Rapport TATIA
Génération de texte par prédiction successive du prochain mot d'un texte à l'aide d'un réseau de neurones récurrents

Khaoula BOUHLAL et Hugo BULZOMI

1 Introduction

Générer du texte nous paraissait être un objectif amusant et l'opportunité de nous familiariser plus en profondeur avec certaines techniques évoquées pendant le cours ce semestre. Nous allons dans ce rapport détailler notre démarche dans la réalisation de ce projet en commençant par la méthode de génération choisie. Nous verrons ensuite comment nous avons mis en place de façon concrète la méthode de génération que nous présentons pour ensuite montrer les problèmes que nous avons rencontrés et enfin comparer nos résultats avec du texte généré par un autre modèle.

2 Méthode de génération de texte

Il existe à priori de nombreuses manière d'aborder ce problème et la méthode choisie va largement affecter les résultats finaux et le type travail à fournir.
L'une des méthodes classiques que nous avons pu trouver en nous documentant consiste à prédire le mot ou caractère suivant dans un texte à partir de ceux qui le précèdent. On peut alors, de prédiction en prédiction, générer un texte entier. Pour nous, les principaux avantages de cette méthode sont les suivants :

1. La taille du texte généré est variable et illimitée.
2. Différents modèles peuvent être employés pour créer la prédiction.
3. N'importe quel type de texte peut être utilisé (pas besoin d'un dataset très structuré).

Comme évoqué plus haut, la prédiction peut se faire à plusieurs niveaux au sein du texte : au niveau des caractères, des mots, des groupes de mots ect... Dans le cadre ce projet nous avons choisi de nous intéresser aux mots du texte.
En effet, nous avons évoqué en classe cette année les techniques utilisées pour convertir les mots en valeurs numériques afin de faciliter leur traitement dans les applications informatique. En particulier, les techniques de conversions de mots en vecteurs (ou word embedding) nous semblaient parfaitement pertinentes à utiliser ici et à motivé notre choix de faire des prédictions sur les mots.
Nous avons donc décidé de générer notre texte en prédisant le mot suivant à partir des mots précédents. Finalement, il nous restait à décider du modèle à employer pour faire nos prédictions. De notre point de vue, il y a un nombre infini de moyen possible de prédire le mot suivant à partir des mots précédents (il est probablement même possible d'accomplir cela sans utiliser de machine-learning). Après documentation plus en profondeur, nous avons opté pour un réseau de neurones récurrents (abrégé RNN). En effet il semblerait que les RNN soient très utilisés pour ce genre de tâche. Tirer des conclusions nécessite parfois de s'appuyer sur la connaissance des évènements précédents ; les RNN disposent d'un état interne qui leurs permettent de garder une trace de ces évènements et sont donc très utiles pour tous les problèmes dit de "reconnaissance de motifs temporels" (temporal pattern recognition).

En conclusion notre méthode de génération de texte peut se décomposer comme suit :

1. Rassembler un corpus de texte.

2. découper ce texte en mots et convertir ces mots en vecteurs grâce à l'une des techniques vues en classe.
3. Entraîner un RNN à prédire le mot suivant à partir des mots précédents (convertis en vecteurs au préalable).
4. Une fois le modèle entraîné, lui donner un court texte en entrée et lui faire générer mot par mot la suite du texte.

Pendant nos recherches, nous avons trouvé sur la documentation de Tensorflow un code exemple qui permet de créer un modèle de génération en faisant des prédictions sur les caractères d'un texte : https ://www.tensorflow.org/text/tutorials/text_generation
On pourra donc comparer le texte généré mot par mot par notre modèle avec le texte produit caractère par caractère par cet autre modèle.

3 Travail concret et détails techniques

3.1 Librairies et outils utilisés

Nous avons jugé Python être le langage le plus adapté pour ce projet. De nombreuses librairies permettent de manipuler des modèles de machine learning et plus spécifiquement des réseaux de neurones, et la gestion des chaînes de caractères et grandement facilitée dans ce langage.

