Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Midterm Report

on

“Psychological Profiling Using NLP and Machine Learning”

[Code no: COMP 488 - Neural Network and Deep Learning]

Submitted By:

Krishna Lamsal (27)

Group: CS (2020)

Submitted to:

Professor. Dr. Bal Krishna Bal

Department of Computer Science and Engineering

Submission Date:

25th December 2024

Abstract

This project aims to predict individuals' MBTI personality types based on their written posts using advanced machine learning techniques. Utilizing deep learning models such as LSTM and BERT, the study explores the potential of automated personality type classification across four primary axes: Introversion-Extraversion, Sensing-Intuition, Thinking-Feeling, and Judging-Perceiving. Psychological profiling with AI offers a unique opportunity to evaluate personality frameworks like the MBTI and their ability to predict language styles and behavior online. Evaluation results highlight the effectiveness of these models in handling textual data for personality classification.

Keywords: MBTI, Deep Learning, LSTM, BERT

Table of Contents

List of Figures

List of Tables

Acronyms/Abbreviations

MBTI	Myers-Briggs Type Indicator
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers

Chapter 1: Introduction

1.1 Background

Personality classification has long been a topic of interest in psychology and social sciences. The Myers-Briggs Type Indicator (MBTI) is a popular framework for categorizing individuals into 16 distinct personality types based on preferences across four dichotomies:

- Introversion (I) vs. Extraversion (E): Focus of energy and attention.
- Sensing (S) vs. Intuition (N): Information processing style.
- Thinking (T) vs. Feeling (F): Decision-making preference.
- Judging (J) vs. Perceiving (P): Approach to structure and organization.

These preferences combine to create 16 personality types: ISTJ, ISFJ, INFJ, INTJ, ISTP, ISFP, INFP, INTP, ESTP, ESFP, ENFP, ENTP, ESTJ, ESFJ, ENFJ, and ENTJ. Each type reflects unique behavioral tendencies and cognitive styles.

With the proliferation of online platforms, vast amounts of textual data are now available, making it possible to apply machine learning methods to infer personality traits from written language. Leveraging textual data for personality classification offers insights for applications in recruitment, targeted marketing, and mental health support.

Deep learning techniques, particularly recurrent neural networks like LSTM and transformer models like BERT, have demonstrated remarkable performance in natural language processing tasks. This project investigates the effectiveness of these models in predicting MBTI personality types, providing a comparative analysis of their strengths and limitations.

1.2 Objectives

- To preprocess and analyze a dataset of user posts tagged with MBTI personality types.
- To implement LSTM and BERT models for personality type classification.
- To evaluate and compare the performance of these models on accuracy and other relevant metrics.
- To provide insights into the potential and limitations of machine learning approaches for personality classification.

1.3 Motivation and Significance

Understanding human personality through text has profound implications for both individuals and organizations. Automated personality classification can enhance user experience in digital platforms by enabling personalized content and recommendations. In recruitment, it can assist in matching candidates with roles suited to their personality traits. Furthermore, mental health professionals can utilize such tools for initial assessments, identifying individuals who may benefit from counseling or support.

The primary motivation for this project stems from a deep interest in psychology and the desire to integrate deep learning into this field. This project bridges the gap between psychology and artificial intelligence, exploring the synergy between linguistic patterns and personality traits. By leveraging state-of-the-art deep learning models, it aims to advance the field of computational psychology and offer practical tools for various domains.

Chapter 2: Literature Review/Related Works

Personality classification has evolved from traditional psychometric assessments to data-driven approaches leveraging machine learning. Early studies focused on feature extraction techniques, such as word frequencies and syntactic structures, to infer personality traits from text. For example, Pennebaker et al. (2003) [1] introduced the Linguistic Inquiry and Word Count (LIWC) tool, which correlates linguistic features with psychological states and personality dimensions.

With the advent of deep learning, models like LSTM have been employed to capture sequential dependencies in text. Xu et al. (2018) [2] demonstrated the application of LSTM for MBTI personality prediction, achieving significant improvements over traditional machine learning approaches. The introduction of transformer models like BERT further revolutionized natural language processing, enabling models to understand context and relationships in text more effectively. Sun et al. (2019) [3] showed that fine-tuning BERT for personality classification tasks yields superior results compared to LSTM.

