaire entre l'utilisateur qui pose les questions et le système qui trouve les réponses à ces questions.

De plus la diversité et la complexité des questions auxquelles le SQR doit répondre empêche l'élaboration d'une liste exhaustive de questions-réponses. C'est pourquoi l'idée est d'utiliser une collection de documents (les pages web pertinentes du site `https://www.bibliotheques.universite-paris-saclay.fr`) comme une forme de base de connaissances pour le SQR dans laquelle il viendra extraire les réponses associées aux questions.

La DiBISO souhaite également avoir une certaine flexibilité dans l'ajout de nouvelles pages à la collection pour pouvoir élargir les connaissances du SQR et ainsi lui permettre d'être en mesure de répondre à de nouvelles questions.

Mes missions étaient donc de déployer un prototype basé sur le DL capable de répondre le mieux possible aux questions et de proposer une interface web associée à une API permettant de dialoguer avec le système afin de collecter les données des conversations qui serviront à son amélioration.

Ainsi pour réaliser cette demande deux problématiques principales se dégagent :

— Comment trouver automatiquement la réponse à une question dans une collection de documents ? Ce qu'on appelle plus communément problème de Question-Réponse dans le domaine Ouvert ou Open domain Question-Answering (OQA).
— Comment déployer un tel système au cœur d'une architecture client-serveur dans le cadre de ce partenariat ?

3.3 État de l'art et contexte

3.3.1 Représentation d'une séquence de mots (tokens)

Généralement une des premières étapes d'un système de TAL pour la tâche d'OQA est la représentation des séquences de mots (phrases, paragraphes etc) en vecteurs. Plusieurs méthodes sont possibles.

Un bag of words ou sac de mots en français (BOW)

Un BOW est un histogramme du nombre d'occurrences de chaque mot dans la séquence ou le document ce qui est très utile en recherche d'information mais cette représentation est trop grossière pour la tâche d'OQA.

Une méthode de pondération de l'anglais term frequency-inverse document frequency (TF-IDF)

TF-IDF modifie les scores du BOW de telle sorte à ce que les mots rares aient un score élevé et les mots communs aient un score faible. Elle est très utile pour mesurer l'importance d'un document au sein d'un corpus par rapport à un terme donné.

$$\text{tfidf}_{i,j} = \underbrace{\frac{n_{i,j}}{\sum_k n_{k,j}}}_{\text{tf}_{i,j}} \times \underbrace{\ln \frac{|D|}{|\{d_j : t_i \in d_j\}|}}_{\text{idf}_i}$$

où

$$\begin{cases} D & \text{le corpus soit l'ensemble des documents} \\ d_j \in D & \text{un document du corpus} \\ \text{tf}_{i,j} & \text{fréquence du terme } i \text{ dans le document } j \\ \text{idf}_i & \text{fréquence inverse de document du terme } i \text{ dans le corpus} \\ \text{n}_{i,j} & \text{nombre d'occurrences du terme } i \text{ dans le document } j \end{cases}$$

Un plongement de mots dans un espace vectoriel dense de nombres réels

Un plongement de mots répond à la nécessité d'obtenir une représentation mathématique du sens des mots en apprentissage automatique ou machine learning en anglais (ML).

En effet, les représentations discrètes liant par exemple chaque mot à un entier différent ne permettent pas de comparer deux mots entre eux. Les plongement de mots se basent donc sur l'hypothèse de Harris stipulant que des mots apparaissant dans des contextes similaires ont des significations apparentées afin d'obtenir de tels vecteurs pour chaque mot.

Ainsi les mots "vacances" et "repos" seront représentés par des vecteurs peu distants l'un par rapport à l'autre dans l'espace vectoriel qui aura été défini.

De plus la représentation de mots par vecteur dense nécessite un modèle qui a préalablement appris à représenter des mots en traitant de très larges corpus. Les modèles les plus utilisés pour générer ces vecteurs de mots sont GloVe, word2Vec et FastText ou encore les modèles Transformer qui peuvent également être utilisés pour la tâche de plongement de mots et qui seront abordés paragraphe 3.3.4.

Focus sur GloVe GloVe est une approche qui se découpe en deux étapes :

1. la construction d'une matrice X « count-based » des co-occurrences globales des mots en utilisant une fenêtre contextuelle glissante. Ainsi $X_{i,j}$ correspond au nombre de fois que le mot i apparaît dans le même contexte que le mot j. Par exemple sur la Figure 1 la matrice X en considérant une fenêtre contextuelle qui se déplace de phrase en phrase.

	Il	dort	Elle	se	baigne
Il	1	1	0	0	0
dort	1	1	0	0	0
Elle	0	0	1	1	1
se	0	0	1	1	1
baigne	0	0	1	1	1

Table 1 – Matrice des co-ocurrences pour "Il dort. Elle se baigne"

2. l'entraînement d'un modèle de régression par moindres carrés à partir de X pour apprendre à construire les représentations vectorielles $\vec{w}_i$ avec :

$$\begin{cases} f & \text{fonction de pondération} \\ b_i & \text{vecteur biais du mot } i \\ Loss = \sum_{i,j \in [\![1,V]\!]} f(X_{i,j})(\vec{w}_i^T \vec{w}_j + b_i + b_j - \log X_{i,j})^2 & \text{à minimiser} \end{cases}$$

