Your Personality Speaks: An Exploration to Explainability of Personality Classification

Kyungwook Lee
20193396
l1905kw@kaist.ac.kr

Jihyeon Lee
20203511
jihyeonlee@kaist.ac.kr

Minseok Choi
20208147
minseok.choi@kaist.ac.kr

Kyungduk Kim
Naver Corp.
kyungduk.kim@navercorp.com

1 Introduction

NLP has brought a significant impact to a wide range of areas in life because language is used everywhere, and deep learning has advanced enough to analyzing them effectively. This study starts with the question: "Can one's writing disclose their personality?" We are interested in finding out if there is a strong relationship between one's use of language and their actual personality. Every human being has an instinctive desire to learn about their own personality along with personalities of people around them. If the NLP technology can derive a person's personality from their writing, it can be used as an auxiliary tool to understand each individual, and going further, diversities in the world.

The theory of personality is very diverse depending on the perspective, but symbolically, the Myers-Briggs Type Indicator (MBTI) is one of the most well-known and widely used descriptors of personality type. MBTI is designed to be used more easily and usefully in everyday life based on C. G. Jung's psychological theory, describing the way people behave and interact with the world around them with four binary categories, as shown in Table 1.

Table 1: MBTI categories

Char	Metric
Energy	Extrovert/Introvert
Information	Sensing/iNtuition
Decision	Thinking/Feeling
Lifestyle	Judging/Perceiving

Four categories make a total of 16 types. The four categories combine to create synergy, creating different tendencies. For example, ESTP and ESTJ differ only in their last personality, but when combined, their individual tendencies vary significantly. Therefore, it was not easy to place 16 personalities in proper positions in coordinates, and it would be more socially meaningful if we extract representative features and relationships by applying NLP to this domain.

In a recently conducted study, Cui and Qi [4] applied numerous machine learning methods on the MBTI personality type prediction. Softmax classifier, Naive Bayes, SVM, and LSTM-based models were adopted, and the LSTM network brought the best performance of 38% test accuracy. In addition to previous work, we focus on explainable aspects and improving model performance.

Our two main goals are (1) to predict one's MBTI personality type and (2) to analyze more than what the MBTI type means. First, we classify the personalities of the people into 16 MBTI types from given texts. A number of experiments have been conducted to detect patterns in people's behavior and have demonstrated that there are different uses of language within each MBTI category. However,

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

they showed somewhat unsatisfactory results in terms of 16 MBTI types classification, which means that they still have difficulty generalizing relationships between language usage and personality. Therefore, we develop a better classification model to understand human behavior. Second, we find words or sentences which have the greatest impact on the output of our model. Conventional deep learning models have been considered as black box models because it is difficult to understand which features have had an effect on the results, making further analysis challenging. Thus, by developing an interpretable MBTI type classifier, we find a preferred word or sentence composition for a particular personality. It is expected that by finding relationships between words and personalities, a simpler and more approximate method of MBTI type classification, which is easier than the current MBTI test, can be suggested.

We explore various neural network models to predict people's MBTI types with their short written text. We formulated the task as a text classification problem having 16 MBTI types as target classes. We use several machine learning algorithms for the baseline of the work. Then, we implement CNN- and RNN-based methods to compare the performance. To find out whether there is a bottleneck problem of RNN models, we also experiment on a self-attention-based encoder – namely, BERT [6] – for representing the input text.

We also investigate various ways to visualize how the model predicts people's MBTI types on the model's inference. Attention mechanism usually allows the model to focus on the relevant expressions of the input text as needed [10]. This makes the attention used as a tool for interpretation of the decision process of our model. We adopt the attention mechanism to show which part of the text plays an important role in predicting the type.

2 Related Work

There are several ways to classify personalities into different categories according to the person's way of thinking and behavior, such as the 5-Factor Inventory and the MBTI, and studies have been conducted to see if there is a difference in language use according to the classified personality. Chang H. Lee [8] analyzed stream-of-consciousness essays using the Korean version of Linguistic Inquiry and Word Count (KLIWC) and observed significant correlation between personality traits and linguistic variables. Based on the potential to language use as a marker of personality, many personality classification systems have been developed. Kim Luyckx [9] conducted 8 binary MBTI classification (e.g. I or not-I) to predict a particular author's personality from a given essay using kNN. Likewise, a number of researchers [3, 1] used various machine learning models and feature vectors to predict a user's binary MBTI type (e.g. E/I) from given tweets. As the deep learning model developed, Cui and Qi [4] predicted one's binary and 16-class MBTI type from one of their social media posts using LSTM as well as various machine learning techniques. Keh and Cheng [7] scraped posts from online forums and not only classified 16-class MBTI type but also generated personality-specific sentences using a pre-trained language model. These previous studies demonstrated somewhat good performance, but there still remains a room for improvement, such as analyzing the differences in language use according to the MBTI personality types.

