Contextualized Medication Event Extraction

Jonathan So JS12684@NYU.EDU
Graduate School of Arts and Science
New York University
New York, NY 10003, USA

Krutika Pandit KAP9623@NYU.EDU
Graduate School of Arts and Science
New York University
New York, NY 10003, USA

Abstract

Unstructured clinical documentation often holds many intricate details regarding medication events that contribute to a comprehensive understanding of a patient's medication history. Using state-of-the-art transformer-based models, we performed named entity recognition and event classification for drug mentions in clinical notes. Model performance was assessed after applying a variety of tuning methods including weighted loss, embedding layer freezing, and early stopping. Overall, BERT models pre-trained on a domain-specific corpus were among the best performers across both tasks.

Project data and code files can be found at: https://github.com/kap9623/DL_Final_2022.

1. Introduction

Provision of quality medical care relies heavily on a comprehensive and accurate understanding of a patient's medication history. Healthcare providers may use this information to better assess the appropriateness of current treatment, detect potential medication-related symptoms, and plan future treatment options (Mahajan et al., 2022). Although medication information can be easily accessed from structured medication orders in electronic health record systems, intricate details of medication events are often captured in unstructured forms of clinical documentation. Recognition and retrieval of medication event information from unstructured data allows for a more complete picture of the patient's medication history.

The purpose of this project was to contextualize medication events in a set of unstructured clinical notes using deep learning methods. The project was segmented into two tasks: Task 1 involved Named Entity Recognition (NER), where all medication mentions were extracted from the clinical notes. Task 2 consisted of Event Classification, where extracted medication mentions were classified as either Disposition (medication change discussed, e.g. *"Start Plavix"*), NoDisposition (no change discussed, e.g. *"continue lipitor"*), or Undetermined (needs more information, e.g. "*Plan: Lasix"*).

Previous work has been done on this topic, although there were some limitations in the information retrieved from unstructured data; many studies had a narrow scope, focusing on specific medication information. For example, some studies only focused on medications involved in a single medical condition, e.g. heart failure (Meystre et al., 2015). Other studies

extracted information about a specific drug, e.g warfarin (Liu et al., 2011), or information relating to a single type of drug events, e.g discontinuation. (Liu et al., 2019) Our project hoped to have a broader scope by providing a more comprehensive analysis, first by capturing all medication mentions, then classifying medication changes across general categories. The results of our project may have practical applications in clinical tasks, such as establishing a longitudinal medication timeline or medication reconciliation.

The goal of this project was to develop and tune a deep learning model with maximal performance across both project tasks. We hypothesized that performing transfer learning using a BERT model pre-trained on a corpus of similar domain would outperform a BERT model pre-trained on a corpus of general domain.

2. Data

Our dataset consisted of 400 clinical notes, provided as part of Task 1 of the Harvard N2C2 2022 Challenge. These notes were selected from the 2014 i2b2/UTHealth Natural Language Processing shared task corpus and included medication information for 240 patients, totalling 4154 drug mentions. The notes were presented in the .txt format, along with a corresponding annotation file in the .ann format (BRAT). The annotation file provided a list of medication mentions in their respective note, in addition to the start and end character indices where the mention was found, and the disposition label for the Event Classification task. The medication mentions in the notes were annotated by a team of three annotators, who utilized a medical extraction model and corrected the results, if necessary. They then manually determined if a medication change was discussed and labeled with the appropriate class. For our project, we utilized the same 70-10-20 train-validation-test split across both tasks, numbering 280, 40, and 80 notes respectively.

3. Methods

All experiments were conducted with state-of-the-art Bidirectional Encoder Representations from Transformer (BERT)-based language models (Wolf et al., 2020). Our baseline was established using BaseBERT (Devlin et al., 2019) pre-trained on BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). The performance of the baseline was compared against three domain-specific pre-trained BERT models - BioBERT (Lee et al., 2019), SciBERT (Beltagy et al., 2019), and ClinicalBERT (Alsentzer et al., 2019) for both tasks.

Prior to model building, data from the clinical note (.txt) and annotation files (.ann) was converted from the BRAT to ConLL format. An example of this conversion is shown in Figure 1. This ensured the input data was compatible with the Simple Transformers module which contains models equipped with features and functionality specific to our named entity recognition (NER) tasks (Rajapakse, 2022).

