Summarization and Generation of Discharge Summary Medical Reports

Koyena Pal
Brown University
Providence, RI
koyena_pal@brown.edu

ABSTRACT

Discharge Summary Report is a form filled out by medical professionals when a patient is discharged from their hospital stay. It details the patient's medical history, current medical status, diagnosis, treatments conducted, and any discharge instructions, to name a few. It is an essential medical document used for record-keeping and reference when future doctors want to understand a patient's medical background. Including many other medical forms, medical professionals often have to manually fill out all these reports, which are very time-consuming and have caused challenges for hospitals to keep their Electronic Health Records up to date. Hence, in this paper, we present a study of creating data setups and implementing text summarization models to generate Discharge Summary Reports automatically. Our experiments demonstrate promising results in generating parts of such reports through models such as BART and T5 with F_1 ROUGE scores as high as 0.388.

INTRODUCTION

When patients are discharged from their hospital stay, one of the documents written by clinical professionals to conclude their treatment is a discharge summary report, as shown in Figures 1[1] and 2. Other medical reports can include but are not limited to nursing notes, echo, consult, and general healthcare information. These documents are essential for record-keeping and are helpful for future doctors to refer to them to understand the patient's circumstances better while they treat them. However, according to healthit.gov, an official website of the US Health Information Technology government, most challenges hospitals face are using and digitally exchanging information among hospitals and public health agencies [17]. Some of the causes listed in this report include lack of capacity (e.g., technical, staffing), interface-related issues (e.g., cost, complexity), vocabulary inconsistencies, and difficulty in extracting relevant information from Electronic Health Records (EHR). To address these issues, we propose a system that can generate text using deep learning models to automatically write one of the important documents the healthcare workers would have to complete as part of clerical reporting duties. This document would be the Discharge Summary report, whose context would be gathered from the Nursing notes present during a patient's stay. This solution can significantly reduce the complexities mentioned earlier, one of them being to reduce the time overhead of healthcare workers to complete EHR records for a patient's stay.

The deep learning models we consider as part of our solution are text summarization models. Within the deep learning research community, there is a lot of study on natural language processing models that aim to improve tasks that involve text. Among them, research studies aim to either build summarization mechanisms and/or apply sequence to sequence or other types of models to generate text for various tasks, such as news summarization [4]. The purpose behind these summarizations is generally to allow the readers to read enough text to understand a situation as quickly as possible so that they can act upon it accordingly. We would like to extend such goals to healthcare professionals in the form of summarizing nursing notes to generate parts of another report named the Discharge Summary Report, which can be used by future doctors to quickly scan information about the patient and make prompt medical decisions. Hence, by implementing such a mechanism, we are aiming to reduce time spent by healthcare workers on preparing and reading EHR reports, which would summarize essential information generated by a chosen language model.

In this paper, we study ways to generate EHR discharge summary reports automatically. We experiment with different deep learning text summarization models and create various data creation setups for training models. By doing so, we find settings that make the most use of publicly available nursing medical text documents to write the corresponding discharge summary documents. Therefore, in this paper, we explore the following research questions:

1. What data gathering setups are effective for making the goal of automatic EHR summarization possible?
2. What parts of the discharge summary report can we automatically generate using nursing notes?
3. What model settings are best for achieving high-quality summaries?

The remainder of this paper is structured as follows: Section 2 discusses related literature on text summarization and previous attempts at EHR record utilization. Section 3 describes the experiment, including more information on the dataset and the resulting setups created on which various models are trained and tested. Section 4 showcases the results gathered from that

