B.Sc. (Business Analytics) Dissertation

Deep learning based models for analysis of clinical notes

By

Wang Tengda

Department of Information Systems and Analytics

School of Computing

National University of Singapore

2020/2021

B.Sc. (Business Analytics) Dissertation

Deep learning based models for analysis of clinical notes

By

Wang Tengda

Department of Information Systems and Analytics

School of Computing

National University of Singapore

2020/2021

Project No: H234200

Advisor: Vaibhav Rajan

Deliverables:

Report: 1 Volume

Abstract

As an essential part of a patient's post-hospital care plan, clinicians need to write summaries when a patient is discharged from the hospital, the process of which has shown to be challenging and time-consuming. Inspired by the AI-assisted note writing, we leverage nursing notes in the Electronic Health Records (EHRs) and evaluate the use of neural abstractive summarization to automatically generate discharge summaries using these notes. Two models have been studied in detail, namely the pointer-generator network and the transformer. We train the models on the publicly available de-identified EHR dataset MIMIC-3 and compare the results with the reference summaries using ROUGE scores. The models are able to learn the templates found in many reference summaries and reproduce the contents to certain extent. However, the accuracy of the generated summaries can be compromised sometimes due to information loss and incorrect prediction of some factual details. We then critically analyze the results obtained and discuss possible interpretation of the trained models. Last but not least, we provide recommendation for future experiments.

Subject Descriptors:

I.1.2 Algorithms

I.2.6 Learning

I.2.7 Natural Language Processing

Keywords:

Machine Learning, Deep Learning, Abstractive Summarization, Electronic Health Records, Discharge Summaries

Implementation Software and Hardware:

Python 3, Ubuntu 18.04.5 LTS, Nvidia CUDA 10.1

Acknowledgement

I would like to extend my utmost appreciation to NUS computing for giving me this chance to work on my final year project. This is a perfect opportunity for my academic and professional development. I'll remember this amazing journey for the rest of my life.

I wish to express my most sincere gratitude to Professor Vaibhav Rajan. Despite a busy schedule, he took time to learn about my progress and offered me timely feedback. His enthusiasm, attention to detail and warning character set a high standard for me to learn from. I truly appreciate the guidance from my mentor Suparna, as well as her willingness to share knowledge and collaborate with me. Many thanks should go to my project evaluator Professor Rudy Setiono for his valuable feedback. My completion of the project could not have been accomplished without their help.

I pay my deepest sense of gratitude to my parents for supporting my education and allowing me to be who I am. My decision to go for an undergraduate degree overseas at NUS would not have been possible without their vision. I would also express thanks to my extended families and friends who have helped me all along.

Table of Contents

1. Introduction

Electronic Health Record (EHR) is an electronic version of patients' medical history and contains the key administrative and clinical information related to patients' hospital stay, including their demographic information, lab test results, treatment procedures, medications, and radiology reports. Machine-learning-based models have shown effectiveness in generating previously unknown insights and supporting clinical decision-making, such as predicting mortality and understanding chronic diseases. In particular, many have leveraged novel natural language processing techniques to uncover the full potential of EHR data in free-text formats, namely the charted nursing notes written by doctors and nurses.

When a hospital stay concludes and the patient is discharged, a summary is created and it serves as the major communication channel between the treatment unit and aftercare unit. Fig. 1 below shows a schematic of a patient trajectory in a hospital. However, this process to generate the discharge summary has proved to be challenging and time-consuming for the following reasons:

1. The summary has to be written collectively as a patient is treated by a team of clinicians.
2. Doctors may forget about patient details and thus, have to refer to the previous patient records, especially when the patient stay is long.
3. Important information can be potentially overlooked in the summary as doctors need to refer to records from different sources and these records are usually lengthy.

Inspired by the neural summarisation models in Natural Language Processing, we came up with the following hypothesis: **can an AI tool be developed to effectively assist this process?** With this aim in mind, we evaluated the use of neural text summarization techniques to automatically generate discharge summaries using all the nursing notes written during the patient's hospital stay.

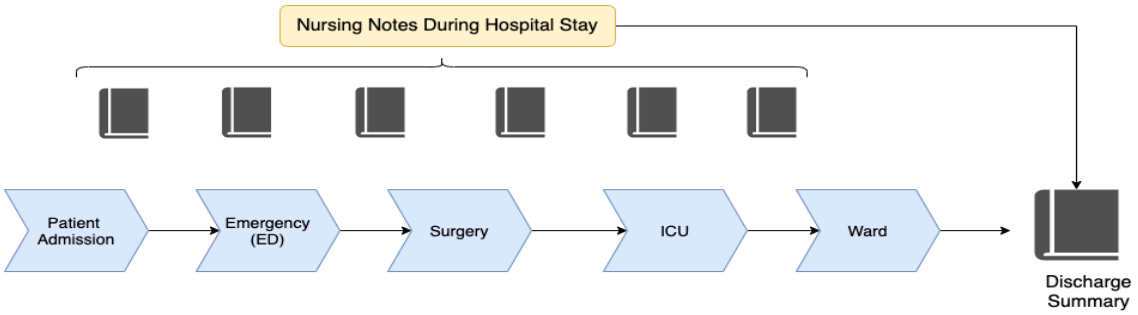

Figure 1: Hospital Stay of a Patient

2. Literature Review

There exists a few prior work for automatic clinical report writing. Biswal et al. (2020) proposed an interactive method that generates clinical reports in a sentence by sentence fashion. It extracts the most relevant sentence templates from previous reports based on doctors' anchor words and fills in the sentences with the feature representation extracted from the input X-rays and electroencephalogram (EEG) data. Liu et al. (2018) proposed a new language modelling task predicting the content of nursing notes conditioned on past data from a patient's medical records, including patient demographics, labs, medications, and past notes written. Liang et al. (2019) leveraged a manually annotated EHR dataset and proposed an automated summarization pipeline that generates extractive summaries relating to a patient's hypertension or mellitus at the point of care.

Outside of the healthcare domain, abstractive text summarization has been well studied in the natural language processing community. It is defined as a task to produce brief and succinct summaries that capture the critical information of the input text. Contrary to extractive summarization which aims to identify, extract and rearrange the important parts of the source documents, the abstractive approach generates summaries that may contain new phrases or sentences not appearing in the source text. Sequence to sequence models (seq2seq), such as Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), have become the predominant approaches for this sequence modelling and transduction task. Meanwhile, attention mechanism has been widely adopted alongside the seq2seq models to align and draw dependencies between the input sequence and output summary. See et al. (2017) presented a seq2seq hybrid pointer-generator architecture with attention and coverage that improves the accuracy of generated summary and reduces repetitions. Shi et al. (2018) provided a literature survey on different seq2seq models for abstractive text summarization, from the viewpoint of network structures, training, and summary generation algorithms. An open-source toolkit, namely Neural Abstractive Text Summarizer (NATS), was developed and tested on the CNN/ Daily Mail, Newsroom, and Bytecup datasets [6].

Despite their popularities, sequence to sequence models with attention generally suffer from the following two limitations. Firstly, it is challenging for the models to draw long-term dependencies in the sequence. Secondly, the sequential nature of their architectures prevents

parallelization during training. Thus, Vaswani et al. (2017) proposed the transformer model, a simple network architecture relying solely on the self-attention mechanism to compute representations of the sequence and dispensing with recurrence entirely [8]. The transformer architecture shows superior quality compared with the seq2seq models, as it allows more parallelization and achieves state-of-the-art results in many language generation tasks, including machine translation, question answering and abstractive summarization.

3. Model, Techniques & Evaluation

We trained a seq2seq Pointer-Generator Network [5] and a Transformer [8] to automatically generate hospital discharge summaries conditioned on the nursing notes. Section 3.1 explains the seq2seq model architecture together with its extensions in detail, including attention mechanism, copying/pointing mechanism and repetition handling techniques. Section 3.2 illustrates the different components of the transformer architecture. Section 3.3 covers the summary generation techniques while section 3.4 explains the evaluation metrics for the summarization task.

3.1. Seq2Seq Model

3.1.1. RNN-based Encoder-Decoder

The Seq2Seq Pointer-Generator Network employs a Recurrent Neural Network (RNN) based Encoder-Decoder Framework. The Encoder is a single-layer bi-directional LSTM that takes a sequence of word embeddings $(x_1, x_2, \dots, x_n)$ as input and produces a sequence of encoder hidden states $(h_1, h_2, \dots, h_n)$. The final hidden state of the encoder h_n can be viewed as the representation of the source document.

The decoder is a single-layer unidirectional LSTM that is used to generate output summary. It takes the last encoder vector h_n as input. At each decoding step t, it receives the word embeddings of the decoded token at the previous step and produces a decoder state s_t. A Multi-Layer Perceptron (MLP) takes the decoder state as input and uses a $softmax$ function to find the most probable word from the vocabulary at the current decoding time step t.

$$P_{vocab} = softmax(W_{MLP} s_t + b_{MLP}) \quad (1)$$

W_{MLP}, b_{MLP} are learnable parameters from the MLP network. The decoder will stop decoding when it produces a special token marking the end of the document.

The training loss at each time step t is the negative log-likelihood of the target word w_t. The training loss of the decoded sequence $(w_1, w_2, \ldots, w_m)$ of length m is defined as the average of the loss at each time step.

$$loss_t = -\log\left(P_{vocab}(w_t)\right) \tag{2}$$

$$loss_m = \frac{1}{m}\sum_{t=1}^{m} loss_t \tag{3}$$

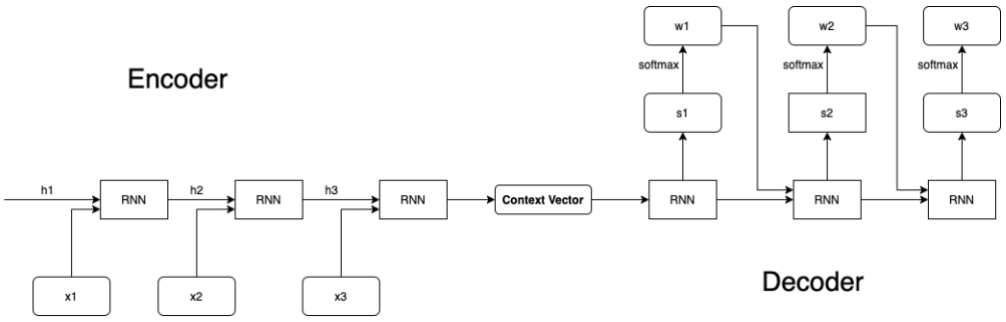

Figure 2: Encoder-Decoder Architecture

3.1.2. Attention Mechanism

Attention distribution is added to the Encoder-Decoder framework to better align the input documents and the output summary. At decoding time t, the attention distribution α^t can be calculated using different methods.

$$e_i^t = \begin{cases} h_i s_t \\ h_i W_{align} s_t \\ V^T \tanh(W_h h_i + W_s s_t + b_{attn}) \end{cases} \tag{4}$$

where $W_{align}, V^T, W_h, W_s, b_{attn}$ are learnable parameters.

