Predicting Myers-Briggs Type Indicator From Written Texts Using Supervised Learning

Linus Kortesalmi, linko538@student.liu.se

***Abstract*—Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.**

I. Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

II. Theory

This section will briefly cover the basics of Myers-Briggs Type Indicator, language modelling, feature representations, common supervised learning models and evaluation metrics.

A. Myers-Briggs Type Indicator

The foundation of the Myers-Briggs Type Indicator (MBTI) is derived from the work of Carl G. Jung [1]. The idea behind MBTI was to make the concepts observed by Jung more accessible to the public [2]. Table I contains the 16 personality types defined by Myers-Briggs [2].

TABLE I. Myers-Briggs Type Indicator Table

ISTJ	ISFJ	INFJ	INTJ
ISTP	ISFP	INFP	INTP
ESTP	ESFP	ENFP	ENTP
ESTJ	ESFJ	ENFJ	ENTJ

1) Introversion and Extroversion: Introversion (I) and extroversion (E) is used to describe what an individual focuses on and draws their energy from. Introversion means that the individual focuses on his/her own "world", which means that they draw energy from ideas and memories and from self-reflection [3]. Extroversion is the opposite side; energy is drawn from activities and events with others [3]. Problems for people with the extroversion type are typically solved by talking and discussing with others out loud [3].

2) Intuition and Sensing: Intuition (N) and sensing (S) describes how individuals handles information. Intuition means that individuals pay more attention to patterns, impressions, theoretical concepts and abstract theories [4]. Sensing is, on the other hand, connected to the five senses. Information is gathered from the physical connection to the world, e.g., what you hear, smell and touch [4].

3) Thinking and Feeling: Thinking (T) and feeling (F) covers how individuals perform decision making. Individuals with the thinking type typically analyse ground truths and use logical reasoning to make a decision; pros and cons are analysed [5]. The feeling type describes people who take the opinion of others into consideration [5]. They weigh the outcome of the situation and try to maintain harmony, by caring about the point-of-view of others [5].

4) Judging and Perceiving: Judging (J) and perceiving (P) is the final pair of the MBTI. This pair is related to how others typically see you. People with the judging trait usually live more structured lives than those with the perceiving trait [6]. The major difference between this pair and the pairs covered before is that this trait is based on the side individuals let others see [6]. For example, an individual can be structured and live ordered lives when alone, but in the interaction with others they tend to be interpreted as spontaneous and open-ended [6].

B. Document Representation

A document d_i can be mathematically represented as a vector $d_i = (w_1, \ldots, w_N)$, where w_i is the i-th word (term) in the document. The words (terms) could, for example, be unigrams (Section II-D2), bigrams (Section II-D4) or letters (Section II-D5).

C. Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) model produces top topic terms for each topic in the model. It can also be used to calculate the topic distribution for a document.

The LDA model, described by Blei et al. in [7], is a probabilistic generative model built on the notion of underlying topics. A document can belong to several topics simultaneously (with a probability $p(\alpha_i)$ of belonging to topic α_i). All the terms in a collection can belong to their own topic(s). Each term in a document in a corpus is assumed to be generated from a topic (following a multinomial distribution with a latent parameter drawn from a Dirichlet distribution). The LDA model can thus for a given document calculate the probability of that document belonging to some topic α_i.

D. Features

1) Normalisation: Normalisation transforms the data values above 0 in the feature vector to be in the range $(0, 1]$.

2) Bag-Of-Words: The bag-of-words model is a simple representation of a document. The model ignores the order of words and the words occurring before the i-th word (w_i):

$$p(w_i|w_1, \dots, w_{i-1}) = p(w_i)$$

The feature vector consists of occurrence counts of words in a dictionary, where each word is represented by an index in the vector. The dictionary can either be predefined or created from the training set. One limitation of the bag-of-words model is how to handle words not present in the training corpus. Unknown words can either be discarded or replaced with an *UNK* tag. The count vector can then be normalised, smoothed or used as-is.

3) TF-IDF: Term frequency and inverted document frequency (TF-IDF) can be constructed from a bag-of-words model. The TF is the word (term) counts for a document (typically normalised) and the IDF is the document frequency for that word

$$IDF_{k_i} = log(N/n_i)$$

where N is the number of documents and n_i is the number of documents where term k_i appears.

4) Bigram Model: The bigram model is similar to the bag-of-words model in Section II-D2. Instead of using counts of single words the bigram model uses pairs of adjacent words as features:

$$p(w_i|w_1, \dots, w_{i-1}) = p(w_i|w_{i-1})$$

The appearance order for different pairs is ignored after construction; it is only the counts of pairs that are interesting.

5) Bag-Of-Letters: The bag-of-letters feature vector is a simplified variant of the Bag-of-words model. The features are constructed from a predefined alphabet consisting of letters and characters. Similar to the bag-of-words model, the symbols in the alphabet are counted for each document. The feature vector can then be normalised or used as-is.

6) Topic Distribution (TD): The LDA model can calculate the probability distribution

$$T_{d_i} = [p(\alpha_1), \dots, p(\alpha_t)]$$

over all t topics ($\alpha_1, \dots, \alpha_t$) for a given document d_i. $p(\alpha_i)$ is the probability that document d_i belongs to topic α_i. The distribution can thus be used to determine which topic(s) the document was generated from. This probability distribution can be directly used as a feature vector for various models.

