EasyChair Preprint

№ 2511

Machine Learning Powered Compatible People Proposer Based on Personality Traits

Gaurav Goswami, Divyanshu Gaur, Eshani Agarwal,
Enosh Kumar and Mukesh Rawat

January 31, 2020

Machine Learning powered Compatible People proposer based on personality traits

Gaurav Goswami
Computer Science and Engineering
Meerut Institute of Engineering and Technology
Meerut, India
gaurav.goswami.cs.2016@miet.ac.in

Divyanshu Gaur
Computer Science and Engineering
Meerut Institute of Engineering and Technology
Meerut, India
divyanshu.gaur.cs.2016@miet.ac.in

Eshani Agarwal
Computer Science and Engineering
Meerut Institute of Engineering and Technology
Meerut, India
eshani.agarwal.cs.2016@miet.ac.in

Enosh Kumar
Computer Science and Engineering
Meerut Institute of Engineering and Technology
Meerut, India
enosh.kumar.cs.2016@miet.ac.in

Dr. Mukesh Rawat
Computer Science and Engineering
Meerut Institute of Engineering and Technology
Meerut, India
mukesh.rawat@miet.ac.in

***ABSTRACT* – The aim is to suggest groups of various people that are compatible working with each other and have similar thinking by analyzing their personality traits obtained or inferred by their speeches, social media updates, views, etc. The textual data (containing non-ASCII characters too) is firstly filtered and then fed to a machine learning model which will spit out 27 different numeric values corresponding to different personality traits. The obtained dataset is then dimensionally reduced for better usage ahead. Then a visual 3D graph is plotted where compatible people groups are displayed as clusters.**
Finally, we can make groups as the clusters in graph which could surely lead to better throughput in the tasks performed by groups.

***Abbreviations used* –** NLU: Natural Language Understanding, PCA: Principal Component Analysis, API: Application Program Interface, 3D: 3-Dimensional, DB: Database

1. Introduction

It is the age of social media liberalism; everyone is free to express what he/she feels, they can say whatever they want to. But the question arises now is: What to do about people working together for same task having different thinking? It leads to poor communication, different working behavior, and different views, choices, likes and dislikes among team members and as a result, team heads to wrong direction. So, in order to avoid all this, we need to make teams (or groups) of people having similar thinking, views and behaviors, choices, likes and dislikes. However, it is not possible to find people with identical thinking, but we could go for similar thinking.
It is obvious and easily understood that people with common thinking and personality traits tend to work efficiently with each other, which could reduce work fatigue and pressure they're carrying in current work culture [7].

2. Application Structure

2.1.Overall Architecture

The application is divided into 4 tiers, frontend, backend, database and API tiers. Frontend just comprises of user interfaces for various operations, whereas backend comprises of algorithms of various operations, such as Authentication, Data Fetch, Data Cleaning, Analysis, Display Card, Generate 3D Graph, Report Generation etc.

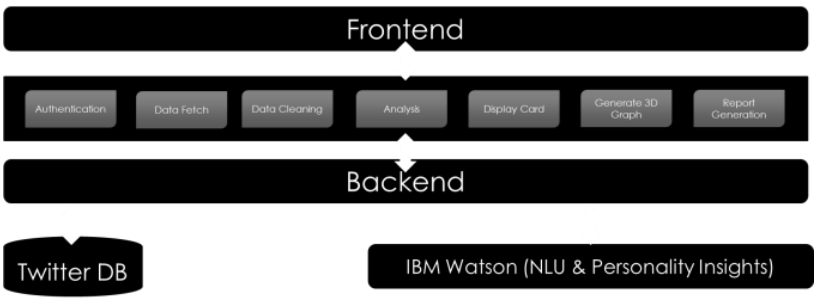


Fig.1: 4-tier architecture of application

2.2.DFD Level-1

The diagram depicts the data and control flow in the application from one point to another point.