$$\alpha^t = softmax(e^t) \tag{5}$$

This attention mechanism guides the decoder which part of the input documents it should focus on at the current decoding step. A context vector at decoding step t can be calculated as the weighted average of the source words based on the attention distribution α^t.

$$c_t = \sum_{i=1}^{n} \alpha_i^t h_i \tag{6}$$

This context vector c_t is concatenated with the decode state vector s_t to generate the summary token, and equation (1) is updated as follows.

$$P_{vocab} = softmax(W_{MLP}(V'[s_t, c_t] + b') + b_{MLP}) \quad (7)$$

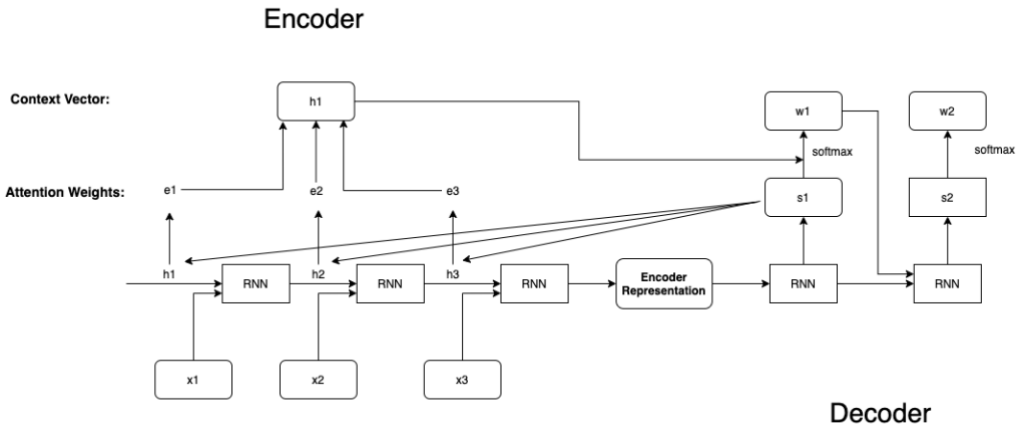

Figure 3: Adding Attention to Encoder-Decoder Framework

3.1.3. Pointer-Generator Network Architecture

Extractive summarization usually performs better than the abstractive approach in terms of ROUGE evaluation as it directly extracts the salient information from the input documents [6]. Meanwhile, the regular RNN-based encoder-decoder framework cannot handle out-of-vocabulary (OOV) words. Pointer-Generator is a novel architecture that combines extractive and abstractive summarization, with a pointer to copy words in the source documents and a generator to produce novel words [5].

At each decoding step t, the model calculates a probability p_{gen} to generate words and $1 - p_{gen}$ to copy words from the source documents based on the attention distribution α^t

$$p_{gen} = \sigma(W_c c_t + W_s s_t + W_x x_t + b_{gen}) \quad (8)$$

where σ is the sigmoid function, c_t is the context vector calculated from the attention distribution, s_t is the decoder state and x_t is the input word embedding at decoder time step t. $W_c,\ W_s,\ W_x,\ b_{gen}$ are learnable parameters.

The extended vocabulary is the combination of all words appearing in the input documents and the constructed vocabulary. The probability distribution of words over this extended vocabulary can be calculated as follows:

$$P(word) = p_{gen} \times P_{vocab}(word) + (1 - p_{gen}) \times \sum_{i:w_i=word} \alpha_i^t \quad (9)$$

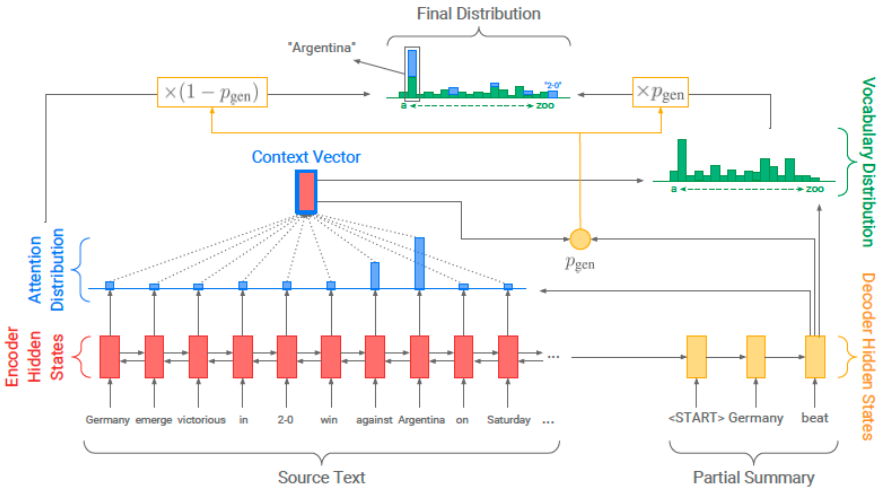

Figure 4: Pointer-Generator [5]

3.1.4. Repetition Handling

Abstractive text summarization usually suffers from repetition of words and sentences in the generated summary, as the attention mechanism tends to ignore the past alignment information [6]. Two repetition handling techniques, namely coverage mechanism and temporal attention, are studied and explored in our project.

3.1.4.1. Temporal Attention

Temporal attention prevents the model from attending the same parts of a source article by normalizing the past attention weights. The temporal attention score can be modified from the original attention score e_i^t in equation (4).

$$e_i^{t\,temporal} = \begin{cases} \exp(e_i^t) & if\ t = 1 \\ \dfrac{\exp(e_i^t)}{\sum_{k=1}^{t-1} e_i^k} & otherwise \end{cases} \tag{10}$$

3.1.4.2. Coverage Mechanism

A coverage vector is defined as the sum of the attention distribution of all previous steps.

$$u^t = \sum_{i=1}^{t-1} \alpha^i \tag{11}$$

It is used as an additional input to equation (4) to calculate the attention score, so that the model is aware of where it has attended to before. In addition to the negative log-likelihood loss in equation (3), a coverage loss is defined below to avoid repeatedly attending to the same place.

$$CovLoss_t = \sum_j \min(\alpha_j^t, u_j^t) \tag{12}$$

3.2. Transformer Model

3.2.1. Self-Attention

Self-attention, or intra-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [8]. It has been widely used and proved useful in many NLP tasks, such as text comprehension, machine translation and abstractive summarization. Transformer adopts self-attention to discard recurrence completely. The section below illustrates the self-attention mechanism used in the model, namely scaled dot-product attention and multi-head attention.

3.2.1.1. Scaled Dot-Product Attention

The input to the scaled dot-product attention consists of a query of dimension d_k and key-value pairs of dimensions d_k, d_v respectively. The query, keys and values are all vectors. This method calculates the dot products between the query and all keys, before dividing the results by $\sqrt{d_k}$. The query and the keys must have the same dimension in order to take dot products. A softmax function is then applied to obtain the weights on the values.

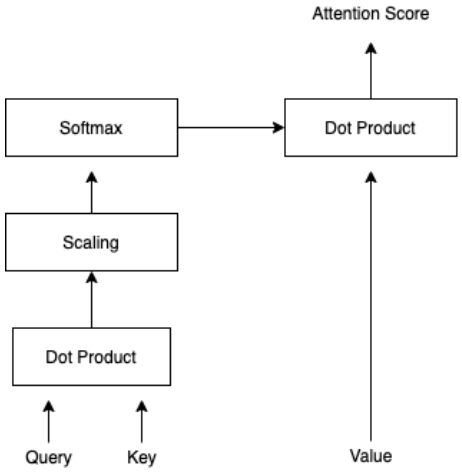

Figure 5: Scaled Dot-Product Attention

In practice, the queries, keys and values are packed into matrix form (Q, K, V respectively) and the scaled dot-product attention is calculated as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{13}$$

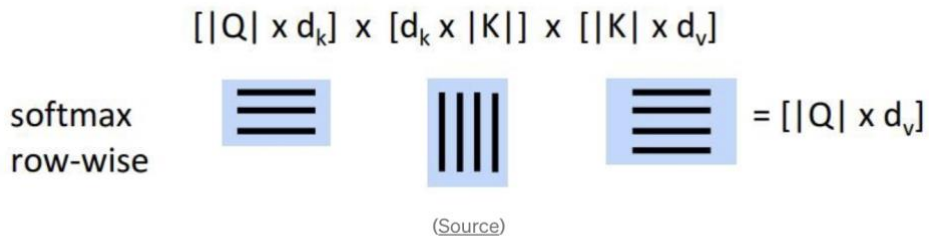

Figure 6: Dot-Product Attention in Matrix Form

The scaling of $\sqrt{d_k}$ is taken to counteract very large dot products and prevent softmax function from falling into regions with extremely small gradients [8].

3.2.1.2. Multi-Head Attention

Instead of computing the attention score once, multi-head attention mechanism calculates the scaled dot-product attention score multiple times in parallel. More specifically, each Q, K, V matrix is projected into lower dimension using different learned weight matrices. This method takes scaled dot-product on each of the projected versions of the queries, keys and values (the result of which is called a head). These heads are then concatenated and linearly projected to get the final results.

$$MultiHead(Q,K,V) = Concat(head_1, \ldots\ldots, head_h)W^O$$
$$where\ head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$$
$$and\ W^O, W^Q, W^K, W^V\ are\ learnable\ projection\ matrices \quad (14)$$

Compared with a single dot-product, the multi-head attention method allows the model to jointly attend to different representations learned.

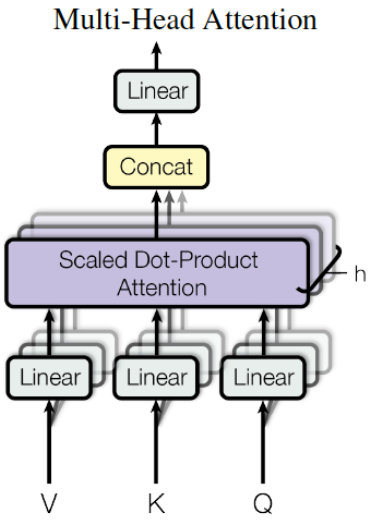

Figure 7: Multi-Head Attention Mechanism [8]

3.2.2. Positional Encoding

Since self-attention mechanism ignores information about the ordering of the input sequence, positional encoding is added to the input embedding to indicate the position of a token within the sequence. Transformer uses the sine and cosine function for positional encodings. They have the same dimensionalities as input embeddings so that the two can be summed directly.

$$PE(pos, 2i) = \sin\left(\frac{pos}{1000^{2i}}\right) \tag{15}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{1000^{2i}}\right) \tag{16}$$

where pos is the position of a word in the sequence and i is the dimension in the embedding.

3.2.3. Transformer Architecture

Fig. 8 below shows a diagram of the transformer architecture. In this section, the different components of the model are explained in detail.