3 Method

3.1 Dataset

We employ the MBTI dataset available from Kaggle, an open-source data platform. The dataset contains texts written from individuals having each type of personality, carefully collected through the Personality Cafe forum. Specifically, the dataset is comprised of over 8,600 rows of data in which each row represents each person's four-letter MBTI code, as well as the last 50 things that they have posted on the forum. However, due to the rarity of certain personalities, the dataset is imbalanced as shown in Table 2. To accommodate the skewness of the data, we incorporate various metrics in addition to accuracy, such as balanced accuracy and F1-score.

3.2 Preprocessing

The dataset comes from online communities, which contains unrefined texts including emojis, URLs, and special characters; therefore, we preprocess the data by omitting and replacing unnecessary

Table 2: Data per label.

MBTI	INFP	INFJ	INTP	INTJ	ENTP	ENFP	ISTP	ISFP
count	1832	1470	1304	1091	685	675	337	271
MBTI	ENTJ	ISTJ	ENFJ	ISFJ	ESTP	ESFP	ESFJ	ESTJ
count	231	205	190	166	89	48	42	39

characters and strings. After filtering, we split our task into two cases: classifying by post or by person.

Preprocessing by post To increase the training samples of the dataset, we split each row of the data by the number of posts it contains (generally 45 to 50). Then the dataset was shuffled and cut into the 80:10:10 ratio for training, validation, and test sets, respectively. Subsequently, we applied the same preprocessing step for BERTweet [5], which tokenizes each post using TweetTokenizer from the NLTK toolkit and uses the emoji package to translate emoticons into text strings. In addition, we converted user mentions, URLs, and progress bars (i.e. 'l') into special tokens `@USER`, `HTTPURL` and `BARS`, respectively.

Preprocessing by person The downside of per-person preprocessing is the relatively small amount of data for classification: we only so have about 8,600 examples to fine-tune our model. Thus, we adopt somewhat of a data augmentation technique as well. More concretely, because the maximum sequence length of the BERT model is 512, it automatically truncates to that length, losing the rest of the information in the dataset. Because the dataset we use has token lengths between 500 to 2,000 per person, we wrap the data in a way that all of the tokens are conserved. For example, if the corresponding label has 2,000 tokens, its text is split into 4 examples containing about 500 tokens each. This approach not only allows fully utilizing the data, but it also maintains the format of per-person training, which can potentially solve the data quantity and accuracy trade-off.

3.3 Models

We describe the details of the models we used in our experiments here. Since none of the previous studies have shown reliable and outstanding results, we focus on evaluating which model performs the best score by using our unique preprocessed data. From the classical methods such as Naive Bayes and KNN, deep learning based models including CNN, LSTM, and BERT are examined.

Baseline For the baselines, Extra Trees Classifier (ETC), Naive Bayes, Logistic Regression (LR), K-Nearest Neighbor (KNN), Extreme Gradient Boosting (XGB) are implemented. All of the models use TF-IDF based feature vectors. In other words, our baselines only consider the appearance of words and do not take into account of the semantics of words that deep learning models can consider. Hyperparameters of ETC, Naive Bayes, LR are set to the same values as experiments from Kaggle Kernels[1], while KNN hyperparameters are set to default and XGB are the same from the post here[2].

LSTM Long Short-Term Memory (LSTM) is a special type of RNN which solves the problem of exploding and vanishing gradients. In this model, we considered our given task as machine translation task, thereby implementing the model having an encoder-decoder structure. The model encodes posts as vectors, and encoded vectors are then translated to their corresponding MBTI types. We also incorporated the attention mechanism used in Bahdanau et al. [2] to see which words have a meaningful effect on the model's output by obtaining their attention scores. Our LSTM model was a 2-layer bidirectional network with embedding and hidden dimension size of 128 and 64, respectively. The model is optimized with Adam optimizer and the (learning rate, maximum token number) pair was set to (0.00002, 100), (0.0002, 512), (0.0005. 1024) for per-post task, per-person task, and per-person-augmented task. Dropout ratio was set to 0.3 and early stopping strategy was performed.

