How Well Does Language-based Community Detection Work for Reddit?

Urvashi Khandelwal
Stanford University
urvashik@stanford.edu

Silei Xu
Stanford University
silei@stanford.edu

Abstract

Online communities in the modern day era are becoming more and more important. This makes it imperative for us to understand the structure of these communities. In addition, content generation sites like Reddit, Tumblr and Quora have an abundance of text in comments and posts which can be used to model the user interactions and network substructures. In this paper, we propose to study community detection in Reddit solely based on language features, to better understand how well language informs the boundaries between different communities. We use supervised prediction tasks and unsupervised community detection to gauge the quality of these features and find that they provide a fairly robust signal in trying to understand and model user interactions in the network.

1 Introduction

One of the most important tasks in understanding a social network is Community Detection. By understanding how the network organizes itself into communities, we gain the insight that can explain the various relations between entities. An important signal that lends itself well to community detection in an online social network is text. Danescu et. al. showed in their work on Beer communities that language plays a key role in identifying the life-cycle of a user within a community (Danescu, 2013). This is an instance of the importance of language in understanding online social networks. For a content generation site like Reddit, we look to utilizing language features from user comments, in order to predict subreddits for test users as well as detect communities within the user population.

Traditionally, community detection algorithms have looked at network structure as well as node attributes. However, in this study we focus our attention on the language features. How well do language similarities connect users with similar interests? This is an important question to ask when trying to understand an individual user's diverse interests as well as which communities suit them best. Such an understanding can aid collaborative filtering tasks, content recommendation as well as personalized search.

The rest of the paper is organized as follows: in Section 2, we first introduce the related works. In Section 3, we explain the dataset, its processing and feature extraction. In Section 4, we give a brief overview of the learning algorithms, proceeding to explain experiments in Section 5, wrapping up with a conclusion in Section 6.

2 Related Work

Online communities have been studied for decades and most of the traditional community detection algorithms put their effort in analyzing graph structure of the data (Fiedler, 1973; Pothen, 1990; Newman, 2004). However, in the real world, apart from the topological structure, we also have content information available to us. In recent years, analysis on community detection on networks with node attributes has gained more and more attention (Zhou, 2009; Yang, 2013). One of the most significant attributes for nodes (users) in content generation sites is their language (Danescu, 2013). In this project, we go one step further by detecting communities solely based on users' language to see how well language informs the user interactions and ground-truth communities.

3 Dataset and Feature Extraction

Reddit is an online content generation website organized by topically specific subreddits, where users can submit posts and have other users com-

Figure 1: A subset of unigrams for a randomly selected user where size of word is correlated to frequency

ment on them. The Reddit comments dataset[1] is a publicly available compilation of comments from nearly 97,000 subreddits since 2008. The number of comments in these communities follow a heavy-tailed distribution, such that only about 7,000 communities have at least 1,000 comments in the year of 2014.

In this paper, we only consider comments posted during 2014. In addition, we consider only midsized communities which are defined as those that had at least 100 thousand comments and at most 1 million comments over the entire year. This choice is driven to avoid extremely large communities where users are likely to feel a lack of loyalty and extremely small communities where the number of comments are too small to justify a wholesome community. After filtering in this way, we end up with **633 communities**.

Next, we looked at the user population for the included communities. First we filtered users by removing bots. Bots were identified as usernames that ended with the string 'bot' or 'Bot', as well as users that posted over 6000 comments in 2014, posted to over 200 communities in 2014 or over 700 comments to a single community (activity representative of bots upon looking at the data). Once we get this raw list of users, we further prune all those who posted less than 1000 comments through the year, in order to have a large number of features per user. In this study, we include **5123 users** with each having posted in at least 10 subreddits. Overall, we have **7,043,339 comments**.

