Reddit

Davide Spallaccini[*]
Department of Computer, Control and Management Engineering
Sapienza University of Rome
Rome, Italy

Beatrice Bevilacqua[†]
Department of Computer, Control and Management Engineering
Sapienza University of Rome
Rome, Italy

Anxhelo Xhebraj[‡]
Department of Computer, Control and Management Engineering
Sapienza University of Rome
Rome, Italy

Abstract

Inspired by the "Community Interaction and Conflict on the Web" paper[4] Machine learning techniques have been the driving component of research in many fields in recent years from advertisement, self-driving cars to healthcare. Although many models have been developed to improve accuracy performances it is often difficult to debug and understand them especially

1 Introduction and Related Work

Reddit is a growing social network based on communities called subreddits where users interact by posting articles and links regarding topics of the community and by commenting them. Interests of communities usually overlap but the opinions on the subject matter may diverge. In this context, a user of one subreddit (source of the link) may post a cross-link (i.e. a link that points to another subreddit which is the target of the link) and lead to a mobilization where the users of the source community interact with the users of the target community in the targeted post. The concept has been explored in the paper "Community Interaction and Conflict on the Web" in which the authors have constructed a dataset starting from 40 months of reddit content such as posts and comments. The resulting dataset is composed of 394,216 instances of mobilizing cross-links represented as id of source post, id of target post and a label telling whether the mobilization was neutral/positive ("non-burst") or negative ("burst").

Starting from the concept of user and community interactions on Reddit we decided to reproduce some of the results of the paper to have an insight of the process and understand the difficulties and challenges of the task. Based on the dataset described above we applied a novel model employing deep learning concepts, using a combination of a convolutional neural network followed by an LSTM recurrent network to predict whether a mobilization is positive or negative, i.e. whether it will lead to a conflict. Additionally, on the target posts of the negative mobilizations we analyzed the kinds of user-user interactions developed by constructing a reply network of the comments for each post and applying personalized PageRank. Finally intrigued by the structure of the social network we decided to find similarities between communities and construct a recommender system to permit the users to discover new communities based on their interaction history.

2 Dataset Retrieval

The "Community Interaction and Conflict on the Web" dataset is provided in a preprocessed form where posts are represented by the indices of the word embeddings of the words forming its title and body and users by a feature vector that describes the activity levels and lexical features of their previous posts. In order to have more flexibility over the design choices of our models we decided to rely on that dataset only as a ground truth and to retrieve the raw content from the source, i.e. the reddit social network. In particular we used the Pushshift API in conjunction with the Python Reddit API Wrapper (PRAW). In this task we encountered our first challenge which was due to

[*] spallaccini.1642557@studenti.uniroma1.it
[†] bevilacqua.1645689@studenti.uniroma1.it
[‡] xhebraj.1643777@studenti.uniroma1.it
[4] Paper link

the rate limiting set by both the APIs which allow respectively an average of 120 requests and 60 requests per minute. Moreover to retrieve the necessary data for each instance of the dataset multiple requests were required. By launching multiple processes in parallel and using multiple accounts we were able to get the most of the information needed.

For the LSTM-based classification task, given the post IDs of the sources of the cross-links provided by the dataset, the Pushshift APIs were used to retrieve: in a first request the text of the title and of the body of the post, in a second one the ids of the top 8 comments present in the post and in a third one the first 512 characters of the body of each comment to avoid giving too much weight to the comments with respect to post text. The process just described unfortunately incurred into some exceptions caused by rate limiting and post deletion but still allowed to collect enough data for the classification as will be described in Section 3.

To reproduce the analysis on the reply network we took only the instances of the dataset that were labeled as "burst" i.e. the ones that produced a negative mobilization. For each of the target posts we used the PRAW library to retrieve its comments and the authors of those comments. Of the authors, only the subset of users that were attackers or defenders were chosen. In order to know whether a user is an attacker or defender we used the Pushshift API to retrieve the comments submitted by the user in the 30 days before the target post submission. An attacker (resp. defender) is a user who has made at least one comment in the source (resp. target) subreddit in the 30 days prior to the cross-link but who did not comment in the target (resp. source) during this time period.

The gathering of the data for the subreddit similarities and the recommender system was done by downloading all the comments submitted to reddit in December 2017. The data for each reddit month is available as a lmza compressed file at pushshift.io in JSONLines format which is the standard for Big Data. In this task we encountered another issue in dataset retrieval since the size of the uncompressed file reached the size of $\approx$ 80GB filling up the available disk space of the machine. We opted for first performing a counting of comments by subreddit to spot the most active subreddits represented in Figure 1 by performing a scan over the compressed dataset since lzma compression permits streaming decompression.