7) Topic Terms (TT): In [8], Chen et al. try to use LDA to improve the performance of Support Vector Machines for text classification. In their work they combine the topic distribution of a document from a LDA model together with the top terms for the topics. The same approach is tested in this work. The topic terms feature vectors are constructed by first calculating the topic distribution T_{d_i} for each document d_i. Then, for each topic α_i, the n top topic terms ($\beta_{i,1}, \dots, \beta_{i,n}$) are used to construct the feature vector. The topic distribution and the topic terms can be retrieved from a LDA model with t topics. The feature vector f_{d_i} for a document d_i is thus defined by

$$f_{d_i} = [\beta_1, \dots, \beta_t]$$

where β_i is short-hand notation for the values $\beta_{i,1}, \dots, \beta_{i,n}$. The dimension of a topic terms feature vector for one document is $t * n$ feature values. Each value in the feature vector is initialised to zero.

The topic distribution determines how many top topic terms shall be counted from each topic in the LDA model. If topic α_i has probability p_i, then the number of top topic terms to be used (n_i) from topic α_i is given by Equation 1.

$$n_i = Round(p_i * n) \tag{1}$$

The $n - n_i$ top terms not used in topic α_i have their feature values set to zero. The feature value for each of the n_i top terms is set according to Equation 2.

$$feature\ value = \begin{cases} 1 + TF, & if\ term\ present \\ 1, & otherwise \end{cases} \tag{2}$$

If the term is present in document d_i it is counted and the feature value is set to $1 + TF$. TF is the number of times the term occurs in document d_i. If the term is not in the document, the feature value for the term is set to 1. If a document d_i is calculated by the LDA to belong to a topic α_i, then Equation 2 would give more weight to the top terms that do occur in d_i, compared to the missing ones.

For example, a LDA with five topics ($t = 5$) and a document d_i with the topic distribution

$$T_{d_i} = [0.01, 0.15, 0.75, 0.09, 0]$$

and $n = 10$ top topic terms would result in a feature vector f_{d_i} where:

- n_1 is rounded down to zero, thus setting all values for the 10 top topic terms in α_1 to zero.
- n_2 is rounded up to two, which means that only the values for the two top terms in α_2 are updated to $1+TF$.

- n_3 is rounded up to 8, which means the values for the 8 top terms in topic α_3 are updated.
- n_4 is rounded up to one, which means that only the value for the top term in topic α_4 is updated.
- n_5 is zero, thus all 10 values are set to zero (same as n_1).

E. Models

Various supervised models can be used for classification. This section will very briefly cover the models used in this work.

1) Linear Support Vector Classification: The linear Support Vector Classifier (LinSVC) can efficiently handle sparse features[1]. The SVC has a slightly different mathematical formulation than the traditional Support Vector Machine (SVM), but the idea behind the model is the same as in the SVM case. A hyperplane is constructed in order to try and separate the different classes with as big of a margin as possible. The margin is defined as the distance from the hyperplane to the nearest point. Slack can be introduced in order to allow for misclassifications, as needed in the case of data not being linearly separable. Regularisation is typically used in order to control the slack, otherwise the margin would be maximised by potentially misclassifying all data points.

2) Logistic Regression Classifier: The Logistic Regression (LR) classifier, also called logit or MaxEnt, is a probabilistic classifier consisting of linear combinations of predictor functions and weights. The predictor function calculates the probability of a data point X_i belonging to a class K_i ($P(K_i|X_i)$), divided by the probability that it belongs to all other classes ($P(\{K \setminus K_i\}|X_i)$).

3) Extra Trees Classifier: The Extra Trees (ET) classifier is a variant of the well-known Random Forest (RF) classifier. Instead of choosing the best decision boundary at each split it continuously chooses attributes and cut-points at random [9]. It can be heavily tuned to the problem at hand with various parameters [9].

4) AdaBoost Classifier: The AdaBoost (AB) classifier is an ensemble method which combines multiple weak learners (estimators) [10]. The later weak learners in the chain are tweaked in order to solve those samples that were misclassified by the previous learners [10]. The result of each weak learner is combined into a weighted sum [10].

5) Gradient Boosting Classifier: The Gradient Boosting (GB) classifier is a variant of the AdaBoost classifier. If the exponential loss function is used, then it turns into the AdaBoost classifier. The model creates multiple decision trees that are, in each iterative stage, fitted with the negative gradient of the multinomial deviance loss function.

F. Exploratory Data Analysis

Exploratory data analysis (EDA) is a broad concept containing various techniques [11], [12], [13] that can be used to explore and analyse data. Some early techniques include stem-and-leaf plots and box plots. EDA can be seen as applying tools and statistics in a meaningful way. It could, for example, be used to perform variance analysis, detect outliers or smooth the data [11], [12], [13].

1) Word clouds: In the field of NLP, the use of word clouds can help visualise important aspects of the data. Word clouds are generated by simply counting the occurrence of each word in the data set and presenting the most common words in an image. The most frequent words are typically presented with a larger font.

G. Evaluation Metrics

Common evaluation metrics in the NLP field are Accuracy (Equation 3), Precision (Equation 4), Recall (Equation 5) and F1 Score (Equation 6). They are computed by calculating the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{6}$$

III. Related Work

In [14], Ma et al. train Neural Networks (NNs) to predict Myers-Briggs personality types based on the writing style. The data used in [14] was manually crafted by generating a mapping between famous authors and their MBTI. The text in books written by the authors were then retrieved and used in the work. The best accuracy obtained by Ma et al. was 37%, using a Recurrent NN with long short-term memory (LSTM).

IV. Method

This section covers the data set used in this work and the preprocessing and feature extraction used. The implementations of the classification models used are briefly mentioned. The work was implemented using Python[2] version 3.6.2.

A. Data Set

The data set used in this work was downloaded from Kaggle[3]. It contains data from 8675 Personality Cafe [4] forum users. Each data entry contains the 50 latest posts from a user, together with the MBTI of the user. 30% of the data was used as a test set and the training set consisted of the remaining 70%.