Finally, we extract language features for each user. Since the language in Reddit is extremely informal, we start by looking at unigrams. For each user, we start by tokenizing each comment using NLTK[2]. A canonical list of stop words is used to filter through the more topical and user specific tokens. Tokens containing punctuation, urls, numbers and non-ascii characters are removed. Finally, nouns and verbs are lemmatized. This generates a list of unigrams with term frequencies. The overall size of the unigram vocabulary is **1,512,526 words**, with **5180.5 average unigrams per user**. For the entire set of users, we proceed to compute Term Frequency-Inverse Document Frequencies as follows:

$$tfidf(w) = (1 + \log(tf(w))) \times \log \frac{N}{df(w) + 1}$$

where, w is a single token, N is the number of users, tf is the term frequency, and df is the document frequency, treating each user's comments collectively as a document. We use $tfidf$ scores as the feature vector for each user. Given the sparsity of the perceived user-unigram matrix, we use SVD to map the features to lower dimensions, making the different users' language more comparable.

4 Learning

In order to gauge the quality of our language-based features, we looked at two types of tasks: supervised prediction and unsupervised community detection.

4.1 Supervised Learning - Multilabel Classification

For supervised prediction of subreddits for test users, we use multilabel classification algorithms since each user posts to multiple subreddits and belongs to mutliple classes.

4.1.1 Decision Trees

Decision Trees (Quinlan, 1986) are described as follows:

1. Start with the entire dataset.
2. Split along the attribute/dimension that provides the highest information gain.
3. Divide examples into children based on the splitting attribute.
4. Recurse on the children, going back to step 2 for each.

Information Gain on set X given that we know the attribute value for A is defined as follows, dependent on the definitions of information entropy (H) and conditional entropy:

$$IG(X|A) = H(X) - H(X|A)$$

$$H(X) = \sum_i p(x_i) \log_2 p(x_i)$$

$$H(X|A) = \sum_{i,j} p(x_i, a_j) \log_2 \frac{p(a_j)}{p(x_i, a_j)}$$

[1] https://www.reddit.com/r/datasets/
[2] www.nltk.org

In the context of our setting, Decision Trees can be thought as splitting based on different topics, thereby creating a kind of a topic hierarchy in the process. The subreddit labels lie in the leaves and each user can belong to multiple subreddits.

4.1.2 Random Forests

Random Forests is an ensemble technique that combines multiple decision trees (Brieman, 2001). Each tree $k = 1, ...n$ is constructed based on i.i.d. random vectors Θ_k, sampled from the feature distribution. The outputs of all trees are combined based on majority vote, with some threshold for multilabel settings. This helps to remove variance by training on different parts of the dataset and averaging over all predictions. In our setting, this is expected to be helpful in reducing overfitting as well as the generalization error.

4.1.3 ML k-NN

Multi Label k-Nearest Neighbor is an extension of the k-NN algorithm. For every query user from the test set, we compute the k nearest neighbors in the feature space, using their class distribution as a prior. Then, the MAP estimate is used to compute the label set for the query user (Zhang, 2007). In our setting, this would help to demonstrate how well the distribution in the feature space represents actual subreddits.

This algorithm should perform better than the two based on decision trees, since topical hierarchies try to isolate a subreddit's language where multiple subreddits might have similar language and this may lead to higher error. On the other hand, ML k-NN attempts to find similar users, which might all be posting to a similar set of subreddits and would have higher confidence in the predictions.

4.2 Unsupervised Learning

Community detection can be seen as a clustering of nodes into sets of similar entities. Hence, in order to detect communities in Reddit, based on the users' language, we tried two unsupervised clustering algorithms.

4.2.1 k-means Clustering

We start with the k-means clustering algorithm to group users into communities, with the objective of using coordinate descent to minimize the within cluster sum of squares metric:

$$J(c, \boldsymbol{\mu}) = \sum_{i=1}^{m} ||\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}}||^2$$

where, J is the distortion function that measures the sum of squares between each n-dimensional example $x^{(i)}$ and the center of the cluster to which it was assigned $\mu_{c^{(i)}}$. In this setting, the $x^{(i)}$'s are feature vectors containing *tfidf* scores for user i. This algorithm performs hard clustering, i.e. every user is assigned to a single cluster.

