Système automatique de traduction

Rapport final de validation

Rapport de synthèse sur les différentes méthodes de traduction explorées dans l'optique d'une traduction de l'anglais au français

MHEDHBI Sonia,
MOREAU Coralie,
SAADAOUI Mounir

Promotion Juillet 2021

0. Contenu

1. Contexte et objectifs

L'objectif du projet est d'adapter un système de traduction au projet de lunettes connectées déjà réalisé. Le système implémenté par ces lunettes permet de localiser, de transcrire la voix d'un interlocuteur et d'afficher la transcription sur des lunettes connectées. Dans ce projet d'amélioration, nous implémenterons un système de traduction qui permettrait d'élargir l'utilisation de ces lunettes à un public plus vaste et permettrait à deux individus ne pratiquant pas la même langue de pouvoir communiquer aisément.

Ce projet de traduction de l'anglais vers le français s'inscrit dans le domaine spécifique des NLP (Natural Langage Processing ou traitement du langage naturel), l'une des tâches les plus populaires en data science. Il s'agit de représenter du texte sous forme mathématique afin d'en extraire la sémantique (la signification) pour son traitement ultérieur.

Les objectifs du projet étaient multiples :

- Choisir un jeu de données adéquat (étendu, diversifié mais de taille raisonnable) ;
- Sélectionner et personnaliser un modèle de NLP ;
- Déterminer une (ou des) métrique(s) visant à analyser et comparer les résultats ;
- Adapter le modèle au système de lunettes connectées.

Nous nous sommes attachés à l'accomplissement de ces objectifs, étant tous trois novices en Data science et en NLP a fortiori.

2. Jeux de données

2.1. Liminaire sur les types de modèles

Ce rapport explicite les différentes méthodes de traduction explorées en vue de traduire l'anglais vers le français. Nous pouvons départager les modèles selon leur capacité à traduire du mot à mot ou séquence par séquence. Nous expliciterons chacun de ces modèles par la suite mais le schéma ci-après permet de saisir d'ores et déjà le cheminement opéré.

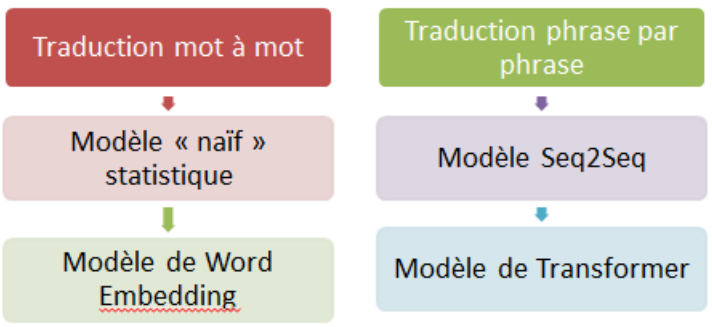


Figure 1: Présentation des modèles testés

En première analyse, on peut déduire qu'il sera difficile d'obtenir des traductions performantes de phrases en procédant en mot à mot. Toutefois, comme nous le verrons, le

modèle de Word Embedding est très important pour la suite car ses fondements sont au cœur des systèmes plus complexes de traduction phrase par phrase.

2.2. Jeux de données utilisés

En raison de la diversité du fonctionnement des modèles élaborés, nous n'avons pas utilisé les mêmes jeux de données tout au long de l'étude.

Ce tableau reprend les infos des jeux de données d'entrainement et de validation utilisés selon les étapes : toutes les données étant en accès libre.

Tableau 1: Tableau listant les jeux de données utilisés comme test et validation selon les étapes de modélisation

Etape de modélisation	Jeux de données d'entrainement	Jeux de données de validation ou métriques de validation
Mot à mot statistique	« Petit ensemble de phrases » : Ensemble relativement réduit de phrases traduites en anglais en en français.	« Dictionnaire Facebook » : Large ensemble de mots anglais proposant diverses traductions en français.
Word Embedding	« Matrices pré-entrainées de Word embedding » : ensemble de mots vectorisés par un entrainement très long (2 000 000 de mots) + « Dictionnaire Facebook ».	« Dictionnaire Facebook »
Seq2Seq	« Petit ensemble de phrases »	Non concerné – tests sur la longueur des phrases et la racine des mots
Seq2Seq	« Grand ensemble de phrases »	Non concerné – tests sur la longueur des phrases et la racine des mots puis applications des scores BLEU et ROUGE
Transformer	« Grand ensemble de phrases »	Non concerné – tests sur la longueur des phrases et la racine des mots puis applications des scores BLEU et ROUGE

Pour information, le « Grand ensemble de phrases » comprend 154 882 phrases et 15 311 mots uniques en Anglais pour 22 773 mots uniques en Français.

2.3. Visualisation du jeu de données « Petit ensemble de phrases »

Nous avons effectué une analyse minutieuse du jeu de données « Petit ensemble de phrases » via les bibliothèques Matplotlib et Seaborn et en avons déduit un certain nombre d'observations utiles pour les étapes de modélisation.

Comparaison du nombre de mots en Français en en Anglais

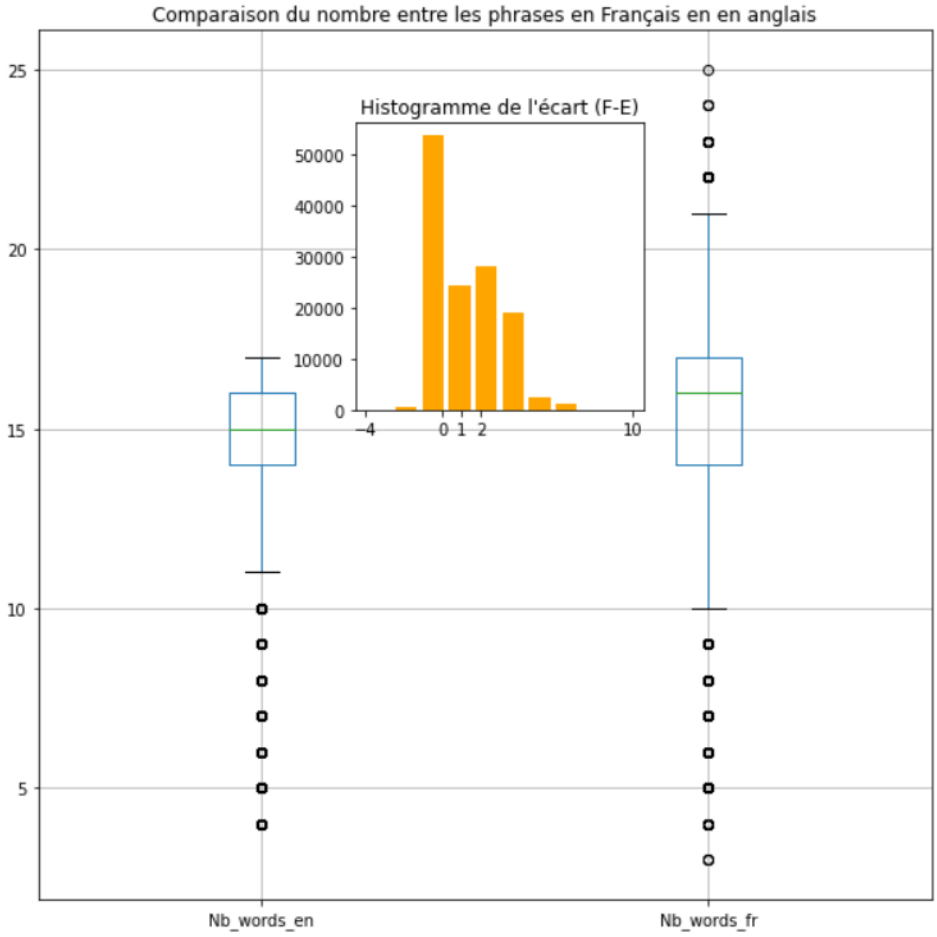


Figure 2: Graphique 1 relatif à l'étude sur le "Petit Ensemble de données"

Observations : La moyenne du nombre de mots par phrase en anglais (13.6) est légèrement inférieure à celle en français (14.8), les médianes sont relativement proches. Plus les phrases sont longues, plus l'écart semble se creuser, l'anglais semble une langue plus « compacte » sur les phrases longues. L'histogramme de l'écart montre qu'il y a beaucoup de phrases où l'écart est très légèrement négatif. Par la suite, nous avons constaté que les phrases interrogatives sont généralement plus longues en anglais qu'en français.

Par ailleurs, le nombre total de mots du jeu de données est très important (1,5 millions environ) mais le nombre de mots uniques est relativement faible (346 en français et 197 en anglais).

