TECHNICAL UNIVERSITY DRESDEN

FACULTY OF COMPUTER SCIENCE
MASTER COMPUTATIONAL LOGIC

Master's Project

for obtaining the academic degree
Master of Science

Automatic Classification of Diagnoses from Patient Records

Garvit Joshi
(Born 10. January 1997 in Indore, India, Mat.-No.: 4825153)

Tutor: Dipl. Inf. Walter Forkel

Dresden, June 4, 2020

Task Description

The given task was to extract medical diagnoses (ICD concepts) from written patient records. Dataset used is (a subset of) the MIMIC-III patient database. The following tasks are performed:

1. The suitability of using a pre-trained BERT model as a Natural Language Inference system, to predict the ICD concepts from patient records in the MIMIC-III database is evaluated.
2. A domain-specific BERT model (BIO BERT) on (a subset of) the MIMIC-III data is trained.
3. Another attention-based model for this task is created and evaluated.
4. Comparison of the performance of these models on the MIMIC-III dataset with the MetaMap tagger is performed

Contents

1 Dataset

The dataset used in the project is created from **MIMIC-III** dataset. **MIMIC-III** (Medical Information Mart for Intensive Care III) is a large, freely-available database comprising deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.[9]

1.1 Data Structure and Modifications done

Following two tables have been used:

Table 1.1: from NOTEEVENTS the following columns are used

SUBJECT_ID	HADM_ID	CATEGORY	DESCRIPTION	TEXT
INT	INT	VARCHAR(50)	VARCHAR(300)	TEXT

1. SUBJECT_ID, ID of the patient. This ID can be used multiple times as a patient can have multiple cases.
2. HADM_ID, ID for patient cases/hospital stay. This ID is unique.
3. CATEGORY, type of note.
4. DESCRIPTION, type of note
5. TEXT, contains the notes.

Table 1.2: from the DIAGNOSES_ICD the following columns are used

SUBJECT_ID	HADM_ID	ICD9_CODE
INT	INT	VARCHAR(10)

1. SUBJECT_ID, ID of the patient.
2. HADM_ID, ID for patient cases/hospital stay.
3. ICD9_CODE, ICD9 codes corresponding to the diagnoses

A joint table is made from both the tables (NOTEEVENTS, DIAGNOSES_ICD) by sorting them on HADM_ID and then performing a full join on the HADM_ID. As it is the unique identifier and then dropping all the rows which had **NAN** values in them. The Sequence_Num allows to get highest priority ICD9_Codes and then the column is dropped in the final table after dropping the rows with lower priority.

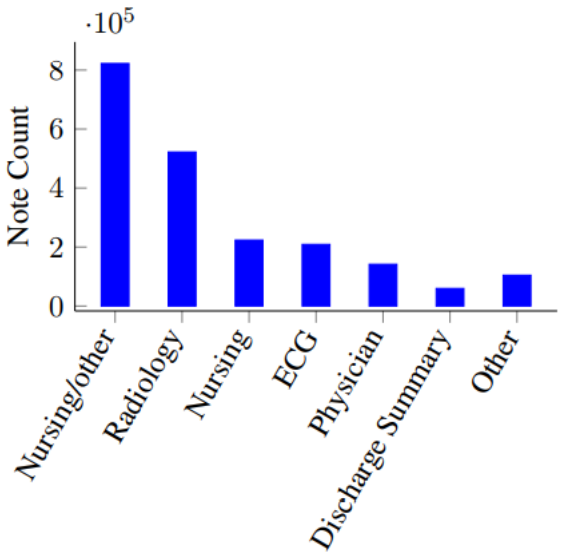

Figure 1.1: Frequency Distribution of Various Categories

1.2 Problems

The table has uneven values for the HADM_ID, and SUBJECT_ID. They are of type INT in NOTEEVENTS and type INT(with a trailing decimal 0) in DIAGNOSES_ICD. To further complicate the things specific IDs are also alphanumeric. These alphanumerical IDs are not so crucial as a frequency analysis shows that they are less frequent.

1.3 Viability for Project Usage

Since there are 13,000 ICD9 Codes, a further frequency filter is performed on the table, thus results obtained contain only five most frequent occurring codes. This step is done to ensure that the multi-class classification performs predictions better, as not all the 13,000 codes often occur and to train a module to classify 13,000 code is complicated with codes having a sparse distribution. An experiment was performed with all 13,000 ICD9_Codes as classes. The result generated from this experiment was

unsatisfactory with an accuracy of .01 %. The X is text, and the y is ICD9_code There are a total of 559816 rows in the table and five ICD9_Codes **4019,3893,966,9672,9604** are in decreasing order of frequency and also labelled as class 0 to 4 respectively.
The following diseases are related to the ICD9_code:

1. 401.9 - Hypertension NOS 92% of Total Data
2. 38.93 - Venous catheterization, not elsewhere classified 2.4% of Total Data
3. 96.6 - Enteral infusion of concentrated nutritional substances 2.1 % of Total Data
4. 96.72 - Continuous invasive mechanical ventilation for 96 consecutive hrs 1.4 % of Total Data
5. 96.04 - Insertion of endotracheal tube 1.2 % of Total Data

1.4 Measures Used for Analysis

Precision vs Recall and F1 scores are used for the following reasons: The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall. This measure is generally the harmonic mean, which, for the case of two numbers, coincides with the square of the geometric mean divided by the arithmetic mean. [8] There are several reasons that the F-score is not optimal in particular circumstances due to its bias as an evaluation metric. These are also known as measures because recall and precision are evenly weighted.
The F_1 Score can be calculated by: $F_1 = \left(\frac{2}{recall^{-1}+precision^{-1}}\right) = 2.\frac{precision.recall}{precision+recall}$

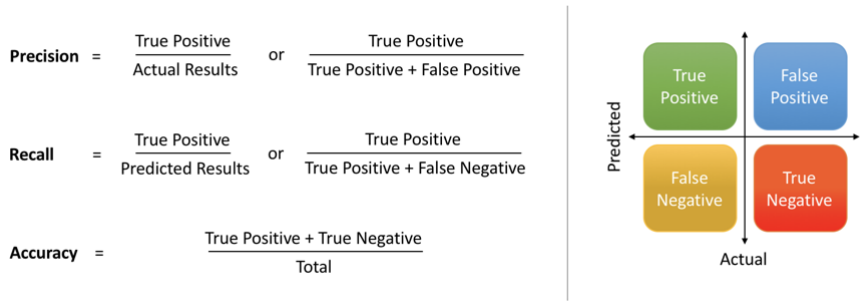

Figure 1.2: Pictorial Representation of the Measures

2 Neural Networks (NNs)

Neural networks are machine learning algorithms that simulate the human brain. A concept of supervised learning is used here where labelled data is used for training. The neural network predicts the labels for the input. The input is X, and the labels are Y. They are split into test and train. The train uses both the input and labels and learns to predict the labels correctly. During the training phase, the labels predicted are scored by a loss function that tells how off the prediction was from the actual value, and then it tries to minimise this value. The concept of transfer learning is used in the project. BERT is used in the project as a pre-trained NN, and we fine-tune it for medical data and then do a classification task on it. The classification is a multi-class classification with four classes being the ICD9 codes that are most frequently used.The BERT is a transformer NN.

2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is developed by Google. It applies bidirectional training of a Transformer to language modelling. The critical difference in this model from the predecessor is that it uses left-to-right and right-to-left training, compared to only either left-to-right or right-to-left training in Word2Vec or GloVe. This model has a more profound sense of language context and flows than simpler unidirectional models. It uses Masked LM (MLM), which is unique to this model.

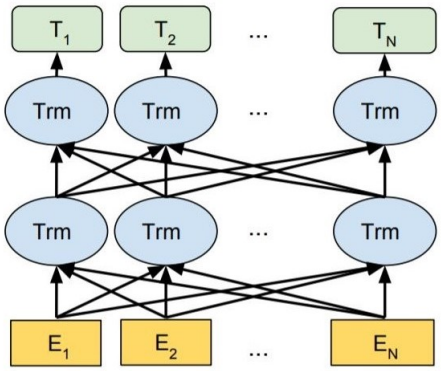

Figure 2.1: Bert Structure[3]

2.1.1 How Does it Work?

It follows two approaches:

1. Feature-based approach
2. Masked LM(MLM)

2.1.1.1 Featured-Based Approach

In a feature-based approach, a pre-trained neural network produces word embeddings which are used in NLP models. BERT uses transformer that learns contextual relation between words or sub-words in text. A transformer has 2 parts encode and decoder, but BERT focuses on the encoder. It encodes the entire sequence of words at once. Thus, it is considered as bidirectional. The context of a word is based on left and right words. The high-level description is that it converts the input text to a sequence of tokens that uses tokenization functions to create tokens. The tokenized input is then embedded into a vector and then processed by the neural network. The output of this is a sequence of vectors sized x where every vector corresponds to an input token with the same index.

