Detection of Myers- Briggs Type Indicator via Text Based Computer-Mediated Communication

Dan Brinks, Hugh White
Department of Electrical Engineering
Stanford University
Palo Alto, CA
{dbrinks, hwhite}@stanford.edu

***Abstract*— In this paper various algorithms for detecting Myers-Briggs Type Indicator based on brief samples of an individual's text communication from the web site Twitter are explored. These algorithms are trained using almost seven-hundred-thousand tweets from four-thousand-six-hundred unique users who have indicated their MBTI results.**

Keywords-MBTI; Myers-Briggs; Twitter; NLP; Naïve Bayes

I. INTRODUCTION

The Myers-Briggs Type Indicator (MBTI) is a psychometric assessment that describes preferences for recharging, understanding information, making decisions, and resolving issues. It is widely used to help people understand themselves, co-operate effectively with others, and present information persuasively. Personality assessment is also increasingly being used in marketing campaigns, having been used even in the 2012 elections [5]. The MBTI assessment uses a 93 question multiple choice test to determine personality type. Results of this assessment are confidential between the individual and the administrator. However, as MBTI makes predictions about how an individual communicates, it is possible for a trained analyst to examine samples of a subject's written communications and determine the subject's MBTI with a certain degree of accuracy. The aim of this project is to develop a computer system that can perform the function of the trained human analyst by predicting MBTI based on text communication.

II. DATA SET

While MBTI results are confidential, many individuals who have taken the assessment openly reveal their Myers-Briggs type in a variety of ways including Twitter. For instance, a Twitter search of "#INFP" finds tweets such as this: "I just reread the Meyers-Briggs description of my #INFP personality type. Its scary-accurate." While not all tweets tagged with a Myers-Briggs four letter personality type are authored by an individual of that personality type, it is a reasonable assumption that the majority of them are. A Python script utilizing the Twitter API was used to search for tweets including a Myers-Briggs Type (MBT) abbreviation. The username and MBT were recorded, resulting in a list of 6,358 users labeled with an MBT. Then, another Python script retrieved the last two-hundred tweets for each of the labeled users. Because Twitter throttles the number of requests per hour, these scripts were run automatically for five weeks. This produced a set of 960,715 tweets, each labeled by MBT.

Data Set Breakdown

Users By Type

Type	Count	Percentage
ENFJ	336	7.3%
ESFP	122	2.7%
INFJ	714	15.5%
ISFP	128	2.8%
ENTJ	279	6.1%
ESTJ	101	2.2%
INTJ	650	14.1%
ISTP	105	2.3%
ENTP	237	5.1%
ESTP	95	2.1%
INTP	423	9.2%
ESFJ	151	3.3%
ISFJ	181	3.9%
ENFP	448	9.7%
ISTJ	183	4.0%
INFP	449	9.8%
Total	4602	100.0%

Users Aggregated by Spectrum

Spectrum	Count	Percentage
E vs I	1769 2866	38% : 62%
N vs S	3536 : 1066	77% : 23%
T vs F	2073 : 2529	45% : 55%
P vs J	2007 : 2595	44% : 56%

Tweets Aggregated by Spectrum

Spectrum	Count	Percentage
E vs I	269014 : 426832	39% : 61%
N vs S	533725 : 162121	77% : 23%
T vs F	317991 : 377855	46% : 54%
P vs J	306234 : 389612	44% : 56%

TABLE I. DATA SET BREAKDOWN BY LABEL AND SPRECTRUM

III. PREPROCESSING

Initial inspection of the data revealed one potential issue. Many of the users labeled "INTP" were not referencing their MBT; instead, they had simply misspelled "into". By retaining only users who hashtagged an MBT or capitalized the entire MBT, 53%

of INTPs and 28% of all users were eliminated. Inspection of the data confirmed this filtering functioned as intended. In addition, any user whose labeling tweet contained two or more different MBTs was rejected. Next, all numbers, links, @<user>, and MBTs[1] were replaced with "NUMBER", "URL", "AT_USER", and "MBT". In addition, all contractions were replaced by their expanded form, and all words were converted to lowercase. Each tweet was tokenized using the Natural Language Toolkit (NLTK) Python library WordPunctTokenizer. Finally, all of a user's tweets were aggregated into a single text block.

IV. PROCESSING PARAMETERIZATION

Several further processing steps were implemented and selectively applied. This enabled the optimum processing parameters to be determined by running every combination and measuring the results. The optional processing steps were as follows:

1. Porter Stemming.
2. Emoticon Substitution. Emoticons were reduced to one of four categories: "SMILEY", "FROWN", "WINK", or "LAUGH".
3. Minimum Token Frequency. Any token occurring with a frequency less than the specified value was rejected.
4. Minimum User Frequency. Any token used by less than the specified fraction of user was rejected.
5. Term Frequency Transform (only relevant for MNEMNB). The frequency of terms by each user was transformed by a log(1 + f) factor (to decrease the effects of heavy tail which occurs in text) as per [1].
6. Inverse Document Frequency Transform (only relevant for MNEMNB). The weight of terms was discounted by their document frequency as per [1].