[1] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[2] https://www.python.org/

[3] https://www.kaggle.com/

[4] http://personalitycafe.com/forum/

B. Exploratory Data Analysis

A simple EDA was performed to better understand the data. This work mainly used word clouds, bar plots and document content analysis.

1) Word clouds: The Python library `word_cloud`[5] was used to generate word clouds. First a word cloud was generated from all the documents in the data set. The idea behind this general word cloud was to see if there were any terms or topics that needed further investigation. After the general word cloud was generated the documents were partitioned on their respective MBTI labels. A word cloud was then generated for each MBTI label. Appendix B contains all the word clouds generated.

C. Preprocessing

The data set was preprocessed according to the following steps:

1) Standardising: The data set was standardised so that any leading or trailing "`'`" characters were removed.

2) Tagging: All URLs in the data set were replaced with a `<URL>` tag. Only URLs were replaced with a tag in this work.

3) Tokenising: The data was tokenised using the tokeniser from the `NLTK`[6] package. The `NLTK` package also contains an English stop word list that were used to remove stop words.

4) Stemming: The Snowball stemmer from the `NLTK` package was used to stem the tokens in the preprocessing step.

D. Features

The features covered in Section II-D were extracted and tested with the models in Section II-E. This section covers the implementation-specific details for the feature extraction used in this work.

1) Filter Levels: Three *filter levels* were introduced in order to try and improve the quality of the different feature models:

a) Normal Filter: This filter level only applies the first two steps in the preprocessing mentioned in Section IV-C.

b) Type Filter: This filter level applies the standard preprocessing, but it also removes any MBTIs found in the data set. The filtering is done by adding the MBTIs to the stop-word list.

c) Extreme Filter: This filter level applies both the standard preprocessing and the removal of MBTIs, as in the case of the Type Filter. It also removes any tokens that occur in less than five documents or in more than 50% of the documents.

2) Latent Dirichlet Allocation: The library `gensim`[7] was used to implement the LDA model. Six LDA models with 10, 16, 25, 50, 75 and 100 topics were trained for each filter level. The LDA models were used to extract the topic terms and topic distribution feature vectors covered in Section II-D.

3) Topic Terms Feature Vector: The topic term feature vector covered in Section II-D7 was implemented using 10 topic terms. The feature vector was normalised for the LinSVC, GB and ET classifiers.

[5] https://github.com/amueller/word_cloud

[6] http://www.nltk.org/

[7] https://radimrehurek.com/gensim/

E. Model Implementation and Training

The models used the implementations in the `scikit-learn`[8] package for Python. The main parameters for each model was tuned using the `GridSearchCV` class, which allows a grid of parameter values to be tested for each model using cross-validation. The cross-validation used in this work was `StratifiedKFold` using five folds. The stratified k-folds tries to maintain the imbalance of the data set in the different folds. The `F1_micro` score was used to choose the best cross-validated model during the `GridSearchCV` parameter tuning. The micro score globally counts all TPs, TNs, FPs and FNs, which is more suitable if the data set is imbalanced.

1) Initial Models: The first part was to use only the standardised and tagged data set (see Section IV-D1a) in a LR model. The classifier was tested using the bag-of-words model, the bigram model and the TF-IDF model. The best LR model obtained through the `GridSearchCV` process was then evaluated on the test set.

The next step was to try the ET classifier on the standardised and tagged data set. The features were extracted by using TF-IDF with a unigram model, followed by truncated Singular Value Decomposition (SVD) to reduce the number of features. `GridSearchCV` was used to optimise the parameters for TF-IDF, truncated SVD and the ET classifier.

2) Type Filter Models: The second part was to filter out the MBTIs from the data set, see Section IV-D1b. Instead of using the built-in tokenisation methods in `scikit-learn`, a custom corpus was built using the `NLTK` library. All the models briefly covered in Section II-E were used on the type-filtered data set. The parameters for each model was individually optimised using `GridSearchCV`.

3) Extreme Filter Models: The final part was to also filter out extreme words from the data set, see Section IV-D1c. The same models are used as in Section IV-E2. The parameters were again optimised using `GridSearchCV`.

V. Results

The results obtained from the initial model (Section IV-E1) are presented in Table II. The Extra Trees (ET) classifier performed best with a cross-validated training accuracy of 64.9%. The Logistic Regression (LR) classifier has a cross-validated training accuracy of 47.5%.

Table III presents the results for the type-filtered models (Section IV-E2). The best type-filtered model is a Linear SVC (LinSVC) model trained with a topic distribution (TD) feature vector for 100 topics. The best cross-validated training accuracy achieved is 28.9%.

Table IV has the results for the extreme-filtered models (Section IV-E3). The best extreme-filtered model is a Gradient Boosting (GB) classifier with a TD for 16 topics as a feature vector. The cross-validated training accuracy achieved with the GB model is 29.9%.

The complete list of results are presented in Appendix A.