3.3.2 Réseaux de neurones récurrents

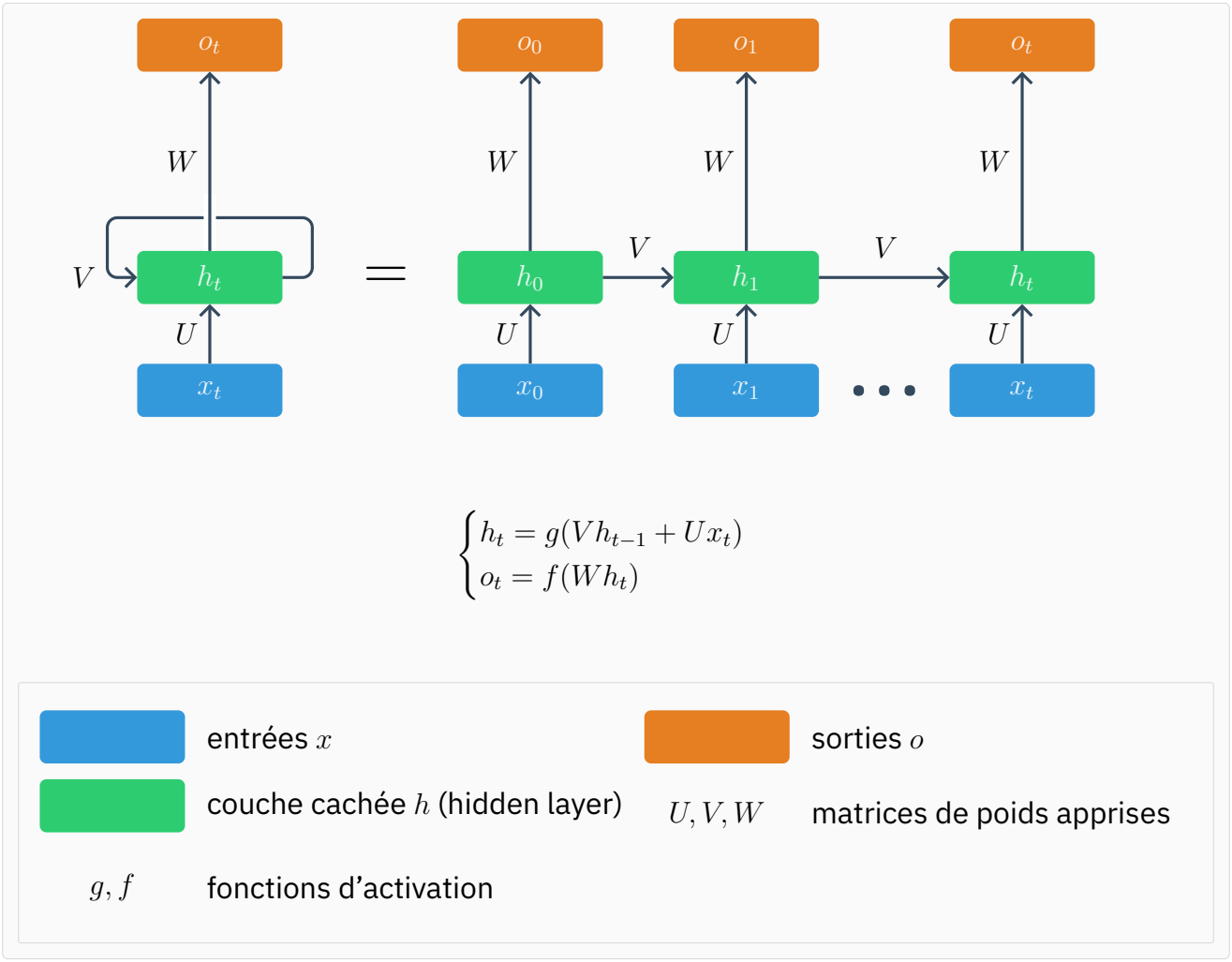


Figure 1 – Exemple du RNN d'Elman

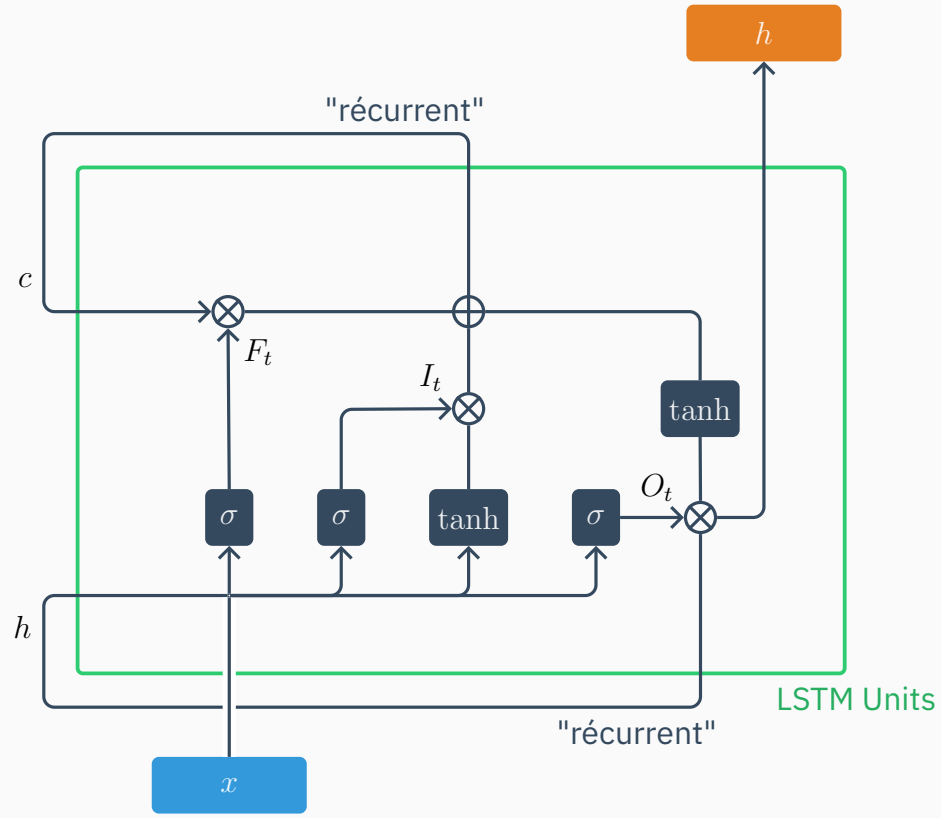
Cependant le BOW, l'approche TF-IDF ou encore GloVe ne prennent pas en compte l'ordre des mots dans une séquence. C'est là qu'intervient le réseau de neurones récurrents (RNN), particulièrement adapté aux données séquentielles.

Le RNN est une variante importante du réseau de neurones classique (à propagation avant) et est très utilisé dans le TAL. Dans la figure 1 le RNN traite chaque token d'une séquence $(x_0, ..., x_t)$ l'un après l'autre en utilisant également le résultat du token précédent en entrée. C'est donc une structure capable de prendre en compte le contexte d'un mot. En effet la sortie o_t dépend à la fois de l'entrée x_t et de la variable cachée h_{t-1}.

Ce modèle est tout d'abord entraîné avec des séquences d'entrée dont on connaît la sortie afin de régler les différents coefficients des matrices de poids : c'est la phase d'apprentissage. Une fois cette étape terminée on présente au modèle des séquences inconnues en récupérant la sortie : c'est la phase de prédiction. Le but étant d'entraîner le RNN de sorte à ce qu'il puisse ensuite généraliser son apprentissage sur un maximum de séquences inconnues.

Cependant le RNN ne va pas capturer les dépendances dans les deux sens et présuppose que le mot arrivant après n'a aucune incidence sur les mots précédents. Il est aussi limité car il ne prend pas bien en compte les dépendances sur le long terme à cause du problème de la disparition du gradient.

3.3.3 RNN à mémoire court-terme et long-terme (LSTM)



$$\begin{cases} F_t = \sigma(x_t * U_f + h_{t-1} * W_f) \\ \bar{C}_t = \tanh(x_t * U_c + h_{t-1} * W_c) \\ I_t = \sigma(x_t * U_i + h_{t-1} * W_i) \\ O_t = \sigma(x_t * U_o + h_{t-1} * W_o) \end{cases} \qquad \begin{cases} c_t = F_t * c_{t-1} + I_t * \bar{C}_t \\ h_t = O_t * \tanh(c_t) \end{cases}$$

entrées x — sorties h

couche cachée récursive

I la porte d'entrée — O la porte de sortie

c le contexte = mémoire — F la porte de remise à zéro

$\begin{cases} U_f, U_c, U_i, U_o \\ W_f, W_c, W_i, W_o \end{cases}$ les matrices de poids

Figure 2 – Schéma d'une couche d'un modèle LSTM (les matrices de poids ne sont pas représentées pour ne pas alourdir le schéma)

L'expérience montre que les RNNs ont du mal à garder une information de contexte pertinente pour le contexte lointain ($t' << t$) d'un événement x_t ce qui est un handicap pour

les phrases longues.

