Feature Extraction from Social Media Posts for Psychometric Typing of Participants

Charles Li[1], Monte Hancock[2](✉), Ben Bowles[3], Olivia Hancock[3], Lesley Perg[3], Payton Brown[3], Asher Burrell[3], Gianella Frank[3], Frankie Stiers[3], Shana Marshall[3], Gale Mercado[3], Alexis-Walid Ahmed[3], Phillip Beckelheimer[3], Samuel Williamson[3], and Rodney Wade[3]

[1] Department of Mathematics and Computer Science,
Mercy College, Dobbs Ferry, USA
[2] George Herbert Walker School of Business,
Webster University, Webster Groves, USA
practicaldatamining@gmail.com
[3] Sirius18, New York, USA

Abstract. Sentiment analysis is an important tool for assessing the dynamic emotional terrain of social media interactions and behaviors [1]. Underlying the shallow emotional phenomenology are deeper and more stable strata, such as culture and psychology.

This work addresses the latter, by applying text mining methods to the assessment of individual psychometrics. A methodology is described for reducing bulk, unstructured text to low-dimensional numeric feature vectors, from which components of the Myers-Briggs Typology Indicator (MBTI) [2] of the text's author can be reliably inferred.

MBTI is a psychometric schema that emerged from the personality theories of Freud and Jung in the early 20th Century, refined and codified by K. C. Briggs and her daughter, I. Briggs-Myers in the 1940's and 50's. This schema positions people along four (nominally independent) axes between pairs of polar motivations/preferences: Extroversion vs. Introversion (E-I); Intuition vs. Sensing (N-S); Feeling vs. Thinking (F-T); and, Judging vs Perceiving (J-P). Under this schema, each person falls into one of 16 psychometric groups, each designated by a four-character string (e.g., INTJ) [3].

Empirical results are shown for text generated during the social media interaction of over 8,600 PersonalityCafe users [4], all of whom are of known MBTI type. Blind tests to validate the features were conducted for a population (balanced by MBTI type), with exemplars based upon text samples having several thousand words each. The feature extraction method presented supports partial (1-letter) MBTI psychometric typing: E-I 95%; J-P 76.25%; F-T 91.25%, N-S 90%. Other results are reported.

Keywords: Social media · Psychometrics · MBTI

D. D. Schmorrow and C. M. Fidopiastis (Eds.): AC 2018, LNAI 10915, pp. 267–286, 2018.
https://doi.org/10.1007/978-3-319-91470-1_23

1 Introduction

1.1 Moving from Voice Back to Text: A World-Wide Cultural Shift

Human collaborative processes are being revolutionized by the emergence of ubiquitous, completely portable social media. Group decision-making, social interaction, educational instruction, and many other directed and undirected cognitive interactions are now conducted without a single spoken word being exchanged.

As a result of this cultural shift, much social, business, and government communication has moved from postal and telephone exchanges to conversation online. This has fostered the development of world-wide open-source social platforms (e.g., Twitter, Facebook, internet Forums) which daily host billions of text-based interactions. Research is ongoing into questions such as:

- What use can be made of this open-source data?
- What can social media text analysis tell us about its author?
- What the limiting factors for social media text analysis (e.g., how much/what type of data are required? Are results repeatable?)

1.2 Popular Interest in Psychometrics Is Growing

Text mining today generally relies on a combination of statistical and graph theoretic schemas for representing latent information. These schemas are parsed and quantified to obtain information about associations, processes, and conditional probabilities for variables of interest. While some automation exists, the final processing and interpretation is largely manual and ad hoc. The state of the art is described in detail in [6].

The Myers-Briggs Typology Indicator (MBTI) is a psychometric schema that emerged from the personality theories of Freud and Jung in the early 20th Century, refined and codified by K. C. Briggs and her daughter, I. Briggs-Myers in the 1940's and 50's. This schema positions people along four (nominally independent) axes between pairs of polar motivations/preferences: Extroversion vs. Introversion (E-I); Intuition vs. Sensing (N-S); Feeling vs. Thinking (F-T); and, Judging vs Perceiving (J-P). Under this schema, each person falls into one of 16 psychometric groups, each designated by a four character string (e.g., INTJ).

MBTI has become a simple and popular schema for self-analysis among the public. Professional determination of MBTI typology is available; but, informal determination of MBTI typology can be accomplished by taking one (or more) of a large number of free, online, machine-scored multiple-choice tests. A typical example consists of 70 binary-choice check-boxes, and can be completed in less than 10 min. As such, the MBTI is generally a "self-reported" assessment.

Modern applications of MBTI include staffing placement for new employees, corporate team building exercises, student assessment for educational planning, foreign intelligence collection, psychological operations in counter-terrorism, and others.

2 Social Media Data

A Social medium is defined here as any venue supporting public-access pseudo-anonymous self-initiated asynchronous text/image sharing. The term platform will be used here to refer to web-based systems that exist to provide public venues for social interaction and informal information exchange. Preeminent examples include Facebook, Twitter, Reddit, Imgur, SnapChat, and many others.

Social media (the plural) have the following attributes:

1. User submissions are referred to variously as:
   (a) "posts" when the venue is a Forum, Blog (Weblog), or a Chat
   (b) "messages" when the venue is a messaging system ("tweets" on Twitter)
2. Submissions might or might not be moderated by venue managers, according to guidelines established by the venue owners, and local, state and federal law.
3. Submissions are often organized into threads by topic and/or conversation. A thread is a collection of posts grouped by time, topic, or type.
4. Access to a venue generally requires some kind of "membership" or "registration".
5. Media often provide various levels of privacy among users ("following, blocking", etc.).

3 Data Source

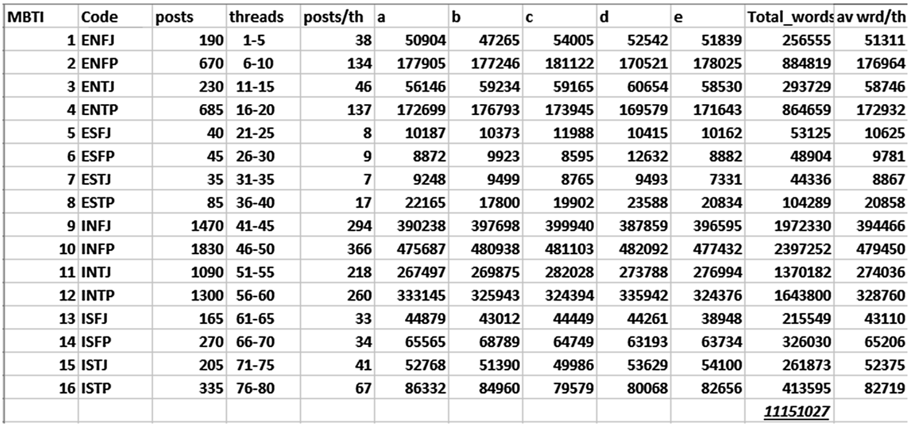
The data for this paper were collected through the PersonalityCafe forum in 2017. It consists of snippets (usually whole sentences) of the last 50 posts made by 8,675 people, all of known MBTI type. The entire 59 MB corpus was posted for download by researchers on a web site for Data Science researchers [7].