4.2.2 EM Algorithm

A soft clustering algorithm which uses coordinate ascent to maximize the following objective function:

$$J(Q, \theta) = \sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

where $x^{(i)}$ is the feature vector containing *tfidf* scores for user i, $z^{(i)}$ is the subreddit being considered, $Q_i(z^{(i)})$ is the distribution over the labels for user i.

5 Experiments and Analytical Discussion

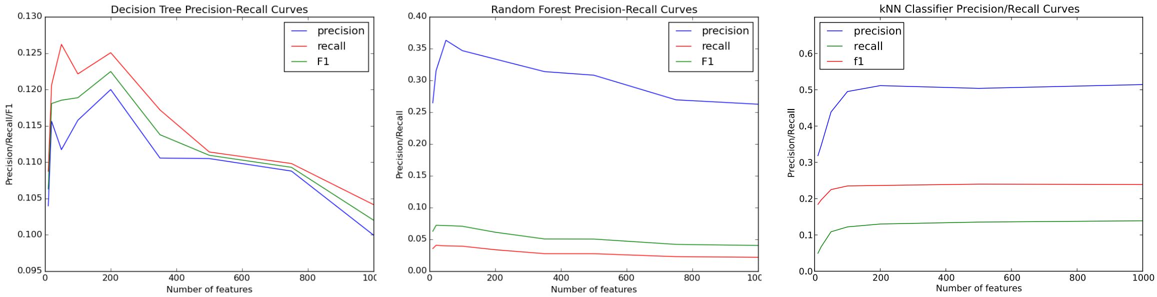
In this section, we present results for the supervised and unsupervised approaches to learning described above. We evaluated both techniques using Precision, Recall and F-1 scores. Precision is defined as the fraction of the subreddit labels predicted that matched the ground truth, Recall is the fraction of ground truth subreddit labels that were predicted and F-1 score is the harmonic mean of Precision and Recall.

5.1 Ground Truth

Ground truth in this dataset exists in the form of sets of subreddits for each user, where a subreddit is included if the user posted to it during the year. This ground truth is useful for the evaluation of the multilabel classification as well as the clustering approaches.

5.2 Prediction Task Results

The prediction task involves training a model to learn which subreddits a user belongs to based on the feature vectors, in order to predict these for a test set of users. We randomly pick 4123 users as our training set and test on the remaining 1000 users using the supervised learning algorithms described in Section 3.1 - decision trees, random



(a) Decision Trees *(b) Random Forests* *(c) k-NN*

Figure 2: Supervised Learning Precision/Recall Curves

forests, and multi-label k-NN. For random forests, the number of trees used was 5. For ML k-NN, the number of neighbors considered was 5. The Precision, Recall, and F-1 scores are shown in Figure 2.

As we can see, the results for decision trees and random forests are relatively better for fewer features. This can be explained by the fact that after performing SVD and picking the top n features, for a small value of n this tends to include the more topically relevant features that help to distinguish between similar subreddits while classifying a new user. For large values of n, we tend to include more noisy features which would cause the model to overfit. For instance, with fewer features we get a shallower decision tree, but as it gets deeper with higher n, performance degrades as picking the correct subreddits at the leaves gets harder due to confusion caused by noisy features.

Using a higher number of decision trees (5 in random forests) helped to increase the precision in the sense that fewer subreddits were being predicted and a larger fraction of these were correct. But Recall was lower since fewer predictions meant retrieving lesser of the ground truth. Overall, fewer noisy results (based on confusion caused by noisy features) meant that using random forests served as an improvement over decision trees, just as expected.

