5-2023

Performance Analysis Of Attention Based Deep Learning Models On Named Entity Recognition In Electronic Health Records

Tariq Abdul Quddoos

PERFORMANCE ANALYSIS OF ATTENTION BASED DEEP LEARNING MODELS ON NAMED ENTITY RECOGNITION IN ELECTRONIC HEALTH RECORDS

A Thesis

by

TARIQ ABDUL-QUDDOOS

Submitted to the Office of Graduate Studies of
Prairie View A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2023

Major Subject: Electrical Engineering

PERFORMANCE ANALYSIS OF ATTENTION BASED DEEP LEARNING MODELS ON NAMED ENTITY RECOGNITION IN ELECTRONIC HEALTH RECORDS

A Thesis

by

TARIQ ABDUL-QUDDOOS

Submitted to the Office of Graduate Studies of
Prairie View A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Dr. Lijun Qian
Chair of Committee

Dr. Xishuang Dong
Committee Member

Dr. Xiangfang Li
Committee Member

Dr. Richard Wilkins
Committee Member

Dr. Annamalai Annamalai
Head of Department

Dr. Pamela Obiomon
Dean, Roy G. Perry College
of Engineering

Dr. Tyrone Tanner
Dean, Graduate Studies

May 2023

Major Subject: Electrical Engineering

ABSTRACT

Performance Analysis of Attention Based Deep Learning Models on Named Entity Recognition in Electronic Health Records

(May 2023)

Tariq Abdul-Quddoos

Chair of Advisory Committee:

Dr. Lijun Qian

Mining Clinical Notes for relevant information has attracted a lot of interest in Natural Language Processing (NLP). Medical documents contain language whose distributions vary from that of the general domain and have a vocabulary that evolves with time. Recently, attention based deep learning language models have become the new state-of-the-art in language modeling capturing strong representations of language with respect to the context it is in, improving on classic clinical NLP task such as medication detection, and medication classification.

In this thesis research, the Harvard Medical School's 2022 National Clinical NLP Challenges (n2c2) is considered where the Contextualized Medication Event Dataset (CMED) has been given for the challenge. CMED is a dataset of unstructured Electronic Health Records (EHRs) and annotated notes that contain task relevant information about the

EHRs. The goal of the challenge is to develop effective solutions for extracting contextual information related to medications from EHRs using data driven methods. In this thesis, variations of Google's attention-based Bert architecture have been applied for this challenge, namely, Bert Base, BioBert, and two variations of Bio+Clinical Bert, that are pre-trained on general domain, biomedical domain, and clinical domain corpora, respectively. They are used to perform named entity recognition (NER) for medication extraction and medical event detection. Pre-processing methods have been developed for breaking down EHRs for compatibility with the Bert model on NER task, and the variations of Bert are fine-tuned with CMED for the n2c2 task. Performance analysis has been carried out using a script based on constructing medical terms from the evaluation portion of CMED with metrics including recall, precision, and F1-Score. The results demonstrate that Bio+Clinical Bert outperforms Bert Base and BioBert, as well as three of the top ten performers in the challenge.

Index terms: Bi-directional encoder representations from transformers, electronic health records, natural language processing, transformer

ACKNOWLEDGMENTS

This research work is supported by NASA under award number 80NSSC22KM0052, and by NSF award 2205891 and 2018945. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA or NSF.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

-

LIST OF ABBREVIATIONS

- BERT - Bi-directional Encoder Representations from Transformers
- BiLSTM - Bidirectional Long Short Term Memory
- CMED - Contextualized Medication Event Dataset
- CRF - Conditional Random Field
- EHR - Electronic Health Record
- GPT - Generative Pre-Trained Transformers
- MLM - Masked Language Modeling
- N2C2 - National Clinical NLP Challenges
- NER - Named Entity Recognition
- NLP - Natural Language Processing
- RNN - Recurrent Neural Network

CHAPTER 1

INTRODUCTION

The digitization of medical information sources has become widely adopted allowing for multiple data sources to be integrated together and ease of data sharing amongst multiple parties. An EHR is one of these digitized sources and is defined as a longitudinal electronic record of patient health information generated by encounters in any care delivery setting, including information on patient demographics, progress notes, problem lists, medications, vital signs, past medical history, immunizations, laboratory data, and radiology reports [6]. A 2017 survey of patient registries in the United States by the National Quality Registry Network found that 68 percent of registries extract some data from EHRs [7], with the increased use of EHR's data driven methods have become of interest amongst researchers for extracting relevant information from them.

Natural Language Processing (NLP) is a field where data driven methods meet linguistics and NLP researchers have sought to utilize machine learning to model and mine EHRs. Machine learning serves to enhance the use of EHRs placing the burden of information extraction on models rather than people. Models have shown success

This thesis style is in accordance with IEEE Journal of Biomedical and Health Informatics in extracting information from EHRs with classic tasks such as entity recognition, question answering, and context classification. NLP is also a field that is quickly progressing giving researchers access to more powerful tools for application on EHRs.

In this work, information extraction tools on EHRs are advanced by applying variations of Bert, an attention based deep learning architecture for medication detection and medication context classification as part of Track 1 of the 2022 National Clinical NLP Challenges(N2C2). For this challenge the Contextualized Medication Event Dataset (CMED) had been released and is a dataset capturing relevant context needed to understand medication changes in clinical narrative [8], containing EHRs and annotated notes on the EHRs with task relevant information.

1.1 National Clinical NLP Challenges 2022

The N2C2 focuses on the study of applying data driven approaches to mining clinical information. Track 1 of the 2022 N2C2 tasked researchers with developing solutions for generating information related to the context of medication mentions in EHRs using data driven methods. A description of each challenge is shown in Section 1.1, with this work covering both task 1 and 2. Track 1 of the 2022 National Natural Language Processing Clinical Challenges (N2C2) focused on data driven approaches to identifying timelines relating to medication changes. There were 32 teams from 19 countries with a total of 211 submissions for this track.

1.1.1 Task 1: Medication Detection. Task 1 for the 2022 N2C2 tasked researchers with identifying medication mentions in EHRs. This task has been well studied by NLP researchers with it also being a task in the 2018 N2C2. Knowing what medications and where they are in EHRs are an essential step with unstructured EHRs for further medication related information extraction.

1.1.2 Task 2: Medication Event Classification. Task 2 for 2022 N2C2 tasked researchers with classifying identified medication mentions from task 1 as having events associated with them. An event refers to any change that has to do with a particular mediation within its context. The three classes for this task are given as disposition, no disposition, and undetermined. Disposition refers to an event occurring with the associated medication, no disposition refers to no event occurring for the associated medication, and undetermined refers to if annotators cannot determine if an event has occurred or not [9].

1.1.3 Task 3: Multi-dimensional Context Classification.

Task 3 for 2022 N2C2 is not covered in this work but is discussed for a full view of the goal of the competition and its relevance to future work. Researchers are tasked with classifying the context of medications that have been labeled with disposition in task 2. The context is classified across 4 dimensions, those dimensions being action, temporality, certainty, and actor. Action refers to type of change being discussed, temporality refers to when the change occurred, certainty refers to if a change was implemented or just discussed, and actor refers to who initiated the change [9].

1.2 Problem Statement

The utilization of deep learning for EHR language modeling has been of interest among researchers for some time, with the N2C2 holding numerous challenges around applied data driven approaches to clinical text mining. Capturing a machine understanding of EHR language is a complex problem due to the vocabulary distribution, numerous language syntax structures, and evolving vocabulary meanings in medical domain text. Language models have shown strong performance on classic task in EHRs such as entity recognition, entity classification, and question answering by capturing contextual information and including domain specific knowledge in word embedding and modeling methods. Recurrent Neural Network (RNN) based architectures have been a popular method in the past for modeling EHR due to their auto-regressive nature or in the case of the RNN variant Bi-LSTM, their ability to capture bi-directional context, but suffer from loss of information as input examples increase with length. This has been overcome by attention-based language models such as GPT and BERT. Attention based models have become the state-of-the-art in language modeling due to their ability to capture full contextual information without the restriction on context length. Strategies have also been developed for pre-training these models, so they are able to capture general representations of language distributions and then be fine-tuned for a given downstream task. In this work variations of the Bert model were applied to accomplish task 1 and task 2 as listed in Sections 1.1.1 and 1.1.2. Variations of Bert pre-trained on general, biomedical, and clinical domain corpora were applied to CMED, using Named Entity Recognition (NER) on both task. This study explored the performance of attention-based models on EHR data along with the differences in performance with regard to pre-training corpora. Pre- and post- processing methods were also of interest in this study given the wide range of syntactic structures used in EHRs, it can be difficult to judge how best the process an entire record.

1.3 Contributions

The contributions of this research are as itemized below:

1. Apply Bert model on EHR dataset for medication related information extraction, with pre-trained variations of Bert applied to see significance of pre-training corpora on fine-tuning task.

2. A challenge with unstructured electronic health records is that they hold a large variation of language syntax, making data cleaning and structuring a challenge. Processing methods were developed for structuring EHRs in formats compatible and effective with the Bert model and required competition evaluation formats.

3. Named Entity Recognition is a popular method for extracting names of entities such as a city, car, or in the case of task 1 medications from text. This method is also extended to task 2, where medications are recognized by their event classes listed in section 1.1.2.

1.4 Outline of the Thesis

The remaining part of this study is structured as follows: Chapter 2 contains the Literature review where relevant research works in the recent past are discussed. The methods used in this study are described in Chapter 3. The data used is also described in this chapter. Chapter 4 provides a description of all the experiments carried out in this research, along with a comparison between the results in this work and the top performing teams from the 2022 N2C2 competition. Chapter 5 concludes this study by summarizing the contributions of this work and highlighting potential research to further build on this work.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

Understanding medication events in clinical narratives is essential to achieve a complete picture of a patient's medication history. A complete picture of medication history is important for health care providers to decide on the proper steps for treatment, identify medication related symptoms, and plan for future treatments [9]. Track 1 of the 2022 National Natural Language Processing Clinical Challenges(N2C2) focused on data driven approaches to identifying timelines relating to medication changes. There were 32 teams from 19 countries with a total of 211 submissions for this track [5], including some of the results shown in this work. The N2C2 focuses on the study of applying data driven approaches to mining clinical information. The two previous challenges were held in 2018 and 2019, the first in 2018, focused on Cohort Selection for Clinical Trials(Track 1) [10] and Adverse Drug Events and Medication Extraction in EHR's(Track 2) [11]. The second in 2019 focusing on Clinical Semantic Textual Similarity(Track 1), Family History Extraction(Track 2), Clinical Concept Normalization(Track 3), and Novel Data Use(Track 4). The N2C2 is an outgrowth of the Informatics for Integrating Biology(i2b2) center which also held clinical related data driven challenges from 2004-2014. The clinical notes dataset CMED used in this study are a portion of the data from the 2014 i2b2/UTHealth Natural Language Processing shared task [12].The 2014 i2b2 corpus is longitudinal corpus of 1304 records representing 296 diabetic patients. The corpus contains three cohorts: patients who have a diagnosis of coronary artery disease (CAD) in their first record and continued to have it in subsequent records; patients who did not have a diagnosis of CAD in the first record, but developed it by the last record and patients who did not have a diagnosis of CAD in any record [12]. Other popular clinical corpora are the Mimic II [13] and Mimic III [14] clinical databases which are collections of nursing notes and discharge summaries gathered from ICUs in the Beth Israel Deconess Medical Center. Another corpora is the THYME corpus and it is a collection of over 1200 notes from the Mayo Clinic, representing patients from the oncology department, specifically those with brain or colon cancer [15].

