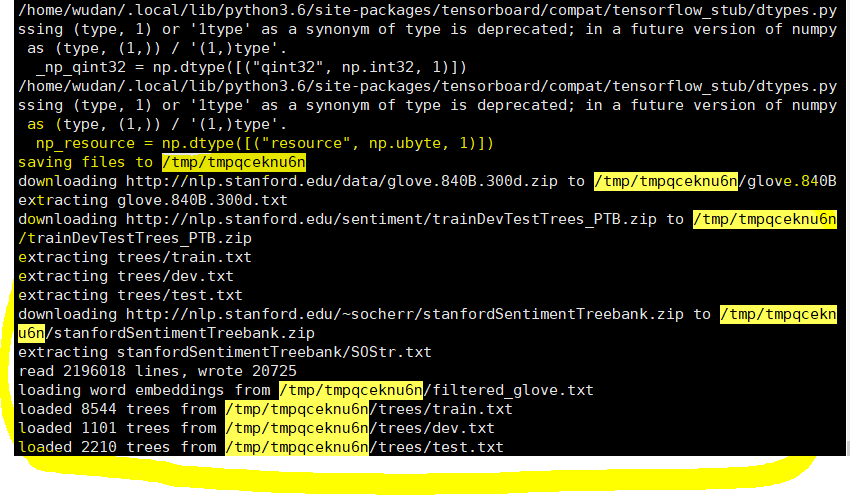
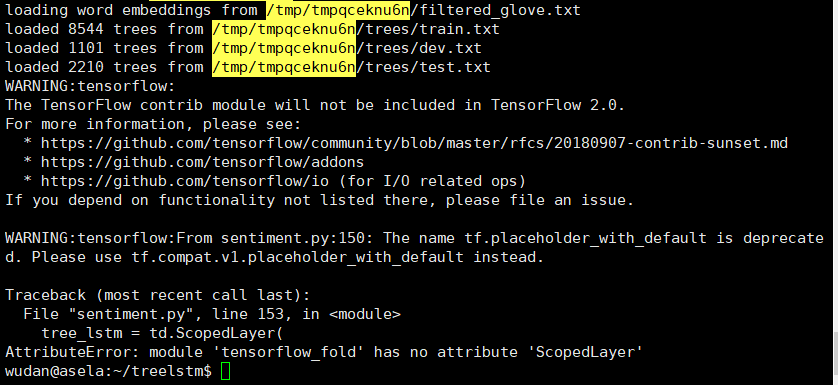
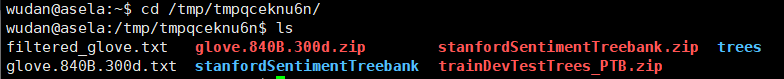
5.8 working on download\_data.py







5.7 working on <https://github.com/tensorflow/fold/blob/master/tensorflow_fold/g3doc/sentiment.ipynb>

Need to manually include this library file:

<https://github.com/tensorflow/fold/tree/master/tensorflow_fold>

LRP for LSTM: <https://github.com/ArrasL/LRP_for_LSTM>

**===========================================================================**

Most Current Questions for Text classification/Prediction:

1. Define LSTM model (sequential keras model)
   1. Input: varying-size window? Dimensions?
   2. Output:
   3. Hyperparameters Tuning:

LSTM hidden layer

dropout layer (intended to reduce model overfitting by the training data)

a dense fully connected layer (to interpret the features extracted by the LSTM hidden layer)

a final output layer (to make predictions)

Epoch and Batch number

1. Decide Algorithms to optimize the network:

(For instance: [Adam version of stochastic gradient descent](https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/))

Categorical cross entropy loss function etc.

1. Model evaluation:

Accuracy on the fit model on the test dataset

* (Notes: neural networks are stochastic, so we need to repeat the evaluation process multiple times and summarize the performance of the model across each of those runs.)
* Summarize by: mean and standard deviation etc.