Le modèle des Long Short Term Memory (LSTM) répond ainsi au besoin d'une meilleure gestion du contexte et des dépendances lors du traitement d'une séquence. Nous avions vu que le RNN avait du mal à capturer les dépendances sur le long terme, c'est pourquoi l'intuition derrière les LSTM est de pouvoir :

— enlever l'information qui n'est plus nécessaire du contexte
— ajouter l'information qui est nécessaire pour une prise de décision ultérieure

Sur la Figure 2 on observe qu'en plus des informations passées d'un instant t à l'instant suivant $t+1$ via la variable cachée $h(t)$ (sert à la mémoire à court terme), ces cellules se transmettent également la variable $c(t)$ correspondant à une forme de mémoire long terme. Ce mécanisme est contrôlé par différentes portes qui permettent de maîtriser l'information qui est retenue :

— F_t la porte de remise à zéro (forget gate) qui permet au réseau de supprimer l'information du contexte qui n'est plus nécessaire
— I_t la porte d'entrée (input gate) pour calculer l'information à extraire du précédent état caché et de l'entrée courante
— O_t la porte de sortie (output gate) qui décide quelle information est requise par l'état caché courant

On voit également apparaître les Bi-LSTM qui connectent deux couches cachées dans des directions opposées afin d'avoir des informations sur le contexte des jetons qui suivent et de ceux qui précèdent à chaque étape.

Cependant les LSTM sont relativement lents lors de la phase d'apprentissage car ils traitent l'information séquentiellement (token par token) et ont encore des difficultés à traiter les longues séquences.

3.3.4 Les Transformers

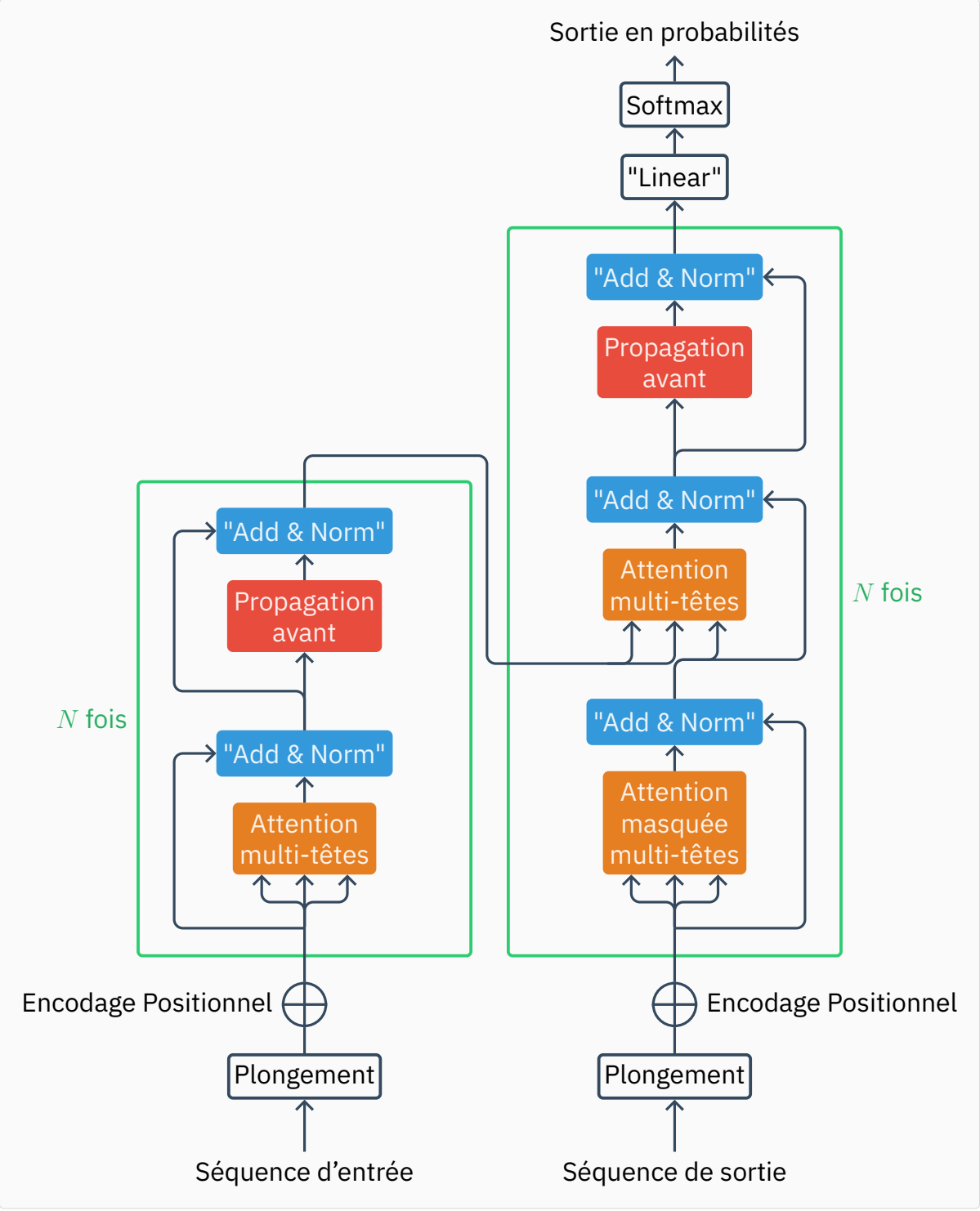


Figure 3 – Schéma d'un transformer

Le transformer est un modèle état-de-l'art basé sur le mécanisme d'attention et dont le but est d'effectuer des tâches manipulant de longues séquences

Comme on peut le voir sur la Figure 3, la séquence d'entrée subit d'abord un traitement (découpage de la séquence en tokens etc) pour ensuite traverser une série d'encodeurs suivie d'une série de décodeurs. À noter toutefois que cette structure varie légèrement selon la version du transformer considérée (GPT, BERT, XLNET, ...)

Lors de la phase d'apprentissage on fournit également au modèle la séquence de sor-

tie correspondante et lors de la phase de prédiction on remplace cette séquence par une matrice qui aura été apprise.

De plus, le transformer traite la séquence comme un tout et cela le rend beaucoup plus rapide à entraîner grâce à la possibilité de paralléliser les calculs à l'inverse du RNN et du LSTM.

Focus sur le mécanisme d'attention Le mécanisme d'attention consiste à « dire au reste du modèle, à quels mots de la séquence B il faut porter le plus d'attention quand on traite un mot de la séquence A ».

Ainsi l'auto-attention (lorsque $A = B$) définit pour chaque mot dans une séquence les autres mots de la séquence qui lui sont pertinents.

De plus, l'intérêt d'utiliser l'attention multi-têtes est de manipuler plusieurs notions de pertinence comme si on résumait l'information obtenue à partir de plusieurs têtes d'attention (chacune se spécialisant indépendamment tout au long de l'apprentissage).

3.3.5 Système de Question-Réponse dans le domaine ouvert

La tâche de question-réponse dans le domaine ouvert est une tâche importante du TAL et a pour but de répondre à une question en langage naturel grâce à un corpus de documents non structurés contrairement à l'extraction de réponse dans le domaine fermé.

Elle fait appel à 3 sous-tâches :

1. l'implication textuelle qui consiste à identifier un document qui contient la réponse à la question
2. l'apprentissage à l'ordonnancement (Learning to rank) des documents selon leur pertinence pour la question donnée. L'approche basique qui est toujours utilisée pour cette sous-tâche consiste à représenter la question et les documents sous forme de vecteurs dans un espace euclidien afin de trier les documents selon la valeur du produit scalaire avec la question
3. et l'extraction de réponse dans un document où les modèles transformers

Focus sur DrQA DrQA est un système pour la compréhension écrite appliquée à la tâche d'OQA.