The average length of the post samples is 1,288 words. All posts are in English. The total size of the corpus is approximately 11.2 million words. Each post sample is tagged with the author's MBTI type. See Table 2.

The number of words vary from one author to another. Further, the number of posts for the different MBTI types are different. Of special concern is the corpus' word count imbalance between some of the MBTI types. The "ES" MBTI types have about 20% or fewer of the words of the other types.

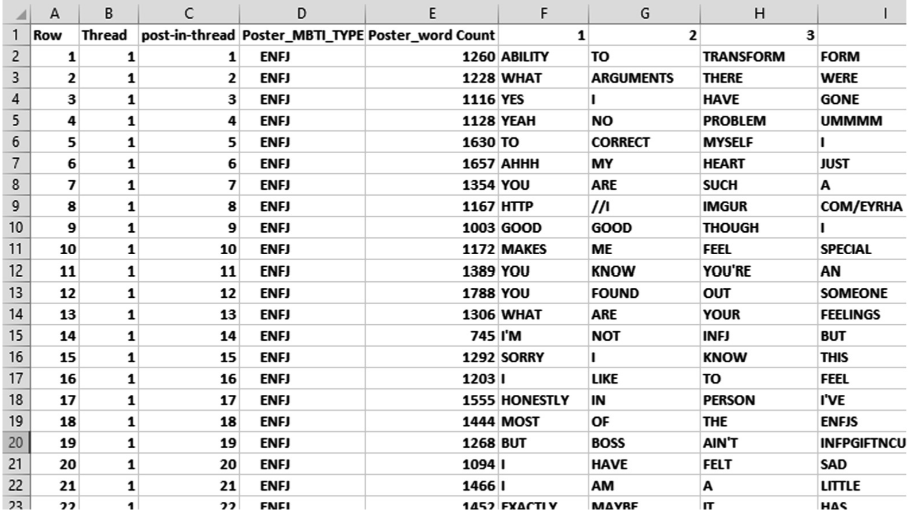
Rebalancing the data set by decimating the large classes would have resulted in most of the data being lost. Rather than do this, it was decided to merge the text of each MBTI type to produce 5 exemplars of that type. With 16 MBTI types, this produced 80 MBTI exemplars, the smallest of which would consist of 7,331 words. The table below gives the specifics for each MBTI type. For example (see row 1 of Table 1 below), the corpus contains 190 ENFJ posters. Merging their text in groups of 38 posters produces 5 ENFJ exemplars (henceforth called threads), averaging 51,311 words each. In this way, the 8,645 posters are aggregated into 80 threads, homogeneous by type, each thread consisting of at least 7,331 words, and as many as 399,940 words.

Table 1. Descriptive Statistics for our Corpus for each MBTI type.



MBTI	Code	posts	threads	posts/th	a	b	c	d	e	Total_words	av wrd/th
1	ENFJ	190	1-5	38	50904	47265	54005	52542	51839	256555	51311
2	ENFP	670	6-10	134	177905	177246	181122	170521	178025	884819	176964
3	ENTJ	230	11-15	46	56146	59234	59165	60654	58530	293729	58746
4	ENTP	685	16-20	137	172699	176793	173945	169579	171643	864659	172932
5	ESFJ	40	21-25	8	10187	10373	11988	10415	10162	53125	10625
6	ESFP	45	26-30	9	8872	9923	8595	12632	8882	48904	9781
7	ESTJ	35	31-35	7	9248	9499	8765	9493	7331	44336	8867
8	ESTP	85	36-40	17	22165	17800	19902	23588	20834	104289	20858
9	INFJ	1470	41-45	294	390238	397698	399940	387859	396595	1972330	394466
10	INFP	1830	46-50	366	475687	480938	481103	482092	477432	2397252	479450
11	INTJ	1090	51-55	218	267497	269875	282028	273788	276994	1370182	274036
12	INTP	1300	56-60	260	333145	325943	324394	335942	324376	1643800	328760
13	ISFJ	165	61-65	33	44879	43012	44449	44261	38948	215549	43110
14	ISFP	270	66-70	34	65565	68789	64749	63193	63734	326030	65206
15	ISTJ	205	71-75	41	52768	51390	49986	53629	54100	261873	52375
16	ISTP	335	76-80	67	86332	84960	79579	80068	82656	413595	82719
										11151027	

Table 2. Each post sample is tagged with the author’s MBTI type



	A	B	C	D	E	F	G	H	I
1	Row	Thread	post-in-thread	Poster_MBTI_TYPE	Poster_word Count	1	2	3	
2	1	1	1	ENFJ	1260	ABILITY	TO	TRANSFORM	FORM
3	2	1	2	ENFJ	1228	WHAT	ARGUMENTS	THERE	WERE
4	3	1	3	ENFJ	1116	YES	I	HAVE	GONE
5	4	1	4	ENFJ	1128	YEAH	NO	PROBLEM	UMMMM
6	5	1	5	ENFJ	1630	TO	CORRECT	MYSELF	I
7	6	1	6	ENFJ	1657	AHHH	MY	HEART	JUST
8	7	1	7	ENFJ	1354	YOU	ARE	SUCH	A
9	8	1	8	ENFJ	1167	HTTP	//I	IMGUR	COM/EYRHA
10	9	1	9	ENFJ	1003	GOOD	GOOD	THOUGH	I
11	10	1	10	ENFJ	1172	MAKES	ME	FEEL	SPECIAL
12	11	1	11	ENFJ	1389	YOU	KNOW	YOU'RE	AN
13	12	1	12	ENFJ	1788	YOU	FOUND	OUT	SOMEONE
14	13	1	13	ENFJ	1306	WHAT	ARE	YOUR	FEELINGS
15	14	1	14	ENFJ	745	I'M	NOT	INFJ	BUT
16	15	1	15	ENFJ	1292	SORRY	I	KNOW	THIS
17	16	1	16	ENFJ	1203	I	LIKE	TO	FEEL
18	17	1	17	ENFJ	1555	HONESTLY	IN	PERSON	I'VE
19	18	1	18	ENFJ	1444	MOST	OF	THE	ENFJS
20	19	1	19	ENFJ	1268	BUT	BOSS	AIN'T	INFPGIFTNCU
21	20	1	20	ENFJ	1094	I	HAVE	FELT	SAD
22	21	1	21	ENFJ	1466	I	AM	A	LITTLE
23	22	1	22	ENFJ	1452	EXACTLY	MAYBE	IT	HAS

The aggregation process produced a balanced data set having 5 threads for each of the 16 MBTI types.