2.1.1.2 Masked LM(MLM)

In this approach, 15% of words in a sequence are taken and replaced with tokens and then the model attempts to predict the original value of the masked words, based on the context by non-masked words in sequence. [3]

1. Classification layer is added on top of encoder output.
2. The output vector is multiplied by the embedding matrix resulting in the transformation to vocabulary dimension.
3. Probability of each word is predicted using SoftMax.

Example:

Input: The $[MASK]_1$ is admitted in ICU. He is unable to $[MASK]_2$.

Labels: $[MASK]_1$ = patient; $[MASK]_2$=breath.

During training, the model takes a pair of sentences as input and try to predict if the second sentence in the pair is the subsequent sentence in the original document. Where 50% of the inputs are paired with the second sentence, Which is the subsequent sentence in the original document. The remaining 50% of inputs with the second sentence, which is a random sentence from the corpus.

It is assumed that the random sequence is disconnected or irreverent to the first sentence. [3][2] For a distinction between two sentences, we process the input before being passed to the model as input:

1. $[CLS]$ is inserted at the beginning and $[SEP]$ at the end of every sentence.
2. Sentence embedding is added to each token indicating sentence A or B. It is similar to vector embedding.

A positional embedding is added to every token to indicate its position in the sequence. The following steps are followed to predict whether the second sentence is connected to the first sentence:

1. The entire input sequence is passed through the Transformer.
2. $[CLS]$ Token output is transformed into a vector of shape 2×1, using a classification layer that utilizes learned matrices of weights and biases.
3. Probability of IsNextSequence with SoftMax is calculated.

Training is done with the above two methods to minimize the combined loss.

Example:

Sentence A : The patient is admitted.

Sentence B: He is unable to breathe.

Label: IsNextSentence

Sentence A: The patient is admitted.

Sentence B: He is discharged.

Label: NotNextSentence

2.1.2 Changes made to the Model

A classification layer has been added on top of the Transformer output for the $[CLS]$ token. This layer predicts 4 classes (ICD9 Codes) as, the codes used in a sentence as the Y column for the prediction task.

2.1.3 Hyper-Parameters for the Model

Table 2.1: Hyper-Parameter BERT

Batch Size	Learning Rate	Training Epochs	Warm-up Proportion
16	$2e-5$	2	0.1

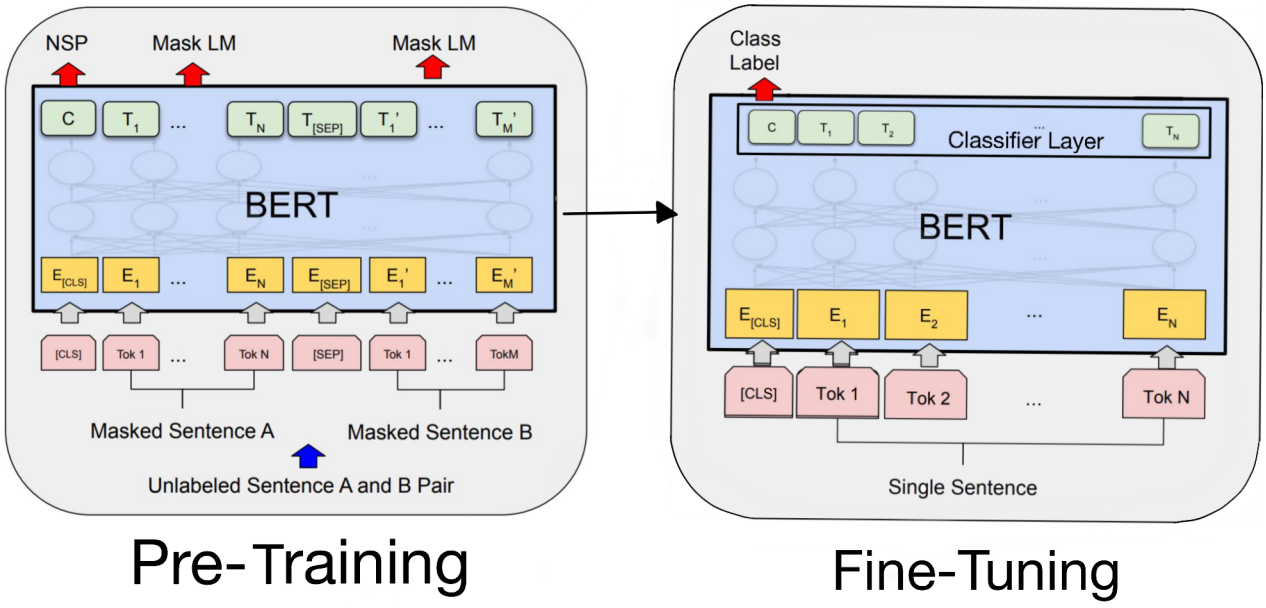

Figure 2.2: BERT Structure Pre-Training and Fine-Tuning for The Task[3]

2.1.4 Results using MIMIC-III

The Size of the following elements are changed by taking a Radomized Sample from The Dataset:

1. Test data.
2. Train data.
3. Total rows used from the dataset, as a direct result form changing of values of 1 and 2.
4. Tokenization Sequence Maximum length

Table 2.2: BERT Results

Sr NO	Precision	Recall	F_1 Score	Support	Accuracy	Overall Size	Sequence Length
1	0.93	1	0.96	103662	0.927	111964	90
2	0.92	1	0.96	916	0.915	1000	100
3	0.93	1	0.96	2788	0.928	3000	150
4	0.94	0.99	0.97	2788	0.928	3000	250
5	0.93	1	0.96	2788	0.927	3000	90

Sr.No refers to the Serial Number in the table.

The Support in the table indicates the total number of Test cases used, i.e. the Test Dataset size.

2.1.4.1 Plots of Performance Measures

As it can be seen from the data, the accuracy increases with the increase in sequence length and the overall f_1 score also increase with the sequence length. The more complex the token sequence is, the better conclusion is drawn from the data. The size of data does not have much impact on the accuracy besides that beyond certain limits the data size becomes too small to train the model correctly, and the accuracy falls. The f_1 score and accuracy are the most important scores, and they tell us how accurately can the values be predicted and what is the ratio of false positives and false negatives and thus also tells how reliable the module is when predicting the result. It is observed that regardless of sequence length, the accuracy will fall if not enough examples are there for training. The module performs best at 250 length. Going above this to max length offered by BERT is not optimal. The size of the vectors becomes too large, and computation becomes slow. Since the aggregate probability is used, this does not affect overall accuracy in a long text. Smaller sequence length is also okay, but below the threshold of 100, the module learns too less. For compensation of loss of information, the number of samples are to be increased, which may or may not be possible in a real-world situation.

