From Social Networks to Sustainable networks: A Machine Learning Perspective

Team Member: Jiangtian Qian, Rui Wu

Abstract

There are growing interests in being involved in the "Green" social network and a lot of researches done to find the sustainability and high-societal impact of these social networks. Our project mainly focuses on finding the pro-social impact of one "Green" social network, Change.org, one of the most active and largest petition website. To be general, our research based on the 5000 pieces of petitions under 41 different topics.We did two different kinds of research-user network and petition network. For user network, we applied node2vec to get feature vector and do the linear regression. For petition network, we applied word2vec to get feature vector and do multi-label regression. The accuracy of user prediction is 0.75 and we adopt f1 score metrics to evaluate our model for multilabel regression task, where micro is 0.95 and macro is 0.66. Additionally, we try to combine node2vec and word2vec features together to see if we can improve our model. After PCA reduction applied on the combined features, the f1 micro is 0.92 and f1 macro is 0.59. Which indicates that the combination can not ensure better performance for the multilabel task. While in binary classification problem like petitions' victory prediction, the combination does improve the accuracy, where the precision increases from 0.51 to 0.58 and the recall increases from 0.54 to 0.61. And we also figure out that hot topic has more chance to win.

1. Introduction and Motivation

In "Green" social network field, there is a growing interest in figure out how the "Green" social network has sustainable and pro-social impact. Participating and broadcasting the content of these networks are the regular activities of these networks. And people often link each other into communities online just like joining clubs in the real world. How do people become a member of the community? Think about it, for example, you log in the Change.org and concerned about some petition; you want to make the contribution to make petition victory, you support this petition or leave the comment to show your interests; now you are in a community of this petition. We collect raw data from Change.org which contains attributes of users and petitions and analyze the user's relationship and a relationship between petitions with machine learning tools.

1.1. Description

In the sustainable online network area, there is a growing concern about how participants influence each other. We have a particular interest in pro-social behavior that how large online social network brings single behavior to others and influence them. Besides, the interesting part of our project is how much influential users will determine the number of signatures a petition could get since all followers of an influential user will also sign a petition that this influential user signs. As we know the number of signatures will determine the success, we can make clear the relationship between the influential user and the victory of a petition. And for the petition part, we want to find the similar petition. What's more? Is there any possibility that certain content may lead to the victory of the petition, for people may more interested in some topics than others. For the Data we used, which are collected from change.org, one of the largest and active petition website. Change.org provides a social networking platform that enables members to create or sign a petition. Besides, members can leave a comment on other people's petition. Petition types vary from animals and politics. The presence of online social network makes Change.org an ideal platform to observe and do research about social network influence pro-social behavior in social communities.

1.2. Main Challenges and approach

It will be interesting, from a theoretical as well as their visibility among users and petitions. To be specific, this project seeks the results of these challenges:

1. Build user's network: Using user's id as the resource and petition's id which user leave a comment below as target. Then we can get property of this network, which can help us do research on the relationship between users and the relationship between user and petition.

2. Topic prediction for users: For the user, we do multi-label prediction using logistic regression since each user's commented petition can have multiple tags.

3. Petition victory prediction from users: Applying machine learning tools like node2vec to get feature vector of users and predict petition victory status.

4. Petition victory prediction from petitions: Applying machine learning tools, word2vec, to get feature vector of each petition and predict petition topics.

5. To make petition's relationship more intuitive, we build petition's network and do research about this network.

6. Improve model: Optimize our machine learning tools and improve the accuracy of our prediction by combining node2vec and word2vec for petition prediction.

For the first challenge, we pre-processed data and keep user's id, the petition creator's related to it. Then we connected them by setting user's id as source and petition creator's id as the target. At last, we use Gephi to visualize this network and analyze network's degree, average length and so on. For the second challenge, we apply node2vec to extract feature vector of each user and match them with tags.