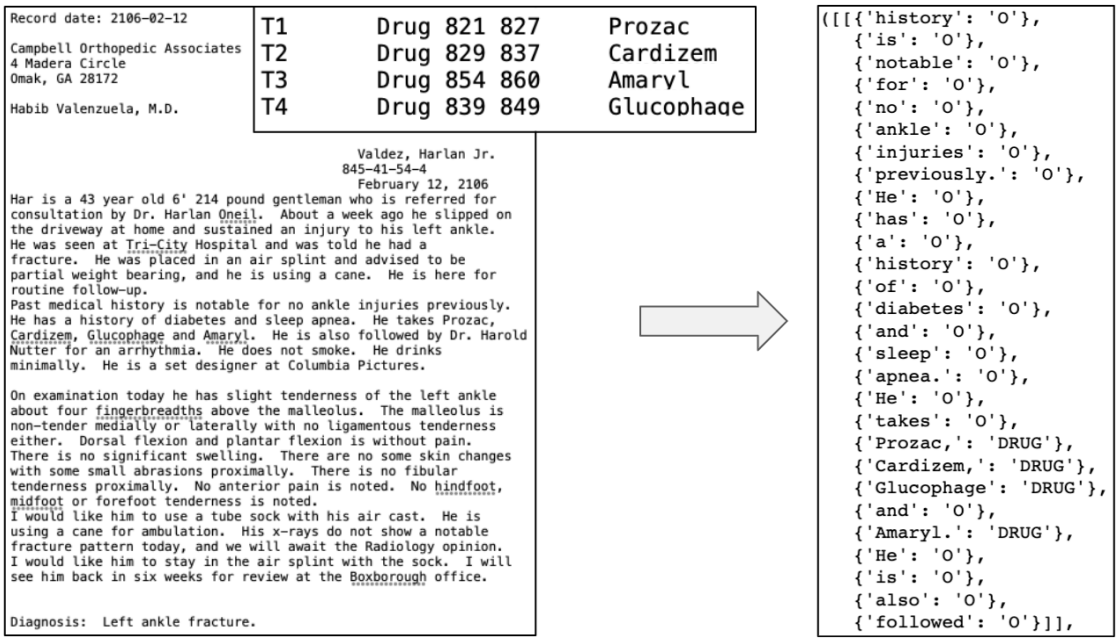

Figure 1: Example conversion from BRAT to ConLL for Task 1. On the left is a sample clinical note with the corresponding annotation file. Each word in the note was extracted and given either an "O" tag (meaning outside the studied named entity) or a "Drug"/"Disposition" tag depending on the task at hand.

First, the ConLL output was converted into a dataframe with a column assigning each token an ID based on the note or patient it belonged to. The four BERT models were tested using default hyperparameters with both note and patient level ID assignment. Next, class imbalance was addressed using weighted loss. Equation 1 was used to calculate each class weight, which was then passed to the model when instantiated. This was followed by a grid search for the best learning rate and batch size. Since running loss while training was substantially lower than the calculated evaluation loss, early stopping was performed with 25 epochs to reduce the possibility of overfitting. For Task 2, an additional L2 regularization step was performed. A complete overview of the model building and tuning process is outlined in Figure 2 in Appendix A. All experiments were conducted using the NERModel architecture in the Simple Transformers module with AdamW as the optimizer and cross entropy loss as the loss function.

Equation 1:

$$classweight = \frac{(number\ of\ samples)}{(number\ of\ classes)(class\ frequency)}$$

4. Results

The following evaluation metrics were used for both medication extraction and event classification tasks: evaluation loss, precision, recall, and F1 score.

Task 1: Medication Extraction

Performance using note and patient level token IDs was collected from all four models. As seen in Table 1, in all four cases, the note level split performed substantially better. Using the note level split, performance was next monitored with weighted loss, freezing, and both. These results, shown in Table 2, were compared against the default performance. BaseBERT and BioBERT performed best with the addition of both weighted loss and embedding layer freezing; however, SciBERT and ClinicalBERT performed best with default parameters. Next, a grid search was performed to tune the learning rate and batch size for each model as shown in Table 6 of Appendix B.

		Test Loss	Precision	Recall	F1-Score
baseBERT	Note-level	0.018	1.0	0.55	**0.71**
	Patient-level	0.099	0.9	0.40	0.55
BioBERT	Note-level	0.012	1.0	0.72	**0.84**
	Patient-level	0.057	0.89	0.59	0.71
sciBERT	Note-level	0.008	0.96	0.89	**0.93**
	Patient-level	0.030	0.89	0.77	0.83
clinicalBERT	Note-level	0.006	0.89	0.89	**0.89**
	Patient-level	0.030	0.89	0.77	0.83

Table 1: Performance across four BERT models with note versus patient level token IDs.

