Northeastern University
Khoury College of
Computer Sciences

Progress Note Understanding: Assessment and Plan Reasoning

CS6120 Natural Language Processing

Team 106 - Bobby Doshi & Harsh Agrawal

1. Abstract

The Progress Note Understanding: Assessment and Plan Reasoning task, part of the 2022 N2C2 shared task, focuses on extracting and understanding the causal relationships within the assessment and plan sections of clinical progress notes. Using annotated data from the MIMIC-III dataset, this project aims to predict the relations between assessment and plan subsections with labels such as Direct, Indirect, Neither, and Not Relevant. Our approach involved fine-tuning multiple NLP models, including BERT [2], BioBERT, ClinicalBERT [3], a Multi-Task BERT model, and a BiLSTM model. While BERT served as a benchmark, we concentrated on improving the performance of smaller language models such as Tiny-ClinicalBERT and Tiny-BioBERT [4, 5], and BiLSTM. These tiny models range in size from 45 MB to 50 MB, enabling us to achieve comparable scores while significantly reducing memory requirements, making the models suitable for mobile devices. Ultimately, we achieved a Macro F1 score of 0.780 with the Multi-Task BERT model. Our work demonstrates the potential of transformer-based models in clinical decision support, enabling automated systems to accurately extract and understand medical information from unstructured text, ultimately supporting clinical workflows and improving patient care. For reference, the highest score in the N2C2 competition was 0.8212, achieved by CMU.

2. Introduction

Healthcare providers generate notes in electronic health records (EHRs) to document daily progress and treatment plans. The Subjective, Objective, Assessment, and Plan (SOAP) format is a widely adopted structure for these notes, with the Assessment and Plan sections being particularly crucial. The Assessment section summarizes the patient's current health problems, while the Plan section details the treatment strategies for each problem.

2.1 Problem Statement:

The challenge is to automatically extract and understand the relationships between the Assessment and Plan sections. Specifically, each plan subsection needs to be categorized based on its relationship to the assessment: Direct, Indirect, Neither, or Not Relevant. This understanding can enhance downstream applications like problem list generation, thus aiding clinical decision-making.

2.2 Background:

Progress notes in EHRs are essential for documenting patient care but are often plagued by issues such as note bloat and information overload, which hinder efficient care delivery. The Assessment and Plan sections, written in free text, encapsulate critical information about a patient's health status and the corresponding medical interventions. However, this information is not easily accessible for automated analysis, necessitating advanced natural language processing (NLP) techniques to extract and interpret the data.

2.3 Previous Work:

The hierarchical annotation framework proposed by Gao et al. (2022) [1] serves as a foundation for this challenge. Previous shared tasks have emphasized the importance of relation extraction and document classification in the clinical domain. The advent of transformer models, such as BERT and its clinical variants (e.g., ClinicalBERT, BioBERT [2, 3]), has significantly advanced NLP capabilities in the biomedical field. These models have demonstrated superior performance in various NLP tasks, including named entity recognition (NER) and relation extraction.

2.4 Approaches Taken by Participants:

The participants of the 2022 N2C2 Track 3 shared task employed various strategies to address the challenge. Most teams used transformer-based models, with BERT and its variants being the most popular choices. Some teams incorporated external medical knowledge sources, such as medical ontologies, to enhance their models' understanding of clinical concepts. For instance, the top-performing team used additional EHR note types and applied Bayesian inference over ensemble BERT models to improve their predictions. Another approach that showed promise was the integration of named entity recognition (NER) tagging to better capture the relationships between concepts [6].

Several teams observed that the ordering information of Plan subsections could be beneficial, but the improvements were not significant. Additionally, the use of Longformer for handling long documents did not yield substantial performance gains compared to BERT models with smaller token limits, as few samples exceeded the token limit [6]. These insights highlight the importance of model architecture and the integration of external knowledge in improving the performance of NLP systems for clinical decision support.

3. Methods

In this section, we discuss the models we employed to tackle the problem of understanding the relationships between the Assessment and Plan sections of clinical progress notes. Given the nature of the task and the constraints of real-world applications, especially in resource-limited settings, we selected a range of models varying in complexity and computational requirements. We focused on leveraging both large transformer-based models and smaller, more efficient models suitable for deployment on low-resource devices.