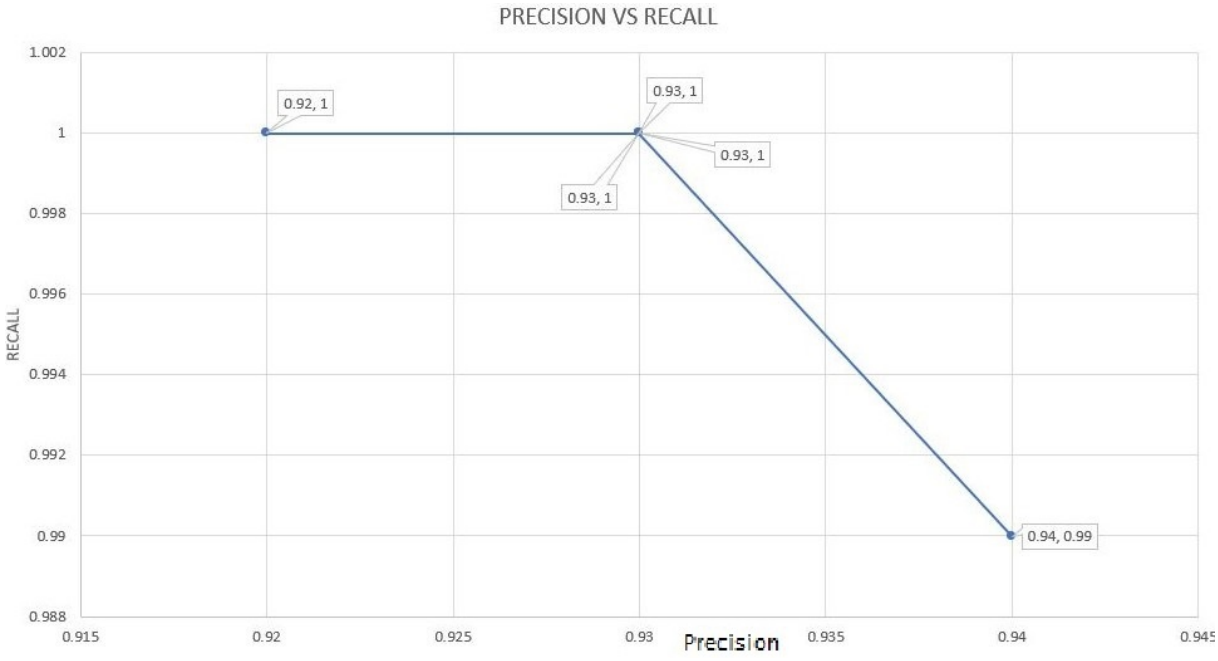

Figure 2.3: BERT Precision vs Recall

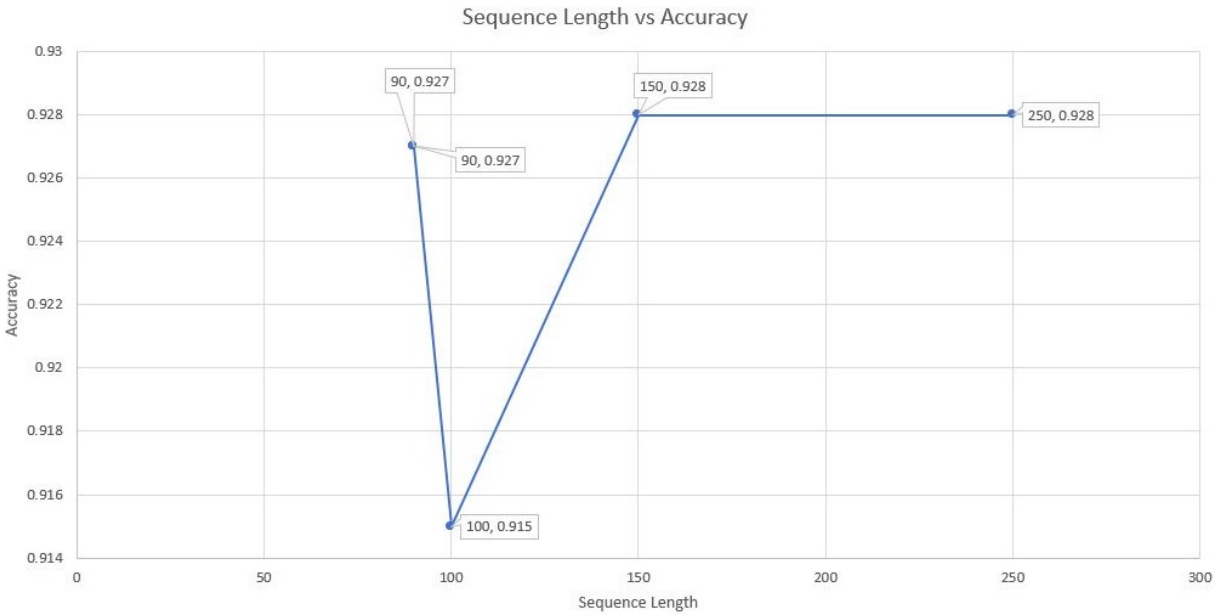

Figure 2.4: BERT Sequence Length vs Accuracy

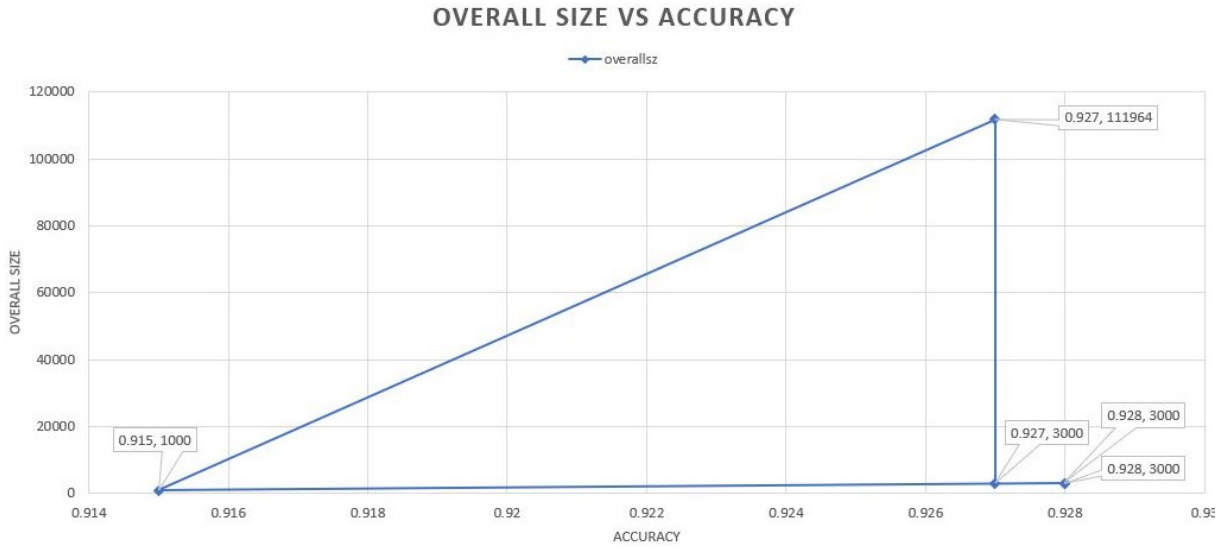

Figure 2.5: BERT Overall Size vs Accuracy

2.2 Bio-BERT

BioBERT is a pre-trained biomedical language representation model for biomedical text mining. It was created in Korea University, Seoul.

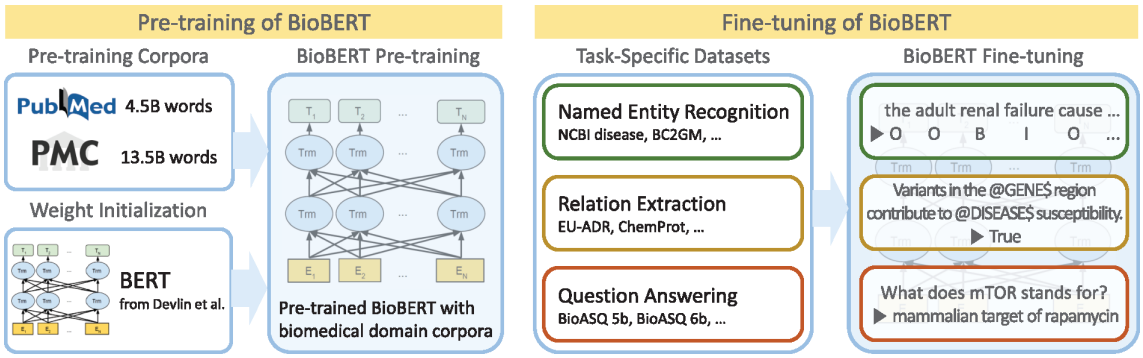

Figure 2.6: BIO-BERT Structure

2.2.1 How it is Different from BERT?

Bio-BERT is a pre-trained language representation model for the biomedical domain. It is initialised with weights from BERT. Bio-BERT is then pre-trained on biomedical domain corpora (PubMed abstracts and PMC full-text articles). There is also a fundamental difference in how BIO-BERT processes Tokens. BERT has a maximum sequence length of 512. The BIO-BERT while training solves this problem by splitting the text into multiple parts and predicting them separately. The final probability is the aggregation of probabilities of all the sub-parts. The BIO-BERT paper mentions that due to concern about relying only on maxima or mean, authors have combined them to get an accurate result. Aggregate probability helps, but the max size is bound to the memory available in the system and the probability obtained is still artificial.[4][5][6]

The Function used is: $P\left(redmit = 1|h_patient\right) = \frac{P^n_{max}+P^n_{mean}n/c}{1+n/c}$

Where P^n is the probability of nth sentence fragment.

2.2.2 Clinical\Bio-BERT

It is a fine-tuned module of BIO-BERT. The clinical narratives, like physician notes, are different in linguistic characteristics from a general and non-clinical biomedical test. Thus, CLINICAL BIO-BERT has been trained on the NOTEVENTS in discharge summary from MIMIC-III dataset.