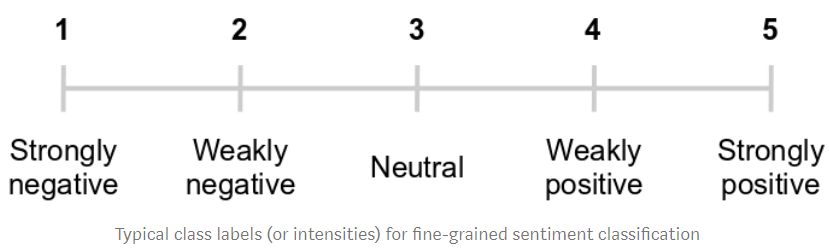
**====================================================================================**

**Stanford Sentiment Treebank: (SST-5 dataset)**

1. **Brief intro:**

**Sentiment treebank introduces fine grained sentiment labels for 215,154 phrases in parse trees of 11,855 sentences. It also presents new challenges for sentiment compositionality.**

**It also provides a suitable benchmark to test our application, “since it was designed to help evaluate a model’s ability to understand representations of sentence structure, rather than just looking at individual words in isolation. SST-5 consists of 11,855 sentences extracted from movie reviews with fine-grained sentiment labels [1–5], as well as 215,154 phrases that compose each sentence in the dataset.”**



**Motivation: dual-polarity sentences such as “*The location was truly disgusting ... but the people there were glorious.*” can confuse binary sentiment classifiers, leading to incorrect class predictions.**

1. **Proposed solution:**

**The Recursive Neural Tensor network (RNTN)**

1. **Performances:**

**When trained on the new treebank, this model outperforms all previous methods on several metrics. It pushes the state of the art in single sentence positive/negative classification from 80% up to 85.4%. The accuracy of predicting fine-grained sentiment labels for all phrases reaches 80.7%, an improvement of 9.7% over bag of features baselines. Lastly, it is the only model that can accurately capture the effects of negation and its scope at various tree levels for both positive and negative phrases.**

1. **What is the state-of-the-art?**

*The original RNTN implemented in the*[*Stanford paper*](https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf)*[Socher et al.] obtained an accuracy of****45.7%****on the full-sentence sentiment classification. More recently, a Bi-attentive Classification Network (BCN) augmented with [ELMo embeddings](https://arxiv.org/pdf/1802.05365v2.pdf" \t "_blank) has been used to*[*achieve a significantly higher accuracy*](https://github.com/sebastianruder/NLP-progress/blob/master/english/sentiment_analysis.md#sst)*of****54.7%****on the SST-5 dataset. The current (as of 2019) state-of-the-art accuracy on the SST-5 dataset is****64.4%****, by*[*a method that uses sentence-level embeddings*](https://arxiv.org/pdf/1806.00807.pdf)*originally designed to solve a paraphrasing task — it ended up doing surprisingly well on fine-grained sentiment analysis as well.*

*Although neural language models have been getting increasingly powerful since 2018, it might take far bigger deep learning models (with far more parameters) augmented with knowledge-based methods (such as graphs) to achieve sufficient semantic context for accuracies of 70–80% in fine-grained sentiment analysis.*

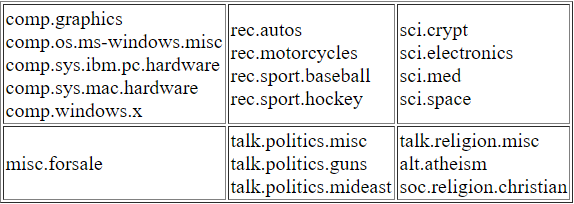
[20 Newsgroup Dataset](http://qwone.com/~jason/20Newsgroups/)**:**

1. **Brief intro:**

**A collection of roughly 20,000 newsgroup documents, this is a popular dataset for experiments in text applications (text classification and text clustering) of machine learning techniques.**