Because the feature extraction is O(n^2) in the number of threads, aggregation also greatly decreased the processing time required for each experiment.

4 Methodology

Reducing the "bag-of-words" threads to a fixed number of numeric features is most naturally accomplished by histogramming. This places words having some common attribute into bins weighted to create distributions from variable length collections of unstructured text.

4.1 Semantic Mapping: The Category File

To avoid the computational complexity of full-scope computational linguistic (which includes parsing, pronominal reference, stemming, synonymy, etc.), the authors have used approximate semantic tagging based upon a word list having pre-assigned "term categories" and numeric weights. This "Category File" is used for assigning semantic tags ("categories") and impact scores ("nuances") to words in threads by a hard match.

The Category File was not created using the MBTI corpus from this work. It was created manually by the authors under a previous effort, using two-years of posts (in colloquial English) from an online Sports Forum. The Forum was first stop-worded to remove "structural terms" (mostly conjunctions, prepositions, articles, etc.); then terms in that 10-million word corpus occurring with frequency above a significance threshold were collected. The resulting set consisted of 4,116 terms. These words were collected into 126 subjectively defined "semantic categories", and each word assigned a "nuance", subjectively expressing its "impactfulness" as an integer in the range −3 to +3. Negative and positive values indicate "bad" and "good" impact, resp. See Table 3.

Terms in a thread that do not occur in the category file are discarded; terms in a thread that are found in the category file are accumulated into the corresponding histogram bin for that thread.

The category file has 4,116 entries (rows), with five values in each row: Term, Part-of-Speech, Term Category, Frequency in Reference Corpus, Nuance. See Table 3.

4.2 Computing the Components of Thread Similarity

The extraction of features must preserve the salient similarities and differences between the entities being analyzed. Typically, the encoding of any discriminating information across the data set is unknown. It is, therefore, customary to select multiple entity attributes, encode them as features, and perform empirical experiments to determine their discriminating power.

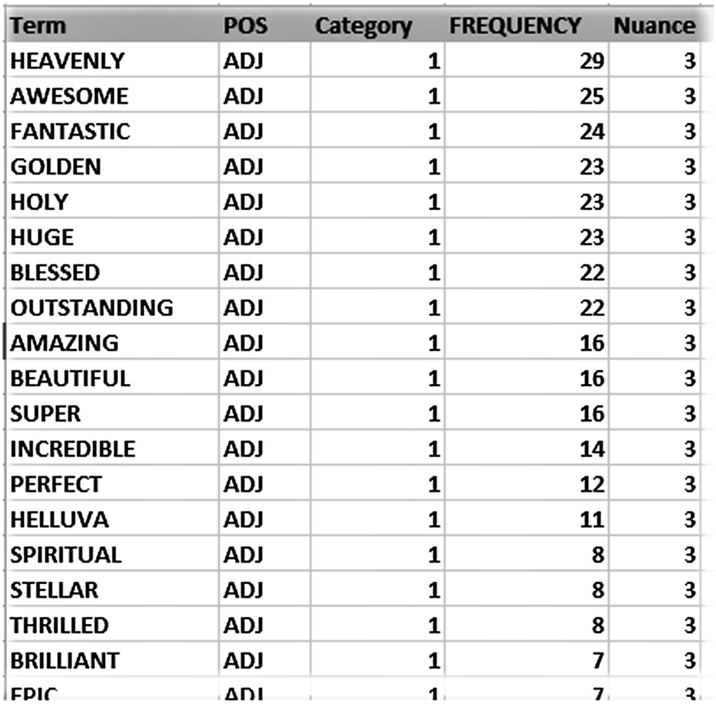
The following term statistics were computed for each term in each thread:

- Term Category (CAT)
- Term Nuance
- Term Frequency (TF)
- Term Inverse Document Frequency (idf)
- Term Frequency times Inverse Document Frequency (Tf.idf)

Each term in a thread will, in general, have different values for these components.

Given that there are thousands of terms in a thread, unifying these components directly (e.g., by placing them into an ordered n-tuple) would produce large,

Table 3. Category File entries give word attributes



Term	POS	Category	FREQUENCY	Nuance
HEAVENLY	ADJ	1	29	3
AWESOME	ADJ	1	25	3
FANTASTIC	ADJ	1	24	3
GOLDEN	ADJ	1	23	3
HOLY	ADJ	1	23	3
HUGE	ADJ	1	23	3
BLESSED	ADJ	1	22	3
OUTSTANDING	ADJ	1	22	3
AMAZING	ADJ	1	16	3
BEAUTIFUL	ADJ	1	16	3
SUPER	ADJ	1	16	3
INCREDIBLE	ADJ	1	14	3
PERFECT	ADJ	1	12	3
HELLUVA	ADJ	1	11	3
SPIRITUAL	ADJ	1	8	3
STELLAR	ADJ	1	8	3
THRILLED	ADJ	1	8	3
BRILLIANT	ADJ	1	7	3
EPIC	ADJ	1	7	3

unsynchronized vectors having different dimensions. Such a representation requires conformation of some kind to make it suitable for thread analysis. Histogramming mitigates this problem.

5 Determining Term Category and Nuance for Each Term in Each Thread

These are determined by hard match to the corresponding term entry in the Category File.

The categories provide a coarse semantic mapping. The 126 categories in the category file include “Salient Adjectives”, “Salient Adverbs”, “Parts of the Body”, “Filial Relationships”, “Major Cities and States”, “Common Names”, etc. The rationale is that pairs of threads using terms in related categories are likely to be addressing the same topic, making them more similar. Whether this is true was investigated by experiment, and is discussed below.

6 Computing Term Frequency, Tf (Document, Term), for Each Word in Each Thread

First, compute T1 for each word in each thread. T1(thread, "word") is the number of times, counting multiplicities, that "word" occurs in the thread under consideration:

```
T1(thread, "word") = # occurrences of "word" in document,
counting multiplicity.
```

The T1 value of the most frequently occurring word in a document is called T2 (thread) for that thread, and is used for normalization:

```
T2(thread) = # occurrences of most frequently occurring
word in the thread
```

We define Term Frequency each term in each thread as:

```
Tf(document, term) = T1(document, term)/T2(document)
```

7 Computing Inverse Document Frequency, Idf, for Each Word in a Document

Let I1 be the number of threads in the entire corpus:

```
I1 = # threads in the corpus
```

Then, for each term in the entire corpus, count the number of threads in the entire corpus that contain that term; call this

```
I2 ("word") = # threads in corpus containing "word"..
```

Form the expression: `idf("word") = log(I1/(1+I2("word")))`

Notice that the argument of the log is essentially the reciprocal of I2/I1, a "document frequency", so idf is called the inverse document frequency. The "1" in the denominator prevents division by zero, and the log pulls the result into a reasonable range, since this ratio is usually very large. It is customary to use either the Ln(x), or Log2(x). The "1" in the denominator is not used by all researchers; this matters little when the number of words is large, as it is in most applications.