2.2 Electronic Health Records Language Modeling

Language modeling of unstructured electronic health records has become a topic of interest amongst NLP researchers. EHRs contain language whose distribution varies from that of the general domain and have a vocabulary that evolves as the medical field evolves, making language modeling a challenge. A combination of word embedding methods and modeling choices have shown to overcome this challenge and achieve strong results on EHR text mining task. In the past recurrent neural networks (RNN) were a popular choice for modeling EHRs and a general choice for many language modeling task. RNNs are a class of neural networks used for modeling sequential data that share parameters across the model [16], making them ideal for language modeling due to the sequential nature of language.

Track 2 of the 2018 N2C2 like track 1 of the 2022 N2C2 involved medication related information extraction and the RNN variant Bidirectional long short-term memory with Condition Random Fields (BiLSTM CRF's), shown in Fig. 2.1 was a popular modeling choice, with 9 of the top teams incorporating them in their system. Conditional random fields (CRF's) in general were extremely popular, with every top-performing team incorporating them in their system [17]. BiLSTM-CRFs use a BiLSTM to create a series of state representations that are then used as in- put into a CRF for labeling [17]. The typical modeling methods used named entity recognition (NER) for labeling and structuring data, where NER is a sub-problem of information extraction and involves processing structured and unstructured documents and identifying expressions that refer to peoples, places, organizations, companies [18] or in this case, medications. A popular method for doing supervised learning with NER is classifying words using the BIO tag format. For each entity class of interest, B-classname would be the label for the beginning of an entity name, I-classname is a label for any word inside the entity class name after the first. O, standing for outside of entity, is the label for any words that do not fit into an entity class of interest.

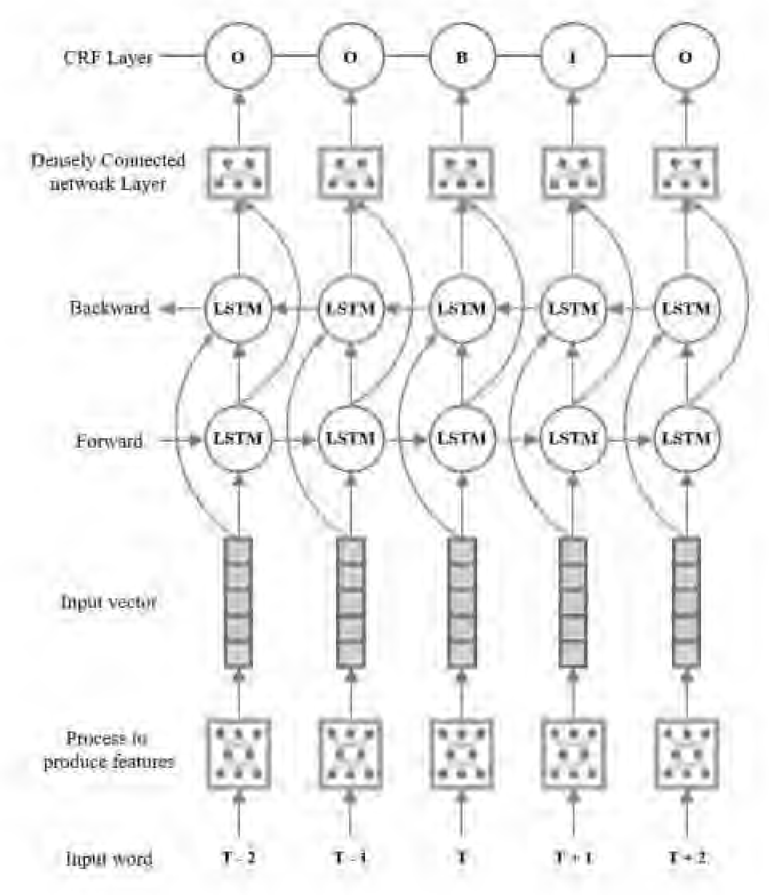


Fig. 2.1. Bi-LSTM CRF Example. Adapted from [1].

For embedding methods most of the top-ranked teams in the 2018 N2C2 utilized the entire MIMIC-III dataset to create pre-trained word embeddings with the Word2Vec package [11]. The Word2Vec package uses methods that are efficient for obtaining dense static embeddings using self-supervised methods, with code and pre-trained embeddings available online [19]. One Word2Vec method for computing embeddings is the skip-gram algorithm and another is fast text. Dai et al. [11] applied a number of different embeddings methods in the 2018 N2C2, with two being the Word2Vec methods. Two more embedding methods used were variations of GloVe embeddings another supervised learning method, the first being ConcatedVec, a concated version of GloVe embeddings and the other being PurifiedVec, which are GloVe embeddings encoded using principal component analysis [11].

Recently attention based models have found wide success in EHR language modeling with architectures such as Generative Pre-Trained Tranformers (GPT) and Bi-directional Encoder Representations from Transformers (BERT). Libbi et al. [20] had applied the GPT varaint GPT2 for generating synthetic EHR data as a solution to overcoming privacy concerns with EHRs [20]. Laws pertaining to patient privacy make releasing clinical notes difficult, and as a result there are relatively few datasets of these notes available to researchers who are not affiliated with medical facilities [12]. EHR's static embeddings such as Word2Vec methods find limitation with evolving vocabulary due to new words being Out of Vocabulary (OOV) or old words adopting an evolved meaning. Attention models such as Bert, solve this using wordpiece embeddings, where any word can be represented as an appended sum of wordpieces and where each wordpiece has a single numerical ID. Attention models and wordpiece embeddings have become the state-of-the art in EHR language modeling and are further examined in depth in the next section.

2.3 Attention Enhanced Language Models

Attention based models have become a standard tool in the deep learning toolkit and the new state-of-the art in language modeling. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors, the output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key [2]. A standard neural network consist of a series of non-linear transformation layers, where each layer produces a fixed-dimensional hidden representation, for tasks with large input spaces. This paradigm makes it hard to control the interaction between components [21]. In the case of language modeling with RNNs, they suffer from a loss of information when input sequence lengths become too long. The transformer is a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output [2]. The transformer architecture is shown in Fig. 2.2, originally applied in language translation, consisting of an encoder and decoder.

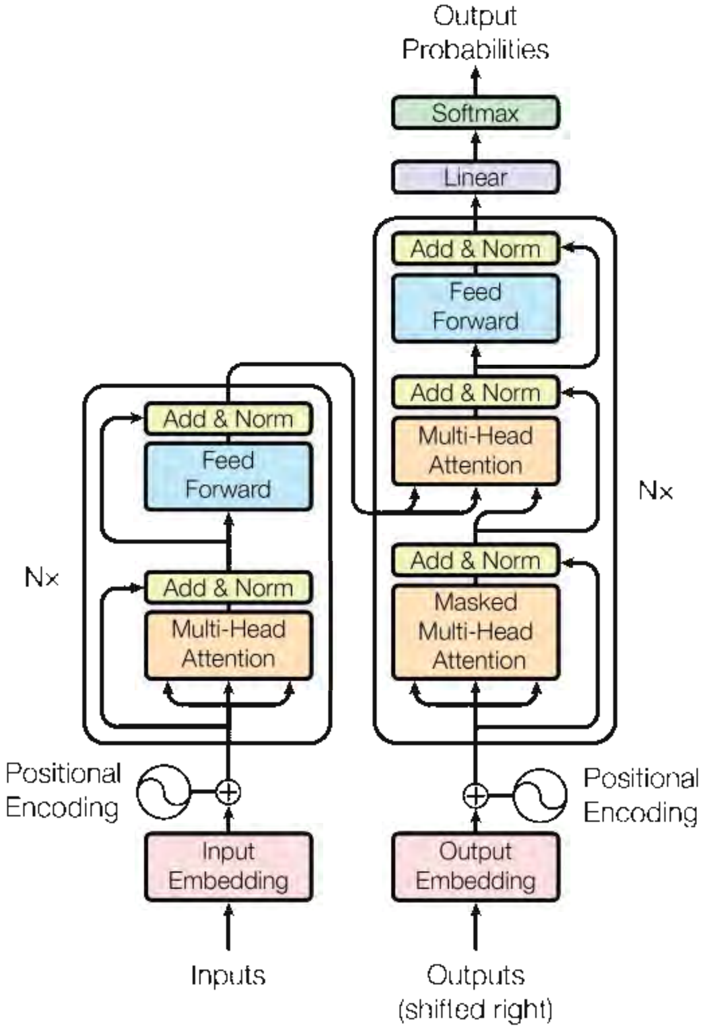


Fig. 2.2. Transformer Architecture. Adapted from [2].

Transformers utilize an attention method called Self-attention, also called intra- attention which is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [2]. This attention mechanism is carried mathematically using a dot-product attention method shown in Fig. 2.3, where Q stands for the query, K for the key, and V for the value. When multiple of these intra-attention computations are done on separate linear transformations of the same input sequence, this is called multi-head attention where the number of heads correspond to the number of intra-attentions done.

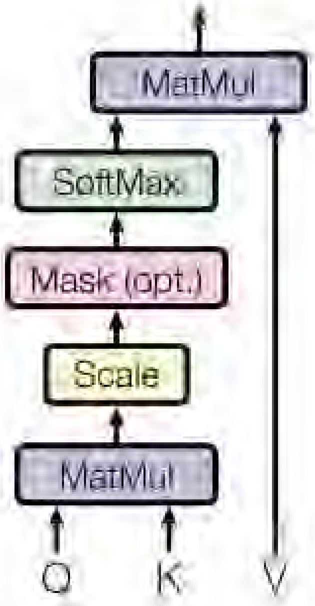


Fig. 2.3. Dot Product Attention. Adapted from [2].

An example of the outcome of a trained multi-head attention with 2 attention heads on a sentence is shown in Fig. 2.4. The darker purple lines indicate that the attention mechanism recognizes a stronger connection between the words in the context they are in.