We also use word2vec and create a correlation-based petition network. Analyze its properties, like degree, centrality, communities. And figure out what the different communities spread among the network. Since we build user and petition model based on the huge data set, say the 30GB text file, collecting and cleaning data is the most time-consuming part.

1.3. Novelty of our research

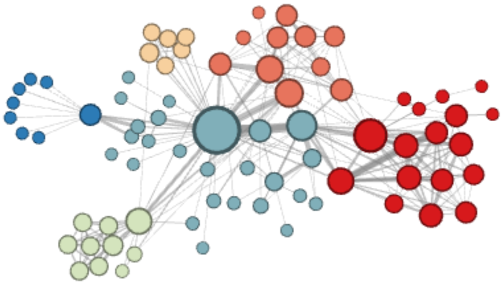


Figure 1. Example: color-coded communities find by Node2vec.

Node2vec, programming in c++, is applied to our users' network to extract features vector. And it will help to detect network structure property like the community as Fig.1 shown. A machine learning algorithm-liner regression is also used to deal with feature vector we have got earlier.

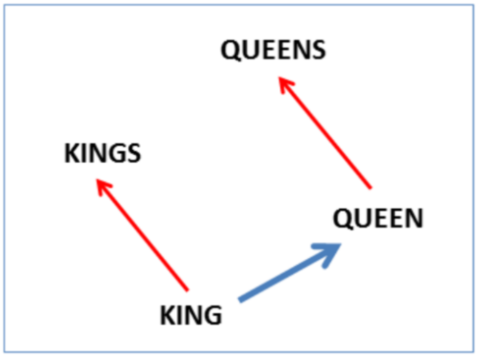


Figure 2. Example: word transfer to vector.

Then, we applied word2vec to petition's description to extract feature vector. After that, we applied multiple-regression to match labels with a petition. For node2vec and word2vec can both extract feature vector, we combine them together and do PCA to get a combined feature vector. When we use this feature vector to do victory prediction, we got higher accuracy.

The similar previous works are most likely using LSI, our work based on more efficient and accurate methods node2vec and word2vec. Node2vec can learn continuous feature representations for the nodes in a given graph, which can lead greater predictive power, like regression problem in users-net to decide users preference. Different from LSI, a count based model, word2vec is a prediction based model, given vector presentation of a word predict the context word vectors, which captures multiple degrees of similarity between words in a better manner. In this vector space, words are close to others if they share the similarity in semantic like KING and QUEEN in Fig.2 and each of them actually is represented in KINGS and QUEENS vector.

2. Previous Work

To make our project in the right way, we did research of previous work on theoretical background and method support for sustainable network research.

2.1. Previous Approaches

During the development of machine learning, methods vary. In Juan Ramos's research paper, he mentioned applying Term Frequency-Inverse Document Frequency(TF-IDF) to determine what words in a corpus of documents might be more favorable to use in a query. As the term implies, TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. The simple structure of TF-IDF cannot effectively reflect the importance of words and the distribution of feature words, which makes it impossible to adjust the weight well. In Dario De Nart's Social Network Analysis(2015), he did several works in literature investigated the activities of research communities based on LSI and get inspect on their emerging trends. But analyze on big data, LSI is not fast enough to compute the result on a standard personal computer. So, we choose node2vec and word2vec, which are more efficient and accurate.

2.2. Related to Our Work

In Ernst Fehrl and Urs Fischbacher's article(2003), they mention that a minority of altruists can force a majority of selfish individuals to cooperate or a few egoists can reduce a large number of positive participant to defect depending on the environment. This is the basic theoretical background for us to study how online networks or communities can support prosocial behaviors in Change.org.

Secondly, in Burt's lecture(1992), he explained the communities with similar property trend to show a prosocial behavior. And Dobin Yim and Siva Viswanathan think a network with greater opportunity for social learning make a participant makes sense about what to do in a prosocial context while removing uncertainty about who is participating. This idea gives us a clear image of relationship between network perspective on prosocial behaviors.