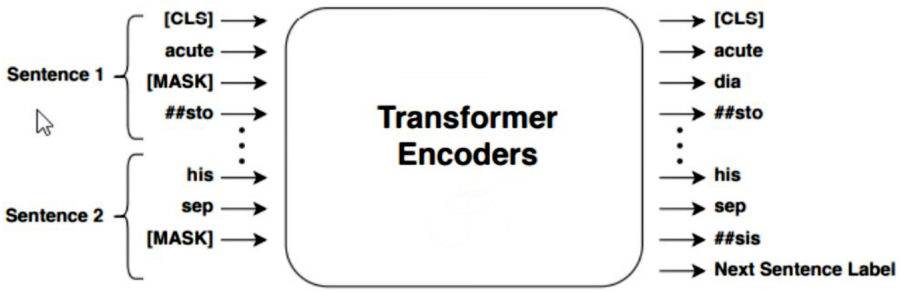

Figure 2.7: BIO-BERT Tokenization Example

2.2.3 Hyper-Parameters for The Model

Table 2.3: Hyper-Parameters Clinical\Bio-BERT

Batch Size	Learning Rate	Training Epochs	Warm-up Proportion
16	$2e-5$	2	0.1

2.2.4 Results using MIMIC-III

The Size of the following elements are changed by taking a Randomized Sample from The Dataset:

1. Test data.
2. Train data.
3. Total rows used from the dataset, as a direct result form changing of values of 1 and 2.
4. Tokenization Sequence Maximum length

Table 2.4: Clinical\Bio-BERT Results

Sr NO	Precision	Recall	F_1 Score	Support	Accuracy	Overall Size	Sequence Length
1	0.93	1	0.96	2788	0.929	3000	90
2	0.93	1	0.96	103662	0.926	111964	100
3	0.92	1	0.96	916	0.915	1000	150
4	0.95	0.99	0.977	2788	0.932	3000	250
5	0.96	0.99	0.98	2788	0.941	3000	90

Sr.No refers to the Serial Number in the table.

The Support in the table indicates the total number of Test cases used, i.e. the Test Dataset size.

2.2.4.1 Plots of Performance Measures

As it can be seen from the data from BIO-BERT, the trend is similar. Accuracy increases with the increase in sequence length, and the overall f1 score also increases with the sequence length. The more complex the token sequence is, the better conclusion is drawn from the data. The size of data does not have much impact on the accuracy besides that beyond certain limits the data size becomes too small to train the model correctly, and the accuracy falls. The f_1 score and accuracy are the most important scores. They tell us how accurately the values can be predicted and what is the ratio of false positives and false negatives. Hence, informing how reliable the module is when predicting the result. It is observed that regardless of sequence length, the accuracy will fall if not enough examples are there for training. The module performs best at 250 length. Going above this to max length offered by BERT is not optimal. The size of the vectors becomes too large, and computation becomes slow and since the aggregate probability is taken, which does not affect overall accuracy in a long text. Smaller sequence length is also okay, but below the threshold of 100, the module learns less. For compensation of loss of information, the number of samples are to be increased, which may or may not be possible in a real-world situation. This model is better than BERT as the model performs better on the 250 token length, and since this one is also doing the aggregate probability, the loss of information is minimal. The increase in performance is because of the increased understanding of the clinical text.

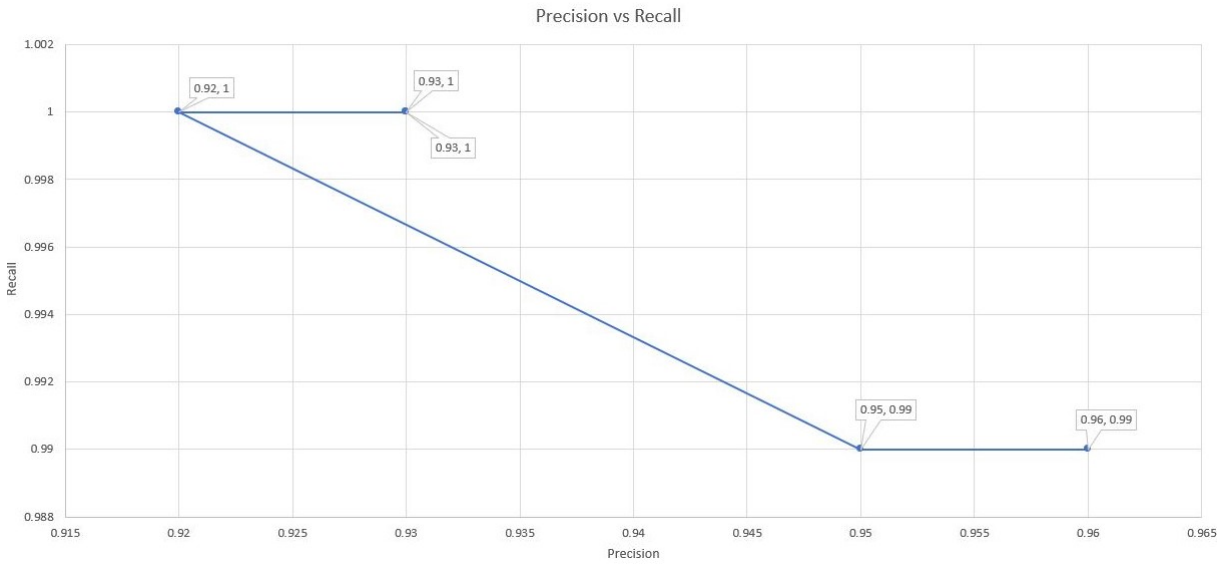

Figure 2.8: BIO-BERT Precision vs Recall

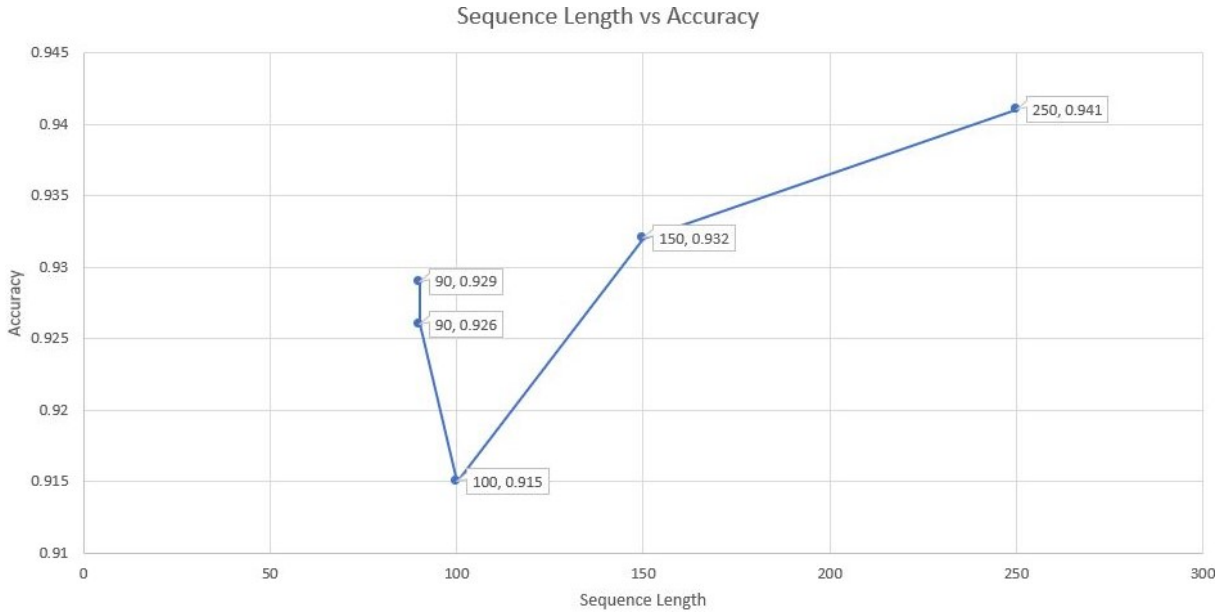

Figure 2.9: BIO-BERT Sequence Length vs Accuracy

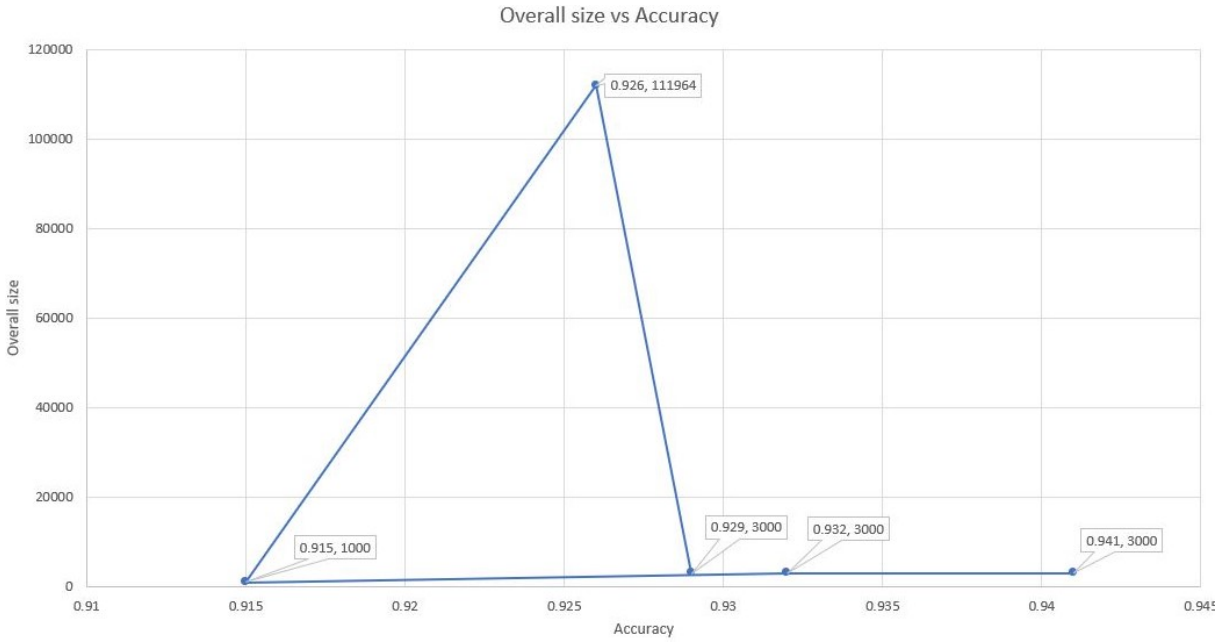

Figure 2.10: BIO-BERT Accuracy vs Overall Size

2.3 Tokenization and Embeddings

2.3.1 Tokenization

BERT is trained on WordPiece tokenization. Thus, it can breakdown word into more than one sub-words. It helps without of vocabulary words. Complicated words are also better represented. Padding is also performed, which adds blank spaces to word sequences smaller than maximum sequence length. The

