Predicting sentiment on 280.000 reddit posts with EDA, graphs, and machine learning

Julian Scheuchenpflug

Abstract

This document contains an in-depth analysis of Stanford's Reddit Hyperlink network, describing the data with explorative data analysis and analyzing its sentiment using graph analysis and machine learning techniques, as well as discussing those findings.

1 Introduction

The underlying data provided for this project is a Reddit Hyperlink network from the SNAP library. This library contains numerous datasets concerning social networks as a "result of [their] research pursuits in analysis of large social and information networks"[1]. In this work, first the data will be described using common methods of explorative data analysis. Then, using a provided gold standard for testing results, the sentiment of the posts will be predicted with methods of graph interpretation as well as common machine learning methods. The results of the latter will then be statistically tested do discern the "best" model regarding the classification of this data. The results and findings will then be discussed. All Figures will be provided on the repository, for readability's sake.

2 The codebase

The codebase to this project can be found here[2]. Some of the train, test and result data has been excluded from this repository due to its sheer size, but can be downloaded here[3]. The code consists of three main notebooks, EDA, sub1_graph and sub2_ML_statistics, as well as the helper script explode_data. The latter was only used to augment the provided data and is just for demonstration purposes, as running it takes rather long and the resulting data is provided. The notebooks contain a lot more derived data than described here, so might be worth a closer look.

3 Explorative data analysis

3.1 Posts and sentiment

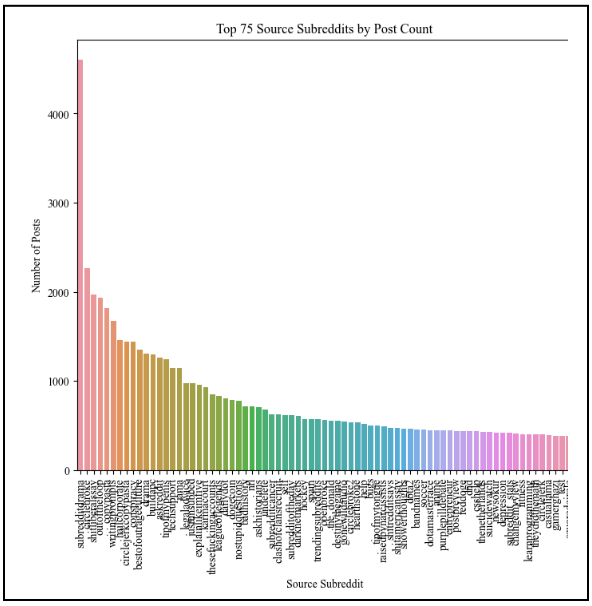

Figure 1: Top 75 source subreddits sorted by number of posts.

The provided data contains 281562 entries of posts from one subreddit referencing another between January 2014 and April 2017. These entries contain the source subreddit, target subreddit, sentiment of the post or message referencing the target, the post id, a timestamp, and a plethora of post properties as listed on the dataset

[1] http://snap.stanford.edu/about.html

[2] https://github.com/niceshice/DS4DH_Abg

[3] https://drive.google.com/drive/folders/1S_wDLVG0l4p2wgRvfu7Nw8Es-_ygs_08?usp=drive_link

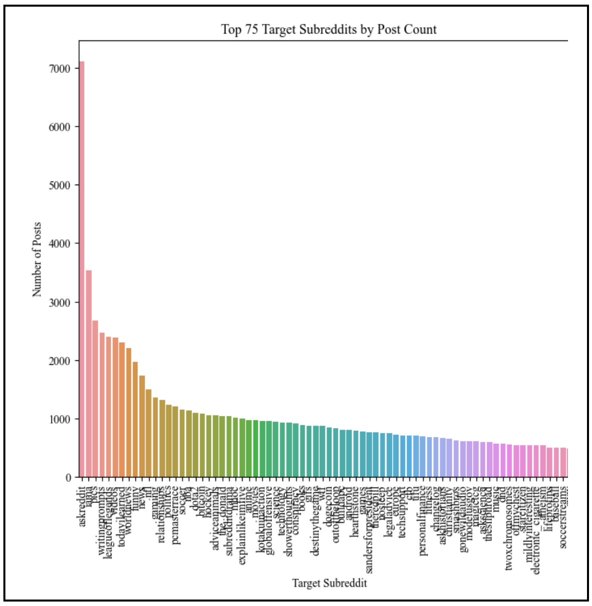

Figure 2: Top 75 target subreddits sorted by number of posts.

