Personality detection from written text

Aniket Pramanick
SR No. 14686
M.Tech(CSE)
Computer Science
aniketp@iisc

Rishi Hazra
SR No. 14542
M.Tech(SE)
System Science
rishihazra@iisc

Sayambhu Sen
SR No. 14351
M.Tech(SE)
System Science
sayambhusen@iisc

Abstract

Personality is a defining aspect of a person. It defines the person's thinking habits, actions, ideas, behaviors as well as reactions to various external and internal stimuli. In this document, we will be exploring various ways of making a judgement about a person's personality based on the written text. Specifically, we will be working on predicting a label for one of the 16 different personality types based on labeled data available at Kaggle. It has various applications like ad targeting based on a person's personality as well as forensic narrowing of suspects.

1 Introduction

Personality is defined as the set of habitual behaviors, cognitions and emotional patterns that evolve from biological and environmental factors.[1] It is an aspect of a person that describes a wide range of behaviours and one can gain a lot of information about a person from an understanding about his personality. In fact, some aspects of a person's personality are inborn while some are acquired through the external environment in which he grew up. [2] Also, it is not a static concept since studies indicate that experience does indeed change a person's personality throughout his lifetime. Certain personality aspects are crucial in deciding many aspects of a person including career choice, relations with friends and family. In fact, often, in many companies, employers look for skills coupled with a certain personality aspect. Indeed skills are not the only factor in deciding a person's employability towards a certain job. In the case of e-marketing, current knowledge about a person's search history are the main things used in targeted ads. In community and social network sites like Facebook, twitter, Quora, stackexchange etc. , users are shown ads based on their clicking behaviors or their search histories. Keywords and hash tags are also often indicative of a person's wants and needs. However, not much is often gauaged from the written text that some people post on such community sites. [3] Not just people, but companies try to build a brand personality as well which decides which market the company is trying to cater to. This can decide what the company decides to use as tag lines or branding.[4]

2 Motivation

Recent reports have revealed how **Cambridge Analytica**, a UK based company and led, at the time, by candidate Donald Trumps key adviser, used psychographic data from Facebook profile to target American voters in the months before the 2016 presidential election with **personalized political messages and influence their voting behavior**.

Many psychologists and social scientists have made their careers analyzing ways to predict our personality and ideologies by asking simple questions. These questions, like the ones used in social media quizzes, do not appear to have obvious connections to politics. Even a decision like, which web browser we are using to read this article, is filled with clues about our personality.

Cambridge Analytica compiled a **shopping list of traits**, that could be predicted about voters by building large psychographic profiles on a national scale. They were then able to create websites, ads and blogs that would attract Facebook users and encourage them to spread the word. *They see it...they click it...they go down the rabbit hole.*

3 Myers-Briggs

The Myers-Briggs Type Indicator(MBTI) is based on Carl Jungs theory of psychological types which states that random variation in behavior is accounted for by the way people use judgement and perception. It is believed to measure the psychological preferences of people and assess the way in which a person perceives and reacts to their environment.There are 16 personality types in 4 different dimensions. The 16 types are typically referred to by an abbreviation of four letter initial letters of each of their four type preferences.

1. E, I : Extroversion, Introversion
2. S, N : Sensing, Intuition
3. T, F : Thinking, Feeling
4. J, P : Judging, Perceiving

Extroversion (E) vs Introversion (I) is a measure of the extent to which a person is ready to interact or reflect. Sensing (S) vs Intuition (N) is a measure of how new information is interpreted by an individual (whether the person trusts their five senses and concrete facts or are instinctive and consider different possibilities) Thinking (T) vs Feeling (F) is a measure of rational decision making ability of a person (whether a person takes logical decisions or considers the needs of the people involved) Judging (J) vs Perceiving (P) is a measure of flexibility and adaptability of a person (whether a person likes to approach situations in a structured manner or is open to choices and adaptable)