Pour l'encodage des mots en vecteurs, nous aurions pu choisir d'analyser notre corpus de texte et de déduire les vecteurs associés à chaque mots grâce à leur contexte d'emplois. Néanmoins, des tables d'encodage pré-entraînées sur des corpus très larges existent déjà et nous permettent d'obtenir un encodage de bien meilleure qualité que ce que nous serions en mesure de produire nous-même. Nous avons donc décidé d'utiliser les représentations vectorielles de GloVe (plus spécifiquement la représentation en vecteurs de 50 dimensions obtenue après analyse des bases de données "Wikipedia 2014" et "Gigaword 5"). Nous n'avons pas inclus le fichier en question à https ://nlp.stanford.edu/projects/glove/

Les trois librairies les plus utilisées en Python pour la manipulation de réseaux de neurones sont Scikit-Learn, Pytorch et Tensorflow. La librairie Scikit-learn, bien que plus simple d'utilisation, ne permet par d'utiliser des RNN, le choix s'est donc fait entre Pytorch et Tensorflow. Ayant trouvé plus de ressources et d'informations pour cette dernière, nous avons finalement opté pour Tensorflow malgré sa réputation d'être un peu plus rigide d'utilisation que Pytorch.

3.2 Le dataset employé

Comme évoqué précédemment n'importe quel texte peut être utilisé pour la méthode de génération que nous avons choisie. Cela dit la sélection du corpus a tout de même été motivée par plusieurs aspects :
Tout d'abord un réseau de neurones, par rapport à d'autres modèles de machine learning a besoin d'une grande quantité de donnée pour s'entraîner. Nous devions donc tout d'abord trouver des textes assez longs.
De plus, pour tout modèle de machine learning l'homogénéité de notre corpus est importante. En effet nous aurions très bien pu sélectionner des textes très divers avec des vocabulaires et des structures différents, mais garder un dataset homogène est de façon générale une bonne chose pour obtenir de bons résultats.
Finalement, l'aspect le plus important pour notre corpus est de garder une orthographe correcte et consistante. En effet, nous avons déjà choisi d'encoder les mots comme des vecteurs. Nous allons donc à un moment ou à un autre, pour un mot de notre texte, rechercher son vecteur associé dans une table de conversion. Si l'orthographe des mots est inconsistante dans notre corpus, nous aurons donc beaucoup de mots pour lesquels nous n'arriverons pas à trouver d'entrée dans notre table de conversion. Tant que ces mots que nous n'arrivons pas à reconnaître restent peu nombreux, nous pouvons leur associer un vecteur par défaut (voir les corriger à la main), mais la qualité de nos prédictions va forcément diminuer si trop de mots inconnus sont trouvés dans notre corpus. Ce dernier point exclu donc les corpus constitués de tweets, de posts sur des forums ou de sms.

Il a finalement été décidé de rassembler, au moins dans un premier temps, un corpus d'horoscopes puisque ces derniers sont faciles à trouver, sont relativement homogènes et contiennent en général peu de fautes d'orthographe. Nous avons donc réunis un corpus d'horoscopes de 641 357 mots et 3 317 373 caractères dont voici quelques phrases exemples :
Wishing you were somewhere else is a recipe for a bad mood !
Be motivated to handle all manner of money issues in january , organize finances , and possibly see some improvement in income.
If you give yourself permission , you can trust your intuition about nearly anything .

Ce corpus est tiré de deux datasets téléchargés sur Kaggle :
https ://www.kaggle.com/shahp7575/horoscopes et https ://www.kaggle.com/thedivtagguy/times-of-india-horoscopes-dec-2019-to-june-2021 dont nous avons ensuite limité la taille pour accélérer l'entraînement.

Nous avons ensuite nettoyé notre corpus mettant toutes les lettres en minuscules et en listant les mots pour lesquels nous n'arrivons pas à trouver de représentation vectorielle dans notre fichier GloVe. Cette liste étant plutôt courte, nous avons trouvé nous-même des synonymes pour lesquels nous avons un vecteur associé. De plus, un caractère de séparation '$' a été ajouté entre les horoscopes. Cet ajout est parfaitement optionnel, mais puisque notre dataset est présenté comme long texte, un caractère de séparation peut permettre au modèle de différencier deux horoscopes différents.
Finalement le texte est découpé en séquences de 21 mots. Les entrées de notre réseau de neurones sont les 20 premiers mots de la séquence et la prédiction porte sur le 21ème. Ainsi notre dataset est prêt pour l'entraînement de notre modèle.
Il est à noter que nous avons gardé toute la ponctuation du texte puisque GloVe considère cette dernière comme des mots à part entière et nous donne donc accès à des vecteurs pour l'encoder.

