Name: Quang Dang

CS 510: NLP - Spring 24

Multiclass Text Classification Using MBTI 500 Dataset: A Comparative Study of CNN and DistilBERT Approaches

1) Abstract

This study explores the problem of multiclass text classification on the MBTI 500 dataset, which consists of recent Twitter and Reddit posts from various personality types. We compare two approaches: a Convolutional Neural Network (CNN) model using Word2Vec for word embeddings, and a pretrained transformer model, DistilBERT, for tokenization and prediction. The CNN model achieved reasonable results, while the DistilBERT approach failed due to resource and time constraints. This paper presents our methodologies, experimental setup, results, and discusses the limitations and potential future improvements.

2) Introduction

a) Problem Statement

The Myers-Briggs Type Indicator (MBTI) is a popular tool for personality assessment, categorizing individuals into 16 distinct personality types based on preferences in four dichotomies: Extraversion (E) vs. Introversion (I), Sensing (S) vs. Intuition (N), Thinking (T) vs. Feeling (F), and Judging (J) vs. Perceiving (P). Accurately classifying text into these 16 MBTI personality types is a challenging multiclass classification problem due to the nuanced and complex nature of human language and personality expression.

b) Importance

Understanding personality through text classification has significant applications in various fields.In Personalized Marketing, tailoring advertisements and recommendations based on personality can increase user engagement and satisfaction. In the field of psychology and mental health, analyzing social media posts for personality traits can help identify individuals in need of psychological support or intervention. Predict personailty can also help enhance user experience on social platforms by providing more personalized interactions and content.Finally, gaining deeper insights into human behavior, communication styles, and social interactions through large-scale text analysis.

c) Key Methods

In this study, we compare two state-of-the-art approaches for multiclass text classification:

I. **Word2Vec + CNN**: This method leverages Word2Vec embeddings to capture semantic relationships between words, followed by a Convolutional Neural Network (CNN) to learn spatial hierarchies and classify text into personality types.
II. **DistilBERT**: A pretrained transformer model that uses attention mechanisms to capture contextual relationships in text. We fine-tune DistilBERT for our

classification task, aiming to leverage its powerful language understanding capabilities.

3) Related Work

a) Text Classification Techniques

Text classification is a fundamental task in Natural Language Processing (NLP) with applications in sentiment analysis, spam detection, and topic categorization. Traditional methods include Bag-of-Words (BoW) which epresents text as a set of word frequencies, ignoring word order and context. A newer method TF-IDF also enhances BoW by weighting terms based on their importance in a document and across the corpus.

Modern techniques employ deep learning models that can capture more complex patterns includes Recurrent Neural Networks (RNNs) that **c**apture sequential dependencies in text but suffer from long-term dependency issues. Convolutional Neural Networks (CNNs), which Initially used in image processing, have shown promise in text classification by capturing local dependencies and hierarchical patterns.

b) Word2Vec and CNN

Word2Vec was introduced by Mikolov et al., Word2Vec models learn word embeddings by predicting a word's context (skip-gram) or the word given its context (CBOW). These embeddings capture semantic similarities between words, enhancing downstream NLP tasks. CNNs, popularized by Kim (2014) for text classification, apply convolutional filters to capture local n-gram features, followed by pooling layers to aggregate information and dense layers for final classification.

c) Transformer Models

Transformers, introduced by Vaswani et al., use self-attention mechanisms to process all words in a sentence simultaneously, capturing long-range dependencies. BERT (Bidirectional Encoder Representations from Transformers) and its distilled version, DistilBERT, pretrain on large corpora to learn general language representations, which can be fine-tuned for specific tasks. While transformers achieve state-of-the-art performance in many NLP tasks, they require significant computational resources for training and fine-tuning, posing challenges for resource-constrained environments.

4) Methods

a) Word2Vec + CNN

Word2Vec is a group of models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. We trained Word2Vec on the MBTI 500 dataset to obtain 100 and 200-dimensional word vectors.

In terms of CNN architecture**,** The CNN model consists of an embedding layer (initialized with Word2Vec embeddings), followed by convolutional layers with fixed filter sizes, max-pooling layers, a dropout layer for regularization, and a fully

connected dense layer for classification. For activation functions, we used ReLU for intermediate layers and softmax for the output layer. The output layer consists of 16 neurons with softmax activation, corresponding to the 16 MBTI personality types.

