project-016: I523
Data Analysis of Interaction Dynamics for Persuasion

F16-DG-4064, Carlos Sathler
Graduate Student, MS in Data Science
Indiana University at Bloomington
cssathler@gmail.com

ABSTRACT

This project explores the work of a group of researchers from Cornell University published on the paper Winning Arguments: Interaction Dynamics and Persuasion Strategies in Good-Faith Online Discussions [1]. The authors researched interaction dynamics leading to persuasion, as well as language conducive to persuasion. On the present project we restrict ourselves to analysing some aspects of interaction dynamics that the authors did not address in their paper, we explore potential bias in the data resulting from highly skilled users, as well as users who may be highly susceptible to persuasion, and we perform predictive analytics based on interaction dynamics of discussions.

1. INTRODUCTION

In February and March of 2016 a group of scientists from Cornell University made the news [2][3] when they reported the results of a study about persuasion in which they used online data collected from a Reddit community called ”Change My View” [4]. The researchers were interested in understanding patterns of interaction and language that ultimately led to one user changing another user's opinion about a given topic. The study behind the news is documented on a paper presented at the 25th International Conference on World Wide Web [1].

Our project uses the same dataset these researches used in their study. But due to time limitations it does not analyze language and how it impacts persuasion. Instead, we explored features of interaction dynamics that we later use for prediction. Additionally, we looked for evidence that certain users wouldn't be skewing the data by possessing advanced persuasion skills, or by being highly malleable. In order to better understand persuasion, in the first case it would make sense to focus on highly skilled persuaders before anything else, and in the latter case it would make sense to ignore users who are too easy persuaded.

In our analysis we tried to answer 5 questions, each of which will be addressed in a separate section of this report:

1. Was the data properly loaded?
2. Are certain users more persuasive than others?
3. Are certain users more susceptible to persuasion?
4. Is persuasion more likely at certain times of the day or days of the week?
5. Is there a relationship between the shape of a discussion tree and persuasion?

The last question was also explored by the original research. We will compare results obtained in our project with those of the original research.

At the end of the analysis we identify a set of features that we use to perform predictive analytics with the Python scikit-learn package.

2. DATASET

In this section we describe the project dataset.

2.1 Source

The dataset for this work was released in January of 2016 and distributed with the original research paper that inspired our project [1]. It can be downloaded from one of the author's (Mr. Chenhao Tan) web page [5] at the following url: https://chenhaot.com/pages/changemyview.html

2.2 Data Structure

The dataset used on the project is a collection of online discussions. Each discussion consists of a tree structure where the root node is the initial post containing an opinion from a community user along with a invitation abbreviated by the initials ”CMV”, as in ”Change My View”.

First users who respond to the post initiating each discussion will have the option to start a new thread under the original post or offer their opinion under an existing thread created by a prior user. This logic does not change as a discussion tree grows; every user has a chance to expand the tree by either adding a sibling node or a child node, each node being a post.

2.3 Definitions

We will borrow some of the definitions from the original research that inspired our project [1, p 616, 617]:

- Discussion Tree - Each discussion originated by a user with the invitation ”CMV” (as in ”Change My View”).
- OP - The author of the post that initiates a discussion tree.
- Original Post - Post by OP that initiates a discussion tree.

- Root Reply - A direct reply to OP post that initiated a discussion tree.
- Path - All replies from OP to a given leaf.
- Challenger - Any user who accepts the invitation from the OP and attempts to change the OP's view.
- Root Challenger - User who posted a Root Reply.

Other terms that are important for the project:

- Comment - A post following the OP original post.
- Reply - Same as Comment.
- Post - Same as Comment
- Delta - An acknowledgement offered by one user to indicate that another user changed their view.

Note that Deltas can be granted by the users responsible for the Original Post to express a Challenger was successful, but they can also be granted by other users to each other. Additional information on how the Reddit community Change My View [4] works can be found in the community Wiki https://www.reddit.com/r/changemyview/wiki/index.

2.4 Dataset Statistics

The following dataset statistics are reported in [1, p 615]:

- Number of Discussion Trees: 20,626
- Number of nodes: 146,847.533
- Number of unique participants in the discussions: 86,888

The research dataset is actually partitioned into a training dataset and a testing dataset ("heldout"). The above numbers are grand totals. We obtained close numbers for these counts (through program "question0.py"). We attribute the minor differences that we found to data filtering and cleaning when loading the data.