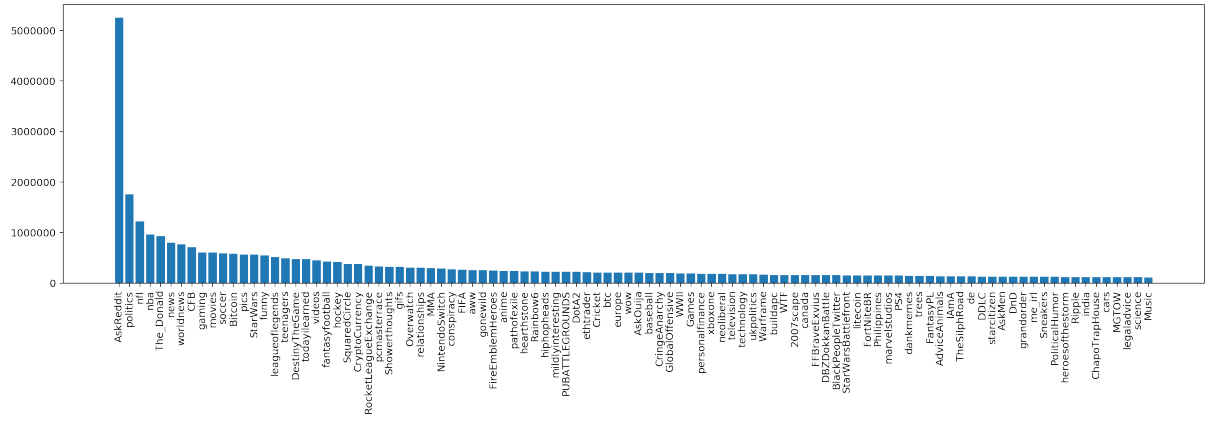

Figure 1: Most active subreddits by number of comments in December 2017

Since processing such dataset was unfeasible on a standard desktop we decided to take a sample of it by keeping each comment in the original dataset with a probability of $\frac{1}{4}$ and transforming the format from JSONLine to csv which reduces the space needed for storage.

3 Mobilization Sentiment Prediction through LSTM

The author's dataset was built with the help of Mechanical Turk crowdworkers that manually annotated almost a thousand of cross-links telling whether the sentiment of the source post towards the target post was negative or positive/neutral reaching a inter-rater agreement of 0.95. Over this sample the authors applied a Random Forest classifier with forests of 400 trees that achieves an accuracy of 0.80 on a 10-fold cross validation. They then used this classifier to build the dataset presented in Section 1. Such dataset was used as a ground truth to train our model. By performing the retrieval described above we obtained a set of instances of which about 95% labeled as positive as announced in the original paper raising another challenge. Such unbalancing between the number of instances of the classes would produce an unreliable evaluation therefore we chose to undersample the retrieved dataset. Instead of using a random heuristic we used the NearMiss version 1 method that reduces the number of positive elements by

keeping only those positive instances for which the average distance to the N closest samples of the negative class is the smallest. After undersampling we obtain a dataset of about 8,000 instances that is split into 80% of training set and 20% of test set.

Instead of employing the large set of features, that by the way are not very well documented, used by the authors in this setting we propose an improved model based on the bare content of the post and its comments in terms of contained words. This is a trend in modern classification methods where deep learning models substitute annoying feature engineering.

Our model is essentially based on a combination of a convolutional neural network followed by a LSTM recurrent network. At the basis of the model we use GloVe word embeddings that allow us to represent each word in the text in a fixed- size, compact and dense vector of 300 floats also taking into account similarity between words. The advantage of GloVe vectors over the simple one-hot representation of words is that these vectors were trained using a neural network so that words that have a similar context are close in the vector space. Then the word embeddings are given in input to the convolutional layer. The purpose of this layer is to capture more general patterns in the data helping the network to better generalize to new examples without excessively specialising on the text of the training data. The result is given in input to the bidirectional LSTM layer which is responsible of learning the patterns in the sequences of words, something that recurrent network were designed to do well. In particular the LSTM architecture allows to "remember" longer sequences learning which parts of the sequences are more important. After some tuning of the hyperparameters we obtained a classifier with the following performances.

	precision	recall	f1-score	support
non-burst	0.87	0.93	0.90	769
burst	0.93	0.87	0.90	807
avg/total	0.90	0.90	0.90	1576

4 Attacker Defender PageRank

To understand whether in the target post of a negative mobilization attackers and defenders talk to each other or they form their own groups thus creating echo chambers we construct a graph of the users' interactions. In fact, reddit comments can be nested and a comment can be in response to another comment. From this scheme, a reply network is constructed where each node is a user, either attacker or defender, and a directed edge from user i to user j indicates that i replied to one of j's comment. The weight of the edge indicates the number of times a user replied.

We quantify the echo chamber by replicating the approach of the paper which is based in the application of personalized PageRank: firstly by restricting the teleport set to the defender nodes (Defender PageRank) and secondly to the attacker nodes (Attacker PageRank). The Defender PageRank (resp. Attacker PageRank) score of a node represents its centrality measured from the perspective of Defender nodes (resp. Attacker nodes).

5 Recommender System

Interested in the properties of the reddit social network, we decided to extend the analysis provided in the reference paper by trying to implement a subreddit recommender system. In order to implement such system we preferred to not perform a crawl of the reddit website since it could have produced a bias in the result coming from the fact that the communities are not necessarily linked. Instead we chose to use the sample of the reddit comments of December 2017 retrieved as explained in Section 2. This is of course an over-semplification of the task but it is based in the often valid assumption that contents of subreddits do not change rapidly at the granularity of the month.

The first phase of the design process was to choose how to tackle the problem. We thought about experimenting a Collaborative Filtering approach but the amount of users in the sample and the sparsity of user-subreddit matrix would have not been significantly effective. Instead we opted for a Content-Based system using subreddit similarities. We decided to represent a subreddit as the TF-IDF vector of the most frequent words in its comments. We split the computation into multiple phases. In the first phase we grouped comments by subreddit. In the second phase we tokenized the comments, removed stop words and performed stemming based on the Porter algorithm. Finally we computed the TF-IDF vector of length 10,000 for each subreddit by considering a vocabulary with the 10,000 terms having the highest TF.

To qualitatively test this representation of subreddits we decided to compute the most similar subreddit for each subreddit. Indeed we've found out similarities between subreddits about sports, news, funny posts etc.

To evaluate