CNN CNN have emerged as widely used architecture for text classification. In order to capture correlation between co-occurred words, CNNs perform excellently in extracting various n-gram features from a sentence through convolution layers, and can learn short and relatively long-range relations through pooling mechanisms. Our CNN model used embedding layer with dimension of

[1] https://www.kaggle.com/lbronchal/what-s-the-personality-of-kaggle-users

[2] https://blog.naver.com/gustn3964/221431714122

Table 3: Accuracies, balanced accuracies, and F1-scores of each model for each task.

	per post			per person			per person augmented		
Model	Acc	Bal	F1	Acc	Bal	F1	Acc	Bal	F1
ExtraTreesClassifier	0.2122	0.0625	0.021883	0.255908	0.088181	0.056466	0.239337	0.072031	0.039353
Naive Bayes	0.245424	0.111546	0.11778	0.390202	0.194249	0.199171	0.439481	0.300848	0.309502
Logistic Regression	0.119364	0.142816	0.098181	0.422478	0.375884	0.319072	0.45317	0.436509	0.36115
K-Nearest Neighbors	0.160154	0.077625	0.076013	0.105476	0.074717	0.053511	0.190634	0.077835	0.073779
XGBoost	0.249794	0.104068	0.103217	0.436888	0.266042	0.289925	0.49755	0.319425	0.355252
LSTM	0.266313	0.106608	0.102832	0.209919	0.068429	0.036696	0.444012	0.202051	0.193108
CNN	0.275097	0.126586	0.136285	0.444957	0.232622	0.245886	0.500288	0.328959	0.359224
BERT-base	**0.304212**	**0.174326**	**0.202375**	**0.5401**	**0.3609**	**0.3803**	**0.5736**	**0.488**	**0.5086**

256, five different filter sizes from one to five, and 100 filters for each size. We also used a dropout operation with 0.1 ratio for the training phase.

BERT Devlin et al. [6] introduced an innovative language model called BERT, which utilized the deep bidirectional training of the Transformer encoder. In the pre-training stage, the BERT learns co-occurrence relations among words by predicting the `[MASK]` tokens and identify the context of the overall data by the `[CLS]` token. However, because of the expense of pre-training, we utilize the pre-trained BERT model from the Transformers library[3], built by Huggingface. Using the pre-trained model, we fine-tune our task using the MBTI dataset, optimizing the model to classify 16 different personalities.

4 Results

4.1 Comparison of different methods and data

We conduct experiments on models described in the previous section and display results as shown in Table 4.1. Because our dataset is skewed, the vanilla accuracy may not be a good indicator of performance; therefore, we also evaluate each of our models on balanced accuracy and F1-score that were not evaluated in previous studies.

For classifying the personality of a post, all of the models perform poorly, and we believe that this is due to the insufficient information each post contains. Although there were many samples to train from, the models do not learn much from a small piece of text. On the other hand, most of the models have a notable improvement in the per-person task.

Regarding the augmented dataset, we observe another performance boost, and it is also worth noting that the balanced accuracies and F1-scores have generally improved across the table. This informs that augmented data was able to learn the feature representations effectively, as well as utilize the increased amount of training samples efficiently. Overall, the BERT model exhibits the highest scores among other models in any task, and this finding makes sense, as it exploits semantics of words from pre-training.

4.2 Analysis

We used integrated gradient (IG) method to find out which words or expressions represents the person's MBTI type well. IG aims to explain the relationship between a model's predictions in terms of its features. It considers a sentence with pad tokens as a baseline sentence. It interpolates a series of word tokens, increasing in intensity, between the baseline sentence and the original sentence. And then, IG tries to get the gradients of these interpolated sentences and approximate the gradients integral using the trapezoidal rule. Figure 1 shows an result of IG on CNN model. It enables to find and display words played an important role in classifying to MBTI type for a given text. IG could be computed as below:

$$IG_m = \frac{(x_m - x'_m)}{S} \sum_{S}^{k=0} \frac{\partial F(x' + \frac{k}{S}(x - x'))}{\partial x_m} \tag{1}$$

where S denotes the number of interpolation steps, and x_m denotes m-th word of given sentence, x' denotes baseline token, F(x) denotes the model result (i.e. forward result of CNN).