[1] Source: http://www.himconnect.ca/

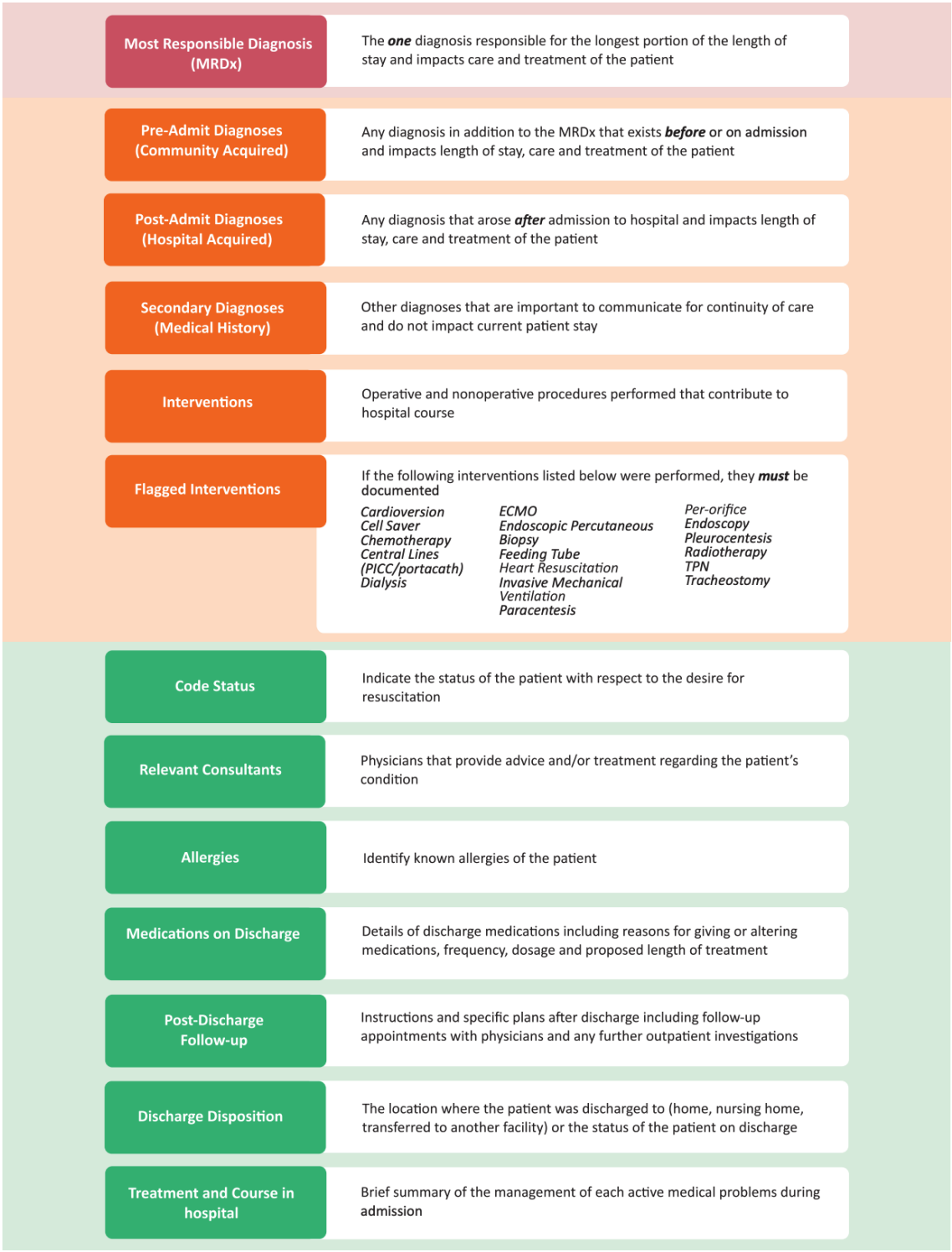

Figure 1. Discharge Summary Report Template

experiment. We continue discussing the implications of the results in Section 5. Finally, in Section 6, we conclude with an outlook on the future directions of our research.

RELATED WORKS

Text Summarization

In this section, we briefly review extractive and abstractive approaches, including several models central to this paper. Recent works on text summarization algorithms can be broadly classified based on the type of summary generated – extractive and abstractive. Extractive summarization involves taking a subset of phrases and sentences from the input documents and concatenating them to form a summary. On the other hand, abstractive summarization algorithms generate summaries based on their vocabulary and the concepts they associate with the input document.

One of the models in the extractive summarization category is the Luhn summarizer [21]. It selects sentences based on the maximum number of significant words present in a particular sentence. The significance of words is determined through Term Frequency - Inverse Document Frequency, also known as TF-IDF [1]. A common baseline model for extractive summarization is LEAD-3, which is another extractive summarization algorithm that takes the first three sentences of the input document and sets them as the document's summary. Another sub-category of algorithms is topic-based approaches. Harabagiu et al. [16], for instance, represents topic themes based on events that frequently occur over a set of documentation. They illustrate five ways of determining such frequencies – topic signatures, enhanced topic signatures, thematic signatures, modeling documents' content structure, and templates. In addition to these approaches, there are also graph-based [11, 3] and discourse-based ones [23]. The former method uses text representation in a graph where words or sentences are represented as nodes and semantically-related text elements are being connected through edges while the latter approach integrates linguistic knowledge to represent the connections within and between sentences. Apart from such methodologies, there are also many approaches based on machine learning, which can range from supervised to unsupervised frameworks. In supervised learning, there is a set of documents and related human-generated summaries with a classification of whether they are a summary or not a summary. Supervised learning algorithms require a lot of amount of such labeled data to learn the intended task. Regression [13], Naïve Bayes classification [12], and Support Vector machines (SVM) [12] are some of the supervised learning approaches. Unlike supervised learning, unsupervised learning does not require training data. It tries to identify patterns using techniques such as clustering [31] and apply them to generate a summary of a given document.

Amongst various abstractive summarization algorithms, a state-of-the-art model for text summarization is the Bidirectional and Auto-Regressive Transformer (BART) [19]. As the name suggests, BART employs a standard Transformer-based neural machine translation architecture with a bidirectional encoder and a left-to-right decoder. Its encoder behaves similar to Bidirectional Encoder Representations from Transformers (BERT), another well-known transformer [9], while its decoding nature resembles that of Generative Pre-trained Transformers (GPT) [24, 25, 6], which is a famous auto-regressor transformer. Other similar recent language models include UniLM [2], MASS [27], and XLNet [32]. Apart from BART, other popular models are Text-To-Text Transfer Transformer (T5) [26] and Longformer [5]. T5 is an encoder-decoder model that is pre-trained on various text-based language tasks, whose input-output definitions are converted into a text-to-text format. Longformer also has a transformer architecture; however, it is modified to process long document texts. Hence, it is a model that is preferred if we are dealing with a large number of lengthy documents. In this paper, we aim to generate medical texts from nursing notes. Since we would like to control the summary size in terms of words and sentences, we use models that can produce abstractive summarizations. In particular, we test our setups on 3 models, namely, BART, T5, and Longformer, because they are popular models in the summarization domain and have similar yet differently built frameworks – BART with BERT and GPT combined encoder-decoder behavior, T5 with various transferred learning task understanding in a unified framework, and Longformer with the ability to handle lengthy documents.