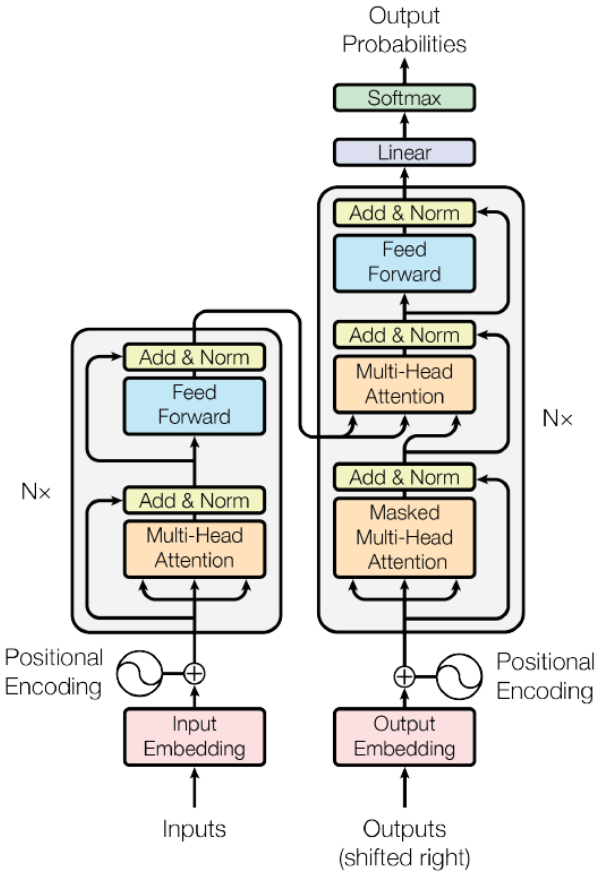

Figure 8: Transformer Architecture [8]

3.2.3.1. Encoder

Encoder (the left part of Fig. 8) can be viewed a stack of N = 6 identical layers. For each layer, it has two sub-layers, namely a multi-head self-attention layer and a point-wise fully-connected feedforward neural network. The encoder employs residual connection and layer normalisation for both two sub-layers.

For the self-attention sub-layer, the queries, keys and values all come from the output of the previous layer in the encoder. Therefore, each position in the encoder can attend to all positions in the previous layer of the encoder.

3.2.3.2. Decoder

Similar to the encoder, decoder (the right part of Fig. 8) also consists of N = 6 identify layers. For each layer, in addition to the two sub-layers (self-attention and feedforward neural network) mentioned in the encoder, it contains another multi-head attention sub-layer, called encoder-decoder attention sub-layer, over the output of the encoder stack.

For the encoder-decoder attention sub-layer, the queries come from the previous decoder layer, while the keys and values come from the output of the encoder. This allows every position in the decoder to attend over all positions in the encoder, which mimics the usual attention mechanism in seq2seq architecture.

The self-attention sub-layer in the decoder is similar to the one in the encoder and allows each position in the decoder to attend to all positions in the decoder up to and including that position. In order to preserve the auto-regressive property in the decoder, masking (setting value to $-\infty$) is added to prevent one position from attending to subsequent positions.

3.2.3.3. Fully connected feedforward neural network

The fully-connected feedforward neural network in both the encoder and decoder applies to all positions independently and identically. This point-wise network consists of two linear transformation layers with RELU activation.

3.2.3.4. Embedding and Softmax Function

Similar to the seq2seq architecture, transformer uses embedding to convert the input token to vector representations. It also leverages the softmax function to predict probabilities of the next token based on the output of the decoder.

3.3. Summary Generation

Beam search is used in the summary generation step. By setting a beam size of K, the algorithm maintains K most probable sequence at every decoding step. At decoding time t, each of the K sequences from step $t-1$ is expanded with K most probable tokens from the current step, resulting in K^2 candidate sequences. Of these K^2 sequences, only the top K ones with the highest log-likelihood are considered and the rest are removed. Beam search can be run efficiently and generally improve the fluency and readability of the generated summary.

Algorithm 1: Beam search algorithm for decoding seq2seq models.

Input: Source article x, beam size B, summary length T, model parameters θ;
Output: B-best summaries;

Initialize:
Output sequences $Q^{\text{seq}} = [\text{SOS}]_{B\times T}$;
Accumulated probabilities $Q^{\text{prob}} = [1.0]_{B\times 1}$;
The last decoded tokens $Q^{\text{word}} = [\text{SOS}]_{B\times 1}$;
States (hidden and cell states for LSTM) $Q^{\text{states}} = [0.0]_{B\times|h_t^d|}$;
Context vectors $Q^{\text{ctx}} = [0.0]_{B\times|z_t^e|}$;

Compute $(h_1^e, h_2^e, \ldots, h_J^e)$ with encoder;
Update Q^{states} with encoder states;
for t=1, T **do**
  Initialize candidates $Q^{\text{cand,seq}}$, $Q^{\text{cand,prob}}$, $Q^{\text{cand,word}}$, $Q^{\text{cand,states}}$, $Q^{\text{cand,ctx}}$ by repeating Q^{seq}, Q^{prob}, Q^{word}, Q^{states} and Q^{ctx} B times, respectively;
  for b=1, B **do**
    Compute $P_\theta(y_{t,b}^{\text{cand}}|y_{<t,b}, x)$ using decoder LSTM cell with input $(h_1^e, h_2^e, \ldots, h_J^e)$, Q_b^{word}, Q_b^{states} and Q_b^{ctx};
    Select the top-B candidate words $y_{t,b,b'}^{\text{cand}}$, where $b' = 1, 2, \ldots, B$;
    Select corresponding probability $P_\theta(y_{t,b,b'}^{\text{cand}}|y_{<t,b}, x)$, hidden states $h_{t,b,b'}^d$, cell states $c_{t,b,b'}^d$ and context vector $z_{t,b,b'}^e$;
    Update elements of $Q_{b',b,t}^{\text{cand,seq}}$, $Q_{b',b}^{\text{cand,word}}$ with $y_{t,b,b'}^{\text{cand}}$;
    Update elements of $Q_{b',b}^{\text{cand,states}}$ with $h_{t,b,b'}^d$ and $c_{t,b,b'}^d$;
    Update elements of $Q_{b',b}^{\text{cand,ctx}}$ with $z_{t,b,b'}^e$;
    Update $Q_{b',b}^{\text{cand,prob}}$ with Eq.(44);
  end
  Flatten $Q^{\text{cand,prob}}$ and choose B best hypotheses;
  Update Q_t^{seq}, Q^{prob}, Q^{word}, Q^{states}, Q^{ctx} with corresponding candidates.
end

Figure 9 Beam Search

3.4. Evaluation

We used ROUGE to evaluate the quality of the summary. ROUGE is a commonly used metric in the text summarization task by comparing the overlapping of n-grams and the longest common subsequence (LCS) between the generated summary and the reference summary. The ROUGE metric measures the overlap in terms of recall, precision, and F1 score. More specifically, a ROUGE-n recall of x% means that x% of the n-grams in the reference summary also present in the generated summary while a ROUGE-n precision of y% means y% of the n-grams in the generated summary can be found in the reference summary. The F1 score is a weighted harmonic mean of precision and recall.

$$ROUGE_n\ recall = \frac{|reference\ summary\ \wedge generated\ summary|}{|reference\ summary|} \tag{17}$$

$$ROUGE_n\ precision = \frac{|reference\ summary\ \wedge generated\ summary|}{|generated\ summary|} \tag{18}$$

$$where\ |X|\ represents\ the\ number\ of\ ngrams\ in\ X$$

$$ROUGE_n\ F1 = \frac{2 \times recall \times precision}{recall + precision} \tag{19}$$

4. Experiments

In our experiment, we trained both a seq2seq Pointer-Generator Network [5] and a Transformer model [8]. The data acquisition and filtering steps can be found in section 4.1. Section 4.2 covers the pre-processing performed for the acquired data. Section 4.3 explains the exploratory data analysis. The predictive results and the interpretation of the two models can be found in section 4.4 and section 4.5 respectively.

4.1. Data Acquisition and Filtering

The data are taken from the NOTEEVENTS table in the MIMIC-3 (`Medical Information Mart for Intensive Care`) database [1]. It is a large de-identified dataset comprising information related to patients admitted to critical care units at a hospital. It includes medications, laboratory measurements, observations, and most importantly in our use case, notes charted by care providers. The dataset has been widely adopted in research since its publication.

A description of the NOTEEVENTS table can be found in Appendix 7.1. The raw notes are stored in the TEXT column. Fig. 10 below shows an example note in the dataset.

```
Admission Date:  [**2151-7-16**]       Discharge Date:  [**2151-8-4**]


Service:
ADDENDUM:

RADIOLOGIC STUDIES:  Radiologic studies also included a chest
CT, which confirmed cavitary lesions in the left lung apex
consistent with infectious process/tuberculosis.  This also
moderate-sized left pleural effusion.

HEAD CT:  Head CT showed no intracranial hemorrhage or mass
effect, but old infarction consistent with past medical
history.

ABDOMINAL CT:  Abdominal CT showed lesions of
T10 and sacrum most likely secondary to osteoporosis. These can
be followed by repeat imaging as an outpatient.



                          [**First Name8 (NamePattern2) **] [**First Name4 (NamePattern1) 1775**] [**Last Name (Nam
ePattern1) **], M.D.  [**MD Number(1) 1776**]

Dictated By:[**Hospital 1807**]
MEDQUIST36

D:  [**2151-8-5**]  12:11
T:  [**2151-8-5**]  12:21
JOB#:  [**Job Number 1808**]
```

Figure 10: Raw Notes in the NOTEEVENTS Table

For each note in the table, it contains different sections including admission detail, past medical history, hospital course, and discharge instructions. A detailed description of these sections can be found in table 1 below.