[8] http://scikit-learn.org/stable/

TABLE II. RESULTS FROM INITIAL MODEL

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
ET	0.649	0.649	0.65	0.65	0.64
LR	0.473	0.475	0.64	0.47	0.51

TABLE III. TOP 5 RESULTS FROM TYPE FILTER MODELS

Model	Topics	Train Acc	Test Acc	Precision	Recall	F1
LinSVC (TD)	100	0.289	0.273	0.28	0.27	0.20
LinSVC (TT)	100	0.288	0.285	0.23	0.29	0.22
GB (TD)	100	0.286	0.273	0.21	0.27	0.22
GB (TT)	100	0.286	0.273	0.21	0.27	0.22
LinSVC (TT)	50	0.277	0.279	0.20	0.28	0.21

TABLE IV. TOP 5 RESULTS FROM EXTREME FILTER MODELS

Model	Topics	Train Acc	Test Acc	Precision	Recall	F1
GB (TD)	16	0.299	0.303	0.27	0.30	0.27
LinSVC (TT)	75	0.294	0.287	0.23	0.29	0.23
GB (TT)	16	0.291	0.297	0.27	0.30	0.26
LinSVC (TT)	16	0.291	0.298	0.25	0.30	0.25
LinSVC (TT)	50	0.289	0.280	0.23	0.28	0.23

VI. DISCUSSION

Why was only URLs tagged?

A. Data Set Limitations

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

VII. CONCLUSION

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

VIII. FUTURE WORK

Use Number tagging and User tagging. Maybe use URL analysis?

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

REFERENCES

[1] C. G. Jung, *Psychological types*. Routledge, 2014.

[2] Myers-Briggs, "MBTI Basics," http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/home.htm?bhcp=1, 2018, [Online; accessed 21-Jan-2018].

[3] ——, "Extraversion or Introversion," http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/extraversion-or-introversion.htm, 2018, [Online; accessed 21-Jan-2018].

[4] ——, "Sensing or Intuition," http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/sensing-or-intuition.htm, 2018, [Online; accessed 21-Jan-2018].

[5] ——, "Thinking or Feeling," http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/thinking-or-feeling.htm, 2018, [Online; accessed 21-Jan-2018].

[6] ——, "Judging or Perceiving," http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/judging-or-perceiving.htm, 2018, [Online; accessed 21-Jan-2018].

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[8] Y. H. Chen and S. F. Li, "Using latent dirichlet allocation to improve text classification performance of support vector machine," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 1280–1286.

[9] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Apr 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6226-1

[10] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[11] D. C. Hoaglin, "John w. tukey and data analysis," *Statistical Science*, pp. 311–318, 2003.

[12] J. W. Tukey, "Exploratory data analysis," 1977.

[13] P. F. Velleman and D. C. Hoaglin, *Applications, basics, and computing of exploratory data analysis*. Duxbury Press, 1981.

[14] A. Ma and G. Liu, "Neural networks in predicting myers brigg personality type from writing style."

APPENDIX A
TYPE FILTER AND EXTREME FILTER RESULTS

This appendix contains the results from using the implemented models on data with different filter levels (see Section IV-D1). Table V and Table VI contain the results from using the Type Filter (Section IV-D1b) and the Extreme Filter (Section IV-D1c), respectively.