The size of the training partition of the raw dataset is 2.23 GB; the size of the test ("heldout") partition of the raw dataset is 300.5 MB.

3. DATA LOAD

In this section we describe the process of loading the raw data for the project and creating our own data structures used for analysis and prediction.

3.1 Source Data Format

Source data is available online in compressed format at [6]. The raw data consists of a list of JSON records each record containing all the information in a discussion tree, including OP, node id for each post, comments for each post and parent id of each node. Delta awards are embedded in the discussion tree comments.

3.2 Load Process

Program "load_data.py" was used to load the data. It traverses the JSON list in the raw data files (one for training data, and one for test data) and creates a number of output files that store all relevant discussion data information that will be used on the project. These files are described below.

Because the amount of data being processed is so large (training data file size is 2.23 GB) even a powerful personal computer (16G of RAM) could not process all JSON records in one shot. So the load program saves discussion data to thousand of temporary files as it reads the raw data. These files are later concatenated into the final output files that will be used in the project. The process is cumbersome but it worked well and it allowed us to conclude the project using a personal computer.

3.3 Data Load Output

Raw data is loaded into three files that we manipulated as SQL tables using the Python Pandas package [7][8] throughout the project. The files store data about discussion trees, discussion tree nodes, and node comments. Because the data is partitioned between a training and a test dataset, and because some of the training data had to be filtered for analysis, a total of 10 data files was created from the raw data. File names corresponding to all data files used in the project are shown in Table 1.

CSV File Name	Description
ops_train	Discussion tree information
ops_train_filtered	Same as above, filtered
ops_test	Same as before, for test data
ops_test_filtered	Same as above, filtered
disc_tree_nodes$_train$	Tree nodes information
disc_tree_nodes_train_filtered	Same as above, filtered
disc_tree_nodes_test	Same as before, for test data
disc_tree_nodes_test_filtered	Same as above, filtered
raw_comments_train	Post comments
raw_comments_test	Same as above, for test data

Table 1: CSV Files Created by "load_data.py"

Tree information is captured in the "ops_*.csv" files which contain data about each discussion root node, including OP id, and URL for the discussion in the CMV website [4]. Tree size is calculated by the load program, as well as whether or not there was a change of view in the discussion tree (Delta award). The specs for this file are shown in Table 2.

Field Name	Description
ID	Node id from raw data
OpAuthor	OP user id
Title	Title of the post
URL	Url of the discussion tree
TreeSize	Number of nodes in discussion tree
DeltaTree_TF	Was there a Delta in this tree?

Table 2: OP Posts CSV File Specs

Tree node information is captured in the "disc_tree_nodes_*.csv" files and contain data about each node of each discussion tree, including node id, parent id, and a number of node attributes such as node level, degree and height [9], some of which will be used to explore interaction dynamics. This information is not present in the raw data so the load program needs to calculates those fields during the load. The specs for the tree node information files are shown in Table 3.

Field Name	Description
ID	Node id from raw data
Parent_id	Node id of parent node
Author	Userid of post author
Op_author	Userid of OP who initiated discussion
PostTimeRaw	Raw time (secs since Jan 1, 1970)
Degree	Number children nodes [9]
Level	Distance to root node [9]
Height	Distance to leaf with highest level [9]
DeltaNode_TF	Was this node awarded a Delta?
DeltaAward_TF	Was this a node awarding a Delta?
Root_id	Node id of the root of this tree

Table 3: Discussion Tree CSV File Specs

Post comment are stored on separately in the "raw_comments _*.csv" files. The specs for the comments files are shown in Table 4.

Field Name	Description
ID	Node id from raw data
Parent_id	Node id of parent node
Author	Userid of post author
Op_author	Userid of OP who initiated discussion
Comm	Comment
Comm_html	Comment with html tags

Table 4: Comments CSV File Specs

3.4 Identifying Deltas

Critical to the data load is the identification of Delta nodes, i.e., those nodes that persuaded the OP. Deltas are captured in a two step process: first the OP replies to the challenger by including a special character in his comment (a "Δ" in most operating systems) to officially signal that he/she has been persuaded to change his/her view. A Delta award by the OP signifies to the challenger "yes, you have changed my view". Second, a "DeltaBot" which runs periodically on the CMV site [4] will confirm the Delta.