3.1 Model Selection

3.1.1 BERT (Bidirectional Encoder Representations from Transformers)

BERT [2] has become a cornerstone in NLP due to its bidirectional nature and deep understanding of context. We used the base BERT model (bert-base-uncased) as our benchmark. Despite its success across various NLP tasks, BERT's size (approximately 110 million parameters) can be a limitation in low-resource environments, which is why we explored more efficient alternatives alongside it. Previous studies have successfully applied BERT to clinical tasks, such as named entity recognition and relation extraction in medical records [3].

3.1.2 Tiny-ClinicalBERT and Tiny-BioBERT

We used two distinct "tiny" models, Tiny-ClinicalBERT and Tiny-BioBERT, to explore the effectiveness of smaller models in a clinical context:

- **Tiny-ClinicalBERT**: This model is a distilled version of BioClinicalBERT, which itself is based on BioBERT but further fine-tuned on clinical text from the MIMIC-III dataset [3]. The distillation was performed for three epochs using a batch size of 192. Tiny-ClinicalBERT uses a unique 'transformer-layer distillation' method that aligns the attention maps and hidden states of the student model (Tiny-ClinicalBERT) with those of the teacher model. The model architecture includes 4 hidden layers, each with a hidden dimension and embedding size of 768, resulting in approximately 15 million parameters. Given its small size, the model is initialized randomly [4].
- **Tiny-BioBERT**: Tiny-BioBERT is a distilled version of BioBERT, designed for use in biomedical NLP tasks. The distillation process involved 100k training steps with a batch size of 192 on the PubMed dataset. Like Tiny-ClinicalBERT, it uses the 'transformer-layer distillation' method to ensure that the attention maps and hidden states closely match those of the full-sized BioBERT model. The architecture of Tiny-BioBERT is similar, with 4 hidden layers and 15 million parameters, and it is also initialized randomly [5].

Both models offer a significant reduction in computational requirements without a substantial loss in performance, making them ideal for real-time applications in healthcare where resource constraints are a concern. Previous work has shown that Tiny-ClinicalBERT can be used effectively for clinical concept extraction [4], while Tiny-BioBERT has been successfully applied to biomedical named entity recognition and relation extraction tasks [5].

3.1.3 BiLSTM with Attention

Recurrent Neural Networks (RNNs) like LSTMs have traditionally been used for sequence data, and their bidirectional variant (BiLSTM) has shown promise in capturing dependencies from both past and future contexts in a sequence. We implemented a BiLSTM model with an attention mechanism to focus on the most relevant parts of the input sequence. Although BiLSTMs are generally less powerful than transformers, they are more computationally efficient, making them suitable for scenarios where processing power is a bottleneck [6]. Previous work has demonstrated the effectiveness of BiLSTM with attention mechanisms in text classification and sequence labeling tasks, particularly in domains requiring sequential data analysis [7].

3.1.4 Multi-Task BERT Model

To improve the model's ability to generalize across different sub-tasks, we employed a Multi-Task BERT model. This approach allowed us to leverage shared representations across related tasks, which is particularly beneficial when dealing with complex domains like healthcare. Multi-task learning has been shown to improve the performance of NLP models by allowing them to learn common features across tasks, which can be crucial when data is sparse [8]. The use of multi-task learning in clinical NLP has been explored in previous studies, demonstrating improvements in tasks such as medical code prediction and clinical outcome forecasting [9].

3.2 Rationale for Model Selection

The selection of models was guided by the following considerations:

- **Domain Specificity**: Models like Tiny-ClinicalBERT and Tiny-BioBERT are pre-trained on specific datasets that are highly relevant to our task. Tiny-ClinicalBERT is trained on the MIMIC-III dataset, which contains de-identified health data from ICU patients, making it particularly suited for clinical text tasks [3]. Tiny-BioBERT, on the other hand, is pre-trained on a combination of biomedical literature, making it more adept at handling biomedical terms and relations [5].
- **Efficiency**: The smaller size of the Tiny models (approximately 45-50 MB) makes them ideal for deployment in environments where computational resources are limited. This efficiency does not come at the expense of performance, as these models have been fine-tuned on relevant clinical data to retain accuracy while reducing computational overhead.
- **Performance**: While transformer-based models like BERT and its variants are state-of-the-art in many NLP tasks, we hypothesized that multi-task learning could further enhance the model's ability to generalize across different aspects of the data, leading to better performance in understanding the nuanced relationships within clinical notes.