TABLE V. RESULTS FROM TYPE FILTERING

Normalized	Topics	Filter Level	Feature Type	Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
No	-	Types	Characters	Linear SVC	0.2621870882740448	0.26930464848252017	0.19	0.27	0.20
No	-	Types	Characters	Logistic Regression	0.11495388669301712	0.1260084517864003	0.25	0.13	0.16
Yes	-	Types	Characters	Extra Trees	0.2439064558629776	0.24587014982712255	0.16	0.25	0.17
No	-	Types	Characters	AdaBoost	0.22266139657444006	0.2155205532078371	0.21	0.22	0.13
Yes	-	Types	Characters	GradBoost	0.24769433465085638	0.2397233960814445	0.17	0.24	0.18
-	100	Types	Topic Distribution	Linear SVC	0.2887022397891963	0.27276219746446406	0.28	0.27	0.20
-	100	Types	Topic Distribution	Logistic Regression	0.13554018445322794	0.13753361505954667	0.25	0.14	0.15
-	100	Types	Topic Distribution	Extra Trees	0.25148221343873517	0.24433346139070303	0.21	0.24	0.21
-	100	Types	Topic Distribution	AdaBoost	0.255764163372859	0.2535535920092201	0.17	0.25	0.14
-	100	Types	Topic Distribution	GradBoost	0.28573781291172595	0.27276219746446406	0.21	0.27	0.22
-	75	Types	Topic Distribution	Linear SVC	0.26515151515151514	0.26661544371878604	0.19	0.27	0.20
-	75	Types	Topic Distribution	Logistic Regression	0.10013175230566534	0.11025739531310026	0.28	0.11	0.13
-	75	Types	Topic Distribution	Extra Trees	0.22990777338603424	0.2320399538993469	0.19	0.23	0.20
-	75	Types	Topic Distribution	AdaBoost	0.2524703557312253	0.2451018056089128	0.15	0.25	0.15
-	75	Types	Topic Distribution	GradBoost	0.2666337285902503	0.25509028044563964	0.22	0.26	0.20
-	50	Types	Topic Distribution	Linear SVC	0.2723978919631094	0.2723780253553592	0.17	0.27	0.19
-	50	Types	Topic Distribution	Logistic Regression	0.15184453227931488	0.16135228582404917	0.30	0.16	0.18
-	50	Types	Topic Distribution	Extra Trees	0.23517786561264822	0.24241260084517863	0.21	0.24	0.22
-	50	Types	Topic Distribution	AdaBoost	0.2689393939393939	0.269688820591625	0.16	0.27	0.19
-	50	Types	Topic Distribution	GradBoost	0.272068511198946	0.276219746446408	0.22	0.28	0.23
-	25	Types	Topic Distribution	Linear SVC	0.26366930171277997	0.263542066845947	0.18	0.26	0.17
-	25	Types	Topic Distribution	Logistic Regression	0.10622529644268774	0.11870918171340761	0.28	0.12	0.14
-	25	Types	Topic Distribution	Extra Trees	0.2017457180500659	0.19708029197080296	0.18	0.20	0.19
-	25	Types	Topic Distribution	AdaBoost	0.2498353096179183	0.24932769880906647	0.17	0.25	0.15
-	25	Types	Topic Distribution	GradBoost	0.25889328063241107	0.2685363042643104	0.23	0.27	0.21
-	16	Types	Topic Distribution	Linear SVC	0.2658102766798419	0.2669996158278909	0.15	0.27	0.18
-	16	Types	Topic Distribution	Logistic Regression	0.14410408432147562	0.14560122935074912	0.17	0.15	0.14
-	16	Types	Topic Distribution	Extra Trees	0.19911067193675888	0.1901651940069151	0.17	0.19	0.18
-	16	Types	Topic Distribution	AdaBoost	0.25757575757575757	0.24625432193622743	0.11	0.25	0.14
-	16	Types	Topic Distribution	GradBoost	0.2682806324110672	0.2673837879369958	0.21	0.27	0.21
-	10	Types	Topic Distribution	Linear SVC	0.26301054018445325	0.26815213215520556	0.14	0.27	0.17
-	10	Types	Topic Distribution	Logistic Regression	0.12269433465085638	0.1244717633499808	0.20	0.12	0.12
-	10	Types	Topic Distribution	Extra Trees	0.18593544137022397	0.1828659239339224	0.17	0.18	0.18
-	10	Types	Topic Distribution	AdaBoost	0.25774044795783924	0.25701114099116407	0.18	0.26	0.19
-	10	Types	Topic Distribution	GradBoost	0.25559947299077734	0.2558586246638494	0.19	0.26	0.20
Yes	100	Types	Topic Terms + TF	Linear SVC	0.2877140974967062	0.2854398770649251	0.23	0.29	0.22
Yes	100	Types	Topic Terms + TF	Logistic Regression	0.13142292490118576	0.13638109873223203	0.24	0.14	0.14
No	100	Types	Topic Terms + TF	Extra Trees	0.25905797101449274	0.26046868997310796	0.21	0.26	0.22
No	100	Types	Topic Terms + TF	AdaBoost	0.2625164690382082	0.26008451786400305	0.14	0.26	0.17
No	100	Types	Topic Terms + TF	GradBoost	0.28573781291172595	0.27276219746446406	0.21	0.27	0.22
Yes	75	Types	Topic Terms + TF	Linear SVC	0.26613965744400525	0.2593161736457933	0.20	0.26	0.20
Yes	75	Types	Topic Terms + TF	Logistic Regression	0.10737812911725955	0.11717249327698809	0.28	0.12	0.14
No	75	Types	Topic Terms + TF	Extra Trees	0.2437417654808959	0.2489435266999616	0.20	0.25	0.21
No	75	Types	Topic Terms + TF	AdaBoost	0.24588274044795783	0.2431809450633884	0.14	0.24	0.15
No	75	Types	Topic Terms + TF	GradBoost	0.2666337285902503	0.25509028044563964	0.22	0.26	0.20
Yes	50	Types	Topic Terms + TF	Linear SVC	0.2765151515151515	0.27890895121014214	0.20	0.28	0.21
Yes	50	Types	Topic Terms + TF	Logistic Regression	0.155467720685112	0.17018824433346139	0.31	0.17	0.19
No	50	Types	Topic Terms + TF	Extra Trees	0.25461133069828723	0.25662696888205916	0.22	0.26	0.22
No	50	Types	Topic Terms + TF	AdaBoost	0.25757575757575757	0.26431041106415676	0.19	0.26	0.14
No	50	Types	Topic Terms + TF	GradBoost	0.272068511198946	0.276219746446408	0.22	0.28	0.23
Yes	25	Types	Topic Terms + TF	Linear SVC	0.2621870882740448	0.2685363042643104	0.21	0.27	0.20
Yes	25	Types	Topic Terms + TF	Logistic Regression	0.10869565217391304	0.1152516327314637	0.26	0.12	0.14
No	25	Types	Topic Terms + TF	Extra Trees	0.2330368906455863	0.24356511717249327	0.20	0.24	0.20
No	25	Types	Topic Terms + TF	AdaBoost	0.2549407114624506	0.253937764118325	0.16	0.25	0.18
No	25	Types	Topic Terms + TF	GradBoost	0.25889328063241107	0.2685363042643104	0.23	0.27	0.21
Yes	16	Types	Topic Terms + TF	Linear SVC	0.274703557312253	0.2831348444102958	0.22	0.28	0.22
Yes	16	Types	Topic Terms + TF	Logistic Regression	0.13339920948616601	0.13138686131386862	0.20	0.13	0.13
No	16	Types	Topic Terms + TF	Extra Trees	0.24571805006587616	0.23933922397233962	0.19	0.24	0.20
No	16	Types	Topic Terms + TF	AdaBoost	0.25329380764163373	0.2431809450633884	0.16	0.24	0.15
No	16	Types	Topic Terms + TF	GradBoost	0.2682806324110672	0.2673837879369958	0.21	0.27	0.21
Yes	10	Types	Topic Terms + TF	Linear SVC	0.26910408432147565	0.27929312331924705	0.18	0.28	0.20
Yes	10	Types	Topic Terms + TF	Logistic Regression	0.12697628458498023	0.13215520553207838	0.22	0.13	0.14
No	10	Types	Topic Terms + TF	Extra Trees	0.24703557312252963	0.24202842873607375	0.19	0.24	0.20
No	10	Types	Topic Terms + TF	AdaBoost	0.2554347826086957	0.25854782942758353	0.16	0.26	0.18
No	10	Types	Topic Terms + TF	GradBoost	0.25559947299077734	0.2558586246638494	0.19	0.26	0.20