Section	Description
Admission Detail	Information related to admission such as admission date, hospital , patient date of birth and sex etc
Major Surgical Procedures	Surgical procedures performed during the patient's hospital stay
History of Present Illness	A brief description of the past illness history of the patient
Past Medical History	Past medications and treatment of the patient
Social History	Social status of the patient (occupation, habits, marital status)
Family History	Family illness history
Brief Hospital Course	A brief description of the patient's hospital stay
Discharge Disposition	Place of discharge
Discharge Medicine	Instructions related to medications after discharge
Discharge Condition	The patient's condition on discharge (stable, death)
Discharge Instructions	Instructions for the after-care unit

Table 1: Description of Sections in Discharge Summaries

In our experiment, the reference summary is taken from the brief hospital course section in the discharge summary, uniquely identified by a hospital episode ID. We took this section as reference as it is a succinct summary of the entire hospital stay. Fig.11 below shows an example hospital course. All nursing notes from the same hospital stay that are written before this discharge summary are concatenated and used as input documents. On average, each concatenated input document contains 2389 tokens and each reference summary has 321 tokens.

```
Brief Hospital Course:
Mr. [**Known lastname 303**] was admitted to the hospital and examined by the
Bariatric service. He had no abdominal pain but had Dilaudid
prior to his transfer.  His cardiac work up was negative and due
to the finding of pneumoperitoneum on abd CT he was taken to the
Operating Room for an exploratory laparotomy. He had a
perforated marginal ulcer and underwent a partial gastrectomy,
omental patch to ulcer and a cholecystectomy. He tolerated the
procedure well and returned to the ICU in stable condition.
His pain was controlled with a Dilaudid PCA and he remained
hemodynamically stable.
He was subsequently transferred to the Surgical floor for
further management.

Due to his recent surgery and length of time prior to taking
fluids he had a PICC line placed for TPN which began on [**2119-5-14**]
through a right AC PICC line.  This was continued until [**2119-5-18**].
In the interim he had an upper GI on [**2119-5-16**] which showed no
anastomotic leak and he gradually began a stage 1 diet.  This
was slowly advanced over the next 24 hours to stage 3 and he had
no trouble with nausea or fullness.

His abdominal wound was healing well and he was up and walking
independently.  He remained free of any pulmonary complications
post op by using his incentive spirometer.  Roxicet was
effective in treating his incisional pain and he was generally
improving every day.

After an uncomplicated course he was discharged to home on
[**2119-5-18**] and will follow up with Dr. [**Last Name (STitle) **] in 2 weeks.
```

Figure 11. Hospital Course Section

For each hospital stay, there are on average 1.13 discharge summaries written. For simplicity, we focus on hospital episodes with only one discharge summary written, which contains 42849 records in total. To ensure that the hospital course is a summary of not just the ICU period but the entire hospital stay, we only retain hospital episodes with discharge summaries written on the same date on which the patient is discharged from the hospital.

4.2. Pre-Processing

We heuristically tokenized the document by parsing it into different segments and chipping away irrelevant structures using regular expressions until only prose is left. Below is the algorithm for heuristic tokenization:

Step 1: Find all section headers.

a. A section header usually ends with ":" and all non-stop words in the header is capitalized

Step 2: Parse the documents into different sections based on the section headers obtained in the step above.

Step 3: For each section found in step two, further parse the section by finding obvious delimiters and remove these delimiters, such as '___' and '-----'.

Step 4: Find bullet list and break the list into one line.

Step 5: Replace confidential information marked as [***XXXX***], including hospital name, patient names and admission date, with a special UNK token.

Step 6: Case-folding: convert all tokens to lower cases.

Step 7: For the prose left after step 6, use spaCy *Med7* library for both sentence level and word level tokenization.

We used the algorithm above to tokenise both the source notes and the reference summaries. In addition, each sentence in the reference summary is separated by the <s> and </s> tokens. The reference summary and the source document are divided by the <sec> token. The processed data format is shown in Fig. 12.

```
<s> sentence one in summary 1 . </s> <s> sentence two in summary 1 </s> <sec> concatenated source document 1 . \n
<s> sentence one in summary 2 . </s> <s> sentence two in summary 2 </s> <sec> concatenated source document 2 . \n
<s> sentence one in summary 3 . </s> <s> sentence two in summary 3 </s> <sec> concatenated source document 3 . \n
<s> sentence one in summary 4 . </s> <s> sentence two in summary 4 </s> <sec> concatenated source document 4 . \n
...
...
```

Figure 12: Data Format

We used pre-trained Flair embedding weights [13] to embed the tokens. These weights have a dimension of 200 and are trained with 5% of PubMed abstracts data using LSTM of 3 hidden layers and 1150 hidden states. Using pre-trained weights allows the model to better capture the syntactic and semantic meaning of a word and could potentially lead to better performance of the neural network.

During the processing steps above, we also created a vocabulary file which is used later during training. We removed stop words from the dictionary. In our case, both source documents and reference summaries share the same vocabulary. The words are sorted in descending order in terms of the frequency in which they appear in the documents. By setting a minimum threshold of 20 (the number of occurrences of each word in the corpus should be larger than or equal to 20), the generated vocabulary has a size of 98,488. The rest of the words are replaced with the out-of-vocabulary token. The most frequent and least frequent words of this vocabulary can be found in Appendix 7.2.

We partitioned the processed data into training, validation, and test set. Each patient episode can appear in one set only. The sizes of these sets are 35517, 3947, and 4385 respectively.

In our experiment, the hospital course section is extracted from the discharge summaries using regular expressions. As the nursing notes in MIMIC-3 do not follow a consistent structure, the current extraction script cannot handle all the variants to effectively locate and retrieve this section. Thus, the reference summaries used may be subject to errors. Furthermore, we also find that these noisy records are either excessively short (only extract part of the hospital course) or excessively long (extract contents not part of this section). Therefore, we filtered the MIMIC-3 data based on the length of the reference summary, and only considered records from the 30th percentile to 90th percentile. We name this dataset as filtered-MIMIC3, which is a subset of the original MIMIC-3 data. Filtered-MIMIC3 contains 26253 records in total and is further partitioned into training, validation and test set with a size of 21264, 2363, 2626 each.

4.3. Exploratory Analysis

To understand how well a summary can be generated from the nursing notes, we studied the correlation between the two data sources by calculating the overlaps in both tokens and medical entities. This overlap is represented by the proportion of tokens/entities in a discharge summary that appears in the corresponding nursing notes.

$$overlap = \frac{|\, summary \ \wedge nursing\ notes|}{|\, summary\,|} \tag{17}$$

$$where\ |X|\ is\ the\ number\ of\ tokens/entities\ in\ document\ X$$

We relied on the spaCy med7 library [12] to extract medical entities from both the summaries and input nursing notes. The extracted entities include dosage, drug, duration, form, frequency, route and strength. A detailed explanation of these entities can be found in Appendix 7.5. During calculation, we removed stop words and punctuations from both input documents and summaries.

Table 2 below shows a summary of the overlap statistics for both tokens and medical entities. On average, about 44% of the tokens and 26% of the medical entities appearing in the summary can be located in the corresponding nursing notes.

	Token Overlap	Entity Overlap
Mean	43.57%	26.06%
Median	41.79%.	16.67%

Table 2: Overlap Statistics

Two histogram plots are shown in Fig. 13, one for tokens overlap (left hand side) and another for entity overlap (right hand side). The x-axis is the overlap statistics measured in percentage term while the y-axis is $\frac{number\ of\ documents\ in\ percentage\ term}{10}$, where 10 is the number of bins in the histogram. For example, more than 40% of the documents have entity overlap smaller than 10%. Meanwhile, almost 18% of the documents have token overlap between 30% and 40%.

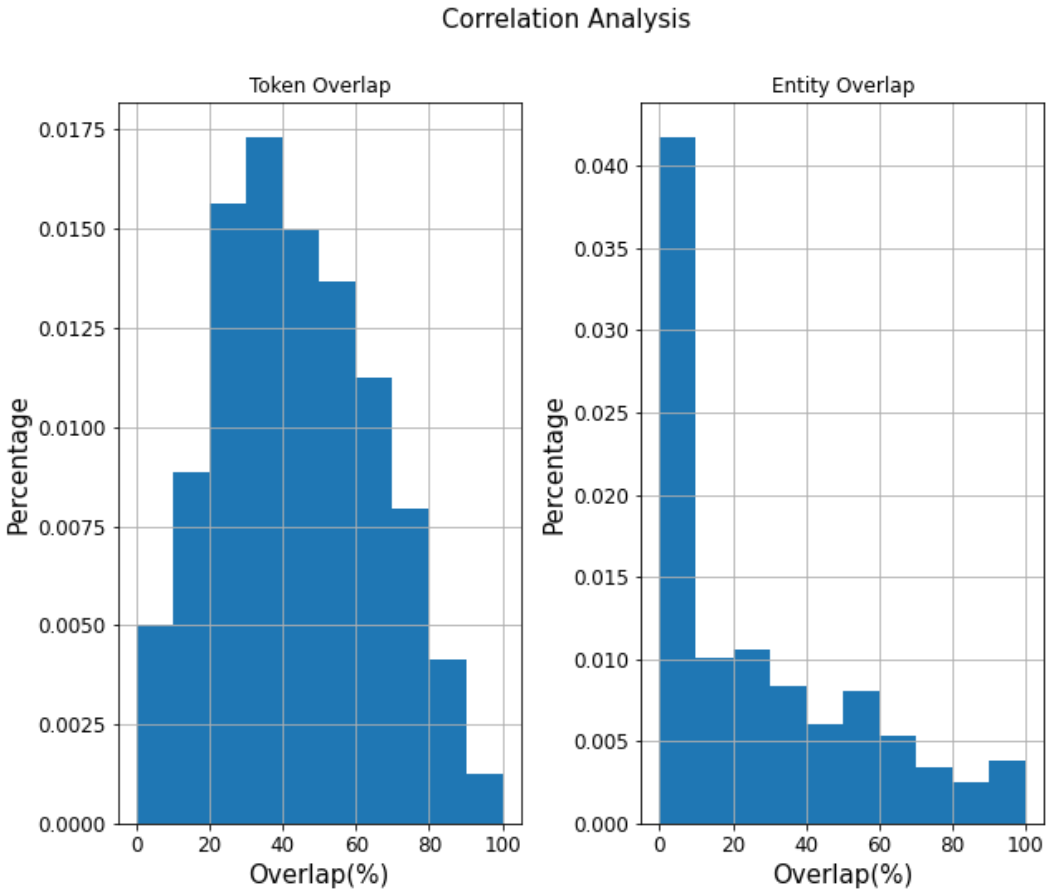

Figure 13:Distribution of Overlaps in MIMIC-3

4.4. Results

This section covers the results from both the pointer-generator and the transformer. Section 4.4.1 describes model training and prediction, with the ROUGE scores on all datasets used in our experiment. Section 4.4.2 analyses the summaries generated by the models.

4.4.1. Predictive Accuracy

Model	Data	ROUGE-L F1 Score
Pointer-Generator	MIMIC-3	**0.19**
	Filtered-MIMIC	**0.14**
Transformer	MIMIC-3	**0.11**
	NEWSROOM	**0.31**

Table 3: ROUGE-L Scores

Pointer-Generator

We trained a pointer-generator network for 35 epochs with a learning rate of 0.0001 and a batch size of 16, on both MIMIC-3 and Filtered-MIMIC3. We used Luong concatenation [16] to calculate attention scores and coverage mechanism to handle repetition. The detailed model architecture can be found in Appendix 7.3. These hyperparameters are chosen so that the average log-likelihood loss during both training and validation can converge, and the validation accuracy is approximately the same as the training accuracy.

Our best performing pointer-generator achieves an average ROUGE-L (Longest Common Subsequence) F1 score of 0.19 on MIMIC-3 test set and 0.14 on Filtered-MIMIC3 test set.

Transformer

The transformer model trained has 6 layers, 8 heads, a batch size of 16 and a model dimension of 256. We trained the model for 30 epochs using Adam optimizer with learning rate decay. Similar to the trained pointer-generator, the chosen hyperparameters ensure that the loss (average log-likelihood loss) during both training and validation can converge, and the validation accuracy is on the same scale as the training accuracy.