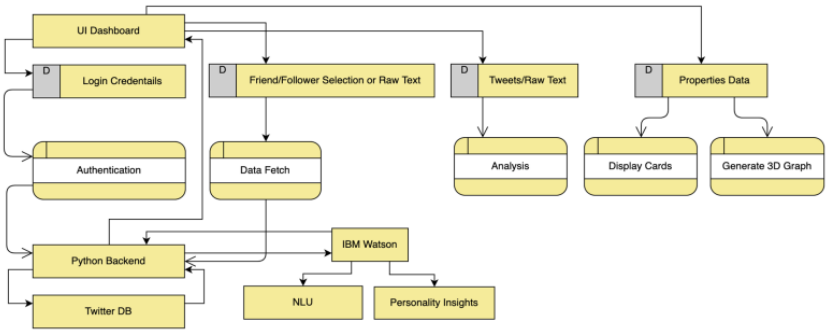


Fig.2: Control and data flow in application

The application runs as follows: User logs in using his/her Twitter credentials and then user may enter any raw text manually or select any friend/follower or own self and backend will fetch all the recent updates to the selected person[1] and then clean the text data and forward it to IBM Watson NLU and Personality Insights services, and after getting response, backend filter out specific 27 personality traits with their corresponding scores. The backend generates cards for each selected person with those personality traits and then the dataset is reduced to 3-dimensions using PCA algorithm and an overall 3D graph is generated which depicts clusters of compatible people (having =< 0.7 units of Euclidean distance). The backend finally generates a report showing what all groups or clusters can be made (if any).

3. Machine Learning Models

We've used two machine learning models which are capable of analyzing textual data and to predict how much they fall in defined categories. These models have variable accuracy, which is dependent on words count of textual data fed to it.

Textual data is cleaned first before sending to NLU Model and Personality Insights Model.

3.1.NLU Model

Analyze text to extract metadata from content such as concepts, entities, keywords, categories, sentiment, emotion, relations, and semantic roles using natural language understanding [3][5].

NLU Model provides these traits:

- Joy
- Anger
- Disgust
- Sadness
- Fear

3.2.Personality Insights Model

Predict personality characteristics, needs, and values via written text. Understand customer habits and preferences on an individual level—at scale [2].

Personality Insights Model provides these traits:

- Openness
- Conscientiousness
- Extraversion
- Agreeableness
- Emotional range
- Challenge
- Closeness
- Curiosity
- Excitement
- Harmony
- Ideal
- Liberty
- Love
- Practicality
- Self-Expression
- Stability
- Structure
- Conservation
- Openness to change
- Hedonism
- Self-Enhancement
- Self-Transcendence

4. Custom Algorithms

4.1. Data Preprocessing Algorithm [8]

- Remove punctuation from words (e.g. ‘what’s’).
- Removing tokens that are just punctuation (e.g. ‘-‘).
- Removing tokens that contain numbers (e.g. ’10/10’).
- Remove tokens that have one character (e.g. ‘a’).
- Remove tokens that don’t have much meaning (e.g. ‘and’).
- Remove any non-ASCII characters (e.g. ‘☺’).
- Encode whole text content to UTF-8, for simplicity.

For better understanding, let’s clean the following textual data:

“It’s all about one’s hard work. Luck is not anything.”

After removing punctuation from words, we get:

“It all about one hard work. Luck is not anything.”

After removing punctuation marks, we get:

“*It all about one hard work Luck is not anything*”

The data doesn’t have any numeric, non-ASCII, meaningless tokens, and single characters. So, we can say the data is cleaned now.

Final cleaned data is:

“It~~’s~~ all about one~~’s~~ hard work~~.~~ Luck is not anything~~.~~.”

4.2. PCA Algorithm

- Find the mean vector.
- Assemble all the data samples in a mean adjusted matrix.
- Create the covariance matrix.
- Compute the Eigen vectors and Eigen values.
- Compute the basis vectors.
- Represent each sample as a linear combination of basis vectors.