Data load was not trivial, particularly the identification of Deltas. Initial counts following our data load were not matching the counts of the original paper [1]. We contacted the authors to ensure we were identifying Deltas in the raw data in the same way they did. The email exchange with them is captured in the Appendix.

3.5 Cleanup and Filtering

The clean up consisted mainly in extracting a handful of relevant fields from a complex JSON record structure. These fields were described above. Additionally, some records in the JSON file were corrupted and that had to be addressed as well by making sure the load program would skip past those corrupt records.

Filtering followed the guidelines put forth in the original article [1], which were reiterated in the email exchange we had with the authors of the article (please refer to Appendix). For the purpose of exploratory data analysis of the factors that drive persuasion, only original posts with at least 10 replies were considered. Additionally, we eliminated from our analysis those discussion trees in which an OP simply posts a challenge but never engages with his/her challengers (no posts after original posts) [1, p 620].

4. EXPLORATORY DATA ANALYSIS

When working with the data we preserved data partitions. All exploratory data analysis was performed using the data files created by the load process (see Load Data Output section of this document). For the first question in the exploratory data analysis we used the unfiltered training datasets. All other questions were answered with the "filtered" training data files.

4.1 Was the data properly loaded?

After confirming the counts for number of discussion trees, number of nodes and unique participants in the discussion trees (through program question0.py) we replicated four plots the authors used in the original paper to describe the raw data [1, p 169]. These plots are shown in Figure 1.

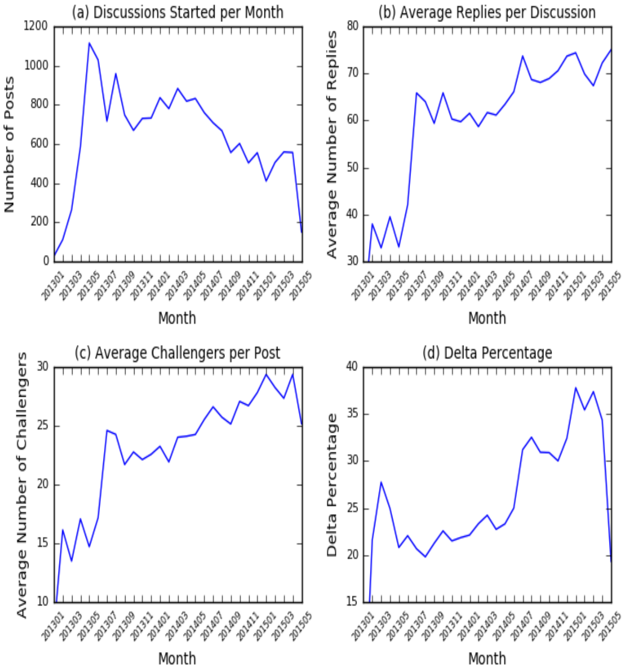

Figure 1: Monthly Activity Metrics [1, p 169]

Plot (a) is created by grouping number of posts per month. A post in this case is any comment by any user captured in the system; plot (b) shows the average number of root replies to original post by OP's, grouped by month; plot (c) shows average number of distinct root challengers by month and plot (d) shows percentage of discussion trees in which the OP awards a Delta to one or more challengers, grouped by month.

The plots produced by our project and the plots presented by the authors in their paper [1, p 169] are almost identical allowing us to conclude that the data load process for our project is correct.

This question was answered with program ”question1.py”.

4.2 Are there highly persuasive users?

This questions asks if there are users that we should study more closely, as opposed to anything else, because they perform significantly better than the rest of the user population. These users would be highly skilled persuaders that use special language, or know exactly when to challenge an OP in order to receive a Delta. Knowing what they do could unlock the secret of successful persuasion.