We focused solely on the dataset provided by the N2C2 Track 3 task and did not supplement it with additional data sources, unlike some participants in the competition [6]. Our goal was to evaluate the effectiveness of the models in a controlled setting, ensuring that any improvements were attributable to the models themselves rather than external data sources.
Each model was fine-tuned on our dataset to predict the relationship labels (Direct, Indirect, Neither, Not Relevant) for the Plan subsections based on the corresponding Assessment section.

4. Data

The dataset used in this study was derived from the N2C2 Track 3 competition, which focuses on understanding the relationships between the Assessment and Plan sections in clinical progress notes. The dataset consists of progress notes extracted from the MIMIC-III database, a widely recognized source of electronic health records (EHR) from patients admitted to the intensive care unit (ICU) at the Beth Israel Deaconess Medical Center .

4.1 Data Composition and Distribution

The dataset provided by the organizers was divided into training, development, and testing sets, with a total of 4,633, 597, and 667 Assessment-Plan (A-P) pairs, respectively. These pairs were extracted from 598, 75, and 86 hospital admissions. The A-P pairs were annotated by medical experts, who assigned each pair one of four relationship labels: "Direct," "Indirect," "Neither," or "Not Relevant" . The distribution of these labels across the dataset is shown in Table 1, which ensures a balanced representation of each class.

To create a more manageable dataset size for our experiments, we sampled 50% of the available data from each category, maintaining the original distribution to ensure the representativeness of the dataset. This stratified sampling approach allowed us to evaluate our models while

keeping the computational requirements feasible.

4.2 Data Preprocessing

Given the complexity of clinical text, especially in the context of progress notes, we employed several preprocessing strategies to clean and structure the data:

1. **Concatenation of Multiline Text**: Clinical notes often contain multiline text, where subsections are split across lines. We concatenated these lines to ensure that each Plan subsection was treated as a cohesive unit, which is crucial for accurately capturing the context of the Plan in relation to the Assessment.
2. **De-identification Removal**: The MIMIC-III dataset includes de-identified patient information marked by tags (e.g., [** **]). We removed these tags to ensure that no personal information was included in the dataset, complying with data privacy regulations and focusing on the textual content relevant to the task.
3. **Text Normalization**: Standard text preprocessing techniques were applied, including lowercasing, punctuation removal, and whitespace normalization. These steps helped standardize the text, reducing variability that could negatively impact model performance.
4. **Feature Engineering**: We added features such as text length and word count for both the Assessment and Plan subsections. These features provided additional context to the models, helping to distinguish between the different relationship categories, especially in cases where the textual content alone might be ambiguous.

4.3 Data Pipeline

We implemented a standardized data pipeline that ensured consistency in data handling across all models, whether transformer-based (e.g., BERT) or non-transformer-based (e.g., BiLSTM). The pipeline includes tokenization, text conversion, and label encoding, essential for preparing the data for input into various models. This standardization was crucial for maintaining the integrity of the experimental results and for ensuring that the models were trained on the same high-quality data.

The dataset class, N2C2Track3Dataset, was designed to handle the data loading, tokenization, and splitting into training, development, and testing sets. This class ensured that the dataset was consistently processed and fed into the models, facilitating seamless integration across different model architectures.

4.4 Data Sharing and Access

Due to the sensitive nature of the MIMIC-III data, the dataset used in this study cannot be shared directly. The raw data is accessible only through the N2C2 competition or PhysioNet, and access requires signing the MIMIC data usage agreement. A sample dataset, compliant with the user agreement, has been attached to the code provided. For access to the full dataset, interested parties should request permission through the appropriate channels, ensuring compliance with all data usage policies.