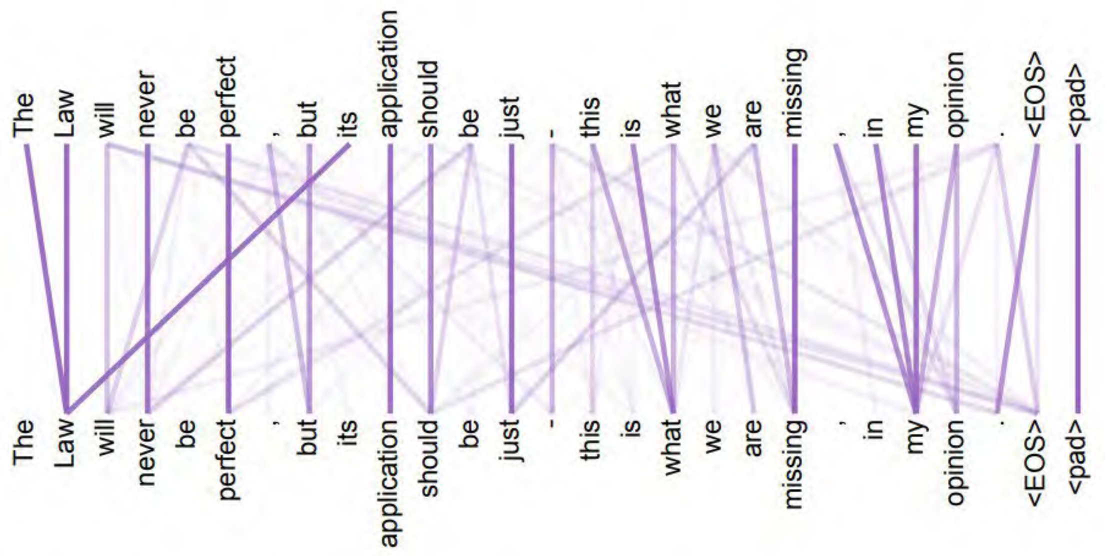


Fig. 2.4. Applied Attention on Sentence Example. Adapted from [2].

Many of the popular state-of-the-art large language models are built using the transformer as the base architecture. GPT makes use of multiple transformer de- coders and like the RNN variant Long-Short Term Memory (LSTM), GPT is a single direction auto-regressive model, with the ability to capture long range dependencies and has become a popular choice for text generation [22]. BERT's model architecture makes use of multiple transformer encoders and like a Bi-LSTM can capture bi-directional dependencies as shown in Fig.2.5. Bert's ability to capture bi-directional long-range dependencies has made it a popular choice for tasks such as NER and next sentence prediction and is the applied model is this work on an NER task.

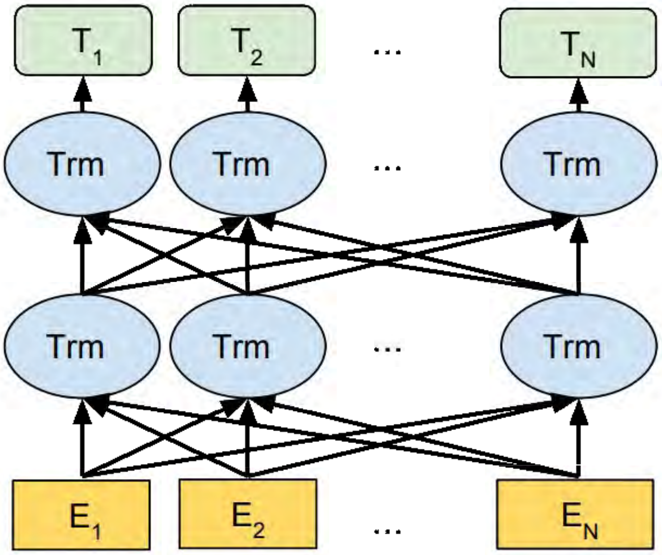


Fig. 2.5. Bert Model Architecture. Adapted from [3].

Bert is also a pre-trained model that can be easily fine-tuned by switching the output layer once Bert weights are initialized in pre-training to one suitable for the applied task. This pre-training allows Bert to capture language distributions, significantly reducing training time for researchers and increasing performance on various tasks. The first task Bert is pre-trained on is Masked Language Modeling (MLM) also called the Cloze task. The Cloze procedure may be defined as a method of mutilating language patterns by deleting parts, and administering it to ”receivers” (readers or listeners) with them attempting to make the patterns whole again [23]. For the use of MLM on Bert some percentages of the input tokens are masked at random, and then those masked tokens are predicted [3]. The second task Bert is pre-trained on is next sentence prediction. This is done because many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the relationship between two sentences, which is not directly captured by language modeling [3].

The applied embedding method for Bert is wordpiece embeddings where words are split into wordpieces based on a pre-defined vocabulary and those wordpieces are converted to a representative numerical ids for modeling. Wordpiece models are generated using a data-driven approach to maximize the language-model likelihood of the training data, given an evolving word definition [24]. Given a training corpus and a number of desired tokens D, the optimization problem is to select D word- pieces such that the resulting corpus is minimal in the number of wordpieces when segmented according to the chosen wordpiece model [24]. An example of wordpiece tokenization and embeddings are shown below:

- Words: Jet makers fued with big orders at stake
- Wordpieces: J ##et makers fe ##ud with big orders at stake
- Embedding Array: [[13784, 10293] [12525] [175, 17226] [1114] [1992] [3791] [1120] [8219]]

Pre-trained Bert models also incorporate special tokens into their embedding vocabulary for making it easier to accomplish a number of downstream tasks. The first token of every sequence is always a special classification token ([CLS]), the final hidden state of Bert corresponding to this token is used as the aggregate sequence representation for classification tasks [3]. Another token is the [SEP] token, used to identify the boundary between two sentences in the same input example. Some other special tokens used are the [MASK], [UNK], and [PAD] tokens.

CHAPTER 3

METHODOLOGIES

The machine learning pipeline consists primarily of a text pre-processing stage, a modeling stage, and a prediction post-processing stage. During pre-processing the EHR text is tagged with a BIO scheme for NER and processed to a form compatible with the Bert architecture. In the modeling stage, the model gives tag probabilities for words in the EHR text. In the post-processing state, predictions are processed in a format compatible with the evaluation methods used. Pipelines were done with both Tensorflow 1 and PyTorch with their processing slightly differing so the following section will focus on the PyTorch pipeline, since both tasks are implemented with PyTorch and only task 1 with TensorFlow.

3.1 Pre-Processing

The pre-processing pipeline is shown in figure 3.1 and in the following order consist of annotation parsing, EHR tagging, EHR sentence/section segmentation, word tokenization, and wordpiece tokenization. All pre-processing is done with Python and is largely rule-based with the exception of sentence segmentation, word tokenization, and wordpiece tokenization being done using libraries with pre-trained models.

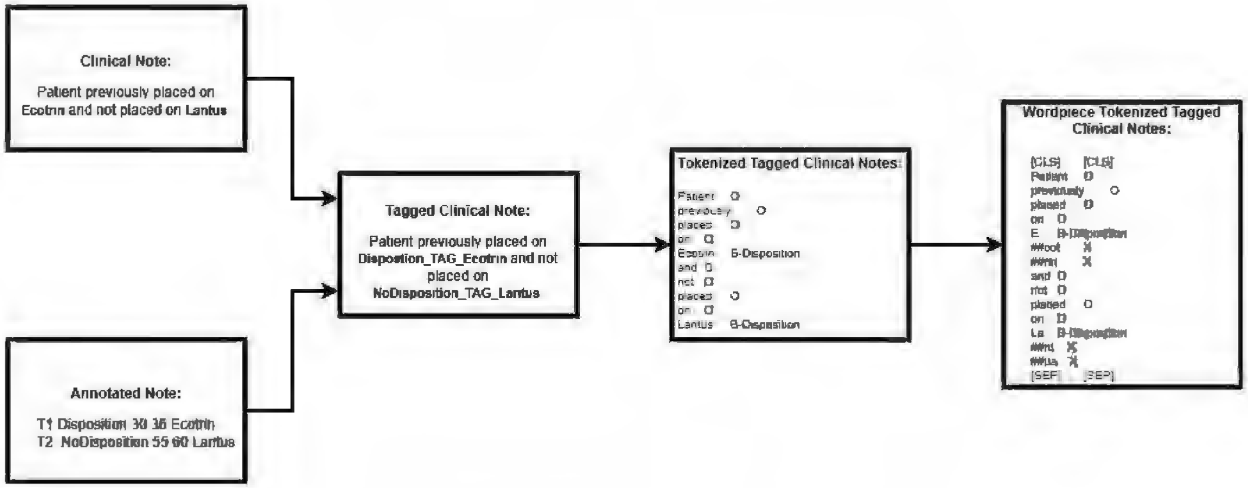

Clinical Note:
Patient previously placed on Ecotrin and not placed on Lantus
Annotated Note:
T1 Disposition 30 36 Ecotrin
T2 NoDisposition 55 60 Lantus
Tagged Clinical Note:
Patient previously placed on Disposition_TAG_Ecotrin and not placed on NoDisposition_TAG_Lantus
Tokenized Tagged Clinical Notes:
Patient O
previously O
placed O
on O
Ecotrin B-Disposition
and O
not O
placed O
on O
Lantus B-Disposition
Wordpiece Tokenized Tagged Clinical Notes:
[CLS] [CLS]
Patient O
previously O
placed O
on O
E B-Disposition
##cot X
##rin X
and O
not O
placed O
on O
La B-Disposition
##nt X
##us X
[SEP] [SEP]

Fig. 3.1. Pre-Processing Pipeline.

Each EHR is accompanied by an annotated file with task relevant information such as medication names, medication character position, and medication event class. For both task 1 and 2 the annotations have the following format:

- (Term Number), (Event Tag), (First Char Index), (Last Char Index), (Medication)

Annotation parsing is done to create an ordered numpy array of all medications from a single EHR with each row of the array in the following format:

- (First Char Index, Last Char Index), (Event Tag), (Medication)

All punctuation and spaces in a medication are replaced by the "_" character. The numpy array is placed in order of largest first char index to smallest first char index for ease of tagging in the EHR. EHR tagging is done by placing the entity tag directly in the unstructured

EHR. The medication in the EHR are replaced by tagged medications in the following format: tag TAG Medication, where tag refers to the actual entity tag (Example: Drug for task 1 or Disposition for task 2) and the string " TAG " is used as a flag to indicate that this word is an entity of interest from the annotation, an example is shown below:

- EHR /w No Tag: Patient placed on Antibiotics for the next 2 months.
- EHR /w Tag: Patient placed on Drug TAG Antibiotics for the next 2 months.