3.3 Création du modèle avec Tensorflow

La structuration d'un réseau de neurones est une tâche assez compliquée pour nous, car il n'y a pas de règles précises à suivre pour choisir le nombre de couches, le nombre de neurones ou même la façon dont les couches sont connectées entre elles. Nous avons donc décidé d'utiliser une architecture simple, similaire à beaucoup d'autres modèles employés pour des problèmes similaires :

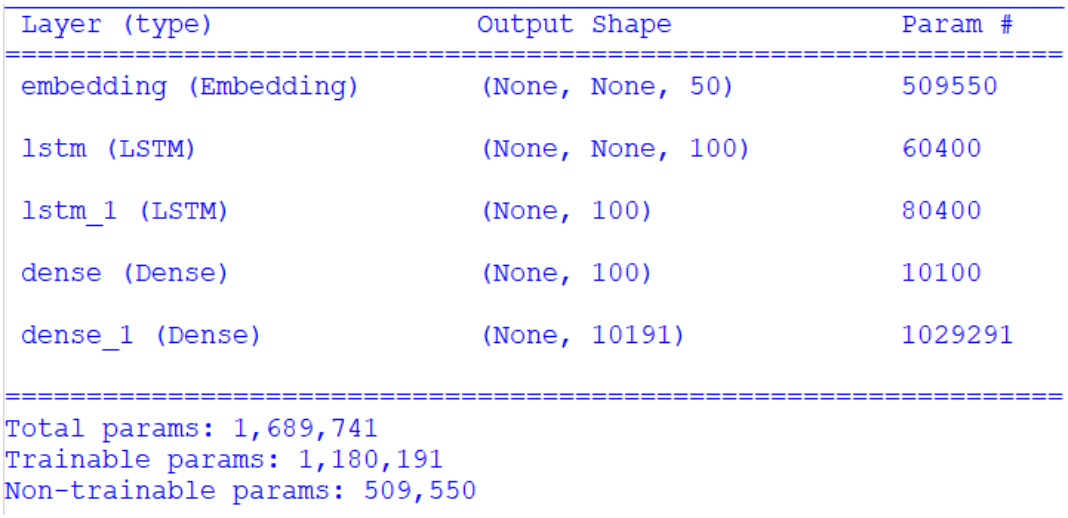
Pour chaque séquence de 21 mots de notre dataset, les 20 premiers seront donnés en entrée de notre modèle. La première couche les convertis en vecteurs en utilisant une matrice de conversion générée à partir des données d'un fichier GloVe. La couche de sortie dispose d'une fonction d'activation de type softmax qui va simplement retenir le neurone avec le plus haut taux d'activation de la couche. Les autres auront tous une valeur de sortie de zéro.

4 Problèmes rencontrés

Nous avons évidemment rencontré certaines difficultés pendant l'élaboration de ce projet.
N'ayant jamais utilisé Tensorflow avant, la relative rigidité de cette librairie nous a déjà demandé un travail d'apprentissage qui a rallongé le temps que nous avons mis à mettre en place tout ce que nous voulions. Mais au-delà de ces difficultés mineures, deux problèmes nous ont véritablement fait perdre du temps.

4.1 Le choix du du type de prédiction

Le modèle que nous avons présenté plus haut est en réalité notre second.
Comme énoncé précédemment, notre modèle prend en entrée des mots (identifiés par des entiers uniques), les convertis en vecteurs grâce à une matrice obtenue avec un fichier GloVe et enfin prédis le mot suivant sous forme d'un encodage one-hot. Le calcul de l'erreur revient alors à retrouver le mot



```
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 50)          509550

lstm (LSTM)                  (None, None, 100)         60400

lstm_1 (LSTM)                (None, 100)               80400

dense (Dense)                (None, 100)               10100

dense_1 (Dense)              (None, 10191)             1029291

=================================================================
Total params: 1,689,741
Trainable params: 1,180,191
Non-trainable params: 509,550
```

FIGURE 1 – Résumé du modèle généré par Tensorflow. On peut voir qu'une première couche sert à encoder en vecteur les mots donnés en entrée. Les paramètres de cette couche ne sont donc pas entraînables. Viennent ensuite deux couches récurrentes de type LSTM. On a choisi de manière arbitraire 100 neurones par couche. Une couche dense de taille égale aux couches récurrentes est ensuite placée entre ces dernières et une couche de sortie de taille égale au vocabulaire du dataset. Cette dernière couche nous donne la version one-hot encodée du mot prédit par le modèle.