Figure 2 shows four histograms that group users based on the percentage of successes getting a Delta per total posts submitted. Plot (a) shows the entire universe of users in the dataset; Histogram (b) shows that the average percentage of successful posts is close to zero; histograms (c) and (d) attempt to identify a potential set of users who could be considered highly skilled persuaders. Particularly (d) shows 366 users who achieved higher success persuading OP’s. They posted an average of 8 posts each and got an average of 1 Delta each. Because this doesn’t seem like a lot of data, it is fair to ask, would these users have performed this well had they posted more challenges? Figure 3 suggests that no, we should not expect that. The figure shows the Spearman Correlation [10] between Delta received and number of posts. The Spearman Correlation coefficient is not close to 1 indicating a week correlation with a p-value of 0.

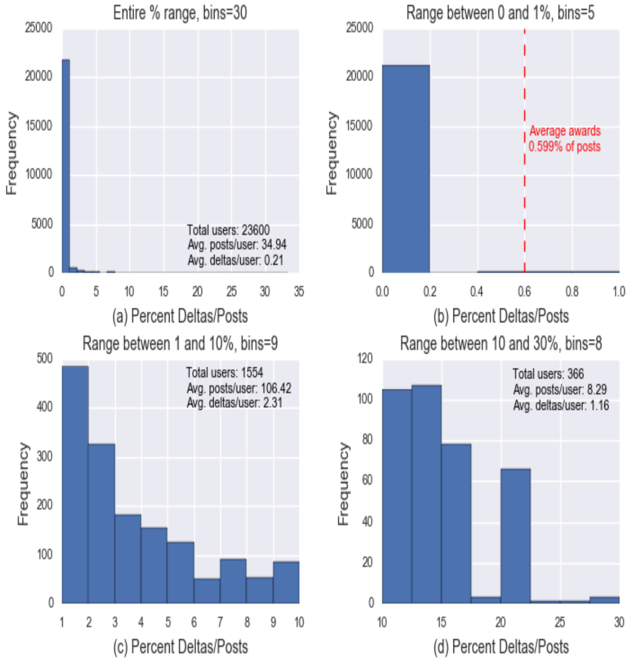

Figure 2: Percentage of Delta Awards for OP’s

We conclude it is not a good strategy to study what makes a post persuasive by focusing on certain presumably highly skilled users; there is no evidence of such users in the project dataset.

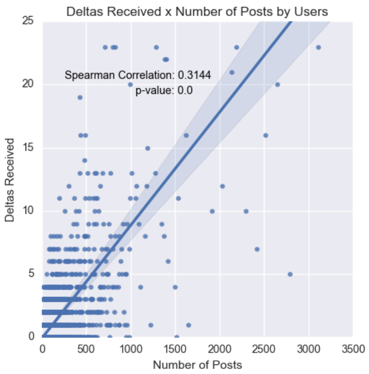

Figure 3: Deltas Received vs. Posts per User

This question was answered with program ”question2.py”

4.3 Are there highly malleable users?

Here we look for indications that certain users may be more malleable, i.e., more easily susceptible to persuasion; we consider the risk of inferring that certain posts’ contents or interaction dynamics lead to persuasion when actually the persuasion could be better explained by the fact certain users are highly malleable. If such users exist we would want to exclude them from the analysis, or focus the analysis on them to study what behavioral patterns can be associated with malleability.

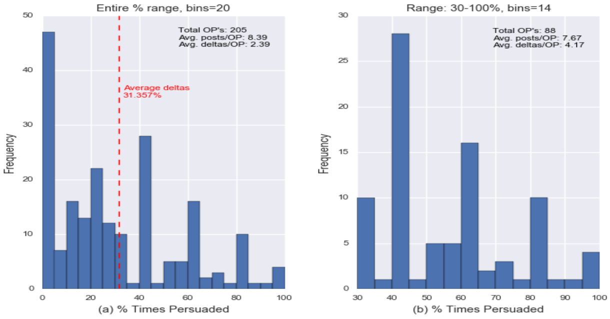

Figure 4: Percentage of Times Authors Persuaded

In order to eliminate infrequent users we only include in this analysis OP’s who initiated at least five discussions during the period of data collection. Only 205 out of 8,299 initiated 5 or more discussions. Figure 4 shows the result of this analysis. Histogram (a) charts these 205 OP’s and their Delta percentage. They awarded a Delta an average of about 31% of the time. Histogram (b) shows OP’s who awarded Deltas above the average. Our data only includes 88 such users and the average number of discussions started by the group is 7.67. These are small numbers considering a total population of 8,299 OP’s. Still, would these users continue the same pattern should they initiate more discussions? Are they clearly more malleable? Figure 5 shows the Spearman Correlation [10] between Delta awards and number of discussion initiated by OP's. There is no significant correlation (0.2355 coefficient; p-value close to 0), so we conclude there is no significant evidence certain users in the population should be considered more malleable.