Prior research(Moon and Sproull 2008) on online network points out it can enhance prosocial behavior by simulating reaching out to many participants and yet close ties between them through feedback. In our research, we filtered out comments of each petition and see how petitions can links supporters together.

Much prior work(Liebrand et al.1986, Batson 1998) touch the marginalized participants but do not research how to induce the prosocial behavior and sustainability behavior. Our research will primarily draw on the dynamics of sustainable pro-social behavior conversely.

3. Approach

3.1. Build Network

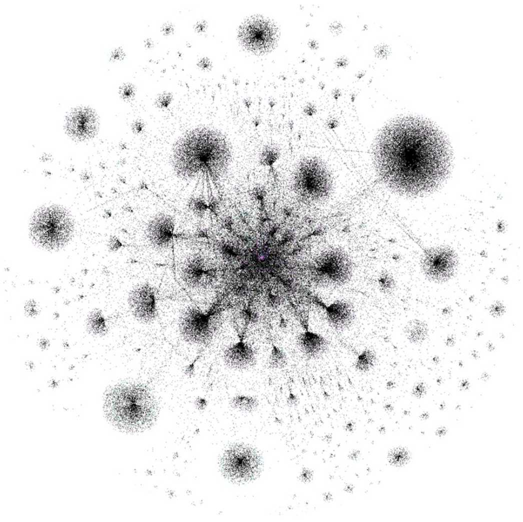


Figure 3. Petition creator and commentators.

Our project starts with collecting the huge amount of data set. These data include information about users and petitions respectively. We do HTML scraping using Python and Change.org's API and collect 40GB data, that is 5000 petitions through our whole project, from September,1,2017 to December,1,2017. For the user, we filter there information: petition creator id and created the location, the comment details, with comment id, the commentator userId and timestamp of the comment. For petition, we filter this information: petition description, petition content, petition id, petition victory status and petition tags. Then we build networks and learn network behavior by machine learning tools. And at the end of the network construction, we use Gephi to visualize the network and calculate network properties.

User information	Petition information
petition creator's id	petition id
location	petiition victory status
comment details	petition tags
comment id	petition description
commentator userId	
timestamp of comment	

Table 1. Filtered information of user and petition.

To build user network, we find the direct relationship between petition creator and commentators and use them as source and the target node in the edge. We build user network as a directed network. For example, we have petition A, and a user B support it and leaves comment, like 'I support you' under petition A. And at the same time, C support A, too. And C leave the comment under A, like 'Go fighting, baby!'. We will let B and C be the source and A be the target and connect them as with a directed edge.

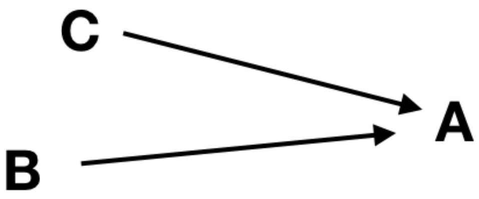


Figure 4. How user network come out.

While for petition network, we extract keywords from petition descriptions and train on those words with word2vec. By computing similarity between each pair of petitions and selection of reasonable threshold as 0.90, we could then build petition network on the basis of their semantics. For example, we choose the threshold as 0.90. Then we will keep petitions whose similarity between each other is higher than 0.90. And we define that there are edges between these kept petitions, and all of their weight is all 1. Petition's network is undirected.

3.2. Tasks and Assumptions

In addition, when achieving the goal to construct a network, we are also interested in finding out networks' attributes. Our main tasks could be split into several parts as showing below

1. Collecting various petitions and users information from Change.org API to lay the basis of the network.

2. Build users' network and have a fundamental knowl-

edge of users network then apply node2vec with edge information as input to extract network features vectors.

3. Users' preference research on users network, combine features from node2vec and linear regression to predict users' belongs to topics.

4. With the help of doc2vec, deriving from word2vec, we are able to find out to which extent each pair of petitions is similar to each other. Based on the mutual value, we then can build the network from petitions data by selecting a reasonable threshold to decide to which extent should we connect two petitions.