	Default			Default + weights			Default + freezing			Default + weights + freezing		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BaseBERT	1.0	0.55	0.71	0.97	0.72	0.83	0.87	0.72	0.79	0.95	0.79	**0.86**
BioBERT	1.0	0.72	0.84	0.88	0.75	0.81	0.90	0.72	0.80	0.89	0.85	**0.87**
SciBERT	0.96	0.89	**0.93**	0.93	0.89	0.91	0.96	0.80	0.87	0.94	0.86	0.90
ClinicalBERT	0.89	0.89	**0.89**	0.86	0.89	0.88	0.97	0.79	0.87	0.86	0.89	0.88

Table 2: Performance across four BERT models for Task 1 comparing the effect of weighted loss and freezing.

Task 2: Event Classification

Due to increased model performance in Task 1 from using note-level over patient-level sequences as input, Task 2 models were only tuned based on performance on note-level input sequences. The performance of all 4 models, shown in Table 3, was captured by training and testing each model using default hyperparameters, default parameters with class weights, default parameters with freezing, and default parameters with both class weights and freezing.

	Default			**Default + weights**			**Default + freezing**			**Default + weights + freezing**		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BaseBERT	0.55	0.37	0.44	0.50	0.33	0.39	0.52	0.24	0.33	0.61	0.5	0.55
BioBERT	0.63	0.33	0.43	0.65	0.52	0.58	0.68	0.46	0.55	0.67	0.57	0.61
SciBERT	0.62	0.54	**0.58**	0.52	0.52	0.52	0.65	0.54	**0.59**	0.65	0.58	0.61
ClinicalBERT	0.63	0.33	0.43	0.67	0.67	**0.67**	0.65	0.43	0.52	0.64	0.63	**0.64**

Table 3: Performance across four BERT models for Task 2 comparing the effect of weighted loss and freezing.

The pretrained ClinicalBERT model with added class weights achieved the highest performance, with a 0.67 F1 score. Thus, this model was selected for hyperparameter tuning, performed through a grid search for a variety of learning rates and batch sizes, with model performances shown in Table 4. The optimal learning rate and batch size of 5e-5 and 32 respectively, were held constant while the number of epochs were tuned, shown in Table 5. The epochs parameter set at 10 and 15 yielded the same model performance, with a final F1 score of 0.71.

BS	lr = 5e-4			lr = 1e-4			**lr = 5e-5**			lr = 1e-5			lr = 5e-6			lr = 1e-6		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
8	0.0	0.0	0.0	0.59	0.63	0.61	0.6	0.59	0.59	0.45	0.59	0.51	0.48	0.59	0.53	0.5	0.33	0.39
16	0.0	0.0	0.0	0.69	0.63	0.66	0.57	0.67	0.62	0.54	0.63	0.58	0.39	0.52	0.45	0.24	0.13	0.17
32	0.0	0.0	0.0	0.67	0.67	0.67	0.70	0.72	**0.71**	0.47	0.61	0.53	0.31	0.41	0.35	0.05	0.13	0.07

Table 4: Tuning the ClinicalBERT with class weights model across various learning rates and batch sizes.

epochs = 5			epochs = 10			epochs = 15			epochs = 20			epochs = 25		
P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.52	0.72	0.61	0.70	0.72	0.71	0.70	0.72	0.71	0.71	0.70	0.70	0.71	0.70	0.70

Table 5: Tuning the ClinicalBERT with class weights model with various epochs at 5e-5 learning rate and 32 batch size.

Due to the higher testing loss versus training loss of this model, we considered overfitting to be a factor limiting model performance. Two methods were employed to correct for model overfitting: first, early stopping was implemented over a training period of 25 epochs with the condition that training would be stopped if model evaluation loss did not decrease across three consecutive epochs. Next, weight decay (L2 regularization) was implemented at various lambda values. Unfortunately, neither method was able to improve model performance.

5. Discussion

Overall, our findings confirm our hypothesis that the best performing models for both tasks were BERT models pre-trained on domain-specific data. For Task 1, SciBERT with default parameters performed best with an F1 score of 0.93, whereas ClinicalBERT with class weights performed best for Task 2 with an F1 score of 0.71. Performance for both tasks may be improved using a pre-trained BERT tokenizer for data processing; however, due to the annotations provided in our dataset, our manual conversion from BRAT to ConLL formats tokenized the clinical notes by white space. Thus, input sequences contained information that did not positively contribute to the context for classification.

