Sparse Embeddings pour la Recherche d'Information

Laure Soulier & Adrien Pouyet *

Objectifs et Compétences Acquises

— Assimiler l'état de l'art sur la représentation du texte en machine learning et en appréhender les enjeux
— Developper une architecture deep learning avec PyTorch
— Implémenter les découvertes récentes en recherche d'information (ranking)

Introduction

En Traitement Automatique de la Langue[1], une des manières de représenter du texte dans une machine est d'utiliser une représentation continue, appelée *embedding*[2]. Comme vous verrez au cours du cours RITAL (qui n'est pas un prérequis pour ce projet), il existe beaucoup de modèles de représentations (Bag of Word, Word2Vec, Doc2Vec, ...) que l'on peut étendre à des phrases ou des textes. Ce projet permet d'explorer une des manières de représenter du texte que l'on qualifiera d'*embedding sparse*.

Travail à réaliser

Pour ce projet, nous aborderons les modèles de recherche d'information à base de néseaux de neurones. Une première partie sera consacrée à l'état de l'art du domaine[1, 2, 5] ainsi que des modèles de représentation de textes [3, 4].

Ensuite, il vous sera demandé de réimplémenter le papier [1]. Il s'agit d'un modèle de ranking en recherche d'information. La recherche d'information est le champs de recherche qui s'intéresse à satisfaire le besoin en information d'un utilisateur ; le ranking est une manière de compléter cette tâche en proposant à l'utilisateur un ordre dans des documents. L'objectif du ranking est de présenter en premier le document le plus pertinent pour ce que l'utilisateur recherche, puis le deuxième plus pertinent, etc. Le meilleur exemple est Google. La particuliarité de ce modèle est qu'il allie la rapidité des méthodes traditionnelles (à base d'index) et la performance des méthodes de machine learning.

Du point de vue méthodologique, nous commencerons par des modèles simples de ranking par réseaux de neurones (à base de convolution et de réseaux recurrents) pour vous familiariser avec le sujet et apprendre PyTorch. Ensuite, il sera temps d'utiliser ces nouvelles connaissances pour implémenter le papier [1]. Si le temps le permet, vous pourrez commencer un travail plus exploratoire ou analytique (selon vos préférences) sur les embeddings. Le code produit devra suivre une architecture logicielle d'une qualité attendue en master.

Références

[1] ZAMANI, Hamed, DEHGHANI, Mostafa, CROFT, W. Bruce, et al. From neural re-ranking to neural ranking : Learning a sparse representation for inverted indexing. In : Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 2018 link

[2] DEHGHANI, Mostafa, ZAMANI, Hamed, SEVERYN, Aliaksei, et al. Neural ranking models with weak supervision. In : Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2017. link

[3] MIKOLOV, Tomas, SUTSKEVER, Ilya, CHEN, Kai, et al. Distributed representations of words and phrases and their compositionality. In : Advances in neural information processing systems. 2013. link

[4] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. link

[5] Huang, P. S., He, X., Gao, J., Deng, L., Acero, A., Heck, L. (2013, October). Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Information Knowledge Management ACM. link

*laure.soulier@lip6.fr, adrien.pouyet@lip6.fr

1. abrégé TAL ou NLP (Natural Language Processing) en anglais
2. *to embody* signifie incarner en anglais