WordCloud des mots utilisés dans le jeu de données

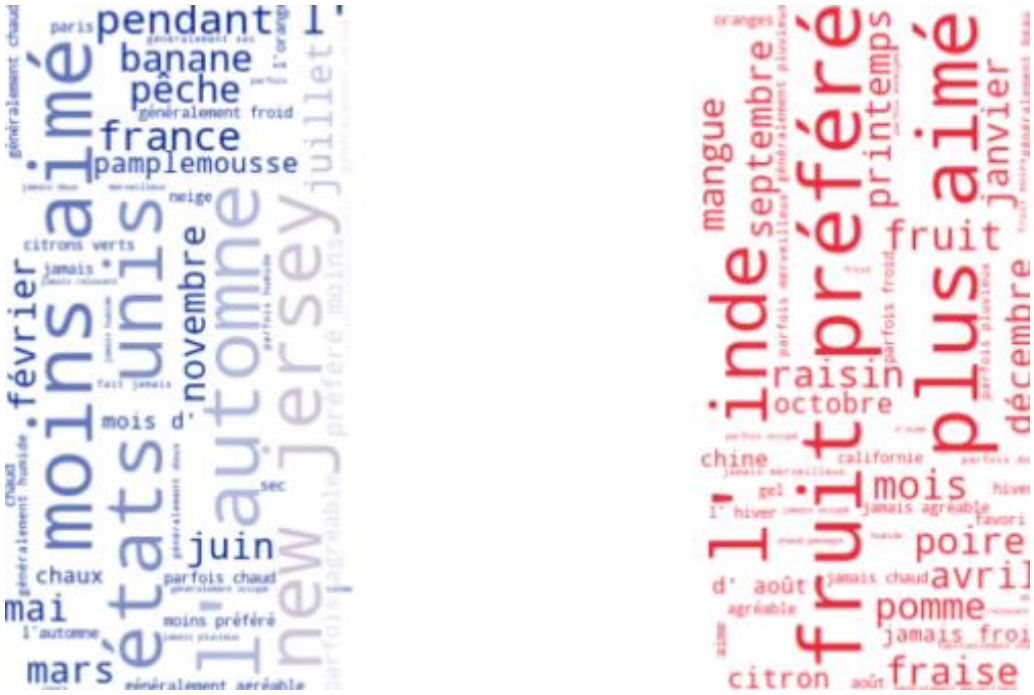


Figure 3: WordCloud des mots français du "Petit Ensemble de phrases"

Figure 4: WordCloud des mots anglais du "Petit Ensemble de phrases"

Observations : Nous observons que les thématiques des 100 mots les plus utilisées du jeu de données sont réduites : fruits, mois, saisons, pays et lieux.

Etude sur la nature des mots

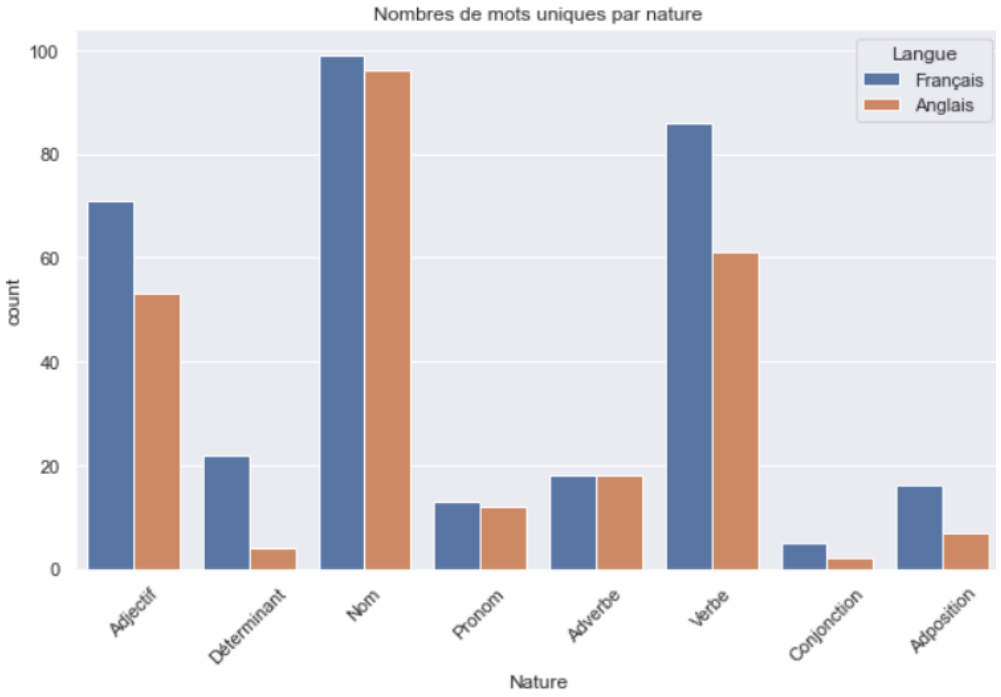


Figure 5: Etude sur la nature des mots sur "Petit Ensemble de données"

Observations : Nous avons pu constater que les noms et verbes sont les types majoritaires dans les deux langues. Dans la quasi-totalité des natures de mots, il y a plus de mots français qu'anglais (donc moins de correspondances).

Extrapolations pour la modélisation :

Ce jeu de données, bien que conséquent, comprend un ensemble de mots très redondant et les thématiques abordées sont assez restreintes. Les phrases sont également relativement courtes.

Si ce jeu de données permettait la première approche « naïve », il ne serait probablement pas pertinent pour des modèles plus élaborés.

Le jeu de données « Grand ensemble de phrases » a été préféré au « Petit ensemble de phrases » pour les modèles de Deep Learning afin d'avoir un ensemble de phrases plus diversifié donc un modèle qui se généralise mieux.

3. Modélisation 1 : Traitement mot par mot

3.1. Principe de la méthode

Dans cette itération, nous avons réalisé notre première modélisation de traduction par un système de traduction mot à mot.

L'hypothèse de base de cette itération est l'hypothèse suivante : "H0 : dans une phrase donnée, les mots à traduire de l'anglais vers le français sont quasiment à la même position (c'est-à-dire dans la même fenêtre de traduction) ».

Concrètement, à chaque mot anglais vont être associés les mots français situés dans la même fenêtre de traduction. En réalisant ce processus pour l'ensemble des phrases du jeu de données, il est possible de générer un dictionnaire avec d'un côté les mots anglais et de l'autre les mots français correspondants avec leur occurrence.

Le schéma ci-après explicite le processus dans le cas d'une traduction très simple.

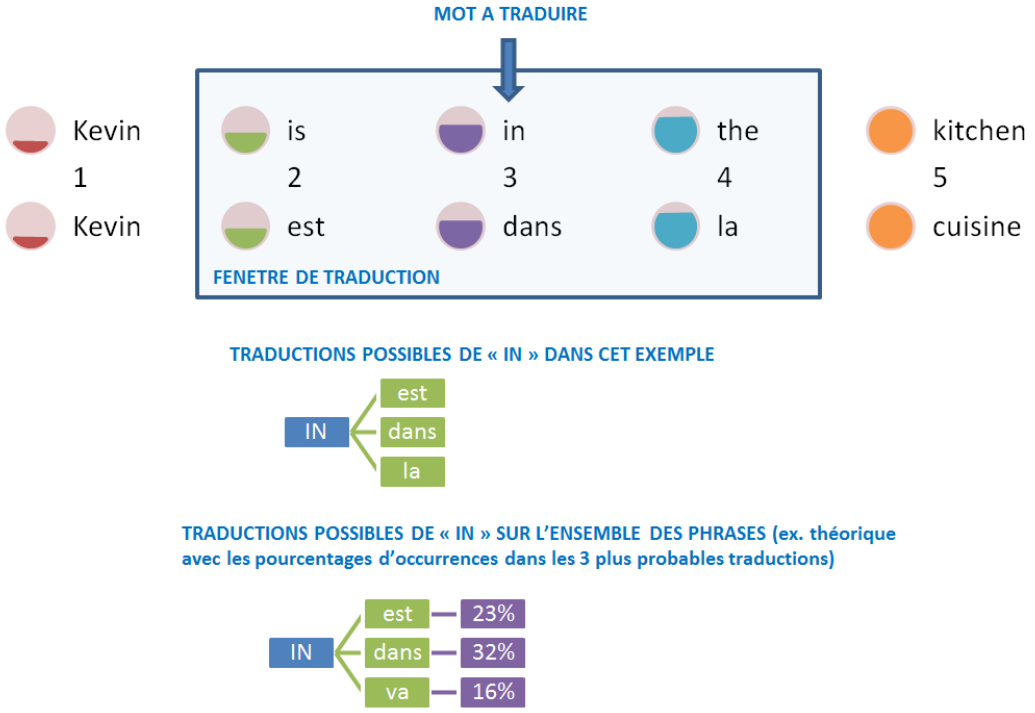


Figure 6: Méthode de traduction mot à mot par utilisation d'une fenêtre de traduction

Cette hypothèse naïve, bien qu'intéressante, ne semble pas vraiment plausible. En effet, dans l'étape de visualisation (et contrairement à l'exemple simpliste ci-dessus), nous avons constaté que le nombre de mots et la structure des phrases sont très différents en français et en anglais.

Nous avons toutefois procédé à l'élaboration de cette méthode, son évaluation puis testé diverses pistes d'amélioration.

3.2. Premiers résultats

A l'issue de l'application de la méthode sur l'ensemble du jeu de données, nous avons obtenu un dictionnaire contenant en clé : 196 mots anglais différents et en valeur : le mot le plus fréquemment rencontré comme possible traduction ainsi que cette occurrence en pourcentage.