8 Computing the Tf.Idf Score for Each Word in Each Thread

Note: Tf.idf (usually vocalized as a 5-letter acronym: “tfidf”) is the standard term for the product (Tf)(idf)

From Tf and idf, we obtain the Tf.idf score for each term in each thread by multiplying Tf ("thread", "word") by idf("word"):

```
Tf.idf(“thread”,”word”) =
(Tf("thread","word"))x(idf("word"))
```

The idea behind the Tf.idf score is that words that occur often in a particular thread, but are relatively rare in other thread, probably have meanings strongly related to the “meaning” of the threads in which they occur often: semantically significant terms in a thread are naturally expected to have high Tf, and high idf in that thread.

9 Developing Weighted Histograms from Which to Extract Features

Each thread can be viewed as a distribution by its word-frequency histogram using the terms in the Category File. This histogram will have 4,116 bins, because the Category File has entries for 4,116 terms.

The unweighted normalized frequency histograms for each MBTI type in the corpus are shown in Fig. 1. Notice the absence of artifacts that might indicate problems arising from disparity in word count among the MBTI types.

These unweighted normalized frequency histograms give the word selection distribution for each MBTI type. To fuse information from other term statistics for analysis, the corresponding terms bins can be weighted by those statistics. For example, if the Nuance scores for each term are multiplied by the corresponding BIN count in the histogram, we obtain a Nuance weighted histogram. This weighting can be carried out with any term statistic. The statistic will have more impact where the BIN count for that term is highest, automatically weighting the statistic by its “prevalence”.

In particular, the term statistics previously described can be used to scale the word-frequency histograms for threads. Thread similarity is performed by viewing the various weighted, term-frequency histograms as vectors in a 4,116 Euclidean space, where many metrics are available.

10 Creating Weighted Histograms for Threads

To create histograms for extracting features for a thread, the following are collected for each word it contains:

- CAT: binary Category (“1” if term is in Category File, else “0”)
- Nuance: the nuance value for the term in range [− 3, +3]

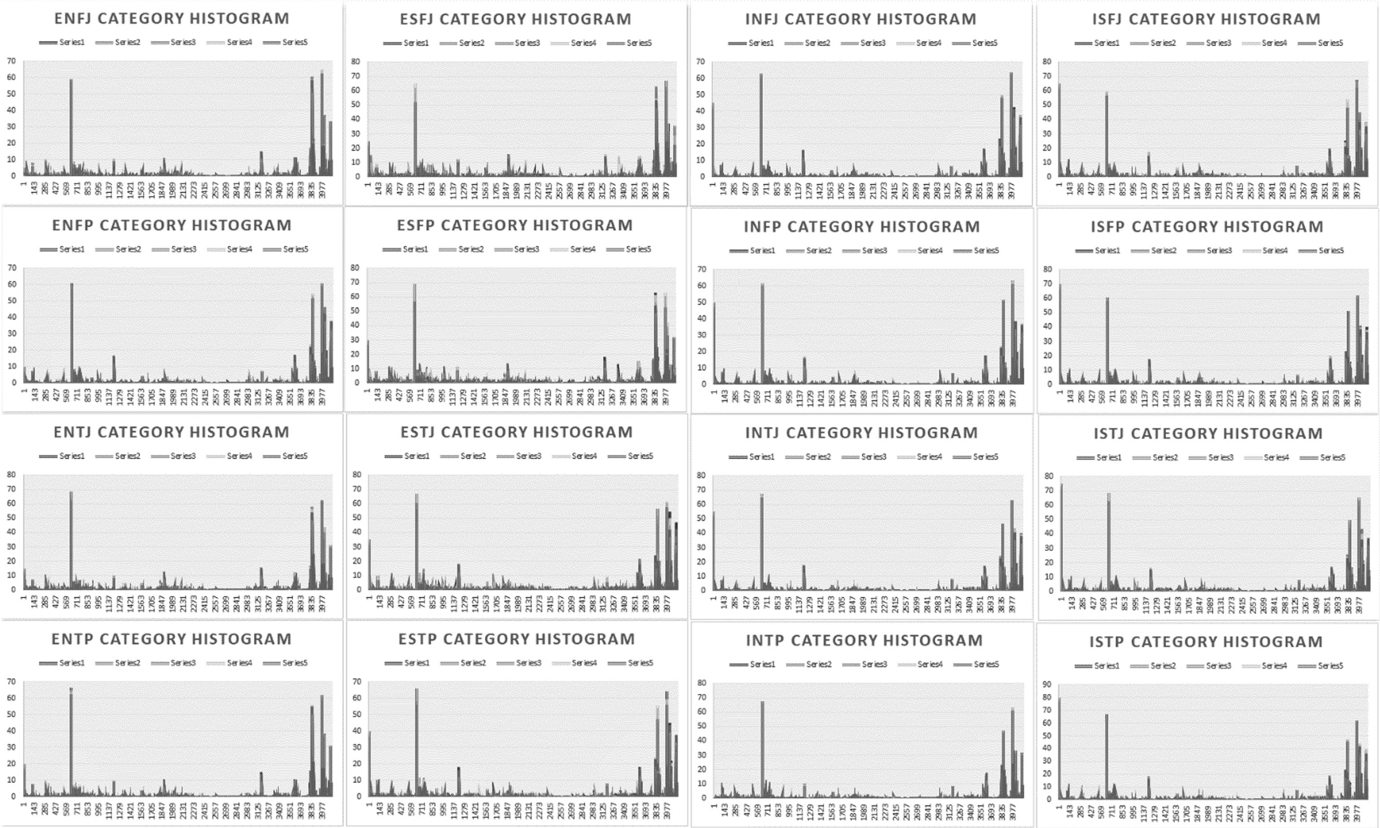


Fig. 1. The unweighted normalized frequency histograms for each MBTI type in the corpus.

- Term Frequency: (Tf) for the term in the document
- Inverse Document Frequency: (idf) for the term in the document
- TF.idf: the product of Tf and idf values for the term

Define the Relative Frequency for a term row as: nu = T1/I2, and accumulate four weighted histograms as follows:

```
CAT(author,tagrows)=nu*Category   (weighted ‶0″ or ‶1″)
NIV(author,tagrows)=nu*nuance     freq.-weighted nuance
TFV(author,tagrows)=nu*Tf         freq.-weighted Tf
IDFV(author,tagrows)=nu*idf       freq.-weighted idf
TFIDFV(author,tagrows)=nu*Tfidf   freq.-weighted Tfidf
```

Inferring Inter-Author Distances from Histograms Twenty-six metric values are then computed from the four histograms above. They fall into six groups, depending upon which of histograms are used:

- take dot products of histograms
- use ArcCos turns dot products into angles between normalized histograms
  - ([-pi, pi] rescaled to [− 1, 1], so 0 implies perpendicularity)
- compute RMS vector distance between histograms as 4,116-vectors
- compute Euclidean distance between histograms as 4,116-vectors

The computation of the feature values is performed by five software routines: Faze_A, Faze_B, Faze_C, Faze_D, and Faze_Alpha. Faze_Alpha is the controlling routine, calling the others in order A through D.