We observe that the performance of the transformer is relatively poor, achieving a ROUGE-L F1 score of 0.11 on the test set. As the model did not perform as expected on the MIMIC-3

dataset, we tested its capabilities on the NEWSROOM, a summarization dataset of 1.3 million articles and summaries written by authors and editors in newsrooms of 38 majors news publications. These high-quality summaries demonstrate high diversity of summarization styles and are widely accepted as the benchmark dataset for the neural summarization task [11].

We pre-processed the NEWSROOM data following the same steps in Section 4.2, except that we directly tokenized the data with the spaCy sentence tokenizer and word tokenizer. We initialized the word embedding for the tokens randomly. The vocabulary file has a size of 209498 after setting a minimum threshold of 15. The rest of the words are replaced with the out-of-vocabulary token. The processed data are partitioned into training, testing and validation with a size of 992943, 108609 and 108652 respectively.

The trained transformer model on NEWSROOM has the same hyperparameter setting as the one on MIMIC-3. Due to time constraint, we trained the model for 10 epochs (instead of 30 epochs) as the data size of NEWSROOM is larger. The model managed to obtain an average ROUGE-L of 0.31 on the test set, much higher than the results on MIMIC-3.

The complete ROUGE evaluation scores can be found in Appendix 7.4.

4.4.2. Analysis of Results

Pointer-Generator

Based on our observation, we find that the generated summaries from pointer-generator network mostly observe the pattern shown below. We hypothesise that this is because the reference summaries share similar vocabularies and many of them also follow the exact same template.

The patient was admitted to the hospital and brought to the operating room on UNK where he suffered from XXX symptoms. He underwent XXX treatment and tolerated the procedure well. The patient was transferred to XXX on day XXX. The patient was discharged from the hospital on day XXX.

Figure 14: Pattern in the Generated Summaries

We also find that summaries with high ROUGE scores are usually short in length. The top two summaries with the highest ROUGE-L score are shown in the table below. These summaries manage to capture the salient information in the documents with minimal errors. They have

relatively higher overlaps with the reference summaries, and they are also fluent and human-readable. However, some factual details such as numbers are produced wrongly.

Example 1 ROUGE-L F1 = 0.728	Reference	**respiratory: infant admitted to newborn intensive care unit and placed on cpap. she remained on cpap** for the first 48 hours of life, **at which time she transitioned to room air. she has remained in room air since that time. she has had no episodes of apnea** and bradycardia of prematurity.
	Generated	**respiratory: the infant was admitted to the newborn intensive care unit and placed on cpap. she remained on cpap** for a total of 24 hours, **at which time she transitioned to room air. she has remained in room air since that time. she has had no apnea** of prematurity. Cardiovascular: she has remained normotensive throughout her neonatal intensive care unit stay. she has had no cardiovascular issues.
	Input	input1
Example 2 ROUGE-L F1 = 0.670	Reference	the patient **underwent a coronary artery bypass graft times three on unk with left internal mammary coronary artery to left anterior descending coronary artery** , saphenous vein graft to obtuse marginal , saphenous vein graft to posterior descending coronary artery . **he tolerated the procedure well and was transferred to the intensive care unit in stable condition . he was extubated on postoperative day** zero . he was stable hemodynamically and **was transferred to the floor on postoperative day one** . his subsequent stay on the floor was uncomplicated . his pacing wires were discontinued on postoperative day three . by this time he was ambulating well and his pain was under control with po analgesics . he was ready for discharge home on postoperative day four .
	Generated	the patient was taken to the operating room on unk, where he **underwent coronary artery bypass grafting times four with left internal mammary artery to the left anterior descending coronary artery**, reverse saphenous vein graft to the obtuse marginal artery, and saphenous vein graft to the posterior descending artery**. the patient tolerated the procedure well and was transferred to the cardiothoracic intensive care unit in stable condition. the patient was extubated on postoperative day** one and **was transferred to the floor on postoperative day one.**
	Input	input2

Table 4: Examples from Pointer-Generator Network on MIMIC-3

Below is one example summary with low ROUGE-L score. It is noticed that this reference summary is much longer than the two examples shown above.

Example 3 ROUGE-L F1 = 0.08	Reference	59 year old man with pmh significant for cad s p cabg and stents ; atrial fibrillation ; etoh abuse ; and recurrent pancreatitis admitted with active severe pancreatitis . . # # gi . # . pancreatitis : due to etoh . amylase and lipase steadily improved since admission to nearly normal levels . no complication seen on abdominal ct . he initially had prerenal azotemia and was bolused , urine output fully recovered by day of discharge . - able to tolerate po clears by day prior to discharge , no epigastric pain reported on discharge on unk , able to ambulate well . - dilaudid pca initially , d c'ed on unk . - had some minor non - bloody diarrhea night

		before discharge perhaps s p stool softeners and laxatives ; k repleted and bowel regimen d c'ed . # # pulmonary . # low grade fever : - initial suspicion of ? aspiration pneumonia based on cxr w rul infiltrates but cxr gradually resolved over course of stay . - continued w some low grades temps and fever until day prior to admission when temp fully normalized . # pulmonary edema : - found to be volume overloaded on exam and by cxr early in stay , perhaps s p aggressive initial rehydration in wake of prerenal azotemia on admission . - tte showed normal ef and no sign of diastolic dysfunction but sub - optimal windows . - excellent urine output by day before discharge w o need for diuresis . . # wheezes on lung exam : - no prior hx of such wheezes . no h o copd . improved with albuterol nebs . - could be due to volume overload , but could also be from metoprolol . - received nebs prn . - consider pulmonary function tests as an outpatient . # # neuro psych tox . # etoh abuse- mr . unk has a history of etoh withdrawal per records and required prolonged intubation for this at an osh . he has a reported history of allergy to ativan , but this actually seems to have been a result of oversedation from too much ativan administered . - he was maintained on a diazepam ciwa scale for etoh withdrawal throughout hospitalization . given hypertension and tachycardia on admission , treated with diazepam 10iv x 1 and monitored throughout . - on unk he had a ? withdrawal picture pulling out iv and becoming somewhat agitated in late afternoon , mildly diaphoretic . - on morning of unk he vigorously expressed desire to leave hospital , became somewhat agitated . however , though hypertensive in both cases of agitation , he had no tachycardia . seemed frustrated with situation but aware and no indication of ms changes or confusion . no indication of dt . - placed on fall precautions . - given thiamine and folate qd . # # renal . # . anion gap- he initially p w a significant anion gap of 28 on his admission labs . likely from saline repletion ; gap resolved by end of admission w no indication of acidosis . . # . hypokalemia- potassium of 3.0 early on day of discharge as noted above , probably s p diarrhea night before . k normalized on po , iv repletion . . # # cardio . # . cad htn- he has a history of one vessel cabg and multiple stents in unk . - ruled out for mi x three sets of cardiac enzymes . - continued on asa . - restarted fenofibrate and quinapril . - was started on metoprolol in the unk for hypertension . htn continued to be a problem during stay and he spiked to 170s on unk midnight . his quinapril dose increased to 10 mg po daily and discharged on 20 mg . sent home on the metoprolol . - recommend follow - up and adjustment of bp meds as needed . . # . atrial fibrillation- he has a long history of atrial fib but is not anticoagulated . - continued on amiodarone . # # heme . # low hct : apparently hemodilution , stable throughout hospitalization to baseline for him , though may benefit from further work - up .
	Generated	the patient was admitted to the icu for close monitoring . the patient was started on a heparin drip . the patient was started on a heparin drip . the patient was transferred to the floor on unk . on unk , the patient was taken to the operating room for exploratory laparotomy and lysis of adhesions . the patient tolerated the procedure well . the patient tolerated the procedure well and was transferred to the icu in stable condition .
	Input	input3

Table 5: Another Example from Pointer-Generator Network on MIMIC-3

The summaries generated by the pointer-generator on Filtered-MIMIC3 are similar to the one on MIMIC-3, in view of both styles and quality. Furthermore, summaries with high ROUGE

scores in Filtered-MIMIC3 are usually short as well. This could be one possible explanation why the ROUGE evaluation score for Filtered-MIMIC3 (0.14) is lower than MIMIC3 (0.19), as Filtered-MIMIC3 excludes summaries in the lower 30th percentile of MIMIC3 with respect to summary length.

One example generated by Pointer-Generator on Filtered-MIMIC3 is shown in Table 6.

Example 4 ROUGE-L F1 = 0.46	Reference	**the patient was a direct admission to the operating room on unk** where the patient **underwent** replacement of ascending aorta and **aortic** hemiarch . **please see the operative report for details** . in summary he had : redo sternotomy , replacement of ascending aorta and hemiarch using deep hypothermic circulatory arrest with a 30-mm vascutek dacron tube graft , catalog number unk , serial number unk . his cardiopulmonary bypass time was 119 minutes , with a crossclamp time of 75 minutes , and circulatory arrest time of 18 minutes . **he tolerated the procedure well and post - operatively was transferred to the cvicu in stable condition for recovery and invasive monitoring** . **pod 1 found the patient extubated , alert and oriented and breathing comfortably . the patient was neurologically intact and hemodynamically stable , weaned from inotropic and vasopressor support** . beta blocker was initiated and the patient was gently diuresed toward the preoperative weight . the patient was transferred to the telemetry floor for further recovery . chest tubes and pacing wires were discontinued without complication . heparin was initiated as a bridge to coumadin for his mechanical valves . the patient was evaluated by the physical therapy service for assistance with strength and mobility . by the time of discharge on pod # 7 the patient was ambulating freely , the wound was healing and pain was controlled with oral analgesics . the patient was discharged on to home with vna services , in good condition with appropriate follow up instructions advised .
	Generated	**the patient was brought to the operating room on unk** where the patient **underwent aortic** valve replacement with coronary artery bypass graft x 4 **. please see operative note for details . overall the patient tolerated the procedure well and post - operatively was transferred to the cvicu in stable condition for recovery and invasive monitoring** . **pod 1 found the patient extubated and alert and oriented and breathing comfortably . the patient was neurologically intact and hemodynamically stable and weaned from inotropic and vasopressor support .**
	Input	input4

Table 6: Example from Pointer-Generator on Filtered-MIMIC3

Transformer

Overall, the generated summaries from the transformer model on MIMIC-3 cannot reproduce the salient information in the reference. They suffer from repetition and their sentence structures lack variations. Table 7 below shows one example.

