IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Text Generation in Clinical NLP

Author:
Yurun SONG

Supervisor:
Dr. Kai Sun

Submitted in partial fulfillment of the requirements for the MSc degree in MSc Advanced Computing of Imperial College London

September 2020

Abstract

With the rise of natural language processing technology, text generator is favored by many researchers. This is because text generators have many advantages. For instance, a good text generator is able to improve both accuracy and processing speed of article summaries. Besides, the content generated by a text generator is less biased than manual abstracts. A series of new summarization models have appeared in the general field, but there are few researches in the specialized field. There are many challenges in transferring a general text generator into specific domains. For example, the training for generator lacks domain-specific data sources and the generator needs to understand the relationship between specific terminologies. In this project, a text generator is created specifically for the clinical domain. The generator automatically generates accurate discharge summaries for patients based on their hospitalization information. Through the ontology extraction of 500K medical samples, and pretraining on the weight of the T5-Based model, the model reaches great performance in our task.

Acknowledgments

I would like to give my sincere gratitude to my supervisor Dr. Kai Sun, for offering her guidance and support throughout this project, and also like to thank Jingqing Zhang for his professional advice and always responding to my questions with enthusiasm and patience.

Contents

Chapter 1

Introduction

In recent years, text generator has become an intense research topic in the field of Natural Language Processing (NLP). Given the volume of textual data is growing every single day, it is essential to reduce these verbose textual material to shorter, more focused abstracts. Text generators are able to capture important details of documents and effectively help us navigate to target text. Besides, a large number of tedious unstructured text are very unreadable for the human reader. The ability to translate these materials into legible and precise summaries is the advantage of text generators.

Following the Encoder-Decoder(Vaswani et al., 2017) framework, many models are developed and refined to improve the accuracy and readability of the generated context. These models achieve remarkable results and break through the limitation of text generation in the general domain. However, for some specific domains such as biomedical and clinical area, the general Encoder-Decoder model cannot directly produce reliable context because of the existence of many medical terminologies and narratives. These models have not learned the relationship between domain-specific terms in the pretraining, which can be time-consuming to relearn and difficult to quickly apply in the downstream tasks. Therefore, a generator specifically designed for the clinical domain is necessary.

The key of text generation, to some extent, is the ability to summarize the input documents and reconstruct useful information. Rather than using the extractive summary, we implemented an abstractive summarization model, whose sequences generated are more readable and innovative.

This project intended to implement an encoder-decoder framework for abstracting clinical information. Text-to-Text Transfer Transformer (T5)(Raffel et al., 2019), derived from the standard transformer(Vaswani et al., 2017) model, pre-trains on abundant unlabeled text and fine-tunes in the multi-tasks downstream. Such kind of transfer learning has shown the outstanding performance in both discrimiative and generative tasks. Like this project, transfer learning plays a very important role in the effective improvement of final grades. Pretraining a large amount of unlabeled clinical data, followed by adjustment of clinical tasks, is the main step of our project.

Two primary data sources are used in the project, MIMIC-III[1] and PubMed Central (PMC)[2]. As a broad public and free-accessible Electric Health Record (EHR) dataset, MIMIC-III stores a large number of clinical events, particularly discharge summary records. Learning hospital courses in the summary records and then generating relevant discharge instructions for patients is the downstream task of this project. Meanwhile, as the major source of pretraining data, PMC provides plenty of free full-text biomedical journal and literature for the pretraining. These biomedical articles contain a large number of medical terms that are predicted by the model during pretraining.

The experiments showed that our pretraining built on the T5 weights delivers the best result, which is superior to T5-Base. The results of Rouge1 36.819, Rouge2 15.496 and RougeL 25.087 were achieved in the experiment. In addition, many other experiments were established to test the capability of low resource summarization. It also showed that our pretraining performs better than that of T5 under different kinds of low resources in this task.

1.1 Objectives and Challenges

For patient safety, the hospital courses register all information about the patients' condition in the hospital, such as medical history, diagnosis, prescription and discharge condition.

These records are detailed but cumbersome. Many patients even have more than one hospitalization record or records from different hospitals. When clinicians write a discharge summary, they must carefully check all the patient's hospital history to prevent accidents. In fact, for clinicians, this is an arduous and low-tech task they must do.

An abstract text generator designed for the clinical field can read medical records and automatically generate reliable discharge summaries, which saves medical staff a lot of time. The text generator can also unify the writing format and provide doctors with more readable guidance when patients go to the hospital for follow-up.