[3]https://github.com/huggingface/transformers

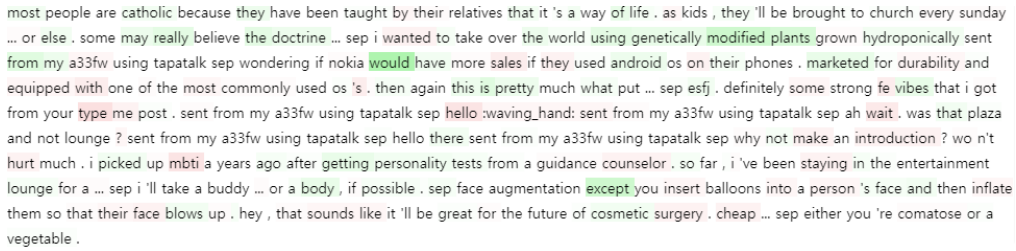

Figure 1: Example result of integrated gradient on CNN model. The posts were written by INTP type user. Words with green background indicate that they have positive influence on classifying the text to answer type, while the words with red background are opposite. Intensity of color means the amount of importance.

We assumed that the language or words used in each MBTI type would be slightly different, and this tendency was important information for classifiers to determine types. So we investigated words which play a crucial role in classifying each type for the whole data as well as each data point. We calculated importance of every word in the test data by applying IG. And then, we could get the top important words by summing the all the importance for each MBTI type. The table 4 shows top 20 important word for classifying type. We found that the words which describe MBTI type itself are top ranked. For example, 'enfj' is top ranked for ENTJ type and 'infj' is top ranked for INFJ type. This is convincing enough because people usually focused on their MBTI type and likely to write about their type. We also found that the words related 'feeling' such as 'feel', 'feels' are 'feelings' are usually highly ranked only for F types (which originally stand for 'feeling') like ENFJ, ENFP, INFJ, INFP and so on.

To see if there is a similar tendency in the posts of people who share many MBTI subtypes, we calculated the similarities between word importances of MBTI types. We compute similarities between one given MBTI type against four groups of types. The first group includes MBTI types which have three same subtypes. For eaxmple, if ISTP is given, ESTP/INTP/ISFP/ISTJ types have three shared subtypes against ISTP. The second group consists of MBTI types which have two same subtypes. The third one has only one shared subtype, and the last one has no shared subtype (i.e. ENFJ for ISTP). Table 5 shows the similarity between one arbitrary type and four groups. We found that the more subtypes the more they share, the higher the similarity they have.

Another thing to note here is, for per-person augmented dataset, rather than the difference between vanilla accuracy of CNN and Bert-base models, the gap between the balanced accuracy and f1 score of models tends to widen. We can assume this is caused by the distinct tokenizing and analysis methods of each model. Table 4 shows that the words meaning each MBTI personality type and their modified forms account for the highest importance both for normalized and unnormalized word importance results. This is possible with CNN model because each word itself is the input, while BERT tokenizer adopt subword tokenizer using BPE which cannot catch words like 'INFP' perfectly. Those MBTI types not normally appear in sentences so might be separated as the tokens. This distinction might make different tendency between two model, CNN classified 16types highly based on those words, but BERT model trained to learn structures of each personality types without explicit words reduce the gap between accuracy and balanced accuracy.

Next, we analyze the effectiveness and generality of our model by evaluating on completely different data: speeches from prominent historical figures. When taking MBTI tests, the tester is sometimes notified of famous people who have the same MBTI type. They may have a different personality type in reality; nevertheless, people consensually agree upon the most probable personality based on how the figure speaks. Therefore, when evaluating speeches, we referred to the Personality Database website[4]. Then, using an attention visualizing tool called *BertViz*, we observe which tokens give meaningful effects on the `[CLS]` token, which is used to classify our given task.

As shown in Figure 2, we see the top 20 tokens (excluding the `[CLS]` token) that attends the most to the `[CLS]` token. As they are from a speech of an INFJ figure, Elie Wiesel, the tokens themselves carry emotions, (e.g. *compassion, judged, cruelly, severely*), as well as hint at the speaker's imaginative

