Uncovering Sentiments using EDGAR Datasets

Author: Sri Krishnamurthy, CFA, QuantUniversity

Notes:

1. Use Python and Keras with a backend of your choice for this project
2. Assignment is due Nov 27th midnight

Case

Sentiment analysis is becoming more and more popular with availability of alternative datasets such as Twitter etc. Companies like Amazon, Yelp, Uber etc. routinely analyze user reviews to understand user sentiments. With chatbots and conversational agents becoming more and more popular, sentiment analysis is gaining mainstream acceptance. Building a sentiment analysis engine from scratch is a challenge. Traditionally, bow models, word embeddings using glove/word2vec have been popular and the Movie review database (http://ai.stanford.edu/~amaas/data/sentiment/) is used to illustrate these concepts. In the last few years, Deep Neural networks and RNNs have been tried for sentiment analysis. Many cloud vendors are offering sentiment analysis APIs[2,3,4,5]. Sentiment analysis is particularly challenging in domains such as finance. Financial Sentiment analysis is a challenging problem considering it is domain specific and there is lack of good labeled data to train. Traditionally, analysts conduct in-depth fundamental analysis and issue guidance and reports which are used by financial professionals for investment analysis. In the last few years, with the growth of social networks and alternative means of getting information about market events, using Alternative data to evaluate sentiments is becoming popular. Twitter, EDGAR and news articles are typical data sources for information and for evaluating sentiments.

In this case, we will review some methods to evaluate sentiment analysis techniques.The goal of this project is to build a sentiment analysis engine that can be used to understand the sentiments of earnings call transcripts.

Data:

1. Obtain the latest earnings call transcript (Q3 2018 for the FANG stocks Google(1), Amazon(2), Facebook(3), Netflix(4), Microsoft(5)). You can get it from seekingalpha.com or EDGAR (sec.gov)
2. Parse it and chunk it to paragraphs.
3. Each team(numbered 1-5) will read the transcript and assign a +/-/Neutral for each paragraph.
4. Your output should be a json formatted file (See document.json attached.) You should post your properly formatted json file to the folder (See blackboard) **by Wednesday Nov 14th**.

Experiments:

We will try 3 types of models. See [1] for examples on how you could build these models.

1. BOW model
2. Word embeddings (GLOVE)
3. RNN

Experiment 1

1. Mix all the paragraphs and split the data to a 80-20 split
2. Build the 3 models listed earlier and compute the confusion matrix for the training and testing datasets.
3. Discuss how your models perform.

Experiment 2: Transfer learning

The data we have is still small. We want to see if we can we can use another larger dataset to learn and test using our datasets and the labels we created. To do that:

1. Build the 3 models listed earlier using the Movie review dataset [6] or from Keras. Compute the confusion matrix for each model
2. Use the 3 models to predict the sentiments for the financial dataset. And compute the confusion matrices for each model.
3. Discuss how your models perform wrt to the training dataset.

Experiment 3: Using APIs

1. Using the Amazon, Google, Microsoft and Watson APIs [2,3,4,5], obtain the sentiment scores for your entire dataset.
2. Normalize the scores and take the average normalized score to determine the sentiments
3. Compute the confusion matrix wrt the original dataset and discuss your results.

Experiment 4: Ensemble learning using AutoML

1. Instead of simple averaging, you wonder if you could build a model that can map the **raw(not normalized)** results from the 4 APIs to the outputs you labeled. i.e.
   a. Inputs: Amazon, Google, IBM, Microsoft scores
   b. Outputs: Sentiment scores you labeled
2. Use TPOT, AutoSKLearn, H2O.ai's APIs and choose the best model.
3. Discuss if this model was better than the metrics you got from experiments 3 (simple averaging)

Final model:

1. Discuss which model from which experiment would you use in production.

Test:

1. Obtain the earnings call transcript for Tesla.
2. Parse and format the file in the json format described before
3. Each team should read the transcript and label the paragraphs with (+/-/neutral) sentiments.
4. Pass this file to the final model you selected. How does it perform (Show confusion matrix)?
5. On November 25th, share your labeled json for the Tesla report with the class.
6. Pass the 4 other files (from other teams) to your model. How does it perform (Show confusion matrix)?
7. Discuss what you learnt from this exercise.
8. Submit all results, python files and report through a google drive to analyticsneu@gmail.com

PREPARATION:

1. REVIEW EXAMPLES HERE: https://github.com/fchollet/deep-learning-with-python-notebooks
2. https://cloud.google.com/natural-language/docs/sentiment-tutorial
3. https://www.ibm.com/watson/services/natural-language-understanding/
4. https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/
5. https://docs.aws.amazon.com/comprehend/latest/dg/API_Reference.html
6. http://ai.stanford.edu/~amaas/data/sentiment/