EHR Summarization and Simplification

Due to previous studies on the challenges of using EHR documents [15, 28, 7], there have been conceptual discussions [30] and prototypes [33, 8] to mitigate issues of utilizing medical records. There are frameworks to summarize scientific literature [10, 14] automatically and to improve its factual correctness. For instance, Zhang et al. [33] propose a training strategy that optimizes a neural summarization model with a factual correctness reward through reinforcement learning. There have also been works to improve the understanding of such documents to a wider audience. For example, Devaraj et al. [8] achieve a baseline for this goal by implementing an encoder-decoder Transformer model with a modified loss function of penalizing the decoder aspect when it creates "technical" tokens. Due to the resulting text having less jargon, the generated document is simplified and relatively easier for a wider audience to read.

Other works, including ours, focus on making the clinical texts more readily available for medical professionals to access, record, and understand the patients' conditions at a more efficient rate. MacAvaney et al. [22], for instance, create an ontology-aware clinical abstractive summarization by extending an abstractive summarization model to include encoding of an ontology report section to aid the decoding process. Such a mechanism allows clinicians' time to be saved while informing them of patients' ontology-related information that they may or may not have considered important. The goal of our study is related but different in terms of automating and making it more convenient for clinical professionals to prepare and use Discharge Summary [29], which are essential documents for efficient and continuous patient care.

EXPERIMENT SETUP

Dataset

The dataset we use in this research is MIMIC-III [18]. It is a large database containing de-identified health-related data pertaining to over forty thousand patients who stayed in critical care units in the Beth Israel Deaconess Medical Center between 2001 and 2012. Amongst the tables present in this database, we used the 'noteevents' table, which contains notes for patients. There are 15 types of notes present, including Discharge summary, Echo, ECG, Nursing, and many more. Amongst them, we used the Discharge Summary and Nursing notes. We design the following setups to analyze various possibilities for compiling a dataset that can be effectively used for abstractive EHR summarization:

- Setup 1: For each patient, we gathered all the patient's nursing notes and placed them together under the 'source' column. As for the corresponding 'target,' we set it to be their most recent discharge summary.
- Setup 2: In this setup for each patient, we combine their earliest and latest nursing notes to be the 'source' part of the source-target pair. To create the 'target,' we use Luhn summarizer, which is one of the extractive summarization models that we reviewed in Section 2. We utilize this model on their most recent discharge summary and set the generated texts as the respective 'target.'
- Setup 3: The source setup is the same as setup 2. As for the 'target,' we extract the first three lines of each section within the most recent discharge summary report.
- Setup 4: For each patient, their most recent nursing note is considered as 'source,' and their 'History of Present Illness' section in their most recent discharge summary report is considered as 'target.'
- Setup 5: This setup is similar to setup 4, except that we include the 'History of Present Illness' as well as the 'Discharge Instructions' sections as part of the respective 'target' text.

Setups 1, 2, and 3 include $6,157$ training data points, while setups 4 and 5 contain $6,132$ and $5,981$ training data points, respectively. For testing purposes, there are $1,000$ patient data reports for each setup. The Python code written to generate these setups is available here[2] . The purpose of creating these five setups is to understand the input combination that a model best understands with respect to the expected output, which is a partial discharge summary report.

Setup 1 takes the fewest steps to create the dataset because the source includes all nursing notes for a set of patients, and the target includes the discharge summary report for the same set of patients. These can be taken directly from the initial dataset, i.e., the MIMIC-III dataset, without further processing of the data.

In setups 2 and 3, additional steps are taken to populate both the source and the target text. We sort the nursing notes because we want the earliest and latest nursing notes as our source. This change aims to see if we can achieve a decent summarization with fewer nursing notes. Additionally, we had another modification in the target text where the summary of the discharge summary report is provided instead of the entire report itself. In setup two, the summary of the report is generated using the Luhn summarizer [21], while setup three uses a modified version of the LEAD-3 pipeline, which gets the first three lines of each section in the discharge summary report rather than the first three lines of the entire document. Both of these summarizers are extractive summarization algorithms. The use of these setups is to determine whether a combination of extractive and abstractive pipelines would create better summary reports than just abstractive ones.