For patients, they can get a discharge summary much faster than human summaries by using generation models. Considering that everyone's condition is different, the summary model can customize discharge instructions according to their condition, which is very helpful for their recovery.

For hospitals, standardized discharge summaries can also facilitate the hospital to file and classify for patients' EHRs.

The goal of our project is to design an abstractive text summarization for clinical domain using the real-word EHRs. According to the hospital course of patients in

[1] MIMIC-III: https://mimic.physionet.org/

[2] PMC: https://www.ncbi.nlm.nih.gov/pmc/

the discharge summaries, a reliable model is designed to generate the relevant discharge instructions for patients.

Additionally, low resource or zero-shot is an important benchmark to evaluate the performance of the model.

The challenges of the project are mainly in data processing, ontology extraction and pretraining methods. Three data sources are prepared for pretraining. As unlabeled medical samples, these data sources need to be cleaned up, extracted and concatenated. Ontology extraction is to collect the medical terms in the documents, which is a tedious task. Because extraction tools and medical vocabulary are difficult to implement in the projects and how to map the extracted terms into the pretraining samples is a challenge.Compared to many alternatives, we chooses the QuickUMLS(Luca Soldaini, 2020) as our extraction tool eventually. Furthermore, it is also challenging to reproduce and make a improvement for the pretraining method of T5 model in the experiments, since T5 was the the best.

1.2 Contribution

The contribution of this projects is mainly in the following three aspects:

1. Our experiments proved that pretraining is meaningful for BHC-DIN task. It accelerated the convergence of downstream task's finetuning. Compared with training from the scratch, pretraining showed absolute advantages.

2. The pretraining which uses T5 weights and 500K medical samples achieved great performance in our downstream task. It generated more readable and personalized discharge instructions for the patients in comparison to other pretraining models.

3. In order to learn the relationship between terminologies, it is not a good idea to apply ontology extraction directly for pretraining. Instead, we should do random sampling to extract span text. In this case, the model can learn both common words and medical terms for the pretraining from the scratch.

Chapter 2

Background

The background chapter details several model architectures and data processing techniques. In the first part, we introduce some fundamental, but essential NLP models and how they inspire our later experiments. Meanwhile, a few data processing and evaluation techniques are summarized in the second part.

2.1 Model Architectures

2.1.1 Summarization model

In general, there are two ways to summarize the document: extractive summary and abstractive summary. Extractive approach assembles summaries directly by copying information fragment from the input text, like hierarchical extractor(Nallapati et al., 2016) and pointer-generator(See et al., 2017). This kind of text summarization model is more likely to get a higher rouge score but merely create novel words and phrases in the output sequence. However, abstractive summarization model can capture critical information without changing document consistency and meaning. At the same time, it can carry on the innovative summary according to the text content. Compared with the extractive model, the sequences generated by the abstractive model are more readable and concise though there are many losses and artifacts exist.
In this project, we focus on the abstractive models rather than extractive one.

2.1.2 Transformer

The sequence transaction models are dominated by complex recurrent and convolutional neural network for a long time. However, these two kinds of the model are restricted by reducing the amount of sequential computation and the number of operation required for learning dependencies between distant positions(Hochreiter et al., 2001). Therefore, (Vaswani et al., 2017) proposes a new architecture called Transformer, which follows the Encoder-Decoder structure but through the attention mechanism.

