NLP Unit 3 Evaluation

Nov 8, 2016, 1:30 pm to 6:30 pm

Objective

In this exercise we will build a RNN based sentiment analyzer. Given a product review text, this model predicts the sentiment as either positive or negative.

Steps

(a) Dataset

i. You are provided 2 files in json format, titled: pos_amazon_cell_phone_reviews.json and neg_amazon_cell_phone_reviews.json. These files respectively contain training examples for positive sentiment and negative sentiment

ii. You are required to select 1000 samples for training and 100 for testing using a random selection process.

iii. Inspect the json file and infer the format which is self explanatory. There are 3 fields in each row that you are required to use. The "text" field is a multi-line text that constitutes the review. The "summary" is a short summary of the multi-line text. Rating of 1 implies negative sentiment and the rating of 5 is positive.

iv. In this evaluation we are concerned only with bipolar classification.

(b) Building RNN

i. Use the RNN that you might have created during U2 Evaluation and select the hyper parameters suitably

ii. You can also use a GRU or any other architecture which is a variant of RNN. It is your choice. But you need to have created it yourself either as a part of U2 or U3.

iii. You need to generate the input from 2 JSON fields: "text" and "summary". Perform any text preprocessing, convert these to vectors using Word2Vec. In parallel, create another dataset where you can use the word2vec representation from: http://www.jnresearchlabs.com:9027. Download the client code and use the Word representation. Later you can compare the performance of your sentiment analyzer using 2 different word representation techniques. Note that the web service returns a 32 element vector and so you should design the rest of the architecture accordingly.

iv. As the text field has many lines, you may clip the number of sentences to 1 or 2 and place an upper limit on number of words. Otherwise the RNN might encounter vanishing gradient problem

v. You can build a GRU (time permitting, optional) in order to get a better result

vi. For the input generated using this process, generate the expected target by using the rating information. This is straightforward as the files are organized as per the label.

(c) Train the system starting with 1000 sequences from the dataset and use the 100 sequences (held out data) to validate.

(d) Measure the accuracy, precision, recall

(e) Experiment with different sizes of dataset rows, different hyper parameter settings

(f) Use the dataset created using lexicalized word reps and check if there is any change with respect to performance of the system

Deliverables

Submit the following by 6:30 pm, 8th Nov 2016:

1. Source code of RNN, sentiment analyzer
2. Source of the test program

Do the following by 10 pm 9th Nov 2016:

1. Post your results and analysis on the Facebook. Regardless of the accuracy you got, explain in the document what went right and where you could have taken a better approach to getting better results.
2. Optionally you can include any graphics, visualization etc

Best wishes from the faculty, enjoy NLP development!

Anantharaman Palacode Narayana Iyer, Chitra GM