Text-Based Prediction of MBTI Personality Types

Project Category: Natural Language Processing
Project Mentor: Michael Xie

Jenny Mai
SUNet ID: mcmai
Department of Computer Science
Stanford University
mcmai@stanford.edu

Emily Nguyen
SUNet ID: eqnguyen
Department of Computer Science
Stanford University
eqnguyen@stanford.edu

Daniel Kim
SUNet ID: ddk
Department of Computer Science
Stanford University
ddk@stanford.edu

1 Key Information to include

- External collaborators or mentors (if you have any): N/A
- Sharing project with another class: No

2 Motivation

Although the concept of personality tests has been around for over 100 years, all of these personality tests follow the same structure: classify the user's personality based on their answers to a fixed set of questions. However, we would like to explore how personality types, more specifically the Myers-Briggs Type Indicator (MBTI), can be indirectly inferred from a user's text. While doing background research on this topic, we came across several projects that classified the user's personality based on their text answers to open-ended questions. Our project aims to predict the user's MBTI personality type based on their tweets. This application has the potential to uncover other personality types indirectly, through whatever people say on the internet (rather than just Twitter).

3 Methodology

3.1 Data collection

We used the (MBTI) Myers-Briggs Personality Type Dataset from Kaggle[1], which includes over 8600 rows of Twitter users' last 50 Twitter posts and their respective MBTI labels. A snippet of the data is shown in A

3.2 Analysis and Preprocessing

Before preprocessing the data, we first explored the general distribution of MBTI personality types in tweets, plotted in B. There is a very uneven distribution in MBTI types, with heavy skew towards introverted types.

[1] https://www.kaggle.com/datasets/datasnaek/mbti-type

Next, to preprocess the data, we extracted each tweet from each user and removed all punctuation, symbols, and hyperlinks, as well as make all words lowercase. This made sure that our classification was accurate based on the actual words within the tweets and make sure that words that have different spellings are classified together. For example, the contractions "don't", "dont", and "Don't" would all be tokenized as "dont".

After cleaning our data, we looked at the most popular words for each of the types C. Interestingly, some of the popular words that each of the types post are the MBTI types themselves.

In addition, we lemmatized our data using NLTK's WordNetLemmatizer and removed all stop words. For example, if we have the word "feet", this would be lemmatized into the word "foot". This process also ensures that words with the same base meanings are classified together to increase the accuracy of the classification.

Finally, we applied TF-IDF weighting on each of the user's posts. This measures how relevant each lemma/string is in each post for each user. However, because there are over $200,000$ words over all posts, we decided to use the top $10,000$ words ordered by their term frequency. This feature extraction ensures the efficiency of our algorithms while also maintaining the most important features in our data.

4 Preliminary experiments

After pre-processing the data, we randomly split the data into training and testing sets, with a ratio of 80:20. We then fit three different classification models on the training set and evaluated their accuracy with the testing set:

Classification Accuracy

Model	Accuracy
Multinomial naive Bayes	0.1100
Multiclass logistic regression	0.1827
Linear SVC	0.2028

4.1 Multinomial naive Bayes

We implemented the multinomial naive Bayes algorithmD as a baseline and resulted in an accuracy of 0.1100. Interestingly, the algorithm predicted many personalities to be ENFP and ENTP (which was false most of the time), even though ENFP and ENTP had relatively low popularity in the dataset. Multinomial naive Bayes does not work too well on this dataset, likely because the conditional independence assumption is incorrect: every pair of words given the personality type is not independent.

4.2 Multiclass logistic regression

Next, we implemented a multiclass logistic regression modelE that used a one-vs-rest scheme and resulted in an accuracy of 0.1827. Unlike multinomial naive bayes, multiclass logistic regression mainly predicted INTJ personality types for the testing set. This makes more sense, as INTJ is the fourth most popular personality type. This model was likely more successful than the previous one because multiclass logistic regression uses less assumptions, and the size of the dataset is sufficiently large.

4.3 Linear support vector classifcation matrix

Next, we implemented a linear SVC alorithmF with a one-vs-rest scheme and resulted in an accuracy of 0.2028. Like multiclass logistic regression, linear SVC also predicted many personality types to be INTJ, however there were also many ENTP predictions. Rather than maximizing the conditional likelihood of the training data in logistic regression, linear SVC is a more geometric approach that

maximizes the margin among class variables, thus it's less prone to outliers. As a result, linear SVC works better with semi-structured data (i.e. the text we are dealing with).

5 Next Steps

Given your preliminary results, what are the next steps that you're considering?

