Subjective Sentiment Analysis with LSTM Networks

Michael Bremner & Justin Brown | May 3, 2019

Abstract

We present an approach to sentiment analysis that seeks to encode the personal subjective opinion of an individual in an LSTM recurrent neural network model. After training a model to learn the subjective sentiment of an individual, the model is applied on previously unseen text samples to rank the samples based on their similarity to the learned sentiment in a feature space. We carry out analysis whereby a set of text by a single author serves as the ground truth sentiment. Our approach supplies a foundation for forthcoming sentiment mining work. The techniques and lessons developed serve an ongoing motivation to mine for interesting or insightful comments based on an abstract personal encoding of the user's definition of useful/clever/insightful. The primary restriction on our approach is the large task of finding and tagging comments that meet a personal thresh-hold of useful or interesting. In the near future, we expect it will be easier to tag and maintain inventories of text data. In the meantime, we supply examples where a single author (ex. Donald Trump) is used as a subjective sentiment anchor. We are then able to mine text data and score comments on their similarity to Donald Trump's use of language. Unreported is an additional case study where the subjective anchor is provided by a collection of personal Youtube comments by one of the authors. The motivation for this work is every comment that we wish we had data-based because we thought it was simply brilliant. We do not have the data we wish we had saved at our fingertips, but we can still build the model we wish we'd have once we have the data.

1. Introduction

Many Sentiment Analysis Models seek an objective definition of sentiment analysis. For example, the LSTM [1] that our model is based off of was initially trained to decipher positive from negative reviews. The sentimental semantics of the individual are averaged with the group and the personal aspect of the opinion is lost.

This work explores an alternative approach, whereby the ground truth for sentiment is defined by the subjective position of a single individual. Any individual may be used as such a subjective anchor, provided the availability of a large enough corpus of personalized text. A model is trained to separate text comments by the ground truth individual from text comments made by anybody else. Untagged text comments are fed to the model & their features are scored & ranked by similarity to the learned ground truth class center.

The goal of the aforementioned process is to establish a foundation for Text Data Mining on the basis of entirely abstract subjective definitions of personal sentiment. We provide a pseudo-subjective example of subjective sentiment mining whereby the ground truth is represented by the authorship of the target individual. Provided a sufficient corpus of text data by a single known author, we study how effectively an LSTM may be deployed to retrieve comments similar to the author.

This work sets a foundation for future work, where the ground truth is established by a dataset of comments tagged by an individual based on some arbitrary personal notion of sentiment. (Ex. People who I don't like when they talk

that way, obvious children, mean-spirited folks, people I think are bots/trolls etc.). The goal is to achieve contextual disambiguation by encoding the personal sentiment interpretation mechanism of the user. That is, all context is resolved by the individual of interest's personal linguistic preferences.

There is a necessary qualitative component of analysis involved at the very end of our process. We feed untagged text into a trained LSTM model and retrieve features. The features are processed and ranked on the basis of the learned class center of interest. The ranking must then be assessed qualitatively based on how well we, the used, believe the model achieved the desired goal.

2. Related Works

Our approach is based on Recurrent Neural Networks. In this type of network, the data is looped over the same nodes/weights a certain amount of times. The benefit of doing this is that a single node can analyze a sequence all at once, rather than one at a time. We use Recurrent Neural Network in order to extract semantic features from temporally-ordered data. In our case, text data, which consists of a wide array of temporal dependencies, some easier to identify than others.

Although Recurrent Neural Networks work well for temporal data, they suffer from the Vanishing Gradient Problem [1][2][3]. When more than one word is fed into a Recurrent Neural Network, by the chain rule, the next word in the sequence decreases the magnitude of the gradient of every word that came prior. The weights in the prior layers, which the next layer depends on, become too difficult to train, and the entire model fails to converge.