L'image ci-après en présente un extrait :

```
'been': ['été', 31.48],
'between': ['entre', 24.07],
'big': ['grande', 13.48],
'bird': ['oiseau', 33.33],
'birds': ['oiseaux', 31.25],
'black': ['voiture', 19.07],
'blue': ['voiture', 17.11],
'busy': ['occupé', 24.42],
'but': ['mais', 26.55],
'california': ['est', 31.37],
'car': ['voiture', 33.33],
'cat': ['chat', 33.33],
'cats': ['chats', 32.81],
'chilly': ['froid', 25.8],
'china': ['chine', 33.05],
'chinese': ['chinois', 18.59],
'cold': ['froid', 26.51],
'december': ['en', 33.2],
'did': ['pourquoi', 15.15],
'difficult': ['difficile', 27.95],
'dislike': ["n'", 25.6],
```

Figure 7: Premier dictionnaire de traduction obtenu par la méthode de mot à mot

En parcourant visuellement la liste des 196 mots traduits, nous constatons plusieurs points :

- Certaines traductions sont manifestement erronées ;
- Les mots français les plus fréquents choisis pour la traduction des mots anglais ont une occurrence comprise entre 13 % et 33 %.

3.3. Mode d'évaluation

Pour évaluer la performance de ces traductions mot à mot, nous avons utilisé le dictionnaire de référence réalisé par Facebook (cf. bibliographie en annexe). Ce dictionnaire est très riche (113 286 mots) dont 174 mots sur les 196 mots anglais différents dont nous avions besoin (soit environ 90%).

Toutefois, nous avons constaté des erreurs plus ou moins importantes :

- Traduction d'un mot par un terme du même champ lexical (ex : cat par félin) ;
- Traduction d'un gérondif par un nom (ex : canoying par canoe) ;
- Traduction d'un mot par ce même mot en anglais (ex : bird par bird), ou bien une fois par un nom et une fois par un adjectif.

Pour nous affranchir de ces biais, nous avons construit un dictionnaire associant à un mot anglais de multiples traductions françaises (via l'utilisation d'une liste), ce qui permet d'obtenir plusieurs choix de traduction pour un même mot.

3.4. Résultats d'évaluation et pistes d'amélioration

En appliquant la méthode d'évaluation présentée, nous obtenons le résultat suivant :

- Les traductions proposées dans le 1er dictionnaire correspondent à celles proposées par le dictionnaire de référence pour 82 mots, soit dans **47 %** des cas.

Ce résultat n'étant pas satisfaisant, nous avons envisagé les pistes d'amélioration suivantes :

Tableau 2: Scoring obtenus en fonction des différentes pistes suivies

Pistes d'amélioration	Observations ou résultats	Conservation de la modification
Augmentation de la fenêtre de traduction	La bonne traduction de chaque mot anglais est quasiment toujours dans la fenêtre de 3 mots sélectionnée dans la version française (seuls 14 mots se trouvent en-dehors de cette fenêtre de 3 mots).	Non
Retrait des mots les plus usuels	Augmentation du taux de bonnes traductions de 47 à 69% (correspondance de 99 mots).	Oui
Ajout d'un seuil d'occurrence fixe de 25% à partir duquel le mot est considéré traduit	Augmentation du taux de bonnes traductions de 47% à 73% mais diminution du nombre de mots total traduit.	Non

3.5. Conclusions

Le taux de réussite est très moyen en dépit des pistes d'amélioration étudiées (pré-étude sur l'influence de la taille de la fenêtre, retrait de certains mots en français, travail sur le seuil d'occurrence).

Cette méthode est peu valeureuse en dépit du faible nombre de mots uniques, du grand nombre d'occurrences de ceux-ci et du fait que les phrases françaises étaient parfois un peu bancales (structurellement plus proches de l'anglais que du français), par exemple : "votre moins aimé fruit est le raisin".

En conclusion, cette méthode est trop fastidieuse, peu fiable et ne peut pas être utilisée à plus grande échelle et poursuivie.

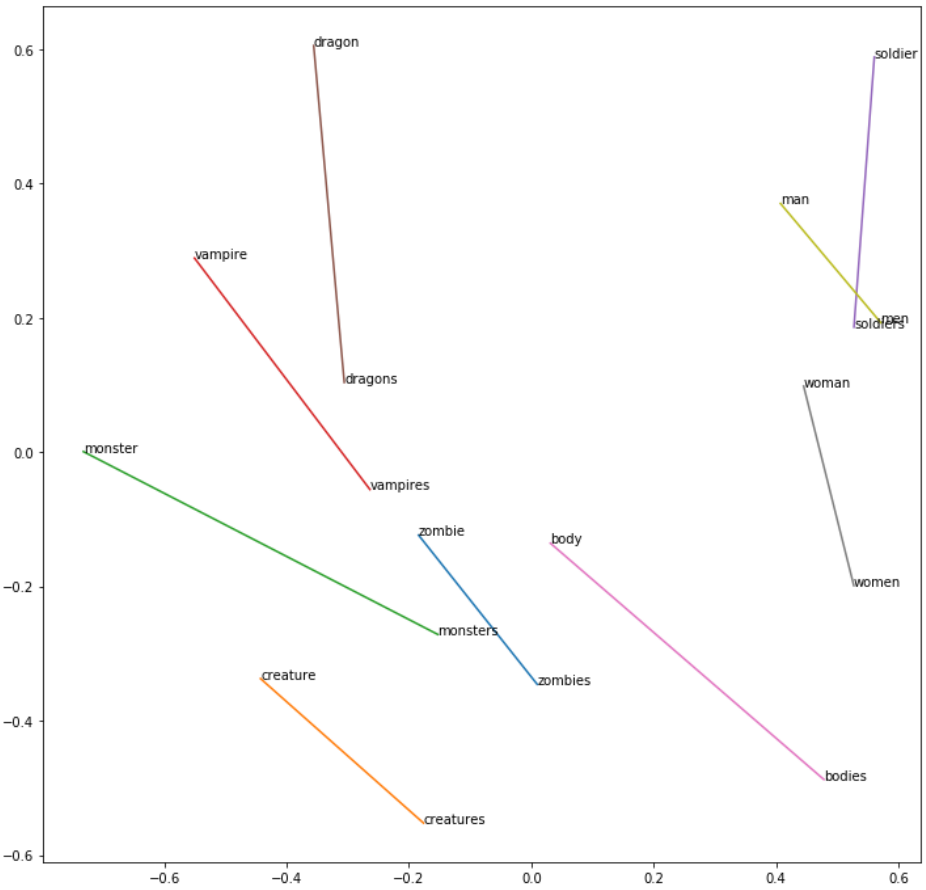
Nous rejetons l'hypothèse H0.

4. Modélisation 2 : Word Embedding

4.1. Principe de la méthode

La méthode de traduction par Word Embedding repose également sur une traduction mot à mot par l'utilisation d'une représentation vectorielle des mots. La représentation vectorielle permet de conserver la signification des mots et leur relation avec d'autres mots.

Figure 8: Visualisation par ACP (Analyse en composantes principales) de mots vectorisés



Nous avons récupéré des matrices d'embedding pré-entrainées en anglais d'une part et en français d'autre part. En fusionnant ces données avec notre dictionnaire de travail (dictionnaire Facebook précédemment évoqué), nous avons généré un corpus de mots identiques dans les deux langues.

Le principe de la traduction consiste en l'identification d'une correspondance entre l'espace vectoriel des mots sources (anglais) et l'espace vectoriel des mots cibles (français). Mathématiquement, cette correspondance se traduit par une matrice de transformation (translation, rotation, etc.).

Trouver la matrice de correspondance W revient à résoudre le problème d'optimisation suivant :

$$\underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WU_1 - U_2\|^2$$

- $O_d(\mathbb{R})$ est l'ensemble des matrices orthogonales.
- *U1* et *U2* sont respectivement les matrices contenant les embeddings des mots présents dans la langue source (anglais) et dans la langue cible (français).

Pour ce faire, il était nécessaire de connaître la traduction de certains mots et nous avons utilisé les mots transparents contenus dans les deux vocabulaires. Ces mots sont aussi appelés "anchors" (ancres) puisqu'ils permettent de faire la liaison entre les deux espaces vectoriels.

Grâce à la matrice obtenue, chaque mot anglais du corpus peut être traduit en français en trouvant les plus proches voisins de la projection.

4.2. Mode d'évaluation

Pour effectuer l'évaluation des résultats, nous avons utilisé le « dictionnaire Facebook ». Concrètement, nous avons vérifié la similarité entre les traductions proposées par notre modèle et celles proposées dans le jeu de données Facebook.

4.3. Résultats d'évaluation et pistes d'amélioration

Nous avons testé notre modèle et obtenu des scores de 47 à 60% selon le nombre de plus proches voisins considérés (jusqu'à cinq voisins).

4.4. Conclusions

Le Word Embedding seul ne permet pas d'obtenir de bons résultats pour une traduction. Cela s'apparente en effet à une technique de traduction mot à mot qui ne prend pas en compte les structures de phrases spécifiques à chaque langue : les mots sont traduits dans leur ordre d'apparition dans la langue originale (à traduire), ce qui peut différer de l'ordre attendu dans la langue de destination. A titre d'exemples, on peut notamment remarquer que :

- L'ordre de la paire adjectif-nom en anglais devrait être traduit dans la plupart des cas en nom-adjectif en français
- L'absence de certains articles-déterminants en anglais quand ils sont indispensables en français.

En conclusion, ce modèle, bien que meilleur que le modèle statistique n'est pas suffisamment efficace. Nous allons donc nous diriger vers des modèles de traduction phrase par phrase.

5. Modélisation 3 : Seq2Seq

5.1. Principe de la méthode Seq2Seq

Les modèles de Deep Learning Seq2Seq (séquences à séquences) sont basés sur un système d'encodeur-décodeur permettant de prendre en compte l'intégralité des phrases et donc de reconstituer les dépendances qui existent en termes de grammaire, syntaxe… dans les phrases traduites.