The discrepancy in performance between tasks may be attributed to the nature of the two tasks (binary versus multi class classification). Compared to Task 1, Task 2 also required additional contextual information to make accurate predictions. In addition to the scope of this project, context can be added by incorporating part-of-speech (POS) or subword/character level embeddings into the training process. This may be especially helpful in identifying medical vocabulary, which relies on prefix and suffix use for drug class naming conventions.

Lastly, the Simple Transformers module used in this project allowed for a streamlined implementation of BERT models. On the other hand, using the PyTorch module would expand the available options for customizing model architecture. For example, adding dropout/maxpool layers may be a better alternative to correct overfitting since our early stopping and regularization methods were unable to improve model performance. Ultimately, our project contributed to the extraction of patient medication history from unstructured clinical notes, with the possibility for future investigations.

Appendix A. Model tuning workflow

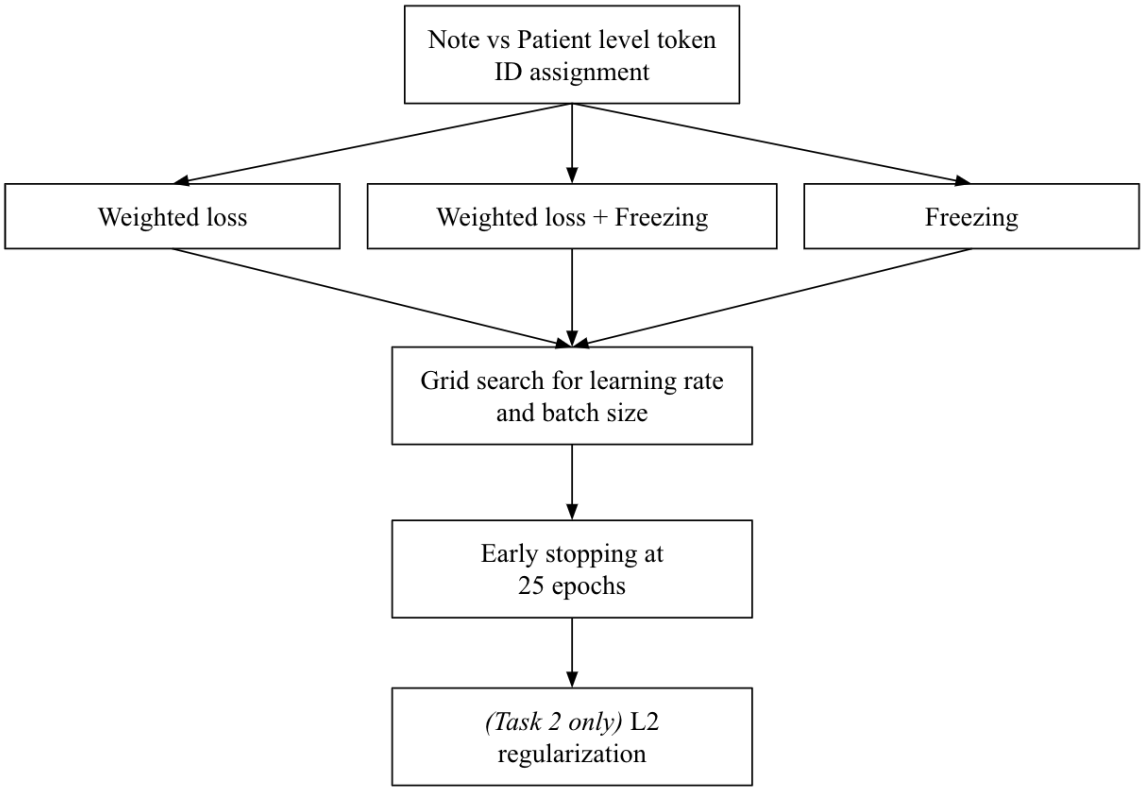

Figure 2: Overview of model building and tuning. The above workflow was used for all four models and both tasks. At each stage, the parameters resulting in the best performance advanced to the next step.