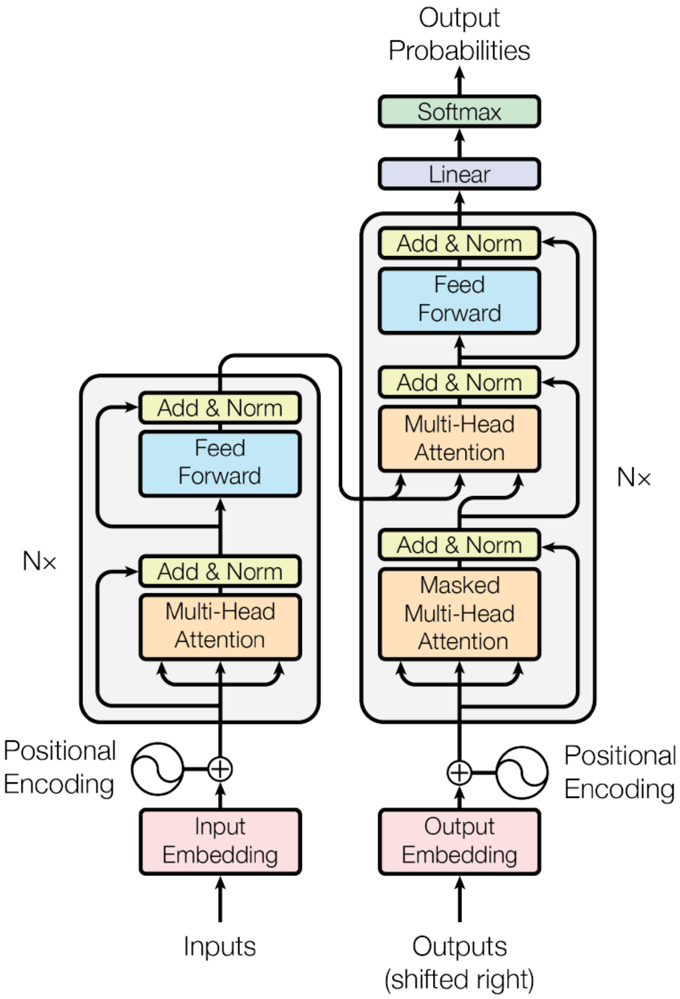


Figure 2.1: The overview of Encoder-Decoder structure of Transformer.(Vaswani et al., 2017)

A standard transformer consists of 6 identical encoder and decoder layers each. From Fig 2.1, the overview of the transformer structure shows the essential components in the encoder and decoder. Briefly, both encoder and decoder have a self-attention mechanism and a position-wise fully-connected feed-forward network for each layer.

Self-attention helps the model to relate different positions of sequence and compute a representation for the sequence. It fully captures the linguistic particularities of each sequence.

Position-wise feed-forward neural network is composed of two linear transformations and a ReLU activation in between to fully explore the relation across the different positions in the sentence.

For the decoder layer, one more sub-layer is required, which is encoder-decoder multi-head attention. This multi-head attention takes the queries from the decoder layer and uses the key and value from the encoder layer, which allows position in the decoder to attend all positions in the encoder input sequence.

Additionally, a residual connection is implemented around each of the two sublayers, followed by layer normalization.

Apart from these modules in the layers, positional encoding plays a significant role to capture the order of the sequence before feeding the input and target embedding into encoder and decoder. Positional encoding needs to consider both the relative and absolute position of the tokens. The sinusoid function is considered to be an appropriate function used in the position encoding.
As a technique to control the range that the model is able to see, masking is widely used in NLP. Transformer accepts two kinds of masks: padding mask and upper triangle sequence mask. Instead of spending many resources on padding position, padding mask forces the model to focus more on the sequence context, especially in the softmax layer.
Upper triangle sequence mask works on the decoder input. In the inference stage, it needs to mask the input tokens and release the output one by one.

In term of training efficiency, Transformer takes priority over other architectures based on recurrent or convolutional layers. Moreover, Transformer can be widely used in many NLP tasks like machine translation, question answering task, and it outperforms almost all the previous ensembles. Many models developed later are based on the attention mechanism or inspired by Transformer structure.

2.1.3 Bidirectional Encoder Representations from Transformers (Bert)

Due to lack of enough labelled training samples in many NLP tasks, direct training on the NLP tasks is not a easy job to achieve the highest scores. Therefore, in order to conceptually understand the context within the limited samples, pretraining on plenty of unlabeled data becomes a widely-used technique. However, how to design a simple but powerful pretraining and its objectives has become a new subject that needs to be studied in depth.
Many language models like ELMo(Peters et al., 2018), Generative Pre-trained Transformer (GPT)(Radford et al., 2018) are unidirectional, in which the token can only use the previous tokens in the attention mechanism. Therefore, (Devlin et al., 2018) proposes a new model architecture called Bert. The bidirectional encoder is able to consider the context from both left and right in all layers. It helps to develop the comprehensive ability of model further and optimizes the token-level tasks in the downstream.