information website[4]. All in all, the data contains posts from 27606 unique sources and 20447 unique targets, yielded from 35465 subreddits in the data as a whole. It should be noted that a big share of the total number of posts comes from a small number of subreddits, or alternatively said: there are a lot of subreddits with a rather low post count. While the average amount of posts per subreddit is 10.2, with a minimum of 1 and a maximum of 4601 (see Figure 1) only at the 90th percentile we see a post count of 16, 156 at the 99th percentile. Interestingly enough the subreddit with the maximum target value is targeted 7121 times, implying a significant focus on targeting this subreddit[5] (see Figure 2). Of all these interactions, 260874 have a positive sentiment, while only 20688 are negative. This seems to indicate a basically positive interaction between these subreddits. This assumption is further strengthened by computing subreddits with more negative than positive interactions, netting 657 source subreddits with 1251 posts, and 539 target subreddits with 830 posts respectively. This also shows that there are very few subreddits being the source (579, 697 posts) or target (488, 549 posts) of purely negative interactions in relation to the aforementioned entirety of posts. Unsurprisingly, having the most source posts overall, *subredditdrama* leads the charge in positive as well as negative source post counts (see Figure 5, Figure 6), while we can observe the same for *askreddit* on the target side respectively (see Figure 3, Figure 4).

3.2 Text features

As well as sentiment, the data entries come with a plethora of text features. These range from rather simple metrics like the number of characters or the average word length to more complicated features like an automated readability index or a whole suite of LWIC generated properties. The focus here will be on the former as understanding LIWC features enough to interpret them meaningfully seems out of scope for this work.

Post lengths average at 2073.5 characters (min: 49, max: 41239, std: 3570.4) or 327.6 words (min: 8, max: 7600, std: 557.3) with an average of 27.4% stopwords (min: 0%, max: 62.6%, std: 13.4%)[6]. All these characteristics are far more evenly distributed among the posts and subreddits than the number of posts, but there are still outliers: *testingground4bots* for example has 178 posts with an average of 37461 characters. As there are only 489 subreddits with more than 10 posts (as source) and a higher average character count than the total average character count plus the standard deviation, these findings can likely be dismissed as outliers.

4 Graph analysis

4.1 Method

Now to analyze the data via graph heuristics. This was achieved using *networkx* 3.1. Each subreddit is added as a node to the graph. Edges are added in the same step, with the additional information of the sentiment being stored in the edge's attribute. Then, three scores are calculated. All of them contain the fraction on edges with a sentiment in relation to all observed edges, positive values for positive, negative for negative:

- Edge score: Sentiment of edges between observed source and observed target. This should determine how positive or

[4] http://snap.stanford.edu/data/soc-RedditHyperlinks.html

[5] Of course, the subreddit is *askreddit*. It shouldn't come as much of a surprise for this sub being targeted as often, given the name and supposed thematic diversity.

[6] For more information on the data, see property_descriptions_trans.csv in the repository.

negative interactions between these two specific subreddits are.

- Source score: Sentiment of the observed source node. This should determine how positive or negative interactions from the source node are.

- Target score: Positivity of the observed target node. This should determine how positive or negative interactions aiming at the target node are.

To put this in words: “Does this source hate this target?”, “Does this source hate?” and “Is this target hated?”. These scores are then combined into a single confidence score, equally containing the average sentiment for this specific edge (and its parallels), the average sentiment of the source node, and the average sentiment of the target node. These edge data come from the training dataset and are tested with the test dataset.

4.2 Testing and results

It should be noted that out of the test dataset, 29 entries are interactions between subreddits that are not in the training dataset, neither as source nor as target subreddit. Their sentiment of course cannot be predicted at all, as there is no training data to go off. Apart from this “lost data”, the rest of the entries will be predicted with a confidence score of varying significance, as there are rather few occurrences of some entries and as such their statistical yield is rather meagre. This brings us to the result of 393 wrong predictions out of 4971, 131 of which have a confidence score of .5, being ambiguous and thusly with this approach can’t be predicted confidently[7]. Funnily enough, if the confidence threshold is lowered to .1 or even .01, the number of wrongly predicted sentiments gets even lower due to the sheer number of positive interactions in the training data.

These results may be able to be refined, possibly with a more sophisticated method for determining the sentiment scores than just taking fractures or weighing them differently in the calculation of the final confidence score, as the “historical” data on interactions between the current source and target nodes could have more decision power over the interaction. That said, training with bigger datasets might be required, or even the culling of nodes below a certain interaction (or edge) threshold, to further assure that the data and scores are actually meaningful.

5 Machine learning

5.1 Method and data

	precision	recall	f1-score	support
-1	0.21	0.14	0.17	382
1	0.93	0.96	0.94	4617
accuracy			0.90	4999
macro avg	0.57	0.50	0.56	4999
weighted avg	0.88	0.90	0.88	4999

Table 1: Exemplary classification report of GaussianNB (Naive Bayes) classifier.