1. Le chercheur de documents de ce système est organisé autour du corpus indexé des pages de Wikipedia et son objectif est de retourner celles les plus pertinentes vis-à-vis de la question.
2. Ensuite l'extracteur de réponse utilise un réseau de neurones Bi-LSTM pour extraire les réponses candidates des pages qui ont été sélectionnées en y assignant un score.
3. La réponse finale retournée sera donc celle qui aura le meilleur score.

Sur la base de données SQuAD 1.1 (Standford Question-Answering Dataset) 70% des réponses données par le système correspondent exactement aux réponses possibles (en effet plusieurs formulations de la réponse sont acceptées).

3.4 Outils et concepts mis à disposition

3.4.1 La machine virtuelle (VM)

Tout au long de ce stage j'ai eu accès à un environnement virtuel situé sur les machines du LISN avec des performances et un stockage conséquents. Cette VM m'a donc permis de déployer et tester les différentes versions de mon architecture en conditions de qualification pour simuler les conditions réelles.

Elle s'est également révélée très utile pour les tâches coûteuses en temps là où cela devient très vite embêtant de garder son ordinateur allumé toute une nuit.

3.4.2 Rasa pour le système de dialogue

Rasa est un framework open-source utilisé pour créer des agents conversationnels. Cet outil permet donc :

1. de se connecter grâce à Rasa Core à plusieurs types de canaux de messagerie comme par exemple Facebook Messenger mais il est également possible d'intégrer son propre canal en utilisant le protocole HTTP ou bien le protocole WebSocket.
2. de déterminer ses propres intentions (exemple : *get_library_address*) et ses propres entités (ex : *library*)
3. de traiter grâce à Rasa NLU la compréhension du langage naturel, ou Natural Language Understanding (NLU) pour extraire les intentions (Intent Detection) et les concepts sémantiques (Slot Filling) des entrées des utilisateurs
   (exemple : « Quelle est l'adresse de la bu d'orsay ? » sera interprétée par Rasa en `{'intention':'get_library_address','entities':{'library':'bu d'orsay'}}`)
4. d'intéragir grâce à Rasa Action Server avec un « action server » local écrit en Python pour effectuer ses propres actions personnalisées (ex : aller chercher l'adresse d'une bibliothèque dans une base de données, faire une requête HTTP pour récupérer les horaires d'une bibliothèque etc)

Ainsi le gros avantage de Rasa est sa versatilité et sa facilité d'utilisation. En effet il est possible d'utiliser dans Rasa NLU des composants faisant appel à des modèles de DL pour la détection d'intention et le Slot Filling ; Rasa se charge alors de proposer des commandes pour automatiser les tests et le fine-tuning (l'ajustement fin des paramètres de modèles déjà entraînés et fonctionnels pour les faire correspondre à des données supplémentaires).

3.4.3 Vespa pour stocker et accéder aux documents

Vespa est un moteur de recherche orienté documents très performant et idéal pour l'implémentation d'un SQR. En effet Vespa dispose de fonctionnalités avancées pour la récupération de documents candidats en utilisant des approches vectorielles, des approches plus traditionnelles ou une combinaison de plusieurs approches.

Par exemple il est possible de récupérer les documents les plus pertinents par rapport à leur distance dans un espace euclidien à une question donnée.

Une fois Vespa instanciée, la personnalisation du moteur de recherche se fait en deux étapes :

1. une première phase de déploiement qui consiste à définir les différents types de documents appelés schémas et leurs champs (voir `Pyvespa Deploy` sur la Figure 4)

2. suivie d'une deuxième phase, l'alimentation de la base de données, qui consiste à fournir les données en précisant pour chaque document à quel schéma il correspond (voir `Pyvespa Feed` sur la Figure 4).

Pour la personnalisation et l'interaction avec Vespa il existe notamment un module Python nommé `pyvespa` qui permet d'utiliser toutes ses particularités.

3.4.4 Hugging Face et les transformers

Hugging Face est une plateforme qui propose des outils pour notamment construire, entraîner et déployer des modèles de ML.

Sur cette plateforme les chercheurs, les entreprises et les particuliers peuvent partager leurs modèles et leurs projets et le tout est accessible depuis une librairie Python. Cela permet ainsi de télécharger et d'utiliser les modèles disponibles.

À noter que Rasa utilise cette librairie pour certains de ses composants. On peut donc changer les paramètres pour pouvoir utiliser le modèle de notre choix à condition qu'il y soit disponible.

3.4.5 Tensorflow serving

Tensorflow serving est un système de déploiement de modèles de ML sur un environnement de production (ici la VM).

Cet outil permet ainsi d'effectuer des requêtes aux modèles déployés, d'organiser plusieurs versions d'un modèle et ce de façon simple et rapide.

3.4.6 Docker pour le déploiement du système

Docker est un ensemble de produits qui utilisent la la conteneurisation informatique pour pouvoir créer des conteneurs qui peuvent être exécutés indépendamment de l'environnement et du système d'exploitation auxquels ils sont soumis. En d'autres termes Docker facilite grandement le déploiement d'applications.

De plus Docker Compose, un des produits Docker, permet de définir et exécuter des applications avec plusieurs conteneurs. Ainsi on peut spécifier au sein d'un fichier de configuration les conteneurs de l'application, les connections entre les différents conteneurs et avec la machine hôte ou encore les commandes à exécuter pour chaque conteneur. Docker Compose est donc très pratique lorsqu'il s'agit de déployer une architecture complexe sur n'importe quel environnement.

3.5 Mise en œuvre de la solution

3.5.1 Données à disposition

Afin de réaliser un tel système, on m'a donné une liste de liens qui correspondaient à certaines pages du site internet de la DiBISO ainsi qu'un jeu de données représentant les questions des utilisateurs.

En effet sur ces pages on pouvait retrouver des explications rédigées sur les services proposés par les bibliothèques de la DiBISO et ces explications répondent à la grande majorité des questions.

En partant de là j'ai donc développé deux scripts Python (voir Figure 4) :

1. le `Scraper` qui prend en entrée un fichier `subdirectories.txt` qui correspond à la liste des liens des pages à récupérer et dont le contenu textuel sera extrait en utilisant les modules de web scraping pour l'enregistrer sous le format `JSON`
2. et le `Splitter` qui prend en entrée le fichier de sortie du `Scraper` et qui scinde chaque page en paragraphes de 2 phrases.

Ce seront ces paragraphes que l'on souhaitera retourner pour présenter une réponse dans son contexte (voir rendu final Figure 15). On obtient donc deux types de documents qui sont les pages et les paragraphes.

3.5.2 Division du problème

En regardant les données à disposition, on s'aperçoit rapidement que beaucoup de questions reviennent souvent et ont une réponse simple. C'est pourquoi j'ai divisé les informations à propos des bibliothèques en deux parties :

1. le corpus de documents (voir `Vespa` dans la Figure 4) qui reprend la sortie du `Scraper` et du `Splitter`.
2. et le fichier `database.json` (voir Figure 4) qui regroupe les informations générales pour chaque bibliothèque comme l'adresse ou les spécialités.

