University of Science
Advanced Program in Computer Science

Thesis Proposal

Thesis' title: USING DEEP LEARNING FOR SENTIMENT ANALYSIS

Advisor: Dr. Nghiem Quoc Minh

Duration: 27/09/2016 - 12/08/2017

Students: Thai Thien (1351040) – Van Duy Vinh (1351050)

Type: Research

Contents:

Introduction:

Sentiment Analysis, also known as opinion mining, is a field, which the primary purpose is to analyze opinions, sentiments or evaluations of people toward products, services, organization, other people, issue, or event. More generally, it analyzes the attitude of a speaker or writer toward a topic. Understanding opinions of others have always been the primary concern for humanity. In real life, we take account of others people opinion before making an important decision; the same apply to organizations which want to know customers' opinions on its products or politicians who stand for the benefit of their party, etc.

Language is the tool human use to communicate most of their thought and emotion. As computer do not have "intuition" or emotion, making a program to understand human language seem to be impossible. Moreover, even human themselves have no idea how their mind work. For this reason, understanding the sentiment of a sentence is itself a very challenging task.

In recent years, with the help from Deep Learning, bigger datasets, and exponentially more computational resource, algorithms for solving Sentiment Analysis are reaching closer and closer to human-level performance. In this thesis, we will use some variations of Recursive Long Short-Term Memory Neural Network to increase the performance of the task sentiment analysis on sentence-level. In particular, we will apply different neural network

structures, and training methods to improve accuracy on the dataset Stanford Sentiment Treebank.

Relative works:

Stanford Sentiment Treebank is first corpus with fully labeled parse tree contains 10,662 sentences of movie review collected from rottentomatoes.com bythen is processed by [3].

Recurrent neural network (RNN) is a suitable to process sequence data (such as sentence). However, when training on very long sequence with RNN, the gradient vector can grow or decay exponentially.

Long Short Term Memory (LSTM) network, which have a memory cell and mechanism to forget old and learn new information, can solve this exploding gradient problem of RNN.

Tree Long Short Term Memory (Tree-LSTM) network is a tree-structure neural network in which each cell is a LSTM cell. Tree-LSTM outperform all other system on Sentiment Classification and Semantic Relatedness task.

Goal:

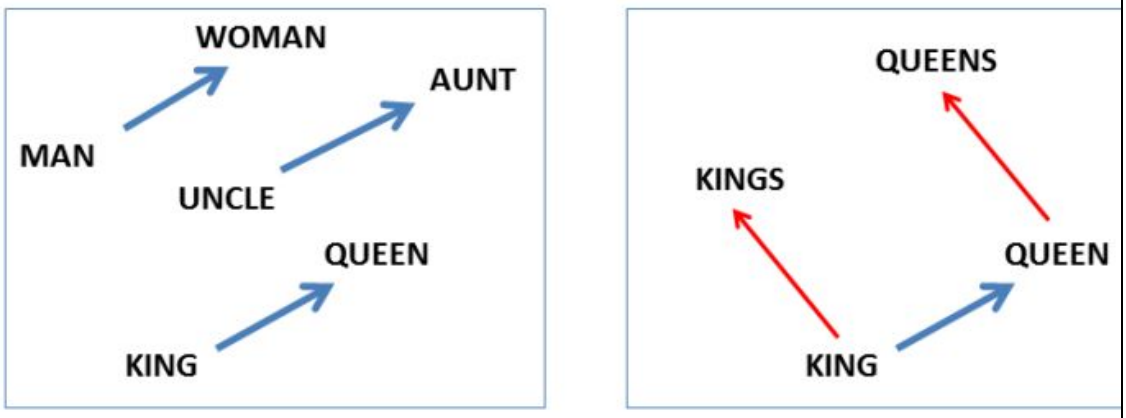
Do experiment on different network models and analyze the accuracy improvement (if there is) on the dataset Stanford Sentiment Treebank

Project details:

The the most influential ideas in this thesis come from recent success in word-level and sentence-level representation learning.

Firstly, the idea of word-level representation learning is that base on analyzing how a word appears in sentences (or which words appear in its context), we can discover the semantic relation between that word and other words. Based on this idea, every word can be present in a rich-feature low-dimensional vector. In this presentation vector space, we can find many relationships between words. For example, presentation vector of words with close

meaning will have a small cosine distance in this space. Even more fascinating, we can find semantic relation like male-female or singular-plural as been demonstrated in Fig 1.



(Mikolov et al., NAACL HLT, 2013)

Fig 1

Secondly, recent advancement in Machine Translation, Sentiment Analysis, and Sentence Relatedness have convincingly show that Recurrent and Recursive Neural Network is good at learning sentences representation. These networks compute a sentence representation by learning how to compose the words presentation in that sentence.

We will adapt these ideas, but try to combine different Recurrent (LSTM, Bi-LSTM , GRU, etc.) and Recursive Neural Network (POS embedding RNN, DRNN) to invent a better model. More particularly, given a sentence, each word presentation will be enriched by cooperating context information into it using a recurrent neural network . After that, the enriched words presentation will be feed to a recursive neural net to classify the sentiment of the sentence. Another idea is to embed the POS-tag information at each node in the parse tree, to have a better word-composer. Finally, we will also analyze the effect of making the recursive neural net “deeper in space” .

Tools:

- ***Programing language:*** Python 2.7, Lua
- ***Anaconda:***

Anaconda is a Python distribution that include library suitable for data science and machine learning such as Numpy, Scipy, OpenBLAS... Anaconda contain all requirement for Theano, another Python library focus on performing, which make install Theano much easier.

- ***Theano:***

Theano is a high performing Python library that can take advantage of GPU for multi-dimensional array computing. Theano integrate tightly with Numpy, a Python library for represent and computing on multi-dimensional array based on c.

- ***Torch7:***

Torch7 is Lua library optimize for machine learning problem, such as multi-dimensional array computing. Similar to Theano, Torch also support running on GPU

- ***Stanford CoreNLP***

Stanford CoreNLP is a toolbox for natural language processing written in Java.

- ***Git and Github***

We use git, a version control system, to manage our source code. We public our source code on Github

Challenges:

Our neural network structures are sophisticated as it combines many different other models, such as Bi-LSTM Deep Recursive Neural Network and Tree-structured LSTM and POS-tag Embedding model.

The implementation utilizes different libraries of different programming language. We use Stanford CoreNLP (written in Java) for preprocessing our data, Torch7 (written in Lua) for implementing one neural network model and Theano (written in Python) for another model. Debugging between different languages make the experiment process slowdown. The

experiments are hard to implement as we are not used to programming deep learning problem yet.

As the data is big, it cost much time and computational resource to train and test a model, as we only know if the model is good or bad based on its performance. At present, we still not be able to take advantage of Theano and Torch7, which optimize to run on GPU, to speed up the running time.

Research time lines:
30/06/2016 - 26/09/2016 Research background, of sentiment analysis and relative work.
27/09/2016 - 31/10/2016 Choosing baseline model and making hypothesis models, which will be implemented and test for performance
01/11/2016 - 07/04/2017 Implementing and doing experiment on our models
02/04/2017 - 01/05/2017 Writing thesis
17/07/2017 - 22/07/2017 Submitting thesis and registration for thesis defend
22/07/2017 - 12/08/2017 Prepare thesis presentation
12/08/2017 - 01/09/2017 Thesis defending

Approved by the advisor(s) **Signature(s) of advisor(s)**	**Ho Chi Minh city, …/…/…** **Signature(s) of student(s)**