For conducting sentiment prediction on the provided gold labels, first the training data is slightly altered and enhanced to use with different classifications. The provided data contains a column ‘POST_PROPERTIES’, that has a comma separated vectors of text features. These are separated into different columns for easier handling during the feature extraction process. Also, the provided timestamp of the post is read as datetime and added as extra features in the columns ‘year’, ‘month’, ‘day’, ‘weekday’, ‘hour’, assuming the time of post could be somewhat relevant[8].

At first, training and test data are loaded and the text features as well as the before extracted time features are declared as features to use. Initially and somewhat simplistically, a variety of classifiers are tested to determine if they yield satisfactory results with the given dataset. Some are discarded without an evaluation of their outcome, as they either do not run entirely[9] or take an impractical amount of time. These steps bring the usable list of classifiers down to the following: AdaBoost, RandomForest, KNeighbours, GaussianNaiveBayes and Quadratic

[7] It should be noted that this algorithm predicts a positive sentiment when the confidence score is 0.5, due to the overall positive nature of the posts. Highering the confidence threshold by .0001 yields more than ten times the wrongly predicted sentiments.

[8] Naïve exemplary assumption: Sentiments could be worse on Mondays. Of course, this does not take into consideration the timezones of posters. Still, time might have an influence on the sentiment of the post.

[9] GaussianProcessClassifier tries to allocate 520 GiB of data. Sadly, the machine this was tested on does not have that amount of RAM or disk space for that matter.

Discriminant. These classifiers are then run multiple times with different random states, to get a distribution of evaluation metrics per class. Even before statistical significance testing of the models, it is readily apparent that almost all the scores for predicting the negative sentiment are horrible (see Table 1). This might be due to the rather low support on this label, at around 7.6% of the data.

5.2 Testing the models

As distributions of scores per classifier are compared, and the data should be non-parametric and is at least ordinal the Mann-Whitney-U-Test[10] is conducted on the distributions of the metrics. This ensures that the "best" classifier with an at least statistically significant difference in relation to other classifiers is used. How much of a difference between the scores (or the mean across the runs) there is will later be looked at.

The analysis shows the AdaBoostClassifier as the best for precision on predicting negative sentiment, as well as recall and f1-score for positive sentiment. RandomForest delivers the best recall value for negative sentiment, although with a sobering best value of .115. It also yields the best score for positive sentiment precision. Best f1-score for negative sentiment is delivered by GaussianNaiveBayes, also merely yielding a top score of .168. As AdaBoost seems do bring the most best values across the classes, it will be seen as a benchmark and also "best" model for this specific data hereafter, not without admitting that a point could be made for choosing RandomForest for having the best f1-score, representing a combination of precision and recall.

That all said, the margins between the absolute best scores of precision predicting positive sentiment for AdaBoost and RandomForest are negligible (RF: .931, AB: .923). Predicting negative sentiments though, AdaBoost loses heavily with scores of .005 for recall and .01 for f1-score. A precision score of .5 for predicting negative is not very high as well. Then again, in a grander scheme of things, all of these values are strikingly low in all of the tested classifiers, leading to believe that either there was no suitable classifier for predicting negative sentiment in the examined classifiers, or the data is insufficient for yielding a satisfactory result for that class.

6 Results and findings

In summary, describing the provided data of reddit posts with sentiment labels and predicting those labels is not an easy task and might require further knowledge of how to handle large masses of data and imbalance in this data. Although thoroughly working with the data and the methods needed for fulfilling the tasks presupposing this specific work, there is always more to be done and learnt.

A prerequisite for working with the data is getting it to be uniform and complete. This was a problem especially in the prediction of sentiment labels via graph heuristics, as some parts of the test data was completely absent from the training data, resulting in having to build around that.

Predicting sentiment from a MultiDiGraph was not immensely fruitful either. With the applied method of giving an interaction a non-weighted score consisting of the edge's cumulative sentiment, the source's cumulative outgoing sentiment and the target's cumulative receiving sentiment, the prediction was not that much better than just assuming every post to have a positive sentiment. Further tuning of the method might be advisable, such as weighing the scores.

The biggest challenge in this whole work was the use of machine learning models. Here it became apparent that the sheer number of available classifiers, not even regarding the specifics of their "preferred" data size, structure or form, was overwhelming and a subject of its own. As the here explored classifiers stem from an overview given on the scikit-learn website [11], they are most certainly incomplete and maybe not fit for the task. On the other hand, the data could be the culprit again.