This produces twenty-seven sets of distance matrices between the 80 threads described in the next section, Feature Extraction Detailed Flow.

11 Feature Extraction Detailed Flow

```
Faze_Alpha:  Sequentially invoke Fazes A - E.
Faze_A: Read operating parameters from disc: Faze.prm
Faze_A: Read semantic tags from disc: tags.csv
Faze_A: Read in the entire text corpus from disc
Faze_A: Create an abridged Text Corpus (aTC): AnnText.csv
Faze_A: aTC all caps, no punctuation, comma delimited
Faze_A: aTC keeps only tagfile words (no misspellings)
Faze_A: Create word histogram for aTC: taghisto.csv
Faze_B: Read operating parameters from disc: Faze.prm
Faze_B: Read wordlist: wordlist.csv
Faze_B: Read the tagfile from disc: tags.csv
Faze_B: Scan wordlist threads: wordlist.csv
Faze_B: Compute term scores for all words in all threads:
Faze_B: T1 = occurrences of term in document
Faze_B: T2 = occurrences most frequently occurring term
Faze_B: Tf(term,document) = T1/T2
Faze_B: I1 = # documents in the corpus
Faze_B: I2 = # documents in corpus containing term
Faze_B: idf(term,corpus) = log(I1/(1+I2))
Faze_B: Tf.idf = Tf(term,thread)*idf(term,corpus)
Faze_B: For each word in each thread, tabulate:
Faze_B: Thread, Category,Nuance, T1 , T2 , I1 , I2,
Tf,idf, Tf.id: stored in file Tfidf.csv
Faze_C: Read operating parameters from disc: Faze.prm
Faze_C: Read in the thread term scores: Tfidf.csv
Faze C: Build the thread vector for this thread by con-
catenating records holding the term scores for each of
the terms in this thread (even when the thread is spread
across rows [and so, multiple same-MBTI-type authors] in
the original text corpus).
Faze_C: Write out the resulting thread vector, appending
records term-by-term, so that a single output row has the
various term scores for all the terms in this thread in
tag file order. This thread vector will be a weighted
roll-up of all the term scores for this thread; it is
similar to a relative frequency histogram for this
thread, and will have 4116 bins.
```

NOTE: histogram entries are freq.-novelty weighted by multiplication by the BIN T1/T2.

By this process, each thread generates five frequency histograms of 4116 BINS:

```
Cat(4116), histogram of term catergory counts
Nu(4116), histogram of term Nuances
TF(4116), histogram of Term Frequencies
IDF(4116), histogram of Inverse Document Frequencies
TFIDF(4116), histogram of term TF.idf Scores
```

Each histogram gives a different information-theoretic view of a thread.

These five histograms are then combined in various ways to obtain hybrid, non-linear distance ‘indicators’ of relatively low dimension. These combination strategies included: Euclidean distance, RMS distance, angle between histograms, and sum-of-squares.

These were collected into six groups, based upon which underlying statistic was used.

For each pair of threads, th1 and th2, twenty-seven thread similarity scores (“Metrics”) were created, as follows:

Group 1 thread similarity scores: Category

```
Metric 1  MIX_SOQ(th1,th2):    Non-Linear Mixing Feature
Metric 2  CAT_Edist(th1,th2):  Euclidean distance between
                               unweighted term histograms
Metric 3  CAT_RMS(th1,th2):    RMS distance between
                               unweighted term histograms
Metric 4  CAT_SOQ(th1,th2):    sum-of-squares of CAT
                               Edist, RMS, CAT_CAT
Metric 5  CAT_CAT(th1,th2):    Cosine Distance between
                               unweighted term histograms
```

```
Group 2 thread similarity scores: Nuance

Metric 6  NIV_Edist(th1,th2):  Euclidean distance between
                               Nuance-weighted term histos
Metric 7  NIV_RMS(th1,th2):    RMS distance between
                               Nuance-weighted term histos
Metric 8  NIV_SOQ(th1,th2):    sum-of-squares of NIV
                               Edist, RMS,_NIV_NIV
Metric 9  NIV_NIV(th1,th2):    Cosine distance between
                               Nuance-weighted term histos
```

Group 3 thread similarity scores: Term Frequency

```
Metric 10 TF_Edist(th1,th2):   Euclidean distance between
                               Tf-weighted term histos
Metric 11 TF_RMS(th1,th2):     RMS-distance between
                               Tf-weighted term histos
Metric 12 TF_SOQ(th1,th2):     sum-of-squares of TF
                               Edist, RMS, Tf_Tf
Metric 13 TF_TF(th1,th2):      Cosine distance between
                               Tf-weighted term histos
```

Group 4 thread similarity scores: Inverse Document Frequency

```
Metric 14 IDF_Edist(th1,th2):  Euclidean distance between
                               Idf-weighted term histos
Metric 15 IDF_RMS(th1,th2):    RMS distance between
                               Idf-weighted term histos
Metric 16 IDF_SOQ(th1,th2):    sum-of-squares of idf
                               Edist, RMS,idf_idf
Metric 17 IDF_IDF(th1,th2):    Cosine distance between
                               idf-weighted term histos
```

Group 5 thread similarity scores: TF.idf

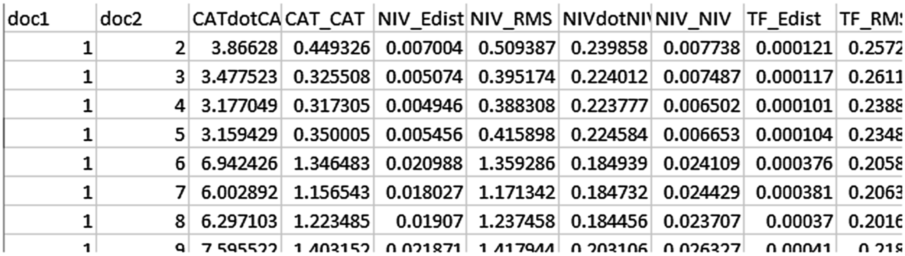
```
Metric 18 TFIDF_Edist(th1,th2):  Euclidean distance between
                               TF.idf-weighted term histos
Metric 19 TFIDF_RMS(th1,th2):  RMS distance between
                               Tf.idf-weighted term histos
Metric 20 TFIDF_SOQ(th1,th2):  sum-of-squares of Tf.idf
                               Edist, RMS, Tf.idf_Tf.idf
Metric 21 TFIDF_TFIDF(th1,th2):  Cosine distance between
                               Tf.idf-weighted term histos
```