For ML k-NN, with a larger number of features, the performance remained fairly stable because the algorithm relies on other similar users to make a prediction and when the feature vector dimensions change, it affects all users in a similar way, showing that similarity of users in the feature vector space is robust in the context of noisy features.

5.3 Community Detection Results

Baseline: We implemented a naive baseline based on random assignment to get a sense of how the clustering algorithms performed in comparison to random guessing. In this baseline, every user is assigned a cluster between 1 and k uniformly at random.

k-means clustering: We adopted Llyod's algorithm for the k-means clustering. The number of clusters were varied from 2 to 600, where each user is assigned to a single cluster.

EM: We used the Gaussian Mixture Model implementation for EM. The number of latent variables were varied from 2 to 600 and a probability was computed for each user's membership for the class.

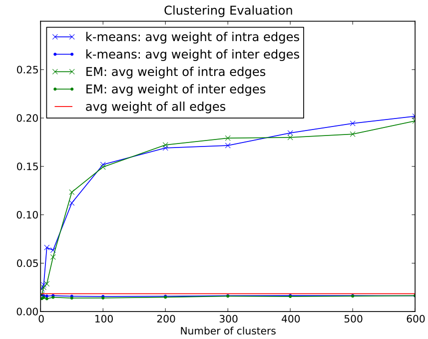
We first evaluate the performance of k-means and EM by calculating the average weight of intra-cluster and inter-cluster edges of detected clusters, where edges are defined as follows. For a user u_i, let C_i denote the set of subreddits u_i commented in, and for each subreddit $c \in C_i$, let $N_i(s)$ denote the number of comments u_i posted in c. Then the weight of the edge between a pair of users u_i and u_j, denoted by $w_{i,j}$, is defined as

$$w_{i,j} = \left(\frac{\sum\limits_{c \in C_i \cap C_j} N_i(s)}{\sum\limits_{c \in C_i} N_i(s)} \right) \times \left(\frac{\sum\limits_{c \in C_i \cap C_j} N_j(s)}{\sum\limits_{c \in C_j} N_j(s)} \right)$$

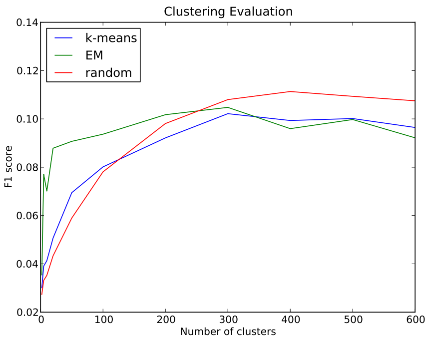
Note that if $C_i \cap C_j = \emptyset$, then $w_{i,j} = 0$, and in case that $C_i = C_j$, we have $w_{i,j} = 1$. The results are shown in Fig. 3a.

The availability of ground-truth subreddits allows us to quantitatively evaluate the performance of these two unsupervised learning algorithms. However, it's hard to find the mapping between detected communities and ground-truth subreddits. Thus, we evaluate the clustering results by calculating the average F-1 score of the best matching ground-truth community to each detected community and the best matching detected community to each ground-truth community:

$$\left(\frac{\sum_{c_i \in C^*} F1(c_i, \hat{c_{m_i}})}{|C^*|} + \frac{\sum_{\hat{c_i} \in \hat{C}} F1(c_{m_i'}, \hat{c_i})}{|\hat{C}|} \right) /2$$



(a) Weight of intra-cluster and inter-cluster edges



(b) Average F1 score

Figure 3: Supervised learning evaluation

where $C^*, \hat{C}$ denote the set of ground-truth and detected communities, respectively, m_i, m'_i denote the best matching: $m_i = \arg\max_{\hat{c} \in \hat{C}} F1(c_i, \hat{c})$, $m'_i = \arg\max_{c \in C^*} F1(c, \hat{c}_i)$. The results are shown in Fig. 3b.