Layer (type)	Output Shape	Param #
embedding (word2vec 200)	(_, 500, 200)	42669000
dropout (0.2)	(_, 500, 200)	
conv1d (128,5,"ReLu")	(_, 496, 128)	128128
max_pooling1d (2)	(_, 248, 128)	
dropout_1 (0.2)	(_, 248, 128)	
conv1d_1 (128,5,"ReLu")	(_, 244, 128)	82048
max_pooling1d_1 (2)	(_, 122, 128)	
flatten	(_, 15616)	
dropout_2(0.2)	(_, 15616)	
dense (128, "ReLu")	(_, 128)	1998976
dropout_3 (0.2)	(_, 128)	
dense_1 (16, "sigmoid)	(_, 16)	2064
Total params: 44,880,216 Trainable params: 2,211,216 Non-trainable params: 42,669,000		

Table1: CNN Architecture

b) DistilBERT

For DistilBERTm we used a pretrained model for both tokenization and inference. DistilBERT's tokenizer was used to convert text into token ids that the model can process. DistilBERT is a smaller, faster, and cheaper version of BERT, maintaining 97% of BERT's performance while being 60% faster. The main challenges of DistilBERT are our resource constraints with limited computational resources(GTX 3050 with only 8GB of RAM) restricted the training time and batch size, impacting the model's ability to converge. The substantial time required for training transformer models with large datasets was a significant barrier.

5) Experiments

a) Preprocessing

The MBTI 500 Dataset: consists of text samples from social media platforms like Twitter and Reddit, categorized into 16 personality types. Preprocessing steps included tokenization, stopword removal, lemmatization, hashtag removal, URL removal, and normalization.

b) Experimental Setup

Hyperparameters:

- CNN: Learning rate (0.001), batch size (32), epochs (20), dropout rate (0.2).
- DistilBERT: Due to resource constraints, training was limited to a smaller batch size (16) and fewer epochs (3).

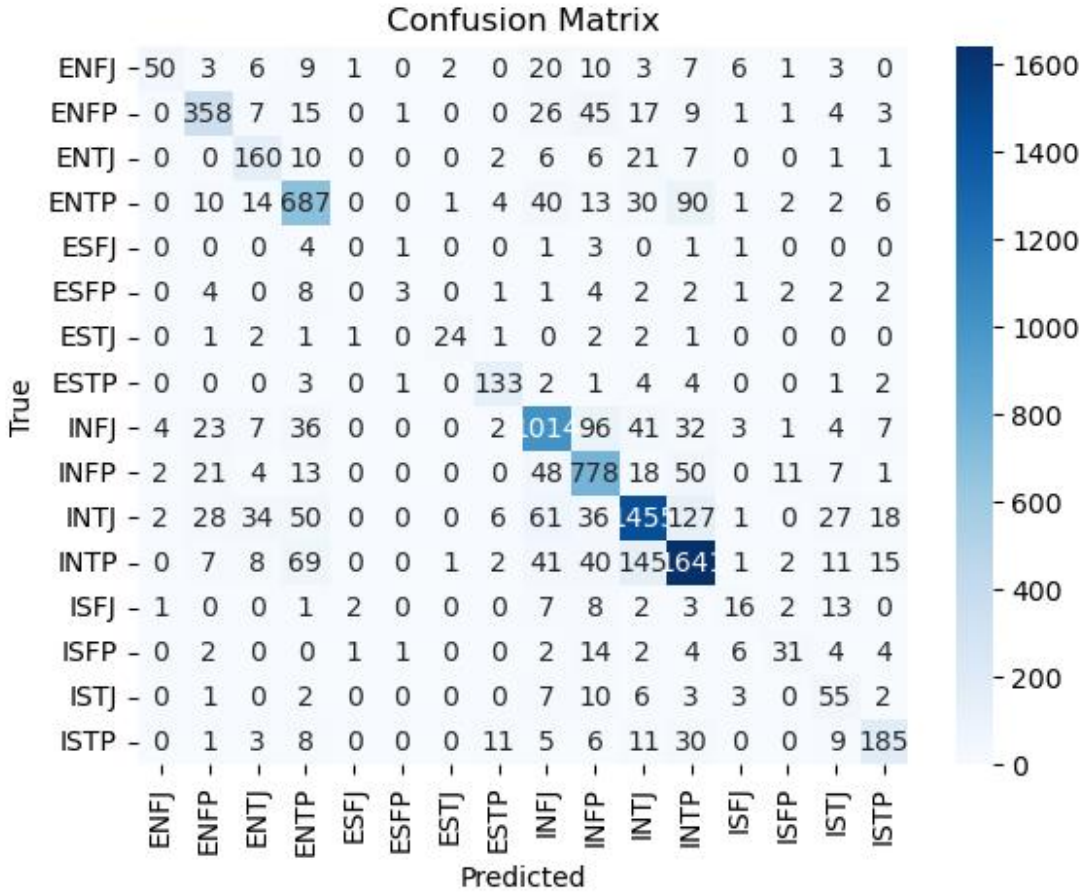
Training and Validation: Data was split into training, validation, and test sets (80/10/10). The CNN model was trained using backpropagation with cross-entropy loss and Adam optimizer. DistilBERT was fine-tuned on the dataset with the same optimizer.