example below shows how to generate token sequences and how the embeddings look like for BERT.

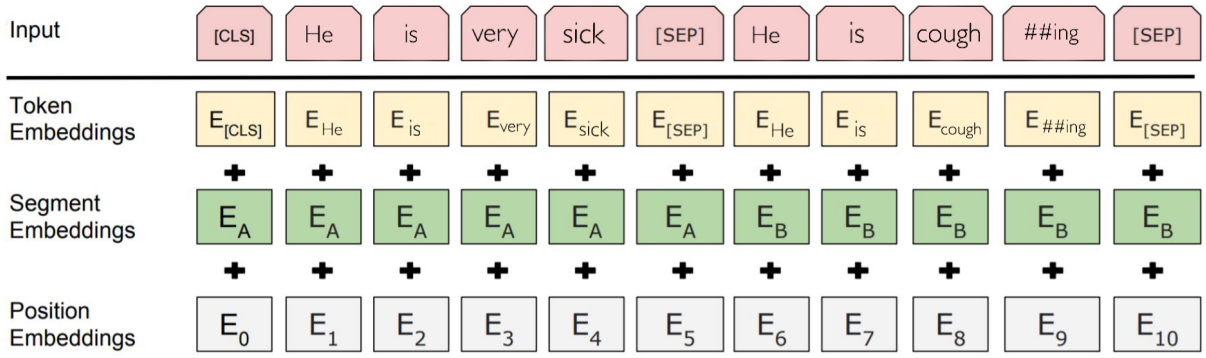

Figure 2.11: Tokenization Example

2.3.2 Word Embedding Generation

Word and sentence embedding vectors are extracted from the given BERT and CLINICAL/BIO-BERT in a bid to compare the two models. These embeddings are useful for a keyword or search expansions, semantic search and information retrieval. In the model, these are used as feature inputs to the NLP module. Unlike word2vec that can annotate a similar to two complete dissimilar but same spelt words.

Example

The duck is yellow.

The guy shouted duck.

[1][2] The embeddings generated in the modules are context-informed. The segment IDs mentioned in the explanation above are o for A and 1 for B, and a description of the text is obtained as an object in encoded layers in 4 dimensions. The dimensions for a sequence length of 250 are:

1. The layer number (24 layers)
2. The batch number (1 sentence)
3. The word/token number (250 tokens in the sentence)
4. The hidden unit/feature number (1024 features)

That is 61,44,000 unique values to represent one sentence. This is one of the reasons why the computation is so slow and memory expensive. The tokenization process above uses a max sequence that determines how long the tokens are and, various sequence lengths are tried for the task. The tokenization sometimes adds empty padding to the text if the length is not as long as the token sequence. This is dependent on the sentence length. Thus, 250 sequence length is optimal for MIMIC-III sentences.

2.3.2.1 How We Create The Word and Sentence Vectors?

The hidden states mentioned above are taken. And the dimensions for out tasks, for a sample sentence from the corpus, the max 250 token lengths is 24x250x1024. A single sentence vector and individual vector for each token are required. For this, it is required to combine the layer vectors. According to the BERT authors concentration of last 4 layers is the best. For **word vectors**, when the last four layers are concatenated, the result is 4x1024=4,096 and shape is 250 x 4,096 and summing of last four layers give the shape 250x 1024 for **sentence vector**, to get a single vector, the second to last hidden layer is averaged to get a single vector of 1024. A check for the contextual dependence of the vectors is performed, and a Cosine Similarity Measure for words picked from the corpus is preformed. This show how well does the system understand the contextual dependence of "in" and "out" of vocabulary words. The creation sentence vector helps in reducing the memory usage of the model.

2.4 Custom Neural Network

A custom neural network based on the principle similar to the transformer was constructed, but it failed to perform satisfactorily with an accuracy of 0.01%. This accuracy is not sufficient to even predict a simple 22 token long vector. This guided the project into using transfer learning rather than creating a solution from scratch.

3 MetaMap Tagger

MetaMap provides access from biomedical text to the concepts in the unified medical language system (UMLS) Meta-thesaurus. It provides a link between a text of biomedical literature and knowledge which includes synonym relationships that are embedded in the Meta-thesaurus.

3.1 How Does it Work ?

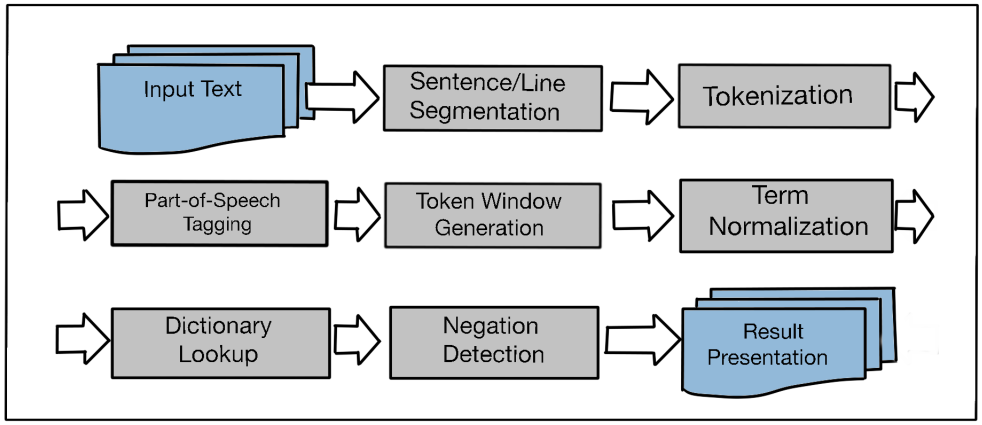

Figure 3.1: Flow Chart MetaMap Tagger

The simplified working is as follows:

lexical/syntactic analysis on Input text is performed as follows:

1. tokenization, sentence boundary determination and acronym/abbreviation are identified
2. part-of-speech is tagged
3. a lexical lookup of input words in the SPECIALIST lexicon
4. syntactic analysis that consists of a shallow parse, which phrases and their lexical heads are identified by the SPECIALIST minimal commitment parser

The phrases found undergo further analysis as follows:

1. variant generation, all variation of phrases are generated.
2. candidate identification, intermediate results consisting of Meta-thesaurus strings that match some

phrase text are computed and evaluated on their match on input.

3. mapping construction, candidates identified in the previous step are combined and evaluated to produce a final result that , optimally best, matches the phrase text.

4. word sense disambiguation, the mappings which have semantic consistency with surrounding text are found and favoured.[4]

Both candidates and the final mappings are evaluated in a linear combination of four measures:

1. centrality

2. variation

3. coverage

4. cohesiveness.

These measures are linguistically inspired. The result thus generated is normalized between 0 to 1000.

4 Comparison BERT vs Clinical\Bio-BERT vs MetaMap Tagger

4.1 Sentence Vectorisation

Table 4.1: Sentence Vector Comparison

Sr NO	NN	Seqlength	Comparison word	Similar	Different
1	BERT	512	Random	0.32	0.31
2	BIOBERT	512	Random	0.56	0.53

This clearly shows that BIO-BERT is better at distinguishing between similar and dissimilar word meanings based on the context of the word. The Sequence length is maximal length that BERT can take.