prédis grâce à son encodage one-hot et à le comparer avec le mot attendu. Peut importe le mot prédis, s'il est différent du mot attendu alors l'erreur sera la même.
On remarque alors que dans ce modèle, les vecteurs GloVe ne sont réellement utilisés que dans les couches cachées du réseau de neurones. Leur propriété de "proximité sémantique entraîne une proximité vectorielle" n'est alors utile que lors de l'application de l'algorithme de back-propagation sur les couches internes.

Notre première idée était un peu différente. Plutôt que de prédire un mot one-hot encodé, nous pensions prédire directement un vecteur. De cette façon le modèle aurait été beaucoup plus transparent ; on convertit tout notre texte en vecteurs et on essaie de prédire le 21ème vecteur en fonction des 20 premiers. Cette approche nécessitait en revanche de calculer l'erreur du modèle en utilisant une fonction de similarité cosinus (cosine similarity) afin de calculer la différence entre le vecteur prédit et celui attendu. Notre motivation était d'avoir une erreur plus mince si le mot prédit avait une sémantique proche du vecteur attendu.
Nous nous sommes néanmoins rendu compte que certains mots avaient un vecteur associé très proche de la majorité des autres mots de notre dataset ! En particulier, le modèle atteignait une erreur moyenne très faible en ne prédisant tout le temps que le mot "but".

```
Cosine similarity between the average vector and the vector associated with 'but
':  0.9580295378102196
```

FIGURE 2 – Similarité calculée entre le vecteur moyen (somme des vecteurs des mots présents dans le texte, divisée par la taille du texte) et le vecteur associé au mot 'but'.

Si notre but n'avait pas été de générer du texte, nous aurions pu "nettoyer" notre dataset de toute conjonction, ponctuation et de façon générale, ne garder que les mots sémantiquements significatifs ; mais nous voulions générer du texte lisible par un être humain et tous ces mots était donc importants pour nous.

C'est cette raison qui nous a poussée à ajouter une couche dense servant à obtenir une prédiction one-hot encodée qui nous permet de transformer notre problème de régression en problème de classification (pour utiliser des termes propres au machine-learning). De cette façon l'erreur sera toujours la même tant que le mot prédit n'est pas le bon, mais l'utilisation en interne des vecteurs GloVe nous permet tout de même de profiter de leurs propriétés.

4.2 La sauvegarde de modèle Tensorflow

Un autre problème majeur que nous avons eu consisté bêtement à la sauvegarde et au chargement de notre modèle grâce à l'API Tensorflow.
Notre stratégie pour évaluer notre modèle au cours de son entraînement consistait simplement à le laisser s'entraîner pendant quelques heures, sauvegarder le modèle grâce à des méthodes Tensorflow, puis plus tard à charger le modèle sauvegardé pour lui faire générer du texte. Cette méthode nous donnait néanmoins des résultats désastreux. Au point où même après entraînement, les performances du modèle pour la génération de texte n'étaient pas si loin de celles du modèle non entraîné et ceux, malgré une bonne précision calculée pendant l'entraînement. Nous aurions pu suspecter un over-fitting sur le dataset si le modèle prédisait les bons mots sur des exemples qui lui avaient déjà été présentés, mais il n'en était rien. Au final, même avec une précision de 50% calculée pendant l'entraînement, celle que nous calculions ne dépassait pas 1% même sur des exemples du dataset de départ.

Après de nombreuses heures, nous nous sommes finalement rendu compte que la sauvegarde et le chargement de notre modèle ne fonctionnait pas correctement. C'est apparemment un problème connu de Tensorflow qui touche les réseaux récurrents et qui semble se résoudre par des manipulations assez diverses : réinstallation totale de Tensorflow voir de Python, utilisation de versions antérieures de certaines librairies... Finalement, après de nombreuses tentatives infructueuses, nous nous sommes résolus à entraîner et tester notre modèle dans le même script python sans le sauvegarder ou le charger entre temps. Le problème que cela entraîne, est que nous ne pouvons pas entraîner le modèle en plusieurs fois. Mais de façon plus problématique, cela signifie que nous devons ré-entraîner notre modèle à chaque fois que nous voulons générer du texte.