Group 6 thread similarity scores: Histogram Dot products

```
Metric 22 NIV_TF_SOQ(th1,th2): sum-of-squares of
                               (NIV_SOQ)*(TF_SOQ)
Metric 23 NIV_IDF_SOQ(th1,th2):  sum-of-squares of
                               (NIV_SOQ)*(IDF_SOQ)
Metric 24 NIV_TFIDF_SOQ(th1,th2):  sum-of-squares of
                           (NIV_SOQ)*(TFIDF_SOQ)
Metric 25 TF_IDF_SOQ(th1,th2): sum-of-squares of
                               (TF_SOQ)*(IDF_SOQ)
Metric 26 TF_TFIDF_SOQ(th1,th2): sum-of-squares of
                               (TF_SOQ)*(TFIDF_SOQ)
Metric 27 IDF_TFIDF_SOQ(th1,th2):  sum-of-squares of
                               (IDF_SOQ)*(TFIDF_SOQ)
```

The pairwise thread similarities ("Metrics") for the 80 threads are placed into twenty-seven 80-by-80 pairwise distance matrices. For example, the 3,160 (non-trivial) RMS distances in inverse document frequency between threads th1 and th2 are in IDF_RMS(th1,th2), Table 4.

Table 4. Distances between pairs of threads



doc1	doc2	CATdotCA	CAT_CAT	NIV_Edist	NIV_RMS	NIVdotNI	NIV_NIV	TF_Edist	TF_RM
1	2	3.86628	0.449326	0.007004	0.509387	0.239858	0.007738	0.000121	0.2572
1	3	3.477523	0.325508	0.005074	0.395174	0.224012	0.007487	0.000117	0.2611
1	4	3.177049	0.317305	0.004946	0.388308	0.223777	0.006502	0.000101	0.2388
1	5	3.159429	0.350005	0.005456	0.415898	0.224584	0.006653	0.000104	0.2348
1	6	6.942426	1.346483	0.020988	1.359286	0.184939	0.024109	0.000376	0.2058
1	7	6.002892	1.156543	0.018027	1.171342	0.184732	0.024429	0.000381	0.2063
1	8	6.297103	1.223485	0.01907	1.237458	0.184456	0.023707	0.00037	0.2016
1	9	7.595522	1.403152	0.021871	1.417944	0.203106	0.026327	0.00041	0.218

Features for clustering and supervised learning are inferred from these distance matrices [8]. This is done by hypothesizing the existence of a low-dimensional point set having the same distance matrix as that developed for the pairs of threads. Such a point set is computed using gradient descent, and the coordinates of these points become abstract features for the corresponding thread.

Each row shows the twenty-seven different distances that have been computed between the pair of threads specified in the two leftmost columns. Twenty-seven, 6-dimensional feature sets were developed for the corpus in this way. Processing time is about 25 min for the 80-thread case described here.

```
Faze_D: Read operating parameters from disc: Faze.prm
Faze_D: Read in the 27 distance matrices, all held in a
single file: Delta_File.csv
Faze_D: For each distance matrix, use gradient descent to
infer low-dimensional Torgerson Coordinates for each
thread.
Faze_D: Save these 27 feature sets on disc for evalua-
tion.
```

Here are the 27 feature sets created for the corpus:
Group 1: Semantic Categories

```
CATdotCAT   inter-thread dot product of category
CAT_CAT     inter-thread Cosine dist. using category
```

Group 2: Term Nuances

```
NIV_Edist  inter-thread Euclidean distance using Nuances
NIV_RMS    inter-thread RMS distance using Nuances
NIVdotNIV  inter-thread dot product of Nuances
NIV_NIV    inter-thread Cosine dist. using Nuances
```

Group 3: Term_Frequencies

```
TFV_Edist  inter-thread Euclidean distance using Tf
TFV_RMS    inter-thread RMS distance using Tf
TFVdotTFV  inter-thread dot product of Tf
TFV_TFV    inter-thread Cosine dist. using Tf
```

Group 4: Inverse Document Frequencies

```
IDFV_Edist  inter-thread Euclidean distance using idf
IDFV_RMS    inter-thread RMS distance using idf
IDFVdotIDFV inter-thread dot product of idf
IDFV_IDFV   inter-thread Cosine dist. using idf
```

Group 5: Tfidfs

```
TFIDFV_Edist inter-thread Euclidean distance using TF.idf
TFIDFV_RMS   inter-thread RMS distance using Tf.idf
TFIDFVdotTFIDFV  inter-thread dot product of Tf.idf
TFIDFV_TFIDFV  inter-thread Cosine dist. using Tf.idf
```

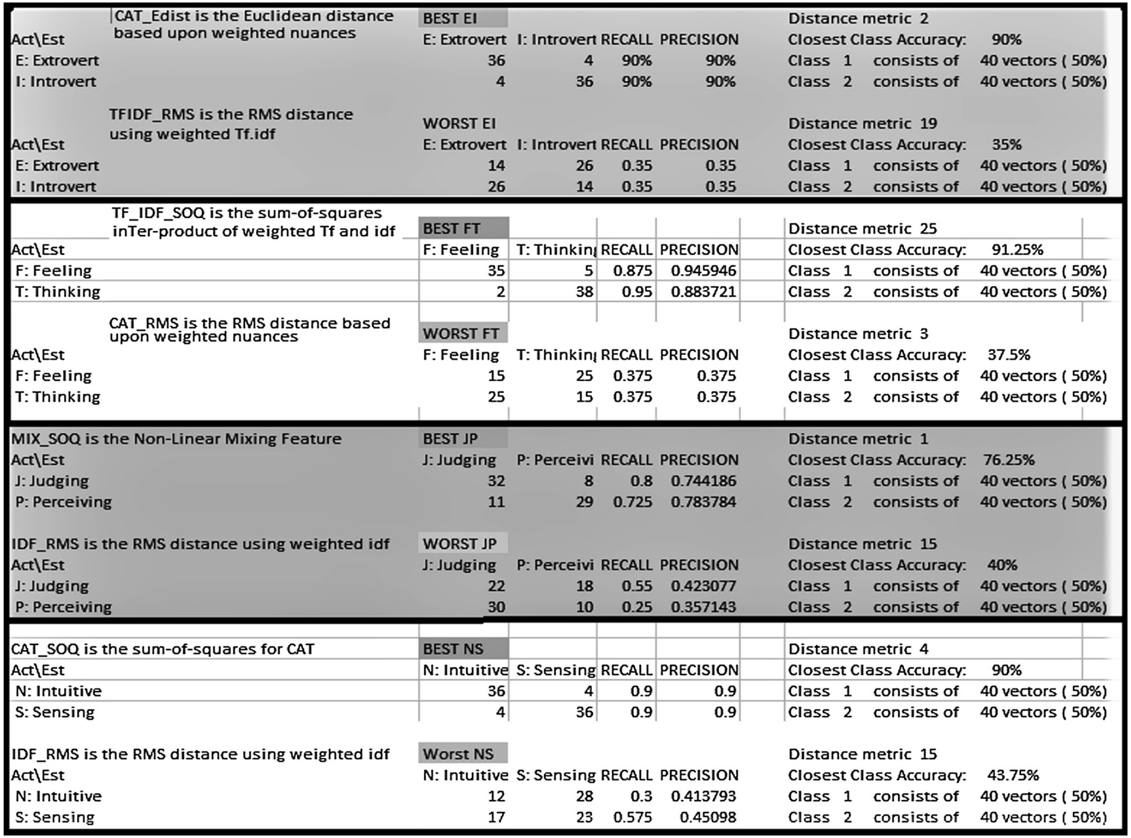
Group 6: Cross_Dots

```
NIVdotTFV    inter-thread dot product of Nuance & Tf
NIVdotIDFV   inter-thread dot product of Nuance & idf
NIVdotTFIDFV inter-thread dot product of Nuance & Tf.ID
TFVdotIDFV   inter-thread dot product of Tf & idf
TFVdotTFIDFV   inter-thread dot product of Tf & Tf.idf
IDFVdotTFIDFV  inter-thread dot product of idf & Tf.idf
```