Objectives are essential for pretraining a deep bidirectional Transformer. Bert has two pretraining objectives. The first one is Masked Language Model (MLM), in which the tokens are randomly masked. The model needs to predict the masked token based on the context. It randomly masks 15% tokens of the input sequence, and in those masked tokens, 80% are masked by [MASK] token, 10% are replaced by a random token and rest of them are unchanged. Another objective is Next Sen-

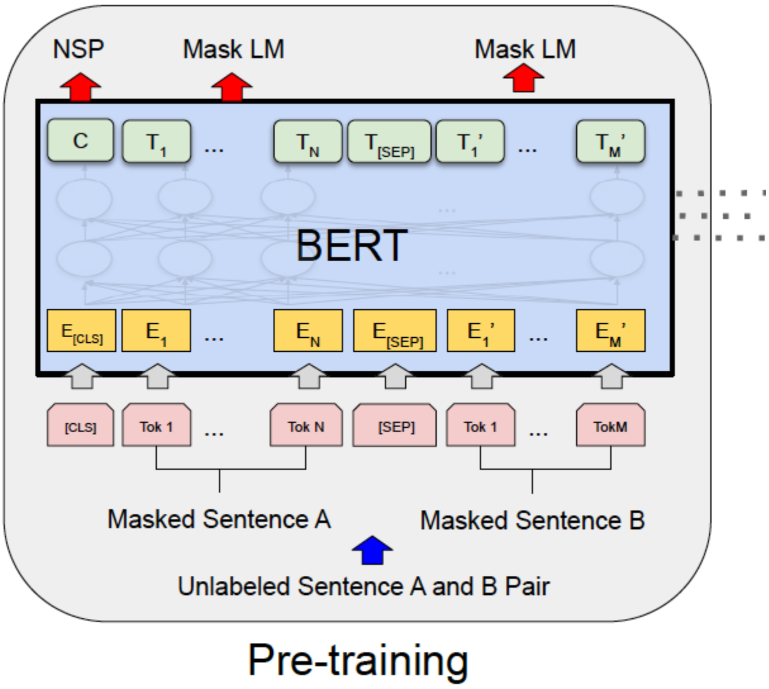


Figure 2.2: The pretraining of Bert model. (Devlin et al., 2018)

tence Prediction (NSP), which tries to understand the relationship between a pair of input sentences. It chooses Sentence A and B from pretraining samples, with 50% of the time B is the next sentence follows A. The model has to predict whether A and B are consecutive.
The pretraining data of Bert comes from the BooksCorpus and English Wikipedia. As an input sequence, two sentences A and B are concatenated, starting with a special token [CLS] and delimited by another token [SEP], see Fig 2.2. Before feeding the tokens into the model, Bert sums the tokens embedding, position embedding and segment embedding up together. Segment embedding indicates which sentence the token belongs to and position embedding randomly initialises the position information of each token. The output contains the decision of NSP in the first token and the result of MLM in the remaining tokens.
Bert uses a bidirectional Transformer encoder as the baseline model. Bert-Base contains 110M parameters, and Bert-Large contains 340M parameters. These two models obtain the best results on the 11 NLP benchmarks. The downstream tasks mainly include sentence pair classification, Question answering, single sentence classification and tagging tasks.

Bert shows that pretraining can significantly contribute to higher scores on many downstream tasks, and demonstrates the effectiveness of token masking technique. Bert becomes a standard reference, and many models subsequently developed follow and improve upon the token masking technique.

2.1.4 BioBert

Due to the word distribution shift from the general domain corpora to biomedical corpora, directly applying the advancements in the NLP model to biomedical text mining often yields unsatisfactory results. Therefore, BioBert is developed to transfer the performance of Bert model from the general domain to the biomedical domain.

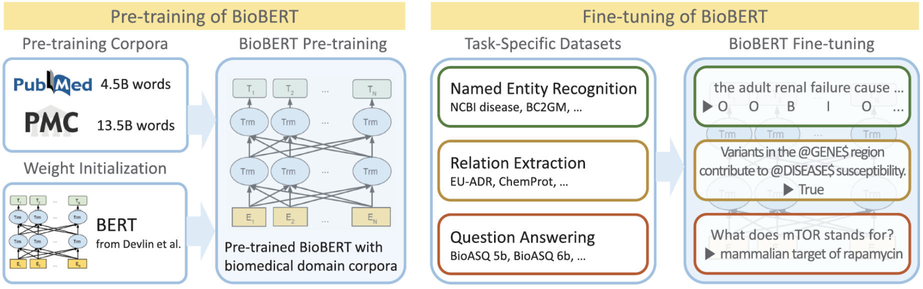


Figure 2.3: The overview of BioBert's pretraining and finetuning. (Lee et al., 2020)