5 Présentation des résultats et analyse

5.1 Nos résultats

Les résultats présentés ci-dessous ont été obtenus après 56 epochs et environ 10h d'entraînement (un peu plus d'une nuit). Après ce temps, la précision du modèle sur notre dataset est de 41% ; le modèle prédit donc le mot correct dans 41% des cas. Afin de prédire du texte, nous donnons en entrée les 20 premiers mots d'une phrase que le modèle doit donc étendre. Étant donné que nous avons gardé la ponctuation dans notre dataset, nous nous attendons à ce que le modèle formule des phrases correctes. Ainsi à partir d'une simple phrase d'une vingtaine de mots, si nous faisons n prédictions successives que nous concaténons, nous devrions obtenir un texte entier.

Voici un exemple pour 40 mots générés avec la phrase donnée en entrée "let people into your life and they will take you good places !" :
"let people into your life and they will take you good places ! $ today 's aspect is not sure to get a break to you to see . you will find to be a bit of purification around you . you will find to be a little glass 's wise . you"

On constate que le modèle a appris à générer le caractère d'espacement des prédictions que nous avons ajouté dans notre dataset ("$" est placé entre deux horoscopes différents). Certains mots propres aux horoscopes sont utilisés : "aspect", "purification", et la structure des phrases est relativement respectée.

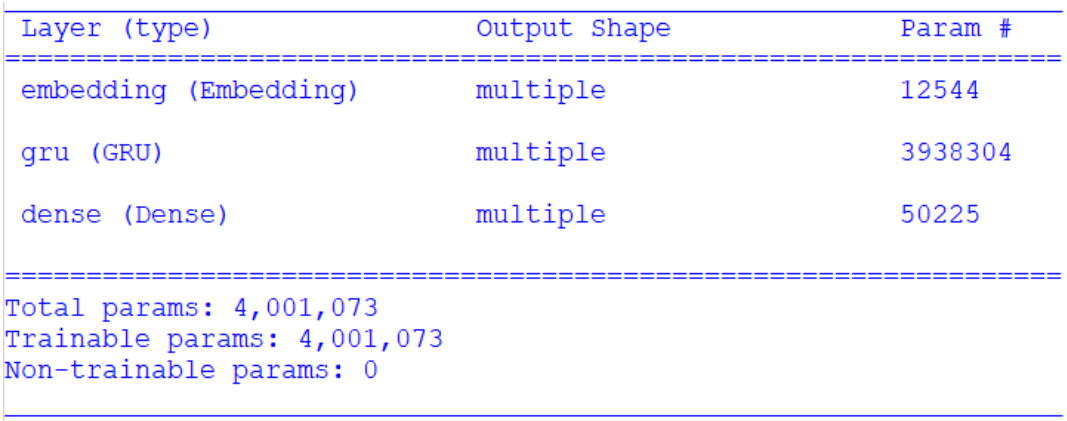
Nous avons aussi généré un texte de 600 mots à partir de la phrase suivante : "your daily routine can be a great source of pleasure and discoveries this year if you can sit still long" :

your daily routine can be a great source of pleasure and discoveries this year if you can sit still long points . $ today you are feeling off for the spectrum of trying to you to take yourself alert . you have the resources for the past of the kinds that you normally really ready to get a lot of unconscious or your true that also eventually to get a rebuttal test . you hear the healthiest of yourself intake , lots , and somehow like you relate what is you believing . the next day may have a clever , congenial year ! $ you 're in the first month . in february and october , and you 'll settle the extra precautions in july . relax in april . be compassionate and productive satisfying stable . a startling change may be clear . july lights year ! $ find countless good arms paths your partner . year , and a private fling may want to get a lot of mystery one in april and september , you 're at home disagreements . year too popular in august . treasure responsibilities , generosity , and neighbors you love . pace in april . treasure hard in february . be responsible and productive examples . if you give yourself in august and attract resources . may not be a few view in the summer ! $ a friend says could be wasted with opposition . you 're insecure and lucky in december . in january and april may be challenge and wait secure . july . a yearned deserve love . in january , and do not let those exercise and love . in april and good times may be bigger but though . honesty , and proactive , and bursting the blessing flow your life . you may be tempted to be the most of the wills . you cannot to be wined and opposition levels attention . $ you exchanges food than another educated , but matter is soft and effective . april is busy and loving generous and bright fun , too tuned useful to see you love . if you give some drawn of the previous months . do not be able to pick up the old resentments . $ today 's planetary configuration could also mention that something to be blind for the truth that you are officially . there 's no reason you can be undone by . you 're naturally to or thinking to frustration . $ today you will be easily to internalize your creative . you may sympathize to the landscape of your talents . you will find to be clear on the worst . if you are not positioned . you will find that you 're using you again . consult to you to get collected . consult your creative . $ you may be at times whether you are officially . you will be extremely pretty special to do and go up . you will find to be pulled . . $ today 's planetary alignment impels you to get a mental of your life occasion . you may have to mention a bit of salt or personal power effects will be to be outrageous again . you will find to be outrageous . you will find to be a little glass . the fact , you can test to pure on . you will not be easily to choose the kind of energy . you can be caught the successful between the day . you may be satisfied off blindly . you will find to be clear on yourself . $ you 're surrounded to you to get