1. There are certain personality types are underrepresented in the data set: ESTJ, ETFJ, ESFP, etc. To address these imbalances, we might use SMOTE (Synthetic Minority Over-sampling Technique), a data augmentation technique that generates synthetic examples from existing examples from minority classes. We will experiment with the degree to which we try to balance the data set. This includes determining the number of classes we want to balance and the level we want to balance them to increase accuracy.
2. Since there are 16 classes we are predicting, we will experiment with breaking down the MBTI labels in our predictions. This means classifying data as 'I' or 'E', 'N' or 'S', 'F' or 'T', and 'P' or 'J'. This way, we will only have to predict among two classes at a time, and we also have more data for each class—the distribution will be more equal, hopefully resulting in a better model.
3. We will further improve the pre-processing of the data with regards to the stop words and regular expressions used to clean the tweets to increase the accuracy. We will also experiment with feature selection.
4. We also plan to experiment with different models that are specifically used in multi-class classification tasks. In particular, XGBoost and Random Forest are techniques are more specialized techniques typically used for text classification that we plan to experiment with. We may also use bigrams when constructing the Naive Bayes Model to address our incorrect assumption that all words are conditionally independent.
5. We will further analyze the current models that we generated and their performance. This includes metrics such as precision-recall and f1-scores.

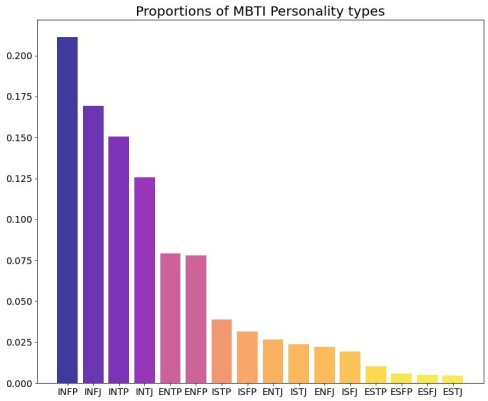
6 Contributions

Jenny: Data exploration/analysis, preprocessing
Daniel: Date exploration/Next Steps
Emily: Preliminary experiments and graphics

A Raw data of MBTI and tweets

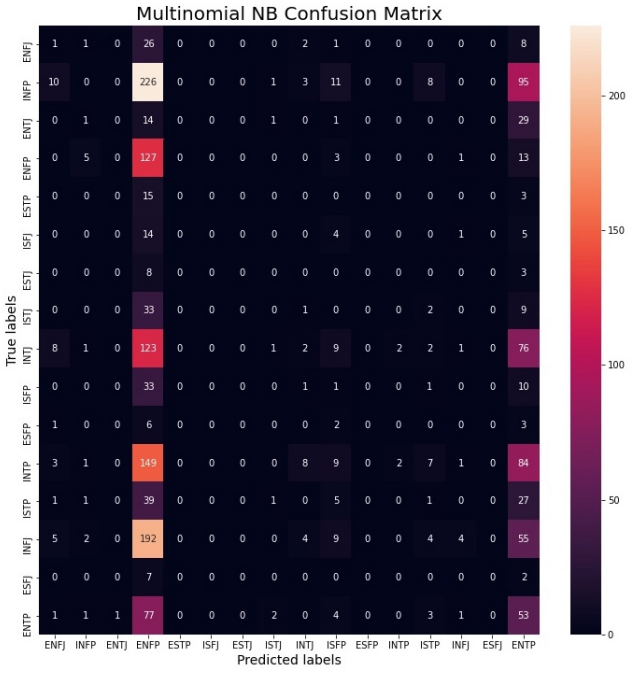
	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...

B Distribution of MBTI personalities

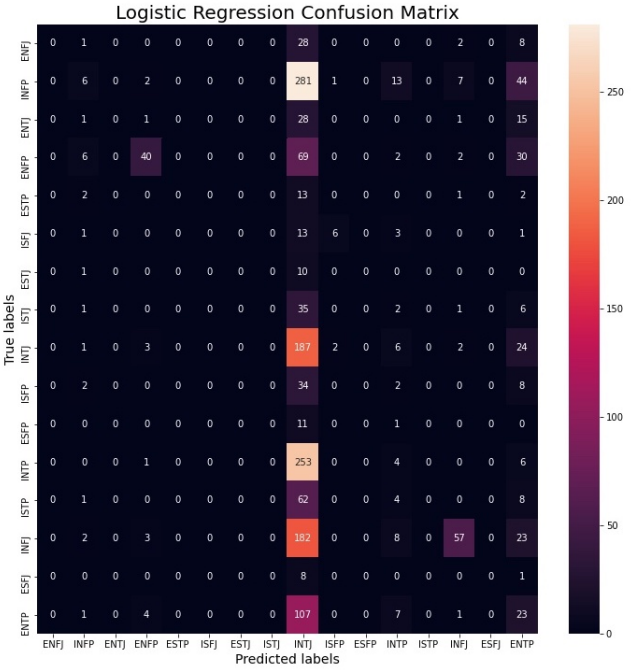


C Most popular words for each personality type

D Multinomial naive bayes confusion matrix



E Multiclass logistic regression confusion matrix



F Linear support vector classification matrix