The experiment is split into two parts, pre-training and fine-tuning of BioBert, see Fig 2.3. For the pretraining, apart from the source data from the Bert (pre-trains on English Wikipedia and BooksCorpus), (Lee et al., 2020) adds the Biomedical corpus from the PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC) to Bert Corpus. Considering the computational efficiency, the model initializes the weight from the Bert-Base, and then pre-trains them on Biomedical corpus. At the same time, in order to guarantee the compatibility between BioBert and Bert, the original vocabulary of Bert-Base is used for tokenization. The structure of BioBert and Bert-Base are identical, containing 12 layers, 768 hidden, 12 heads and about 110M parameters.
In the downstream, three biomedical text mining tasks, including Named entity recognition (NER), Relation extraction (RE) and Question answering (QA) were implemented to evaluate the performance of BioBert.

The final results show that BioBert recognizes more biomedical named entities than Bert, and the exact boundaries of these entities can be easily found in NER. For QA task, Biobert is more likely to give correct answers to simple biomedical questions and provides longer named entities in the answer.
It turns out that BioBert outperforms previous models and shows pretraining Bert on biomedical corpora is crucial. These procedures inspire the experiment of this project, and the same idea is applied to the T5 pretraining. Additionally, (Lee et al., 2020) mentions that using case-sensitive vocabulary results in slightly better performances in downstream tasks.

2.1.5 ClinicalBert

Due to the differences in linguistic features between the general text and clinical text, a specialized Bert model is required in the clinical area.
(Alsentzer et al., 2019) wants to find the performance of Bert-Base and BioBert in the clinical notes and discharge summaries. The author generates two kinds of contextual clinical and biomedical embedding, initialized from the Bert-Base and BioBert.

The main idea behind clinicalBert is very close to BioBert, in which pre-trains more speciality corpora based on the general text and then uses transfer learning to test the performance on various tasks.
Regarding the dataset, clinical notes from the MIMIC-III are used for two variants of Bert, clinicalBert and discharge summary Bert. These two models use all notes and Discharge summary type notes respectively.
It finds out there is a negligible impact relative to the various task corpora between these two pretraining embeddings.
The fine-tuning phase consists of several tasks, such as NER, medical natural language inference task and de-identification (de-ID) task. In MIMIC-III, protected health information (PHI) is identified and replaced with sentinel markers. However, in the de-ID task, PHI is masked by synthetic, but realistic PHI. (Alsentzer et al., 2019) believes such an approach could keep the underlying sentence structure unchanged and is particularly useful to context embedding models such as Bert. Eventually, the result proves that clinical BERT is unsuccessful in such de-ID dataset. In addition, (Alsentzer et al., 2019) implements the clinical and Discharge summary data on the BioBert to improve the performance further, see Fig 2.4 The performance shows that Bert achieves the best results in three out of five tasks using the bio and Discharge summary.

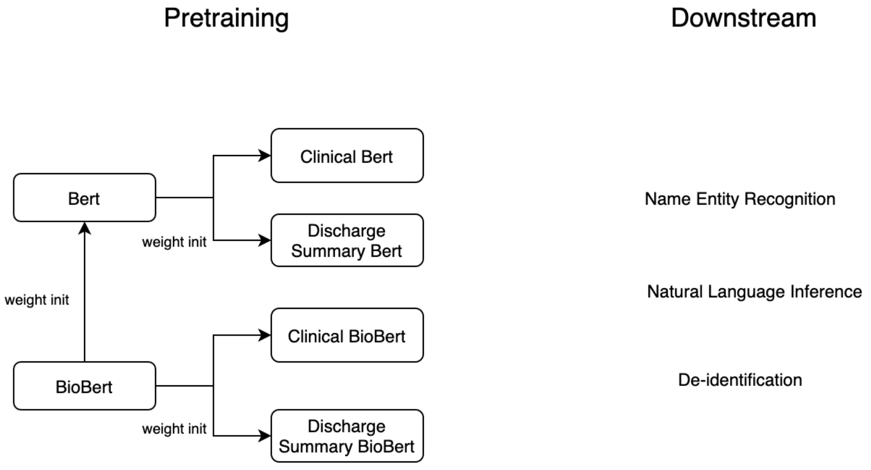


Figure 2.4: The relationship between Bert, BioBert and ClinicalBert.

ClinicalBert demonstrates the promising of domain-specific contextual embeddings for non de-ID clinical task, which indirectly provides evidence of the feasibility of our project.