MBTI	Archetype	Traits
INTJ	Architect	Imaginative and strategic thinkers
INTP	Logician	Innovative inventors with a thirst for knowledge
ENTJ	Commander	Bold, imaginative and strong-willed
ENTP	Debater	Smart and curious thinkers
INFJ	Advocate	Quiet and mystical
INFP	Mediator	Poetic, kind and altruistic
ENFJ	Protagonist	Innovative inventors with a thirst for knowledge
ENFP	Campaigner	Enthusiastic, creative and sociable
ISTJ	Logistician	Practical and fact-minded
INFJ	Defender	Very dedicated and warm
ESTJ	Executive	Excellent administrators
ESFJ	Consul	Extraordinary caring and social
ISTP	Virtuoso	Bold and practical experimenters
ISFP	Adventurer	Flexible and charming artists
ESTP	Entrepreneur	Smart, energetic and perceptive
ESFP	Entertainer	Spontaneous, energetic , enthusiastic

Table 1: Personality types

3.1 Prior Work

The main difficulty in trying to find a technique to detect personality is not the fact that current machine learning algorithms are not powerful enough. Rather it has to do with the fact that the relation between written text and personality is not clear at all. While many heuristics have been used for detection not too many algorithms have been used for detection. The age old methods used by psychologists include filling a questionnaire or showing certain pictures and sometimes even analysis from handwriting for predicting about a personality.

Indeed many current algorithms get the best predictions from using questionnaire based data.[7] Being able to extract text data has the significant advantage of being scalable since, any sort of questionnaire is a constrained environment [5]. Text data can be mined tremendously from social networking sites instead which contain a lot more information and can be obtained more easily. Of course, labeling of such data is difficult because of which we use a the *Kaggle* dataset for obtaining the texts with labels[7].

Word level embeddings along with document level embeddings have already been used for understanding the way in which text structure is used. Indeed choice of words along with structure of text are some of the main factors which determine personality. Document level features are also

indicative of personality like document length, average size of sentences [8]. Some other features like location of words in a sentence are also indicative of personality of a person. For example , and extrovert may say "I think it is possible" , while an introvert may say "It is possible, I think" The placement of a word may be indicative of a personality more than the word itself. Thus, not only must the presence of a feature be taken into account but it's temporal position as well. Thus, algorithms must take into account temporal behaviors as well. Simple temporal information is present in n-gram data of written text but this model can incorporate at best 5 to 10 words.

Indeed this has been tried out using LSTMs and seen to give reasonably accurate outputs of about 37 percent[5]. While deep learning methods do use time as as a feature for learning temporal features[8] [6], those methods have some repetition in the data used. In the case of text data, the biggest problem is that even in the case of using word embeddings there is hardly any continuity between the data and time. The representation of time in text documents can be either of the form of position in sentence or position in paragraph or position in document itself. Indeed incorporation of temporal features itself is itself quite a challenging aspect worth exploring[10].

4 Dataset and Problem Description

In this work , we are using the *Facebook* data as our dataset. 7 The good thing about our dataset is that all of these users have confirmed personality labels. The dataset consists of two columns, type and posts. The 'type' column consists of the confirmed personality type labels of the users of which each row consists of a user. The 'posts' column is such that each row is a continuous body of text consisting of the last 50 posts of the user corresponding to a personality label in type. The posts are separated by the '| | |' symbol.

Now, we have some choices about our model of classifier. We can make it such that it is a word level classifier and uses features such as bag of words or it can use the same words but it will be input as a sequence like in RNNs or LSTMs. We will show why exactly, word level features like bag of words or tfidf are not a good candidate solution for our model. Our task is to look at this body of text in a certain manner such that the features can be used classify this to among the 16 different personality types. We can do this in two ways.