[4]https://www.personality-database.com/

Table 4: Top 20 important words for classifying MBTI types

	ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP
1	enfj	enfp	entj	entp	intps	esfp	estj	estp
2	!	!	entjs	entps	isfj	!	istj	estps
3	enfjs	enfps	an	the	welcome	?	an	fun
4	feel	an	intj	an	esfp	:d	enfp	an
5	an	feel	mbti	you	intp	typed	istp	istp
6	infj	i	.	?	m	intj	not	shit
7	all	fun	not	fun	from	her	all	got
8	really	:d	shit	sep	as	enfp	!	car
9	time	love	sep	shit	maybe	m	infp	type
10	their	lol	be	m	!	entp	by	:)
11	you	intjs	would	we	things	hello	entj	him
12	thank	fi	by	!	enfjs	but	as	entp
13	lol	thank	?	if	time	that	would	re
14	feelings	esfp	based	your	se	estp	their	as
15	are	maybe	/	use	will	by	entp	say
16	hello	and	what	see	but	you	be	except
17	is	went	will	fuck	i	need	?	lol
18	shy	ne	entp	maybe	fun	feel	will	of
19	i	here	does	how	we	in	isfj	can
20	s	feels	feel	are	an	never	going	just
	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
1	infj	infp	intj	intp	isfj	isfp	istj	istp
2	infjs	infps	intjs	intps	fe	!	istjs	istps
3	feel	feel	.	)	an	feel	happy	.
4	!	really	)	would	feel	really	you	,
5	my	feeling	?	the	!	)	posting	ll
6	-	and	not	is	would	music	mbti	really
7	you	!	by	,	like	m	the	shit
8	se	,	are	sep	entj	ll	rant	maybe
9	life	-	s	use	her	istp	being	got
10	fe	music	would	all	?	feelings	back	fuck
11	)	by	no	ti	from	esfp	fun	out
12	and	love	the	this	based	fun	like	the
13	of	my	,	.	maybe	welcome	do	fun
14	could	m	fi	could	enjoy	.	not	on
15	feels	this	world	to	went	like	enjoy	they
16	but	time	if	they	si	more	/	httpurl
17	feelings	maybe	female	female	on	infp	.	ti
18	really	but	do	god	beautiful	stuff	on	what
19	lol	fi	a	just	fellow	family	my	going
20	hello	life	use	an	for	infj	really	car

Table 5: Cosine similarity of word importances between types.

Num. of shared subtypes	avg. similarity	type examples (example for ISTP)
3	0.05919	ISTP vs. [ESTP, INTP, ISFP, ISTJ]
2	-0.02059	ISTP vs. [ENTP, ESFP, INFP, ESTJ, INTJ, ISFJ]
1	-0.04799	ISTP vs. [ENFP, ENTJ, ESFJ, INFJ]
0	-0.05033	ISTP vs. [ENFJ]

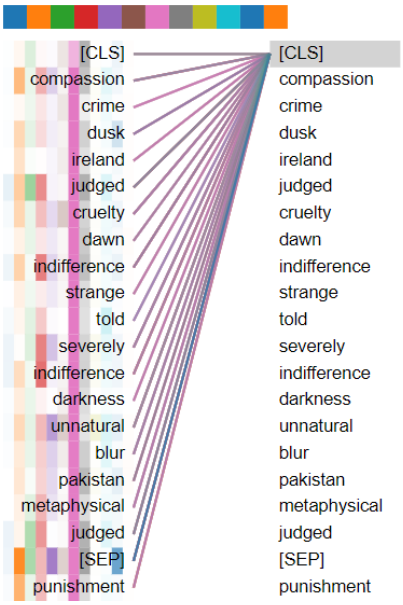

Figure 2: Top 20 tokens whose attention weights to the [CLS] token are the largest. Tokens are extracted from Elie Wiesel's speech *The Perils of Indifference*, who is thought to be an INFJ person. Each color represents one of the 12 heads from BERT, and the intensity shows how strong the weights are. Tokens are sorted from top to bottom in descending order.

perspective (e.g. *dusk, dawn, darkness, metaphysical*). Although this reasoning could be arguable, it is worth noting enough that such tokens are relatively weighted high when the model classifies a text to a certain personality type.

5 Conclusion

In this project, we explored various machine learning and deep learning models to predict one's MBTI type and the best model (BERT) achieved an accuracy of almost 60%. Furthermore, we analyzed language use corresponding to personality and showed that there are differences between some personalities. If more time to tune hyperparameters and larger and more balanced data are given, it might result in better performance. Using pre-trained language model for English Tweets [5] may also increase the model's performance. LIWC dictionary that defines each word to one or more categories is paid so we haven't used it in this project, but LIWC dictionary can be useful for finding correlations between language use and personality.