2.1.6 SpanBert

SpanBert, an extension of Bert, demonstrates a novel pretraining method to present and predict the masked span text with outstanding performance. In summary, (Joshi et al., 2020) makes contributions to pretraining from three aspects based on Bert.

First, (Joshi et al., 2020) chooses to mask the contiguous span of text, instead of randomly masking a single token in Bert (MLM). Both Bert and SpanBert mask 15% of the tokens of the entire documents. The span length is sampled based on the geometric distribution, with the maximum span length of 10, minimum span length of 1 and a distribution probability 0.2, see Fig 2.5. The possibility of choosing span length corresponds to the span length itself. The longer the span length is, the less likely it is to be chosen. On average, 3.8 tokens are masked per sequence. Secondly, an auxiliary objective, Span Boundary Objective (SBO), is proposed, which uses the boundary tokens to predict each masked token in the span. It forces the end of the span to summarize as much of the internal span content as possible using the vector embedding of boundary token and the position embedding of target token. The final loss of masked token is composed of MLM loss and auxiliary loss from the SBO. For example, from the Fig 2.6, the loss for token *football* consists of MLM loss and SBO loss. The MLM loss is the log likelihood of predicting the masked token given the x7 value while SBO loss is computed through the boundary token x4, x9 and position embedding of football token.

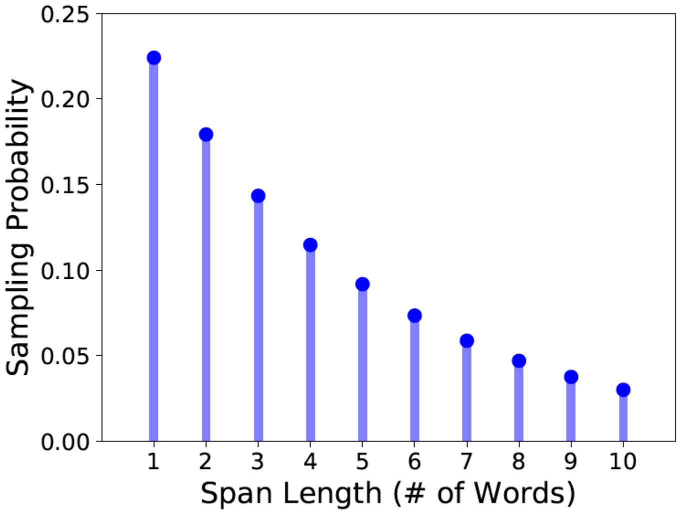


Figure 2.5: The geometric distribution of span length. (Joshi et al., 2020)

Last but not least, the pretraining of Bert uses two sequences as the input, and designs an objective that trains the model to predict whether these two sequences are connected (NSP). However, SpanBert discards this bi-sequences training and adopts the single sequence training, which shows superior performance.

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$

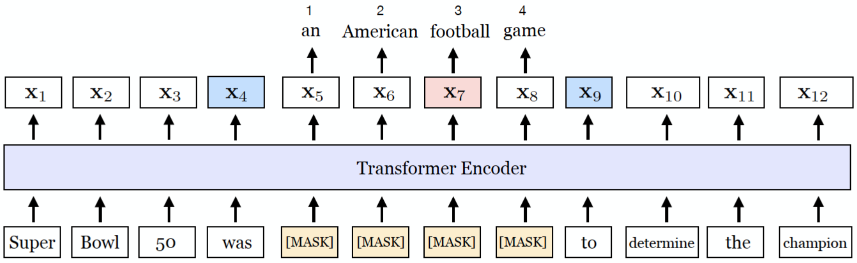


Figure 2.6: An instance of how to compute loss for SpanBert. (Joshi et al., 2020)

Through the above three methods, SpanBert re-implements the same architecture as Bert, and pre-trains on the same corpus but completely outperforms Bert in almost every task. It turns out that span selection can reach substantially higher scores and improve the comprehensive ability of the model. SpanBert and many models afterwards indicate that masking a span of tokens is more effective than single token.

2.1.7 Ontology-Aware Clinical Abstractive Summarization

In order to generate accurate summaries from the clinical reports, the paper (MacAvaney et al., 2019) proposes a pointer generator model augmented with domain-specific ontological information. Precise content is considered to be especially crucial in the medical area and improving the summary completeness through medical ontology becomes their central concept.