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

Table 2: Overview of data

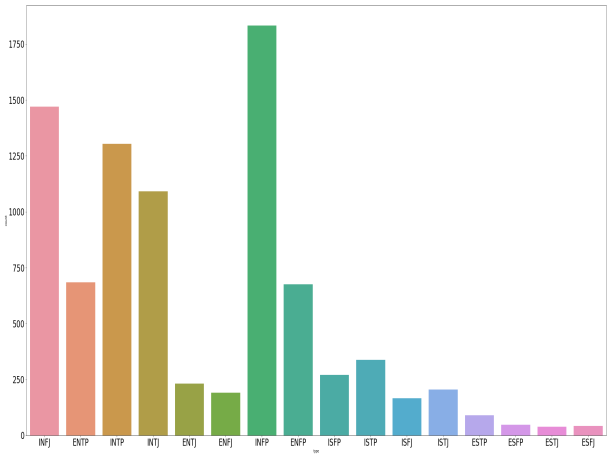

Figure 1: Label Distribution of dataset

Either we can provide a one hot encoding from among the 16 different personality tasks or we can predict one of the 4 dimensions of the output such that each of the four positions represent a separate label. For *E.g.*, we can have 'ENTJ' to be equal to '1111' or we can have 'INTJ' equal to '0111'. In this sort of labelling, we are more robust to errors as part of the labels being correct will be less penalized for *e.g.* 0111 is quite close to 1111. On the other hand, a single one hot vector label will be used will be such that even a single label misclassified will be marked as wrong classification and penalized completely.

4.1 Preprocessing

1. Converting all words to lower case
2. Removing all urls and replacing them with the word 'link'
3. Removing all '| | |' and replacing them with

space

4. Removing all punctuation
5. Removing all white spaces, line breaks etc.
6. Removing all repeated characters.
7. Lemmatizing all words.
8. Using max matching algorithm to get as many single frequency words to separate recognizable words. This was a very crucial step as it helped in drastically

All these different types of preprocessing helped in reducing the word vocabulary size from 1,72,900 to 40,000.

For the word level model, **parts of speech (POS) tagging** which is a basic form of syntactic analysis, is done on the data. We use *nltk* pos-tagger. Use of POS taggers gives us a lot of insight into the data and helps us model the sequence in a better way by acting as a attention feature in our **Attention model**. It has been found that people who use common nouns more often in their language tend to be in **E**xtroversion, **T**hinking or **J**udging type 2, 5. Introverted people use more pronouns but less common nouns. Interjection, which includes *lol*, *haha*, *FTW*, *yea*, is more likely to be used by people who are in Sensing and Perceiving type. *Emoticon* is more likely to be used by people who are in **S**ensing and **F**eeling type while numbers are more likely to be used by people who are in **S**ensing and **T**hinking type 3. Those seemingly random online behaviors, such as use of pronouns, *emoticons*, and nouns can be somehow explained by the psychological traits of the users 4.

Figure 2: Extroversion(E) vs Introversion(I)

4.2 Word level LSTM model

The word level LSTM model takes in as input, a sequence of words, and gives as output, a corresponding label. Thus, it is an acceptor model of

Figure 3: Sensation(S) vs Intuition(I)

Figure 4: Thinking(T) vs Feeling(F)

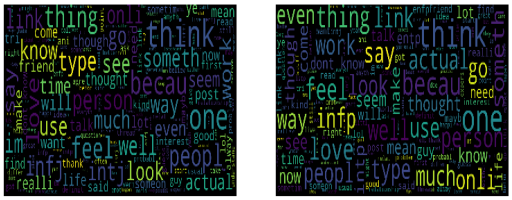

Figure 5: Judgement(J) vs Perception(P)

sequence neural network. The model is as shown below. We have made it in such a way that the input sequence is of length 500. Each of the words are such that they have an index corresponding to a word in the dictionary. An embedding layer is used in the first layer such that it is a large horizontal input which corresponds to a one hot input. The output is of a variable size 'embedding size'. Thus, the embedding size corresponds to the size of the word embedding. Essentially, we are learning the embedding for the task itself.