The parsed annotation is contained in a numpy array ordered from largest to smallest character index so medications that appear last in the EHR are tagged first, that way character positions of other medications that need to be tagged in the EHR are not changed. When tagging medications in the order they appear in the EHR, a count is needed for how many extra characters have been added to the EHR after tagging, starting from the last medications eliminates that need. The tagged EHRs are then segmented using a combination of rules and models from the Punkt package. The EHR is first separated into sections based on three or more new line characters. The Punkt sentence tokenizer is then applied on the sections, which uses a language-independent, unsupervised approach to sentence boundary detection. It is based on the assumption that a large number of ambiguities in the determination of sentence boundaries can be eliminated once abbreviations have been identified [25]. Although functioning for this work, the punkt tokenizer applied is a general domain tokenizer, not capturing all the different syntax structures held in an EHR. Each sentence returned by the tokenizer is an input example for modeling. For each EHR an additional input example is added to the beginning of each list of examples containing information on the EHR record id, this record id is used during post-processing and will be further discussed in the post-processing section (3.3). Each sentence is then tokenized using the Punkt word tokenizer and all punctuation is filtered out. All stop words are kept due to the embedding method being wordpiece embeddings, where most stop words are only representedby a single numerical ID and in the interest of keeping all sentence context for the attention mechanism applied in the model. All words are then tagged using the BIO format for NER, with this method being applied for both task 1 and task 2. The tags for task 1 after applying the BIO tagging were B-Drug, I-Drug, and O. For task 2 the BIO tags were B-Disposition, I-Disposition, B-NoDisposition, I-NoDisposition, B-Undetermined, I-Undetermined, and O. All word tokens were further tokenized into wordpieces, with each applied variation of Bert having its own pre-trained wordpiece tokenizer. A [CLS] token (Classification token) is added to the beginning of each sequence and a [SEP] (Sentence separator token) is added to the end of each sequence. [PAD] tokens are then added to each after the [SEP] token to meet the desired embedding length, for this work all embedding lengths are set to 512.

3.2 Modeling

The Bert architecture captures strong representations of text using a self-attention based deep learning approach. This approach has shown to overcome the limitations of RNN based architectures that have a harder time capturing dependencies when they are too far apart. The modeling pipeline for both tasks can be seen in Fig. 3.2, consisting of the Bert input layer from Fig. 3.3, a Bert encoder, a feed-forward network, and a softmax layer.

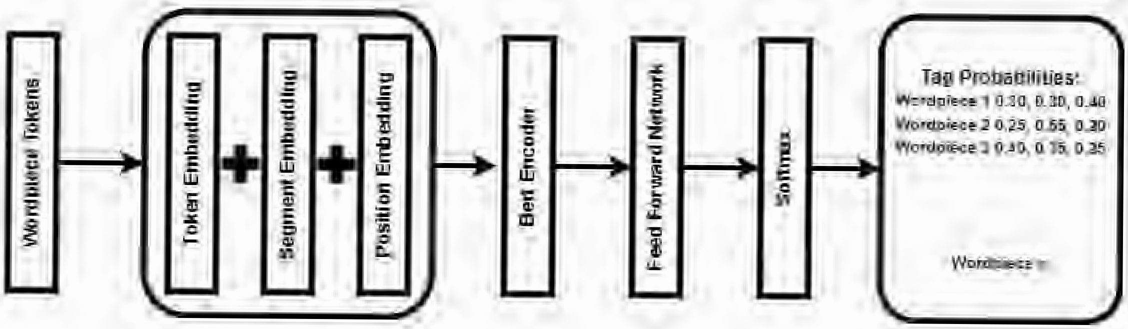


Fig. 3.2. Modeling Pipeline.

The modeling pipeline for all three tasks consist of a pre-trained Bert encoder with a feed-forward network or in the case of task 3 a Support Vector Machine for decoding into predictions. All Bert encoders follow the Bert Base architecture containing a total of 110M parameters, made up of 12 transformer blocks each with 12 self-attention heads and hidden size for all layers in the model is 768 [3].

All wordpiece sequences are encoded to their input ids, where the input ids are a single numerical id for each wordpiece. The input ids (token embeddings) are summed together with segment embeddings, and positional embeddings. The segment embeddings are used to separate different sentences in the same sequence where each sentence would have a different segment embedding, in the case of this work all segment embeddings are the same. The positional embeddings are used since the Bert model contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, some information must be injected about the relative or absolute position of the tokens in the sequence [2]. A linear layer is also applied before the Bert encoder for the models hidden size of 768 to be met.

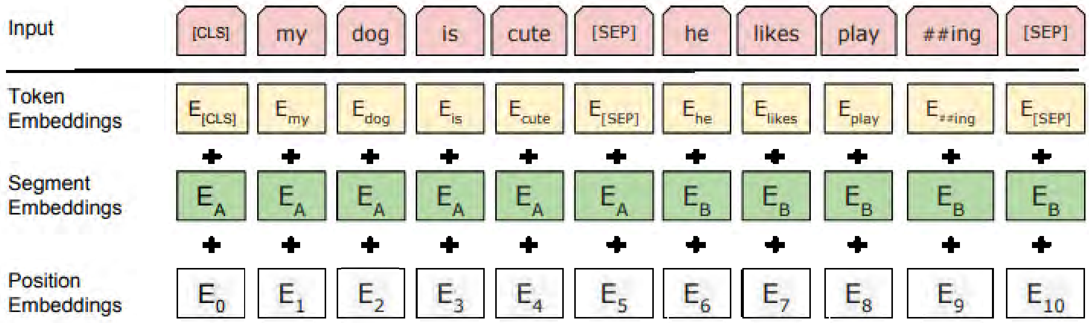


Fig. 3.3. Bert Input Layer. Adapted from [3].

The hidden layers consist of only the Bert encoder and output layer consist of a feed-forward neural network. The Bert encoder for a single training example gives a matrix with dimensions: (embedding length x 768). The feed-forward network consists of two layers, the first with a hidden size of 256 and the second with a hidden size equivalent to the number of tags for the task. The feed-forward network's output goes into a SoftMax layer for tag probabilities, with the SoftMax returning 3 probabilities for task 1 and 7 probabilities for task 2, the tag with the highest probability is used as the prediction.

3.2.1 Bert Base. The Bert base model applied is pre-trained on general domain corpora keeping the case of all words so casing is kept for the fine-tuning done with CMED. The model was pre-trained on the following general domain corpora:

- BooksCorpus [26] - 800M Words
- English Wikipedia [27] - 2.5M Words

Pre-training was done on 4 cloud TPU's in pod configurations (16 TPU chips total) over 4 days [3].

3.2.2 BioBert.

BioBERT is the first domain-specific Bert based model pre- trained on biomedical data [4]. The applied model is pre-trained with all lower case words so fine-tuning on CMED is done the same way. BioBert uses the previously explained Bert base model weights and does additionally pre-training on the following biomedical corpora:

- PubMed Abstracts [4] - (4.5B Words)
- PMC Full-text articles [4] - (13.5B Words)

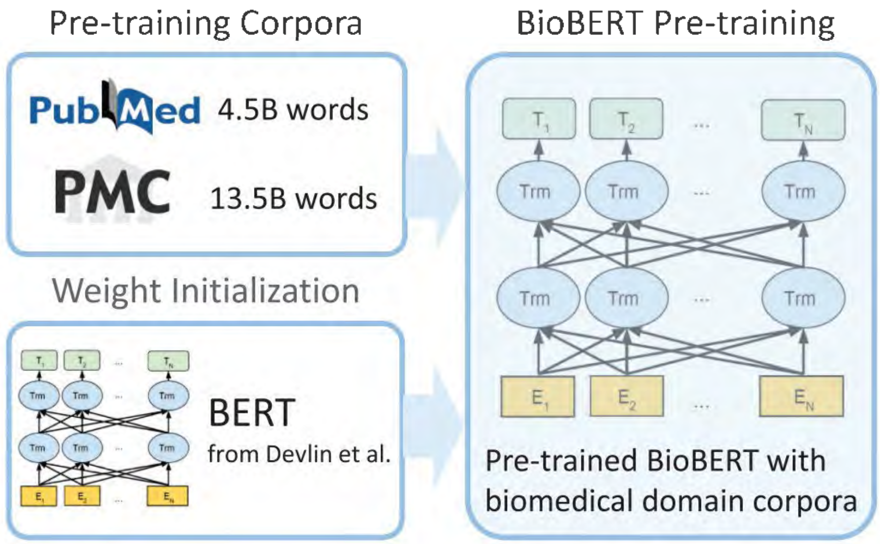


Fig. 3.4. BioBERT Pre-training. Adapted from [4].

3.2.3 Bio+Clinical Bert.

Bio+Clinical Bert [28] is another

domain specific Bert base model pre-trained on clinical notes. The applied model is pre-trained with all lower case words so fine-tuning on CMED is done the same way. Bio+Clinical Bert uses the previously explained BioBert model weights and undergoes further pre-training on the following clinical corpus.

- MIMIC-III [14] v1.4 database (2M Clinical Notes)

There are two variations of Bio+Clinical Bert, one trained on all MIMIC-III notes and another only on MIMIC-III discharge summaries. These will be referred to as Bio+Clinical Bert All Notes and Bio+Clinical Bert Discharge Notes. Pre-training was done on a single GeForce GTX TITAN X 12 GB GPU over 18 days [28].

3.3 Post-Processing

The post-processing pipeline is shown in Fig. 3.5 and consist of word reconstruction from word pieces and placing predictions in an annotated format for compatibility with an evaluation script released for the 2022 N2C2 competition.

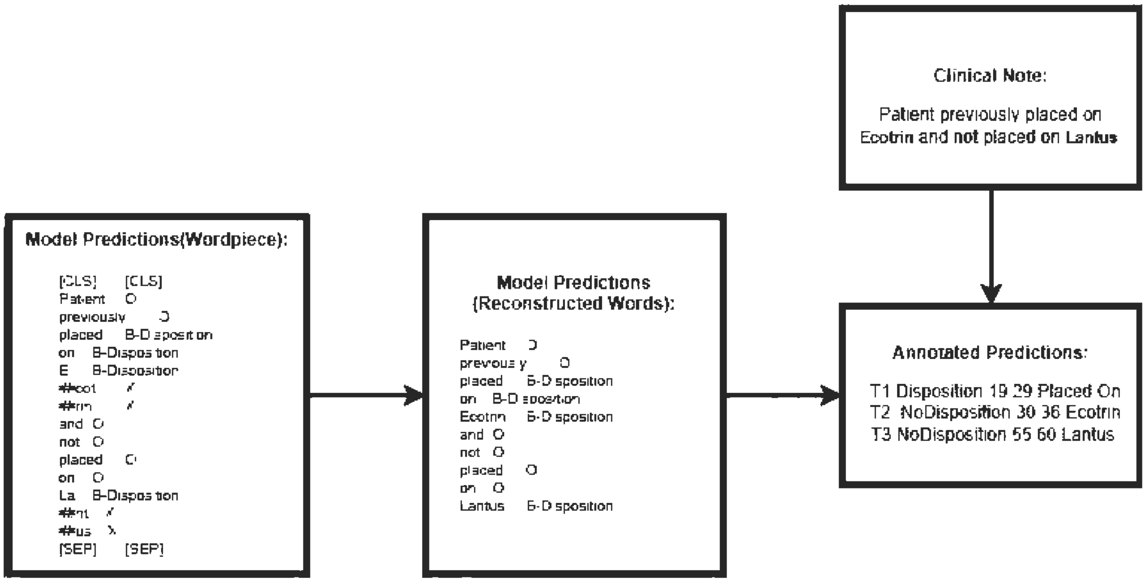


Fig. 3.5. Post Processing Pipeline.