This question was answered with program "question3.py".

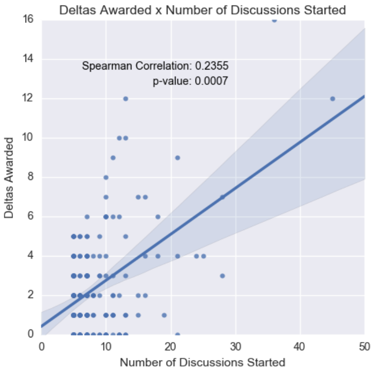

Figure 5: Deltas Awarded vs. Posts per User

4.4 Is persuasion more likely at certain times of the day or day of the week?

Will challengers have a better chance of persuading the OP based on when they post their challenge? Figure 6 tries to answer that question. Histogram (a) shows a distribution that is mostly uniform, with a slightly higher incidence of Deltas on Mondays; histogram (b) indicates most persuasive posts occur at the beginning or at the end of the day.

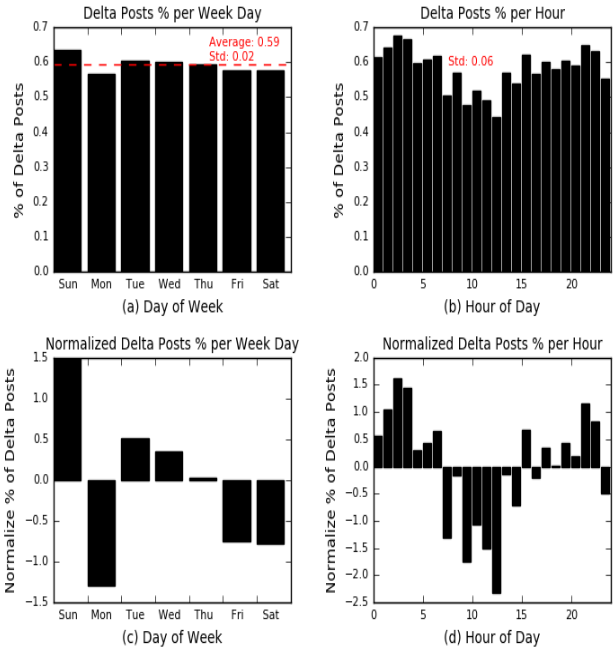

Figure 6: Persuasion vs. Time and Week Day

This question was answered with program "question4.py".

4.5 Is there a relationship between the shape of the discussion tree and persuasion?

We use two attributes of tree nodes [9] to explore the shape of the discussion tree where a Delta was a warded. The attributes are node level (root level = 0) and node degree (number of children).

The authors of the original paper [1] conducted a line of inquiry similar to the line of inquiry we explore with the present question. They found evidence that a larger number of challengers will increase the probability of persuasion and that in general the earlier a challenger enters a discussion the higher his/her chance of persuading the OP [1, p 616-617].

Histogram (a) in Figure 7 shows a decreasing incidence of Delta awards for posts that happens "deeper" in the discussion; the second histogram on the figure shows that nodes with more siblings have a higher chance of being persuasive. We realize that this analysis includes post-hoc comments (children nodes created after the Delta award), but do to time constraints we decided not to remediate this issue.

This question was answered with program "question5.py".

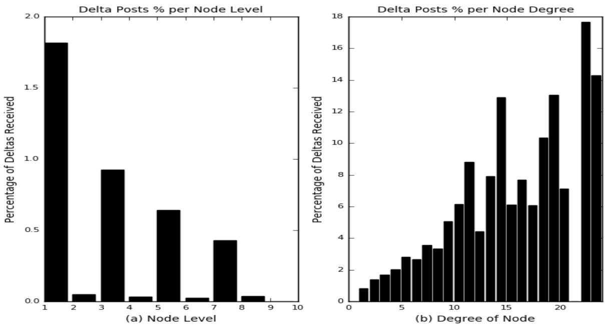

Figure 7: Persuasion vs. Node Level and Degree

5. PREDICTIVE ANALYTICS

For the first round of predictive analytics we only included the features analysed above: time of post, day of the week; level and degree of post [9]. We improved our prediction metrics by adding two more features: proximity to closest OP post and number of words in post. The idea of including number of words in the posts comes from the original article [1, p 620].