4.3 Character level

The preprocessing for character level model is slightly different. We used a *regex* type parser to only keep the website's name instead of all the extra writing. E.g. 'http://www.youtube.com/watch?v=qsXHcwe3krw' was replaced by 'http://www.youtube.com/'. Thus, it starts looking for http and ends till the first /.

Character level models have the advantage that they can capture even finer information like spelling mistakes, repeated characters, use of con-

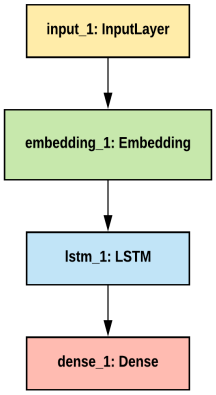

Figure 6: LSTM model (word and character level)

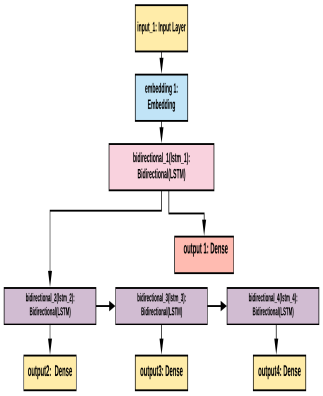

Figure 7: LSTM model (word and character level) stacked deep model

nected words, length of sentences etc. Thus, for deeper models, we have that this can capture a higher level of information. [**?**] Here also we are using an embedding layer. The character level is such that the size of the character dictionary is very small and hence, we have a much better generalization ability with the deep neural network.

We thus also implement a deeper connected model which can capture a higher level of information.

5 Implemented models

We have implemented the word level LSTM model and the character level LSTM model. We have implemented both the normal and the deep level stacked model. The deep model has been made in such a way that the lower levels predict one of the labels while the upper labels predict the upper labels progressively. In fact, prediction of the upper labels aid in the prediction of the lower labels, similar to how we can have learning POS tags can aid in learning the NER tags.

5.1 Novelty: Attention Model with POS tagging

The proposed model uses LSTM as the Basic Classifier. The architecture of the modeled Network is given in Figure 11.

The first LSTM is for obtaining a representation for each of the posts. The sequence of POS tags of each of the words in each of the posts is used to train the *Attention Weights* from the second LSTM. Then using a *BiLSTM* is used to predict *Personality Labels*. In the *BiLSTM* concatenated outputs from the two LSTMs are used along

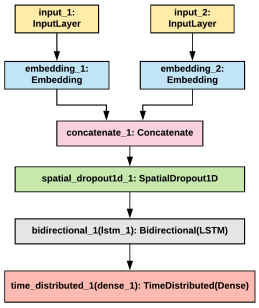

Figure 8: Attention Model

with a Softmax at the output and tanh at all other hidden layers.

The *BiLSTM* is fed with the concatenated outputs of the other two LSTMs and a soft-max function for classification at the output layer. In all other hidden layer tanh activation is used.

Model Implementation: Let *S* be the set of posts and *Y* be the set of unique personality types. A post s_k is essentially an ordered sequence of words.

$$s_k = w_{k1}w_{k2}w_{k3}...w_{k1_k}$$

where l_k is the length of s_k. The features of a word forms a word vector v^{w_i} of dimension 25.

Corresponding to each post, we get a sequence of POS tags

$$pos_k = p_{k1}p_{k2}p_{k3}...p_{k1_k}$$

The features of a POS tag forms a vector t^{p_i} of dimension 5.

The vectors pos_s and s_k are concatenated to form a representation r_k. The final *BiLSTM*

projects r_k onto the target space Y of classes through a soft-max layer.

A LSTM has five types of gates in each node i represented by five vectors including an input gate i_i, a forget gate f_i, an output gate o_i and a candidate memory cell gate c_i. Values are updated or forgot using memory cell gates.