6) Discussion and Analysis

a) Overall Performance

The model achieved a test accuracy of 76.42%, which is a substantial result for a multiclass classification problem with 16 classes.The test loss of 0.8739 indicates the model's performance in terms of error, with lower values indicating better performance. The relatively low loss value suggests that the model has learned useful patterns from the data.

b) Confusion Matrix Analysis



(Figure 1) Confusion matrix illustrates the distribution of true labels versus predicted labels. The diagonal elements represent the correctly classified instances for each personality type, while the off-diagonal elements indicate misclassifications.

Here are some key observations as can be seen from this matrix:

- **High Accuracy for INTP and INTJ**: The model shows strong performance in predicting INTP and INTJ types, with accuracies of 76% and 80%, respectively.

This suggests that the features and patterns associated with these personality types are well captured by the model.

- **Challenges with ESFJ and ESFP**: The model struggles significantly with the ESFJ and ESFP types, with very few correct predictions (1 and 3 respectively). The small sample size for these types (15 and 36 instances) might contribute to this poor performance, highlighting the need for more balanced data or additional techniques to handle class imbalance.
- **Moderate Performance for INFJ, INFP, ENTP, and ENFP**: The model shows reasonably good performance for these types, with precision and recall values in the range of 0.69 to 0.80. However, there are still notable misclassifications, such as INFJ being confused with INFP, reflecting the subtle differences in text features between these personality types.

c) Classification Report Analysis

Classification Report:

	precision	recall	f1-score	support
ENFJ	0.51	0.52	0.52	121
ENFP	0.75	0.79	0.77	487
ENTJ	0.65	0.74	0.69	214
ENTP	0.75	0.75	0.75	900
ESFJ	0.14	0.09	0.11	11
ESFP	0.20	0.06	0.10	32
ESTJ	0.70	0.60	0.65	35
ESTP	0.92	0.79	0.85	151
INFJ	0.85	0.75	0.80	1270
INFP	0.83	0.69	0.75	953
INTJ	0.80	0.83	0.81	1845
INTP	0.78	0.87	0.82	1983
ISFJ	0.31	0.22	0.26	55
ISFP	0.32	0.63	0.43	71
ISTJ	0.42	0.62	0.50	89
ISTP	0.80	0.66	0.72	269
accuracy			0.77	8486
macro avg	0.61	0.60	0.60	8486
weighted avg	0.78	0.77	0.77	8486

Table2. Classification Report

The classification report provides detailed metrics for each personality type, including precision, recall, and F1-score. Key observations include:

- **High Precision and Recall for INTP and INTJ**: Both INTP and INTJ types exhibit high precision (0.78 and 0.80) and recall (0.87 and 0.83), resulting in high F1-scores (0.82 and 0.81). This aligns with the confusion matrix findings, reinforcing the model's capability to accurately classify these types.
- **Low Performance for ESFJ and ESFP**: The ESFJ type has a precision, recall, and F1-score of 0.00, indicating that the model fails to correctly identify any instances of this type. Similarly, ESFP shows low precision (0.20), recall (0.08), and F1-score (0.12). These low scores highlight the need for improved handling of underrepresented classes.

- **Balanced Performance Across Most Types**: For the majority of the personality types, the model achieves balanced precision and recall scores, resulting in reasonable F1-scores. This indicates that the model is generally effective in distinguishing between different personality types, although there is room for improvement in certain areas.

d) Error Analysis

The error analysis reveals several important insights:

- **Class Imbalance**: The dataset is imbalanced, with certain personality types having significantly fewer instances. This imbalance likely contributes to the poor performance for types like ESFJ and ESFP.
- **Confusion Between Similar Types**: The model often confuses similar personality types, such as INFJ and INFP. This suggests that the textual features distinguishing these types are subtle and might require more sophisticated feature extraction or additional data for better differentiation.
- **Resource Constraints**: The resource constraints faced during the experiment, particularly with the DistilBERT approach, highlight the importance of adequate computational resources for training large models. Future work should consider more powerful hardware and longer training times to fully leverage transformer models' capabilities.

Future Work

To address the identified limitations and improve model performance, future work should consider the following:

1. **Handling Class Imbalance**: Implement techniques such as oversampling, undersampling, or synthetic data generation to balance the dataset and improve performance for underrepresented classes.
2. **Advanced Feature Extraction**: Explore more sophisticated feature extraction methods, such as transformer-based embeddings, to capture nuanced textual features.
3. **Resource Optimization**: Utilize more powerful computational resources (Talapas) and optimized training strategies to fully exploit the potential of transformer models like DistilBERT.
4. **Data Augmentation**: Apply data augmentation techniques to increase the diversity and quantity of training data, potentially improving the model's generalization capabilities.
5. **Ensemble Methods**: Combine multiple models using ensemble techniques to leverage their strengths and achieve more robust performance.

References

[1] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781
[2] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)
[3] Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108

[4] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119)
[5] Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759
[6] Skowron, M., Tkalčič, M., Ferwerda, B., Schedl, M. (2016). Fusing Social Media Cues: Personality Prediction from Twitter and Instagram. Proceedings of the 25th International Conference Companion on World Wide Web