TABLE VI. RESULTS FROM EXTREME FILTERING

Normalized	Topics	Filter Level	Feature Type	Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
No	-	Extremes	Characters	Linear SVC	0.2621870882740448	0.26930464848252017	0.19	0.27	0.20
No	-	Extremes	Characters	Logistic Regression	0.11495388669301712	0.1260084517864003	0.25	0.13	0.16
Yes	-	Extremes	Characters	Extra Trees	0.2513175230566535	0.2427967729542835	0.18	0.24	0.17
No	-	Extremes	Characters	AdaBoost	0.2185441370223979	0.21667306953515175	0.13	0.22	0.15
Yes	-	Extremes	Characters	GradBoost	0.24341238471673254	0.24932769880906647	0.21	0.25	0.20
-	100	Extremes	Topic Distribution	Linear SVC	0.2768445322793149	0.27276219746446406	0.21	0.27	0.20
-	100	Extremes	Topic Distribution	Logistic Regression	0.1541501976284585	0.15097963887821744	0.26	0.15	0.18
-	100	Extremes	Topic Distribution	Extra Trees	0.25774044795783924	0.2535535920092201	0.22	0.25	0.22
-	100	Extremes	Topic Distribution	AdaBoost	0.25	0.25163273146369575	0.14	0.25	0.16
-	100	Extremes	Topic Distribution	GradBoost	0.26498682476943347	0.2742988859008836	0.25	0.27	0.22
-	75	Extremes	Topic Distribution	Linear SVC	0.283596837944664	0.2908182865923934	0.23	0.29	0.23
-	75	Extremes	Topic Distribution	Logistic Regression	0.15151515151515152	0.15328467153284672	0.29	0.15	0.18
-	75	Extremes	Topic Distribution	Extra Trees	0.27618577075098816	0.28006146753745675	0.24	0.28	0.24
-	75	Extremes	Topic Distribution	AdaBoost	0.266304347826087	0.2612370341913177	0.17	0.26	0.17
-	75	Extremes	Topic Distribution	GradBoost	0.28343214756258234	0.2885132539377641	0.24	0.29	0.25
-	50	Extremes	Topic Distribution	Linear SVC	0.2817852437417655	0.28275067230119094	0.21	0.28	0.22
-	50	Extremes	Topic Distribution	Logistic Regression	0.14015151515151514	0.14022281982328083	0.27	0.14	0.16
-	50	Extremes	Topic Distribution	Extra Trees	0.26811594202898553	0.2612370341913177	0.21	0.26	0.22
-	50	Extremes	Topic Distribution	AdaBoost	0.255764163372859	0.2581636573184787	0.16	0.26	0.18
-	50	Extremes	Topic Distribution	GradBoost	0.28540843214756256	0.29273914713791777	0.25	0.29	0.25
-	25	Extremes	Topic Distribution	Linear SVC	0.2840909090909091	0.2908182865923934	0.22	0.29	0.24
-	25	Extremes	Topic Distribution	Logistic Regression	0.1529973649538867	0.1590472531694199	0.28	0.16	0.18
-	25	Extremes	Topic Distribution	Extra Trees	0.25461133069828723	0.26776796004610065	0.23	0.27	0.23
-	25	Extremes	Topic Distribution	AdaBoost	0.26399868247694336	0.2719938532462543	0.22	0.27	0.20
-	25	Extremes	Topic Distribution	GradBoost	0.2875494071146245	0.2981175566653861	0.24	0.30	0.25
-	16	Extremes	Topic Distribution	Linear SVC	0.28804347826086957	0.2892815981559739	0.21	0.29	0.22
-	16	Extremes	Topic Distribution	Logistic Regression	0.13768115942028986	0.1371494429504418	0.28	0.14	0.13
-	16	Extremes	Topic Distribution	Extra Trees	0.25214097496706195	0.2558586246638494	0.22	0.26	0.23
-	16	Extremes	Topic Distribution	AdaBoost	0.2628458498023715	0.25662696888205916	0.19	0.26	0.18
-	16	Extremes	Topic Distribution	GradBoost	0.29907773386034253	0.3031117940837495	0.27	0.30	0.27
-	10	Extremes	Topic Distribution	Linear SVC	0.2702569169960474	0.2704571648098348	0.19	0.27	0.19
-	10	Extremes	Topic Distribution	Logistic Regression	0.12928194993412384	0.13446023818670763	0.27	0.13	0.16
-	10	Extremes	Topic Distribution	Extra Trees	0.19746376811594202	0.21744141375336154	0.20	0.22	0.21
-	10	Extremes	Topic Distribution	AdaBoost	0.26103425559947296	0.2685363042643104	0.18	0.27	0.20
-	10	Extremes	Topic Distribution	GradBoost	0.2755270092226614	0.28275067230119094	0.22	0.28	0.24
Yes	100	Extremes	Topic Terms + TF	Linear SVC	0.2845849802371542	0.2746830580099885	0.22	0.27	0.23
No	100	Extremes	Topic Terms + TF	Logistic Regression	0.1615612648221344	0.16442566269688821	0.28	0.16	0.19
Yes	100	Extremes	Topic Terms + TF	Extra Trees	0.27108036890645587	0.26776796004610065	0.22	0.27	0.23
No	100	Extremes	Topic Terms + TF	AdaBoost	0.24687088274044797	0.25662696888205916	0.19	0.26	0.16
Yes	100	Extremes	Topic Terms + TF	GradBoost	0.27322134387351776	0.2812139838647714	0.24	0.28	0.24
Yes	75	Extremes	Topic Terms + TF	Linear SVC	0.2936429512516469	0.28659239339223974	0.23	0.29	0.23
No	75	Extremes	Topic Terms + TF	Logistic Regression	0.15942028985507245	0.14521705724164424	0.26	0.15	0.17
Yes	75	Extremes	Topic Terms + TF	Extra Trees	0.2781620553359684	0.28006146753745675	0.24	0.28	0.24
No	75	Extremes	Topic Terms + TF	AdaBoost	0.258399209486166	0.2581636573184787	0.16	0.26	0.17
Yes	75	Extremes	Topic Terms + TF	GradBoost	0.2765151515151515	0.28659239339223974	0.24	0.29	0.25
Yes	50	Extremes	Topic Terms + TF	Linear SVC	0.2890316205533597	0.28044563964656166	0.23	0.28	0.23
No	50	Extremes	Topic Terms + TF	Logistic Regression	0.1529973649538867	0.1498271225509028	0.29	0.15	0.18
Yes	50	Extremes	Topic Terms + TF	Extra Trees	0.2763504611330698	0.27775643488282753	0.23	0.28	0.23
No	50	Extremes	Topic Terms + TF	AdaBoost	0.2549407114624506	0.2597003457548982	0.12	0.26	0.14
Yes	50	Extremes	Topic Terms + TF	GradBoost	0.28573781291172595	0.28620822128313483	0.24	0.29	0.24
Yes	25	Extremes	Topic Terms + TF	Linear SVC	0.28804347826086957	0.2908182865923934	0.22	0.29	0.24
No	25	Extremes	Topic Terms + TF	Logistic Regression	0.12878787878787878	0.12946600076834422	0.27	0.13	0.16
Yes	25	Extremes	Topic Terms + TF	Extra Trees	0.274703557312253	0.28236650019208603	0.23	0.28	0.24
No	25	Extremes	Topic Terms + TF	AdaBoost	0.2564229249011858	0.26392623895505185	0.18	0.26	0.19
Yes	25	Extremes	Topic Terms + TF	GradBoost	0.2824440052700922	0.2896657702650788	0.24	0.29	0.25
Yes	16	Extremes	Topic Terms + TF	Linear SVC	0.2906785243741766	0.2981175566653861	0.25	0.30	0.25
No	16	Extremes	Topic Terms + TF	Logistic Regression	0.15349143610013175	0.15635804840568573	0.29	0.16	0.19
Yes	16	Extremes	Topic Terms + TF	Extra Trees	0.27585638998682477	0.2850557049558202	0.23	0.29	0.24
No	16	Extremes	Topic Terms + TF	AdaBoost	0.26235177865612647	0.269688820591625	0.20	0.27	0.22
Yes	16	Extremes	Topic Terms + TF	GradBoost	0.29117259552042163	0.29658086822896657	0.27	0.30	0.26
Yes	10	Extremes	Topic Terms + TF	Linear SVC	0.27618577075098816	0.2758355743373031	0.22	0.28	0.21
No	10	Extremes	Topic Terms + TF	Logistic Regression	0.11940052700922266	0.11794083749519785	0.25	0.12	0.14
Yes	10	Extremes	Topic Terms + TF	Extra Trees	0.25922266139657446	0.26392623895505185	0.22	0.26	0.22
No	10	Extremes	Topic Terms + TF	AdaBoost	0.2572463768115942	0.2547061083365348	0.16	0.25	0.16
Yes	10	Extremes	Topic Terms + TF	GradBoost	0.28194993412384717	0.2819823280829812	0.25	0.28	0.24

