Final Project Report for CSC 544

Zheng Tang
zhengtang@email.arizona.edu

Introduction

Recurrent neural networks (RNNs) have been successfully proved to be one of the best tools for natural language processing (NLP) tasks and achieved perfect results than other methods. However, the lack of interpretability limits further improvements on their architectures. I found a paper written by Ming et.al. [1], which presents a visual analytics method for understanding and comparing RNN models for NLP tasks. I decide to re-implement the visualization tool described in this paper using the techniques they mentioned.

Background Techniques and Tools

- Recurrent Neural Network
- Co-Clustering
- Comparative Visualization
- D3
- Scikit-learn
- Pytorch
- Highchart

Achieved Goals

- Implemented the LSTM Model for POS Tagging task
- Created dataset for visualization
- Implemented the Word Clouds Layout
- Implemented the memory chips layout for hidden state clusters
- Implemented the Co-Clustering connections Layout
- Implemented the chart for visualizing word's responses according to hidden dimensions

Data and Codes

Data:

- data.js: Prepared data for visualizing.
  - words: contains all input words. It is an array of json objects. Each includes three fields.
    - name: the word itself
    - weight: 20-Euclidean distance between the word and centroid of the cluster, where 20 is the upper bound of the distance

    - ∗ responses: the responses score to each hidden dimensions
  - hidden_units: hidden units id in clusters
  - edges: aggregate responses between a word cluster Wi and a hidden state cluster Hj
- lstm.model: pre-trained LSTM model

Codes:

- lstm.py: lstm model training and testing. (You may need data for this, but I didn't put those huge dataset in GitHub repo)
- vectorizer.py: served for lstm.py as a vectorizing tool.
- utils.py: turn lstm.model to data we need, but I need to manually change this json format to data.js.
- data_viz.js: the visualization code to present all these data
- index.html: the front end of visualization, you just need to open it in any browser.

Result

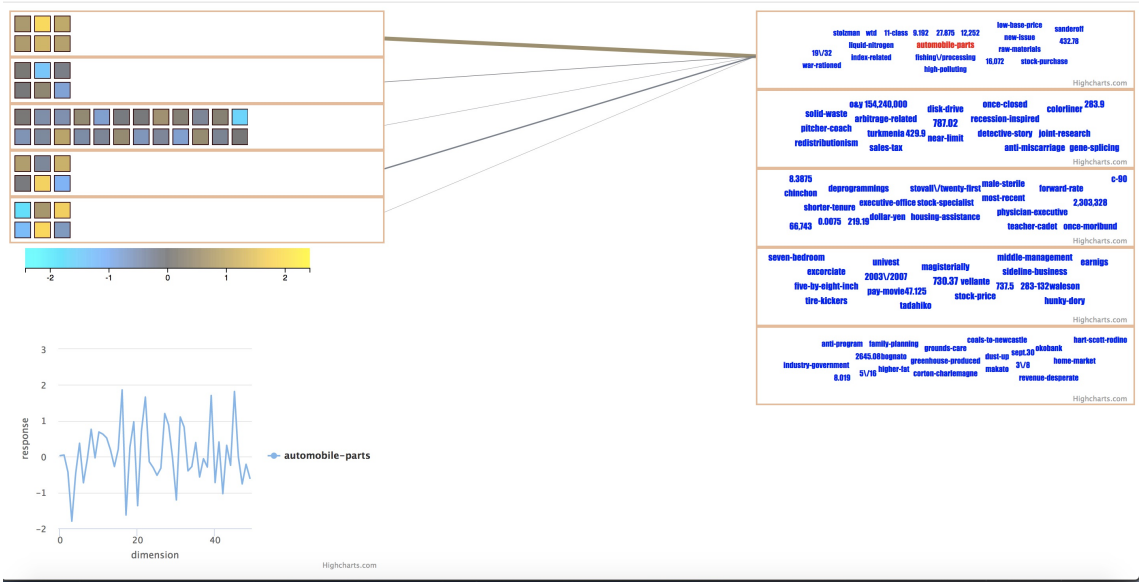

Figure 1: The screenshot of my project. The top left part is the hidden states with a color legend. In its bottom, is the response chart for selected word. The right part is a interactive word clouds, you can click on any word in it. Between word clouds and hidden states is the edges represent the aggregate responses between selected word cluster and all hidden state clusters.

Future Works

First, I want to improve the interactive functions for this visualization tool. So we can change model type and hyper parameters to train different models.
Also, the Glyph for test sequence is very interesting. I'd like to add this part in the future.

References

[1] Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, and Yuanzhe Chen. Understanding hidden memories of recurrent neural networks. *dim*, 1(1.0):0–5.