5. Results

The performance of the models was evaluated based on three key metrics: Accuracy, F1 Score, and Macro F1 Score. The results are summarized in the following tables:

MODEL	Accuracy	F1 Score	Macro F1 Score
BiLSTM	0.70	0.69	0.619
ClinicalBERT	0.72	0.68	0.611
BioBERT	0.60	0.61	0.534
BERT	0.82	0.76	0.662
Multi-Task BERT	0.85	0.78	0.78

Classification Reports :

	precision	recall	f1-score	support
Direct	0.87	0.77	0.82	26
Indirect	0.00	0.00	0.00	29
Neither	0.91	1.00	0.95	42
Not Relevant	0.79	1.00	0.88	103
accuracy			0.82	200
macro avg	0.64	0.69	0.66	200
weighted avg	0.71	0.82	0.76	200

Macro F1 Score: 0.662803466374895

Fig 1 - BERT Model

Direct	1.00	0.35	0.51	26
Indirect	0.56	0.66	0.60	29
Neither	0.58	0.50	0.54	42
Not Relevant	0.76	0.89	0.82	103
accuracy			0.70	200
macro avg	0.73	0.60	0.62	200
weighted avg	0.73	0.70	0.69	200

Macro F1 Score: 0.6193376068376069

Fig 2- BiLSTM Model

	precision	recall	f1-score	support
Direct	0.49	1.00	0.66	26
Indirect	0.24	0.41	0.31	29
Neither	0.80	0.29	0.42	42
Not Relevant	0.84	0.68	0.75	103
accuracy			0.60	200
macro avg	0.59	0.59	0.53	200
weighted avg	0.70	0.60	0.61	200

Macro F1 Score: 0.5349152398538829

Fig 3 - BioBERT

	precision	recall	f1-score	support
Direct	0.68	1.00	0.81	26
Indirect	0.53	0.31	0.39	29
Neither	0.67	0.29	0.40	42
Not Relevant	0.76	0.94	0.84	103
accuracy			0.72	200
macro avg	0.66	0.63	0.61	200
weighted avg	0.70	0.72	0.68	200

Macro F1 Score: 0.611820652173913

Fig 4 - Clinical BERT

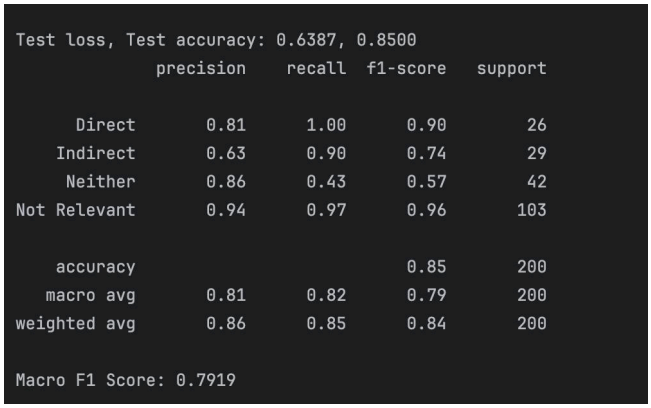

Fig 5 - Multi-task BERT

6. Discussion

The results indicate that transformer-based models, particularly BERT and Multi-Task BERT, significantly outperformed other models in understanding the relationships between the Assessment and Plan sections in clinical progress notes. The Multi-Task BERT model achieved the highest overall performance with an Accuracy of 85% and a Macro F1 Score of 0.7919, demonstrating its ability to effectively generalize across multiple tasks. This model excelled particularly in the "Not Relevant" and "Direct" categories, achieving F1-scores of 0.96 and 0.90, respectively. These results suggest that leveraging multi-task learning can enhance the model's understanding of complex and nuanced relationships in clinical text, making it a valuable approach for such tasks.

The standard BERT model also performed well, with an Accuracy of 82% and a Macro F1 Score of 0.662. It showed strong performance in the "Neither" and "Not Relevant" categories, with F1-scores of 0.95 and 0.88, respectively. However, it struggled significantly with the "Indirect" category, failing to predict any positive samples accurately, which was a common challenge across models. This highlights BERT's robustness as a general-purpose transformer model in the clinical domain, though it also underscores the need for further refinement to improve performance in more challenging categories like "Indirect."

In comparison, domain-specific models like ClinicalBERT and BioBERT, although pre-trained on relevant biomedical literature, did not achieve comparable results. ClinicalBERT achieved an Accuracy of 72% and a Macro F1 Score of 0.611, while BioBERT achieved a lower Accuracy of 60% and a Macro F1 Score of 0.534. These results indicate that while domain-specific pre-training is beneficial, it may not always translate into superior performance on complex tasks requiring deep contextual understanding, particularly when fine-tuning is not extensively tailored to the specific task at hand.

The BiLSTM model, included as a baseline, showed competitive results with an Accuracy of 70% and a Macro F1 Score of 0.619. While this performance is lower than that of the transformer models, it demonstrates the model's ability to capture sequence dependencies effectively, albeit with less sophistication. The BiLSTM model showed balanced performance across categories but, like the transformer models, struggled with the "Indirect" category.

