Comparaison des RNN, LSTM et des GRU dans le cadre du pos-tagging de texte

NOELE Elodie, TABOGA Vincent, WOZNIAK Anne-Laure

[elodie.noele@polymtl.ca](mailto:elodie.noele@polymtl.ca)

[vincent.taboga@polymtl.ca](mailto:vincent.taboga@polymtl.ca)

[anne-laure.wozniak@polymtl.ca](mailto:anne-laure.wozniak@polymtl.ca)

Résumé

Ce rapport vise à étudier les performances de différents modèles de RNN, de GRU et de LSTM dans le cadre du pos-tagging de texte. Le pos-tagging de texte consiste à associer chaque mot d’un texte à une entité grammaticale bien définie. Dans ces modèles, on retrouve les modèles vanilla RNN, RNN à deux couches avec dropout, RNN bidirectionnels, les LSTM, les LSTM bidirectionnels, les LSTM bidirectionnels à deux couches, les vanilla GRU, les GRU bidirectionnels et les GRU à deux couches avec dropout. Ce rapport montre que des performances significatives sont obtenus lorsque l’on utilise le modèle ??? ( A compléter).

Introduction*[[1]](#footnote-1)*

Le POS-Tagging de texte est une étape préliminaire à la traduction de texte d’une langue à l’autre. C’est une des tâches courantes du traitement du langage. Cette technique peut aussi être utilisée pour identifier les mots clés d’une phrase sur un moteur de recherche et ainsi, mieux cibler la demande d’un utilisateur.

Le POS-Tagging est un domaine couramment étudié en intelligence artificielle, puisqu’il donne, généralement, de bien meilleurs résultats que le pos-tagging avec des algorithmes « classiques » comme celui proposé par la bibliothèque NLTK de Python.

Dans cet article, nous proposons différents types de réseaux neuronaux récurrents que nous évaluons sur une expérience de POS-tagging. Dans ces modèles, on peut retrouver les vanilla RNN, les RNN à deux couches, les RNN bidirectionnel, les LSTM simples, les LSTM à deux couches, les LSTM bidirectionnel, les GRU simples, les GRU bidirectionnel et les GRU à deux couches. Nous évaluons ensuite ces réseaux sur différents corpus de texte et nous comparons nos résultats avec les résultats des travaux antérieurs.

L’article est donc organisé de la manière suivante : en premier lieu, nous présentons les travaux antérieurs, puis les modèles retenus, les expériences menées sur ces modèles, nos résultats que nous analysons et que nous critiques dans la section Discussion. Enfin, en dernier section, nous concluons.

Travaux antérieurs

Les travaux antérieurs à notre projet font état de l’utilisation de réseaux neuronaux de type LSTM pour le pos-tagging de texte. En effet, ce sont ces réseaux avec les GRU qui montrent les meilleurs résultats dans le cadre du pos-tagging. Plus particulièrement, les réseaux neuronaux de type LSTM bidirectionnel couplés à des méthodes statistiques telle que le CRF (Wei Xu et al.) obtiennent de bien meilleurs résultats que les GRU simples ou les LSTM bidirectionnel (Xuezhe Ma and Eduard Hovy)

A compléter si nécessaire

This Is an Example of a Figure Caption. Use of the figure caption style is required.

Modèles retenus

RNN

Les réseaux neuronaux récurrents sont des réseaux dont les connexions forment des boucles. Ils possèdent donc un état interne, ce qui en font des candidats propices pour le traitement du langage naturel. Ces réseaux appliquent des opérations matricielles au terme d’observation o et aux unités cachées h de l’état passé. Les résultats de ces opérations sont ensuite passés en entrée de la fonction d’activation (ici log-softmax).

Avec :

, la matrice du réseau

, la matrice des poids de l’unité cachée

, la matrice des poids de la sortie

, le terme de biais de l’unité cachée

, le terme de biais de la sortie

LSTM

Les LSTM sont des types de RNN qui évitent le problème du gradient explosif ou évanescent lors de l’entraînement des RNN. Ils permettent de préserver la norme de l’erreur lors de la rétropropagation du gradient. Avec une erreur qui n’explose pas ou qui ne disparaît pas, les LSTM peuvent donc apprendre sur de larges périodes de temps et avec plus de couches, ce qui leur permet mieux réaliser les tâches de traitement du langage naturel.

En comparaison des RNN, les LSTM possèdent en plus une cellule mémoire qui permet de garder ou non l’état passé du neurone, permettant ainsi une meilleure propagation des informations à travers les couches.

GRU

Une autre manière d’éviter le problème du gradient évanescent est d’utiliser des GRU. Contrairement aux LSTM, les GRU possèdent une porte de mise à jour et une porte de mise à zéro, qui sont utilisées afin de décider si l’état actuel du neurone doit être passé en argument de la fonction d’activation. Ils ne retiennent donc que les informations a priori utiles pour l’apprentissage du réseau.

En outre, les GRU sont plus rapides à entraîner que les LSTM.

Réseau à deux couches

Les réseaux à deux couches sont un type de réseau neuronal où l’information circule de la couche d’entrée, vers les couches cachées puis la couche de sortie. C’est un réseau de type feedforward. Comparé au réseau neuronal dit “simple”, les réseaux à deux couches, couplé à la rétropropagation du gradient, permettent aux réseaux neuronaux d’effectuer des taches plus abstraites, comme l’exercice du POS-Tagging qui est un problème non linéaire.