Setups 4 and 5 also require sorting of nursing notes in order to use the most recent nursing note as the source text. Instead of extracting certain sentences from the entire document, these setups focus on one (setup 4) or two (setup 5) sections of the discharge summary report. The purpose of these setups is to train models to produce parts of the reports instead of the entire report. While nursing notes have information that is present in the discharge summary report, they do not have all information. Hence, we choose to look into those sections that have overlapping information in both document types. One of such sections is 'History of Present Illness.' Both setups have this section as the target text. This requires extra text processing because we need to parse through the discharge summary report and divide it into sections and their related texts. In setup 5, we also include 'Discharge Instructions' as the second section to generate text. By adding another section, we want to understand if the generated reports are significantly better than having the model focus on producing text for one section.

Metrics

To quantitatively measure our generated summaries for each of the data setup and model combinations, the set of metrics we use is ROUGE, namely ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-L-SUM [20]. ROUGE-1 measures the number of matching unigrams between the generated text and the actual text. The actual text in our experiments would either be the entire discharge summary report (setups 1, 2, and 3) or particular sections of the report (setups 4 and 5). The difference between ROUGE-1 and ROUGE-2 is that ROUGE-2 measures the number of matching bigrams instead of unigrams. Unlike finding fixed n-gram matches, ROUGE-L computes the longest common subsequence (LCS) between the model output and the actual text. In other words, it calculates the number of the longest word sequence that is shared between the two texts. ROUGE-L-SUM also calculates this, but it applies on a summary level. This means that it splits the sentences in the text based on newline characters and takes the union LCS matches between actual (reference) text and every model-generated sentence.

The calculated scores are recall, precision, and F_1. Recall calculates the number of matches found in both texts and divides it by the number of n-grams in the reference text. Precision calculates the matches as well, but it instead divides it by the number of n-grams in the generated text. The F_1

[2]Google Colab Links: part 1, part 2

score uses both the recall and precision values to provide us an insight into how many words a model is able to capture without generating irrelevant words. To calculate F_1, the following equation is used.

Models

To automate the text generation of discharge summary reports, we select three text summarization models that would be trained and tested against the setups described in the previous sub-section. The models that are used for this purpose are BART [19], T5 [26], and Longformer [5]. There are multiple reasons for selecting these models. BART is known as being the state-of-the-art in different domains for producing abstractive summarizations. Since the nursing notes have overlapping but not the entire information required to fill up this report, we want the chosen summarization models to extract certain key information but understand the overall idea described in the nursing notes and phrase its understanding into the report. BART's encoder-decoder behavior emulates the properties of BERT [9] and GPT [25], which are well-known language-based task models. T5 has a unified transferred learning framework which has been trained on huge datasets suitable for various text generation tasks, which makes it a suitable model to train and test on EHR summarization, another language-based task. As for Longformer, the benefit of this model is that it can encode and decode multiple lengthy text documents, which previously mentioned transformer-based models have trouble doing so due to the quadratic increase in scale and complexity caused by their attention mechanisms. For these reasons, we train and test these three models and find out which of them is most suitable for each setup and overall for EHR summarization and Discharge Summary text generation.

Technical Implementation

All the training and testing have been conducted using Python Google Colab notebooks. The models use Hugging Face packages[3]. For BART, we use BartForConditionalGeneration and BartTokenizer classes[4]. We utilize transfer learning by starting our model training using the values present in the "facebook/bart-base" model[5]. For T5, we useT5ForConditionalGeneration and T5ForConditionalGeneration classes[6]. Similar to BART, we fine-tune a pre-trained model, which for T5 is the "t5-base" model[7]. For Longformer, we use the variant known as Longformer-Encoder-Decoder (LED), which is mentioned in the Longformer paper [5]. This variant is suitable for summarization texts as it has the support for performing long document generative sequence-to-sequence tasks. Hence the Hugging Face classes that we use for LED are LEDForConditionalGeneration and LEDTokenizer[8]. As for the pre-trained model, we use "allenai/led-base-16384"[9]. We chose our pre-trained models based on the number of downloads recorded by Hugging Face and on the memory limit present in Google Colab.

Number of Beams	Chosen Setup No. 3			
	rouge1	*rouge2*	*rougeL*	*rougeLsum*
1	0.214	0.175	0.206	0.206
2	0.218	0.178	0.210	0.210
3	0.220	0.180	0.212	0.212
4	0.219	0.179	0.210	0.211
5	0.220	0.180	0.212	0.212
6	0.221	0.180	0.212	0.213
7	0.221	0.180	0.212	0.212
8	0.220	0.180	0.213	0.212
9	0.221	0.181	0.213	0.212
10	**0.221**	**0.181**	**0.213**	**0.213**

Table 1. Beam Search Range

In each of these models, there are a couple of parameters we needed to change in order to maximize our summary generation accuracy. One of such parameters is known as `num_beams`. This parameter represents the number of beams (or possibilities) that the model will use while conducting a beam search, which is a method to reduce the likelihood of missing a highly probable word sequence: the model will keep `num_beams` number of hypotheses at each time step and eventually choose the hypothesis that gives the overall highest word sequence probability. Hence, to determine this number, we trained and tested one of the models, BART, with one of our setups, Setup 3, with beam values 1 through 10. The set of metrics we used to test the resulting trained model is known as Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [20]. It works by comparing the generated summary text against the reference summary provided. In Table 1, we display our experimental results where we show how the change in `num_beams` value affects the ROUGE scores for the BART model, which is trained using Setup 3. Based on the results, the `num_beams` value 10 provides the highest value in the entire set of ROUGE scores. Hence, when we generate our summaries, the parameter value we assign our `num_beams` to is 10.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

In our scripts, we calculate ROUGE using the package provided by Hugging Face which is a wrapper around Google Research's re-implementation of ROUGE[10].