Example 5 ROUGE-L F1 = 0.28	Reference	respiratory : the patient was initially brought to the nicu on room air , was placed on cpap . on day of life # 2 was weaned back to room air . the patient never required intubation or surfactant therapy throughout his hospital stay and has been doing well on room air . cardiovascular : the patient has been noted to have a soft , intermittent murmur located at the left upper sternal border . noted to have stable blood pressures and heart rate . fluids , electrolytes and nutrition : the patient was initially made n.p.o . and started on iv fluids of d10-w . on day of life # 2 the patient was started p.o . pg feeds of breast milk and or enfamil premature . on day of life # 16 the patient reached full p.o . feeds of breast milk and or enfamil premature 24 . weight at the time of discharge is currently 2480 g
	Generated	respiratory**: the infant was initially placed on cpap and then weaned to room air on day of life 1** and has been stable on room air since that time . the infant has been stable on room air cardiovascular . the infant has been hemodynamically stable throughout hospitalization fluids electrolytes and nutrition . the infant was initially npo on admission enteral feedings were initiated on day of life 2 and advanced to full enteral feedings on day of life 3 . the infant has been ad lib feeding breast milk or enfamil 24 calories were increased to 24 calories per ounce of breast milk 24 calories per ounce of breast milk 24 calories per ounce the infant is currently feeding ad lib p o feeding breast milk 24 calories per ounce of breast milk 24 calories per ounce with iron supplementation . the infant is voiding and stooling normally gastrointestinal . the infant has not required any phototherapy hematology . the infant has not required any blood transfusions infectious disease cbc and blood culture were screened on admission to the nicu . the infant was started on ampicillin and gentamycin at the time of admission . the infant was clinically well and the infant has not had any further issues . the infant has not had any further issues . sensory hearing screen has not yet been performed psychosocial . the infant has been involved with the infant .
	Input	input5

Table 7: Example from Transformer on MIMIC-3

As for the NEWSROOM data, the generated summaries from transformer are fluent and human-readable without repetition. Moreover, important factual details such as quantities and dates are also preserved correctly. Below are some example summaries generated by the transformer on NEWSROOM.

Example 5 ROUGE-L F1 = 0.8	Reference	**a dramatic plunge in the value of bitcoin has led an internet forum** to issue a **suicide warning** to **investors of the virtual currency**
	Generated	**a dramatic plunge in the value of bitcoin has led an internet forum** devoted to the **virtual currency** to **issue a suicide warning to its investors** .
	Input	screenshot of suicide warning on r / bitcoin , reddit \'s forum for the virtual currency (reddit) a dramatic plunge in the value of bitcoin has led an internet forum devoted to the virtual currency to issue a suicide warning to its investors . bitcoin \'s value fell by half after btc china , china \'s biggest bitcoin trading platform , announced on wednesday that it had banned deposits in yuan following new restrictions reportedly imposed by the central bank . shortly after the collapse , a post appeared at the top of reddit \'s bitcoin forum with details

		about suicide - prevention hotlines in a number of countries , including australia . uk man scours landfill for bitcoin fortune " remember , it \'s just money , " the post advised . a subsequent post by the forum \'s moderators temporarily banned " memes , price posts and emotional posts " to " reduce the volume of short , meaningless submissions " and focus on bitcoin news . however , another post criticised the " short - sightedness " of the suicide warning and the forum \'s reaction to the bitcoin plunge . by late wednesday bitcoin \'s value stood at 2245 yuan ($ 416) , down almost 50 percent from the start of the week and more than 70 percent lower than its all - time peak of 7588.88 yuan ($ 1406) on november 30 . launched in 2009 as the invention of a mysterious computer guru who goes by the pseudonym satoshi nakamoto , bitcoins can be exchanged online for real money or used to buy goods and services on the internet . the currency is not regulated by any government . do you have any news photos or videos ?
Example 6 ROUGE-L F1 = 0.89	Reference	apollo global management and cedar fair said on **tuesday** that the two have agreed to end their proposed **$ 2.4 billion** leveraged buyout **after the amusement park operator 's investors** indicated **dissatisfaction with the deal** .
	Generated	global management and cedar fair , the amusement park operator , said on **tuesday** that the two had agreed to end their proposed **$ 2.4 billion** leveraged buyout **after cedar fair s investors** indicated **dissatisfaction with the deal** .
	Input	apollo global management and cedar fair , the amusement park operator , said on tuesday that the two had agreed to end their proposed $ 2.4 billion leveraged buyout after cedar fair s investors indicated dissatisfaction with the deal . cedar fair s investors , including a major stakeholder , q investments , had voiced discontent with the $ 11.50 a unit that apollo would pay under the deal . two - thirds of cedar fair unit holders needed to approve the deal for it to go through . shares in cedar fair rose nearly 1.6 percent , to $ 12.38 , tuesday morning , after having fallen more than 7 percent earlier in the day . the board has heard from cedar fair unit holders and it is apparent that the merger transaction does not have the required level of investor support , dick kinzel , cedar fair s chairman and chief executive , said in a statement . cedar fair , whose board had supported the transaction , will cancel a unit holder meeting scheduled for thursday . the meeting had been postponed from last month , as the company sought to gain more time to solicit votes . as part of the deal agreement , cedar fair has agreed to reimburse apollo for $ 6.5 million in expenses . cedar fair announced concurrently that it had enacted a shareholder rights plan to deflect hostile takeovers , which would kick in if an investor gained a 20 percent stake in the company . the plan is scheduled to expire in 2013 . go to joint press release via pr newswire go to previous item from the deal professor.

Table 8: Examples from Transformer on NEWSROOM

4.5. Interpretation

After investigation, we attribute the unsatisfactory model performance on the MIMIC-3 dataset to the following two reasons. Firstly, some of the information in the reference summaries are missing in the input documents (Section 4.5.1). Secondly, the processed sequences in the MIMIC-3 dataset are too long for the trained model to generate accurate results (Section 4.5.2).

4.5.1. Information Loss

For the MIMIC-3 dataset, about 56% of the tokens and 74% of the medical entities appearing in the summary cannot be located in the corresponding nursing notes according to the correlation analysis in section 4.3. While for the NEWSROOM data, only about 13% of the tokens in the summary are absent in the input documents on average. As seen from the histogram plot in Fig. 15, most of the documents in NEWSROOM have token overlap greater than 90%

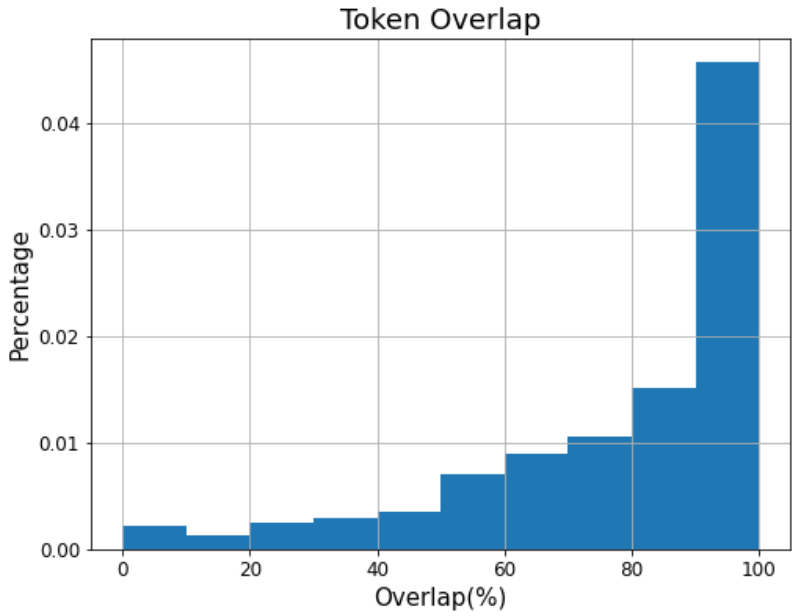

Figure 15. Distribution of Token Overlap for NEWSROOM

To this end, we believe that the amount of information loss in the MIMIC-3 data restrains the models from generating summaries close to the reference, even though abstractive summarization can produce novel words/phrases not appearing in the source documents.

4.5.2. Long Context Sequence

The table below summarizes the average sequence length measured in terms of the number of tokens after stop words and punctuations removal for both MIMIC-3 and NEWSROOM.

Avg Sequence Length	Input Documents	Reference Summaries
MIMIC-3	2389	321
NEWSROOM	312	15

Table 4: Average Sequence Length

As shown in the table, the average sequence length in MIMIC-3 is much longer compared with NEWSROOM, for both input documents and the reference summaries. Due to the sequential nature of RNN based models, RNN cannot be parallelized within a sequence along the time-step dimension during training and validation, and hence training on long sequences becomes particularly challenging due to computation time and memory constraints of GPUs [6]. Meanwhile, although the transformer model is more parallelizable compared with RNNs, one major limitation of the model is the quadratic dependency of memory on the sequence length due to its full attention calculation [8, 14].

Therefore, it is hard for both models to learn long-term dependencies in the sequence. In fact, as we are training the model on a single Tesla V100S-PCIE-32GB GPU, we need to truncate the sequence length to 1000 for input sequence, 400 for summary sequence and reduce the batch size to 16 to avoid memory issues. This means information beyond this threshold will be lost and as a result, the trained models have reduced capabilities dealing with longer sequences.

As a verification of the above hypothesis, for the pointer-generator model on MIMIC-3 data, we have found a negative correlation of -0.68 between the ROUGE-L score and the summary length

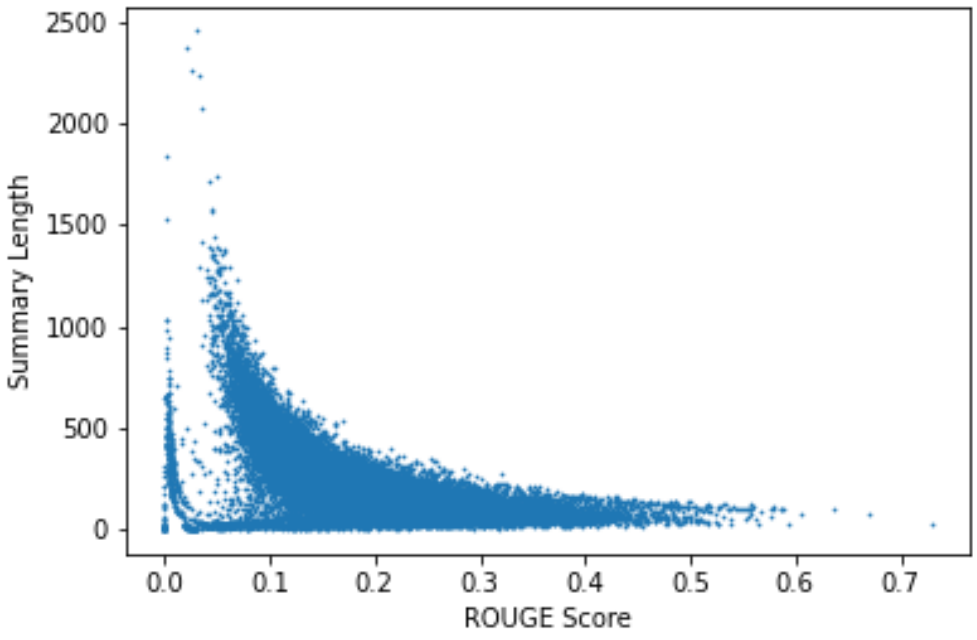

Figure 16. Scatter Plot between Summary Length and ROUGE Score

5. Conclusion

5.1. Summary

In conclusion, we introduced a new conditional language modelling task on Electronic Health Records (EHRs). This project focused on evaluating the use of neural abstractive summarization to automatically generate hospital discharge summaries conditioned on nursing notes in the de-identified MIMIC-3 dataset. We first demonstrated how to effectively process and represent the unstructured text in EHRs. We then explored the use of the pointer-generator network and the transformer model for abstractive summarization. The trained pointer-generator network can generate human-readable summaries of a fixed pattern. Although the preliminary results from the transformer showed no improvement over the pointer-generator network, we conducted a proof-of-concept experiment on the capability of the transformer by training the same model on the NEWSROOM dataset. Last but not least, we investigated and analyzed why the two models failed to generate satisfactory results on the MIMIC-3 dataset.