1. **Organizations**

**partitioned (more or less) according to subject matter:**



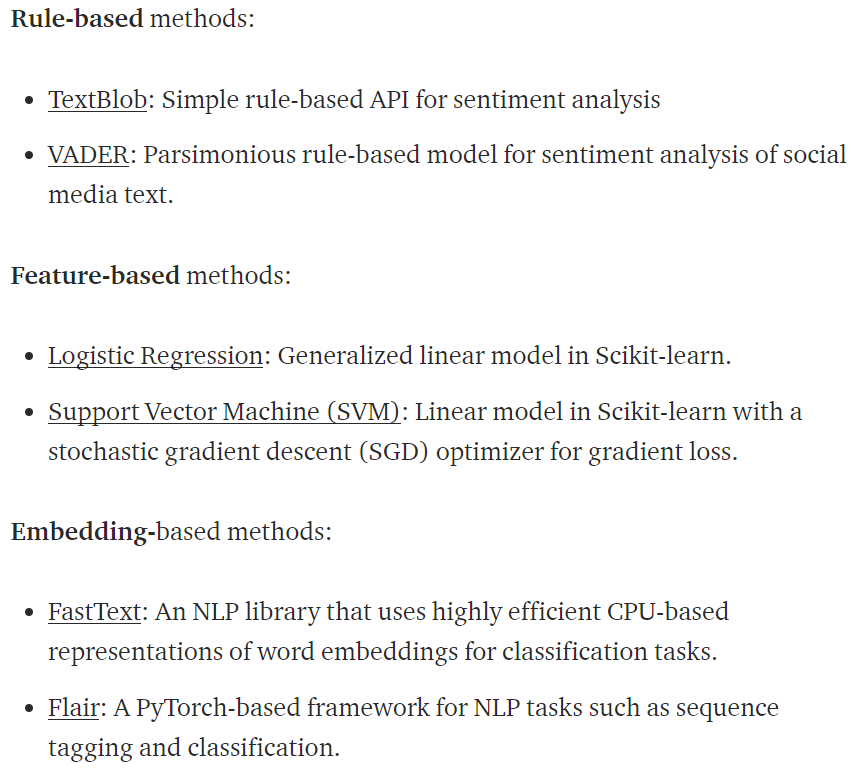
**Yahoo! Answers Topic Classification Dataset:**

**=====================================================================================**

**Fine-grained Sentiment Analysis in Python (**[Part I](https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4)**)**

<https://github.com/prrao87/fine-grained-sentiment>

1. **NLP natural language processing: a field in machine learning with the ability of a computer to understand, analyze, manipulate, and potentially generate human language.**
2. **Different types of model for NLP:**



1. Transform the Dataset to Tabular Form:

Why? ------to evaluate those NLP models and compare how each one differs from the other

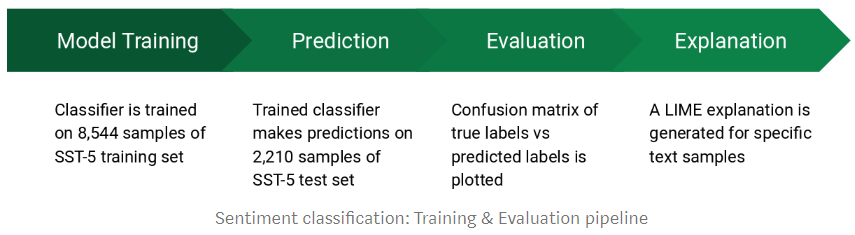
How? ------only use just the complete samples in the training dataset while ignore the component phrases

---------------the tree structure of phrases is converted to raw text and its associated class label using the pytreebank library.