5. Use vectors from doc2vec for regression task for subsequent machine learning process, like binary classification for victory. Since one petition may belong to several different topics, a multilabel linear regression is a better choice for training algorithm.

6. Fix the experimental setup: fix petitions' tags and Report F1 micro and F1 macro for multilabel classification.

7. Apply node2vec on petitions' network and contrast regression performance on victory prediction with word2vec in terms of F1 micro and F1 macro.

8. Simply combine word2vec and node2vec features with PCA reduction for victory prediction and figure out which method could provide the best performance in regression task.

9. Collect more petitions' information with victory status and scale up the regression problem to see the model performance stability.

As we already know that words may change with time, which will affect our accuracy. A large proportion of words are stable over time: we do not except most common words (food, car etc..) to undergo important changes in meaning over time. We focus on the word embedding from the 20th century, which should also limit the variability of the vocabulary.

Also, there is no punishment for individual activities, a potential bias of gaming behavior will challenge the validity of our results. So we assume users sign in a petition will definitely comment under that petition and willing participant in.

3.3. Approach Details

We did research both on a user and petition relationships.

3.3.1 User research approach

In users' network analysis part, we select features dimension as 24 and length of walk per source as 3 for node2vec model. By inputting the edge information from a subset of 100,000 users(due to the fact that a single petition attracts more users than expectation), we could get feature vectors for each selected node and use them as sample input for topic prediction in the regression model. In the label vector, where one indicates the existence of one of a topic, like human right or immigration. This rule also works in the prediction output from the testing case. To evaluate model performance, we split those users features by choosing 0.4 of them to be tests and rest of them for training process to learn. Finally, we come up with precision and recall value for binary classification task to evaluate our model performance.

Words classification
HTML tags, $< \backslash a >< \backslash p >, < \backslash t >$
preposition, like, to
adjectivelike amazing, wonderful
interrogative, when, where, which
yes, no
name of people
time, year, month
and so on

Table 2. Meaningless words should be discards.

3.3.2 Petition research approach

A simple example of how word2vec work, is presented like this. If we have an expression and operation like vector(”King”) - vector(”Man”) + vector(”Woman”), where each string is represented in vector space in the same dimension we have pre-selected. After the operation and computation process, the outcome should be vector representation that is closely similar to vector(”Queen).

Based on this mechanism, we pick out petition ID and petitions' description from raw data as target dataset that work for the analysis task. First, we need to preprocess original descriptions by removing common words and HTML tags in the JSON file, like a preposition and HTML tags like $< \backslash a >< \backslash p >, < \backslash t >$, which are meaningless in semantic.

Each petition then is represented in one labeled sentence of clean keywords by the unique identifier, which is its petition ID. What we mean by clean is that words from petitions' description that could effectively conclude and stand for the core thought from that petition.

After constructing all sentences into one ”document”, we train it with the assist of doc2vec as a black box technique, a word2vec based model. During the training process, the model incorporates and establish the corresponding vocabulary for all petitions, where we choose feature dimension to be 128 and window size to be 8.

At the end of the training process, we could get vector representation for each sentence in 128 dimensions as we defined. And those features serve as input for downstream machine learning tasks, like multi-label classification. Along the training task, we attain similarity value for each pair of petitions based on the vocabulary pool. The value ranges from zero to one, the more close to one the more similar is this pair of petitions. By looking up into petitions, we then select a threshold 0.9 that could meet the requirement to reasonably distinguish between various and less connected petitions. This decision value determines in which extent we could link two petitions to build petitions network.

Finally in a multi-label regression model, due to highly freedom in constructing tags, we need to remove users own created tags and remain the website provided selection to avoid sparse tag matrix. Based on integrated tags, we again split the dataset into training and test set and evaluate the model performance in terms of F1 micro and F1 macro.