EXPERIMENTAL RESULTS

In Tables 2, 3, 4, 5, and 6, we display ROUGE scores, namely the F_1 scores in the mid percentile range following common practice, for all setup and model combinations. Each of these

[3] Hugging Face Website
[4] Bart-related Hugging Face documentation
[5] Bart-base model card link
[6] T5-related Hugging Face documentation
[7] T5-base model card link
[8] LED-related Hugging Face documentation
[9] LED-base-16384 model card link
[10] Hugging Face ROUGE

values is approximated to their third significant figure. We evaluate the summaries generated by BART pre-trained, BART fine-tuned, T5 pre-trained, T5 fine-tuned, Longformer pre-trained, and Longformer fine-tuned models for each data setup. The pre-trained versions mean that we just load a pre-trained model (for instance, t5-base for T5) and test the model using our testing dataset for a particular data setup. The fine-tuned versions imply that we load a pre-trained model, train it further using our training and validation datasets for the given data setup, and then finally test the newly trained model. The tables have two bold rows that are populated with ROUGE values. The first bold row showcases the highest F_1 scores achieved in the given setup. The second bold row represents the average values of F_1 scores for each metric. This is calculated by taking the average over all the mid-F_1 scores recorded in all the models, i.e., the pre-trained and fine-tuned versions of the BART, T5, and Longformer models.

For Setup 1, Table 2 indicates that T5 fine-tuned model achieves the highest scores in the ROUGE metric set. To understand whether this result is significant, we compare the T5 fine-tuned version's low-percentile F_1 score against the high-percentile F_1 scores achieved by other models. We find that these values are significantly higher than all the other models except for BART's fine-tuned version. Between T5 fine-tuned and BART fine-tuned models, the difference in the former's low-percentile scores and the latter's high-percentile scores is about 0.001, which is not a significant difference. This behavior is also reflected in other setups, namely Setups 2 and 5. Table 3 reflects the F_1 scores achieved by the models when they are trained and tested using Setup 2 data setup, while Table 6 does the same, but with Setup 5 data setup. Like Setup 1, T5's fine-tuned version achieves the highest scores in Setups 2 and 5, which are significant compared to all other models except for BART's fine-tuned version. This means that we can use T5 or BART to achieve comparatively high ROUGE scores for these data setups.

For Setup 3, Table 4 indicates that BART fine-tuned model achieves the highest scores in the ROUGE metric set. This result is significant against all pre-trained versions but not fine-tuned ones. T5 and Longformer fine-tuned models achieve similar results, and the high-percentile F_1 ROUGE scores of these models are higher than the low-percentile F_1 ROUGE scores of BART's fine-tuned version. This implies that we can interchangeably use any of the fine-tuned models to achieve similar ROUGE results.

For Setup 4, Table 5 indicates that BART fine-tuned model also achieves the highest scores in the ROUGE metric set. Similar to Setup 3, this result is significant against all pre-trained versions and mildly significant against the T5 fine-tuned version. However, unlike Setup 3, this model is significant against Longformer for ROUGE-1, ROUGE-L, and ROUGE-L-SUM scores, but not in terms of ROUGE-2 scores. Since the majority of scores are significant, the BART fine-tuned model scores can be considered significantly higher than Longformer's fine-tuned version. Hence, similar to other setups, we can utilize T5 or BART to gain high ROUGE scores whilst generating a section of the discharge summary report using Setup 4.

In addition to testing how well the models have performed in each setup, we also see how much easier it is for models to train and generate text. The way we find this is by comparing the average F_1 values attained in each setup. Setup 2 achieves the lowest while Setup 5 achieves the highest average F_1 score. This implies that text summarization models, in general, have a better chance of generating Discharge Summary texts in data creation Setup 5 than in Setup 2.

EVALUATION

In the results section, we quantitatively look at how each setup and model combination perform. This section goes through some generated text examples and qualitatively evaluates the highest scoring model's outputs.

In Figure 2, we illustrate an instance of a generated summary (on the right side of the table in the figure), which can compared against the actual report (on the left side of the table in the figure). This example was taken from one of the generated test outputs by the highest performing model for setup 1, which is T5. As we can see, this model is able to recognize and identify that the Discharge Summary has a structure with some sections such as Service, Allergies, Past Medical History and more. It was also able to correctly determine the related information for age, service, and allergies. It also recognizes that certain numbers are important. For instance, it kept some lab result display results such as 99.8, HR 95, and BP 180/160. However, there are other instances where the service type or other information can be wrong. To understand how often an information is accurate, we calculate the accuracy of information for a particular section, namely "Service." Amongst the model's generated texts, 508 out of 1000 have "Service" sections described. 282 out of 508 (55%) have the right service type described. This implies that even if the model recognizes this section, it may not always identify the actual type of service conducted on the patient. These behavior traits also appear in other setup and model combinations.