APPENDIX B
WORD CLOUDS

This section presents the word clouds generated as part of the exploratory data analysis. Figure 1 shows the word cloud for all the documents in the data set. Figures 2 to 17 show the word clouds generated from documents when partitioned according to the Myers-Briggs type label of the documents. The word cloud for each partitioned data set clearly shows the MBTI label partitioning the data set to also be a common word in that partition of documents.

Fig. 1. word cloud generated from all the documents in the data set. Visualises the most common words in the whole data set.

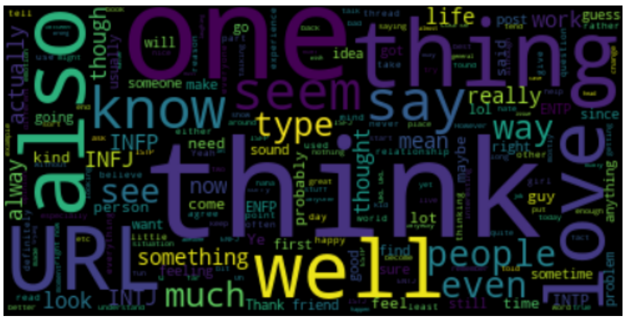

Fig. 2. word cloud generated from all documents with the ISTJ Myers-Briggs type indicator. As we can see, "ISTJ" is a common word for this subset of documents.

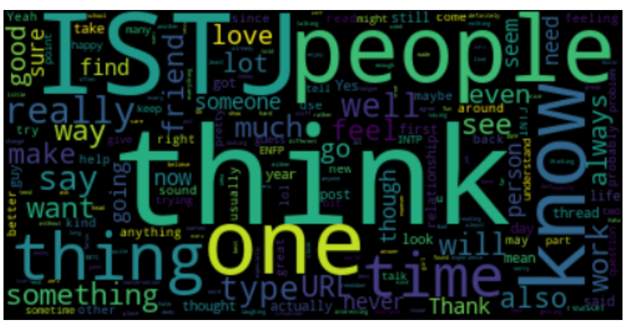

Fig. 3. word cloud generated from all documents with the ISFJ Myers-Briggs type indicator. As we can see, "ISFJ" is a common word for this subset of documents.