Despite these advancements, challenges remain. The MBTI framework itself faces criticism for its binary categorizations and lack of empirical validation. Moreover, imbalanced datasets and the ambiguity of language can impact model performance. This project builds upon previous work by addressing these challenges through robust preprocessing techniques and comparative analysis of LSTM and BERT models.

Chapter 3: Procedure and Methods

3.1 Dataset Acquisition and Preprocessing

The dataset used for this study was sourced from Kaggle and consists of two columns: "Type" (MBTI types) and "Posts." The "Posts" column contains text written by individuals from an online forum dedicated to personality discussions. These posts provide insights into the language style and behavior of individuals with specific MBTI types.

The preprocessing steps include:

- Removing unwanted characters, spaces, links, and explicit mentions of MBTI types.
- Tokenizing the text and removing stop words using the NLTK library.
- Lemmatizing words to their base forms to reduce linguistic variability.
- Splitting the "Type" column into four binary axes: Introversion/Extraversion (I/E), Sensing/Intuition (S/N), Thinking/Feeling (T/F), and Judging/Perceiving (J/P).

	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...

Figure 3.1.1. Original Dataset

	IE	NS	TF	JP	cleaned_text
0	1	1	0	1	moment sportscenter top ten play prankswhat li...
1	0	1	1	0	finding lack post alarming sex boring position...
2	1	1	1	0	good one course say know blessing curse absolu...
3	1	1	1	1	dear enjoyed conversation day esoteric gabbing...
4	0	1	1	1	fired another silly misconception approaching ...

Figure 3.1.2. Processed Dataset

3.2 Model Architecture and Training

3.2.1 LSTM Model

The LSTM model captures sequential dependencies in text, making it suitable for analyzing language data. Bidirectional LSTM layers were used to process textual input in both forward and backward directions, allowing the model to understand contextual relationships comprehensively. The architecture incorporated dropout layers for regularization, preventing overfitting by deactivating neurons randomly during training. This ensured the model could generalize well to unseen data.

The embedding layer transformed the tokenized text into dense vector representations, which were then passed through multiple LSTM layers. These layers handled the sequential nature of the data, identifying patterns indicative of each MBTI personality axis. Dense layers with sigmoid activations were used at the output, producing binary classifications for each of the four personality axes.

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_text (InputLayer)	(None, 1)	0	-
text_vectorization… (TextVectorization)	(None, 250)	0	input_text[0][0]
embedding (Embedding)	(None, 250, 128)	1,280,000	text_vectorizati…
bidirectional (Bidirectional)	(None, 250, 400)	526,400	embedding[0][0]
dropout (Dropout)	(None, 250, 400)	0	bidirectional[0]…
bidirectional_1 (Bidirectional)	(None, 64)	110,848	dropout[0][0]
dropout_1 (Dropout)	(None, 64)	0	bidirectional_1[…
dense (Dense)	(None, 20)	1,300	dropout_1[0][0]
IE_output (Dense)	(None, 1)	21	dense[0][0]
NS_output (Dense)	(None, 1)	21	dense[0][0]
TF_output (Dense)	(None, 1)	21	dense[0][0]
JP_output (Dense)	(None, 1)	21	dense[0][0]

Total params: 1,918,632 (7.32 MB)
Trainable params: 1,918,632 (7.32 MB)
Non-trainable params: 0 (0.00 B)

Figure 3.2.1. Model-1 architecture

3.2.2 BERT Model

BERT, a transformer-based architecture, was selected for its state-of-the-art performance in natural language understanding tasks. BERT's contextual embeddings and its ability to understand intricate language patterns provided a significant advantage over traditional models. Unlike LSTM, BERT employs self-attention mechanisms to capture contextual information from the entire text simultaneously, making it highly effective for understanding complex sentence structures.

The pretrained BERT model was fine-tuned on the MBTI dataset. Input text was processed into token IDs, attention masks, and token type IDs, which BERT uses to encode contextual embeddings. The CLS token's output, representing the entire input sequence, was passed through a dropout layer for regularization before being classified into the four personality axes using dense layers.