Cela se fait en deux grandes étapes :

ENCODAGE : Lecture de la phrase originale puis transformation dans un vecteur sens	DECODAGE: Traduction des phrases à partir des outputs de l'encodeur et des traductions déjà existantes dans le jeu de données :
• La vectorisation de la phrase se fait dans la couche embedding • Une ou plusieurs couches de réseau de neurones récurrents (RNN), particulièrement adaptées à l'apprentissage sur séquences, permettent le deep learning sur ces vecteurs.	• La couche d'embedding permet la vectorisation des phrases traduites du jeu de données • Ces vecteurs et les vecteurs en output de l'encodeur passent par une ou plusieurs couches RNN dans le décodeur • La couche de projection, une simple couche dense, permet ensuite d'associer l'output de la couche RNN à l'espace vocabulaire dans la langue de destination.

Figure 9: Explications du modèle de Seq2Seq

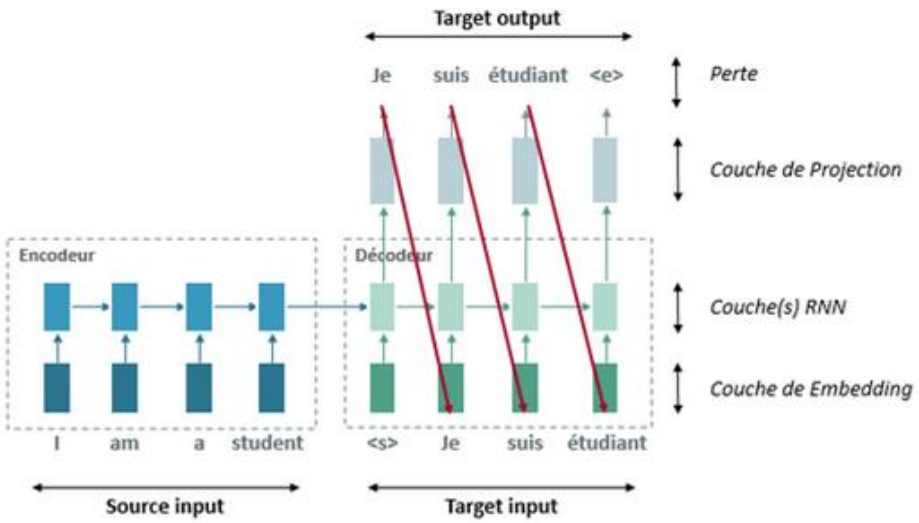


Figure 10: Modèle Seq2Seq « simple »

5.2. Principe des mécanismes d'attention

Au simple modèle Seq2Seq peuvent être ajoutés des mécanismes d'attention permettant d'améliorer les performances du modèle basique.

Développés par Bahdanau et al. En 2014, puis par Luong et al. En 2015, ces mécanismes permettent de pallier les difficultés rencontrées par le modèle Seq2Seq sur les phrases plus longues et plus complexes en poussant le modèle à « prêter attention » aux mots les plus importants de la phrase originale pendant la traduction : le modèle fait ainsi des liaisons directes entre la phrase originale et sa traduction (input du décodeur) dans les couches RNN du décodeur.

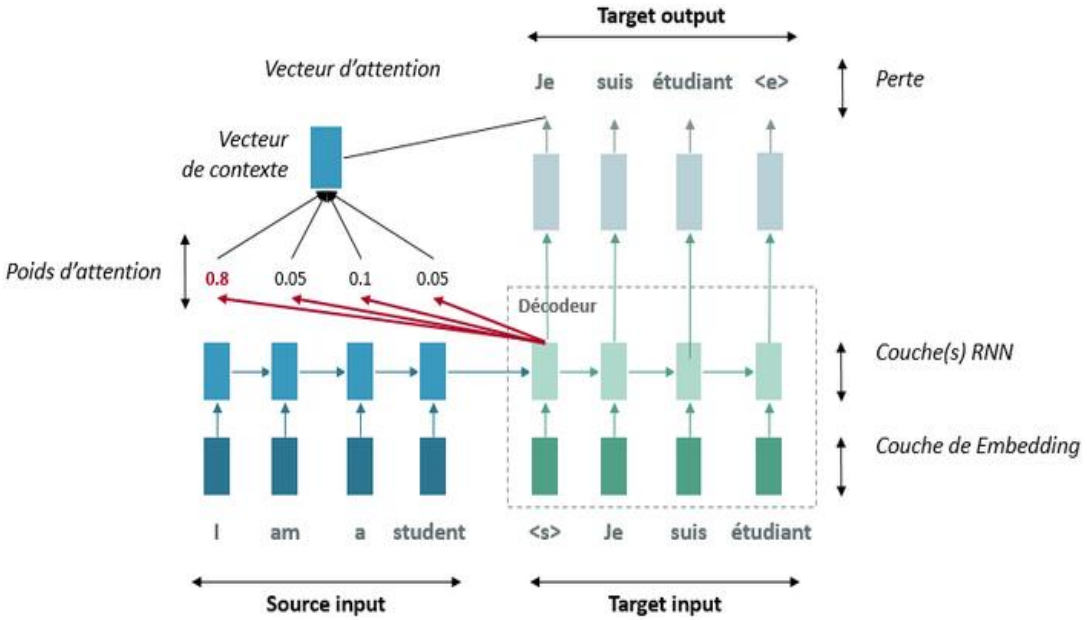


Figure 11: Modèle Seq2Seq avec mécanisme d'attention

Les mécanismes d'attention s'intègrent au niveau du décodeur en créant deux vecteurs consécutifs :

- **Un vecteur d'alignement** qui associe à chaque mot de la phrase originale la probabilité (ou score) que le mot traduit corresponde à celui de la phrase originale : on parle alors de poids d'attention.
- **Un vecteur de contexte** s'obtient par produit scalaire du vecteur d'alignement et de l'output de l'encodeur. C'est ce vecteur qui sert d'input dans les couches RNN du décodeur.

5.3. Sélection du modèle

Nous avons implémenté un modèle Seq2Seq avec mécanisme d'attention. Pour cela, nous avons choisi les caractéristiques du modèle, notamment au niveau du type et du nombre de couches RNN et le type de mécanismes d'attention retenus :

- Pour les couches RNN, le choix s'est fait principalement entre une couche simple, une LSTM ou une GRU.
- Pour le mécanisme d'attention, le choix s'est fait principalement entre Bahdanau et Luong.

Au vu des phrases du dataset utilisé (phrases simples, répétitives, peu de vocabulaire), nous avons implémenté un seul modèle Seq2Seq utilisant les caractéristiques suivantes :

		Caractéristiques
Dataset utilisé		Dataset n°1
Encodeur	Nbre de couches RNN	1
	Type de couches RNN	GRU
Décodeur	Mécanismes d'attention	Bahdanau
	Nbre de couches RNN	1
	Type de couches RNN	GRU

5.4. Modalités d'entrainement et de validation

Après un split du jeu de données en jeu d'entrainement (80%) et de test (20%) et prétraitement de ces données (nettoyage des chaines de caractères puis tokenisation), l'entrainement du modèle s'est fait sur le jeu d'entrainement pendant 10 époques avec la fonction de perte CategorialCrossEntropy et l'optimiseur Adam.

Une fonction traduction basée sur le choix du mot le plus probable en traduction du mot original (approche Greedy) est ensuite appliquée au jeu de test pour évaluer le score du modèle. Pour cela, on évalue les phrases sorties du modèle selon deux critères :

- Le nombre de mots dans les phrases sorties du modèle (y_pred) en comparaison avec le nombre de mots dans les traductions équivalentes du jeu de données (y_test)
- Le stemming des mots hors mots stop_words en obtenant le taux des racines y_pred sur celles de y_test
- ➢ La moyenne des deux critères constituera le score de ce modèle.

Cette méthode de calcul du score s'explique par le fait que la comparaison des phrases y_pred et y_test est compliquée par le manque de contexte : dans le cas où l'une des phrases est traduite au féminin et l'autre au masculin, il est possible que les deux solutions soient admissibles, sans qu'elles ne soient totalement identiques.

Ce score permet de juger la similarité dans la construction des phrases et dans leur sens global.

5.5. Résultats

On remarque que la majorité des phrases y_pred ne présente aucune différence dans leur nombre de mots avec les phrases y_test. Globalement, les scores en termes de différence de mots et de racines de mots sont très bons : le score global du modèle s'élève à 95 %.

Cependant, ce modèle est surentrainé : il est constitué de près de 130 000 phrases de structure simple et très similaire pour un vocabulaire d'environ 300 mots. Lorsque l'on essaie d'autres phrases (même très simples) hors du dataset, les résultats sont très mauvais.

Exemples : "united states is rainy" : "il est pluvieux en espagnol"
"the united states is rainy" : "il n'aime pas les raisons".

Par ailleurs, il présente :

- Quelques erreurs de traductions manifestes :
  - Phrase anglaise : china is mild during september .
  - Phrase traduite : chine est doux au mois de septembre.
  - Sortie du modèle : ils n aimons les etats unis est doux au mois de septembre, les mangues et en chine est doux au mois de septembre .
- Répétitions de pattern, qui peut être dues à l'utilisation de l'approche « greedy » :
  - Phrase traduite : il veut aller en inde
  - Sortie du modèle : il est alle en inde en inde en inde [….] en inde

Le modèle semble plus performant que dans les itérations précédentes. Il est toutefois difficile d'estimer si le modèle serait réellement performant avec un jeu d'entraînement plus diversifié.