4.2 Comparison of Results BERT vs Clinical BIO-BERT (via plots)

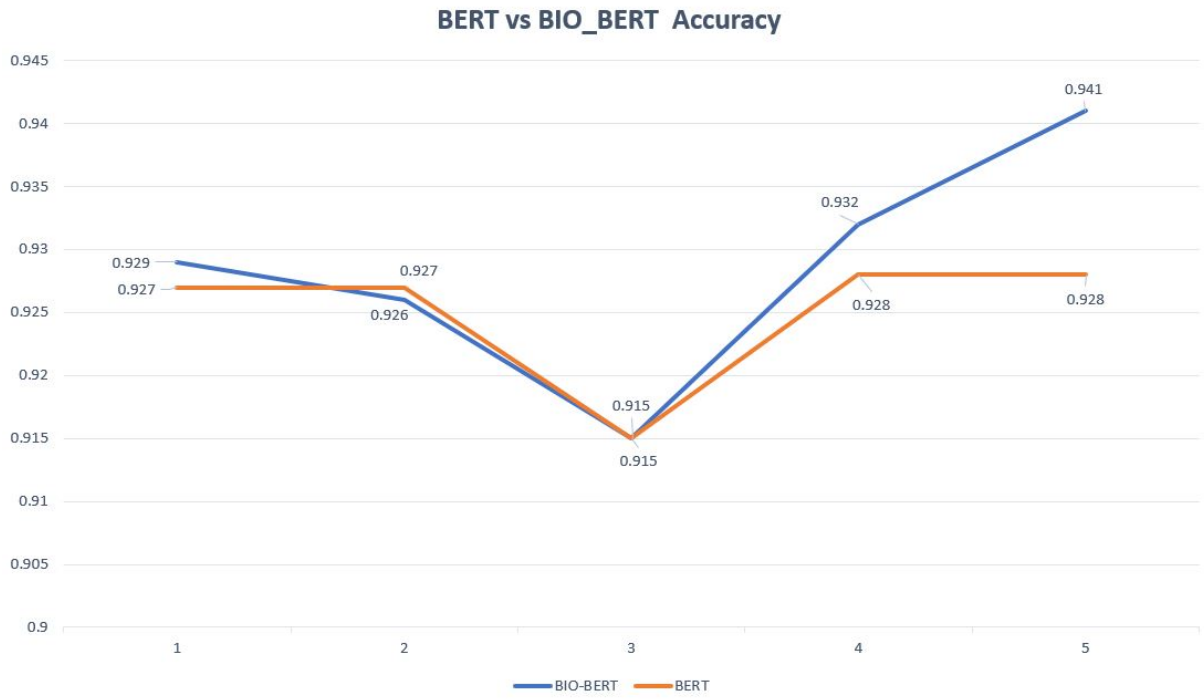

Figure 4.1: Accuracy

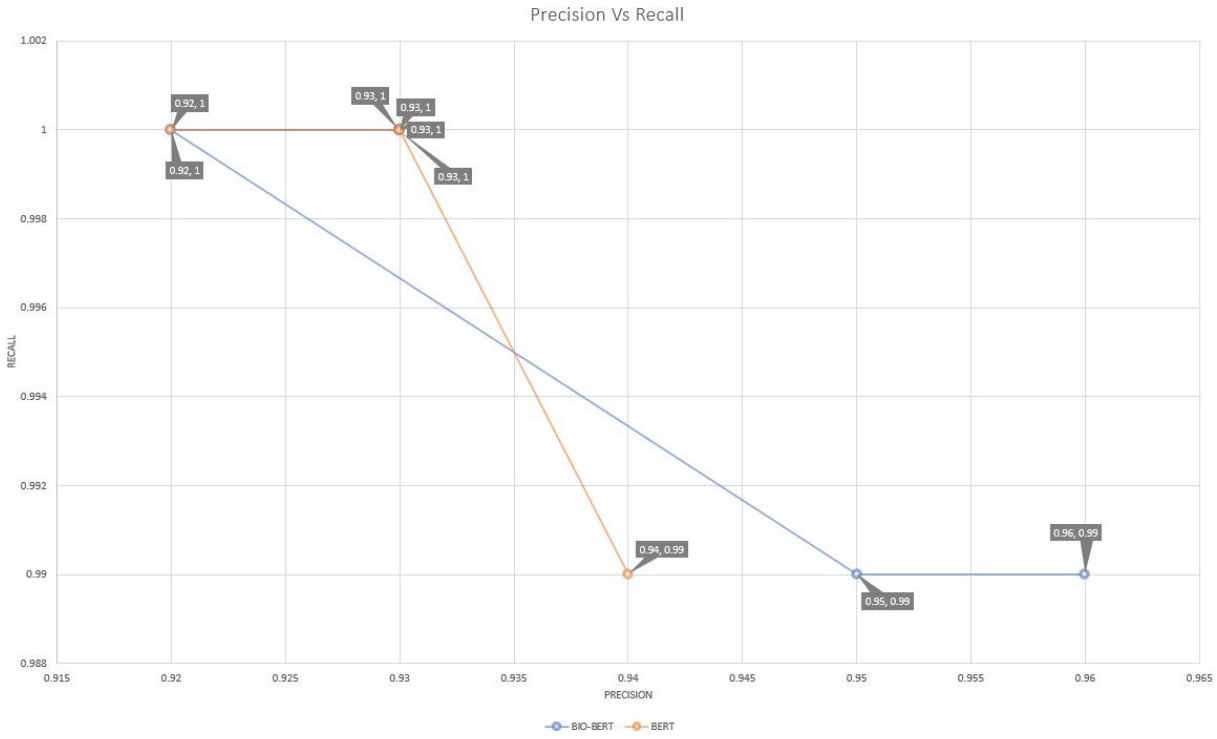

Figure 4.2: Precision vs Recall

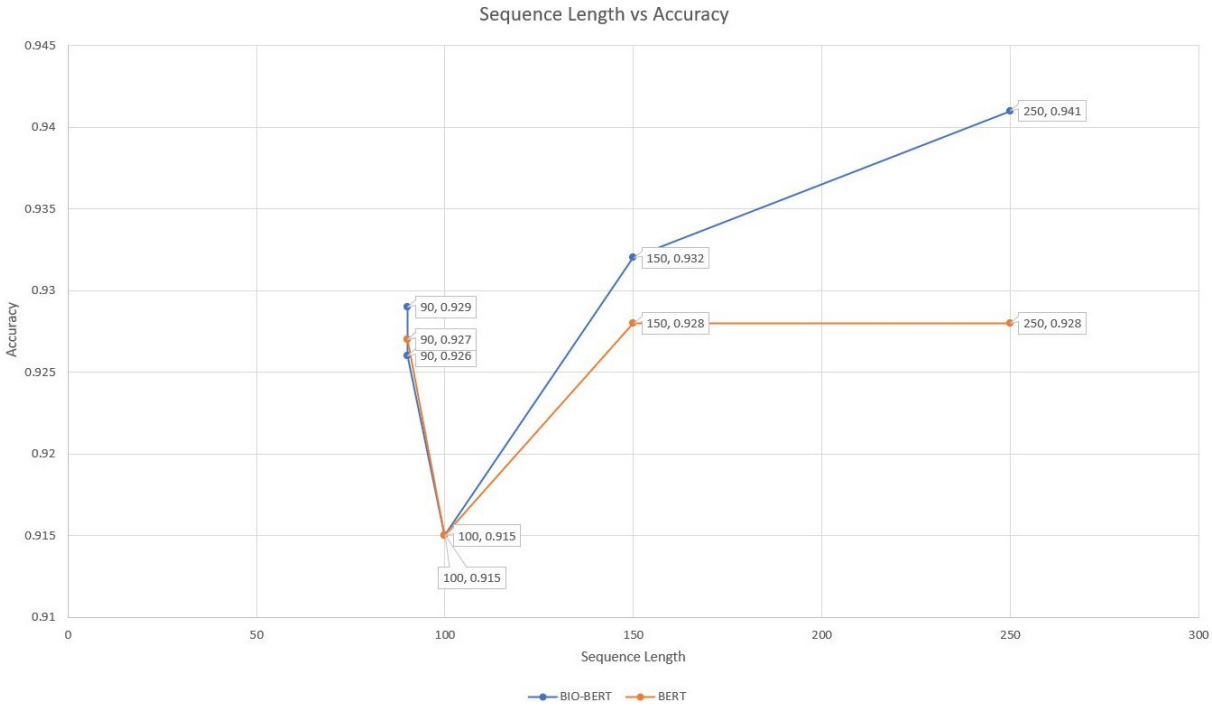

Figure 4.3: Sequence Length vs Accuracy

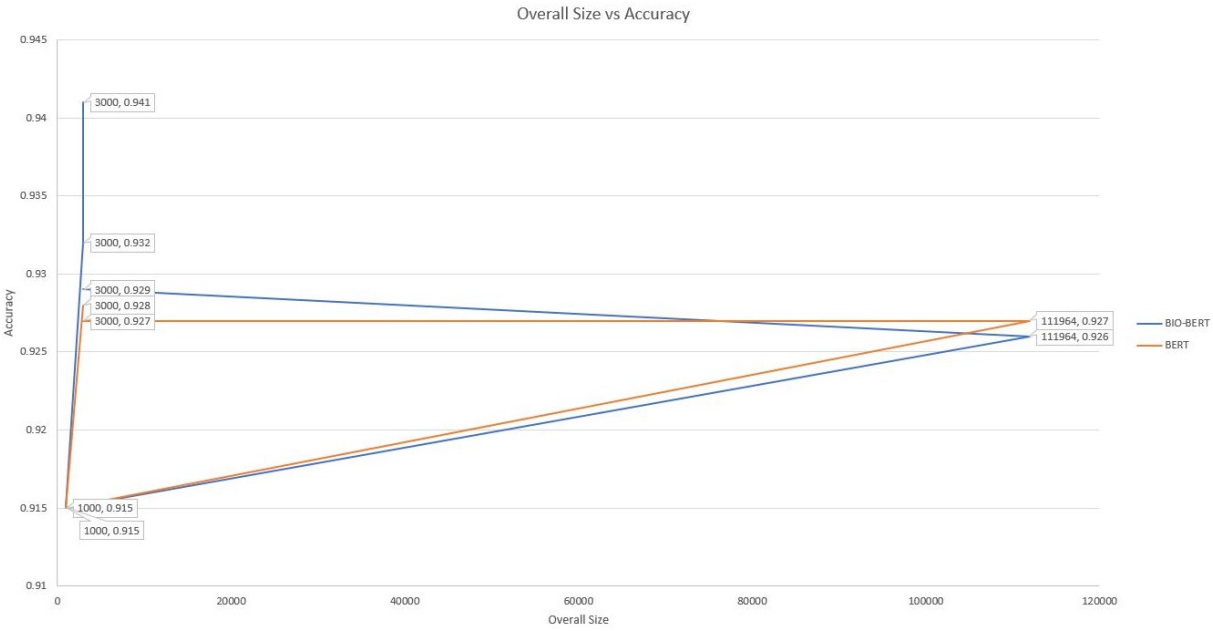

Figure 4.4: Overall Size vs Accuracy

4.3 Indirect Comparison with MetaMap tagger

Direct comparison between both the Neural Networks(BERT and BIO-BERT) with MetaMap Tagger is not possible. The direct evaluation of MetaMap tagger to manually constructed gold standards to a meta thesaurus have not been created for ICD9 Codes. Hence the only way to evaluate is indirect; we apply it to a task and compare before and after performance. Since there is no comparison of BERT to golden standard, they are compared on the basis of their performance on the I2B2 2010 task of concept extraction and entity recognition.

Dataset	**Baseline**			**FT-BERT**				**FC-BERT**		
		CNN-RNN	**JOINT**	**Cased**	**Uncased**	**Bio**	**MIMIC**	**Cased**	**Uncased**	**MIMIC**
n2c2	P	0.9673	0.9715	0.9811	0.9803	0.9798	**0.9838**	0.9632	0.9653	0.9662
	R	0.8878	**0.9079**	0.899	0.9021	0.9018	0.9015	0.8524	0.8952	0.8769
	F	0.9258	0.9386	0.9383	0.9396	0.9392	**0.9409**	0.9044	0.9289	0.9194

Figure 4.5: BERT Methods on I2B2, P= Precision, R= Recall, F= F_1 score[5]

Model	MedNLI	i2b2 2006	i2b2 2010	i2b2 2012	i2b2 2014
BERT	77.6%	93.9	83.5	75.9	92.8
BioBERT	80.8%	**94.8**	86.5	78.9	**93.0**
Clinical BERT	80.8%	91.5	86.4	78.5	92.6
Discharge Summary BERT	80.6%	91.9	86.4	78.4	92.8
Bio+Clinical BERT	**82.7%**	94.7	87.2	**78.9**	92.5
Bio+Discharge Summary BERT	**82.7%**	94.8	**87.8**	78.9	92.7

Figure 4.6: Performance Comparison on The I2B2 Task[4]

Method	**i2b2 2010**		**i2b2 2012**		**Semeval 2014 Task 7**		**Semeval 2015 Task 14**	
	General	MIMIC	General	MIMIC	General	MIMIC	General	MIMIC
word2vec	80.38	84.32	71.07	75.09	72.2	77.48	73.09	76.42
GloVe	84.08	85.07	74.95	75.27	70.22	77.73	72.13	76.68
fastText	83.46	84.19	73.24	74.83	69.87	76.47	72.67	77.85
ELMo	83.83	87.8	76.61	80.5	72.27	78.58	75.15	80.46
BERTbase	84.33	89.55	76.62	80.34	76.76	80.07	77.57	80.67
BERTlarge	85.48	**90.25**	78.14	**80.91**	78.75	**80.74**	77.97	**81.65**
BioBERT	84.76	-	77.77	-	77.91	-	79.97	-

Figure 4.7: Performance Comparison of Model fine-tuned on MIMIC-3 and General model on I2B2 2010 challenges[5]

Collection/Tool	MetaMap			cTAKES (DL)			DNorm			MetaMap Lite		
	P	R	F-1	P	R	F-1	P	R	F-1	P	R	F-1
BioScope (negation)	43.7	34.4	38.5							85.2	37.9	**52.4**
NCBI disease	60.3	68.3	64.1	47.0	53.8	47.4	74.1	67.6	70.7	73.1	71.9	**72.5**
ShARe (entities)	59.5	48.1	53.2	46.3	46.2	46.2	N/A	N/A	N/A	74.2	42.1	**53.8**
i2b2 2010 (entities)	38.1	35.7	36.8	31.9	34.1	32.9	N/A	N/A	N/A	47.0	31.9	**38.0**
i2b2 2010 (negation)	40.2	32.2	38.3							53.8	38.0	**44.6**
LHC clinical articles	58.8	77.2	66.8	42.6	59.9	49.8	71.5	58.2	64.2	69.4	74.9	**70.0**
LHC biological articles	46.8	75.6	57.8	47.1	60.6	53.0	67.7	62.8	65.2	67.5	77.9	**72.4**

Figure 4.8: Entity Recognition and Negation Detection Results[2]

4.3.1 I2B2 2010

In The 2010 i2b2/VA Workshop on Natural Language Processing Challenges for Clinical Records three tasks were presented: [10]

1. A concept extraction task focused on the extraction of medical concepts from patient reports
2. An assertion classification task focused on assigning assertion types for medical problem concepts