From Fig. 3a, one can observe that both k-means and EM algorithms perform much better than randomly guessing cluster assignments: the average weight of intra-cluster edges is much higher than inter-cluster edges, whereas the two metrics should be similar if we cluster users randomly. From Fig. 3b, one can see that both algorithms perform poorly when trying to match the ground truth precisely. The algorithms perform relatively better when trying to discover fewer clusters. This is because the similar subreddits have similar language which allows them to be grouped to form super communities and the clustering algorithms seem to be better at detecting these clusters than the more fine-grained subreddits that we are working with.

5.4 User Activity

The prediction task and community detection results collectively demonstrate that the unigram *tfidf* scores do not form a strong set of features. One reason for this can be that we are constructing these features by using comments posted throughout the year. However, as shown in Fig. 4, users post comments to different subreddits at different times of the year. Mixing signals from throughout the year might be adding more noise and it could be interesting to consider comments and ground truth from specific time-windows (day, week etc.).

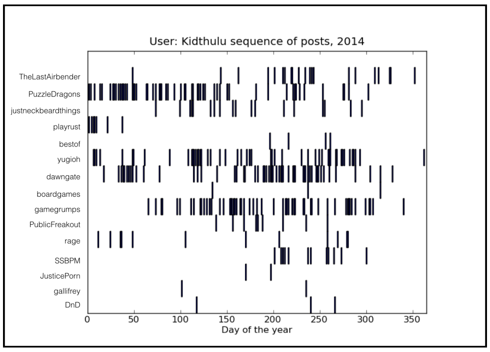


Figure 4: Event Plot showing the comments posted to different subreddits for a randomly selected user who posted 501 comments to 15 subreddits through the year.

6 Conclusion and Future Work

From this study, we see that language in the comments holds signals that inform the process for modeling user interactions in Reddit. If we consider the task of trying to understand the interactions based on similar interests, rather than the precise subreddits, these techniques hold a lot of potential. In this case, it might work to our advantage to try and cluster the ground truth into groups based on similar topics. Another possible future direction could look at making the language features more robust, i.e. considering bigrams, processing the comments using more advanced natural language processing techniques as well as setting up word vectors to match language based on semantic meaning. Finally, it might also be beneficial to consider time-windows for the comments. This would not only improve the current study, but also allow us to understand how these user interactions change over time. In conclusion, our project served as a good starting point for looking at text to model interactions between users in the Reddit network and after confirming that the language features provide robust signals, we can pursue many future directions to make a more compelling argument about how they can be useful.

References

Fiedler, Miroslav. 1973. *Algebraic connectivity of graphs*. Czechoslovak mathematical journal.

Pothen, Alex, Horst D. Simon, and Kang-Pu Liou. 1990 *Partitioning sparse matrices with eigenvectors of graphs*. SIAM journal on matrix analysis and applications.

Newman, Mark EJ. 2004 *Fast algorithm for detecting community structure in networks*. Physical review E.

Yang, J. and McAuley, J. and Leskovec, J. 2013. *Community Detection in Networks with Node Attributes*. IEEE International Conference On Data Mining, Dallas, TX, USA.

Zhou, Y. Cheng, H. and Yu, J. 2009. *Graph Clustering Based on Structural/Attribute Similarities*. Proceedings of VLDB Endowment, Lyon, France

Zhang, M. and Zhou, Z. 2007. *ML-KNN: A lazy learning approach to multi-label learning*. Pattern Recognition

Brieman, L. 2001. *Random Forests*.

Quinlan, J. R. 1986. *Induction of Decision Trees*. Machine Learning

Pedregosa, F. et. al. 2011. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research

Danescu-Niculescu-Mizil, C. and West, R. and Jurafsky, D. and Leskovec, J. and Potts, C. 2013. *No Country for Old Members: User lifecycle and linguistic change in online communities*. ACM International Conference on World Wide Web, Rio de Janeiro, Brazil.