Appendix B. Additional results

		lr = 5e-4			lr = 1e-4			lr = 5e-5			lr = 1e-5			lr = 5e-6			lr = 1e-6		
Model	**bs**	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Base BERT (W+F)	**8**	0.00	0.00	0.00	0.89	0.51	0.65	0.95	0.79	**0.86**	0.89	0.66	0.76	0.77	0.72	0.75	0.52	0.70	0.60
	16	0.00	0.00	0.00	0.80	0.75	0.77	0.87	0.72	0.79	0.87	0.70	0.78	0.75	0.70	0.73	0.41	0.68	0.51
	32	0.00	0.00	0.00	0.95	0.79	**0.86**	0.94	0.72	0.82	0.77	0.70	0.73	0.68	0.64	0.66	0.65	0.33	0.43
Bio BERT (W+F)	**8**	0.00	0.00	0.00	0.85	0.75	0.80	0.95	0.75	0.83	0.87	0.83	0.85	0.95	0.83	0.89	0.71	0.72	0.72
	16	0.00	0.00	0.00	0.97	0.77	0.86	0.91	0.81	0.85	0.95	0.85	**0.90**	0.83	0.81	0.82	0.65	0.66	0.65
	32	0.03	0.21	0.05	0.89	0.85	0.87	0.93	0.81	0.86	0.84	0.81	0.83	0.83	0.83	0.83	0.73	0.34	0.46
Sci BERT (No W + No F)	**8**	0.00	0.00	0.00	0.96	0.80	0.87	0.92	0.80	0.85	0.94	0.84	0.89	0.90	0.86	0.88	0.89	0.56	0.69
	16	0.00	0.00	0.00	0.90	0.86	0.88	0.92	0.86	0.89	0.96	0.82	0.88	0.97	0.78	0.87	1.00	0.15	0.25
	32	0.00	0.00	0.00	0.96	0.89	**0.93**	0.96	0.87	0.91	0.96	0.84	0.89	0.90	0.66	0.76	0.00	0.00	0.00
Clinical BERT (No W + No F)	**8**	0.00	0.00	0.00	0.95	0.43	0.59	0.95	0.81	0.87	0.91	0.85	0.88	0.88	0.81	0.84	1.0	0.23	0.38
	16	0.00	0.00	0.00	0.96	0.89	**0.92**	0.97	0.70	0.82	0.93	0.81	0.86	0.95	0.79	0.86	0.00	0.00	0.00
	32	0.56	0.58	0.57	0.89	0.89	0.89	0.98	0.87	**0.92**	0.90	0.79	0.84	0.96	0.57	0.72	0.00	0.00	0.00

Table 6: Tuning learning rate and batch size. W and F indicate the application of weighted loss and freezing, respectively.

References

Mahajan, D., Liang, J. J. & Tsou, C.-H. Toward Understanding Clinical Context of Medication Change Events in Clinical Narratives. AMIA Annu Symp Proc 2021, 833–842 (2022).

Turchin, A., Shubina, M., Breydo, E., Pendergrass, M. L. & Einbinder, J. S. Comparison of information content of structured and narrative text data sources on the example of medication intensification. J Am Med Inform Assoc 16, 362–370 (2009).

Jensen, P. B., Jensen, L. J. & Brunak, S. Mining electronic health records: towards better research applications and clinical care. Nat Rev Genet 13, 395–405 (2012).

Meystre, S. M. et al. Heart Failure Medications Detection and Prescription Status Classification in Clinical Narrative Documents. Stud Health Technol Inform 216, 609–613 (2015).

Liu, M. et al. Modeling Drug Exposure Data in Electronic Medical Records: an Application to Warfarin. AMIA Annu Symp Proc 2011, 815–823 (2011).

Liu, F. et al. Learning to detect and understand drug discontinuation events from clinical narratives. J Am Med Inform Assoc 26, 943–951 (2019).

Wolf, T. et al. Transformers: State-of-the-Art Natural Language Processing. in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations 38–45 (Association for Computational Linguistics, 2020). doi:10.18653/v1/2020.emnlp-demos.6.

Zhu, Y. et al. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. arXiv:1506.06724 [cs] (2015).

Rajapakse, T. NER Model. Simple Transformers https://simpletransformers.ai/docs/ner-model/ (2022).

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] (2019).

Lee, J. et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics btz682 (2019) doi:10.1093/bioinformatics/btz682.

Beltagy, I., Lo, K. & Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. arXiv:1903.10676 [cs] (2019).

Alsentzer, E. et al. Publicly Available Clinical BERT Embeddings. arXiv:1904.03323 [cs] (2019).

Contribution Statement

Initial data cleaning and tokenization was completed by both authors. Model building and tuning for Task 1 was performed by Krutika Pandit. Model building and tuning for Task 2 was performed by Jonathan So.