The memory cell c_i is updated as

$$c_i = f_i \odot c_{i-1} + i_i \odot c_i$$

The hidden state is obtained as

$$h_{w_i} = o_i tanh(f_i \odot c_i)$$

The forget gate f_i keeps the log term memory.

The POS sequence tag information for weight training. The attention weight α_{w_i} for each hidden layer is calculated as

$$\alpha_{w_i} = \frac{exp(f(h_{w_i}, p_i))}{\sum_{i=1}^{l_i} exp(f(h_{w_i}, p_i))}$$

Where f in a score function to depict the importance of a word.

Formally, the concatenated representation s_k is a weighted sum of hidden states, given as

$$s_k = \sum_i \alpha_{w_i} h_{w_i}$$

. In the final layer of LSTM d_k is the projection of outputs given as

$$d_k = tanh(\hat{W}_h(s_k \oplus w_i) + \hat{b}_k)$$

Finally, prediction for any label y ϵ Y is predicted as

$$P(y|d_k) = \frac{e^{d_k^T W_y}}{\sum_{l=1}^{Y} e^{d_k^T W_y}}$$

Where W_l is the soft-max weight labels.

5.2 Excluding failed trials

We tried performing truncated SVD by reducing the data distribution along the axis having the highest variance. In order to visualize the high dimensional data, t-distributed Stochastic Neighbor Embedding is also used. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. The clustering in the Figure 10 is not proper, which tells us that projecting the data into lower dimension is not useful and we must design our model for high dimensional data.

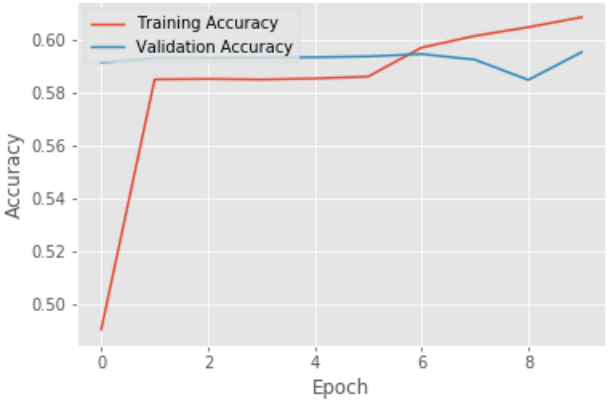

Figure 9: Accuracy vs. Epoch Graph

1. Using SVM and xgboost on tfidf reduced vectors : Now, the above datasets are made using bag of words or tfidf vectorizers. As can be seen in the pictures, the datasets are highly intermingled even in the highest variance directions using such features. We are getting the tfidf vectors of the posts after using a fit transform. Then on these vectors, we are using svd fit transform. As can be seen, we see that even in the highest variance direction, the data is highly inseparable. Thus, SVM may not be very successful in separating such data. In fact, we have also tried to use the kaggle famed XGBoost to model it but it has been seen to get similar 25 %.

2. Using Word Level LSTM models : We have done the preprocessing as described above and then trained the model for about 5 epochs. Now, we can train the model to have outputs of the form of a 16 bit one hot vector or a 4 dimensional Binary vector. As it is we find that the model for the word level becomes inaccurate for very large embedding sizes due to the large size of the vocabulary. In fact, while training accuracy becomes very large of the form of 80% the validation accuracy is very low of 20%. However, we find that the accuracy for the outputs of 4 dimensional outputs are slightly higher of the form of 70%.