V. ALGORITHM DEVELOPMENT

The initial algorithm employed the Naïve Bayes Classifier provided by the NLTK, which uses the multi-variate Bernoulli event model. It was discovered that this classifier yielded poor test results as classification accuracy was worse than the prior distribution. Even the training results were only on par with the prior expectation. Analysis showed that the classifier was almost universally making its decisions based on the number of words retained for each user. For instance, one test showed that the average number of tokens for a user chosen to be an extrovert was forty-three, while the average number of tokens for chosen introverts was over three-hundred. This shortcoming is a result of the calculations performed by the multi-variate Bernoulli event model: that providing more features for a label to the algorithm increases the probability of that label being selected.

To combat the issues inherent in the multi-variate event model, a multinomial event model classifier was written since NLTK provides no such functionality. This classifier avoided the selection dichotomy based on number of tokens and therefore improved performance to a level roughly on par with the prior distributions. However, the training accuracy had improved to a level well above the priors which indicates that the classifier was functioning nominally. In order to improve performance, the variance between training and testing was addressed.

Training Accuracy by Classifier

Classifier	E vs I	N vs S	T vs F	P vs J
Multinomial Event Model Naive Bayes	96.0%	83.4%	84.6%	75.9%
L2-regularized logistic regression (primal)	99.8%	99.8%	100.0%	99.8%
L2-regularized L2-loss SV classification (dual)	99.8%	99.9%	99.9%	99.9%
L2-regularized L2-loss SV classification (primal)	99.8%	99.9%	99.9%	99.9%
L2-regularized L1-loss SV classification (dual)	99.9%	99.9%	99.9%	99.9%
SV classification by Crammer and Singer	100.0%	100.0%	100.0%	100.0%
L1-regularized L2-loss SV classification	100.0%	100.0%	100.0%	100.0%
L1-regularized logistic regression	99.9%	99.9%	99.8%	99.9%
L2-regularized logistic regression (dual)	100.0%	100.0%	100.0%	100.0%

TABLE II. TRAINING ACCURACY BY CLASSIFER

[1] Leaving the MBTs unmodified caused the classifiers to perform significantly better, but the authors felt this enabled the classifiers to "cheat", and thus should be disallowed.

The first solution to a high variance problem is more data. Unfortunately, Twitter places a cap on data retrieval requests, and only a limited number of users tweet their MBTI information. This meant that the only way to acquire more data was to give the Twitter scraping scripts more time to do their work. However, even after tripling the number of collected tweets, performance remained constant.

Another solution to high variance is decreasing the feature set size. By modifying the preprocessing steps, a parameterized number of features could be fed to the classifier in order to determine the optimal number of features. Additionally, several transforms detailed in [1] were added to the classifier in hopes of improving the performance. Furthermore, the algorithm was modified to use confidence metrics in its classification and instructed to only make a determination for users about which it had a strong degree of certainty. This yielded significant improvements to training accuracy (sometimes above 99%!). Ultimately, however, none of these options improved testing behavior to any significant degree.

To verify the accuracy of the multinomial event model classifier, SVM and logistic regression classifiers provided by liblinear[3] were employed. These classifiers suffered similar issues to Naïve Bayes—relatively high training accuracy but low test accuracy across a broad range of tokenizing and feature set size parameterization.

VI. Results

Once processed, the data was fed into multiple classifiers. The performance of each classifier was measured using 70/30 hold-out cross-validation. A full list of classifiers and performance metrics are shown in Table III.

VII. Analysis

There are several possible reasons why the machine classifier did not achieve better performance. One explanation is that a large portion of tweets are noise with respect to MBTI. This is the result of several factors. Perhaps the most difficult to overcome is the inherent breviloquence of tweets. Because Twitter imposes an 140 character limit on each tweet, users are forced to express themselves succinctly, which causes stylistic elements common in prose to be suppressed as each thought is compressed down to its raw elements. This means that the number of indicator words were few and infrequent. In addition, a large percentage of tokens in tweets are not English words, but twitter handles being retweeted or URLs. Thus, while a user's tweet set may contain a thousand tokens, a significant subset is unique to that individual user, and cannot be used for correlation. Further, due to retweeting, a user's tweet may not be expressing his or her own thoughts, but those of a different individual.