5.2. Future Recommendation

As shown in section 4.5, our trained pointer-generator network and transformer suffered from information loss and long context sequence. Future efforts could attempt to address these two limitations.

For example, it is recommended to include additional data sources in the input documents as a supplement of nursing notes. These data sources may include patient demographic information and lab test results available in other tables from the MIMIC-3 dataset. However, this may bring additional challenges to combine unstructured text with structured data during training. Meanwhile, as mentioned in section 4.2, the current hospital course extraction script is not robust enough to handle all variants, and thus the reference summaries used in our experiment may contain errors. Although we did a simple filtering based on the summary length, the generated summaries show no improvement in terms of ROUGE scores (see section 4.4). Therefore, we recommend future attempts to refine this extraction process and improve the overall data quality.

On the other hand, as both the pointer-generator network and transformer cannot take advantage of long context sequence, we recommend using more advanced models to better

handle long input sequence for future experimentations, such as BigBird sparse attention mechanism [14] and transformer with memory-compressed attention [15].

6. References

1. Johnson, A., Pollard, T., Shen, L. *et al.* MIMIC-III, a freely accessible critical care database. *Sci Data* **3,** 160035 (2016). https://doi.org/10.1038/sdata.2016.35
2. Siddharth Biswal et al. *CLARA: Clinical Report Auto-completion. 2020. arXiv: 2002.11701* [cs.LG].
3. Peter J. Liu. "Learning to Write Notes in Electronic Health Records". In: *arxiv* (2018). URL: https://arxiv.org/abs/1808.02622
4. Jennifer Liang, Ching-Huei Tsou, and Anaya Poddar. "A Novel System for Extractive Clinical Note Summarization using HER Data". In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop.* Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 46-54. DOI: 10 . 18643/v1/w19-1906. URL: https//www.aclweb.org/anthology/W19-1906.
5. Abigail See, Peter J, Liu, and Christopher D. Manning. *Get To The Point: Summarization with Pointer-Generator Networks.* 2017. arXiv: 1704.04368 [cs.CL].
6. Tian Shi et al. *Neural Abstractive Text Summarization with Sequence-to-Sequence Models.* 2020. arXiv: 1812.02303 [cs.CL].
7. Tian Shi, Ping Wang, and Chandan K. Reddy. *LeafNATS: An Open-Source Toolkit and Live Demo System for Neural Abstractive Text Summarization.* 2019. arXiv: 1906.01512 [cs.CL].
8. Ashish Vaswani et al. *Attention Is All You Need.* 2017. arXiv: 1706.03762 [cs.CL].
9. Peter J. Liu et al. *Generating Wikipedia by Summarizing Long Sequences.* 2018. arXiv: 1801.10198 [cs.CL].
10. Emily Alsentzer et al. *Publicly Available Clinical BERT Embeddings.* 2019. arXiv: 1904.03323 [cs.CL].
11. Max Grunsky, Mor Naaman, and Yoav Artzi. *Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies.* 2020. arXiv: 1804.11283 [cs.CL].
12. Andrey Komilitzin et al. *Med7: a transferable clinical natual language processing model for electronic health reocrds.* 2020. arXiv: 2003.01271 [cs.CL].

13. Alan Akbik, Duncan Blythe, and Roland Vollgraf. “Contextual String Embeddings for Sequence Labeling”. In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638-1649.
14. Manzil Zaheer et al. *Big Bird: Transformers for Longer Sequence.* 2021. arXiv: 2017 . 14062 [cs.LG]
15. Peter J. Liu et al. *Generating Wikipedia by Summarizing long Sequences.* 2018. arXiv: 1801.10198 [cs.CL]
16. Minh-Tang Luong, Hieu Pham, and Christopher D. Manning. *Effective Approaches to Attention-based Neural Machine Translation.* 2015. arXiv: 1508 . 04025 [cs.CL]

7. Appendix

7.1. NOTEEVENTS TABLE

SUBJECT_ID: patient ID, unique to a patient

HADM_ID: hospital episode ID, unique to a hospital stay

CHARTDATE: date at which the note was charted

CATEGORY: the type of the note recorded

ISERREOR: whether the note is an error, as identified by a physician

TEXT: contains the note contents

7.2. Vocabulary File

- MIMIC-3

	Most frequent 10 words	Least frequent 10 words
1	in	nicarpidine
2	on	sandbag
3	pt	breath;congestive
4	mg	nursey
5	and	snuggle
6	am	rn.p
7	ml	51p
8	left	bm32bp
9	right	sweetease
10	patient	beclavent

- NEWSROOM

	Most frequent 10 words	Least frequent 10 words
1	,	grinspan
2	the	bardini
3	.	surinamese
4	to	getequal
5	a	semitruck
6	of	hydractinia
7	and	gruenewald

8	in	kaoir
9	that	tursunov
10	-	wetware

7.3. Model Architecture

- Pointer-Generator Network

```
The vocabulary size: 98488
{}
{'decoder2proj': Linear(in_features=256, out_features=128, bias=False),
 'embedding': natsEmbedding(
  (embedding): Embedding(98488, 128)
  (proj2vocab): Linear(in_features=128, out_features=98488, bias=True)
),
 'encoder': natsEncoder(
  (encoder): LSTM(128, 256, batch_first=True, bidirectional=True)
),
 'encoder2decoder': natsEncoder2Decoder(
  (encoder2decoder): Linear(in_features=512, out_features=256, bias=True)
  (encoder2decoder_c): Linear(in_features=512, out_features=256, bias=True)
),
 'pgdecoder': PointerGeneratorDecoder(
  (rnn_): LSTMCell(384, 256)
  (encoder_attn_layer): AttentionEncoder(
    (attn_en_in): Linear(in_features=512, out_features=256, bias=True)
    (attn_de_in): Linear(in_features=256, out_features=256, bias=False)
    (attn_cv_in): Linear(in_features=1, out_features=256, bias=False)
    (attn_warp_in): Linear(in_features=256, out_features=1, bias=False)
  )
  (decoder_attn_layer): AttentionDecoder(
    (attn_en_in): Linear(in_features=256, out_features=256, bias=True)
    (attn_de_in): Linear(in_features=256, out_features=256, bias=False)
    (attn_warp_in): Linear(in_features=256, out_features=1, bias=False)
  )
  (attn_out): Linear(in_features=1024, out_features=256, bias=True)
  (pt_out): Linear(in_features=1152, out_features=1, bias=True)
)}
```

7.4. Detailed ROUGE Scores

- Pointer-Generator on MIMIC-3

```
---------------------------------------------
1 ROUGE-1 Average_R: 0.51919 (95%-conf.int. 0.51404 - 0.52412)
1 ROUGE-1 Average_P: 0.13759 (95%-conf.int. 0.13488 - 0.14066)
1 ROUGE-1 Average_F: 0.19451 (95%-conf.int. 0.19152 - 0.19753)
---------------------------------------------
1 ROUGE-2 Average_R: 0.17557 (95%-conf.int. 0.17175 - 0.17939)
1 ROUGE-2 Average_P: 0.04634 (95%-conf.int. 0.04481 - 0.04798)
1 ROUGE-2 Average_F: 0.06580 (95%-conf.int. 0.06396 - 0.06771)
---------------------------------------------
1 ROUGE-3 Average_R: 0.07372 (95%-conf.int. 0.07095 - 0.07664)
1 ROUGE-3 Average_P: 0.02085 (95%-conf.int. 0.01968 - 0.02217)
1 ROUGE-3 Average_F: 0.02930 (95%-conf.int. 0.02781 - 0.03082)
---------------------------------------------
1 ROUGE-4 Average_R: 0.03780 (95%-conf.int. 0.03552 - 0.04038)
1 ROUGE-4 Average_P: 0.01145 (95%-conf.int. 0.01047 - 0.01255)
1 ROUGE-4 Average_F: 0.01599 (95%-conf.int. 0.01468 - 0.01736)
---------------------------------------------
1 ROUGE-L Average_R: 0.49749 (95%-conf.int. 0.49250 - 0.50241)
1 ROUGE-L Average_P: 0.13089 (95%-conf.int. 0.12828 - 0.13379)
1 ROUGE-L Average_F: 0.18548 (95%-conf.int. 0.18261 - 0.18837)
---------------------------------------------
1 ROUGE-W-1.2 Average_R: 0.19350 (95%-conf.int. 0.19133 - 0.19556)
1 ROUGE-W-1.2 Average_P: 0.08644 (95%-conf.int. 0.08461 - 0.08853)
1 ROUGE-W-1.2 Average_F: 0.10402 (95%-conf.int. 0.10258 - 0.10558)
---------------------------------------------
1 ROUGE-S* Average_R: 0.29566 (95%-conf.int. 0.29054 - 0.30056)
1 ROUGE-S* Average_P: 0.02589 (95%-conf.int. 0.02474 - 0.02722)
1 ROUGE-S* Average_F: 0.03633 (95%-conf.int. 0.03509 - 0.03767)
---------------------------------------------
1 ROUGE-SU* Average_R: 0.30274 (95%-conf.int. 0.29789 - 0.30759)
1 ROUGE-SU* Average_P: 0.02769 (95%-conf.int. 0.02644 - 0.02912)
1 ROUGE-SU* Average_F: 0.03804 (95%-conf.int. 0.03677 - 0.03943)
```