For sequences that are short enough the network may be trainable, but the model will likely exhibit a recency bias toward the last few words in the sequence. For example, given the input sentence “I hate you, I hate everything, this show is garbage man I love delicious tasty beer”, an RNN might indicate positive sentiment despite the obvious negativity at the beginning of the sequence. Long Term Short Memory (LSTM) Networks are one solution to recency bias and vanishing gradients

LSTMs are a form of Recurrent Neural Network, except each node has 3 additional trainable-parameters. These are the input gate, output gate, and forget gate. In a traditional RNN, the input goes into a node, and the output always flows out of the cell. In an LSTM, the three gates decide whether or not the information coming in is important or not. If it is important, the cell will remember that data, and the output re-enters the loop. This has the effect of allowing the important parts of a sentence be focused on, while filtering out irrelevant parts. The forget gate is also important as it allows the node to decide whether or not the past information is still relevant for the new information. For instance, if the network encounters a period that indicates the end of a sentence and the start of a new sentence, perhaps it needs to forget the last sentence and focus on the new one. Most importantly, for text & language research especially, LSTMs offer a solution to recency bias and the Vanishing Gradient Problem by remembering different parts of sequences for various sequences lengths. LSTMs still aren’t perfect, however, and

can still only hold so much information in its cells at once, similar to how humans can only hold so many numbers in their head at once. For that reason, we limit our sentences to a maximum of 500 characters.

In [5], they use LSTMs to capture context sensitive information from a well-known IMDB dataset and attempt to predict the positivity score of reviews based on LSTM generated features. Their work forms the basis of our approach for generating LSTM sentiment features. Our work varies in that the ground truth of our model is the individualized taste of the end-user; whereas in [5] the ground truth is an average of all opinions (essentially attempting to infer global models / definitions of sentiment.)

3. Methodology

Our overall objective here is to give our network a comment, and have it predict whether it is a Trump tweet or a YouTube comment. However, the goal of our methodology is not necessarily so much to maximize accuracy. The goal is to modify the network parameters in a way that separates the data interestingly. As such, there is an element of qualitative feedback to the model development process. We produce a model, examine the features visually, and assess whether or not the top ranked sentences are useful or representative of what we are looking for.

3.1 Data Collection

Our two main sources of data came from YouTube comments, and President Trump's twitter. In order to collect a large amount of random YouTube comments, we made use of the YouTube API v3. This API allowed us to scrape over 400 000 individual comments from a variety of prominent sources:

> *PewDiePie ,SMOSH, CollegeHumor, FailArmy, JennaMarbles, Fallon, Conan, Corden, Kimmel, Ellen, AsapSCIENCE, SciShow, Numberphile, ScienceChannel, Veritasium, TYT, ABCNews, CNN, FoxNews, AlexJones, Vox.*

The corpus of Trump Tweets is provided by www.trumptwitterarchive.com, providing us access to over 33 000 Trump Tweets.

3.2 Cleaning & Data Evaluation

One of the biggest challenges we face during this project was cleaning and sanitizing the data. The main issue came from the fact that the two dataset came from entirely different mediums. YouTube and Twitter comments have some intrinsic differences between them that make them easily identifiable without even considering the meaning or sentiment of the comment itself. For instance, YouTube comments have timestamps in the form of "minute:second", which Twitter comments rarely have. Conversely, Twitter make use of the @-notation to reference other users, which YouTube doesn't have. These identifiers cause the network to ignore the subtle differences we are interested in teasing out. We may observe the difference in learning by examining a PCA projection of the features (Figure 1).