Model: "functional_2"

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	(None, 128)	0	-
attention_mask (InputLayer)	(None, 128)	0	-
token_type_ids (InputLayer)	(None, 128)	0	-
bert_layer_1 (BertLayer)	(None, 128, 1024)	0	input_word_ids[0… attention_mask[0… token_type_ids[0…
get_item_1 (GetItem)	(None, 1024)	0	bert_layer_1[0][…
dropout_3 (Dropout)	(None, 1024)	0	get_item_1[0][0]
IE_output (Dense)	(None, 1)	1,025	dropout_3[0][0]
NS_output (Dense)	(None, 1)	1,025	dropout_3[0][0]
TF_output (Dense)	(None, 1)	1,025	dropout_3[0][0]
JP_output (Dense)	(None, 1)	1,025	dropout_3[0][0]

Total params: 4,100 (16.02 KB)
Trainable params: 4,100 (16.02 KB)
Non-trainable params: 0 (0.00 B)

Figure 3.2.2. Model-2 Architecture

3.3 Evaluation

Both models are evaluated on a test dataset to measure their accuracy and assess their ability to predict MBTI personality types. Comparative results and key findings will be discussed in subsequent chapters.

Chapter 4: Results and Findings

The models were evaluated based on their ability to classify MBTI personality types along the four axes:

- Introversion (I) vs. Extraversion (E)
- Judging (J) vs. Perceiving (P)
- Sensing (S) vs. Intuition (N)
- Thinking (T) vs. Feeling (F)

The evaluation focused on the following metrics:

- **Accuracy**: The proportion of correct predictions to the total number of predictions.
- **Loss**: The model's objective loss function value, indicating how well the predictions align with actual labels.

4.1 Results for LSTM Model

The following table summarizes the performance of the LSTM model:

Axis	Accuracy	Loss
Introversion-Extraversion (I/E)	78.28%	0.5243
Judging-Perceiving (J/P)	60.31%	0.6667
Sensing-Intuition (S/N)	85.76%	0.4122
Thinking-Feeling (T/F)	55.44%	0.6892
Overall Loss	-	2.2925

Table 4.1.1. Result of LSTM Model

Introversion-Extraversion (I/E):

The model achieved a strong accuracy of 78.28% for the I/E axis, indicating it effectively captures differences in language use between introverted and extroverted individuals.

A lower loss (0.5243) supports the conclusion that the model's predictions are stable and accurate for this axis.

Judging-Perceiving (J/P):

Accuracy was moderate at 60.31%, with a relatively higher loss of 0.6667. This suggests the language features distinguishing J/P are less pronounced, potentially due to overlap in linguistic patterns.

Sensing-Intuition (S/N):

The model performed exceptionally well on this axis, achieving the highest accuracy of 85.76% and the lowest loss of 0.4122. This result reflects the model's ability to distinguish between Sensing and Intuition effectively based on textual cues.

Thinking-Feeling (T/F):

Accuracy was lower at 55.44%, with a loss of 0.6892. This axis proved more challenging for the model, possibly due to subtler linguistic differences between the two types.

Overall Loss:

The cumulative loss for the model across all four axes was 2.2925, which is within acceptable limits for multi-task classification. However, the relatively lower performance on J/P and T/F axes indicates room for improvement.

4.2 Results for BERT Model

The following table summarizes the performance of the BERT model:

Axis	Accuracy	Loss
Introversion-Extraversion (I/E)	78.28%	0.5216
Judging-Perceiving (J/P)	62.01%	0.6641
Sensing-Intuition (S/N)	85.76%	0.4133
Thinking-Feeling (T/F)	54.26%	0.6868
Overall Loss	-	2.2858

Table 4.2.1. Result of Bert Model

Introversion-Extraversion (I/E):

The BERT model achieved an accuracy of 78.28%, nearly identical to the LSTM model, with a slightly lower loss of 0.5216, indicating consistent and stable predictions for this axis.

Judging-Perceiving (J/P):

Accuracy improved to 62.01%, with a loss of 0.6641. BERT's attention mechanism likely contributed to better distinguishing between J/P categories compared to the LSTM model.