Tableau 3 : Scoring du Seq2Seq

Phrase_a_traduire	Phrase_traduite	Sortie_modele	nb_words_cible	nb_words_mod	Différence
<start> the grapefruit is my favorite fruit ,...	le pamplemousse est mon fruit prefere , mais...	le pamplemousse est mon fruit prefere , mais l...	14	14	0
<start> china is usually hot during summer , ...	chine est generalement chaud pendant l ete ,...	chine est generalement chaud pendant l ete , e...	16	16	0
<start> your favorite fruit is the lemon , bu...	votre fruit prefere est le citron , mais leu...	votre fruit prefere est le citron , mais leur ...	14	14	0
<start> her favorite fruit is the strawberry ...	son fruit prefere est la fraise , mais votre...	son fruit prefere est la fraise , mais votre f...	14	14	0
<start> the united states is usually mild dur...	les etats unis est generalement doux pendant...	les etats unis est generalement doux pendant l...	18	18	0

racine_cible	racine_mod	score_racine	score_diff	score_tot
["pamplemouss", "fruit", "prefer", "orang", "f...	["pamplemouss", "fruit", "prefer", "orang", "f...	1.000000	1.0	1.000000
["chin", "general", "chaud", "pend", "jam", "a...	["chin", "general", "chaud", "pend", "jam", "a...	1.000000	1.0	1.000000
["fruit", "prefer", "citron", "prefer", "raisin"]	["fruit", "prefer", "citron", "favor", "raisin"]	0.800000	1.0	0.900000
["fruit", "prefer", "frais", "favor", "banan"]	["fruit", "prefer", "frais", "favor", "banan"]	1.000000	1.0	1.000000
["etat", "unis", "general", "doux", "pend", "h...	["etat", "unis", "general", "doux", "pend", "h...	0.888889	1.0	0.944444

5.6. Conclusions

Ce modèle, bien que meilleur que les précédents, nécessite des améliorations :

- Essai sur un dataset plus diversifié
- Changement d'approche : passage de l'approche Greedy à l'approche beam-search
- Essai avec un transformer

6. Modélisation 4 Beam Search et Décodeur

Dans cette itération, afin d'éviter le sur-apprentissage des données, nous reprenons le modèle Seq2Seq que nous appliquerons sur un ensemble de données plus important et diversifié nommé « Grand ensemble de phrases ».

Ce modèle nous servira de référence pour la comparaison des modèles entraînés pour cette itération :

- Modèle Seq2Seq classique – approche Greedy
- Modèle Seq2Seq avec un beam-search decoder
- Modèle de Transformer

Par ailleurs, en plus du test de performance constituant notre méthode de scoring implémenté dans l'itération précédente, nous utiliserons des métriques de scoring largement diffusées et acceptées pour l'évaluation de la génération automatique de texte, notamment les travaux de traduction, à savoir le score BLEU et le score ROUGE.

6.1. Les nouveaux scores utilisés

Score BLEU

Le score BLEU (Bilingual Evaluation Understudy Score) est le score de référence pour évaluer les traductions émises par les systèmes de traductions automatiques (phrases candidates) par rapport à des phrases traduites dites de référence.

Ce score a été mis en place par les travaux de Kishore Papineni, et al. en 2002 dans "BLEU : a Method for Automatic Evaluation of Machine Translation".

Il permet de calculer un ratio des n-grams communs aux phrases candidates et de référence sur les n-grams présents dans les références (n-grams représentant des groupes de tokens de n tokens) sans distinction d'ordre, et ressort un score compris entre 0 et 1, un score de 1 constituant une correspondance parfaite.

En affectant des poids sur les n premiers niveaux de n-grams, on peut obtenir des scores individuels n-grams lorsque l'intégralité des poids est affectée au niveau d'un unique niveau n de n-grams, et des scores cumulatifs n-grams lorsque les poids sont affectés sur plusieurs niveaux n de n-grams (le score représente alors la moyenne géométrique des scores de chaque niveau). Par défaut, les poids de la fonction utilisée (sentence_bleu) sont de 0.25 sur chacun des niveaux de 1-gram à 4-gram.

Score ROUGE

Le score ROUGE (Recall-Oriented Understudy for Gisting Evaluation) s'utilise principalement dans les travaux de résumés automatique de texte mais également pour les traductions automatiques. Comme pour le score BLEU, le but est de comparer la traduction du modèle à une traduction de référence, et la méthode peut s'utiliser sur différents niveaux de n-grams.

Ce score permet alors d'obtenir trois métriques :

- Le rappel (recall) qui se calcule par le ratio du nombre de n-grams communs aux deux phrases sur le nombre de n-grams dans la phrase de référence. Cela permet de juger si la traduction du modèle contient bien l'ensemble des mots de la phrase de référence.

Cependant, cette métrique seule ne permettra pas de juger de la qualité de la traduction : en effet, le modèle peut ressortir des phrases très longues qui contiennent l'ensemble des mots de la référence sans que la traduction ne soit pertinente : la précision permet alors d'affiner la qualité de la traduction.

- La précision qui se calcule par le ratio du nombre de n-grams communs aux deux phrases sur le nombre de n-grams dans la traduction du modèle. Cela permet de juger de la pertinence des mots utilisés dans la traduction du modèle.
- Le f1_score qui correspond à la moyenne harmonique des deux métriques précédentes.

Comme pour le score BLEU, on applique un score ROUGE-1 (sur 1-gram). On considère le f1_score comme le score ROUGE du modèle.

Explications des n-grams avec un exemple, ici le score ROUGE 2-grams :

Les 2-grams représentent des groupes de 2 mots qui s'enchainent dans chaque phrase.

	Référence	Modèle
Phrase à comparer	the cat was under the bed	the cat was found under the bed
Liste des 2-grams	the cat, cat was, was under, under the, the bed	the cat, cat was, was found, found under, under the, the bed
Nbre de 2-grams en commun	4	4
Nbre total de 2-grams	5	6
Métrique ROUGE-2	Recall = 4/5	Precision = 4/6

6.2. Modèle Seq2Seq classique – approche Greedy

Comme énoncé précédemment, le modèle Seq2Seq a de nouveau été entraîné sur un jeu de données plus important

		Caractéristiques
Dataset utilisé		« Grand ensemble de phrases »
Encodeur	Nbre de couches RNN	1
	Type de couches RNN	GRU
Décodeur	Mécanismes d'attention	Bahdanau
	Nbre de couches RNN	1
	Type de couches RNN	GRU

Résultats

Le modèle Seq2Seq obtient les scores suivants :

- Test de performance : 85 %
  - Score BLEU : 76 %
  - Score ROUGE : 72 %

Tests sur les phrases précédemment traduites par le modèle Seq2Seq entraîné avec le jeu de données « petit ensemble de phrases » :

« The United States is rainy » : « les etats unis est pluvieux . »

« United States is rainy. » : » les etats unis est pluvieux . »

Nous remarquons que le score du test de performance a diminué mais reste bon. Cependant, le modèle se généralise mieux et est plus robuste que le précédent.

6.3. Modèle Seq2Seq avec un Beam Search decoder

Introduction

Bien que le score obtenu avec le modèle Seq2Seq soit satisfaisant, nous remarquons néanmoins que le modèle se généralise très mal et que la traduction de phrases non présentes dans le jeu de données est souvent fausse. Nous avons choisi de modifier le modèle RNN en remplaçant les cellules GRU par des cellules LSTM ainsi que la méthode de traduction en optant pour la méthode Beam Search Decoder.

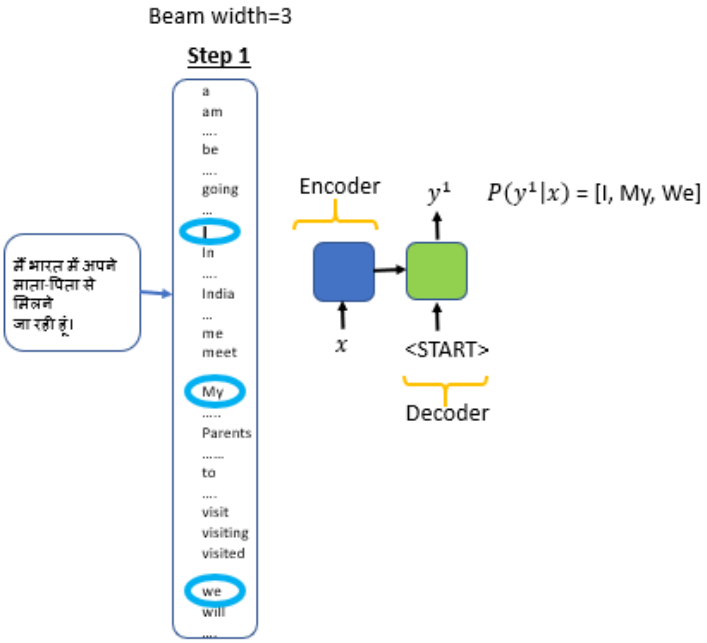
Principe du Beam Search Decoder

Lors de la modélisation précédente, nous avons appliqué la méthode dite Greedy pour obtenir la traduction d'une phrase. Cette méthode est la manière la plus basique de traduire une phrase à l'aide d'un RNN. Il s'agit tout simplement d'une traduction qui consiste à sélectionner, pour chaque mot de la phrase en entrée, le mot qui a la plus grande probabilité d'être la bonne traduction.
Le Beam Search Decoder fonctionne d'une manière différente. Son but va être de trouver la combinaison de mots ayant la plus grande probabilité d'être la traduction de la phrase en entrée.