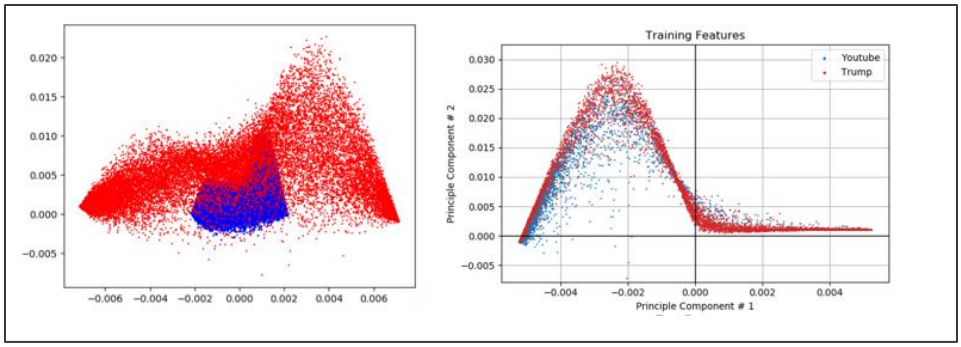


Figure 1 Contrast of sigmoid feature spread, unrestricted vocab (left), restricted vocab (right)

The plot on the left depicts features extracted from a network trained on a vocab which maintained most all domain unique identifiers (hashtags, @s and timestamps). With the identifiers, the network is able to learn a relatively clean separation between YouTube comments and Trumps tweets, compared to the separation on the graph to the right. If we wanted to just maximize accuracy of our classifier, we would leave in the identifiers; however, we wish for the network to learn more than simply the lowest hanging fruit that is available in the data. Interesting separations of the data are preferred to trivial ones. For that reason, we seek to remove obvious identifiers. Other identifiers we removed include: fancy uni-code punctuation, foreign-language symbols, several forms of ascii-art, and extremely long numbers.

3.3 Training

The core of our network is an LSTM that was pre-trained on the IMDB dataset with the WikiText-2 vocabulary. At the end of our network are two fully-connected layers which are used to classify the sentences and a single binary classification node. The subjective anchor always forms the positive (1) class while the contrastive set (in this case Youtube comments) makes up class (0). The network is trained binary cross-entropy loss with a learning rate of 0.005 in all cases.

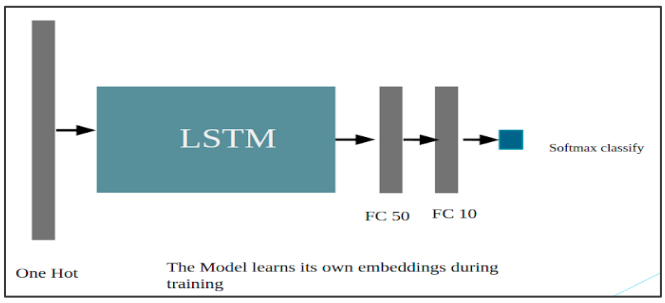


Figure 2 Our LSTM Model for Sentiment Classification

We use the pre-trained features to initialize our own model. Pre-trained LSTM features allow the model learn primary the difference between the two classes, while leveraging a great deal of linguistic semantics already embedded in the features.

3.3.1. Sigmoid vs ReLU activation

The choice of activation function in a neural network is crucial to how well information flows through the network. The sigmoid function produces a dense feature representation between 0 and 1, whereas ReLU produces an unbounded sparse feature representation between 0 and positive infinity. From our experiments, we found that the ReLU activation tended to separate the two classes in the feature space better than the sigmoid activation function. The ReLU function constricts the features to the same quadrant. Unable to spread in any direction anymore, the features tend to form an angular separation which lends itself to the cosine similarity metric.

3.3.2. Size of FC Layers

The LSTM outputs a feature vector of length 200. We add two dense layers on the end to capture the features and classify. The first layer is always *sigmoid* activated, while the final feature layer may by *sigmoid* or *ReLU.* We started with a large number of nodes on each layer (200 & 100) and reduced the number of layers gradually, observing little to no change in representative accuracy. We settled on lengths of 50 and 10, adequately reducing the memory footprint of the features.

3.3.3 Regularization

Several methods of regularization are considered. We added weight decay to the model, which drastically reduced separation of classes in the feature space, which made similarity evaluation near meaningless. We decided that over-fitting is not a problem and that weight-decay is prohibitive for our data. For LSTM networks, dropout regularization is an appropriate choice over batch normalization. Batch normalization is messy when training an LSTM network because of the bucketing process used to unroll several networks during a single batch. It is not recommended to batch-normalized across variably sized networks. The dropout rate during all training is 0.50.