Final list of post features selected for prediction:

- Day of the week
- Time of day
- Post level
- Post degree
- Number of words in post
- Distance to closes OP post

We used four classification algorithms for prediction:

- Logistic Regression (l2)
- KNN Classifier (k=5)
- Decision Tree Classifier
- Random Forest

For preprocessing we standardized all features and we created dummy variables for the day of the week feature. We also imputed values for number of words for some posts that were missing comments using mean value.

Accuracy of predictive models was high, but the high accuracy can be attributed to the low incidence of persuasive posts. For the filtered training dataset, the percentage of persuasive posts is a mere 0.71% (after removal of DeltaBot posts, and posts by the OP). So, predicting all posts as non-persuasive would automatically yield a 99.21% accuracy. Considering the characteristics of the project data, a more valuable performance measure will effectively identify persuasive posts.

With that in mind, we used Python's confusion matrix functionality [11] to evaluate our models and we focused on Positive Predictive Value and Sensitivity.

Below we explain the metrics used in our evaluation [12]. Results are summarized in table 5.

- TP - True Positives
  Number of correct predictions of persuasive posts.
- TN - True Negatives
  Number of correct predictions of non-persuasive posts.
- FP - False Positives
  Number of non-persuasive posts mistaken as persuasive.
- FN - False Negatives
  Number of persuasive posts mistaken as non-persuasive.
- Accuracy = (TP+TN) / (TP+TN+FP+FN)
  Percentage of correct predictions.
- Positive Predictive Value = TP / (TP+FP)
  Accuracy in predicting persuasive posts.
- Sensitivity = TP / (TP+FN)
  Accuracy in identifying persuasive posts.

Predictive analytics was implemented with "prediction.py".

6. COMPARISON WITH ORIGINAL RESEARCH

Our project reproduces basic analysis from the original paper [1] in Figure 1 and it extends the analysis documented in the paper by asking the questions 2 through 5 described in the introduction to this report.

Model	Accuracy	Predictive Value	Sensitivity
Logistic Regression	98.6027%	32.0000%	0.5253%
KNN Classifier	98.5972%	14.2857%	0.1970%
Decision Tree Classifier	98.4322%	10.8%	1.7728%
Random Forest	98.4623%	9.4527%	1.2475%

Table 5: Prediction Results ("predict.py")

One could argue that question 5 is approached in the original paper; the authors analyze "entry order" [1, p 616] and "number of challengers" [p 617] which can be roughly equated to "level" and "degree" [9] used on our project. But we consider our approach original because "level" and "degree" are not directly explored at each node by the authors of the original paper.

Regarding predictive analytics, our project approach is mostly original. It focuses on features relating to interaction dynamics that were not documented in [1]. The only exception is number of words in posts, which we included in our prediction based on findings documented in the original paper [p 618]. Unlike the original research, we did not use language features in your predictions.

We did not seek access to the code used in the original research. The only code on our project that can be attributed to the authors deals with the manipulation of tar files for download of raw data for the project. The code was obtained from a blog authored by one of the authors [13]. The very small part of the code shared on the blog that was reused on our project is cited as a comment in program "util_download_data.py". Some code snippets from StackOverflow [14] were also reused and they are properly cited as code comments where appropriate.

7. CONCLUSIONS

Our results highlight the limitation of doing prediction on post persuasiveness with interaction dynamics' features alone. While accuracy was relatively high, positive predictive value was very low and so was sensitivity in all models. It would have been interesting, with more time, to explore additional aspects of interaction dynamics, such as degree of parent nodes. Yet, based on our project results we feel confident that a successful predictive model of persuasiveness of CMV Reddit posts [5] will have to include language features.