3. A relation classification task focused on assigning relation types that hold between medical problems, tests, and treatments.

4.3.2 Observations

The Result that can be drawn from the tables above is that Between BERT and BIOBERT and Clinical BERT the neural network that is best for our task is Clinical BERT as it has an accuracy of 94.1%. For general medical tasks involving the recognition of context BIO- BERT is better than BERT. For more general tasks, BERT is Better.

The MetaMap Tagger's performance can only be estimated to be somewhere below BERT for medical code prediction, but it's the best for entity recognition and concept identification for Tokens. However, the relationship between the ICD9 codes and the Text Context is better understood by the Neural Networks.

4.4 Comparion MetaMap vs BERT and BIO-BERT (via Examples)

TEXT 1 (taken verbatim from the database)

```
[**2124-8-1**] 3:22 AM
CHEST (PORTABLE AP)          Clip # [**Clip Number (Radiology)
13427**]
Reason: please eval for change, new infiltrate
Admitting Diagnosis: ASTHMA;COPD EXACERBATION
_____________________________________________________________
[**Hospital 2**] MEDICAL CONDITION:
87 year old woman with TBM, COPD a/w parainfluenza, admitted
REASON FOR THIS EXAMINATION:
please eval for change, new infiltrate
_____________________________________________________________
FINAL REPORT
HISTORY:  Tracheobronchomalacia with pneumonia. FINDINGS:
In comparison with the study of [**3-31**], the monitoring and
support devices remain in place.  There is again hyperexpansion
of the lungs. Cardiac silhouette is mildly enlarged and there is
some increased prominence of pulmonary vessels. This could
represent elevated pulmonary venous pressure, though some of
this could be a manifestation of the underlying pulmonary
pathology. Obscuration of both hemidiaphragms suggests
bibasilar atelectasis and pleural effusion.
```

Expected Result: 4019

Results:

BERT:4019 Unspecified essential hypertension

BIOBERT:4019 Unspecified essential hypertension

MetaMap Tagger:

SNOMED ID CUI SNOMED CT Concept Name

392521001 C0262926 History of contextual qualifier

233604007 C0032285 Pneumonia

75540009 C0205250 qualifier value elevated pulmonary venous pressure (related to hypertension)

TEXT 2 (taken verbatim from the database)

```
pmicu npn 7p-7a
pt's sedation was increasd w/the addition of ativan ~8pm. she
received several boluses of both fentanyl and ativan to
improve comfort and sedation level prior to extubation. this
was accomplished ~10pm, and the pt was extubated w/o incident
w/her family at her bedside. although she immediately
desaturated and has maintained sats in the 60's on room air,
she remains hemodynamically stable. she continues
to receive only ativan and fentanyl qtts; all other meds have
been discontinued. in addition, she essentially has not had a
uo for several hours now. currently, she appears to be
comfortable and her family remains at her side.
continue comfort measures.
```

Expected Result:966 Enteral infusion of concentrated nutritional substances

Results:

BERT:966

BIOBERT:966

MetaMap Tagger: SNOMED ID CUI SNOMED CT Concept Name

271280005 C0553891 Removal of endotracheal tube (Shows Removal)

It is observed that the MetaMap works on the simple structured report but fails on a report with an ambiguous structure. The Neural Networks perform optimally in both instances. The Result generated is opposite of the required result. The text was fed line by line for the above result.