Ainsi j'ai pu répartir à la fois l'ensemble de questions qui m'a été fourni et celles que j'ai générées sur une sélection de 7 intentions :

1. *greet* lorsque l'intention est de saluer le SQR (réponse générique sans passer par le serveur d'actions)
2. *bot_challenge* lorsque l'utilisateur souhaite en savoir plus sur le SQR (réponse générique sans passer par le serveur d'actions)
3. *get_timetable_of_library* quand il s'agit d'obtenir les horaires ou l'emploi du temps d'une bibliothèque. Cette intention est reliée directement à une action faisant une requête au service web Affluences qui propose une API pour les horaires des espaces publics. Ainsi grâce au Slot Filling le SQR peut récupérer le nom de la bibliothèque dont parle l'utilisateur pour lui retourner son emploi du temps.
4. *search_library_fields* si on veut connaître les spécialités d'une bibliothèque en particulier (extraction de l'information dans la base de données statique)
5. *search_library_address* quand l'utilisateur souhaite obtenir l'adresse d'une bibliothèque (extraction de l'information dans la base de données statique)
6. *search_libraries_from_field* chaque fois qu'on souhaitera connaître l'ensemble des bibliothèques qui ont des ressources dans un domaine en particulier (extraction de l'information dans la base de données statique)
7. *out_of_scope* lorsqu'il faut aller extraire la réponse dans le corpus de documents.

3.5.3 Apport du DL au système

Focus sur l'intention *out_of_scope* Cette intention est directement reliée à une action qui a pour but de retourner la réponse extraite, le contenu textuel du paragraphe d'où provient la réponse et le lien vers la page internet correspondante.

Ainsi la récupération de la réponse se fait grâce à 2 modèles :

1. le modèle multilingue *sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2* (disponible sur Hugging Face) de [RG19] pour plonger les paragraphes dans un espace vectoriel avec 384 dimensions ainsi chaque vecteur représentant un paragraphe est stocké dans Vespa. Ensuite en utilisant le même modèle pour plonger la question de l'utilisateur on peut demander à Vespa les 5 paragraphes dont les distances euclidiennes à la question sont les plus faibles. On obtient donc les 5 paragraphes censés être les plus pertinents vis-à-vis de la question.
2. le modèle *etalab-ia/camembert-base-squadFR-fquad-piaf* (disponible sur Hugging Face) qui est le modèle transformer CamemBERT de [Mar+20] après un fine-tuning sur 3 jeux de données de questions-réponses en français ([Ker+20], [dHo+20] et [Kab18]). Ce modèle prend en entrée un paragraphe et une question et renvoie ainsi le score et la portion du paragraphe la plus probable. Par conséquent en comparant les scores des réponses candidates extraites des 5 paragraphes grâce au modèle déployé sur le conteneur Tensorflow serving (voir Figure 4) on peut alors sélectionner la meilleure réponse.

Une fois la meilleure réponse identifiée, le système retourne également l'historique des scores pour les analyses.

Focus sur la détection d'intentions et le Slot Filling Comme présenté au paragraphe 3.4.2, l'outil Rasa est doté d'un ensemble de composants pour le Slot Filling et la détection d'intention.

Pour ce faire j'ai d'abord essayé d'utiliser les composants par défaut qui sont notamment basés sur des expressions régulières avancées. Seulement les résultats n'étant pas concluant, Sahar Ghannay, Sophie Rosset et Christophe Servan m'ont conseillé d'utiliser le modèle transformer FrALBERT [CSR21] qui est justement originaire du LISN et également disponible sur Hugging Face.

Cependant le composant `LanguageModelFeaturizer` de Rasa ne peut charger depuis Hugging Face que les modèles BERT, GPT, GPT-2, XLNET, DistilBERT, et roBERTa. Ainsi il m'a fallu le modifier en rajoutant les fonctions nécessaires pour qu'il puisse prendre en charge les modèles ALBERT.

Une fois l'outil Rasa modifié pour prendre en charge les modèles ALBERT et donc pouvoir y utiliser FrALBERT j'ai créé une image Docker avec cette version de Rasa personnalisée. Grâce à cela j'ai pu fine-tuner FrALBERT avec 12 312 questions sur les tâches de détection d'intention et de Slot Filling.

C'est à cette étape qu'une VM est très utile car cela a pris une dizaine d'heures et que j'ai dû réitérer la procédure pour comparer les résultats de Fralbert à ceux de CamemBERT [Mar+20] (voir Figures 6, 7, 2 et 8).