The author trains and evaluates the model on the real world radiology report dataset. Each report describes clinical findings as input and an impression summary as target sequence (136 and 37 tokens on average individually).
Beyond that, two domain-specific medical ontologies are employed, UMLS and RadLex. UMLS (Unified Medical Language System) is a general medical ontology while RadLex is an ontology specifically focus on radiology field. A UMLS concept matcher, quick-UMLS is used to extract UMLS concepts from the clinical findings with a threshold of 0.7 and a window size of 3. The RadLex is utilized to evaluate these generated impression summaries.
Pointer Generator network (PG) (See et al., 2017) is a kind of summarization model that follows the prefix language model framework highly dependent on bi-directional LSTMs rather than attention mechanisms and feedforward networks in Transformer. As the Fig 2.7 shown, PG receives the source text from the bidirectional encoder and fills the summary into the undirectional decoder using teacher forcing. Given that the encoder's hidden state is Bi-LSTMs, the source text is fully visible to each other. Also, attention mechanism is applied to the decoder output over the source text, and the attention score is used to compute a context vector. Additionally, a

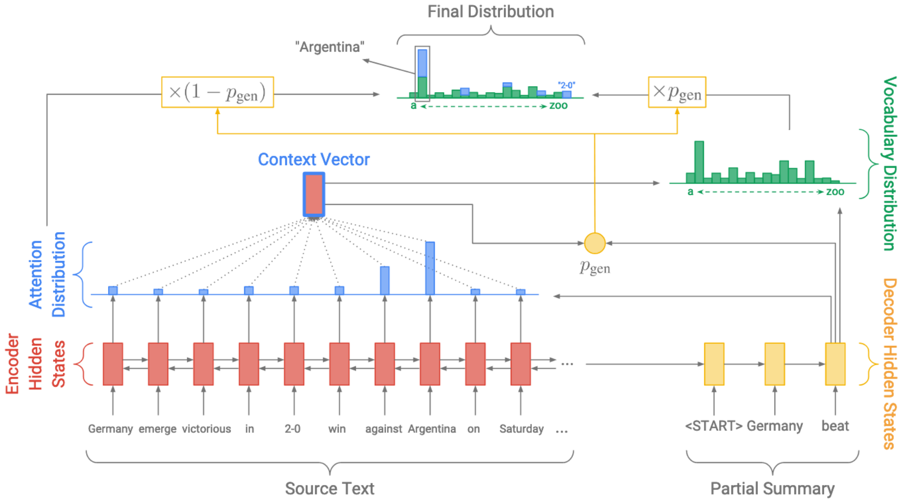


Figure 2.7: The structure of Pointer Generator network. (See et al., 2017)

copy mechanism allows the decoder to directly copy text from the source text where appropriate, which highly improves the rouge scores.
Apart from the standard pointer generator, directly taking the input sequences, the extension ontology-aware pointer generator (ontology PG) bring in a new encoder, which only accepts the sequences with the ontology terms. It introduces a mapping function, only outputs an encoded sequence if it appears in the ontology and skips it otherwise. Meanwhile, an additional context vector is generated to store the domain-ontology information and support the decoding process. Therefore, there are two context vectors work as input in the decoder Bi-LSTMs.

This ontology PG aims the real world radiology reports and shows a significant improvement in comparison with previous work in terms of rouge scores and achieves the best in the radiology dataset.

2.1.8 Masked sequence to sequence pretraining (MASS)

Although Bert achieves outstanding performance by pretraining on the bidirectional encoder, directly applying Bert like language understanding model to language generation is not feasible as generation model handles encoder-decoder learning framework while Bert only adopts a single encoder. In addition, low resources or zero-shot is an essential benchmark in the generation tasks due to lack of enough training samples. Design an efficient pretraining method using plenty of unsupervised data and fine-tuning on the low resources downstream task is crucial for summarization tasks. Hence, (Song et al., 2019) proposes a Masked sequence to sequence pretraining (MASS) method with a novel objective for language generation.

Mass chooses Transformer as the baseline model in the experiment. The encoder takes a sentence with a masked fragment (several consecutive tokens) as input, and its decoder predicts this masked fragment conditioned on the representations from the encoder, see Fig 2.8 This implementation is simple but powerful.