To make results more convinced, we do another three contrast experiment. First contrast experiment, we train the petition network and get 128 dimension feature vector with node2vec and combine it with 128 dimension feature vector we mentioned above(Got by word2vec) to get 256 dimension feature vector. After that, we do PCA to reduce this combined feature vector to 128 dimensions. And we do accuracy, f1 micro and f1 macro. By doing this, we can get a general idea would combine feature vector affect these three results or not. The second contrast experiment, we set all the prediction values to 0 and calculate accuracy, f1 macro and f1 micro. By doing this, we can figure out our model's performance. The third contrast experiment, we use feature vector which extracted by node2vec only to calculate three parameters. By doing this, we can compare the overall model performance.

4. Setup and Results

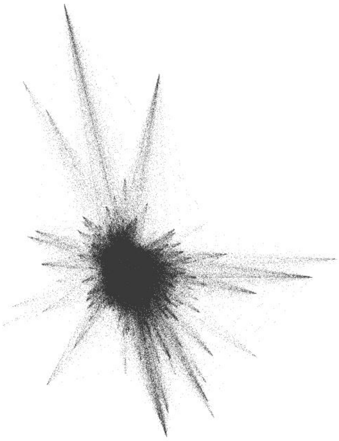


Figure 5. Petitions Network from 21 Topics

Attributes	Value
Average node degree	2.419
Network diameter	5
Average path length	1.682
Average clustering coefficient	0
Modularity	0.888
Number of communities	403

Table 3. Users network properties.

After we collecting about 1,200 petitions under human rights and immigration topic, we constructed 650,000 relations among users into social network visualized as Fig.3.

The attributes from user network is listed in table 1. And the degree distribution in log-log space is shown as Fig.6.

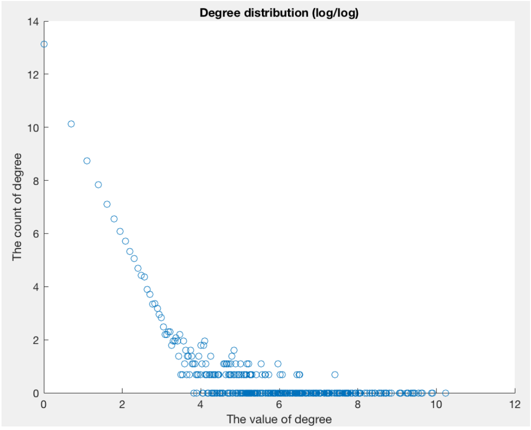


Figure 6. The degree distribution of users network in loglog space.

Then we picked out a subset of 100,000 users and applied linear regression on it with feature vectors from node2vec as input, we got an accuracy of 0.76 in prediction for users' preference for topics.

By looking up into petitions, we find that similarity values between petitions from same topics lie around 0.85. And for those totally independent and diverse from others, similarity values between them could be as low as to 0.5. We then select a decision boundary of 0.85 to cut off various and less connected petitions.

Based on the threshold we choose, we then built the network for 5,033 petitions from 21 topics shown as Fig.7 and the degree distribution as Fig.8. Fundamental properties are listed in table 4. After word2vec training process applied on petitions' description, we gained vector representation of keywords and used them for downstream machine learning process. Additionally, we dug into a subset of 1,540 petitions to find out what is most popular and influential content that contributes to petition victory.

Attributes	Value
Average node degree	138
Network diameter	16
Average path length	3.43
Average clustering coefficient	0.25
Modularity	0.557
Number of communities	14

Table 4. Petitions network properties.

