(May 23, 2018)

Project Title (Tentative) : Toward developing a medical Natural Language Generation system using Deep Learning and Reinforcement Learning

Abstract

We aim to build a prototype framework for Natural Language Generation (NLG) for the medical field leveraging recent advances in Deep Learning and Reinforcement Learning. By defining the task of NLG as sequential decision-making process, we implement two baselines, one built upon Long Short-Term Memory (LSTM)-based Recurrent Neural Network (RNN) and the other with Monte Carlo Tree Search (MCTS). Later, other enhanced variants based on MCTS are further pursued while research focuses for roles of evaluation metric, target corpus, and algorithmic benefits, etc are actively undertaking.

References

LSTM RNN

* <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
* Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
* More to come

MCTS

* Kumagai, K., Kobayashi, I., Mochihashi, D., Asoh, H., Nakamura, T., & Nagai, T. (2016). Human-like Natural Language Generation Using Monte Carlo Tree Search. In *Proceedings of the INLG 2016 Workshop on Computational Creativity in Natural Language Generation* (pp. 11-18).
* More to come

Deep Reinforcement learning

1. Some sort of Introduction (Only for your information, Not for scientific works)

* <https://deepmind.com/>
* (Just one hit randomly) <https://www.wired.com/story/greedy-brittle-opaque-and-shallow-the-downsides-to-deep-learning/>

Deep Learning

1. Artificial Neural Network models (since 2006, emphasized in its deep architecture)
   * End-to-end training with backpropagation
   * Scalable for large datasets
2. Great performance for human-like perception problems (vision, NLP, audio, speech, etc). \*biological sequences such as genomic and proteins. \*some interests on graph datasets
3. Machine Learning framework with representation learning, offering effective methods for unsupervised learning and other advanced methods such as meta learning, transfer learning, active learning, multi-task learning, multi modal learning, etc
4. Deep Reinforcement Learning
   * Deep Learning topics with Reinforcement Learning techniques
   * Reinforcement Learning combined with Deep Learning-based function approximation

* DNN, RNN, CNN, and my hybrid or complex architectures
* RNN – Long Short-Term Memory (LSTM)

Reinforcement Learning

1. MDP vs. POMDP
2. Monte-Carlo Tree Search

Technical Requirements

Pre-requisite

1. PyTorch
2. Spacy
3. GitHub
4. Jupyter
5. Anadonda/PIP
6. Linux commands : ssh, etc

Note : there are numerous tutorials and youtube clips you can watch for more information. And, the most efficient way to master and to remember what you know is to repeat the same thing many times!

1. GitHub

<https://www.atlassian.com/git?utm_source=basic-git-commands&utm_medium=link&utm_campaign=git-microsite&_ga=2.75964245.1174987269.1527087707-1877757822.1527087707>

Our repository location : http://github.com/

1. Python
   1. Shell vs. Jupyter notebook
   2. Package management with pip

https://pip.pypa.io/en/stable/

* 1. Environment with anaconda

<https://conda.io/docs/_downloads/conda-cheatsheet.pdf>

1. Jupyter