3.5.4 Architecture du système

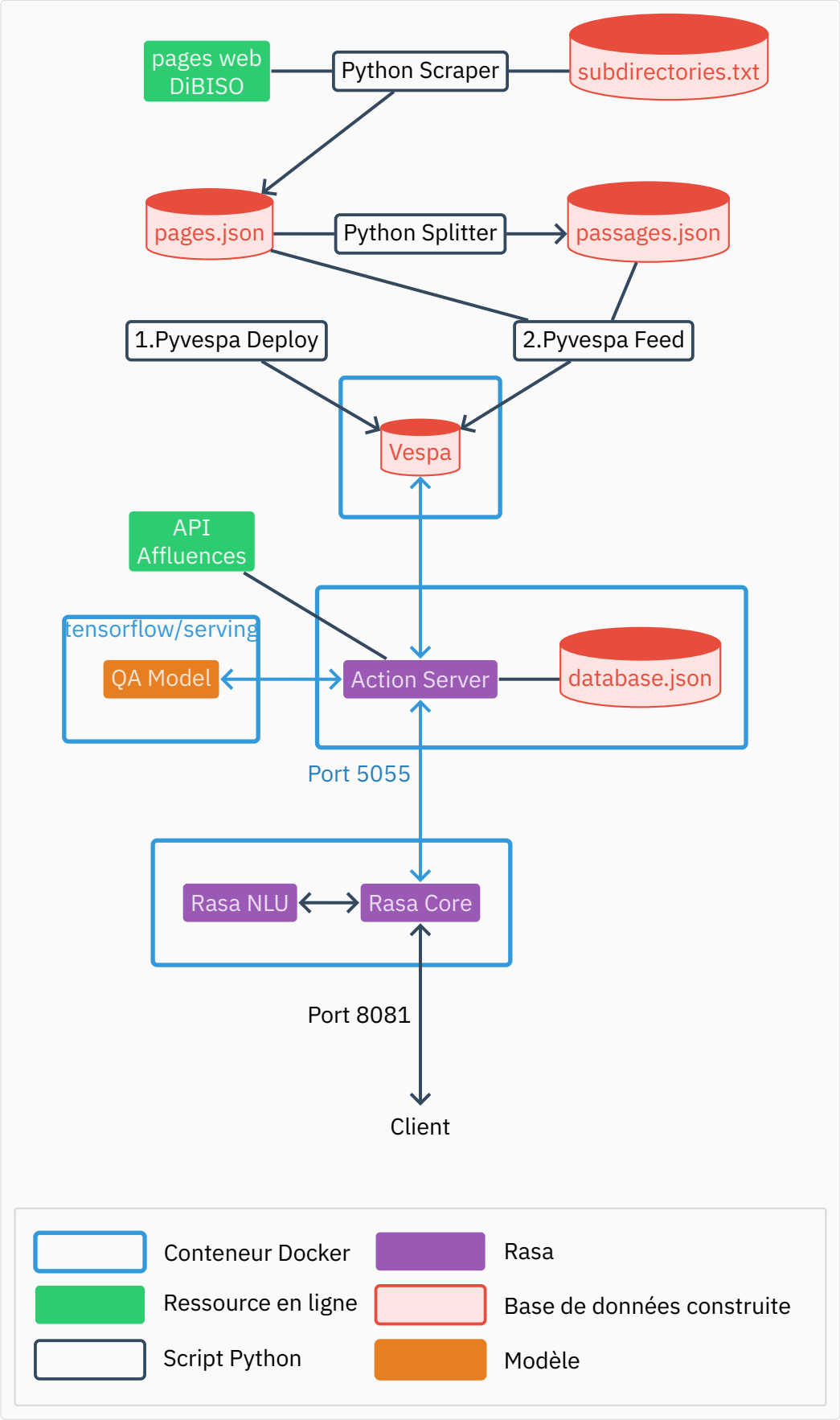


Figure 4 – Architecture du système

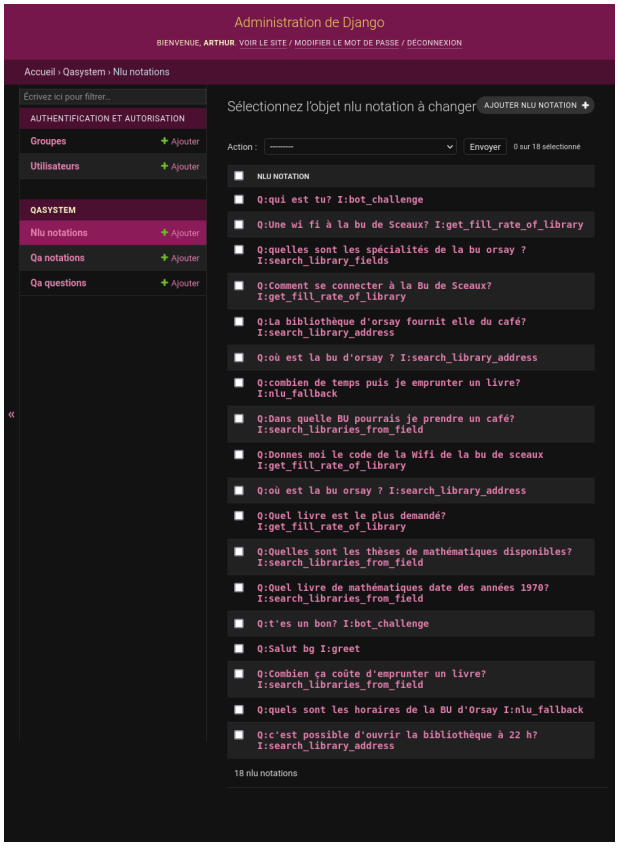
3.5.5 Développement d'une application web pour la mise en place de modules d'annotations

Pour pouvoir lancer une campagne d'annotations auprès de la DiBISO et ainsi récolter un ensemble de données qui serviront à améliorer le système (grâce au fine tuning des modèles) j'ai développé une application web grâce au framework python open-source appelé `Django`.

Cette application web permet donc :

- de gérer plusieurs utilisateurs avec un système d'authentification de sorte à pouvoir par exemple supprimer toutes les annotations d'un utilisateur si les données montrent qu'il n'a pas bien suivi les instructions,
- d'intéragir avec le SQR via une interface graphique simple,
- d'effectuer des annotations pour la détection d'intention mais aussi pour la sélection de documents et l'extraction de réponses (voir Figure 5a),
- de permettre aux utilisateurs "staff" de visualiser, modifier, supprimer, et exporter les annotations (voir Figure 5b),
- et de pouvoir rajouter des pages pour les informations annexes (description de l'éventuelle campagne et des instructions pour l'annotation, exemples de questions posées au modèle, visualisation du contenu textuel qui a été extrait des pages etc).

(a) Question-réponse avec annotations



(b) Panneau administrateur

Figure 5 – Application web pour le SQR

3.6 Résultats et discussion

Pour évaluer et faire du fine-tuning sur FrALBERT [CSR21] j'ai constitué un ensemble de questions annotées pour chaque intention, et un ensemble de variations pour chaque entité. Ensuite je l'ai divisé en deux pour avoir un corpus dit d'entraînement et un corpus de test.

3.6.1 Résultats de la détection d'intention

Rappel, Précision et Score F1

$$\text{Rappel}_i = \frac{\text{nb de questions correctement attribuées à l'intention } i}{\text{nb de questions appartenant à l'intention } i}$$

$$\text{Précision}_i = \frac{\text{nb de questions correctement attribuées à l'intention } i}{\text{nb de questions attribuées à l'intention } i}$$

$$\text{Score F1}_i = 2 \times \frac{\text{Précision}_i \times \text{Rappel}_i}{\text{Précision}_i + \text{Rappel}_i}$$

Intention	Ex. fine-tuning	Ex. Test	Précision	Rappel	F1-Score
greet	9	3	0.750	1.000	0.857
bot_challenge	43	2	1.000	0.500	0.667
get_timetable_of_library	2734	138	0.893	0.667	0.763
search_library_fields	4911	136	0.814	1.000	0.898
search_library_address	2607	248	0.975	0.931	0.953
search_libraries_from_field	514	65	0.964	0.815	0.883
out_of_scope	1494	374	0.902	0.963	0.931
Moyenne	**12 312**	**966**	**0.900**	**0.840**	**0.850**
Moyenne pondérée			**0.911**	**0.907**	**0.904**

Figure 6 – Résultats du module de détection d'intention avec FrALBERT

Intention	Ex. fine-tuning	Ex. Test	Précision	Rappel	F1-Score
greet	9	3	1.000	0.333	0.500
bot_challenge	43	2	0.500	0.500	0.500
get_timetable_of_library	2734	138	0.614	0.818	0.702
search_library_fields	4911	136	0.814	1.000	0.898
search_library_address	2607	248	0.933	0.339	0.497
search_libraries_from_field	514	65	0.000	0.000	0.000
out_of_scope	1494	374	0.625	0.938	0.750
Moyenne	**12 312**	**966**	**0.664**	**0.547**	**0.555**
Moyenne pondérée			**0.711**	**0.696**	**0.652**

Figure 7 – Résultats du module de détection d'intention avec CamemBERT

Tout d'abord on peut dire que d'après la Figure 6 les résultats de FrALBERT [CSR21] pour la détection d'intention sont plutôt bons. Il est aussi intéressant de les comparer à ceux

de CamemBERT [Mar+20] qui est un modèle plus grand avec 110 millions de paramètres contre 12 millions pour FrALBERT. Ainsi en comparant avec la Figure 7 on remarque que FrALBERT est ici beaucoup plus performant.

Cependant le fait que CamemBERT ait des résultats si bas alors qu'il est censé être légèrement plus performant est assez équivoque. C'est pourquoi je remettrais plutôt en question mon implémentation de CamemBERT sur Rasa abordée au paragraphe 3.5.3 mais par manque de temps et en voyant les résultats de FrALBERT je ne m'y suis pas attardé.

En outre le modèle est capable de bien identifier les intentions qui n'ont pas beaucoup d'exemples par rapport aux autres comme *greet* qui a un rappel de 1,000. À noter également que le modèle va avoir tendance à se tromper davantage si l'intention de l'utilisateur est de récupérer l'emploi du temps (*get_timetable_of_library*) on pourrait alors essayer de se pencher sur une façon de mieux définir cette intention en la scindant.

De plus l'intention *out_of_scope* qui est beaucoup plus floue et difficile à définir que les autres est étonnamment bien identifiée par le modèle avec un score F1 supérieur à 0,9.