1. EDA exploratory data analysis

Train\_test size, multi-class samples distribution (balanced or not?),

1. Many of the really short text samples belong to the neutral class (#3), so we can create a new column to stores the string length, using the length to sort rows.
2. The dataset labels are not perfect
3. Methodology:



Code:

<https://github.com/TheSarang/CIFAR10-Image-Classification>

<https://github.com/susanli2016/NLP-with-Python/blob/master/Multi-Class%20Text%20Classification%20LSTM%20Consumer%20complaints.ipynb>

<https://github.com/jiweil/Visualizing-and-Understanding-Neural-Models-in-NLP>

papers:

[Explaining Recurrent Neural Network Predictions in Sentiment Analysis](https://arxiv.org/pdf/1706.07206.pdf)

[Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank](https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf)

[Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks](https://arxiv.org/abs/1503.00075)

tutorial:

video tutorial : <https://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv>

<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

<https://towardsdatascience.com/multi-class-text-classification-with-lstm-1590bee1bd17>

Stochastic Gradient Descent, epoch, and batch

1. Stochastic Gradient Descent:
   1. An optimization algorithm to train ML models (ANN in deep learning etc.)
   2. Goal: to find a set of internal model parameters

Performance measure such as logarithmic loss or mean squared error

* 1. Optimization means searching process
  2. Iterative: means each discrete step, including calculating, comparing, and updating
  3. Update procedure: depends on different algorithms. For ANN is backpropagation update algorithm

1. Sample:
   1. Contains inputs and outputs
   2. Other names: an instance, an observation, an input vector, or a feature vector.
2. Batch:
   1. The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.
   2. “for loop”
   3. A training dataset can be divided into one or more batches.
   4. batch gradient descent VS stochastic gradient descent
3. epoch:
   1. The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset.
   2. a for-loop over the number of epochs where each loop proceeds over the training dataset. Within this for-loop is another nested for-loop that iterates over each batch of samples, where one batch has the specified “batch size” number of samples.
4. Epoch VS batch:
   1. The batch size is a number of samples processed before the model is updated.
   2. The number of epochs is the number of complete passes through the training dataset.

Paper reading: A Benchmark for Interpretability Methods in Deep Neural Networks

Abstract:

Problem definition:

1. Understanding what features are important helps improve our models, builds trust in the model prediction and isolates undesirable behavior.
2. A commonly used strategy is to remove the supposedly informative features from the input and look at how the classifier degrades.

this approach clearly violates one of the key assumptions in machine learning: the training and evaluation data come from the same distribution.

1. ResNet-50 experiment was implemented
2. ROAR=RemOve And Retrain. For each feature importance estimator, ROAR replaces the fraction of the pixels estimated to be most important with a fixed uninformative value.

Tutorial: Develop RNN models for human activities recognition time series classification

Link: <https://machinelearningmastery.com/how-to-develop-rnn-models-for-human-activity-recognition-time-series-classification/>

1. Problem definition:

To classify sequences of accelerometer data recorded by specialized harnesses or smart phones into known well-defined movements

1. Solutions:
   1. hand crafting features from the time series data based on fixed-sized windows and training machine learning models, such as ensembles of decision trees.
      1. The difficulty is that this feature engineering requires strong expertise in the field.
   2. recurrent neural networks (like as LSTMs and variations) that make use of one-dimensional convolutional neural networks or CNNs.

They perform well on challenging activity recognition tasks with little or no data feature engineering, instead using feature learning on raw data.

1. Main Contents: three RNN architectures for modeling an activity recognition time series classification problem

Long Short-Term Memory Recurrent Neural Network for human activity recognition.

a one-dimensional Convolutional Neural Network LSTM, or CNN-LSTM, model.

a one-dimensional Convolutional LSTM, or ConvLSTM, model for the same problem.

1. Human Activity Recognition (HAR)

* Definition: to predict what a person is doing based on a trace of their movement using sensors.

Example dataset: 30 subjects; video recorded for each subject performing the activities and the movement data was labeled manually from these videos.