3.4 Extract & Evaluate Features (Quantitative & Qualitative analysis)

We are interested in features which exhibit interesting groupings of the comment data. We apply Principal Component Analysis (PCA) to generate a visually observable representation of the feature space. PCA searches for a lower dimensional component basis that captures as much variance as possible that exists within the higher dimensional vector data. We map our length 10 feature vectors to a 2D vector space, compute class centroids, and evaluate the similarity of comments by cosine similarity. In order to do this, it is crucial that the features exhibit some clear separation, otherwise the similarity measure is meaningless.

4. Results

We evaluate the quality of the data separation as well as the classification of untagged comments against the learned class center of Donald Trump.

Sigmoid Features

The figure below plot shows the length 10 feature vectors mapped to a 2D space. Each point on the plot represents a comment from either Trump or YouTube. There are many interesting components to these figures. There is a strong separation with nearly 90 % of the Trump comments clearly separated from everything else. This is clear enough separation to apply cosine similarity around the class centers.

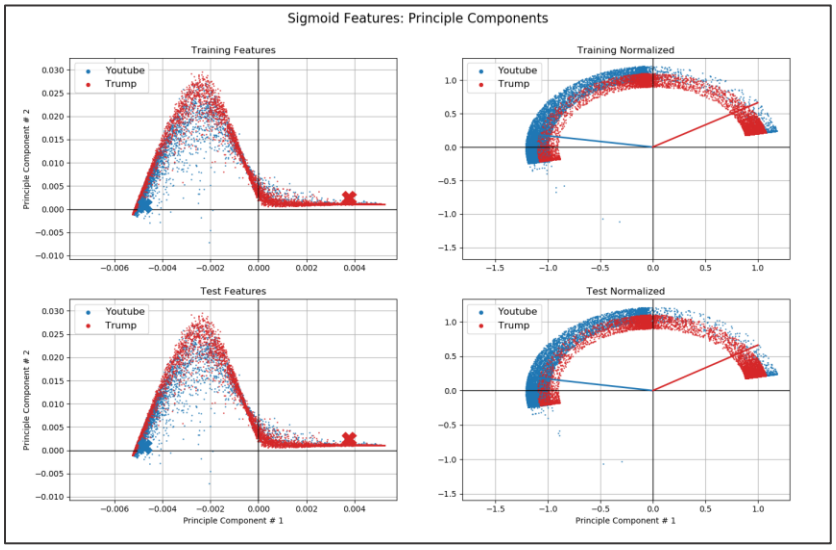
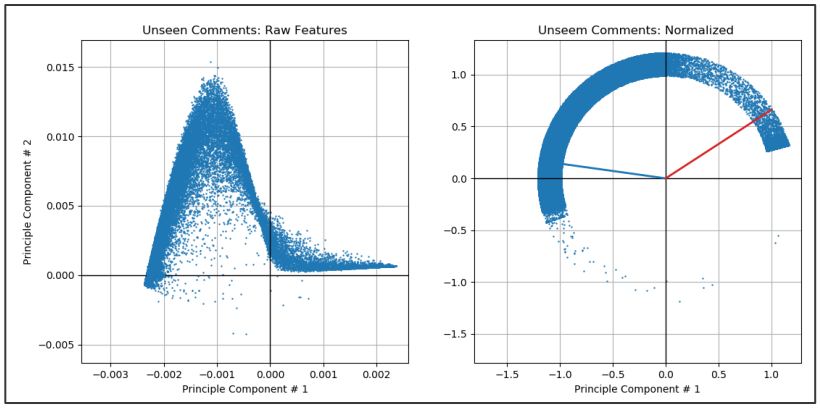


Figure 3: Test & Training Features, Below: Unseen Comments (from Youtube)



Top & Lowest Ranked Comments:

0.999	*i clicked on this because i 'm fat elipsis this did not help but still good video ðŸ˜Œ elipsis ðŸ˜- ðŸ˜- ðŸ˜- ðŸ˜- ðŸ˜-*
0.999	*um elipsis they had us solve that in high school elipsis took most of the class elipsis 5 seconds*
0.999	*can you make a video explaining how dog breeds are different than species cause i 'm really confused on how a great dane is technically the same animal as a chihuahua*
0.999	come on elipsis pandas are begging to go extinct
0.999	*i think the same study that showed more pain and heat pain sensitivity of redheads also showed slightly less sensitivity to electrical pain reception if i remember correctly . really need to read that*

up . anyway next time i hurt myself or don 't wanna wash dishes in hat water i got the perfect excuse to whine : d

---

-0.999 *so beautifull jen . .*
-0.999 *you learn something new everyday . .*
-0.999 *did you smile or did you lose*
-0.999 *you or right*
-0.999 *feels like brooklyn 99*
-0.999 *so mike wasn 't with him*
-0.999 you 'r e the bomb dude . . wow

The top ranked sentences generally would fit into a tweet while the lower ranked ones are shorted. The short stubby comments are simply not Trump's style. There is nothing particularly interesting sentimentally about the way the text is separated.

ReLU Features

We now evaluate the same data on an LSTM trained with a ReLU activation on the FC 10 layer. The red and blue radial lines on the normalized plot lines represent the centroids of each class. Both ReLU and sigmoid give similar performance in terms of classification accuracy; however, ReLU shows a more interesting tendency to separate the data. Having nowhere else to go, comments are forced along a boundary at 90 degrees (some exceed 90 degrees as an artifact of PCA).

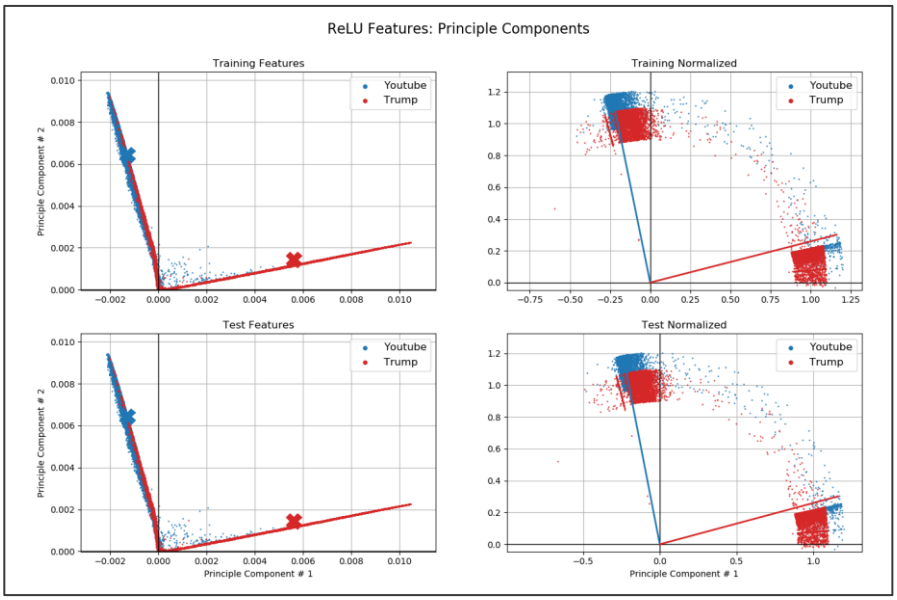
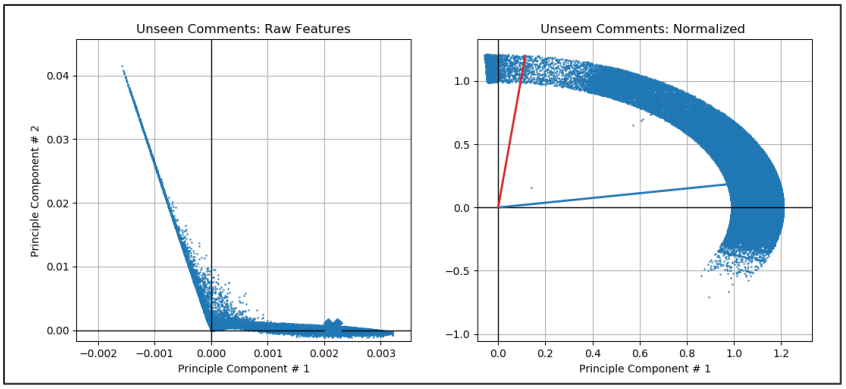