Based on the results, Setup 5 achieves the highest average F_1 score. Hence, Figure 3 takes a closer look at an example taken from one of the generated test outputs by the highest performing model for setup 5, which is also T5. This instance illustrates that the model was able to determine the content of two sections, i.e., "History of Present Illness" and "Discharge Instruction" as intended with this particular data setup. While both sections are generally accurate, the latter section has general instructions that may or may not represent the patient's actual discharge instructions. Nevertheless, this example shows promise with respect to generating sections within the Discharge Summary Report.

CONCLUSIONS AND FUTURE WORKS

This paper investigates the use of various data creation setups and training of text summarization models to automatically generate a Discharge Summary Report, which is a significant Electronic Health Record used by medical professionals. Our experiments show that fine-tuning T5 and BART models significantly create better texts than their pre-trained versions and other models. As for data creation setups, we find that

Model	Setup 1			
	rouge1	*rouge2*	*rougeL*	*rougeLsum*
BART (pre-trained)	0.144	0.076	0.124	0.123
BART (after fine-tuned training)	0.361	0.234	0.329	0.329
T5 (pre-trained)	0.00382	0.000110	0.00340	0.00336
T5 (after fine-tuned training)	**0.383**	**0.238**	**0.349**	**0.349**
Longformer (pre-trained)	0.0560	0.0240	0.0489	0.0487
Longformer (after fine-tuned training)	0.273	0.150	0.227	0.227
Average F_1 Score	**0.203**	**0.120**	**0.180**	**0.180**

Table 2. fMeasure Mid Scores of rouge scores achieved by each model in setup 1

Model	Setup 2			
	rouge1	*rouge2*	*rougeL*	*rougeLsum*
BART (pre-trained)	0.142	0.0767	0.122	0.122
BART (after fine-tuned training)	0.210	0.119	0.172	0.173
T5 (pre-trained)	0.00502	0.000268	0.00443	0.00442
T5 (after fine-tuned training)	**0.223**	**0.125**	**0.184**	**0.184**
Longformer (pre-trained)	0.0549	0.0234	0.0475	0.0358
Longformer (after fine-tuned training)	0.182	0.0802	0.138	0.138
Average F_1 Score	**0.136**	**0.0708**	**0.111**	**0.109**

Table 3. fMeasure Mid Scores of rouge scores achieved by each model in setup 2

Model	Setup 3			
	rouge1	*rouge2*	*rougeL*	*rougeLsum*
BART (pre-trained)	0.142	0.0763	0.122	0.121
BART (after fine-tuned training)	**0.221**	**0.181**	**0.213**	**0.213**
T5 (pre-trained)	0.00500	0.000268	0.00442	0.00441
T5 (after fine-tuned training)	0.213	0.169	0.203	0.203
Longformer (pre-trained)	0.0550	0.0235	0.0476	0.0358
Longformer (after fine-tuned training)	0.221	0.180	0.212	0.212
Average F_1 Score	**0.143**	**0.105**	**0.134**	**0.132**

Table 4. fMeasure Mid Scores of rouge scores achieved by each model in setup 3

Model	Setup 4			
	rouge1	*rouge2*	*rougeL*	*rougeLsum*
BART (pre-trained)	0.264	0.166	0.246	0.246
BART (after fine-tuned training)	**0.360**	**0.224**	**0.336**	**0.336**
T5 (pre-trained)	0.00619	0.000177	0.00569	0.00566
T5 (after fine-tuned training)	0.352	0.218	0.329	0.328
Longformer (pre-trained)	0.108	0.0590	0.103	0.103
Longformer (after fine-tuned training)	0.271	0.114	0.243	0.244
Average F_1 Score	**0.227**	**0.130**	**0.210**	**0.210**