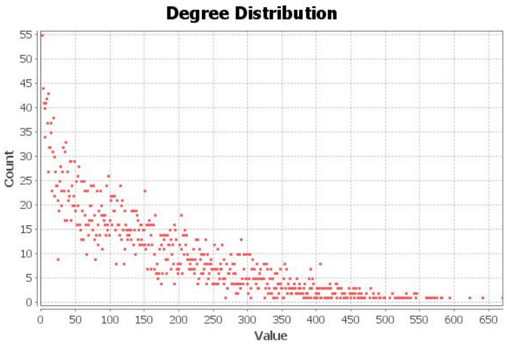
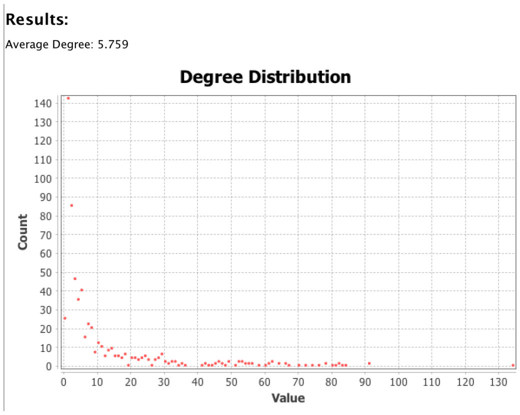


Figure 7. Degree Distribution in Petition Network


Results:

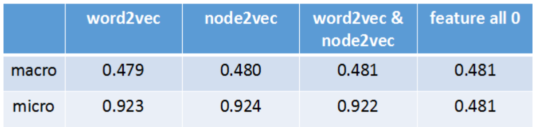
Average Degree: 5.759

Figure 8. Degree Distribution from Petition Network with Victory Status

We applied multilabel regression on entire petitions network, where each petition may own several tags. If a certain petition contained one tag like ”animals”, then the value from the corresponding index of label matrix was set to one. The Same rule applied to the prediction process, one indicated the existence of that tag word. We kept all the official tags, that is to say, capital ones, and the vocabulary size reached 1494.

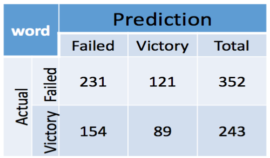
The F1 micro is 0.99 and F1 macro is 0.71 for this task. And this is due to the high diversity size of tags pool and the label matrix is sparse. So we shrink our tags to most frequent 20 ones and fix our experimental set up by keeping petitions with those tags only. This time, the dataset consists of 2557 petitions and we implement the same multilabel regression task. Meanwhile, we tried three ways with different features from training process and evaluate the model in terms of f1 score. When combining both features from word2vec and node2vec for the same multilabel task, the performance has not improved and slightly degenerated. The entire statistic results from the different features input for multi-tags prediction is shown as Fig.8. Where we add extra to experiment with all tags to be zero, which may help us to interpret our model performance.

Besides, we also stuck on researchers in the victory status for petitions. In this section, we started from a dataset of 1539 petitions with their victory status. The degree distribution from this small group of petitions is shown as Fig.7. Although the size of our dataset is limited, we are still able to tell the fitting line goes in the same trend as the power law. To fully understand how binary regression model performed, we split the whole dataset in the same way as before. Label vector values only consisted of binary value, where one indicated the victory status and zero stood for



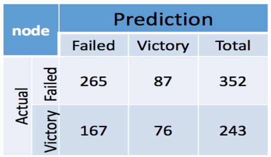
	word2vec	node2vec	word2vec & node2vec	feature all 0
macro	0.479	0.480	0.481	0.481
micro	0.923	0.924	0.922	0.481

Figure 9. Multilabel regression for tags prediction



word		Prediction		
		Failed	Victory	Total
Actual	Failed	231	121	352
	Victory	154	89	243

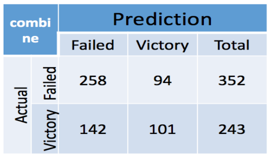
Figure 10. Only word2vec features for victory prediction



node		Prediction		
		Failed	Victory	Total
Actual	Failed	265	87	352
	Victory	167	76	243

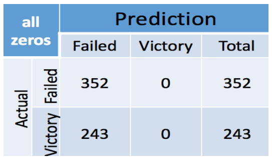
Figure 11. Only node2vec features for victory prediction

petition failure. Interestingly, the precision and recall value both increased, not same as the multilabel problem. The statistic result from original regression problem relied on word2vec and node2vec is shown as Fig.9 and Fig.10 respectively. The result of combining features for regression task is shown as Fig.11 and Fig.12 with all zeros label.