Figure 3: Above: Test & Training Features, Below: Unseen Comments (from Youtube)



Top & Lowest Ranked Comments:

0.999	*wow people just need to find a flaw in everyone don 't they i mean really complaining about the bags under her eyes and saying they won 't get her anywhere or anyone neither will that negative attitude : d*
0.999	*make more dank memes those are the best*
0.999	*omg is that the boy from stranger things edit : please stop replying yes i now clearly know it is in fact him but thank you for the extra confirmation*
0.999	*lucky find elipsis most missing people end up dead*
10.999	*diet is incredibly important . lactose intolerance sensitivity to certain foods like peanuts deficiency in zinc or vitamin d are just some of the causes .*
0.999	*next up elipsis fidget spinner spins faster than the speed of light*
0.999	*b . as you are increasing the downward force of the ping pong ball but keeping the acrylic ball the same as the previous experiment*

-0.04	*wow ðŸ˜® wow elipsis just wow*
-0.04	*white people . lol*
-0.048	*trevor so fine*
-0.052	*c . just guessing*
-0.057	*man cordan is killing it . really great great for the late late show . btw put this on itunes and i buy it*
-0.057	*orange man bad*
-0.060	*wow . . cool*
-0.061	*c . balanced*

Not shown (for space), below the bottom of the list is a collection of the longest and most complex sequences in the dataset which are ranked even more dissimilar to the Trump center than the short ones. It is clear that the model is learning to classify based on the length of the sequences. All moderately long comments with a decent bit of sense and structure have a chance to be classified as Trump. Despite extensive pre-processing and careful sampling, our model learned only to separate the most abundantly clear aspects of each dataset. Please see the appendix for a full documentation of ranked sentences.

5. Conclusion

This undertaking began with high ambition and ended up serving several difficult lesson in cross-domain text analysis. We found that when we under-regularized our model and applied minimal cleaning to the data, the model found trivial ways to separate the data. When we applied heavy regularization and over-pruned the text, the model had difficulty splitting the text. It is difficult and arduous to strike the necessary balance that delivers interesting results. The considerations given in the pre-processing stage are absolutely critical to the quality of the results. We posit that we did not observe meaningful or interesting separation in the data because we were unable to impose a meaningful constraint on the training process. For all we know, the model learned the difference between the two domains rather than the difference in authorship. The network needs motivation to learn in a meaningful way so that it does not learn unintended lessons. Though the results are mostly disappointing, we've gleaned a far richer understanding for text processing with neural networks. Future work will seek to incorporate +/- sentiment prediction data with that of a subjective anchor (such as Donald Trump), thereby blending the global and local definition of sentiment as a means to motivate training. Finally, we intend not to tackle any more cross domain tasks (Twitter vs Youtube data) until we've done successful work with data from like domains.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term memory. Neural computation, 9(8):1735–1780, 1997.

[2] Sepp Hochreiter, Bengio Yoshua, Frasconi Paolo, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer and Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, 2001.

[3] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning, pages 1310–1318, 2013

[4] Bo Pang and Lillian Lee, Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrieval, 2008. Describes approaches and challenges to building a sentiment mining application

[5] Sentiment Analysis with Long Short-Term Memory networks Research Paper Business Analytics Vrije Universiteit Amsterdam Author: Fenna Miedema Supervisor: Prof. dr. Sandjai Bhulai August 1, 2018

[6]Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network Alex Sherstinsky Directly Software, Inc.

[7] Distributed Representations of Words and Phrases and their Compositionality (Mikolov et al. 2013)