Pour cela, nous allons indiquer à l'algorithme un entier comme paramètre k. Ensuite, pour le premier mot à traduire, il va mettre en mémoire les k mots ayant la plus grande probabilité d'être la bonne traduction. Jusque-là, cette méthode ne diffère pas beaucoup de la méthode Greedy. C'est à partir du deuxième mot que le Beam Search Decoder va commencer à fonctionner différemment. En effet, il ne prendra pas les k mots ayant la plus grande probabilité d'être la traduction du deuxième mot, mais il va créer une multitude de combinaisons (toutes les combinaisons possibles entre les k traductions du premier mot et toutes les traductions possibles du deuxième mot) et calculer une nouvelle probabilité de traduction pour chaque combinaison. Il va ensuite garder en mémoire les k meilleures combinaisons de mots et réitérer cette étape pour chaque mot de la phrase.

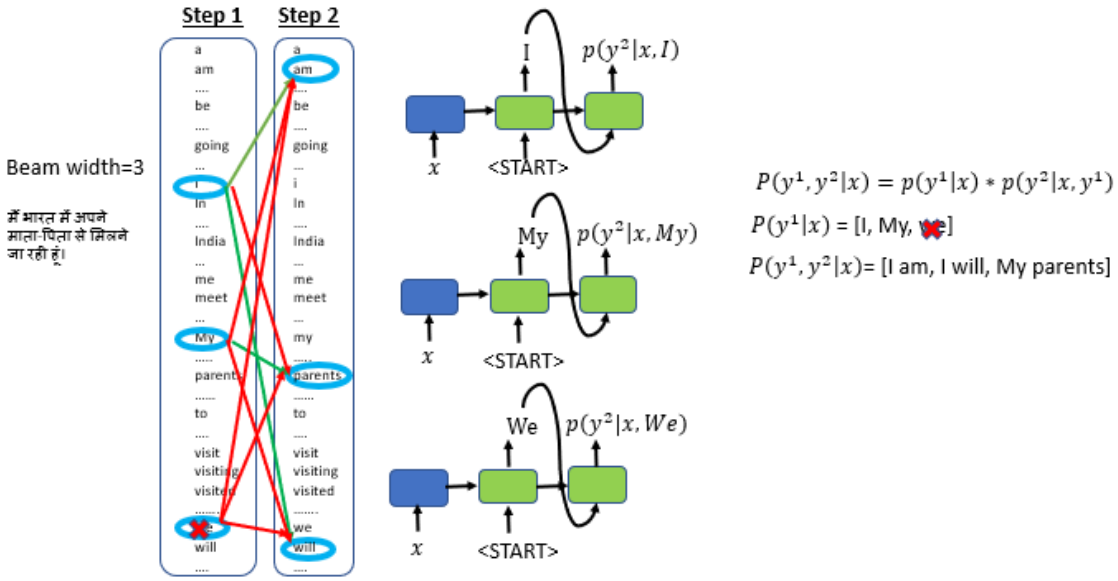
Voici 3 figures tirées de l'article « An intuitive explanation of Beam Search » de Khandelwal R., expliquant le fonctionnement du Beam Search Decoder en trois étapes pour un paramètre k = 3 :

Figure 12: Etape 1 : Sélection des 3 mots ayant la plus grande probabilité d'être les traductions du premier mot.



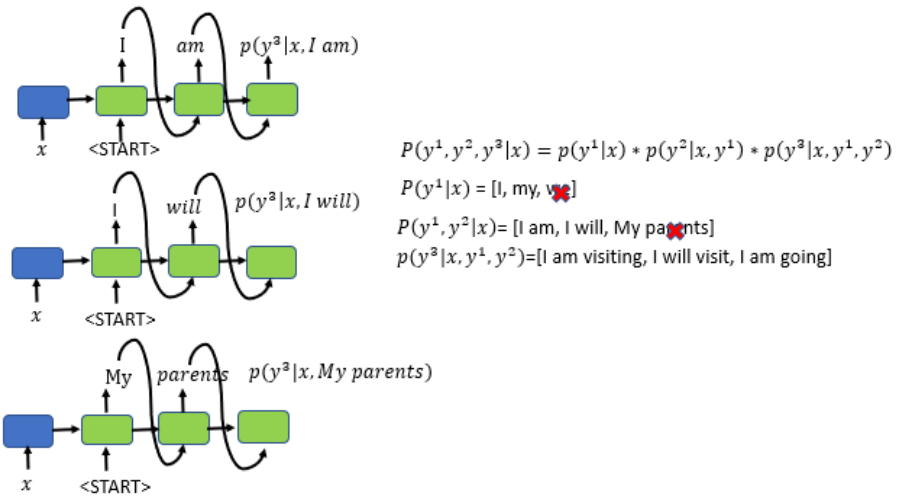
A ce stade, les 3 meilleures combinaisons sont donc [I, My, We].

Figure 13: Etape 2 : Sélection des 3 meilleures combinaisons entre le premier et deuxième mot



A ce stade, les 3 meilleures combinaisons sont donc [I am, I will, My parents]. Nous pouvons remarquer que celles-ci ne comprennent pas le mot 'We'. Il est donc supprimé des traductions possibles.

Figure 14: Etape 3 : Sélection des 3 meilleures combinaisons entre les 3 meilleures combinaisons précédentes et les traductions possibles pour le 3ème mot.



A ce stade, les 3 meilleures combinaisons sont donc [I am visiting, I will visit, I am going]. Nous remarquons ici que la combinaison 'My parents' mémorisée précédemment n'est pas présente dans les 3 meilleures combinaisons. Elle est donc également oubliée. L'algorithme va procéder ainsi jusqu'à ce qu'il considère que la phrase est finie. Il donnera donc en sortie les trois phrases ayant le plus de chance d'être la traduction finale de la phrase donnée en entrée.

L'avantage de cette méthode repose dans le fait que chaque traduction de mot n'est pas choisie indépendamment de tous les autres mots, mais c'est bien la combinaison de tous les mots qui est prise en compte. Cela peut nous laisser penser que la cohérence des phrases traduites devrait être renforcée. C'est ce que nous allons vérifier en appliquant cette méthode afin de traduire les phrases de notre jeu de test en en comparant les résultats obtenus par la méthode Beam et la méthode Greedy.

Nous reprenons la base du modèle Seq2Seq précédent que nous modifions. Nous obtenons finalement ce modèle :

		Caractéristiques
Dataset utilisé		Dataset n°3
Encodeur	Nbre de couches RNN	1
	Type de couches RNN	LSTM
Décodeur	Mécanismes d'attention	Luong
	Nbre de couches RNN	1
	Type de couches RNN	LSTM

Nous effectuerons le même prétraitement des données que pour le modèle Seq2Seq et nous entrainerons le modèle pendant 10 époques, avec la fonction de perte CategoricalCrossEntropy et l'optimiseur Adam.

Résultats

Le modèle Seq2Seq avec Beam Search obtient les scores suivants :

- Test de performance : 87 %
  - Score BLEU : 78 %
  - Score ROUGE : 80 %

	Phrase_a_traduire	Phrase_traduite	Sortie_modele	score_tot	Score_bleu	Rouge_f1_score
0	<start> i love fried bananas . <end>	j adore les bananes frites .	j adore les bananes frites .	1.000000	1.000000	1.000000
1	<start> i want to be alone for a while . <end>	je veux etre seule un moment .	je veux etre seul un moment .	1.000000	0.857143	0.857143
2	<start> i could never do that sort of thing	ce genre de chose ne m etait encore jamais a...	je ne pourrais jamais faire ce type de chose .	0.727273	0.542902	0.571429
3	<start> i was offered the choice of tea or co...	on m a propose le choix entre un the et un c...	on m a propose le choix entre un cafe .	0.769231	0.740818	0.909091
4	<start> are you at home ? <end>	tu es chez toi ?	etes vous chez vous ?	0.700000	0.400000	0.444444
5	<start> my friend helped me . <end>	mon amie m aida .	mon ami m a aidee .	0.800000	0.500000	0.545455
6	<start> i can t find tom anywhere . <end>	je ne trouve tom nulle part .	je ne peux trouver tom nulle part .	0.928571	0.750000	0.800000
7	<start> you re bad . <end>	tu es vilain .	tu es vilain .	1.000000	1.000000	1.000000
8	<start> i d like to eat something . <end>	je voudrais manger quelque chose .	j aimerais manger quelque chose .	0.833333	0.666667	0.666667
9	<start> i picked the lock . <end>	j ai crochete la serrure .	j ai crochete la serrure .	1.000000	1.000000	1.000000

Nous remarquons une amélioration des résultats par rapport au modèle précédent. Toutefois comme le modèle est différent (LSTM à la place de GRU et mécanisme d'attention différent), nous ne sommes pas en mesure d'affirmer si l'amélioration des résultats est due à la modification du modèle ou à l'application du Beam Search Decoder.