12 Feature Validation Experiment

To determine whether feature sets derived by the method described here captured information distinguishing threads (and so, their authors) by MBTI type, each feature set was subjected to leave-one-out validation using a simple nearest neighbor classifier. See Table 5 (Fig. 2).

Table 5. The highest and lowest accuracy confusion matrices (showing precision, recall, and classification accuracy) for the single-letter MBTI classes.



	CAT_Edist is the Euclidean distance based upon weighted nuances	BEST EI				Distance metric 2		
Act\Est		E: Extrovert	I: Introvert	RECALL	PRECISION	Closest Class Accuracy:		90%
E: Extrovert		36	4	90%	90%	Class 1	consists of	40 vectors (50%)
I: Introvert		4	36	90%	90%	Class 2	consists of	40 vectors (50%)
	TFIDF_RMS is the RMS distance using weighted Tf.idf	WORST EI				Distance metric 19		
Act\Est		E: Extrovert	I: Introvert	RECALL	PRECISION	Closest Class Accuracy:		35%
E: Extrovert		14	26	0.35	0.35	Class 1	consists of	40 vectors (50%)
I: Introvert		26	14	0.35	0.35	Class 2	consists of	40 vectors (50%)
	TF_IDF_SOQ is the sum-of-squares inTer-product of weighted Tf and idf	BEST FT				Distance metric 25		
Act\Est		F: Feeling	T: Thinkin	RECALL	PRECISION	Closest Class Accuracy:		91.25%
F: Feeling		35	5	0.875	0.945946	Class 1	consists of	40 vectors (50%)
T: Thinking		2	38	0.95	0.883721	Class 2	consists of	40 vectors (50%)
	CAT_RMS is the RMS distance based upon weighted nuances	WORST FT				Distance metric 3		
Act\Est		F: Feeling	T: Thinkin	RECALL	PRECISION	Closest Class Accuracy:		37.5%
F: Feeling		15	25	0.375	0.375	Class 1	consists of	40 vectors (50%)
T: Thinking		25	15	0.375	0.375	Class 2	consists of	40 vectors (50%)
MIX_SOQ is the Non-Linear Mixing Feature		BEST JP				Distance metric 1		
Act\Est		J: Judging	P: Perceivi	RECALL	PRECISION	Closest Class Accuracy:		76.25%
J: Judging		32	8	0.8	0.744186	Class 1	consists of	40 vectors (50%)
P: Perceiving		11	29	0.725	0.783784	Class 2	consists of	40 vectors (50%)
IDF_RMS is the RMS distance using weighted idf		WORST JP				Distance metric 15		
Act\Est		J: Judging	P: Perceivi	RECALL	PRECISION	Closest Class Accuracy:		40%
J: Judging		22	18	0.55	0.423077	Class 1	consists of	40 vectors (50%)
P: Perceiving		30	10	0.25	0.357143	Class 2	consists of	40 vectors (50%)
CAT_SOQ is the sum-of-squares for CAT		BEST NS				Distance metric 4		
Act\Est		N: Intuitive	S: Sensing	RECALL	PRECISION	Closest Class Accuracy:		90%
N: Intuitive		36	4	0.9	0.9	Class 1	consists of	40 vectors (50%)
S: Sensing		4	36	0.9	0.9	Class 2	consists of	40 vectors (50%)
IDF_RMS is the RMS distance using weighted idf		Worst NS				Distance metric 15		
Act\Est		N: Intuitive	S: Sensing	RECALL	PRECISION	Closest Class Accuracy:		43.75%
N: Intuitive		12	28	0.3	0.413793	Class 1	consists of	40 vectors (50%)
S: Sensing		17	23	0.575	0.45098	Class 2	consists of	40 vectors (50%)

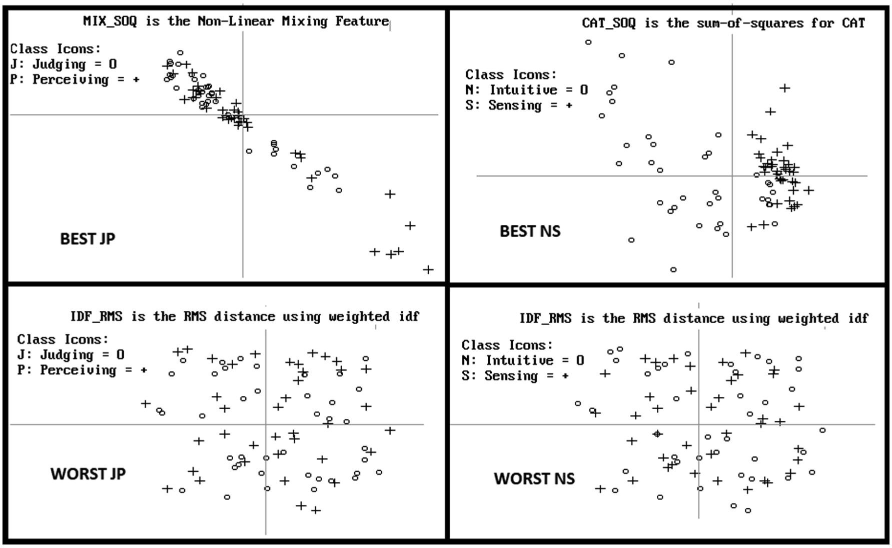


Fig. 2. Here are some typical pair-plots projected on the first two of 6 features generated.