For better understanding, let’s reduce the following data using PCA to 3 dimensional data [4]:

#	A	B	C	D
0	2	1	7	7
1	1	2	3	3
2	4	5	7	2
3	3	3	2	4

Mean vector (= $[(1/n)*\Sigma X_i]$):
[2.5, 2.75, 4.75, 4]

Centered column dataset (= $[X_i - \mu]$)

#	A	B	C	D
0	-0.5	-1.75	2.25	3
1	-1.5	-0.75	-1.75	-1
2	1.5	2.25	2.25	-2
3	0.5	0.25	-2.75	0

Covariance Matrix (= $[(1/n-1)*\Sigma(X_i - \mu_x)*(Y_i - \mu_y)]$)

#	A	B	C	D
0	1.66666667	1.83333333	1.16666667	-1
1	1.83333333	2.91666667	0.58333333	-3
2	1.16666667	0.58333333	6.91666667	1.33333333
3	-1	-3	1.33333333	4.66666667

Eigen Decomposition of Covariance Matrix:

Eigen Vectors:

#	A	B	C	D
0	0.56833128	-0.72801744	-0.37173503	-0.09382559
1	-0.73762144	-0.27508833	-0.49677336	-0.36531801
2	0.03627646	0.30097755	-0.69778696	0.64899144
3	-0.36276464	-0.55111663	0.35794225	0.66071902

Eigen Values:

[-1.11022302e-15, 1.12007468e+00, 7.26952363e+00, 7.77706835e+00]

Here, we see last two columns have high rank of all others, so taking those two columns only.

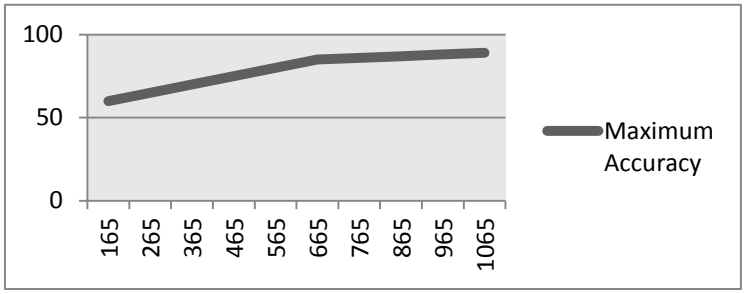
Projected Dataset

#	~~A~~	~~B~~	C	D
0	~~-4.44089210e-16~~	~~-1.30737111e-01~~	5.59026969e-01	4.12860711e+00
1	~~-2.77555756e-16~~	~~1.32274832e+00~~	1.79336750e+00	-1.38172714e+00
2	~~-2.22044605e-16~~	~~6.84578439e-02~~	-3.96124776e+00	-8.23911213e-01
3	~~9.02056208e-16~~	~~-1.26046906e+00~~	1.60885330e+00	-1.92296875e+00

5. Accuracy Experiments

IBM Watson Personality Insights and NLU were experimented with text data of different lengths, and it was found that accuracy differs among them as follows:

Word Count	Accuracy (approximate)
167	Around 60% - 65%
270	Around 66% - 68%
331	Around 69% - 74%
489	Around 75% - 83%
562	Up to 84%
892	Up to 87%



Above experiment clearly shows that up to a maximum extent (above which accuracy could not be increased further) is directly proportional to the word count. Mathematically,

$$\mathbf{A \propto n}$$
(up to an extent)

A = accuracy of model
n = word count

CONCLUSION

Hence, it can be concluded by above work that we can form groups of compatible people by analyzing their speech, views, social media updates etc. and comparing their personality traits with those of others which could prove to be best and most efficient groups. And also, accuracy of the groups' formation can be increased by increasing the text content.

REFERENCES:

[1] 'Tweepy', http://docs.tweepy.org/en/v3.5.0/getting_started.html
[2] 'IBM Watson Personality Insights', https://cloud.ibm.com/docs/services/personality-insights/getting-started.html
[3] 'IBM Watson NLU', https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-getting-started#getting-started
[4] 'PCA', https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial
[5] B. Liu, "Handbook Chapter: Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing," Handbook of Natural Language Processing. Marcel Dekker, Inc. New York, NY, USA, 2009
[6] V. Garousi and M. V. Mäntylä, "Citations, research topics and active countries in software engineering: A bibliometrics study," Computer Science Review, vol. 19, pp. 56–77, 2016.
[7] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and trends in information retrieval, vol.2, no. 1–2, pp. 1–135, 2008.
[8] 'Data Preprocessing', https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html