combi ne		Prediction		
		Failed	Victory	Total
Actual	Failed	258	94	352
	Victory	142	101	243

Figure 12. Combined features for victory prediction



all zeros		Prediction		
		Failed	Victory	Total
Actual	Failed	352	0	352
	Victory	243	0	243

Figure 13. Combined features for all zero victory prediction in all zeros

4.1. Analysis

As an average degree in users-net is low, we may reason that it is due to the small size of our current data set. Because in the real world, people are more likely to be interested in various topics. Hubs with the great degree of 30,000 located at popular and people concerned topics. We can easily tell there are various communities spreading among the network, which means people share the same interest in the certain aspect that they get involved more than one activity.

In petitions network, the average degree is relatively high, which means petitions, in general, are parallel with others. And degree distribution from petitions network follows the power law, where we may infer petitions from Change.org finally form into a scale-free network. The number of community detected from Gephi is reasonable and close to the number of topics.

Regression model from users network to predict topic preference performed well and this contributes to finding out new users most likely behavior. This accuracy value is 0.76 and this could be improved if we try higher dimension, which may contribute to better-capturing features from users.

In regression task for petitions network to append multi tags for each petition, in order to balance highly sparse label matrix where we had an uneven class distribution, the cost of false positives and false negatives were very different, so we chose F1 micro and F1 macro to evaluate the model. Based on the result, which implied that word2vec is able to capture petitions' descriptions semantic features and could serve as good representation for machine learning process.

We tried to combine features for both binary and multilabel task, and we were not surprised to find that simply combined features were not good for multilabel classification while could improve the performance for victory prediction. The possible explanation is that predicting victory is harder. To predict topic, the features we get from doc2vec directly correspond to the content of the petitions and so it is not very hard to predict the tags. But for victory prediction, doc2vec features, which only contain the English language based features, and thus are not sufficient to determine whether a petition will become victorious. Similarly, the social network features obtained from node2vec are not sufficient in themselves to predict victory. Therefore, predicting victory is harder as it is governed by many hidden factors which are loosely connected to networks but not completely determined from the network information. And this is precisely the reason why the accuracy of a combined model should better than the individual models since it combines information from both language and social network.

In the analysis of the relation between victory and petition content, we find out the five most involving categories are economic justice, criminal justice, education, women, health and safety as shown in Fig.13. Petitions with these content are more likely to get a victory. While popular topic like Donald Trump with over 20,000 petitions actually gets the lower chance of winning the petition. In which extent are petitioned popular among others cannot ensure the success of the event.

4.2. Limitation and Solutions

Due to the relatively small size of the petition, the vocabulary corpus is not comprehensive enough. And this makes it difficult to directly pick up same parameters as the default setting to train the data. It is obvious that various combinations of feature dimension and window sizes will give out different performance. A better and possible way is to validate our model by computing accuracy while we apply the model on tests, which means extra efforts in constructing words pool and benchmark to evaluate the similarity.

Another problem that derives from the previous limitation is that we are not able to totally rely on the word2vec results, where we use the similarity values between each pair of petitions from our training process for petitions network construction. While in subsequent regression tasks, we adopt features from petitions' network in node2vec training process without any hesitation. And this could lead to uncertainty in research results due to the dependent relationship between word2vec and node2vec, how we choose the threshold to link two petitions could make a big change in our network structure and turn the results upside down.