3.6.2 Résultats du slot filling

Entité	Ex. Test	Précision	Rappel	F1-Score
library	1560	0.970	0.778	0.863
month	16	0.000	0.000	0.000
day_of_the_week	34	1.000	0.765	0.867
field	435	0.953	0.876	0.913
Moyenne	**2045**	**0.731**	**0.605**	**0.661**
Moyenne pondérée		**0.959**	**0.792**	**0.867**

Table 2 – Résultats du Slot Filling avec FrALBERT

Entité	Ex. Test	Précision	Rappel	F1-Score
library	1560	0.906	0.760	0.826
month	16	0.000	0.000	0.000
day_of_the_week	34	0.692	0.529	0.600
field	435	0.701	0.425	0.529
Moyenne	**2045**	**0.575**	**0.429**	**0.489**
Moyenne pondérée		**0.852**	**0.679**	**0.753**

Figure 8 – Résultats du Slot Filling avec CamemBERT

Avant toute chose, je tenais à préciser que l'effet boîte noire de Rasa qui propose toutes les commandes pour les tests m'ont empêché de comprendre pourquoi dans les résultats pour le Slot Filling il y a un total de 1560 exemples afin de tester la reconnaissance du champ *library* alors qu'il n'est pas censé y en avoir plus de 966. En effet il y a 966 exemples de questions et chaque question contient au plus une fois le champ *library* donc cela paraît impossible.

Malgré cela, sur les Figures 2 et 8 tous les champs *library* et *field* qui sont ceux qui nous intéressent vraiment ici pour les actions ont un score F1 très correct pour FrALBERT [CSR21] car en effet les résultats de CamemBERT [Mar+20] restent en dessous.

3.6.3 Résultats du sélecteur de passages

Afin d'évaluer le sélecteur de passages mais aussi le module d'extraction (voir 3.6.4), j'ai parcouru le corpus de documents pour obtenir un ensemble de questions auxquelles le système peut répondre. Ainsi j'ai répertorié 150 questions avec la page et le passage correspondants et c'est à partir de ce jeu de données que j'ai effectué les évaluations suivantes.

Le rang réciproque moyen (ou Mean Reciprocal Rank) (MRR) mesure l'efficacité d'un sélecteur de k documents (ici ce sont des pages et des passages) pour chaque élément appartenant à un échantillon Q de questions.

En effet pour chaque question on récupère le rang r_i du premier document pertinent, c'est à dire du premier document qui contient une réponse à la question. Ensuite on calcule son rang réciproque en prenant son inverse (si aucun document pertinent n'est présent alors on prend un rang réciproque nul).

$$\text{MRR@}k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{r_i}$$

$$\begin{cases} Q = \text{ famille des questions} \\ k = \text{ le nombre de pages ou passages récupérés pour chaque question} \\ r_i = \text{ rang du premier document (page ou passage) pertinent} \end{cases}$$

k	Valeur
MRR@3	0.733
MRR@5	0.742
MRR@10	0.751
MRR@20	0.753

Figure 9 – MRR des pages

k	Valeur
MRR@3	0.357
MRR@5	0.369
MRR@10	0.386
MRR@20	0.395

Figure 10 – MRR des passages

Figure 11 – MRR en faisant varier k le nombre de passages récupérés

Comme la sélection se fait sur les passages il s'agit surtout d'analyser la Figure 10. On y voit que le paragraphe pertinent se situe généralement autour de la deuxième et troisième place donc c'est un choix approprié de récupérer uniquement les 5 premiers paragraphes pour l'étape suivante qui est l'extraction de la réponse.

De plus une question admet parfois plusieurs réponses alors que dans le cas présent je n'ai mentionné qu'un paragraphe pertinent pour chaque question. J'ai fait ce choix par simplicité car cela devient très vite fastidieux de retrouver toutes les réponses possibles à une question dans des centaines de paragraphes qui se recoupent. Malgré tout il faut tenir compte de cette erreur qui joue en notre faveur car elle nous fait sous-estimer les performances du sélecteur de passages.

Rappel mesure la proportion de questions où un document pertinent est présent dans la sélection (de k documents) peu importe sa position.

En effet on souhaite pouvoir retrouver un document pertinent dans la sélection à chaque fois et sa position n'a pas d'importance car le choix se fera fait à partir du score de l'extracteur de réponses.

$$\text{Rappel@}k = \frac{\text{nb de sélections comportant au moins 1 document pertinent}}{\text{nb de sélections}}$$

k	Valeur
Recall@1	0.654
Recall@3	0.827
Recall@5	0.866
Recall@10	0.933
Recall@20	0.955
Recall@50	0.972
Recall@100	1.000

Figure 12 – Rappel des pages

k	Valeur
Recall@1	0.296
Recall@3	0.430
Recall@5	0.486
Recall@10	0.603
Recall@20	0.737
Recall@50	0.872
Recall@100	0.961

Figure 13 – Rappel des passages

Figure 14 – Rappel en faisant varier k le nombre de passages récupérés

Sur la Figure 12 on remarque que le Recall@5 est très correct pour les pages. Ainsi un utilisateur voyant une réponse fausse pourra au moins être redirigé sur la bonne page. En effet comme le montre la Figure 13, seulement 49% des questions trouvent leur réponse dans les 5 paragraphes pour l'extraction ce qui est insuffisant surtout lorsque l'on sait que les erreurs de chaque module se multiplient. C'est pourquoi il serait peut-être intéressant de récupérer davantage de paragraphes pour l'extraction et obtenir par exemple les résultats du Recall@20 de 73%.

3.6.4 Résultats du module d'extraction de réponses

Sur les 150 questions répertoriées, le système trouve une réponse correcte pour 62 d'entre elles. Il y a donc 41,3% des questions qui sont bien traitées ce qui me semble être un résultat honnête étant donné que pour l'instant la campagne d'annotation et d'amélioration du système n'a pas encore eu lieu. De plus j'ai sélectionné une poignée de questions auxquelles le système a correctement répondu et que l'on peut retrouver sur la Figure 15 afin d'avoir une idée de son potentiel.

Bien évidemment je n'ai pas compté les questions auxquelles le système ne peut pas répondre par manque de données dans les passages. En effet si un utilisateur souhaite connaître l'auteur d'un livre, il est impossible de lui répondre avec le contenu textuel des pages qui ont été récupérées. Cependant il est possible d'ajouter du contenu textuel afin de permettre au système de répondre à de nouveaux types de question comme celui-ci. C'est un des grands avantages de ce genre d'architecture.

3.6.5 Rendu final

Figure 15 – Captures d'écran du SQR déployé

4 Conclusion

À l'issue de ce stage j'ai donc pu mettre en place prototype de système conversationnel et de question-réponse sachant s'adapter à n'importe quel corpus de documents pouvant ainsi répondre automatiquement aux questions en langage naturel des utilisateurs.

Ce prototype a été appliqué aux besoin de la DiBISO et s'appuie pour l'instant sur le contenu textuel du site internet associé [DiB]. Il est ainsi en mesure de détecter l'intention de l'utilisateur et de lui proposer une réponse.

Pour l'instant ce modèle est perfectible c'est pourquoi j'ai mis en place une application web pour intéragir avec le système et annoter ses réponses. En effet au terme de ce stage commencera une campagne d'annotation avec le personnel de la DiBISO pour construire un ensemble de données qui serviront à améliorer le système.