8. SELF-EVALUATION AND REFLECTION ON PROJECT ACHIEVEMENTS

Most of the goals set forth in the project proposal were achieved. There was no need to use regular expressions on the project, so that learning objective was not achieved. On the project proposal, it was stated that language analysis would only occur "time permitting". In the end, there was no time for natural language analysis. The data load for the project was complex. The size of the project raw data required a design change that delayed the project; we had to create temporary files on disk during the load, since a personal computer could not process the project raw data in memory. Additionally, the load takes 3 hours and test cycles were long.

All in all, we learned a great deal on this project, about Python [15][16][13][14], Pandas [7][17][8], Numpy [18], Matplotlib [19] and Searborn [20], Scikit-Learn [21] and machine learning in Python [22]. And we reached a meaningful conclusion: language features must to be incorporated in predictive models of persuasiveness of posts in the CMV Reddit community [4].

9. REFERENCES

[1] C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, and L. Lee, "Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions," in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 613–624. [Online]. Available: http://dx.doi.org/10.1145/2872427.2883081

[2] Ana Swanson, The Washington Post, "How to change someone's mind, according to science," February 2016. [Online]. Available: https://www.washingtonpost.com/news/wonk/wp/2016/02/10/how-to-change-someones-mind-according-to-science/

[3] Tanya Basu, Science of Us, "A subreddit sparked a scientific inquiry into how to change someone's mind," February 2016. [Online]. Available: http://nymag.com/scienceofus/2016/02/subreddit-sparked-a-study-on-changing-minds.html

[4] "Change My View (CMV) Reddit Community," Web Page. [Online]. Available: https://www.reddit.com/r/changemyview/

[5] C. Tan, "Chenhao Tan's Homepage - changemyview," Web Page. [Online]. Available: https://chenhaot.com/pages/changemyview.html

[6] "Data for the CMV Research," Downloadable File, https://chenhaot.com/data/cmv/cmv.tar.bz2. [Online]. Available: https://chenhaot.com/data/cmv/cmv.tar.bz2

[7] "Python data analysis library - pandas: Python data analysis library." [Online]. Available: http://pandas.pydata.org/

[8] "Comparison with SQL - pandas 0.18.1 documentation." [Online]. Available: http://pandas.pydata.org/pandas-docs/version/0.18.1/comparison_with_sql.html

[9] "Tree Terminology :: Data Structures." [Online]. Available: http://btechsmartclass.com/DS/U3_T1.html

[10] "Spearman's rank correlation coefficient - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient

[11] "Confusion matrix - scikit-learn 0.18.1 documentation." [Online]. Available: http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

[12] "Confusion matrix - wikipedia," Web Page. [Online]. Available: https://en.wikipedia.org/wiki/Confusion_matrix

[13] V. Niculae, "Winning Arguments and Attitude Change on Reddit," Blog. [Online]. Available: https://vene.ro/blog/winning-arguments-attitude-change-reddit-cmv.html

[14] "Stackoverflow," a couple citations, including specific URLs appear in some of the programs created for the project. [Online]. Available: http://stackoverflow.com/

[15] "Python 2.7.12 documentation." [Online]. Available: https://docs.python.org/2/

[16] "The python tutorial - python 2.7.12 documentation." [Online]. Available: https://docs.python.org/2/tutorial/index.html

[17] "Working with DataFrames." [Online]. Available: http://www.gregreda.com/2013/10/26/working-with-pandas-dataframes/

[18] "NumPy - NumPy." [Online]. Available: http://www.numpy.org/

[19] "matplotlib: python plotting - Matplotlib 1.5.3 documentation." [Online]. Available: http://matplotlib.org/

[20] "Seaborn: statistical data visualization - seaborn 0.7.1 documentation." [Online]. Available: http://seaborn.pydata.org/index.html

[21] "scikit-learn Tutorials - scikit-learn 0.18.1 documentation." [Online]. Available: http://scikit-learn.org/stable/tutorial/index.html

[22] S. Raschka, "Python Machine Learning," 2015.

APPENDIX

Hi Carlos,

Thank you very much for carefully digging into these! The release data for prediction actually went through several filtering. I went back to check our code. I think that the delta percentage should be correct.