A second explanation is that while tweets may have MBTI relevant information buried in them, this information is not accessible through simple word frequency. As an example, consider the following tweet: "Everyone is like *I hate Obama* or *I hate Romney* and I'm over here like *I love pizza*". The user, as a perceiver, is contrasting the strong judgments of others with her own preference for food. However, word frequency analysis shows "hate hate love", strong words signaling towards a judger. Thus, a decision based on word frequency alone would mislabel this tweet. It is possible that advanced Natural Language Processing (NLP) techniques that isolate clauses, identify sentence structure, and recognize negators could result in useful features, but such techniques were outside the scope of this project.

Performance by Classifier

Classifier	E vs I	N vs S	T vs F	P vs J
Multinomial Event Model Naive Bayes	63.9%	74.6%	60.8%	58.5%
L2-regularized logistic regression (primal)	60.3%	70.7%	59.4%	56.1%
L2-regularized L2-loss SV classification (dual)	56.9%	67.5%	59.3%	54.1%
L2-regularized L2-loss SV classification (primal)	58.8%	69.5%	59.0%	55.9%
L2-regularized L1-loss SV classification (dual)	56.8%	67.6%	59.6%	54.5%
SV classification by Crammer and Singer	56.8%	67.7%	59.4%	54.5%
L1-regularized L2-loss SV classification	59.4%	68.3%	56.8%	56.1%
L1-regularized logistic regression	60.9%	70.5%	58.5%	56.3%
L2-regularized logistic regression (dual)	59.2%	69.6%	59.0%	55.0%

TABLE III. Training Accuracy by Classifer

VIII. COMPARISON WITH HUMAN EXPERTS

Twenty users were randomly selected from our dataset. For each of these users, thirty of their tweets were randomly chosen and presented to two human experts trained in MBTI administration and analysis. These same tweets were also supplied to the Multinomial Event Model Naïve Bayes classifier. The results of both the human and machine selections are shown in table IV.

Human #1 averaged 60%, Human #2 averaged 61.25%, and the MNEMNB Classifier averaged 66.25%. While the machine classifier did outperform the humans, the performance difference is small enough that this result is probably not significant, especially considering the small sample size. The human experts found some users in the test set to be unclassifiable and were forced to make a random guess. On other tweets, the experts indicated they had a low confidence in their predictions for a certain spectrum. While in general, the experts' confidence in their predictions corresponded with their results, there were users that the experts were shocked to learn they had mislabeled. This supports the idea that while some users have tweets that contain personality indicators, others do not, or worse, contain misinformation.

Comparison of Human Experts Vs Machine			
Spectrum	**Human 1**	**Human 2**	**MNEMNB**
E vs I	50.0%	40.0%	55.0%
N vs S	50.0%	90.0%	90.0%
T vs F	80.0%	65.0%	55.0%
P vs J	60.0%	50.0%	65.0%

TABLE IV. COMPARISON OF HUMAN EXPERTS VS MACHINE

When successful, the human experts tended to draw upon context information in the tweets. For example, in the tweet "Got to rush and crowd on public transport... Life sucks", the expert concluded the user was an introvert since they were referencing "crowd" and "public" in the context of "Life sucks". This indicates that by using only word frequencies, the MNEMNB classifier is rejecting potentially useful information.

IX. CONCLUSION

In this project nearly one million tweets from six-thousand users were retrieved for the purpose of Myers-Briggs Type detection. A Multinomial Event Model Naive Bayes classifier was developed, trained, and tested on this data. The performance of this classifier on training data was quite good, but the classifier failed to achieve excellent results for test data. A variety of preprocessing techniques were parameterized and attempted in all combinations without noticeable effect. While this appears to be a high-variance problem, adding more data and limiting the number of features did not improve performance. Fixes for known weaknesses in Naïve Bayes classifiers were also attempted, but again performance remained roughly constant. To ensure that the performance of the Naïve Bayes classifier was not due to implementation error, the training and test data was fed into several classifiers provided by liblinear with similar results. Further analysis of these results lead the authors to conclude that while there may be MBTI information in tweets, it is not prevalent and may require advanced natural language processing techniques to uncover.

X. ACKNOWLEGDEMENTS

Special thanks to David Patterson for providing insight regarding MBTI and tweet data and providing suggestions for improving the preprocessing. Thanks to both David Patterson and Lloyd Patterson for serving as the human experts and analyzing 20 pages of raw tweets.

XI. REFERENCES

[1] J. Rennie, L Shih, J. Teevan, and D. Karger, "Tackling the poor assumptions of naïve bayes text classifiers," Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[2] J. Perkins, Python Text Processing with NLTK2.0 Cookbook. Olton, Birmingham, UK: Packt Publishing, 2010

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9(2008), 1871-1874. Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear

[4] M. Russel, Mining the Social Web. Sebastopol, CA: O'Reilly Media, 2012

[5] C. Parsons and M. Memoli. "President Obama tailors outreach for select group." The Los Angeles Times. 8 August 2012. Web. 12 December 2012. http://articles.latimes.com/2012/aug/08/nation/la-na-obama-narrowcasting-20120808.