NIFTY Assignment - Sentiment Analysis

Saxon Knight
University of Hawai’i Maui College
University of Hawai’i at Manoa

Table of Contents

Background

Sentiment Analysis may be performed as an application of Machine Learning (ML) to large bodies of text, such as those found in large consumer review datasets, in order to determine sentiment (positive, negative, sarcastic, etc.) and gain feedback. The use of Machine Learning techniques in this endeavor allows for much larger quantities of data to be processed than would be practical for human evaluators working directly with the data. With recent advances in Machine Learning in the form of new and powerful frameworks, it is relatively simple to set up a machine to perform analysis on text in a way previously confined to the domain of common-sense, human interpretation of opinions, feelings, etc.

Meta Information

Summary	Sentiment Analysis -- Create and use a neural network model which is capable of inferring positive or negative sentiment from strings of coherent text.
Topics	File I/O, parsing text, various machine learning topics.
Audience	Students who are familiar with Python and interested in Machine Learning or Data Science.
Difficulty	This is an intermediate to advanced assignment distilled into a high-level step-by-step with room to innovate. A student familiar with the frameworks involved may finish in a day or less, while others may need to spend more time researching. A good working knowledge is well-earned and allows intuitive tuning of hyperparameters and potential implementations of more advanced mechanisms (RNNs, LSTMs, etc.) to improve performance.
Strengths	Introduction to an application of Machine Learning, exercises file input/output, string parsing, working with arrays, and use of third-party frameworks.
Weaknesses	May be difficult for the student to grasp the inner workings without introduction and

	overview. In the endeavor of improved performance, heavy reliance on independent research and effort to gain a working knowledge of the concepts involved.
Dependencies	TensorFlow, Keras deep learning library (included in TensorFlow), and sentiment-labelled dataset(s) (provided).
Variants	Focus on file I/O, string parsing, and data massaging/preparation.

Handout

Assignment Instructions

1. If not already present, you will need to install Python (3.x preferred) and TensorFlow (GPU version recommended but not required).
   1.1. https://www.python.org/downloads/
   1.2. https://www.tensorflow.org/install/
2. Select one or more datasets to train on from any source available. The goal is to find a mapping from pieces of text to a kind of sentiment (binary [0 negative, 1 positive] is ideal). The IMDb, Amazon, and Yelp review datasets are a great choice.
   2.1. http://ai.stanford.edu/~amaas/data/sentiment/
   2.2. https://s3.amazonaws.com/amazon-reviews-pds/readme.html
   2.3. https://www.yelp.com/dataset
3. Massage the datasets into a consistent format so that your program may easily read them in (e.g. datum<tab>label).
4. Load in the datasets (data and labels).
5. Separate and store the data and labels into variables (mappings to sentiment).
6. Optionally make helper functions to clean up the data for use (remove artifacts such as numeric codes and the like).
7. Build your neural net model.
   7.1. A relatively straightforward word2vec based approach may use the Keras library (included with TensorFlow). To do so, you will need to create a unique vocabulary list from the pieces of text and associate each word in the vocabulary list with a numeric ID (the index of the word in the list should work fine). You should also include a special value for unknown words (words not present in the vocabulary list) that may be encountered when you attempt inference on arbitrary input.
   7.2. Prepare data and labels for training: encode the text data into numeric IDs and pad the sequences (each sentence or so) so they all have the same length.
8. Train your model.

9. Make a function to use your model for inference on input.
10. Test it out on some arbitrary statements with known or intended sentiment.
11. Play around with hyperparameters to see if you can improve performance accuracy.

Sample data files

Data used for this assignment may vary, but for a successful attempt at inference on sentiments, datasets which are used should be well curated and have many examples for the neural net model to learn from. Some examples are the IMDb, Amazon, and Yelp reviews datasets. The datasets included in this assignment are a combination of those, which was originally for a paper by Kotzias et. al.
https://www.kaggle.com/marklvl/sentiment-labelled-sentences-data-set/version/2#.

Starter and support code files

Starter / support code for this assignment is available in the GitHub repository at https://github.com/UHMC/nifty-sentiment-analysis.

Model grading criteria

Upon successful completion of this assignment, the student's program should demonstrate reasonable accuracy in determining the sentiment (positive, negative) of a statement similar to those used in the training datasets. Getting far enough to perform any inference at all, whether or not it is accurate, still demonstrates a level of competence in file input/output, string parsing, mapping of one datatype to another using arrays, use of third-party frameworks, and a reasonable effort. In some cases, all that is necessary to improve performance is the tuning of the neural network model hyperparameters (number of hidden neurons, dimensionality of word embeddings, etc.).

Runnable demo application

A runnable demo application is available in the GitHub repository at https://github.com/UHMC/nifty-sentiment-analysis.