Labels: Walking; Walking Upstairs; Walking Downstairs; Sitting; Standing; Laying

Raw Data: the x, y, and z accelerometer data (linear acceleration) and gyroscopic data (angular velocity); 50 Hz (50 data points per second)

70-30% train-test split

1. LSTM: are able to learn and remember over long sequences (time steps) of input data.

The model can support multiple parallel sequences of input data, such as each axis of the accelerometer and gyroscope data.

1. Preprocessing and Loading Video data **(IMPORTANT)!!!**
2. Convolutional Neural Network (CNN) layers for feature extraction on input data

**Convolutional LSTM and CNN LSTM:**

1. **spatio-temporal data:**

**Noted that LSTM reads the data directly in order to calculate internal state and state transitions**

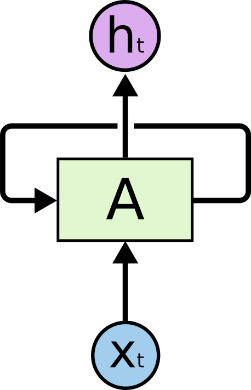
**CNN LSTM interprets the output from CNN models**

**Convolutional LSTM uses convolutions directly as part of reading input into the LSTM units themselves. (**[tutorial link](https://arxiv.org/abs/1506.04214v1)**)**

Tutorial: understanding LSTM networks

Link: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

1. Recurrent Neural Networks:



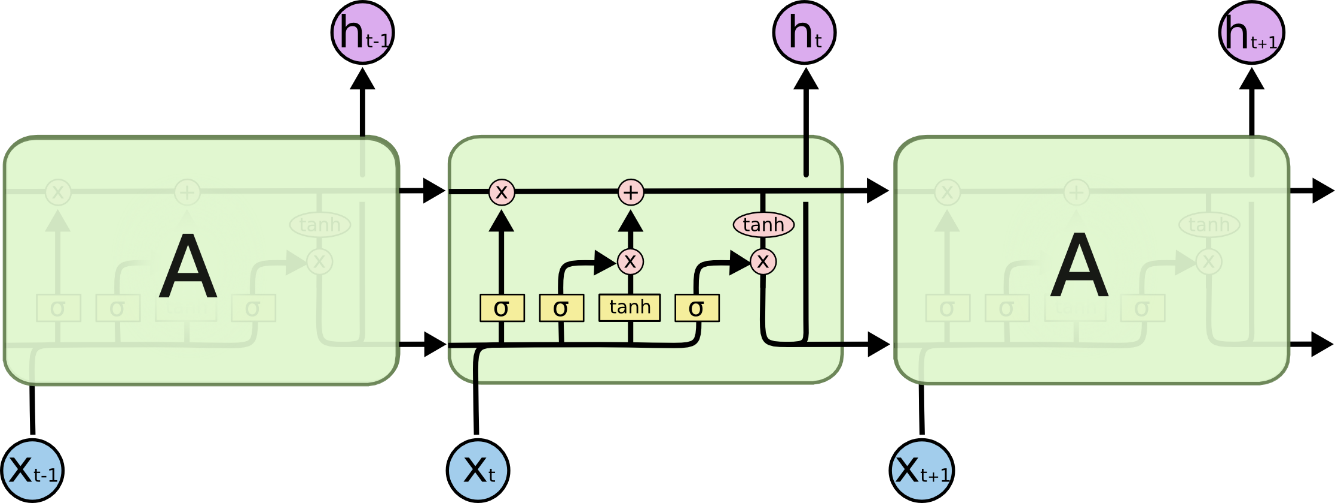
[Application fields](http://karpathy.github.io/2015/05/21/rnn-effectiveness/): speech recognition, language modeling, translation, image captioning, etc.

Feature: connect previous information to the present task.

Disadvantages: in case of the gap between the relevant information and the place that it is needed is large, RNN is unable to learn to connect the information.

1. LSTM networks:

A special type of RNN, capable of learning long-term dependencies.



Noted that figure above is just one version of LSTM. There are lots of LSTM variants that tackle long-term dependencies.