Overall, the results affirm the effectiveness of transformer models, particularly when fine-tuned for specific tasks like clinical reasoning. The consistent difficulty across all models in accurately predicting the "Indirect" label points to an area that requires further research and possibly new approaches to better capture this complex relationship in clinical notes. The Multi-Task BERT model's superior performance underscores the value of multi-task learning in enhancing model accuracy and generalizability in clinical NLP tasks. Future work could explore further fine-tuning and model optimization, particularly for challenging categories, to close the gap between the current performance and the top-performing models in the N2C2 competition.

References

1. Gao, Yanjun, Dmitriy Dligach, Timothy Miller, Samuel Tesch, Ryan Laffin, Matthew M. Churpek, and Majid Afshar. “Hierarchical Annotation for Building A Suite of Clinical Natural Language Processing Tasks: Progress Note Understanding.” arXiv preprint arXiv:2204.03035 (2022).
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).
3. Gao, Jifan, Shilu He, Junjie Hu, and Guanhua Chen. “A hybrid system to understand the relations between assessments and plans in progress notes.” Journal of Biomedical Informatics 141 (2023): 104363.
4. Rohanian, O., Nouriborji, M., Jauncey, H., Kouchaki, S., Nooralahzadeh, F., Clifton, L., … & Clifton, D. A. (2023). Lightweight transformers for clinical natural language processing. Natural Language Engineering, 1-28.
5. Rohanian, O., Nouriborji, M., Kouchaki, S., & Clifton, D. A. (2023). On the effectiveness of compact biomedical transformers. Bioinformatics, 39(3), btad103.
6. Gao, Yanjun, Dmitriy Dligach, Timothy Miller, Matthew M Churpek, Ozlem Uzuner, and Majid Afshar. “Progress Note Understanding – Assessment and Plan Reasoning: Overview of the 2022 N2C2 Track 3 Shared Task.” arXiv preprint arXiv:2303.08038 (2023). Retrieved from [https://arxiv.org/abs/2303.08038].
7. Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. B. A. (2019). Publicly available clinical BERT embeddings. Proceedings of the 2nd Clinical Natural Language Processing Workshop (pp. 72-78).
8. Si, Y., Wang, J., Xu, H., & Roberts, K. (2019). Enhancing clinical concept extraction with contextual embeddings. Journal of the American Medical Informatics Association, 26(11), 1297-1304.
9. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240.
10. Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (pp. 2873-2879).

Appendix

1. Structure of the Code

The project is structured to evaluate the performance of various models on the N2C2 dataset. The different models implemented include BERT, BiLSTM, BioBERT, ClinicalBERT, and a multi-task model. Additionally, there are scripts for data preparation and preprocessing. Below is an overview of the key components:

Code Structure:

- **Data Preprocessing and Preparation**:
  - `data_pre_processing.py`: Handles data loading, cleaning, feature engineering, and saving the processed dataset.
  - `n2c2_dataset.py`: Defines the custom dataset class `N2C2Track3Dataset` used for loading and tokenizing the dataset, compatible with both PyTorch and TensorFlow.
- **Model Implementations**:
  - `bert_model.py`: Implements a BERT model using the Hugging Face Transformers library for sequence classification on the N2C2 dataset.
  - `BiLSTM_model.py`: Implements a Bidirectional LSTM model with an attention mechanism for sequence classification.
  - `biobert_model.py`: Implements a BioBERT model, fine-tuned on the N2C2 dataset, for medical text classification.
  - `clinicalbert_model.py`: Implements a ClinicalBERT model, specifically tuned for clinical data from the N2C2 dataset.
  - `multi_task_model.py`: Implements a custom multi-task BERT model designed to handle multiple tasks simultaneously, leveraging shared layers for efficiency.

External Dependencies and Libraries

The project uses a variety of external libraries, primarily for machine learning, natural language processing, and data manipulation. The key dependencies are listed in the `requirements.txt` file:

- **Pandas**: For data loading, manipulation, and preprocessing.
- **Numpy**: For numerical operations and handling arrays.
- **Scikit-learn**: For model evaluation, including classification reports and F1-score calculations.
- **Transformers**: From Hugging Face, for implementing and fine-tuning various BERT models.
- **TensorFlow**: For training and evaluating the BERT, BioBERT, ClinicalBERT, and multi-task models.
- **PyTorch**: For handling the custom dataset class and supporting model implementations.
- **Torch**: Core library used for PyTorch-based models, such as the BiLSTM model.