Word reconstruction was done by identifying wordpieces with a ”##” string at the beginning of them and appending them to the previous wordpiece, that string is the identifier for a wordpiece that comes after the first wordpiece in a single word. Only the prediction from the first wordpieces was used to classify the entire word but strategies for using all wordpieces for prediction is of interest. Evaluation was done with on the predicted NER tags and an official evaluation script was provided by the 2022 N2C2 that evaluates predictions with respect to their character positions in the EHRs. NER evaluation was done for a view of model performance from only the tag perspective, before additional post-processing was done to construct entire medication names. For NER evaluation on the test data only word reconstruction was needed, and their respective predicted tags are compared with the ground truth tags. For evaluation on the test data using the script provided by the 2022 N2C2 each detected medications is placed in the same annotated format as the training annotation files are in:

- (Term Number), (Event Tag), (First Char Index), (Last Char Index), (Medication)

For task 1 instead of an event tag the word Drug would be in its place. All words are in the order they are in as they appear in their respective EHRs so they are all placed into a list, along with their predicted tags. First the record ID token is searched for to identify what EHR the following words belong to and an .ann file is created for that specific EHR. The list is then searched through, looking for the B- tag for the beginning of an entity and if an I- tag follows than the word with that tag is appended to the previous word with a space in between. The EHR is then searched through to find the first and last character positions of the identified medication. For avoidance of giving medications with the same names the same character position's, after a medication is found in the EHR, the next search will begin one character after that medications last character position. This process is continued until another record ID token is found, which will prompt the system to close the current annotation file and begin a new one. A flaw in this system is that there is no method for recovering punctuation as all are removed during pre-processing. Moreover, if a word is the same as a medication~~s~~ but not considered a medication, its character position could be used if it appears before the actual medication and after the previously found medication. This is troublesome because the evaluation script evaluates based on character positions and not entity name by itself.

CHAPTER 4

EXPERIMENTAL RESULTS & PERFORMANCE ANALYSIS

4.1 Experimental Setup

The Contextualized Medication Event Dataset (CMED) was released for track 1 of the 2022 N2C2 and is a dataset that captures relevant context of medication changes documented in clinical notes [9]. CMED consist of 500 clinical notes and annotated notes with task relevant information, with 9,012 medication mentions across all the notes. The class distributions of CMED for task 1 and task 2 of the 2022 n2c2 are shown in Table I.

TABLE I

CMED TASK 1 AND TASK 2 DISTRIBUTION. ADAPTED FROM [5]

Task 1	Label	Train	Test	Total
	Drug	7229	1783	9012
Task 2	Label	Train	Test	Total
	Disposition	1412	335	1747
	NoDisposition	5260	1326	6586
	Undetermined	557	122	679

The notes are split into 400 for training and 100 for testing. The training set sets contain 7,229 medication mentions and the testing set contains 1,783 medication mentions. For task 2 there is a total of 1,747 entities labeled Disposition, 6,586 labeled NoDisposition, and 679 labeled Undetermined. The tags for task 1 after applying the BIO tagging explained in section 3.1 and their distributions are show in table II.

TABLE II

TASK 1 NER TAG COUNTS

Tags	B-Drug	I-Drug	O
Training Count	7135	1222	260808
Test Count	1764	280	68379
Total Count	8899	1502	329187

Something to point out is that the pre-processing methods are not capturing all the medications. The number of B-Drug tags should be equivalent to the Drug label in Table 4.1. For the training set there are 94 missing and in the test set there are 19 missing. There is also a large class imbalance between all 3 tags with this being due to the combination of BIO tagging and the nature of unstructured real-world data. The O tag is several orders larger than other tags as it represents all words not part of a medication mentioned. Also, the B-drug is much larger than I-Drug, with most medications being single word entities.

The tags for task 2 after applying the BIO tagging explained in Section 3.1 and their distributions are show in Tables III and IV

TABLE III

TASK 2 NER TAG COUNTS

Tags	B-Disposition	I-Disposition	B-NoDisposition	I-NoDisposition
Train Count	1318	154	5260	1010
Test Count	316	24	1326	255
Total Count	1634	178	6586	1265

TABLE IV

TASK 2 NER TAG DISTRIBUTION CONT

Tags	B-Undetermined	I-Undetermined	O
Train Count	557	58	260808
Test Count	122	1	68379
Total Count	679	59	329187

Like task 1, all entities for task 2 are not captured in pre-processing, with entities missing from Disposition. Disposition has a total of 113 entities missing with 96 from the training set and 19 missing from the test set. Also, like task 1, a class imbalance exist with O being several times larger than other classes and B-NoDisposition being much larger than other classes also. Although strong results are achieved with the current pre-processing methods, it is of interest for future work to recover the missing medication mentioned and applying methods for balancing classes.

Fine-tuning is done over 10 epochs on CMED, with a batch size of 100 in sequences and a distributed strategy over four Nvidia V100 Tensor Core GPUs is used. Each GPU creates a copy of the model, and the input batch is split equally amongst the copies. The primary machine learning pipeline is implemented with PyTorch version 1.2.0 and another pipeline has been implemented using TensorFlow 1 and all models along with their wordpiece tokenizers used are from their Hugging- Face implementations accessible through the transformers library. Transformers is a library dedicated to supporting Transformer-based architectures and facilitating the distribution of pre-trained models, it is an ongoing effort maintained by the team of engineers and researchers at Hugging Face with support from a community of over 400 external contributors [29].

4.2 Evaluation Metrics

For training the loss metric used in cross-entropy loss and is defined as a non-symmetric measure of the difference between two probability distributions [30] and is shown in Equation 4.1.

$$CrossEntropyLoss = -y\ true * log(y\ predict) \quad (4.1)$$

For testing the metrics used are precision, recall, and F-score, with only F1-scores being recorded in this section. Precision is defined as the probability that an object is relevant given that it is returned by the system [31] or as the number of true positives (correct labels) for a particular label out of all entities returned by a system and is shown in Equation 4.2.

$$Precision = TruePositive/(TruePositive + FalsePositive) \quad (4.2)$$

Recall is defined as the probability that a relevant object is returned by a system [31] or the number of true positives(correct labels) for a particular label out of all the entities that actually have that been returned by a system and is shown in Equation 4.3.

$$Recall = TruePositive/(TruePositive + FalseNegative) \quad (4.3)$$

An F1-Score is a measure of a models accuracy on a dataset and is defined as the harmonic mean between precision and recall and is shown in Equation 4.4, in this work both micro and macro F1-Scores are used. The micro score is defined as weighted harmonic mean of the precision and recall for all tags and is the one shown in Equation 4.4.

$$MicroF1Score = 2(Precicion * Recall)/(Precision + Recall) \quad (4.4)$$

The macro score is defined as the average micro F1-Score for each tag and is shown in Equation 4.5.

$$MacroF1Score = sum(F1Score)/NumTags \quad (4.5)$$

Evaluation is done on both NER tagged predictions and reconstructed annotated predictions. For the annotated predictions, each metric is evaluated using lenient and strict scores, meaning that if a medication is partially detected then it counts in lenient scores where with strict the entire medication will need to be detected.

4.3 Task 1 Results

Results shown for task 1 are the training results, NER test results, and annotated predictions test results. All results are based on PyTorch implementation, with TensorFlow implementation results included in the annotation results. The results are then analyzed in relation to the 2022 N2C2 results statistics and compared with the competition's top performing teams. Training loss results for all applied PyTorch models can be seen in Fig. 4.1. The training loss shown is the average cross entropy per-step, with a total of 2150 steps. All models show convergence at a similar rate, with Bert showing the slowest, which is an expected result due to Bert being pre-trained on only general domain corpora. Both BioBert and Bio+Clinical Bert Discharge showing convergence around 0.008, where Bert converges around 0.012 and Bio+Clinical Bert All Notes showing convergence around 0.015.

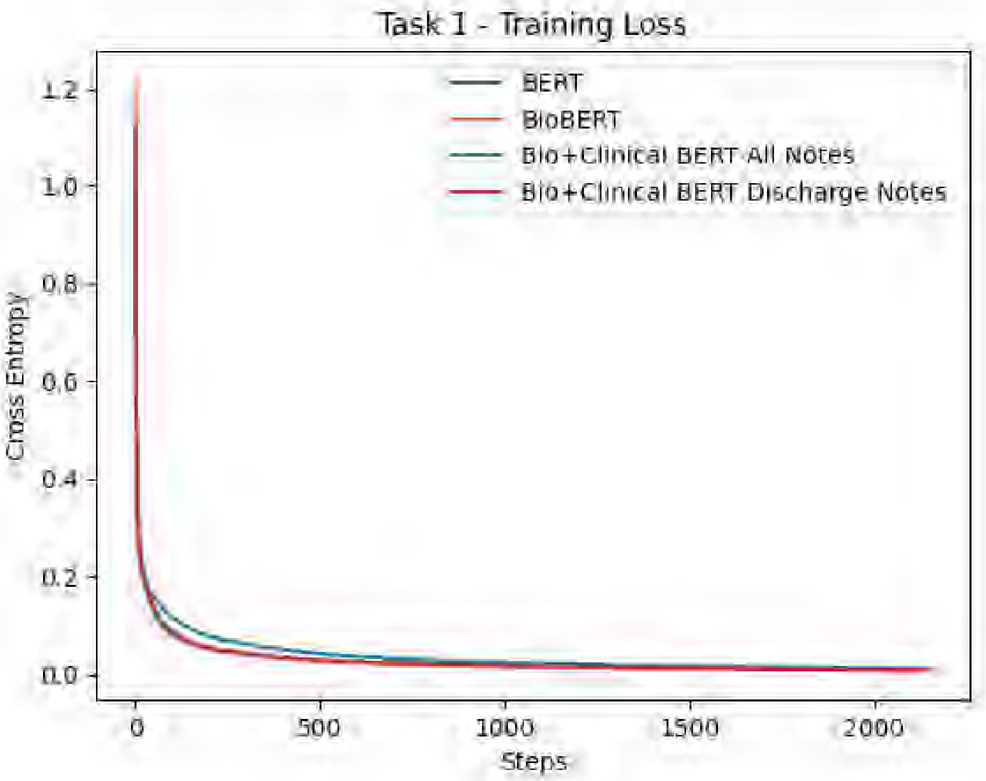


Fig. 4.1. Task 1 Training Loss.

