Modelling Valence and Arousal in Facebook posts

By Shane Talbot, Nathan Kelderman

# Introduction

Sentiment analysis is a large area of research, in which researchers are studying and implementing methods with which to determine the implications of text. This topic has multiple use cases ranging from product feedback to political campaigning. This is a great task for machine learning as the models that can be designed are able to learn intricate details about language that humans don’t always pick up on. There are two main theories regarding emotion in psychology: the first assumes that emotions are discrete and finite, in contrast, the second suggests that emotions lay on a range of different scales. In general, sentiment analysis research has been focused on the former definition of emotion.

As such, we wanted to replicate the works of H. Andrew Schwartz et al. in the paper *Modelling Valence and Arousal in Facebook posts* as well as attempt a few methods to take the study further. Schwartz’s purpose for the paper was to introduce a new dataset as well as train a model on the data. The results could then be further applied for situation such as mental illness detection and large-scale psychological studies.

The dataset for this problem is a set of posts taken from Facebook and annotated with a valence and arousal score by independent raters with training in psychology. The goal is to generate a couple models to determine if the study can be taken further and expanded upon. The output of these models will be accuracy as well as r-values, which can be compared to the results of the models from Schwartz’s study.

# Dataset

The dataset used consists of 2895 Facebook posts. The data was carefully annotated by 2 professionals and both annotations included in the dataset. There were certain steps taken to ensure the data did not include the real names that were in the posts and that the writers of the post were anonymized. There was a variety of ages and genders used as authors and no 2 posts were written by the same author.

The values used to annotate the messages were valence (sentiment) and arousal (intensity), both on a scale of 1-9, with 1 being negative for valence and neutral for arousal, and 9 for positive for valence and very high for arousal. The data was split into training and test data with an 80/20 split. It was then processed using a variety of techniques. The first was to make all the messages lowercase. This creates an even playing field for all words to have a face value as well as reduced the amount of unique words and therefore greatly decreasing training time. The second was to apply a regex to remove anything that wasn’t a letter or a number, as these do not affect the valence or arousal of a sentence. The third and final technique was to remove all the stop words from the messages. For the Bag Of Words (BOW) model, the next step was to create a bag of words that the model could then use. This was done using the sklearn CountVectorizer function. For the RNN, we tokenized and only kept the top 2000 most popular words. The reason for picking out the top 2000 words is that at a certain point, words can be meaningless to a model in training. If, for example, a word is only used once in the whole dataset, the importance of that word cannot be determined with a significant amount of certainty. Whereas if a word is used many times in a dataset, then that word can be linked to a specific output with a higher degree of certainty.

TODO: talk about the data balancing.

# Models

We decided to attempt 2 models in our study. The first model is an RNN model. The network was a sequential network with the following layers: embedding layer, a dropout layer with a dropout of 0.3, 2 LSTM layers and a dense softmax activation layer. The model was compiled using the Adam optimizer and used the categorical cross entropy loss function. The BOW model used a simple logistic regression classifier.

Both models output an accuracy score and an r-value. We also used KFold validation on both of the models with a split of 10. What this does is it splits the data up *k* times (10 in this case) and it will run through 10 training cycles, each time picking a different set of as the validation set and using the rest to train. For example, the first run would take set 1 as the validation set and train on sets 2-10.

# Results

When starting with the RNN, we struggled to get a working model that produced desirable outputs. The accuracy was typically around 40% for valence and 20% for arousal. We tried numerous different tactics to increase this, ranging from preprocessing the data differently to adding/removing layers and hyper-parameter tuning, but these only gave marginal changes in the outputs. We then tried to recreate a similar model that Schwartz created, a BOW model. This did not produce any better results. We then noticed that the outputs for a given message typically centered around certain values. For valence it was 5 and for arousal it was 2. When plotting the outputs against the output on a heatmap, the following image was given:

Chart, histogram

Description automatically generated

Figure : Valence Outputs

This shows that the guesses for valence were almost all 5, with a few being 4/6 and less being 3/7 and none being 1,2,8,9. This was when we realized that the dataset was skewed. We decided the data needed to be balanced, as described in the dataset section above. After balancing the data, we obtained a 20% increase on both valence and arousal outputs. These values were now comparable to Schwartz’s results for arousal and even exceeded the results for valence. Schwartz obtained an r-value of .85 for arousal and .65 for valence. After balancing the data, we achieved an r-value of 0.76 for arousal and 0.94 for valence. Both the RNN and the BOW models output similar values after balancing the data.

# Conclusion

In conclusion, we attempted to use the data from Schwartz’s study to recreate the results and try to push them even further. We spent a lot of time trying to get our models to work better. We realized though that it wasn’t specifically our models that weren’t working, but that the data was skewed and disproportionate. After adjusting the, we were able to achieve much better results and even in some cases better than the results we were trying to match.