You can install all the required dependencies using the provided `requirements.txt` file: `pip install -r requirements.txt`

Description of Classes, Functions, and Methods Developed

- `get_args()`: Common function across scripts to parse command-line arguments for configuring paths, model parameters, and other settings.
- `preprocess_text()` **(in** `data_pre_processing.py`**)**: Cleans text data by removing extra whitespace, with optional steps to remove punctuation, numbers, and convert text to lowercase.
- `concatenate_multiline_text()` **(in** `data_pre_processing.py`**)**: Combines lines of text starting with white spaces or tabs into a single line, facilitating cleaner input for

models.
- `remove_mimic_deid()` **(in** `data_pre_processing.py`**)**: Removes de-identified information from the MIMIC dataset using regex.
- `add_text_features()` **(in** `data_pre_processing.py`**)**: Adds additional features such as text length and word count for the 'Assessment' and 'PlanSubsection' columns.
- `generator()` **(used in model scripts)**: A generator function yielding batches of data for training or evaluation, compatible with TensorFlow's data pipeline.
- `MultiTaskBERTModel` **(in** `multi_task_model.py`**)**: A custom TensorFlow model that extends BERT for multi-task learning, including shared layers and task-specific output layers.
- `N2C2Track3Dataset` **(in** `n2c2_dataset.py`**)**: A custom PyTorch dataset class that handles loading, tokenizing, and preparing the N2C2 dataset for model training and evaluation.

Inputs and Outputs

- **Inputs**:
  - **Raw Data CSV File**: (`n2c2_sample_raw.csv`) - Contains a subset from the original dataset (due to access restrictions) with text fields for 'Assessment' and 'PlanSubsection'.
  - **Pretrained BERT Models**: BERT variants such as `nlpie/tiny-clinicalbert`, `biobert`, etc., are used for tokenization and model initialization.
  - **Command-line Arguments**: Various parameters including file paths, batch size, epochs, learning rate, and whether to use a local model.
- **Outputs**:
  - **Cleaned Data CSV File**: (`cleaned_dataset.csv`) - The output after data preprocessing and feature engineering.
  - **Trained Models**: The trained models are saved to the specified directory if the local model flag is enabled.
  - **Evaluation Metrics**: Metrics such as accuracy, classification reports, and macro-averaged F1-score are printed to the console.

2. User Manual on How to Run the Code

Prerequisites

Ensure you have Python installed (preferably version 3.7 or higher)

Steps to Run the Code

1. **Data Preprocessing**:

Use the `data_pre_processing.py` script to load, clean, and preprocess the raw data. This script will output a cleaned dataset CSV file:

```
python data_pre_processing.py --path_data data/n2c2_sample_raw.csv
--cleaned_data_path data/cleaned_dataset.csv
```

2. **Training the Models**:

Each model script (`bert_model.py`, `BiLSTM_model.py`, `biobert_model.py`, `clinicalbert_model.py`, `multi_task_model.py`) can be used to train a specific model. For example, to train the BERT model:

```
python bert_model.py --path_data data/cleaned_dataset.csv --batch_size 4
--epochs 5
```

Replace `bert_model.py` with the appropriate script name to train other models.

3. **Evaluating the Models**:

After training, the models can be evaluated on the test dataset. For example, to evaluate the BERT model:

```
python bert_model.py --mode test --path_data data/cleaned_dataset.csv
--batch_size 4
S
```

4. **Evaluating the best Multi-task BERT Model**:

To evaluate the best model on any text for prediction, use this script -

```
python multitask_demo_file.py --model_dir  --text_a "$SAMPLE ASSESSMENT"
--text_b "$SAMPLE PLANSUBSECTION"
```

5. **Custom Dataset Usage**:

The `N2C2Track3Dataset` class in `n2c2_dataset.py` can be imported and used in your PyTorch-based projects as follows:

```
from n2c2_dataset import N2C2Track3Dataset

train_dataset = N2C2Track3Dataset(args, mode='train')

test_dataset = N2C2Track3Dataset(args, mode='test')
```