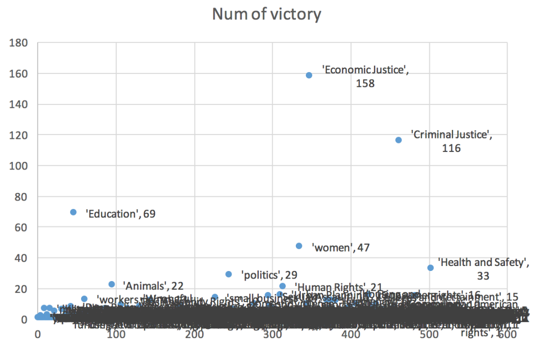


Figure 14. Histogram of tags distribution in petitions subset

5. Conclusion and plan

5.1. Summarize

By data collection from Change.org, we catch real-world phenomena and analyze it with an assist of machine learning tools. After model training, we are easy to visualize intangible social network and find it is close to scale-free network, where hubs locate at active people and trendy topics. Communities come into existence due to the prosocial and common behaviors among people, those densely connected group are more likely to invoke neighbors to take similar actions. The popularity of an event cannot lead to the success of the event. And Word2vec is a good method in languages analysis for semantic features extraction, while less competitive in collecting features from network properties.

5.2. Future Plan

Based on the testing result from regression problem we have got so far, we are more interested in finding out multiple analysis perspectives for networks, especially for petitions' victory prediction. Thus we decide to keep on the comparison between word2vec, node2vec, and combination of word2vec and node2vec that applied on petitions' network, with the adoption of F1 score metric we are able to determine which one could outperform among others to construct feature representation of the network. Thus we would keep collecting more petitions data and scaling up our problem to see whether the model performance is stable or would get better. Real world social networks are more closed to scale free network, both for users and petitions from Change.org online website. Petitions with popular content like economic justice and education are more inclined to get victory.

5.3. Contribution

Each of us collected as much as petition data from different topics. W learned implement details of the machine learning tool and gave out data format requirements so that Q focused on pre-processing data for subsequent training. Both of us involved in network analysis. We shared responsibility for report and presentation preparation.

5.4. What we have learned

This is an interesting project and related to the real world. The idea that everything can be abstracted to a network, not only the social media, the integrated circuit, even the nervous tissue, the brain is really impressive. We faced a lot of challenges when beginning this project, we have no idea about machine learning at first, but Kartikeya helps us. He is really a nice TA and gave us the general idea about node2vec and word2vec and faced challenges with us! And the data set is really huge, we need tons of hours to run, like run accuracy. We think if we can optimize the code, it will be faster. For the data is from the real world, there may be a lot of junk data, so the model is not that good, but just fine. We think there must be a better way to clean this data quickly and efficiently. Finally, when complete this project, we learn more than just doing course project, the different test style like presentation, demo and poster makes us understand better of our research and think aloud. We appreciate all the effort Radu and TAs made. Thank you very much.

References

[1] J. L. Aditya Grover. node2vec: Scalable feature learning for networks. 2016.

[2] K. D. M. M. R. T. Allcott, H and A. Szeidl. Community size and network closure. *American Economic Review Papers and Proceeding*, 97:80–85, Nov 2007.

[3] C. D. Batson. Altruism and prosocial behavior. *The Handbook of Social Psychology (4th ed.)*, pages 282–316, Oct 1998.

[4] R. S. BURT. The social structure of competition. 1992.

[5] S. V. D.Yim. Networks of green people: Dynamic network closure and prosocial behavior in online communities. *Conference on Information Systems and Technology*, 2010.

[6] U. F. Ernst Fehrl. The nature of human altruism. *Nature*, pages 785–791, Oct 2003.

[7] J. Farell. Network structure and influence of the climate change counter-movement. *Nature Climate Change*, Nov 2015.

[8] S. Moon. The role of feedback in managing the internet-based volunteer work force. 2008.

[9] T. M. Quoc Le. Distributed representation os sentences and documents. May 2014.

[10] G. A. G. Thomas Dietz, Paul C. Stern. Social structural and social psychological bases of environmental concern. *Environment and Behavior*, pages 450–472, July 1998.

[11] G. C. J. D. Tomas Mikolov, Kai Chen. Efficient estimation of word representation in vector space. Sep 2013.