Sensing-Intuition (S/N):

Performance remained consistent, with an accuracy of 85.76% and a loss of 0.4133. This reflects the model's ability to leverage contextual embeddings for accurate predictions.

Thinking-Feeling (T/F):

Accuracy was slightly lower at 54.26%, with a loss of 0.6868. Similar to the LSTM model, this axis remains the most challenging for classification.

Overall Loss: The cumulative loss for BERT was 2.2858, slightly lower than the LSTM model, suggesting marginally better performance across all axes.

4.3 Contrast Between LSTM and BERT Models

1. **Overall Performance:** Both models achieved comparable results, but BERT slightly outperformed LSTM in terms of lower overall loss and improved accuracy for the J/P axis.
2. **Effectiveness for Challenging Axes:** Neither model excelled in the T/F axis, but BERT demonstrated marginally better accuracy and loss compared to LSTM.
3. **Architectural Advantages:** LSTM's sequential processing excels in capturing patterns over time, but BERT's self-attention mechanism enables it to consider the entire context simultaneously, contributing to its superior performance on J/P.

4.4 Discussion on Architectures

- **LSTM Strengths:** Sequential dependency modeling and strong performance on S/N.
- **BERT Strengths:** Contextual embedding and superior handling of overlapping linguistic features, as demonstrated in J/P.
- **Areas for Improvement:** Both models need enhancements for T/F classification. Incorporating external linguistic resources or multitask learning could be explored.

Chapter 5: Conclusion

From this experiment we can conclude that there is some correlation between the type of content people write and their personality trait. Although our models couldn't capture all the nuances of texts for a more extensive psychological profiling, it showed that there is potential for AI in the field of psychology.

Both the LSTM and BERT models have strengths and weaknesses in predicting MBTI personality types based on text data. While both models performed strongly on the I/E and S/N axes, the J/P and T/F axes posed more significant challenges. The BERT model demonstrated a slight edge over LSTM in terms of overall accuracy and loss, owing to its contextual understanding and attention mechanism.

5.1 Limitations of the Models

- **Low Accuracy on T/F Axis:** Both models struggled with the T/F axis, highlighting the difficulty of capturing the subtle linguistic cues that differentiate Thinking from Feeling.
- **Imbalanced Dataset:** The dataset used for training may have imbalances in type distribution, which could affect the models' ability to generalize.
- **Limited External Context:** The models rely solely on the text data provided without leveraging additional external context or domain-specific knowledge.
- **Complexity of Personality Traits:** Personality traits are inherently complex and multi-dimensional, making it challenging for any model to achieve high accuracy across all axes.

5.2 Future Recommendations

- **Enhanced Preprocessing:** Consider using advanced NLP preprocessing techniques, such as domain-specific lexicons or semantic role labeling, to capture nuanced differences in language use.
- **Hybrid Models:** Explore hybrid architectures that combine the strengths of LSTM and BERT for improved performance.
- **Class-Specific Augmentation:** Implement data augmentation strategies targeted at underrepresented classes to address dataset imbalances.

- **Transfer Learning:** Leverage pre-trained models fine-tuned on domain-specific data to enhance the models' contextual understanding.
- **Explainability:** Integrate explainable AI techniques to better understand the linguistic features influencing predictions, which could provide valuable insights for improving model design.
- **Extended Validation:** Conduct validation on additional datasets or real-world text samples to assess generalizability and robustness.

By addressing these limitations and implementing the recommendations, future iterations of this research can achieve more accurate and reliable personality predictions, advancing the integration of machine learning in psychological profiling.

Chapter 6: References

1. Pennebaker, J. W., Booth, R. J., & Francis, M. E. (2003). Linguistic Inquiry and Word Count (LIWC): LIWC2007 Operator's Manual. Austin, TX: LIWC.net.
2. Xu, C., Yu, Z., Cai, Y., & Guo, B. (2018). Deep Learning for Personality Prediction: Insights and Improvements. Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI).
3. Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? Proceedings of the China National Conference on Chinese Computational Linguistics.

Appendix

GitHub Link: https://github.com/Krishna1032/MBTI_Prediction_DL.git