Il aurait été possible d’utiliser des réseaux à plus de deux couches, mais, nos experiences nous ont montré que les résultats n’étaient pas forcément meilleurs, notamment dans le cas des RNN où nous pourrions faire face à un problème de gradient explosif ou évanescent.

Réseau bidirectionnel

Dans le cadre du POS-tagging, il peut être bénéfique de savoir quels sont les mots avant et quels sont les mots après le mot que l’on souhaite classifier. C’est pourquoi l’utilisation des réseaux neuronaux bidirectionnels se prête volontiers à cette tâche.

Dans un réseau bidirectionnel, les informations passées et futures sont mémorisées en mémoires dans deux sous-réseaux avant et arrière. Il fonctionne en trois étapes :

* Le réseau arrière est utilisé pour déterminer les tags et les états de la couche cachée (s’il y en a, par exemple pour les LSTM)
* Ensuite, ces tags et ces états de la couche caché sont utilisés comme données d’entrée pour le réseau avant.
* Enfin, le modèle global parcourt la séquence complète en utilisant les tags et les états des réseaux avant et arrière.

L’avantage de ce type de réseau est qu’il permet d’avoir une « compréhension » plus fine de la phrase à tagger et donc d’éviter les erreurs. Cette procédure a été définie dans l’article de **Schuster & Paliwal, 1997**.

Expériences menées

Pour tester nos réseaux neuronaux récurrents, nous avons choisi d’utiliser la bibliothèque NLTK de Python qui propose un corpus de texte Brown dont le pos-tagging a déjà été réalisé. Ce corpus de texte possède plus de 50 000 phrases issues de la littérarure anglaise. Nous avons donc fait le choix de tester nos modèles sur ces corpus en prenant **un ensemble d’entraînement de 5000 phrases**, **un ensemble de validation de 2000 phrases** et **un ensemble de tests de 1000 phrases**.

En outre, pour avoir une représentation plus fine de la performance de nos modèles nous avons choisi de tester nos modèles sur un second ensemble de tests composé de 500 phrases courtes puis de 500 phrases longues. Cela avait pour but de mettre en lumière la capacité de chacun des réseaux à ne pas oublier la contribution des premiers termes de la phrase lue.

En termes d’architecture du réseau, nous évaluons la perte de nos modèles avec une fonction

Résultats

|  |  |
| --- | --- |
| **Corpus** | **Brown Corpus** |
| **Vanilla RNN** | 72,16 % |
| **RNN à deux couches** | 73,31 % |
| **RNN bidirectionnel** | 78,11 % |
| **LSTM simples** |  |
| **LTSM à deux couches** |  |
| **LSTM bidirectionnel** |  |
| **GRU simple** | 75,36 % |
| **GRU bidirectionnel** | 75,59 % |
| **GRU à deux couches** | 80,92 % |

Si nous comparons nos résultats par classes de modèles, on peut retenir les RNN bidirectionnel avec un résultat de 78,11 %, les LSTM ??? avec ???, les GRU à deux couches avec un résultat de 80,92%

Discussion

Comparaison avec l’existant

Dans la littérature sur le pos-tagging de texte avec les réseaux neuronaux, la majorité des réseaux qui obtiennent les meilleurs scores sont de types LSTM. Sur un corpus similaire au notre, Wei Xu et al. obtiennent un score de près de 98% en utilisant un LSTM-CRF, c’est-à-dire un LSTM couplé à des champs aléatoires conditionnels, permettant de mieux prendre en compte la contribution des différentes mos d’une phrase.

Dans les autres articles du même acabit, les réseaux obtennan les meilleurs résultats sont ceux de type LSTM bidirectionnel et GRU simple.

Dans notre cas, A COMPLETER

Limites de notre approche

Notre approche se limite à des modèles de réseaux neuronaux « simples », cependant nous avons réalisé des tests avec des modèles de type RNN, LSTM et GRU à 10 couches et notre F1-score accuracy ne dépassait les 30 %.

Nous aurions pu tester nos modèles sur des ensembles plus grands et sur plus d’ensembles, cependant, les contraintes ont été le manque de temps, puisque chacun des modèles met environ 3 heures à s’entraîner voire plus dans le cas des modèles de type LTSM. Aussi, nous avons été limités par la performance de nos ordinateurs.

Conclusion

Explication de pourquoi les RNN sont moins bons que les LTSM et les GRU

Références

Juan Antonio Pérez-Ortiz, Mikel L. Forcada, *Part-of-Speech Tagging with Recurrent Neural Networks*, Departament de e Llenguatges i Sistemes Informatics, Universitat d’Alacant, E-03071 Alacant, Spain

Ashish Vaswani, Yonatan Bisk, University of Southern California, Kenji Sagae, Kitt.ai, Ryan Musa, University of Illinois at Urbana-Champaign, *Supertagging with LSTMs*

Xuezhe Ma and Eduard Hovy*, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF,* Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Zhiheng Huang, Wei Xu, Kai Yu, *Bidirectional LSTM-CRF Models for Sequence Tagging*, Baidu research

M. Schuster and K. K. Paliwal*, Bidirectional recurrent neural networks*, in IEEE Transactions on Signal Processing*,* vol. 45, no. 11, pp. 2673-2681, Nov 1997.

Ian H.Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal, Data Mining (Fourth Edition) *Pratical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2017

1. Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved. [↑](#footnote-ref-1)