All in all, this was quite an intensive confrontation with the basics of data science methods for the Digital Humanities and certainly helps as a basis for discovering and learning more about this highly interesting topic.

[10] Also called Wilcoxon rank-sum test.
[11] https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Appendices

A Figures

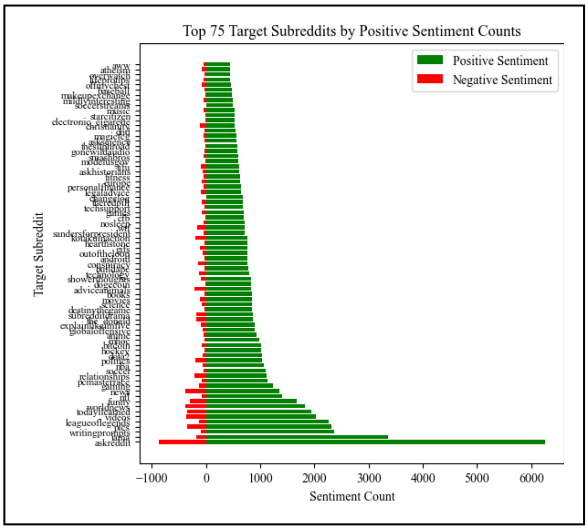

Figure 3: Top 75 target subreddits sorted by positive sentiment count.

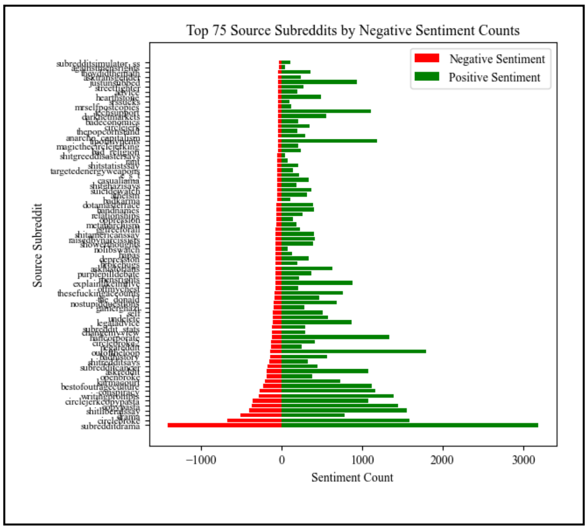

Figure 6: Top 75 source subreddits sorted by positive sentiment count.

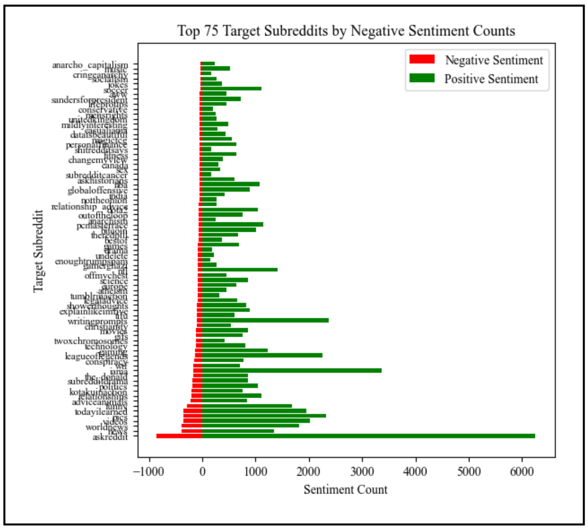

Figure 5: Top 75 target subreddits sorted by negative sentiment counts.

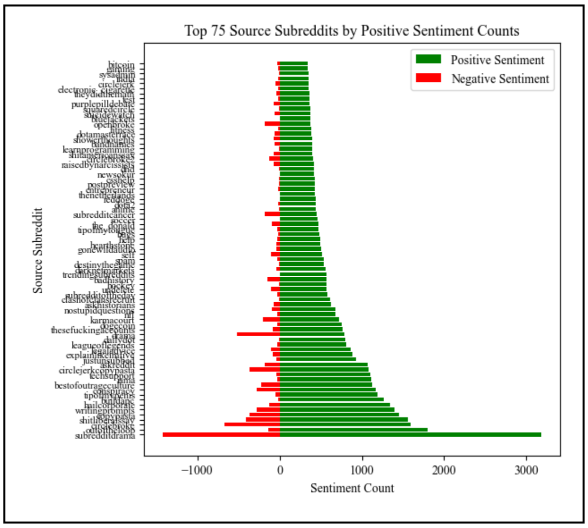

Figure 4: Top 75 source subreddits sorted by negative sentiment counts.