Table 5. fMeasure Mid Scores of rouge scores achieved by each model in setup 4

Actual Discharge Summary Report	Generated Text
Service: MEDICINE **Allergies:** Penicillins / Latex / Sulfa (Sulfonamide Antibiotics) / Shellfish Derived **Major Surgical or Invasive Procedure:** s/p ERCP with metal stent placement [**2907-3-3**]. **History of Present Illness:** Ms. [**Known patient lastname 35261**] is a 77 year old female with pancreatic cancer with liver and lung metastases with recent failure to gemcitabine treatment who presents with new onset jaundice two days ago. She reports feeling fatigued and having a poor a[** Location **]ie ovr the last three days, and noted jaundice on the morning of [**2907-2-28**]. She denies fevers, but does report night sweats that have been going on for weeks. She had one episode of nausea and vomiting after taking a pain pill on the evening of the [**2907-3-1**]. She reports [**Location (un) 685**] colored stool but denies pruritis or dark colored urine. She has chronic abdominal pain related to her cancer, but does not report any change in her abdominal pain. She has never had an episode like this before.In the ED, initial vitals were T 99.8, HR 95, BP 180/106, RR 18, 99% on RA. Her labs were notable for leukocytosis, elevated LFTs with an obstructive pattern. Her [**Location (un) **] u/s showed a common bile duct dilation with obstruction at the level of a large pancreatic mass. She was given Cipro/Flagyl and morphine for pain. ERCP was consulted in the ED and plans to peform ERCP on [**2907-3-3**]. Upon arrival to the [**Hospital Unit Name 2**], she was in no acute distress. **Past Medical History:** 1. Metastatic pancreatic cancer with liver mets, diagnosed [**10-11**]. She failed gemcitabine. She is currently enrolled in hospice but is also interested in considering treatment options. 2. Lupus - no current treatment 3. Hypertension. 4. Hypercholesterolemia. 5. GERD. 6. Hypothyroidism. 7. COPD. 8. History of prior TIAs. …	**Service:** MEDICINE **Allergies:** Sulfa, latex, pcn and shellfish **Major Surgical or Invasive Procedure:** Sphincterotomy **History of Present Illness:** 77 yo female with history of poor a[** Location **]e, 2 days of jaundice. No fevers noted. [**Location (un) **] colored stool. Unchanged chronic abdominal pain. 99.8, HR95, BP 180/106. 99% on RA. WBC of 16.9, tbili 6.2, alk p 461, RUQ U/S with CBD dilation and obstruction at level of pancreatic mass. Pt. admitted to the micu for further management. **Past Medical History:** Metastatic Pancreatic Cancer - [**Last Name (NamePattern1) **] - Hyperlipidemia, hypercholesterolemia - afebrile - Leukocytosis - Pancreas cancer - metastatic pancreatitis - malignant neoplasm - ….

Figure 2. Example of T5's Generated Text using data Setup 1. The actual Discharge Summary Report is on the left side of this figure

Model	Setup 5			
	rouge1	*rouge2*	*rougeL*	*rougeLsum*
BART (pre-trained)	0.266	0.167	0.251	0.250
BART (after fine-tuned training)	0.379	0.236	0.353	0.354
T5 (pre-trained)	0.00590	0.000227	0.00547	0.00545
T5 (after fine-tuned training)	**0.388**	**0.254**	**0.364**	**0.365**
Longformer (pre-trained)	0.106	0.0581	0.101	0.124
Longformer (after fine-tuned training)	0.222	0.105	0.202	0.202
Average F_1 Score	**0.228**	**0.137**	**0.213**	**0.217**

Table 6. fMeasure Mid Scores of rouge scores achieved by each model in setup 5

Actual Discharge Summary Report	Generated Text
Mr. [**Known patient lastname 35644**] is a 28 year old man from [**Country **]. He states he had a fall from a skateboard at 6 am [**2784-10-5**] on his way to work (fishing company employee and some question as to if infection happened at work and was exercabated by fall). … Keep your incisions clean and dry Please take all medication as prescribed If you have any increased redness, drainage, or swelling, or … Please take only as directed and do not drive or operate any machinery while taking this medication. There is a 72 hour (Monday through Friday, 9am to 4pm) response time for prescription refil requests.	Mr. [**Known patient lastname **] is a 28 yo male from [**Country **] who presented to [**Hospital **] Hospital with c/o fall from skateboard at 6 am Friday on his way to work (fishing company employee). … Please shower daily including washing incisions gently with mild soap, no baths or swimming Monitor wounds No driving for 1 month Please call with any questions or concerns [**Telephone/Fax (1) 181**]

Figure 3. Example of T5's Generated Text using data Setup 5. The actual Discharge Summary Sections are on the left side of this figure

utilizing all nursing notes (setup 1) and focusing on particular sections (setups 4 and 5) can be helpful to generate more readable discharge sections. We actualize and show promising results in our work to create custom datasets and train them on text summarizations to automatically generate medical documents. To continue the long-term goal of generating EHR documents, the research community can perhaps collaborate with medical professionals and create summarization-based medical datasets and even better summarization models to generate more readable and accurate medical documents.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Professor Carsten Eickhoff since this project would not exist without his support and expertise. I would also like to thank my mentor, Dr. Seyed Ali Bahrainian, who has been incredibly helpful throughout my thesis journey, by guiding me through various experiment setups and discussing next steps with me at every stage of this project. I would also like to thank my readers for taking the time to read and provide useful feedback to improve this research presentation.

REFERENCES

[1] Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing Management* 39, 1 (2003), 45–65. DOI: http://dx.doi.org/https://doi.org/10.1016/S0306-4573(02)00021-3

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. (2020). DOI: http://dx.doi.org/10.48550/ARXIV.2002.12804

[3] Elena Baralis, Luca Cagliero, Naeem Ahmed Mahoto, and Alessandro Fiori. 2013. GraphSum: Discovering correlations among multiple terms for graph-based summarization. *Inf. Sci.* 249 (2013), 96–109.