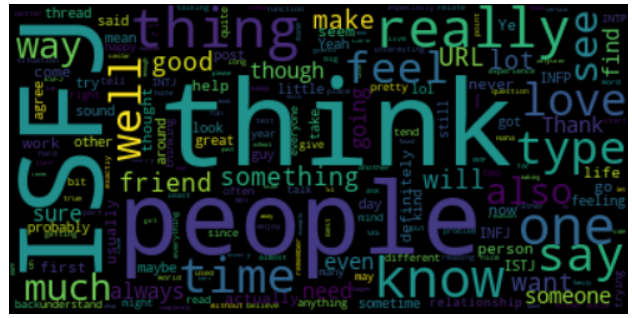

Fig. 4. word cloud generated from all documents with the INFJ Myers-Briggs type indicator. As we can see, "INFJ" is a common word for this subset of documents.

Fig. 5. word cloud generated from all documents with the INTJ Myers-Briggs type indicator. As we can see, "INTJ" is a common word for this subset of documents.

Fig. 6. word cloud generated from all documents with the ISTP Myers-Briggs type indicator. As we can see, "ISTP" is a common word for this subset of documents.

Fig. 7. word cloud generated from all documents with the ISFP Myers-Briggs type indicator. As we can see, "ISFP" is a common word for this subset of documents.

Fig. 8. word cloud generated from all documents with the INFP Myers-Briggs type indicator. As we can see, "INFP" is a common word for this subset of documents.

Fig. 9. word cloud generated from all documents with the INTP Myers-Briggs type indicator. As we can see, "INTP" is a common word for this subset of documents.

Fig. 10. word cloud generated from all documents with the ESTP Myers-Briggs type indicator. As we can see, "ESTP" is a common word for this subset of documents.

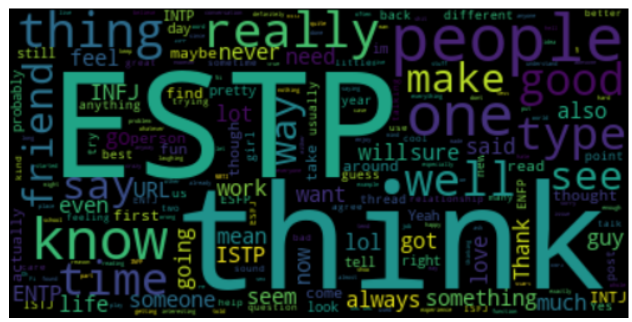

Fig. 11. word cloud generated from all documents with the ESFP Myers-Briggs type indicator. As we can see, "ESFP" is a common word for this subset of documents.

Fig. 12. word cloud generated from all documents with the ENFP Myers-Briggs type indicator. As we can see, "ENFP" is a common word for this subset of documents.

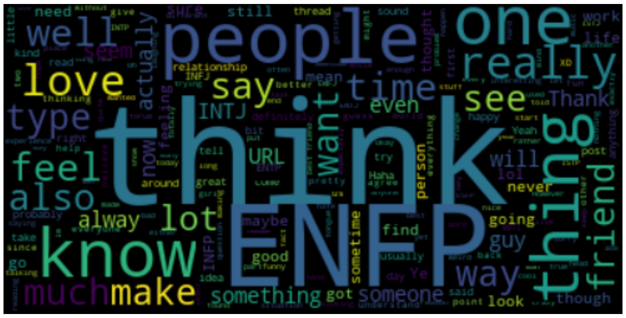

Fig. 13. word cloud generated from all documents with the ENTP Myers-Briggs type indicator. As we can see, "ENTP" is a common word for this subset of documents.

Fig. 14. word cloud generated from all documents with the ESTJ Myers-Briggs type indicator. As we can see, "ESTJ" is a common word for this subset of documents.

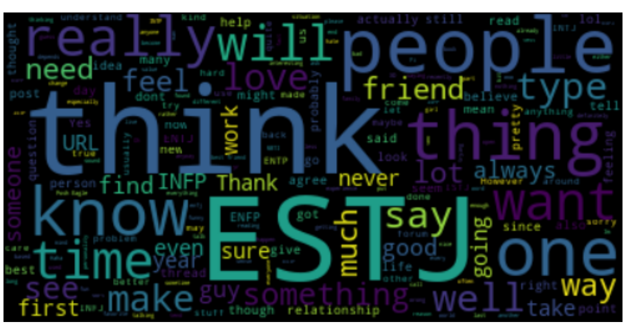

Fig. 15. word cloud generated from all documents with the ESFJ Myers-Briggs type indicator. As we can see, "ESFJ" is a common word for this subset of documents.

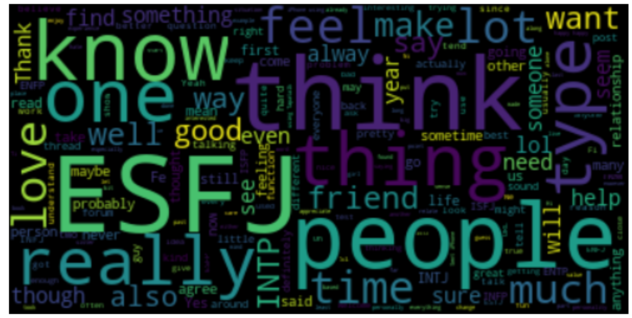

Fig. 16. word cloud generated from all documents with the ENFJ Myers-Briggs type indicator. As we can see, "ENFJ" is a common word for this subset of documents.

Fig. 17. word cloud generated from all documents with the ENTJ Myers-Briggs type indicator. As we can see, "ENTJ" is a common word for this subset of documents.