Developing a Neural Natural Language Generation Approach for a Medical Spoken Dialogue System with PyTorch

Sarah Days-Merrill¹, Joohyun Kim²

¹ Department of Mathematics, Bridgewater State University
² Center for Computation and Technology, Louisiana State University

Abstract

We aim to develop a Neural Natural Language Generation (NLG) approach using PyTorch, following the structure from Tsung Hsien Wen’s RNN model written in Theano. Defining NLG as a sequence generation process, we plan to implement an Encoder-Decoder with Attention model using Long Short-Term Memory – based Recurrent Neural Network. Using this approach, as opposed to rule – and template – based methods, allows us to highlight representation learning within the hidden states of our model. Applications for our medical Spoken Dialogue System include medical image interpretation, clinical decision support, and medical Q & A.

Outline

I. PyTorch vs Theano
   - Dynamic graph support: PyTorch vs Theano
   - Uses Tensor: PyTorch vs Theano
   - Built-in functions – Parameters defined behind the scenes: PyTorch vs Theano
   - Newer (Released Jan 2017): PyTorch vs Theano

II. Program development environment
   - i. DevOp repository: GitHub
   - ii. Integrated IDE: Jupyter Notebook
   - iii. Virtual Environment and package management: Anaconda
   - iv. Python libraries: torch(0.4.0), NumPy, SpaCy, etc.

III. Baseline Model
   - i. Encoder-Decoder with Attention model

(Soft) Attention Layout

LSTM-based RNN Architecture

Diagram from https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

Diagram from https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg

Attention equations to calculate weight and embedded dialogue act type

\[ \beta_i = \text{tanh}(W_{wax}x_{i-1} + W_{wio}e_i) \]

\[ w_{e,i} = e_i^n / \sum e_i^n \]

\[ d_i = a_i \oplus \sum w_{e,i}e_i \]

References


Acknowledgements

Work supported by the National Science Foundation (NSF) award #ACI-1560410