NER results for task 1 are shown in Table V for each word tag. Overall, Bio+Clinical Bert Discharge Notes show the best performance with an average F1-score across all tags of 0.895, with BioBert following closely behind with an average score of 0.89. Bert shows the worst performance with an average score of 0.86. All models show good performance for both tags with the models making the best predictions for B-Drug entities with an average of 0.957, where I-Drug has an average of 0.8.

TABLE V

TASK 1 NER F1 SCORES (PYTORCH IMPLEMENTATIONS ONLY)

Model	**B-Drug**	**I-Drug**
Bert Base	0.94	0.78
BioBert	0.96	0.82
Bio+Clincial Bert All Notes	0.96	0.78
Bio+Clinical Bert Discharge Notes	**0.97**	**0.82**

The task 1 results with annotated medication predictions are shown in Table VI. The TensorFlow implementation of Bio+Clinical Bert Discharge Notes shows the best lenient score of 0.969 and the PyTorch implementation of Bio+Clinical Bert All Notes shows the best strict score of 0.937. The worst results are shown by Bert Base with a lenient score of 0.946 and a strict score of 0.910. On average models have a lenient score of 0.963 and strict score of 0.926.

TABLE VI

TASK 1 ANNOTATED PREDICTIONS F1 SCORES

Model	**Lenient F1**	**Strict F1**
Bert Base(PyTorch)	0.946	0.910
BioBert(TF1)	0.963	0.924
BioBert(PyTorch)	0.961	0.925
Bio+Clinical Bert All Notes (TF1)	0.966	0.926
Bio+Clinical Bert All Notes (PyTorch)	0.968	**0.937**
Bio+Clinical Bert Discharge Notes (TF1)	**0.969**	0.931
Bio+Clinical Bert Discharge Notes (PyTorch)	0.968	0.932

The aggregate performance stats for task 1 can be seen in Table VII, with the TensorFlow implementations included in the distributions. For this task there were 28 participating teams with a total of 76 unique systems [5]. The average lenient and strict scores for our systems are higher than the competition average with every system outperforming the average lenient score and 3 outperforming the average strict score.

TABLE VII

2022 N2C2 TRACK 1 TASK 1 AGGREGATE STATS. ADAPTED FROM [5].

Metric	**Max**	**Min**	**Mean**	**StdDev**
Strict F1	0.9716	0.0913	0.9238	0.1437
Lenient F1	0.9846	0.0945	0.9586	0.1429

The results for the top performing teams can be seen in Fig. 4.2. The top performing team is the Toyota Technological Institute Nagoya with a lenient score of 0.9846 and a strict score of 0.9716. The Bio+Clinical Bert Discharge (TensorFlow) outperforms 3 of the top 10 teams in lenient scores and Bio+Clinical Bert All Notes (PyTorch) outperforms 2 of the top 10 teams in strict scores.

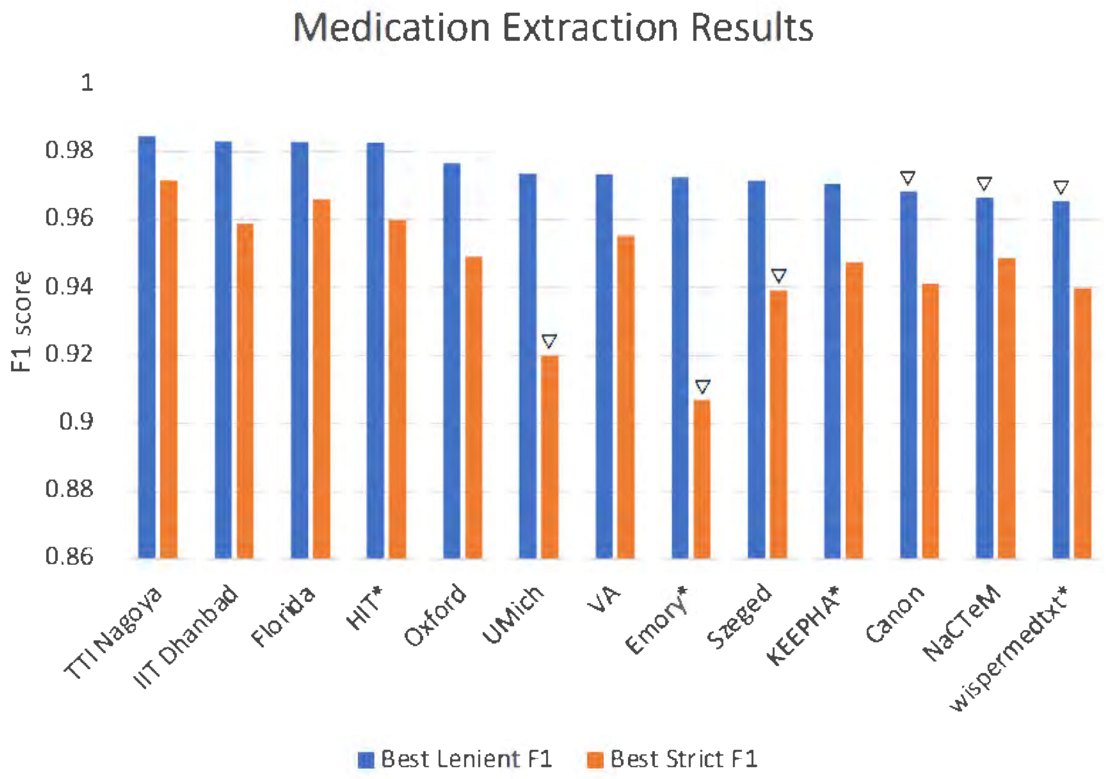


Fig. 4.2. Top Teams Results for Task 1. Adapted from [5].

4.4 Task 2 Results

Results shown for task 2 are the training results, NER test results, and annotated predictions test results. All results are based on PyTorch implementation, with no TensorFlow implementation done for this task. The lenient (micro & macro) results are then analyzed in relation to the 2022 N2C2 results statistics and com- pared with the competitions top performing teams.

Training loss results for all applied PyTorch models can be seen in Fig. 4.3. The training loss shown is the average cross entropy per step, with a total of 2150 steps. All models show convergence at a similar rate, with Bio+Clinical Bert Discharge Notes showing the slowest convergence, which is an unexpected result as it pre-trained already on clinical data on top of the training corpus that all other models are pre-trained on. Both BioBert and Bio+Clinical Bert All Notes show convergence around 0.02. Where Bert converges around 0.027, Bio+Clinical Bert Discharge Notes showing convergence around 0.035. All convergence values are also much higher than those of task 1 although both being trained with the same NER strategy, this is expected due to task 2 having 7 tags as opposed to task 1 having 3 tags.

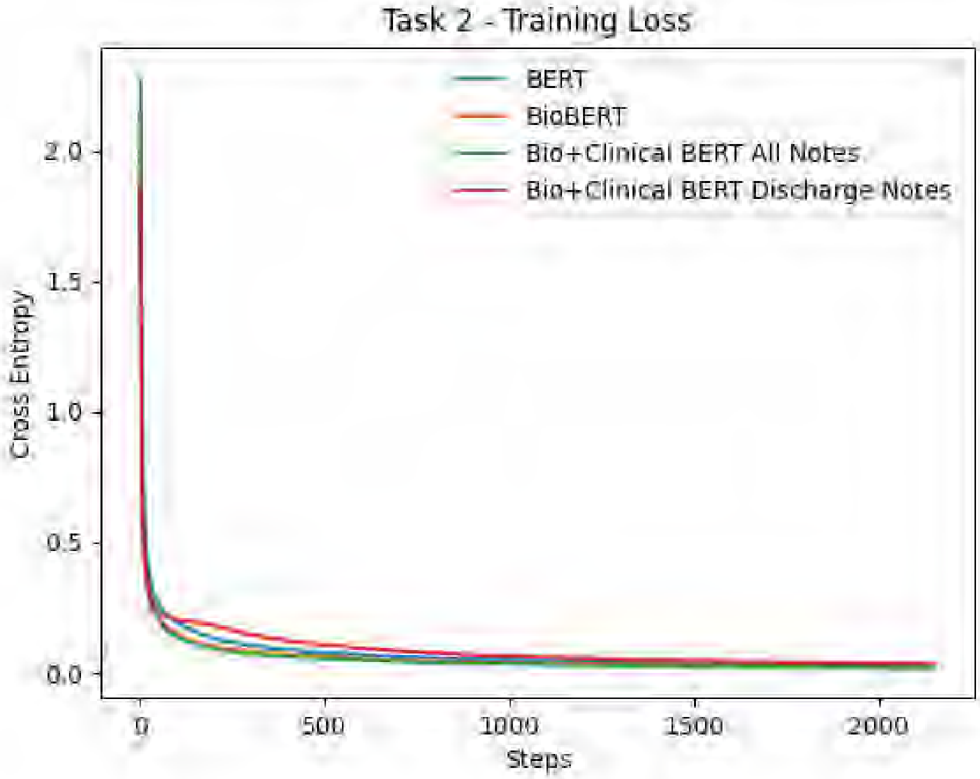


Fig. 4.3. Task 2 Training Loss.

NER results for task 2 are shown in Tables VIII and IX for each word tag. Overall Bio+Clinical Bert All Notes show the best performance with an average score of 0.62, with Bio+Clinical Bert Discharge Notes following closely behind with an average score of 0.615. BioBert shows the worst performance with an average score of 0.59 with Bert close behind with an average score of 0.60. Although BioBert on average is outperformed by Bert, this is largely due to the I-Unknown tag, with this tag only containing one example and Bert being the only applied model making the correct prediction on that tag. For each tag on average models have scores of 0.75 for B-Disp, 0.60 for I-Disp, 0.91 for B-NoDisp, 0.81 for I-NoDisp, 0.51 for B-Undetermined, and 0.07 for I-Undetermined.