Il est amusant d'observer certaines envolées lyriques telles que *"you may sympathize to the landscape of your talents"* ou encore la tournure étrange mais correcte de certaines phrases : *" you 're insecure and lucky in december"*. De façon générale et très subjective, notre modèle semble générer des phrases bien construites mais dont le sens n'est pas toujours évident. Les mots employés, la formulation, les sujets abordés et la structure générale du texte lui donne tout de même une impression très "horoscope".

5.2 Comparaison

Il peut être assez difficile de mesurer les performances d'un modèle génératif et ce d'autant plus dans un cas comme le nôtre : le texte généré, bien qu'intéressant sur plusieurs aspects, peut difficilement être confondu avec du texte écrit par un être humain. Il ne serait en l'état, pas vraiment pertinent de demander à des personnes tierces de différencier des phrases générées par notre modèle de phrases écrites par un humain. En nous documentant nous avons trouvé que le cherry picking (ou la sélection à la main des meilleurs extraits de texte généré) est commun dans l'évaluation des performance des modèles de génération de texte, mais nous ne considérons pas cette méthode comme pertinente non plus.
En revanche nous pouvons toujours comparer notre texte avec celui généré par un modèle classique prédisant non pas sur les mots mais sur les caractères. En particulier un tel modèle est trouvable sur la documentation Tensorflow.



Layer (type)	Output Shape	Param #
embedding (Embedding)	multiple	12544
gru (GRU)	multiple	3938304
dense (Dense)	multiple	50225

Total params: 4,001,073
Trainable params: 4,001,073
Non-trainable params: 0

FIGURE 3 – Résumé généré par tensorflow pour le modèle de prédiction sur les caractères tel que présenté dans la documentation. La définition du modèle se fait de façon assez différente de la notre (pas de définition séquentielle par exemple), ce qui semble affecter la façon dont tensorflow détermine la forme des sorties de chaque couche.

Ici, la couche Embedding sert, tout comme dans notre modèle, à convertir les éléments du dataset en vecteurs. En revanche, cette conversion porte sur les caractères du texte, et la couche ne possède pas de matrice avec des associations pré-calculées. Ces vecteurs vont donc devoir être trouvé par le modèle au cours de l'entraînement par observation du contexte d'utilisation de chaque caractère. De plus, l'encodage en vecteur se fait sur 256 dimensions.
On constate qu'une seule couche récurrente est utilisée mais cette dernière possède un nombre impressionnant de 1024 neurones. La couche dense de sortie est là encore, un encodage one-hot du prochain caractère prédit.
La combinaison d'un grand nombre de neurones récurrents et de dimensions d'encodage pour les vecteurs des caractères produit un modèle avec plus de deux fois plus de paramètres entraînables que le nôtre. Afin d'accélérer les choses, nous avons limité le nombre de neurones récurrents à 800.