For the OP data: If you run 'wc -l train_op_data.jsonlist', the result should be 10743, which says that the overall average delta ratio is 30%. The filtering roughly includes two parts:

1. there are enough unique commenters (at least 10) to make sure that people actually tried to convince the OP

2. the OP did not say that they changed their opinions in the original post in edits.

For the training period data: deltabot is actually DeltaBot. 'bzcat train_period_data.jsonlist.bz2 | grep -i "deltabot" | wc -l' gives me 6088.

Hope that addressed your concerns! Let me know if you have more questions.

–Chenhao

On Sun, Oct 30, 2016 at 9:49 PM, Carlos Sathler <cssathler@gmail.com> wrote:
Good morning Vlad and Chenhao,

Thanks very much for your replies and offer to share your code. I think I’m fine with identifying delta posts.

My issue seems to be understanding ”delta percentage” as depicted in plot 3(d) from the paper, which I include below.

Please let me share with you without referring to my code, why I was expecting an average percentage between 17% and 22% at most.

I’m doing a grep of ”deltabot” on file ”all/train_period_data.jsonlist” and I get 4052 hits.

```
$ grep deltabot train_period_data.jsonlist | wc -l
4052
```

I’m doing a grep of delta labels on file ”op_task/train_op_data.jsonlist” and I get 3191 hits.

```
$ grep ’”delta_label”: true’ train_op_data.jsonlist | wc -l
3191
```

Considering the number of discussion trees in the data set is 18,363, I was expecting a delta percentage between 17 and 22

Do you know what I’m missing?

Thanks again,
Carlos Sathler

From: Chenhao Tan <——@——.com>
Date: Friday, October 28, 2016 at 4:29 PM
To: Vlad Niculae <—-@—–.ro>
Cc: Carlos Sathler <csathler@umail.iu.edu>, Carlos Sathler <cssathler@gmail.com>
Subject: Re: CMV dataset question: how many deltas did you count?

HI Carlos,

Thank you for your interests! Based on CMV’s description, there are various ways to input delta and on different systems it may reflect differently in the final utf-8 encoded text from Reddit. What we used is to identify posts by DeltaBot and check the author of the parent comment of DeltaBot’s comment. I am happy to share the code if you still have problems.

”
How to award a delta:
Reply to the user(s) who changed your view with ∆ included in your comment, which can be achieved through one of the following:
Copy/paste - ∆
&#8710; (Unicode - remember the semi-colon! - Windows, Mac, Linux, and Smartphones)
Option/Alt+J (Mac)
Ctrl+Shift+u2206 (Linux)
!delta (for mobile users, but please use ∆ if you can)
”

Thanks again!

–Chenhao

On Thu, Oct 27, 2016 at 7:20 AM, Vlad Niculae <—-@—-> wrote:
Hi Carlos,

Thank you for writing! I think this is a question of filtering, but I don’t know the answer off the top of my head. I’m CC’ing Chenhao, who might remember better. If not, we’ll surely be able to look it up.

Yours,
Vlad

On Thu, Oct 27, 2016 at 1:55 AM, Carlos Sathler <csathler@umail.iu.edu> wrote:
> Vlad,
>
> Please allow me to introduce myself. My name is Carlos Sathler and I am a
> graduate student at Indiana University. I’m writing to you because I’m
> studying the paper ”Winning Arguments...” and the dataset you made available
> in your blog. I’m very interested in the topic and fascinated by your
> team’s research.
>
> As I work to make sure I loaded the data correctly, I’m running into issues
> replicating the plot (d) in figure 3. I wasn’t able to capture as many
> deltas as you report in the paper. All my other counts and plots so far
> have matched yours.
>
> I’m able to count 2,269 discussion trees where one or more deltas were
> awarded by the OP; and I’m able to count 3,495 nodes awarded a delta by the
> OP. As a result, my percentages are smaller than what appears in Fig 3 (d)
> of your paper.
>
> Here’s how I’m identifying a delta post:
> if (string.find(comment, ’&amp;#8710’) != -1) and the author of the comment
> is the OP I consider that the comment contains a delta that I must attribute
> to the parent post.
>
> I’m wondering if I’m missing anything. How many deltas did you count at the
> discussion tree node level? And how about at the tree level?
>
> I would greatly appreciate any feedback.
>
> Thank you.
> Carlos Sathler