TABLE VIII

TASK 2 NER F1 SCORES (PYTORCH IMPLEMENTATIONS ONLY)

Model	**B-Disp**	**I-Disp**	**B-NoDisp**	**I-NoDisp**
Bert Base	0.71	0.55	0.89	0.76
BioBert	0.73	0.60	0.90	0.84
Bio+Clinical Bert All Notes	**0.78**	**0.67**	**0.93**	0.80
Bio+Clinical Bert Discharge Notes	0.78	0.59	0.92	**0.85**

TABLE IX

TASK 2 NER F1 SCORES CONT. (PYTORCH IMPLIMENTATIONS ONLY)

Model	**B-Undetermined**	**I-Undetermined**
Bert Base	0.45	**0.29**
BioBert	0.49	0.00
Bio+Clinical Bert All Notes	**0.55**	0.00
Bio+Clinical Bert Discharge Notes	0.55	0.00

The task 2 results with annotated medication predictions are shown in Tables X and XI. Bio+Clinical Bert All Notes show the best results for micro lenient scores of 0.875 and Bio+Clinical Bert Discharge for micro strict with a score of 0.825. The worst results are shown by Bert with a micro lenient score of 0.828 and a micro strict score of 0.792. The same trend is shown for macro scores with the Bio+Clinical Bert's showing the best performance and Bert having the worst performance. For micro scores models show an average lenient score of 0.856 and average strict score of 0.825. For macro scores models show an average lenient score of 0.721 and average strict score of 0.704.

TABLE X

ANNOTATED MEDICATION TASK 2 MICRO F1 SCORES

Model	**Micro Lenient F1**	**Micro Strict F1**
Bert	0.828	0.792
BioBert	0.846	0.819
Bio+Clinical Bert All Notes	**0.875**	0.840
Bio+Clinical Bert Discharge Notes	0.873	**0.848**

TABLE XI

ANNOTATED MEDICATION TASK 2 MACRO F1 SCORES

Model	**Macro Lenient F1**	**Macro Strict F1**
Bert	0.681	0.661
BioBert	0.705	0.689
Bio+Clinical Bert All Notes	**0.752**	0.731
Bio+Clinical Bert Discharge Notes	0.749	**0.734**

The aggregate performance stats for task 2 can be seen in Table XII the models in this work are not in the distribution. For this task there were 19 participating teams with a total of 52 unique systems [5]. The average for all systems in this work outperformed the average macro and micro lenient scores in the competition, with all systems outperforming the average for micro and the same for the macro scores except for Bert.

TABLE XII

2022 N2C2 TRACK 1 TASK 2 AGGREGATE STATS. ADAPTED FROM [5].

Metric	**Max**	**Min**	**Mean**	**StdDev**
Lenient micro F1	0.9225	0.2170	0.8232	0.1249
Lenient macro F1	0.8348	0.2666	0.6928	0.1347

The results for the top performing teams can be seen in Fig. 4.4. The top performing team is the Toyota Technological Institute

Nagoya with a micro lenient score of 0.9225. Bio+Clinical Bert All Notes outperforms 3 of the top 10 teams in micro lenient scores and Bio+Clinical Bert Discharge Notes outperforms 1 of the top 10 teams in macro lenient scores.

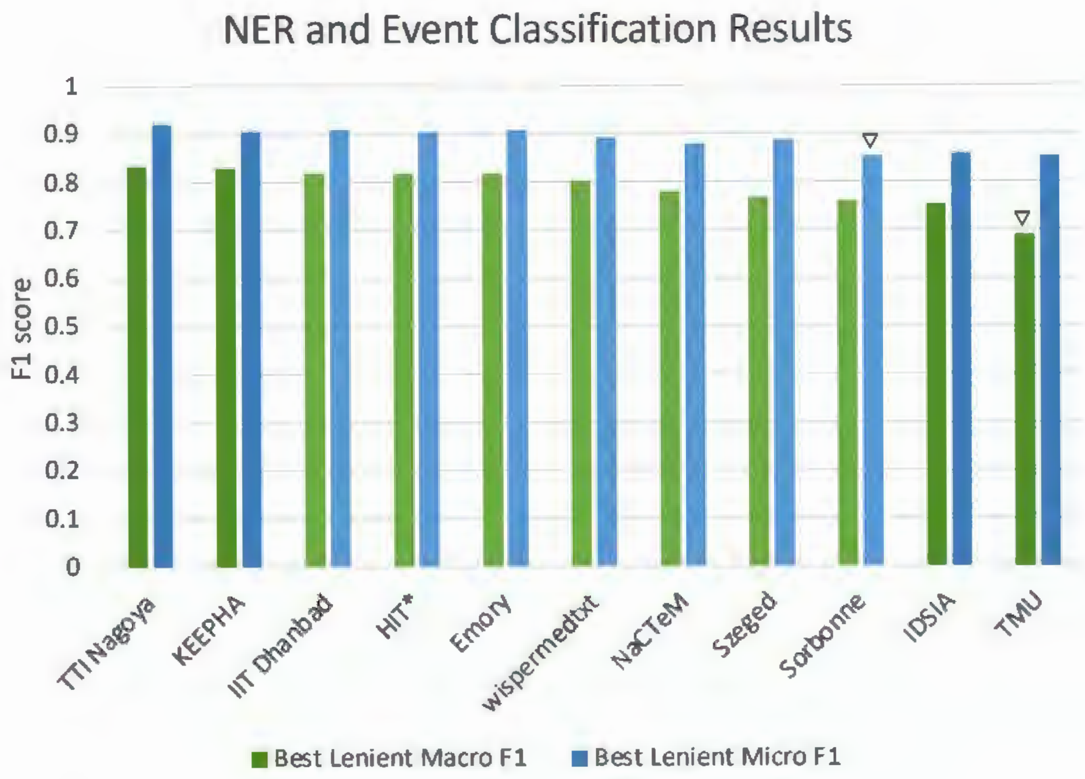


Fig. 4.4. Top Teams Results for Task 2. Adapted from [5].

4.5 Discussion

The models in this show performance comparable to teams in the top 10 ranking for task 1 and task 2 of the 2022 N2C2 competition. For task 1 all models showed comparable results for lenient and strict scores but slightly differed between the same model with different implementations. This is thought to be due to differences in processing methods used to accommodate the respective framework requirements. The TensorFlow implementation is applied with the use of scripts from Google research, where in the PyTorch implementation all processing scripts is created by researchers, with the Bert variation imported as a model layer from the transformers library.

The NER results for task 1 in Table 4.5 show that the PyTorch implementation did not perform as well on I-Drug tag, with all scores .8+-.2. This looks to be a primary cause for lower strict scores as strict scores require the character position of an entire medication. The confusion matrix for Bio+Clinical Bert Discharge Notes for task 1 is shown in 4.5, out of the 280 entities with the ground-truth I-Drug 65 or 23% are mislabeled, bringing down the recall score, with this looking to be a strong factor reducing the F1-score.

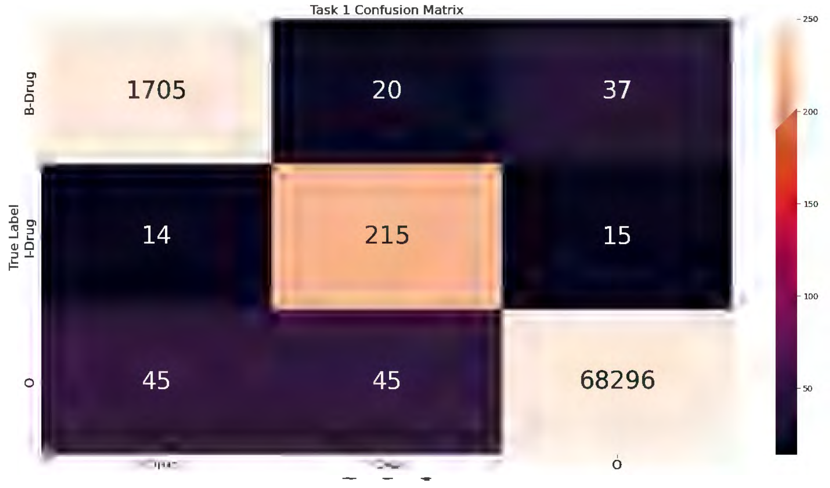
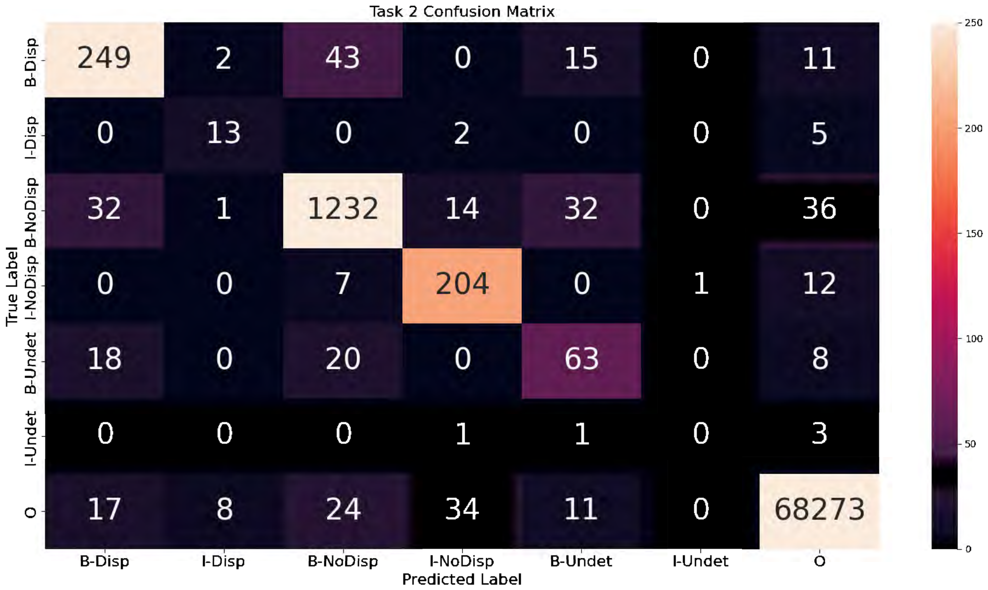


Fig. 4.5. Task 1 Confusion Matrix.

The NER results for task 2 shown in Table 4.5 show moderate performance for all tags, with the worst performance on I-Disposition and B-Undetermined, I-Undetermined not included because only 1 entity with label in test set. For Bio+Clinical Bert All Notes, I-Disposition has a score of .67 which looks to be a result of sample size for entities with the tag, as the test set has a total of 24 with 13 labeled correctly as shown in Fig. 4.6. As for B-Undetermined as shown in Fig. 4.6, 32 of out 122 entities with the ground truth B-Undetermined are given B-NoDisposition. Some thoughts on why this score is lower is that the nature of B-Undetermined entities is ambiguous meaning that they lie somewhere between between Disposition and NoDisposition.



True Label \ Predicted Label	B-Disp	I-Disp	B-NoDisp	I-NoDisp	B-Undet	I-Undet	O
B-Disp	249	2	43	0	15	0	11
I-Disp	0	13	0	2	0	0	5
B-NoDisp	32	1	1232	14	32	0	36
I-NoDisp	0	0	7	204	0	1	12
B-Undet	18	0	20	0	63	0	8
I-Undet	0	0	0	1	1	0	3
O	17	8	24	34	11	0	68273

Fig. 4.6. Task 2 Confusion Matrix.