- Pointer-Generator Network on Filtered-MIMIC3

```
1 ROUGE-1 Average_R: 0.48526 (95%-conf.int. 0.48338 - 0.48719)
1 ROUGE-1 Average_P: 0.08932 (95%-conf.int. 0.08884 - 0.08985)
1 ROUGE-1 Average_F: 0.14823 (95%-conf.int. 0.14752 - 0.14906)
---------------------------------------------
1 ROUGE-2 Average_R: 0.13882 (95%-conf.int. 0.13760 - 0.14018)
1 ROUGE-2 Average_P: 0.02558 (95%-conf.int. 0.02529 - 0.02586)
1 ROUGE-2 Average_F: 0.04245 (95%-conf.int. 0.04202 - 0.04290)
---------------------------------------------
1 ROUGE-3 Average_R: 0.05064 (95%-conf.int. 0.04984 - 0.05154)
1 ROUGE-3 Average_P: 0.00953 (95%-conf.int. 0.00935 - 0.00971)
1 ROUGE-3 Average_F: 0.01576 (95%-conf.int. 0.01548 - 0.01605)
---------------------------------------------
1 ROUGE-4 Average_R: 0.02314 (95%-conf.int. 0.02255 - 0.02375)
1 ROUGE-4 Average_P: 0.00448 (95%-conf.int. 0.00434 - 0.00462)
1 ROUGE-4 Average_F: 0.00738 (95%-conf.int. 0.00716 - 0.00760)
---------------------------------------------
1 ROUGE-L Average_R: 0.46668 (95%-conf.int. 0.46487 - 0.46857)
1 ROUGE-L Average_P: 0.08577 (95%-conf.int. 0.08529 - 0.08626)
1 ROUGE-L Average_F: 0.14237 (95%-conf.int. 0.14167 - 0.14314)
---------------------------------------------
1 ROUGE-W-1.2 Average_R: 0.18410 (95%-conf.int. 0.18323 - 0.18495)
1 ROUGE-W-1.2 Average_P: 0.05510 (95%-conf.int. 0.05478 - 0.05542)
1 ROUGE-W-1.2 Average_F: 0.08250 (95%-conf.int. 0.08209 - 0.08294)
---------------------------------------------
1 ROUGE-S* Average_R: 0.26012 (95%-conf.int. 0.25842 - 0.26199)
1 ROUGE-S* Average_P: 0.00952 (95%-conf.int. 0.00942 - 0.00963)
1 ROUGE-S* Average_F: 0.01805 (95%-conf.int. 0.01787 - 0.01824)
---------------------------------------------
1 ROUGE-SU* Average_R: 0.26733 (95%-conf.int. 0.26560 - 0.26915)
1 ROUGE-SU* Average_P: 0.00998 (95%-conf.int. 0.00988 - 0.01008)
1 ROUGE-SU* Average_F: 0.01889 (95%-conf.int. 0.01871 - 0.01909)
```

- Transformer on MIMIC-3

```
1 ROUGE-1 Average_R: 0.14314 (95%-conf.int. 0.13599 - 0.15012)
1 ROUGE-1 Average_P: 0.18977 (95%-conf.int. 0.18520 - 0.19442)
1 ROUGE-1 Average_F: 0.13995 (95%-conf.int. 0.13578 - 0.14389)
---------------------------------------------
1 ROUGE-2 Average_R: 0.02763 (95%-conf.int. 0.02509 - 0.03004)
1 ROUGE-2 Average_P: 0.03585 (95%-conf.int. 0.03354 - 0.03834)
1 ROUGE-2 Average_F: 0.02570 (95%-conf.int. 0.02408 - 0.02738)
---------------------------------------------
1 ROUGE-3 Average_R: 0.00965 (95%-conf.int. 0.00837 - 0.01090)
1 ROUGE-3 Average_P: 0.01210 (95%-conf.int. 0.01073 - 0.01360)
1 ROUGE-3 Average_F: 0.00863 (95%-conf.int. 0.00781 - 0.00952)
---------------------------------------------
1 ROUGE-4 Average_R: 0.00397 (95%-conf.int. 0.00329 - 0.00469)
1 ROUGE-4 Average_P: 0.00487 (95%-conf.int. 0.00414 - 0.00567)
1 ROUGE-4 Average_F: 0.00343 (95%-conf.int. 0.00298 - 0.00394)
---------------------------------------------
1 ROUGE-L Average_R: 0.10704 (95%-conf.int. 0.10207 - 0.11197)
1 ROUGE-L Average_P: 0.14434 (95%-conf.int. 0.14082 - 0.14784)
1 ROUGE-L Average_F: 0.10560 (95%-conf.int. 0.10267 - 0.10831)
---------------------------------------------
1 ROUGE-W-1.2 Average_R: 0.01848 (95%-conf.int. 0.01738 - 0.01966)
1 ROUGE-W-1.2 Average_P: 0.08563 (95%-conf.int. 0.08300 - 0.08857)
1 ROUGE-W-1.2 Average_F: 0.02627 (95%-conf.int. 0.02527 - 0.02724)
---------------------------------------------
1 ROUGE-S* Average_R: 0.03685 (95%-conf.int. 0.03289 - 0.04092)
1 ROUGE-S* Average_P: 0.04557 (95%-conf.int. 0.04330 - 0.04814)
1 ROUGE-S* Average_F: 0.02486 (95%-conf.int. 0.02344 - 0.02628)
---------------------------------------------
1 ROUGE-SU* Average_R: 0.03746 (95%-conf.int. 0.03345 - 0.04159)
1 ROUGE-SU* Average_P: 0.04750 (95%-conf.int. 0.04507 - 0.05015)
1 ROUGE-SU* Average_F: 0.02540 (95%-conf.int. 0.02396 - 0.02682)
```

- Transformer on NEWSROOM

```
1 ROUGE-1 Average_R: 0.26703 (95%-conf.int. 0.25276 - 0.28101)
1 ROUGE-1 Average_P: 0.57844 (95%-conf.int. 0.55924 - 0.59580)
1 ROUGE-1 Average_F: 0.33349 (95%-conf.int. 0.31859 - 0.34697)
---------------------------------------------
1 ROUGE-2 Average_R: 0.18867 (95%-conf.int. 0.17275 - 0.20407)
1 ROUGE-2 Average_P: 0.37015 (95%-conf.int. 0.34594 - 0.39268)
1 ROUGE-2 Average_F: 0.22799 (95%-conf.int. 0.21113 - 0.24457)
---------------------------------------------
1 ROUGE-3 Average_R: 0.16626 (95%-conf.int. 0.15013 - 0.18200)
1 ROUGE-3 Average_P: 0.31480 (95%-conf.int. 0.29038 - 0.33836)
1 ROUGE-3 Average_F: 0.19805 (95%-conf.int. 0.18130 - 0.21497)
---------------------------------------------
1 ROUGE-4 Average_R: 0.15456 (95%-conf.int. 0.13873 - 0.17036)
1 ROUGE-4 Average_P: 0.28868 (95%-conf.int. 0.26536 - 0.31275)
1 ROUGE-4 Average_F: 0.18265 (95%-conf.int. 0.16577 - 0.19942)
---------------------------------------------
1 ROUGE-L Average_R: 0.24767 (95%-conf.int. 0.23327 - 0.26230)
1 ROUGE-L Average_P: 0.52824 (95%-conf.int. 0.50821 - 0.54743)
1 ROUGE-L Average_F: 0.30742 (95%-conf.int. 0.29191 - 0.32151)
---------------------------------------------
1 ROUGE-W-1.2 Average_R: 0.10982 (95%-conf.int. 0.10274 - 0.11641)
1 ROUGE-W-1.2 Average_P: 0.45166 (95%-conf.int. 0.43260 - 0.47057)
1 ROUGE-W-1.2 Average_F: 0.16415 (95%-conf.int. 0.15466 - 0.17284)
---------------------------------------------
1 ROUGE-S* Average_R: 0.12127 (95%-conf.int. 0.10772 - 0.13507)
1 ROUGE-S* Average_P: 0.38725 (95%-conf.int. 0.36331 - 0.41021)
1 ROUGE-S* Average_F: 0.14055 (95%-conf.int. 0.12710 - 0.15342)
---------------------------------------------
1 ROUGE-SU* Average_R: 0.12612 (95%-conf.int. 0.11260 - 0.13999)
1 ROUGE-SU* Average_P: 0.40636 (95%-conf.int. 0.38323 - 0.42882)
1 ROUGE-SU* Average_F: 0.14732 (95%-conf.int. 0.13402 - 0.16012)
```

7.5. Medical Terms Extraction

Label	Concept	Description
DOSAGE	1-2, sliding scale, taper, bolus, thirty (30) ml	The total amount of a drug administered
DRUG	aspirin, lisinopril, prednisone, vitamin b, flagyl	Generic or brand name of the medication
DURATION	for 3 days, 7 days, chronic, x5 days, for five more days	The length of time that the drug was prescribed for
FORM	tablet, capsule, solution, puff, adhesive patch, disk with device	A particular configuration of the drug which it is marketed for use
FREQUENCY	once a day, b.i.d., prn, q6h, hs, every six (6) hours as needed	The dosage regimen at which the medication should be administered
ROUTE	iv, p.o. (by mouth), gtt, nasal canula, injection,	The path by which the drug is taken into the body
STRENGTH	5mg, 100 unit/ml, 50mg/2ml, 0.05%, 25-50mg	The amount of drug in a given dosage

7.6. Code Files Explanation

All codes for this project are stored in the home/tengda/summarizer folder

/data: raw data files

- NOTEEVENTS.csv
- /NEWSROOM
    - train.jsonl.gz
    - dev.jsonl.gz
    - test.jsonl.gz

/transformer_summarizer

- /data: all processed data files and pre-trained vectors used by the model
- /checkpoints: storing the model training checkpoints
- /scripts:
    - model.py: transformer architecture implementation
    - utils.py: utility functions and data structures used in model training
    - data_prep.py: to prepare and pre-process the data
    - summarizer.py: handle model training, validation and testing
    - eval_pyrouge.py: evaluate the generated summaries
- /rouge_scores: storing the evaluation scores for different experiments

/pointer_generator_net

- /model: Pointer-Generator Network (LeafNATS) implementation with training, validation, testing and ROUGE evaluation
- /data: all processed data files and pre-trained vectors used by the model
- /results: storing results file during experimentation, including trained model, generated summaries and ROUGE evaluation
- /img: storing plots and visualisations

- /experiment: all IPython notebooks used for exploration and experimentation
- /scripts
    - correlation_analysis.py: to conduct the correlation analysis for exploratory data analysis
    - data_prep.py: to create vocabulary file and generated data as input to the summarizer
    - format_notes.py: to clean up the nursing notes and discharge summaries
    - heuristic_tokenize.py: to heuristically tokenize the raw nursing notes
    - utils.py: utility functions and data structures used

7.7. List of Technical Knowledge

- Deep Learning for Natural Language Processing

I took the Stanford CS224N online course to understand deep learning techniques in natural language processing. These topics include word vectors, basic neural networks, RNN, CNN, machine translation, language model, dependency parsing, constituency parsing, Seq2Seq model, attention mechanism, and transformer network.

- Pytorch

I used Pytorch library to build and train the pointer-generator network.

- Tensorflow

I used Tensorflow library to build and train the transformer model.

- Linux & Tmux

I utilised Linux and Tmux to connect to the lab server so that I can train the models with GPU.

B.Sc. (Business Analytics) Dissertation

Deep learning based models for analysis of clinical notes

By

Wang Tengda

Department of Information Systems and Analytics

School of Computing

National University of Singapore

2020/2021