

Agents conversationnels à base de réseaux de neurones artificiels profonds

Guillaume Chevalier

Université Laval

Baccalauréat en génie logiciel

guillaume.chevalier.2@ulaval.ca

Samuel Cabral Cruz

Université Laval

Baccalauréat en génie logiciel

samuel.cabral-cruz.1@ulaval.ca

Abstract

1 Introduction

2 Développement

2.1 Traitement d'un intrant vocal

2.2 Extraction des composantes de l'intrant et des sources d'information à analyser

Une fois que la requête de l'utilisateur aura été convertie sous une forme textuelle facilement manipulable par un ordinateur, nous pourrions, dès lors, utiliser le plongement induit par l'étape précédente. Une autre approche consiste à reprendre cette sortie pour ensuite la fournir à une nouvelle structure qui se chargera d'aller extraire de nouvelles composantes qui aideront certainement à obtenir de meilleurs résultats pour la suite du processus.

À ce stade, nous devons comprendre que le signal est encore purement textuel et nous n'avons pour seule information qu'une décomposition des mots qui forment la demande reçue. Cependant, les langages sont formés de davantage de subtilités qu'un simple enchaînement de mots les uns après les autres. En effet, chaque mot joue un rôle précis dans la structure de la phrase et apporte une nuance particulière au contexte générale de celle-ci ou encore du texte avec une plus faible portée. C'est exactement ce que les travaux de ... visaient à faire. Ainsi, en ..., ce groupe de chercheurs a fait la publication d'un article détaillant leur approche fondée sur l'utilisation de réseaux de neurones. En plus de faire état de leurs travaux, ce groupe est aussi à l'origine d'outil qui est encore à ce jour considéré comme un incontournable : (Mikolov et al., 2013).

Malgré le fait que cet article pour sur les approches neuronales, cet outil a plutôt fait la

démonstration que des approches plus simplistes sont parfois plus adaptées. Comme le montre les figures Figure 1 et Figure 2, Word2vec se fonde sur la combinaison de deux approches nommées skip-gram et bag of words consistant simplement a.

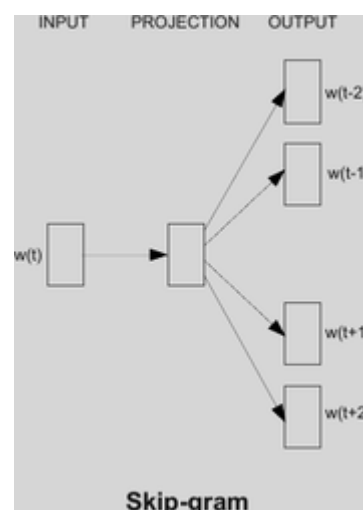


FIGURE 1 – Architecture de la méthode de prédiction Skip-gram

En fournissant la requête reçue à cet outil, il sera donc possible d'extraire les composantes sémantiques et syntaxiques sous-entendues par cette dernière. Par la suite, ces nouvelles composantes seront combinées à celle que nous avons déjà obtenues à l'étape précédente. En procédant avec cette seconde approche, nous réaliserons certainement un gain majeur au niveau de la performance des prochaines étapes en raison de l'ajout important de dimensionnalités qui fourniront beaucoup plus de flexibilité aux réseaux de neurones suivantes qui devront à leur tour détecter les nuances du langage. À titre d'exemple, lorsqu'un utilisateur demandera à l'assistant si ce dernier peut lui indiquer l'horaire du cinéma le plus prêt de sa position, l'assistant

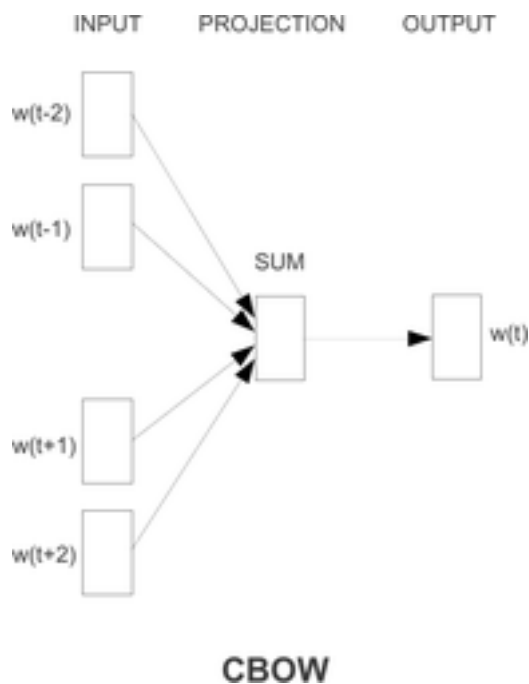


FIGURE 2 – Architecture de la méthode de prédiction CBOW

devra comprendre la nuance que ce qui intéresse vraiment l'utilisateur est l'horaire et non pas l'évaluation booléenne de sa capacité à s'acquitter de cette tâche. Par contre, dans le cas où l'utilisateur demanderait à l'assistant si ce dernier peut le connecté à l'Internet, l'assistant devra dans ce cas faire l'évaluation de sa capacité et répondre en par affirmation à notre cher utilisateur. [Référence au transfer learning (TODO)]

Mais qu'en est-il de nos sources d'informations? En fait, le processus entier bénéficiera certainement qu'un travail similaire soit fait à ce niveau aussi. Pour ce faire, deux approches s'offrent encore à nous. La première consistant encore une fois à utiliser word2vec et la seconde repose sur le même principe, mais à un niveau supérieur d'abstraction en considérant cette fois l'utilité de chacune des phrases dans le texte plutôt que de se concentrer sur le rôle de chaque mot dans chaque phrase.

Il y a 2 façons d'avoir l'embedding du corpus de texte (ex : tout wikipédia) lequel sera utilisé pour répondre à la question. Word2vec et infersent. Expliquer les niveaux d'embedding de word2vec et infersent (word-level et sentence-level). Plugger ça dans un RNN ou autre chose, tel que vu à la section suivante.

2.3 Interpréter la requête et cibler le contenu d'intérêt pour y répondre

2.4 Formulation d'une réponse à partir de l'information d'intérêt retenu

2.5 Retourner la réponse textuelle sous la forme d'un signal audio

Conclusion

Remerciements

References

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*.