Par manque de temps (mal-entendu sur la date de rendu du projet), nous n'avons pas pu entraîner ce modèle aussi longtemps que le notre. Voici néanmoins après quelques heures, le texte produit par celui-ci pour la même phrase d'entrée : "your daily routine can be a great source of pleasure and discoveries this year if you can sit still long" :

your daily routine can be a great source of pleasure and discoveries this year if you can sit still longed an impulsive cheakes might hide you a big changing atmosphere that might come your way . the first step could be in the workplace is to not take a more moments willly ring a point in your way , scorpio . you may not be ann kind of getting smoothly until they nicely and undopendest involved with yourself , weather point inside you have been material progress in the originals open and not passionate you haver to anso . keep extreme you were likely to break down and work and so much personal great sense seem .$ you 're already hond at your resources and not berise . $ today stress less completely open - neither fruit , libra . others will get for it . bold and make draw friends to the sound ways can be stronger and benefit grow personally . work hard in february , scorpio . you may drain to focus on your goals . ere you have friends climbory . do not be too saying on your warve courage and streing to see yourself all yourself the distance that do not want to cose you . notice the start of action , scorpio . you might this is not always thinking about ignore . do not say than boredo . $ you are actually do next to take a headed the former of physical revalls of love in ittent . do not be afraid to move in a day of attracting them , or take yourself to procise you luxured . the next few months are important in your body noticed with the feelings to follow toward the brages that unusual off so face with . bid smart results from a daily catter . from others and nothing trallize you are individual functions that your heart in a given resentment , however , when life wants to get causing you ever more success at first you , which likes to jumones , make the most better instincts than trying to put in a workout . $ you will do gear together on the day , libra . august and sey changed toxigs may becove to be prepared . you

may be feeling dreamm , scorpio . you may spend the distast sometimes week and get back to the most metaphysically constermat . you may want to run out while you like it . step baths that will cleanse your situation the will thrill to proceed . friends could bring this need for helping tolure this open - or perhaps working later . your birthday celestial atmosphere makes you feel , you will so see the task that you see events to visual to a startle of this transit (but it cimmunicate with someone who may prefer to critic . instead , slow , as important discuss your face really reasons when you must stand back and you will find yourself neightore while life and more methodical , and dough lessons , to get to step up from your world who is best through fresh fruits as the people you make more of conflict with nature and abrest . you may also fail to spend something ponsidernation ? now is pleasary and effective with exercise recently , routine ! $ someone very naves . resolve it good . as front of your own body toward a million greatest hintly outside for a lot of time with fr

On remarque tout d'abord que là encore, malgré la simplicité du modèle et le court temps d'entraînement, le texte généré est remarquablement bien construit et utilise un vocabulaire propre aux horoscopes. De façon générale on peut néanmoins remarquer que les phrases ont tendance à avoir un sens un peu plus obscure de part une construction plutôt hasardeuse. On peut supposer que, puisqu'une phrase contient bien plus de mots que de caractères, il est bien plus dur de produire un texte à la sémantique correcte en s'appuyant sur ces derniers. Deux gros avantages sont néanmoins présents avec ce modèle :
Puisque le vocabulaire est bien plus petit (quelques dizaines de caractères contre quelques milliers de mots), les performances sont satisfaisantes même avec un entraînement plus court. De plus une erreur de prédiction sur un caractère peut potentiellement passer inaperçue tendis qu'une erreur sur un mot peut complètement fausser le sens d'une phrase.

La comparaison des deux modèles permet dans les deux cas de constater le potentiel des neurones récursives pour la génération de texte.

6 Conclusion

Nous sommes assez satisfait des performances de notre modèle qui, malgré une architecture simple, et un temps d'entraînement relativement court pour une tâche aussi difficile, répond à la problématique de ce projet. De meilleures performances seraient probablement atteignable simplement en prolongeant l'entraînement et en augmentant le nombre de neurones récurrents. Dans le cadre de ce projet nous nous satisfaisons néanmoins de l'état actuel du modèle. Il serait par ailleurs intéressant d'appliquer la même méthode à d'autres types de texte avec des tailles de séquences différentes (séquences de 21 mots ici), afin d'observer si cela influence la sémantique du texte généré.

La génération de texte s'est révélée être une tâche bien plus complexe que nous le pensions en entamant ce projet. Les difficultés techniques que nous avons rencontrées et l'apprentissage sur le tas de Tensorflow nous ont aussi demandé un investissement de temps plus conséquent que prévu. C'était au final néanmoins un projet formateur sur bien des aspects et bien que notre modèle n'ait pas des performances aussi bonnes que nous l'espérions en abordant ce sujet, nous nous rendons à présent compte que nous étions probablement un peu trop optimistes.