TEXT 3 (taken verbatim from the database)

```
Respiratory Care Note
Patient remains intubated and ventilated on psv
13/ 50%/ 5 of cpap. RR somewhat high in mid-20s
with tidal volumes around 500cc. Had periods of
desaturation down to 89-90, suctioned for thick
tan sputum., Saturation more stable now at 97%.
```

Expected Result:3893 Venous catheterization, not elsewhere classified

Results:

BERT:3893

BIOBERT:3893

MetaMap Tagger: SNOMED ID CUI SNOMED CT Concept Name

12484000 C0035239 Respiratory care and adjustment procedure respiratory care note patient remains

53950000 C0035239 Respiratory therapy procedure respiratory care note patient remains

47545007 C0199451 Continuous positive airway pressure ventilation treatment regime/therapy ventilated on psv 13/ 50%/ 5 of cpap rr

The ICD9 Code shows venous catheterization and the text data does not explicitly mention this.This is captured in form of Ventilation for patient by MetaMap Tagger. The NNs predicted the expected value but there is no apparent relation to text.

TEXT 4 (taken verbatim from the database)

```
Sinus tachycardia. Baseline artifact.
Borderline left atrial abnormality.
Left ventricular hypertrophy with prominent
mid-precordial voltage. Compared to
the previous tracing
```

Expected Result: 9672 Continuous invasive mechanical ventilation for 96 consecutive hours or more

Results:

BERT:9672

BIOBERT:9672

MetaMap Tagger: SNOMED ID CUI SNOMED CT Concept Name

SNOMED ID CUI SNOMED CT Concept Name Category Phrase Number of Subconcepts 118622000 C0016169 Fistula morphologic abnormality sinus tachycardia baseline artifact borderline left atrial abnormality left ventricular hypertrophy with prominent mid precordial voltage
428794004 C0016169 Fistula disorder sinus tachycardia baseline artifact borderline left atrial abnormality left ventricular hypertrophy with prominent mid precordial voltage

The ICD9 Code shows continues ventilation this is not apparent from the text but can be easily concurred by a human.NNs seems to have concurred the same relation as humans. MetaMapp Tagger cannot detect this relationship and seems to repeat the entire text structure as some body structure related code.

Disclaimer The ICD9_Code shown in above cases are most relevant to the case. The Modified data set used to train the BERT modules uses the most significant ICD9_Code. The ICD9_Code used are representing the conclusive diagnosis; hence only one ICD9_Code per summary is used.MetaMapp Tagger tends to show more than one outputs because there is no priority order used.

5 Conclusion

The expected result is that Clinical BERT is going to perform better than BIO-BERT, which in turn is going to perform better than BERT. The Neural Network, which was built from scratch, was expected to fail. The reason for this expectation was that the BERT is trained on the general corpus and does not contain medical vocabulary and since BERT has a context-based approach even after fine-tuning, it won't be that apt to understand the bio-medical vocabulary. BERT will not be able to give proper context to biomedical words. Example The man had pulmonary oedema. This sentence will be a bit strange to BERT as it does not encounter these words, reducing it's a chance to understand the sentence. The Custom Neural Network has no training similar to BERT and training it on the data similar to google requires a lot of resources. BIO and CLINICAL\BIO BERT are trained on a medical corpus, and hence they will be more apt to understand the context to medical language. This gives them an advantage in identifying and assigning a proper token and thus creating a better vector. CLINICAL BERT is better as it is trained on the specific columns of the data being used and fine-tuning it to predict ICD9 code is very efficient as it can easily establish relations between the codes and the given text context pattern. MetaMap Tagger uses a different approach. It performs a lexical/syntactic analysis and uses these tokens generated to match them to the description of the ICD9 codes. In turn, it generates the code since it is dependent on the text being coherent and believes that the information does not have any syntactical anomaly. In cases of lengthy and incoherent text or text with anomalies, Metamap Tagger should be slower. There are two ways to compare the NNs to MetaMap Tagger:

1. Measure their performance to a human-annotated gold standard.
2. Measure model's performance with and without the MetaMap tagger

The results that are obtained were quite close to the expected value; The BERT performs slightly less than it's standardised performance on the regular and medical text. Clinical\Bio-BERT performs equal to its standardised performance on the medical text. It performs the best among all the modules used. It has worked as expected, a bit better than BERT. Since the Meta-Tagger's performance is not known for the project, the performance estimation is not as expected on the text with anomalies as it generates faulty output. There is no direct measure available but an indirect measure can be done from its performance on I2B2 2010 entities. This gives an idea of how well it can identify the context dependencies.

6 Appendix

6.1 Definitions

A few concepts used in the report are explained briefly. The RNNs (Recurrent Neural network) loops the output to input to simulate a memory element. On this principle, a Long-Short Term Memory (LSTM) is built. LSTMs modify the information my multiplication and addition and generate a cell state. Attention involves focusing on a subset of the text to solve the problems of LSTMs CNNs (convolutional neural networks) utilises a convolution function to convolve over the output and then feeds it back to the input. Transformers use these CNNs with self-attention or multi attention and from an encoding layer called encode and decoding layer called decoder, and they are joined together to form a transformer. Detailed information on BERT, BIO-BERT, MetaMap Tagger and other neural networks can be found from the references.[7, 3, 6, 2]

6.2 Intermediate Tokenization Results

```
Tokenization BIOBERT
INFO:tensorflow:tokens: [CLS] [ * * 215 ##1
 - 5 - 29 * * ] 11
 : 09 am ct c - spine w / o contrast clip #
 [ * * clip number
  ( radio ##logy ) 236 ##55 * * ] reason :
   pl ##s eva ##l
  interval change admitting diagnosis :
  arrest
\ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
 _ _ _ _ _ _ _ _ [ * * hospital 3 * * ] medical
 condition :
  84 year old woman with den ##s fx , acute ##ly
```

```
   not moving
   upper ex ##tre ##mit ##ies reason for this
   examination : pl ##s eva ##l interval change
    no contra ##ind ##ication ##s for iv contrast

INFO:tensorflow:tokens: [CLS] [ * * 215 ##1 -
5 - 29 * * ] 11
: 09 am ct c - spine w / o contrast clip # [
 * * clip number
( radio ##logy ) 236 ##55 * * ] reason : pl
##s eva ##l
interval change admitting diagnosis :
arrest
_ [ * * hospital 3 * * ] medical condition :
 84 year old
woman with den ##s fx , acute ##ly not
moving upper ex
##tre ##mit ##ies reason for this
examination : pl ##s
eva ##l interval change no contra
##ind ##ication ##s for iv contrast
```

Bibliography

[1] Neural Machine Translation of Rare Words with Subword Units,
Rico Sennrich and Barry Haddow and Alexandra Birch,
School of Informatics, University of Edinburgh
https://arxiv.org/pdf/1508.07909.pdf

[2] MetaMap Lite: an evaluation of a new Java implementation of MetaMap
Dina Demner-Fushman, Willie J Rogers, and Alan R Aronson
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6080672/

[3] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
https://arxiv.org/pdf/1810.04805.pdf

[4] Enhancing Clinical Concept Extraction with Contextual Embeddings
Yuqi Si Jingqi Wang Hua Xu Kirk Roberts
School of Biomedical Informatics
https://arxiv.org/pdf/1902.08691.pdf

[5] Relation Extraction from Clinical Narratives Using Pre-trained Language Models
Qiang Wei, Zongcheng Ji,Yuqi Si, Jingcheng Du, Jingqi Wang, Firat Tiryaki, Stephen Wu, Cui Tao, Kirk Roberts, and Hua Xu,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7153059/

[6] BioBERT: a pre-trained biomedical language representation model for biomedical text mining
Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim ,Sunkyu Kim, Chan Ho So and Jaewoo Kang,
https://academic.oup.com/bioinformatics/article/36/4/1234/5566506

[7] Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning
https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90

[8] Precision and recall

https://en.wikipedia.org/wiki/Precision_and_recall

[9] MIMIC-III

https://mimic.physionet.org/about/mimic/

[10] 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3168320/