13 Conclusions

The "metric numbers" here refer to those defined in Sect. 11 above.

The most successful metrics for distinguishing extroversion/introversion (E/I) are metrics 1, 2, 4 and 5. The recall for extroverts range from 80% to 95% among these metrics, while the recall for introverts range from 82.5% to 90%. The precision for extroverts range from 84% to 90% among these metrics, while the precision for introverts range from 81.8% to 94.3%. The other metrics were not as successful in distinguishing E/I and the recall and precision figures from most of those metrics were typically in the 40% to 70% range.

Metrics 1 and 2 compare the category bits of one class to another, while metrics 4 and 5 compare their weighted nuances. The data suggests that there is a clear distinction between the words favored by the extroverts and those favored by the introverts. The category bits involve sorting words into various categories based on their meanings. The strong results from the metrics involving category bits could possibly mean the extroverts in the data set were gravitated toward discussing certain subjects and the introverts toward other subjects, or perhaps extroverts and introverts write/speak differently. The results from metrics 4 and 5, which involve comparing weight nuances, would also follow as a result if this supposition were true.

While some of the other metrics also involve the weighted nuances of the words, these metrics typically compare the weighted nuances with some other factor that is stripped of the semantics, stripped of the meanings of the words. For example, the least successful metric for distinguishing E/I is metric 19, which compares the weighted nuances for one class against the term frequencies of words for the other class. The recall and precision figures for both extroverts and introverts were all 35% using this metric.

The most successful metric for distinguishing feeling/thinking (F/T) is metric 4. The recall for feeling is 87.5% while the recall for thinking is 95%. The precision for feeling is 94.6% while the precision for thinking is 88.4%. Note that metric 4 involves the weighted nuances of words, so this result is not surprising.

Metrics 1, 2, 5, 11, 14, 15, 18, 19 and 26 have recall and precision figures for F/T typically in the 70% to 80% range. The other unsuccessful metrics have recall and precision figures typically in the 40% to 70% range. Metrics 1, 2 and 5 emphasize the effect of the semantic content of words and thus were expected to perform respectably in distinguishing F/T. Metrics 11, 14, 15, 18, 19 and 26 involve non-semantic factors such as the inverse document frequency and term frequency of words. These other factors show up in both the semi-successful as well as the unsuccessful metrics for distinguishing F/T, so the importance of these factors is not without question.

The most successful metrics for distinguishing judging/perceiving (J/P) are metrics 1 and 2. The recall and precision figures for judging and perceiving for both of these metrics range from 70% to 80%. The recall and precision figures for the less successful metrics were mostly in the 50% to 70% range, with quite a few in the 60% to 70% range.

These results suggest that while the categories of words used in metrics 1 and 2 was the most successful factor in distinguishing J/P, many of the other metrics were not too far behind. Furthermore, success in the 70% to 80% range when it comes to recall and precision is not extremely remarkable in the first place. These results possibly suggest that J/P is not something that is strongly distinguishable from the words that people use, but rather from their actions or behaviors instead. Note that the MBTI test itself asks questions related to actions and behaviors when assessing J/P.

The most successful metrics for distinguishing intuitive/sensing (N/S) are metrics 1, 2, 4, 5, 20 and 23. The recall and precision figures for N/S typically range from 80% to 90% for these metrics, although some of the figures were as high as 97.5%, such as the case of the recall for intuitive for metric 5. The less successful metrics had recall and precision figures typically in the 50% to 70% range. The success of metrics 1, 2, 4 and 5 in distinguishing N/S suggests that people who classify as intuitives write/speak differently from those who classify as sensing, or perhaps they gravitate toward different subjects for discussion.

Of particular note is metric 23, which compares the term frequency of words in one class to the tf.idf scores of another. This is a non-semantic factor and the recall and precision figures for N/S ranged from 82.5% to 95% for metric 23.

Overall, metrics 1 and 2 were the most successful in distinguishing all four of the pairs E/I, F/T, J/P and N/S. This suggests the various pairs can potentially be distinguished by the way people write/speak, or perhaps by the subjects they are interested in discussing. J/P was the hardest pair to distinguish since it involves looking at people's actions and behaviors to distinguish, not just words.

14 Future Work

Some potential directions to go for future work include analyzing combinations of the various Myers-Briggs binary classifications, such as NT or SJ. There is literature to suggest that such pairings are significant. For example, people who type as NTs are referred to as “rationals” in Keirsey’s temperament sorter, and those who type as SJ are called “guardians”.

Another possible direction for future work is to study the verbosity of people as it relates to their Myers-Briggs personality types. For example, it is plausible that introverts are more verbose in writing than extroverts, given the introverts’ preference for writing over speaking. This direction might shine light on the role played by the non-semantic factors studied in this project such as term frequency, and this direction might greater fine-tune the metrics used to distinguish personality types.

Acknowledgements. We gratefully acknowledge the support of the Sirius Project, and the INTJForum [5].

References

1. https://www.thebalance.com/what-is-social-sentiment-and-why-is-it-important-3960082
2. http://www.myersbriggs.org
3. http://www.humanmetrics.com
4. http://PersonalityCafe.com
5. http://IntjForum.com/
6. Lee, S., Song, J., Kim, Y.: An empirical comparison of four text mining methods, 2010Utility. In: 18th International Conference on Human Computer Interaction, Toronto, Canada, July 2016
7. https://www.kaggle.com
8. Lebanon, G.: Information geometry, the embedding principle, and document classification. In: Proceedings of the 2nd International Symposium on Information Geometry and its Applications, pp. 101–108 (2005)

Bibliography

9. Hancock, M., Sessions, C., Lo, C., Rajwani, S., Kresses, E., Bleasdale, C., Strohschein, D.: Stability of a type of cross-cultural emotion modeling in social media. In: Schmorrow, D.D., Fidopiastis, C.M. (eds.) AC 2015. LNCS (LNAI), vol. 9183, pp. 410–417. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20816-9_39
10. Hancock, M.: Practical Data Mining. CRC Press, Boca Raton (2011)
11. Delmater, R., Hancock, M.: Data Mining Explained: A Manager’s Guide to Customer-Centric Business Intelligence. Digital Press, Boston (2001)

12. Hancock, M., et al.: Field-theoretic modeling method for emotional context in social media: theory and case study. In: Schmorrow, D.D., Fidopiastis, C.M. (eds.) AC 2015. LNCS (LNAI), vol. 9183, pp. 418–425. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20816-9_40

13. Carapinha, F., et al.: Modeling of social media behaviors using only account metadata. In: Schmorrow, D.D., Fidopiastis, C.M. (eds.) AC 2016. LNCS (LNAI), vol. 9744, pp. 393–401. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39952-2_38