3. Using Character Level LSTM models : We have also made the model such that it accepts character sequences. However, in this case, there is a problem that the sequence length in the acceptor model is much larger. Thus,

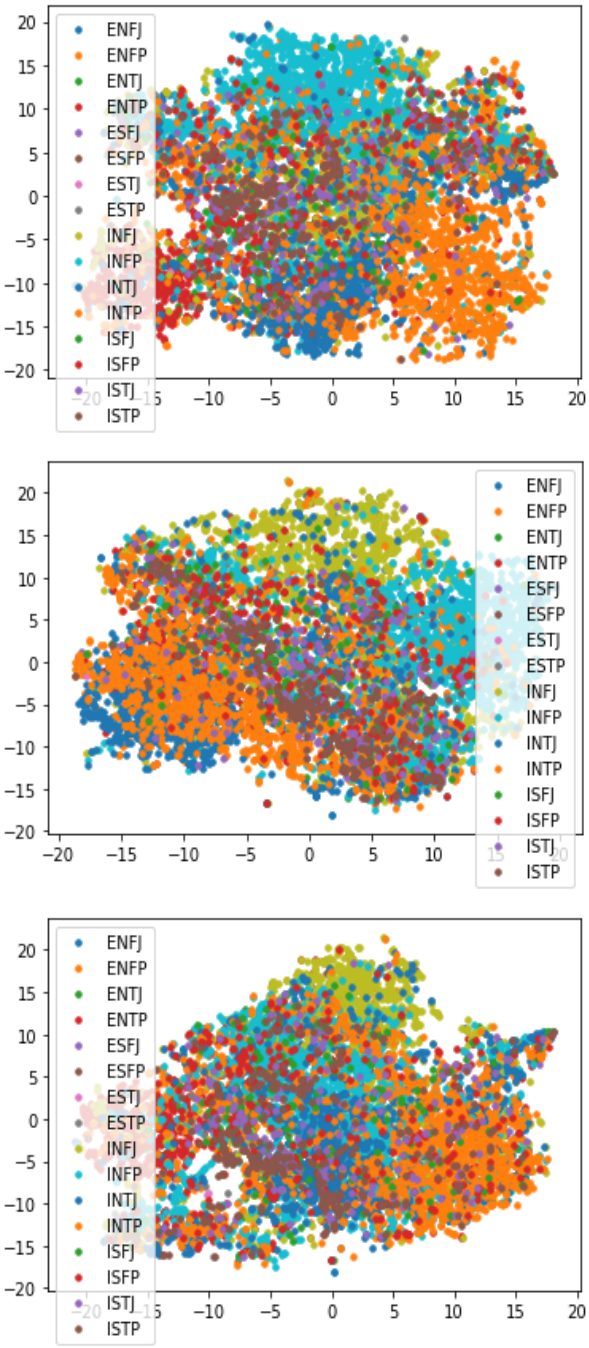

Figure 10: TSNE scores along highest variance

there is a problem of vanishing gradients in such a model despite the use of LSTMs. In fact, there is a problem of having such models taking way too much time to train and so we can only test for smaller test sizes. However, the 16 bit outputs are still of the order of less than 40 %.

4. Using Stacked LSTM models : We have tried to use deep stacked models for LSTM in word and character levels but even here we found the model to be too heavy for character model. However, for smaller data sizes , we found that the accuracy for the 4 dimensional model was quite good of the order of 75 to 85% for each of the labels. However, in each of the cases, we found that the high accuracy was because some labels being very high in the training set due to which the model was biased towards getting that output. However, for larger sizes of the model and larger training sets, we see that such a model has good potential since it captures the interdependencies between the labels quite well. In fact, we found different accuracies for the labels once the target positions were changed.