An issue observed from the evaluation data is that the models do not detect well on short sequences. Often medications are missed for sequences of word length 3 or less. This is also thought to be an issue when using a general tokenizer like the Punkt tokenizer as previously mentioned in section 3.1. The Punkt tokenizer does not capture all the different language syntax structures in an EHR when tokenizing and often splitting sections up based on appearance of sentence ending punctuation. This can be a problem due to periods showing up in different contexts such as lists not necessarily indicating the end of a sentence, making many input sequences shorter than they may need to be. Tools specific to breaking down clinical text such as Medspacy [32] are available and of interest for how they affect system performance in comparison to the Punkt tools. There can also be medications names such as “insulin” where in one context, can be a medication and in another context would not be a medication, models have shown to have trouble with these kinds or words. Punctuation removal also shows some risk for strict scores where some medications include punctuation such as “/” or “-”.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Researchers had been tasked with identifying medication mentions and events regarding changes in medication for Track 1 of the 2022 National Clinical NLP Challenges. The Contextualized Medication Event Dataset (CMED) has been pro- vided to accomplish these tasks. In this research work, the Variations of the Bert model, namely, Bert base, BioBert, Bio+Clinical Bert with MIMIC III Discharge notes, and Bio+Clinical Bert with all MIMIC III notes were fine-tuned on CMED and applied for identifying medication mentions in clinical notes and identifying if an event is associated with the medication. Both tasks were formulated as an NER task with a BIO tag scheme for entities of interest. Results for task 1 showed that the best performance achieved were from the TensorFlow implementation of Bio+Clinical Bert with Discharge MIMIC III notes with a lenient F1-score of 0.969 and the PyTorch implementation of Bio+Clinical Bert with all MIMIC III notes with a strict F1-score of 0.937. As far as the overall competition, these models outperform 3 of the top 10 teams for lenient scores and 2 out of the top 10 for strict scores. On average the proposed models had a lenient score of 0.963 and 0.926 for this task. Results for task 2 showed the best results from

Bio+Clinical Bert with All MIMIC III notes with a micro lenient F1-score of 0.875 and Bio+Clinical Bert with MIMIC III Discharge Notes with a micro strict F1-score of 0.848. For the overall competition this model outperforms 3 out of the top 10 teams for micro lenient scores. On average the proposed models have a micro lenint score of 0.856 and a micro strict score of 0.825 for task 2.

5.2 Future Work

Although models achieve results comparable to those in the top 10 ranking of the competition, further work is of interest to make the proposed approach perform better. Class imbalance has been noted as an issue for both tasks, with this reflected in the results achieved for labels. Methods for dealing with the class imbalance are of interest. In the 2018 N2C2 class imbalance was tackled using different sampling methods, cost-sensitive learning, ensemble learning, and one-class classification [33]. Applying a clinical tokenizer such as Medspacy is of interest as a replacement for the Punkt tokenizer. It has been mentioned that a general tokenizer such as Punkt's does not capture all of the syntax structures in EHRs. Medspacy combines statistical and symbolic methods and is created by a team of practitioners at the Veterans Health Administration and University of Utah and is a Spacy-based [34] based library containing components targeting medical text [32]. Another variation of the Bert model, MedBert is of interest. MedBert is trained on similar sources of all other applied variations of Bert with the addition of previously released N2C2 clinical data [35]. Other further work of interest is reconstruction of entire medications with their punctuation to increase strict scores as punctuation is removed during pre-processing, developing methods for better medication detection in short sequences to increase recall, utilizing filtering methods such as pos-tagging to filter out mis-tagged words to increase precision, and developing pipeline for task 3 of the 2022 N2C2 competition.

REFERENCES

[1] G. Kim, C. Lee, J. Jo, and H. Lim, "Automatic extraction of named entities of cyber threats using a deep bi-lstm-crf network," International journal of machine learning and cybernetics, vol. 11, pp. 2341–2355, 2020.

[2] A. Vaswani, N. Shazeer, and N. Parmar et al., "Attention is all you need," in 31st Conference on Neural Information Processing, 2017.

[3] J.Devlin, M.Chang, K.Lee, , and K.Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," Clinical Orthopaedics and Related Research, vol. 1810.04805, 2018.

[4] J. Lee, W. Yoon, and S. Kim et al., "Biobert: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, 2020.

[5] D. Mahajan, J. Liang, C. Tsou, and O. Uzuner, "n2c2 2022 challenge: Track 1 contextualized medication event extraction," Presented as the 2023 N2C2 Workshop, Washington DC", 2023.

[6] V. Ehrenstein, H. Kharrazi, and H. Lehmann et al., Tools and Technologies for Registry Interoperability, Registries for Evaluating Patient Outcomes: A User's Guide, 3rd Edition. Rockville: Agency for Healthcare Research and Quality, 2019.

[7] S. Blumenthal, "The use of clinical registries in the United States: A landscape survey," Journal of Electronic Health Data and Methods, vol. 5, 2017.

[8] D. Mahajan, J. Liang, and C. Tsou, "Toward understanding clinical context of medication change events in clinical narratives," pp. 1–8, 2022.

[9] D. Mahajan, J. Liang, and C. Tsou, "Toward understanding clinical context of medication change events in clinical narratives," in Proceedings of Machine Learning Research, pp. 1–8, 2021.

[10] A. Stubbs, M. Filannino, and E.Soysal et al., "Cohort selection for clinical trials: n2c2 2018 shared task track 1," Journal of the American Medical Infor- matics Association, vol. 26, pp. 1163–1171, 09 2019.

[11] H. Dai, C. Su, and C. Wu, "Adverse drug event and medication extraction in electronic health records via a cascading architecture with different labeling models and word

embeddings," Journal of the American Medical Informatics Association, vol. 27, no. 1, pp. 47–55, 2020.

[12] V. Kumar, A. Stubbs, S. Shaw, and O. Uzuner, "Creation of a new longitdinal corpus of clinical narratives," Journal of Biomedical Informatics, vol. 58, no. 5, pp. S6–S10, 2015.

[13] J. Lee, D. Scott, and M. Villarroel et al., "Open-access mimic-ii database for intensive care research," in 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 8315–8318, 2011.

[14] A. Johnson and T. Shen et al., "Mimic-iii, a freely accessible critical care database," in Scientific Data 3, 2016.

[15] W. Styler and S. Bethard et al., "Temporal Annotation in the Clinical Domain," Transactions of the Association for Computational Linguistics, vol. 2, pp. 143–154, 04 2014.

[16] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

[17] S. Henry, K. Buchan, M. Filannino, A. Stubbs, and O. Uzuner, "2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records," Journal of the American Medical Informatics Association, vol. 27, no. 1, pp. 3–12, 2020.

[18] A.Mansouri, L.Affendey, and A.Mamt, “Named entity recognition ap- proaches,” International Journal of Computer Science and Network Security, vol. 8, pp. 339–344, 2008.

[19] D. Jurafsky and J. Martin, Speech and Language Processing. 2020.

[20] C. Libbi, J. Trienes, D. Trieschnigg, and C. Seifert, “Generating synthetic training data for supervised de-identification of electronic health records,” Fu- ture Internet, vol. 13, no. 136, 2021.

[21] Y.Kim, C.Denton, L.Hoang, and A.Rush, “Structured attention networks,” in 5th International Conference on Learning Representations, 2017.

[22] A.Radford, K.Narasimhan, T. Salimans, and I.Sutskever, “Improving language understanding by generative pre-training,” 2018.

[23] W. L. Taylor, ““cloze procedure”: A new tool for measuring readability,” Journalism & Mass Communication Quarterly, vol. 30, pp. 415 – 433, 1953.

[24] Y.Wu, M.Schuster, and Z.Chen et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” Clinical Orthopaedics and Related Research, vol. abs/1609.08144, 2016.

[25] T.Kiss and J.Strunk, “Unsupervised Multilingual Sentence Boundary Detec- tion,” Computational Linguistics, vol. 32, pp.

485–525, 12 2006.

[26] Y. Zhu, R. Kiros, and R. Zemel et al., "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in The IEEE International Conference on Computer Vision (ICCV), December 2015.

[27] W. Foundation, "Wikimedia downloads."

[28] E. Alsentzer, J. Murphy, and W. Boag et al., "Publicly available clinical BERT embeddings," in Proceedings of the 2nd Clinical Natural Language Processing Workshop, (Minneapolis, Minnesota, USA), pp. 72–78, Association for Com- putational Linguistics, June 2019.

[29] T. Wolf, L. Debut, and V.Sanh et al., "Huggingface's transformers: State-of- the-art natural language processing," arXiv preprint arXiv:1910.03771, 2019.

[30] X. Chen, S. Kar, and D. Ralescu, "Cross-entropy measure of uncertain vari- ables," Information Sciences, vol. 201, pp. 53–60, 2012.

[31] C. Goutte and E. Gaussier, "A probailistic interpretation of precision, recall and f-score, with implication for evaluation," in Advances in Information Re- trieval, pp. 345–359, 2005.

[32] H.Eyre et al., "Launching into clinical space with medspacy: a new clinical text processing toolkit in python," in AMIA Annual

Symposium Proceedings, vol. 2021, p. 438, American Medical Informatics Association, 2021.

[33] S.Santiso, A.Casillas, and A.Perez, “The class imbalance problem detecting ad- verse drug reactions in electronic health records,” Health Informatics Journal, vol. 25, pp. 1768 – 1778, 2019.

[34] Y. Vasiliev, Natural language processing with Python and spaCy: A practical introduction. No Starch Press, 2020.

[35] C. Vasantharajan, K. Tun, and H. ThiNga et al., “Medbert: A pre-trained language model for biomedical named entity recognition,” in 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Confer- ence (APSIPA ASC), pp. 1482–1488, 2022.

CURRICULUM VITAE

Tariq Abdul-Quddoos

Department of Electrical and Computer
Engineering Roy G. Perry College of
Engineering
Prairie View A&M University,
Texas Email:
tariqaq98@gmail.com
LinkedIn:https://www.linkedi
n.com/in/ tariq-abdul-
quddoos-510566155/

Education

M.S. Electrical Engineering, Prairie View A&M University, 2023.

B.S. Electrical Engineering, St.Cloud State University, 2021

Employment

Prairie View A&M University - CREDIT Center, Research Assistant, 2021– 2023.

Los Alamos National Laboratory, Los Alamos, New Mexico, Research Intern, Summer 2022.

Emerson Automation Solutions, Chanhassen, Minnesota, Electrical Engineering Co-Op, Jan. 2021–Aug. 2021.

Publications

T. Abdul-Quddoos, C.Zemelka, P.Lee "Characterizing the Dynamic Response of a Foam-Based Testbed with Material, Geometric, & Experimental Uncer- tainties", Los Alamos National Laboratory ,In International Modal Analysis Conference(IMAC) 2023.

Award & Recognition

IBM Masters Fellowship, 2022