[4] Hunar Batra, Akansha Jain, Gargi Bisht, Khushi Srivastava, Meenakshi Bharadwaj, Deepali Bajaj, and Urmil Bharti. 2021. CoVShorts: News Summarization Application Based on Deep NLP Transformers for SARS-CoV-2. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. 1–6. DOI: http://dx.doi.org/10.1109/ICRITO51393.2021.9596520

[5] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. (2020). DOI:http://dx.doi.org/10.48550/ARXIV.2004.05150

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz

Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. (2020). DOI: http://dx.doi.org/10.48550/ARXIV.2005.14165

[7] Kabiru Danladi Garba and Idris Yahaya. 2018. SIGNIFICANCE AND CHALLENGES OF MEDICAL RECORDS: A SYSTEMATIC LITERATURE REVIEW. *Advances in Librarianship* 9 (06 2018), 26–31.

[8] Ashwin Devaraj, Iain J. Marshall, Byron C. Wallace, and Junyi Jessy Li. 2021. Paragraph-level Simplification of Medical Texts. (2021). DOI: http://dx.doi.org/10.48550/ARXIV.2104.05767

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. DOI: http://dx.doi.org/10.18653/v1/N19-1423

[10] N. Elhadad, M.-Y. Kan, J.L. Klavans, and K.R. McKeown. 2005. Customization in a unified framework for summarizing medical literature. *Artificial Intelligence in Medicine* 33, 2 (2005), 179–198. DOI: http://dx.doi.org/https://doi.org/10.1016/j.artmed.2004.07.018 Information Extraction and Summarization from Medical Documents.

[11] Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *J. Artif. Intell. Res.* 22 (2004), 457–479.

[12] Mohamed Fattah. 2014. A hybrid machine learning model for multi-document summarization. *Applied Intelligence* 40 (06 2014). DOI: http://dx.doi.org/10.1007/s10489-013-0490-0

[13] Mohamed Abdel Fattah and Fuji Ren. 2009. GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Comput. Speech Lang.* 23 (2009), 126–144.

[14] Robert Gaizauskas, Patrick Herring, Michael Oakes, Michelline Beaulieu, Peter Willett, Helene Fowkes, and Anna Jonsson. 2001. Intelligent Access to Text: Integrating Information Extraction Technology into Text Browsers. In *Proceedings of the First International Conference on Human Language Technology Research*. https://aclanthology.org/H01-1040

[15] Amanda Hall and Graham Walton. 2004. Information overload within the health care system: a literature review. *Health information and libraries journal* 21 (07 2004), 102–8. DOI: http://dx.doi.org/10.1111/j.1471-1842.2004.00506.x

[16] Sanda Harabagiu and Finley Lacatusu. 2005. Topic Themes for Multi-Document Summarization. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*. Association for Computing Machinery, New York, NY, USA, 202–209. DOI:http://dx.doi.org/10.1145/1076034.1076071

[17] HealthIT.gov. 2021. Challenges to Public Health Reporting Experienced by Non-Federal Acute Care Hospitals, 2019. Online. (16 September 2021). Retrieved May 15, 2022 from https://www.healthit.gov/data/data-briefs/challenges-public-health-reporting-experienced\-non-federal-acute-care-hospitals.

[18] Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9.

[19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. (2019). DOI: http://dx.doi.org/10.48550/ARXIV.1910.13461

[20] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[21] Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM J. Res. Dev.* 2 (1958), 159–165.

[22] Sean MacAvaney, Sajad Sotudeh, Arman Cohan, Nazli Goharian, Ish Talati, and Ross W. Filice. 2019. Ontology-Aware Clinical Abstractive Summarization. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1013–1016. DOI: http://dx.doi.org/10.1145/3331184.3331319

[23] WILLIAM MANN and Sandra Thompson. 1988. Rethorical Structure Theory: Toward a functional theory of text organization. *Text* 8 (01 1988), 243–281. DOI: http://dx.doi.org/10.1515/text.1.1988.8.3.243

[24] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

[25] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[26] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. (2019). DOI: http://dx.doi.org/10.48550/ARXIV.1910.10683

[27] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. (2019). DOI:http://dx.doi.org/10.48550/ARXIV.1905.02450

[28] Tielman T. Van Vleck, Adam Wilcox, Peter D. Stetson, Stephen B. Johnson, and Noémie Elhadad. 2008. Content and structure of clinical problem lists: a corpus analysis. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium* (2008), 753–757.

[29] Carl van Walraven and Ella Rokosh. 1999. What Is Necessary for High-Quality Discharge Summaries? *American Journal of Medical Quality* 14, 4 (1999), 160–169. DOI: http://dx.doi.org/10.1177/106286069901400403 PMID: 10452133.

[30] Tielman Van Vleck, Daniel M. Stein, Peter D. Stetson, and Stephen B. Johnson. 2007. Assessing Data Relevance For Automated Generation Of A Clinical Summary. *AMIA ... Annual Symposium proceedings. AMIA Symposium* (2007), 761–5.

[31] L Yang, X Cai, Y Zhang, and Peng Shi. 2014. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information Sciences* 260 (2014), 37 – 50. DOI: http://dx.doi.org/10.1016/j.ins.2013.11.026

[32] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. (2019). DOI: http://dx.doi.org/10.48550/ARXIV.1906.08237

[33] Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis P. Langlotz. 2019. Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports. (2019). DOI: http://dx.doi.org/10.48550/ARXIV.1911.02541