References

[1] Mohammad Hossein Amirhosseini and Hassan Kazemian. Machine learning approach to personality type prediction based on the myers-briggs type indicator ®. *Multimodal Technologies and Interaction*, 4, 03 2020.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] S. Bharadwaj, S. Sridhar, R. Choudhary, and R. Srinath. Persona traits identification based on myers-briggs type indicator(mbti) - a text classification approach. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1076–1082, 2018.

[4] Brandon Cui and Calvin Qi. Survey analysis of machine learning methods for natural language processing for mbti personality type prediction, 2017.

[5] Thanh Vu Dat Quoc Nguyen and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. *arXiv preprint*, arXiv:2005.10200, 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Sedrick Scott Keh and I-Tsun Cheng. Myers-briggs personality classification and personality-specific language generation using pre-trained language models, 07 2019.

[8] Chang H. Lee, Kyungil Kim, Seok Seo Young, and Cindy K. Chung. The relations between personality and language use. *Journal of General Psychology*, 134(4):405–413, October 2007.

[9] Kim Luyckx and Walter Daelemans. Personae: a corpus for author and personality prediction from text. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[10] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

Appendix

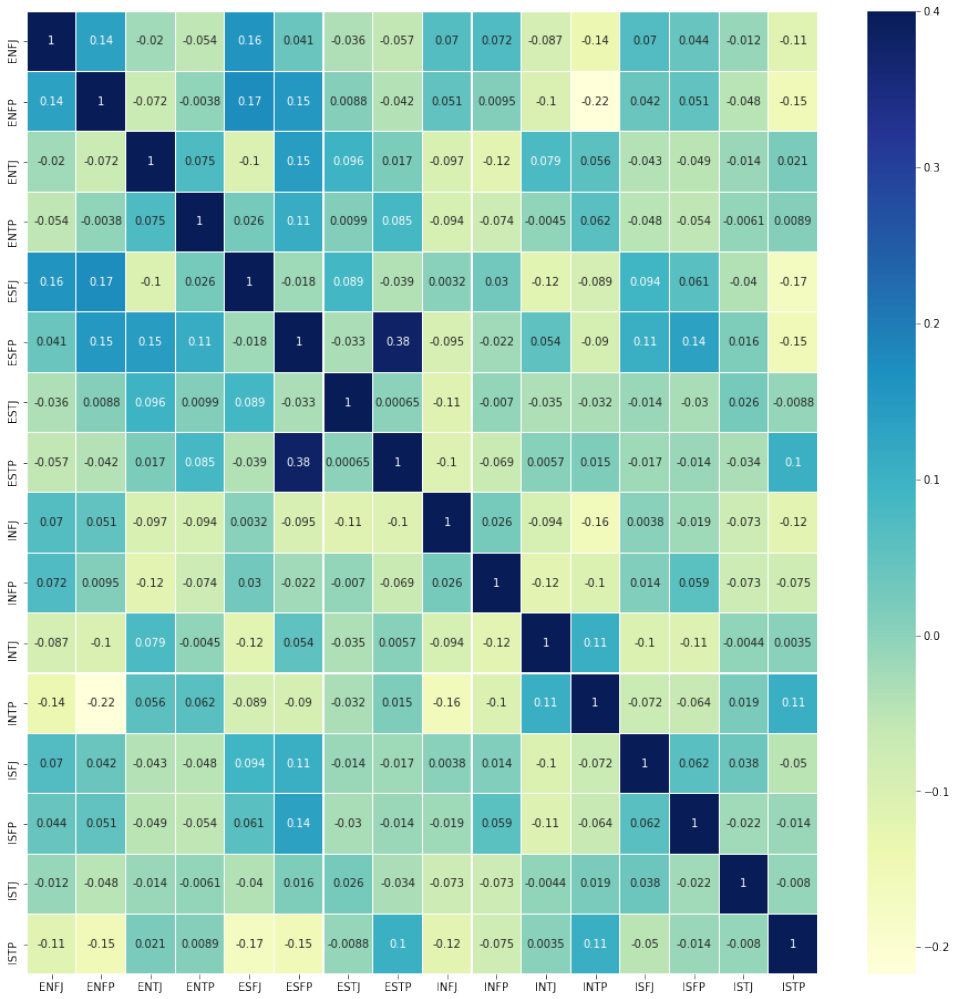

Figure 3: Similarity matrix between MBTI types. This figure shows the consine similarities between word importance vector of each type.