6.4. Modèle de transformer

Principe d'un transformer

La méthode de Transformer est une technique plus évoluée pour traiter des données séquentielles, son avantage majeur résidant dans le fait de n'utiliser que des mécanismes d'attention, sans couches récurrentes (trop longues à entraîner). Les mécanismes d'attention sont précieux en traitement du langage car permettent aux réseaux de neurones d'identifier les mots les plus importants.

Bloc Encodeur

Le bloc Encodeur récupère l'information dans la phrase à traduire. Il est ainsi composé :

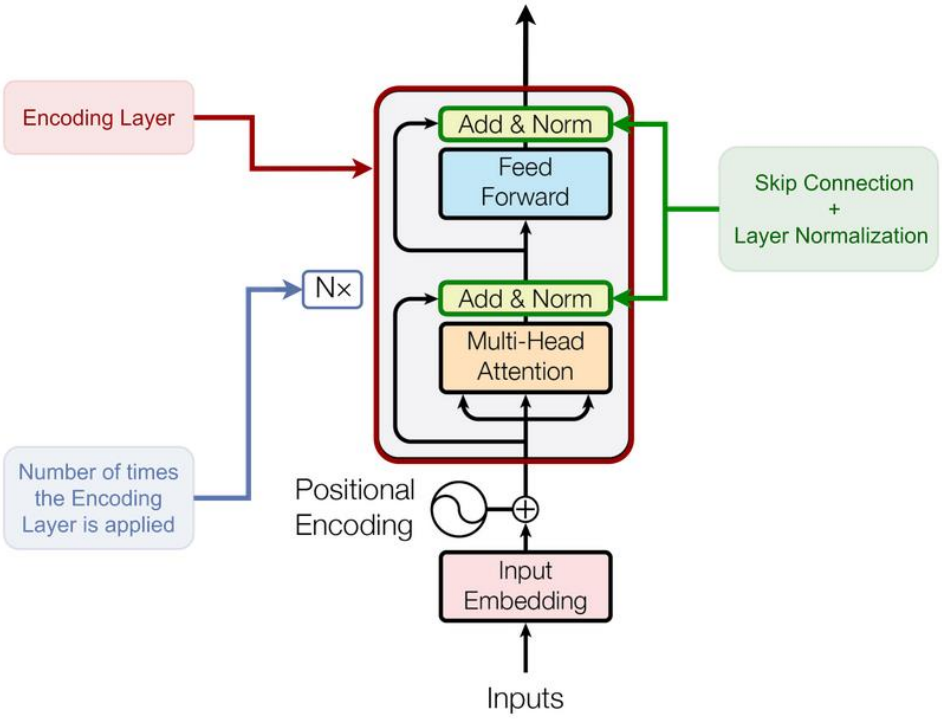


Figure 15: Schéma simplifié du bloc Encodeur

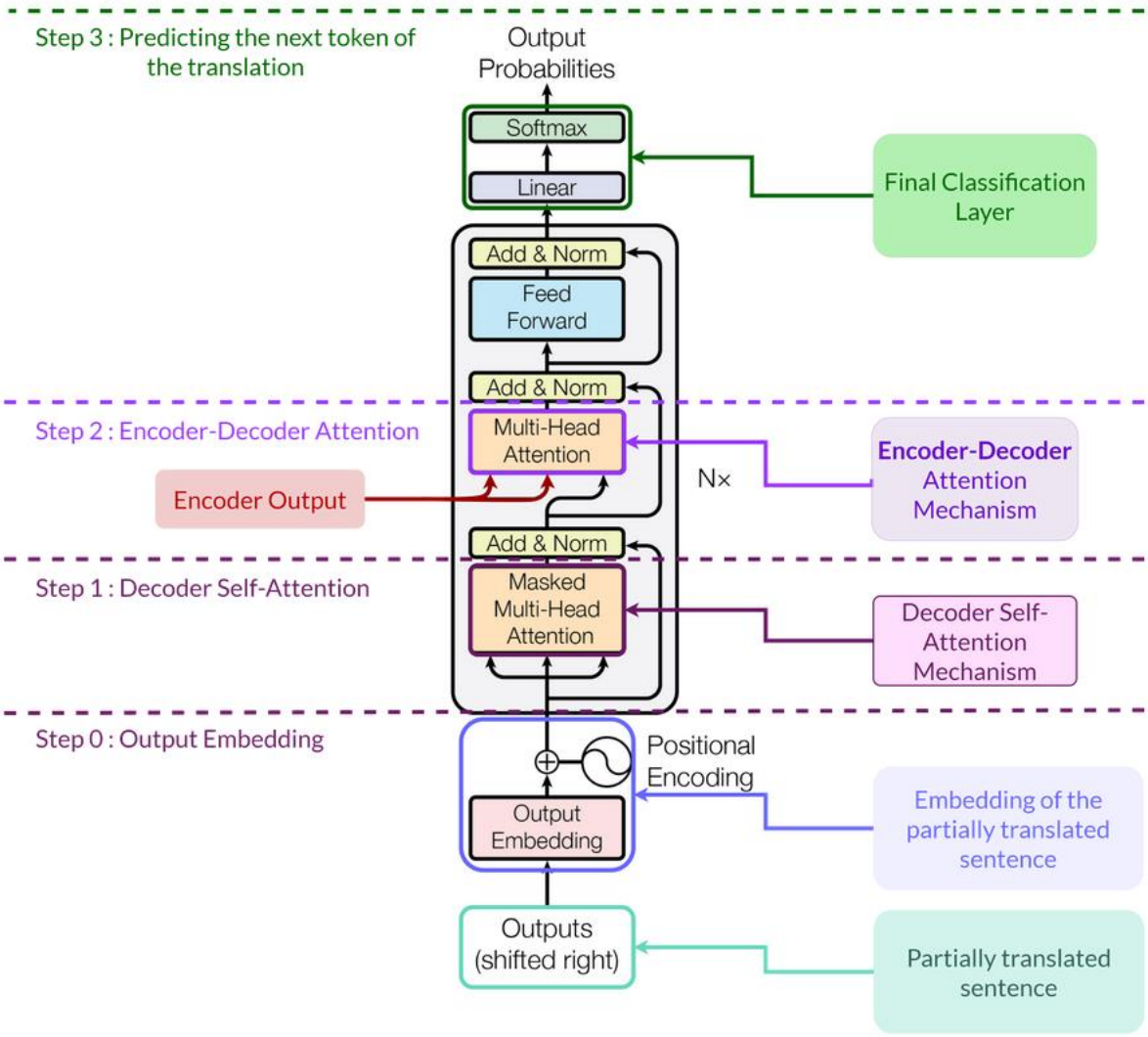
Les phrases à traduire sont tout d'abord encodées par une couche d'embedding. Contrairement au modèle Seq2Seq, le transformer traduit l'ensemble des mots en parallèle. Il faut donc rajouter à l'embedding le codage de la position des mots dans la phrase.

La couche d'encodage contient les mécanismes suivants (cf. annexe pour schéma total) :

- L'attention multi-têtes lance plusieurs mécanismes en parallèle s'attachant à des dépendances différentes entre les mots (long/court terme, grammaticale, syntaxique) ;
- Des couches de Dropout et Layer Normalisation vont servir à régulariser le modèle ;
- Des Skip Connections sont rajoutées pour éviter le problème de vanishing gradient à cause des petits gradients de la fonction softmax ;
- Un réseau dense (Feed Forward Neural Network).

Bloc Décodeur

Le bloc Décodeur constitue le 2ème volet de l'architecture Transformer, et est ainsi composé :



0 : L'entrée du Transformer contient à la fois la phrase à traduire et la traduction partielle de celle-ci (mode de fonctionnement **Autorégressif**). Cette entrée est transformée via une couche d'embedding et le codage de la position des mots de la phrase partiellement traduite.

1 : L'étape d'Auto-Attention permet au Décodeur de compiler les informations en sa possession (phrase partiellement traduite, encodage du mot à traduire) puis d'interagir avec l'Encodeur grâce à un vecteur de requête.

2 : Dans le mécanisme d'attention Encodeur-Décodeur, l'Encodeur fournit des vecteurs avec des clés et valeurs permettant in fine de réaliser un vecteur produit.

3 : Une classification classique (avec couche dense et activation **Softmax**) est ensuite réalisée afin de déterminer le prochain mot traduit.

Entrainement du modèle

Génération de masques

Les masques sont indispensables au fonctionnement du Transformer car permettent d'orienter vers les endroits où les mécanismes d'attention doivent regarder. Le masque dit de « Look-ahead » permet de cacher une partie de la phrase traduite afin d'entrainer le modèle en situation réelle.

Paramètres

Les paramètres suivants ont été sélectionnés :

Paramètres	Sélection
Split du jeu de données	Entrainement (80%) ; test (20%)
Fonction de coût	Entropie croisée
Optimiseur	Learning rate dynamique personnalisé
Epochs	10

Performance du modèle

Le modèle a été testé de la même manière que le modèle Seq2Seq.