6 Results

	E	**I**		**N**	**S**
E	182	200	N	889	600
I	522	831	S	133	113

Table 3: SVD of E vs I and N vs S

	T	**F**		**J**	**P**
T	377	421	J	332	337
F	550	387	P	487	579

Table 4: SVD T vs F and J vs P

	E	**I**		**N**	**S**
E	224	158	N	755	734
I	652	701	S	62	184

Table 5: LSTM (word) E vs I and N and S

	T	**F**		**J**	**P**
T	544	254	J	427	242
F	413	524	P	629	437

Table 6: LSTM (word) T vs F and J vs P

	E	**I**		**N**	**S**
E	201	181	N	1480	9
I	454	899	S	244	2

Table 7: LSTM (stacked word) E vs I and N and S

	T	F
T	575	223
F	285	652

	J	P
J	528	141
P	334	732

Table 8: LSTM (stacked word) T vs F and J vs P

	ENTJ	INTJ	ESTJ	ISTJ	ENFJ	INFJ	ESFJ	ISFJ	ENTP	INTP	ESTP	ISTP	ENFP	INFP	ESFP	ISFP
ENTJ	0	5	0	0	0	0	2	0	0	0	0	0	0	0	0	0
INTJ	0	117	1	4	0	6	0	4	0	3	0	0	0	0	0	0
ESTJ	3	0	27	0	0	0	3	0	1	4	0	1	0	3	2	0
ISTJ	4	3		191	0	0	0	55	0	5	0	23	0	1	8	3
ENFJ	0	1	1	0	0	0	1	0	0	0	2	0	1	0	0	2
INFJ	0	6	0	3	0	24	0	0	2	0	0	2	3	0	1	0
ESFJ	3	1	11	0	0	0	32	0	0	2	0	0	1	4	0	13
ISFJ	11	0	23	0	2	2	0	122	0	0	7	0	0	13	0	13
ENTP	20	0	12	0	4	0	8	0	56	0	0	0	14	0	0	11
INTP	0	1	0	2	3	0	2	0	0	33	0	0	1	0	2	0
ESTP	0	5	0	2	0	1	6	0	0	0	22	0	0	0	9	0
ISTP	21	0	12	0	14	0	1	23	14	0	20	208	23	0	23	11
ENFP	2	0	4	0	1	0	0	3	2	0	1	0	0	0	0	2
INFP	0	1	0	3	0	0	2	1	0	0	0	0	0	0	0	0
ESFP	0	0	1	0	0	0	0	0	1	2	0	1	0	3	42	3
ISFP	17	0	0	16	0	34	5	0	2	0	0	1	13	0	3	197

Figure 11: Confusion matrix attention

7 Conclusion

Following our extensive study on our models we can see that attention played quite a vital role in our prediction. This can be attributed to the fact that our model has a very long sequence consisting of words or characters. Even LSTM cells which have memory cannot capture sufficient information due to such long sequences especially since we see that it is an acceptor model. Attention models can help in understanding vital parts of our sequence. Also, the level of preprocessing plays an extremely important role here since the level of preprocessing can contribute to huge changes in accuracy.

However, it can be seen that character level models while being longer can capture even more information about the personality type since it takes into account spelling mistakes as well. Currently, due to our limitations of time and resources we could not build the character level classification model but we have observed sufficiently good outputs.

Several ideas can be used as future direction for our research.

1. Use CRF since, then even in the case skewed labels in training data we get results like the ones we have.
2. Use CNN to capture local information in the sequence. Right now we did not have the time to implement the CNN, but most current research methods point towards the use of CNN in concatenation with Mairesse features as a good method for personality classification.
3. Use CNN for feature extraction followed by sequence modelling with the help of LSTM.
4. Further exploration of character level deep modelling methods.

References

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

[2] www.wikipedia.com/personality detection.

[3] Subir Bandyopadhyay. *Contemporary Research in E-Branding*. Indiana University Northwest, USA.

[4] Dusan Bosnjakovic. How can machine learning be applied to temporal data?

[5] Navonil Majumder et. el. *Deep Learning-Based Document Modeling for Personality Detection from Text*. Stanford University.

[6] Paul Griffiths. The distinction between innate and acquired characteristics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

[7] Nathan O. Hodas, Ryan Butner, and Courtney Corley. How a user's personality influences content engagement in social media. *CoRR*, abs/1609.00108, 2016.

[8] Anthony Ma. Neural networks in predicting myers brigg personality type from writing style. 2017.

[9] Krikor B. Ozanya Patricia Scully. Deep neural networks for learning spatio-temporal features from tomography sensors.