De plus les résultats montrent que le sélecteur de passages est limitant pour le reste du système. C'est pourquoi on pourrait envisager une autre approche pour récupérer les passages les plus pertinents en combinant l'approche vectorielle basée sur le plongement de paragraphes à la méthode TF-IDF de manière à ce qu'on ait une requête hybride tenant compte à la fois du sens et des mots clés du passage.

Le rendu est donc une première esquisse qui évoluera au sein du partenariat entre le LISN et la DiBISO pour un jour être utilisé et permettre aux étudiants ainsi qu'au personnel d'accéder plus facilement à l'information.

5 Bibliographie

Références

[Che+18] Howard Cheung et al. « A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation ». In : *Applied Energy* 222 (2018), p. 329-342. issn : 0306-2619. doi : https://doi.org/10.1016/j.apenergy.2018.03.138. url : https://www.sciencedirect.com/science/article/pii/S0306261918304768.

[Kab18] Ali Kabbadj. *Something new in French Text Mining and Information Extraction (Universal Chatbot) : Largest QA French training dataset (110 000+)*. Sous la dir. de linkedin.com. [Online ; posted 11-November-2018]. Nov. 2018. url : https://www.linkedin.com/pulse/something-new-french-text-mining-information-chatbot-largest-kabbadj.

[RG19] Nils Reimers et Iryna Gurevych. « Sentence-BERT : Sentence Embeddings using Siamese BERT-Networks ». In : *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, nov. 2019. url : http://arxiv.org/abs/1908.10084.

[Rol+19] David Rolnick et al. *Tackling Climate Change with Machine Learning*. 2019. doi : 10.48550/ARXIV.1906.05433. url : https://arxiv.org/abs/1906.05433.

[Sch+19] Roy Schwartz et al. *Green AI*. 2019. doi : 10.48550/ARXIV.1907.10597. url : https://arxiv.org/abs/1907.10597.

[dHo+20] Martin d'Hoffschmidt et al. « FQuAD : French Question Answering Dataset ». In : *ArXiv* abs/2002.06071 (2020).

[Ker+20] Rachel Keraron et al. « Project PIAF : Building a Native French QA Dataset ». In : *LREC*. European Language Resources Association, 2020, p. 5481-5490.

[Mar+20] Louis Martin et al. « CamemBERT : a Tasty French Language Model ». In : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[CSR21] Oralie Cattan, Christophe Servan et Sophie Rosset. « On the Usability of Transformers based models for a French Question-Answering task ». In : *Recent Advances in Natural Language Processing, RANLP 2021*. Online, sept. 2021.

[DiB] DiBISO. *Site des bibliothèques universitaires Paris-Saclay*. url : https://www.bibliotheques.universite-paris-saclay.fr/.

6 Glossaire

Acronymes

API Interface de Programmation d'Application. 5, 15

BOW bag of words ou sac de mots en français. 6, 8

DiBISO Direction des Bibliothèques, de l'Information et de la Science Ouverte. 3–5, 15, 18, 24

DL Deep Learning ou Apprentissage Profond en français. 2, 4, 5, 13, 16, 28

ILES Information Langue Écrite et Signée. 4, 5

LISN Laboratoire Interdisciplinaire des Sciences du Numérique. 3–5, 13, 16, 24, 28

LSTM Long Short Term Memory. 10, 12

ML apprentissage automatique ou machine learning en anglais. 6, 14

MRR rang réciproque moyen (ou Mean Reciprocal Rank). 21

NLU compréhension du langage naturel, ou Natural Language Understanding. 13

OQA Question-Réponse dans le domaine Ouvert ou Open domain Question-Answering. 5, 6, 12

RNN réseau de neurones récurrents. 8–10, 12

SQR Système de Question-Réponse. 3–5, 13, 15, 18, 23

TAL Traitement Automatique des Langues. 4–6, 8, 12, 27

TF-IDF une méthode de pondération de l'anglais term frequency-inverse document frequency. 6, 8, 24

VM machine virtuelle. 2, 4, 13, 16

Définitions

fine-tuning l'ajustement fin des paramètres de modèles déjà entraînés et fonctionnels pour les faire correspondre à des données supplémentaires. 13, 16, 19

la conteneurisation informatique est une méthode qui propose une manière de virtualiser des ressources de manière légère, avec une isolation garantie par le système d'exploitation. Ces ressources sont ainsi plus facilement portables d'un système à un autre. C'est un puissant accélérateur de développement d'applications. 4, 14

plongement de mots ou word embedding en anglais est représentation de tout mot par un vecteur dense de nombres réels (les mots présents dans des contextes similaires ont une représentation relativement proche dans l'espace vectoriel). 6

Slot Filling identification des différents "slots" correspondant aux différents paramètres de la requête de l'utilisateur. 13, 15, 16, 20

token (ou jeton en français) une chaîne de caractères qui joue un certain rôle dans un langage écrit (ex: "vacances", "!") et correspond généralement à un mot. 2, 6, 8, 10, 11

transformer modèle état-de-l'art basé sur l'apprentissage profond et sur le mécanisme d'attention dans le but de traiter des séquences pour effectuer certaines tâches du TAL comme l'extraction de réponse dans un texte. 2, 11, 12, 14

7 Annexes

7.1 Annexe DD&RS

Pression du secteur sur l'environnement Le domaine de l'intelligence artificielle et plus particulièrement du DL a un impact non négligeable sur l'environnement. En effet qu'il s'agisse de la consommation d'énergie des équipements (entre 70 et 80% pour les serveurs, entre 7 et 20% pour l'équipement du réseau et entre 10 et 20 % pour la coordination électrique selon [Che+18]) , du coût de leur fabrication et de leur fin de vie ou de la pollution générée par l'extraction des matières premières nécessaires les conséquences de l'utilisation de ce genre de programmes sont à prendre en compte.

Cependant il est également possible d'utiliser l'intelligence artificielle afin d'avoir un impact positif sur l'environnement [Rol+19] par exemple en utilisant la surveillance avec télédétection (repérage des déforestations, collection des données sur un bâtiment, estimation des dégâts après des catastrophes etc) ou en optimisant des systèmes pour améliorer leur efficacité et moins polluer.

FrALBERT vs CamemBERT FrALBERT [CSR21] contient 12 millions de paramètres contre 110 millions pour CamemBERT [Mar+20]. Or nous avons l'équation simplifiée [Sch+19] :

$$\text{Coût} = E \times D \times H \quad \text{où} \quad \begin{cases} E = \text{coût d'un exemple} \\ D = \text{taille du corpus d'entraînement} \\ H = \text{nombre d'hyperparamètres} \end{cases}$$

Donc en faisant baisser le nombres de paramètres on réduit le coût environnemental du modèle. En effet FrALBERT a une consommation de 0.57 kWh contre 1.08 kWh pour CamemBERT (voir Table 3 de [CSR21]).

Actions du LISN Les facettes de l'engagement environnemental du LISN sont multiples et parmi elles on retrouve :

1. des thèmes de recherche orientés sur les impacts environnementaux du numérique
2. la cellule Science Responsable pour réfléchir aux conséquences des activités des chercheurs sur l'environnement (ex : expérimentation Labo 1point5 en vue de réduire l'émission des gaz à effet de serre des laboratoires)
3. des enseignements autour du développement durable et des impacts du numérique
4. la réalisation d'un bilan carbone chaque année