Résultats

Résultat du Transformer

	Phrase_a_traduire	Phrase_traduite	Sortie_modele	score_tot	Score_bleu	Rouge_f1_score
0	you are old enough to know better than to act ...	tu es assez âgée pour savoir qu'il ne faut pas...	tu es assez âgée pour savoir que de prendre co...	0.708333	0.498055	0.521739
1	all the money was spent on clothes	tout l'argent a été dépensé dans des vêtements	tout l'argent a été passé à des vêtements	0.875000	0.750000	0.750000
2	he got off the train	il descendit du train	il a éteint le train	0.625000	0.400000	0.444444
3	have you ever been run over	avez vous jamais été renversée	as tu jamais été au courant	0.600000	0.333333	0.363636
4	are you sure you can handle this	êtes vous sûrs de pouvoir gérer ceci	êtes vous sûres de pouvoir gérer ceci	1.000000	0.857143	0.857143

Le transformer obtient les scores suivants :

- Test de performance : 76 %
  - Score BLEU : 60 %
  - Score ROUGE : 64 %

Globalement, le transformer obtient un score d'environ 60 %, ce qui reste convenable au vu du temps d'entraînement (10 époques seulement) et du jeu de données (moins de 155000 phrases dans le corpus).

7. Bilan et suite du projet

7.1. Conclusions

Ce projet nous a permis de réaliser les actions suivantes :

- Choisir un jeu de données adéquat, à savoir le « Grand Ensemble de phrases » ;
- Sélectionner et personnaliser un modèle de NLP, à savoir le Seq2Seq
- Sélectionner une méthode de traduction, à savoir le Beam Search Decoder
- Déterminer une (ou des) métrique(s) visant à analyser et comparer les résultats, à savoir les scores dit BLEUS et ROUGES.

Il s'est déroulé sur 9 semaines, le diagramme de Gantt en annexe en reprend les principales tâches.

Nous n'avons cependant pas pu adapter le modèle au système de lunettes connectées. Pour ce faire, il aurait fallu travailler sur le traitement des sorties du modèles de traduction des lunettes. En effet, les transcriptions de l'oral vers l'écrit génèrent des phrases moins structurées et parfois bancales. Il aurait fallu corriger ces données et/ou entrainer le modèle sur de telles phrases.

Toutefois, ce projet regroupe néanmoins un panel intéressant de modèles de Deep Learning utilisés pour le NLP à ce jour.

7.2. Challenges du projet

Les principales gageures de ce projet étaient :

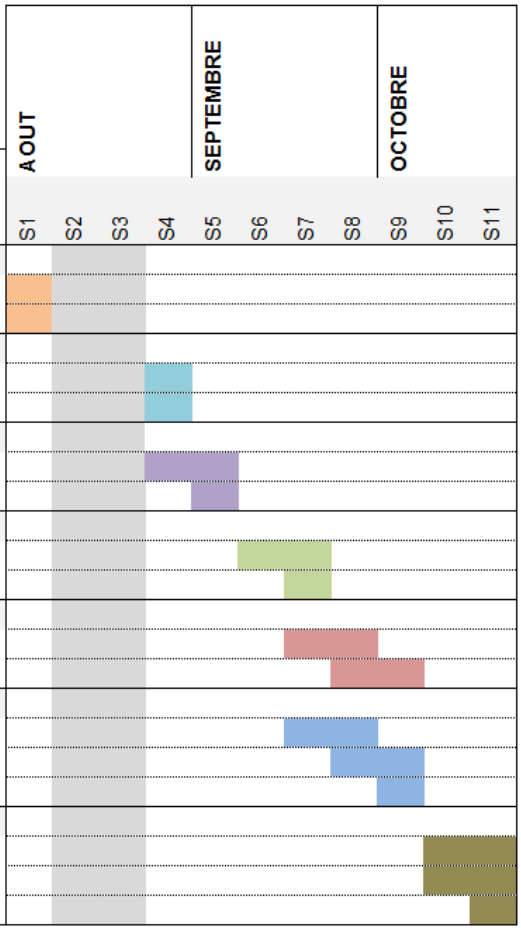
- Devancer le planning de formation afin de maîtriser au plus vite les sujets de Deep Learning ;
- Réfléchir aux paramètres du modèle avant de lancer son exécution
  En effet les temps d'entrainement des modèles sur le jeu de données « grand ensemble de phrases » est extrêmement long :
  - Entrainement du modèle Seq2Seq classique : 3h
  - Entrainement du modèle Seq2Seq Beam : 6h
  - Entrainement du Transformer : 3h
- Posséder de relativement larges puissances de stockages et de puissances computationnelles.
- Repérer les erreurs dans des codes très longs, imbriqués et les uns dans les autres et sans pouvoir faire de micro-tests aisément.
- Difficultés à implémenter certains morceaux de codes (issus de recherches sur internet) dans nos modèles déjà complexes.

8. Annexes :

8.1. Diagramme de Gantt

Diagramme de Gantt

	Détail des tâches	Livrables
1	**Exploration des données**	**Rendu de visualisation**
1.1	Compréhension globale du sujet	
1.2	Exploration et visualisation des données	
2	**Modélisation statistique**	**Rendu de modélisation**
2.1	Modélisation statistique	
2.2	Création du GitHub	
3	**Modélisation Word Embedding**	**Rendu de modélisation**
3.1	Apprentissage nécessaire	
3.2	Modélisation Word Embedding	
4	**Modélisation Seq2Seq**	**Rendu de modélisation**
4.1	Apprentissage nécessaire	
4.2	Modélisation Seq2Seq	
5	**Modélisation Transformer + BeamSearch**	**Rendu de modélisation**
5.1	Apprentissage nécessaire	
5.2	Modélisation Transformer + BeamSearch	
6	**Rédaction des rapports**	**Rapport de modélisation / final**
6.1	Rapport de modélisation	
6.2	Rapport final	
6.3	Nettoyage des codes / GitHub	
7	**Préparation soutenance**	**Streamlit + Powerpoint**
7.1	Réalisation du streamlit	
7.2	Réalisation du powerpoint	
7.3	Préparation discours	



8.2. Annexe : Bibliographie

Data

- « Petit ensemble de phrases » :
  https://github.com/susanli2016/NLP-with-Python/tree/master/data
  small_vocab_fr, small_vocab_en
- « Matrices pré-entrainées de Word embedding » :
  https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.fr.300.vec.gz
  https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.vec.gz
- « Dictionnaire Facebook » :
  https://github.com/facebookresearch/MUSE
- « Grand ensemble de phrases » :
  https://github.com/SamLynnEvans/Transformer/tree/master/data

Bibliographie

- Traduction mot à mot:
  https://datascientest.com/nlp-word-translation
  https://docs.google.com/document/d/1ldPQgScSAyaY0Fj2cChuCXmQpl5NulRlrco1VK5gxl8/edit
- RNN et LSTM:
  https://towardsdatascience.com/using-rnns-for-machine-translation-11ddded78ddf
  https://datascientest.com/fonctionnement-des-reseaux-neurones
- Machine Translation:
  https://towardsdatascience.com/neural-machine-translation-15ecf6b0b
- Score BLEU
https://machinelearningmastery.com/calculate-bleu-score-for-text-python/
- Score ROUGE
https://pypi.org/project/rouge/
https://kavita-ganesan.com/what-is-rouge-and-how-it-works-for-evaluation-of-summaries/

- « An intuitive explanation of Beam Search », Khandelwal R.
  https://towardsdatascience.com/an-intuitive-explanation-of-beam-search-9b1d744e7a0f

- Tutoriel Beam Search :
  https://www.tensorflow.org/addons/tutorials/networks_seq2seq_nmt#setup

8.3. Annexe : Transformer

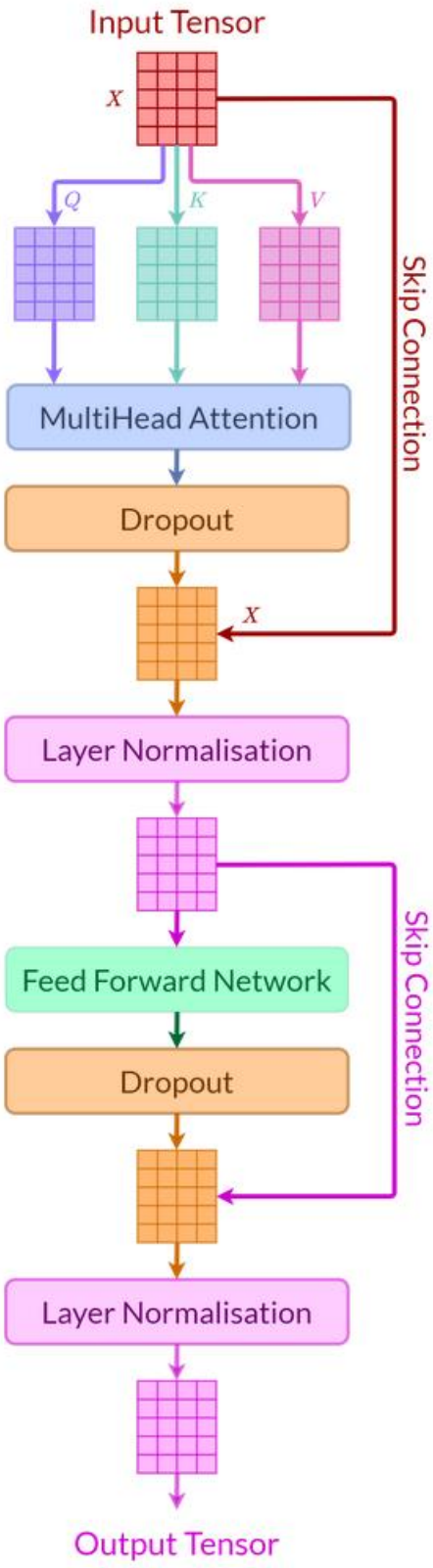


Figure 16: Schéma plus précis de l'encodeur