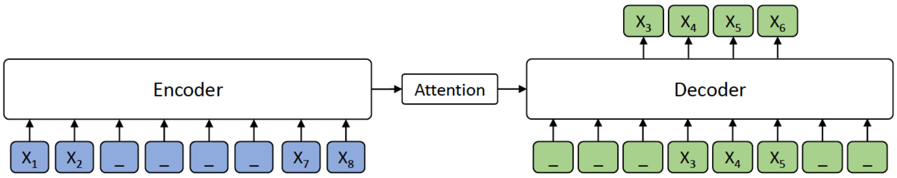


Figure 2.8: The Encoder-Decoder framework of Mass. (Song et al., 2019)

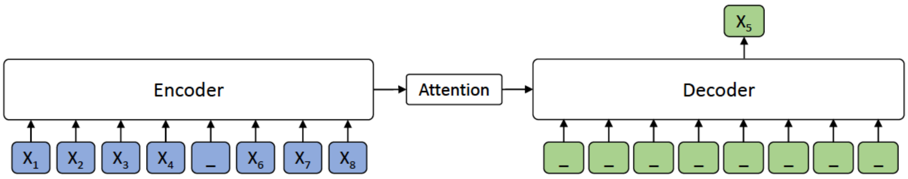


Figure 2.9: The Encoder-Decoder framework of Bert style. (Song et al., 2019)

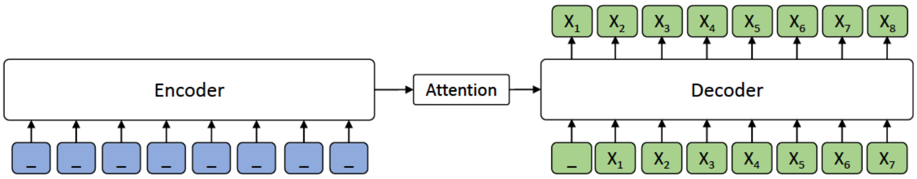


Figure 2.10: The Encoder-Decoder framework of GPT style. (Song et al., 2019)

Mass pretrains on the WMT monolingual corpus and fine-tuning on the three different language generation tasks including Neural Machine Translation, text summarization and conversational response generation.
(Song et al., 2019) makes a comparison between the encoder-decoder framework for Bert-like (Fig 2.9), GPT-like (Fig 2.10) and Mass (Fig 2.8), shown in the [Fig]. The masked fragment in the encoder and decoder is different. For the Bert-like model, the span length (k) is only one, which means only a token is masked once. While the GPT-like model completely masks all the tokens in the encode and the capability of prediction only rely on the decoder. For the Mass, all masked tokens are replaced by the same special symbol ¡M¿ and the span length k is adjustable. The experiments turn out that the MASS achieves the best performance on the majority of downstream tasks when k is nearly 50% of the sentence length m.
In the low-resource scenarios, MASS samples 10K, 100K, 1M pair of sentences from the training data in comparison with the standard Transformer on text summarization task. The experiment result shows that Mass is superior to the Transformer on the both BLUE and Rouge score with different scales of paired samples.
Additionally, the reason why jointly pretrain both encoder and decoder is better than

individually pretraining is significant. First, Mass aims to force the encoder to understand the meaning of those unmasked tokens and drives the decoder to extract useful information from the encoder side at the same time. Secondly, masking the unmasked context in the encoder side forces the decoder to lean more on the encoder representation rather than the information from previous tokens in the prediction. At last, instead of predicting discrete tokens, the consecutive tokens can make the decoder gains better language modelling capability.

On the encoder-decoder framework, Mass achieves better leveraging between encoder and decoder through the joint pretraining. It also proves that masking a span of tokens can achieve significant improvements over other models with or without pretraining methods.

2.1.9 Bart

Rather than using MLM and NSP noising methods in the Bert, Bart implements a denoising autoencoder and an auto-regressive decoder to explore more new transformations. (Lewis et al., 2019) believes there is a significant potential for developing many other noising approaches. Moreover, although the outcome from the Bert is desirable, applying these techniques to encoder-decoder frameworks is worth studying.

The architecture of Bart is, to a large extent, a composite model from the Bert and GPT, a bidirectional encoder and an auto-regressive decoder, as shown in Fig 2.11 However, there is also some slight difference. For example, the ReLU activation is replaced in the GPT decoder. Another cross attention in each decoder layer is implemented over the final layer of the encoder. Hence, Bart encoder is 10% larger than that of Bert. Bart takes the corrupted documents as the input to the encoder, the original documents as the target to the decoder, and relies on the corrupted encoder information and previous tokens to predict the document.
There are two main steps for Bart pretraining. The first step is to apply a random

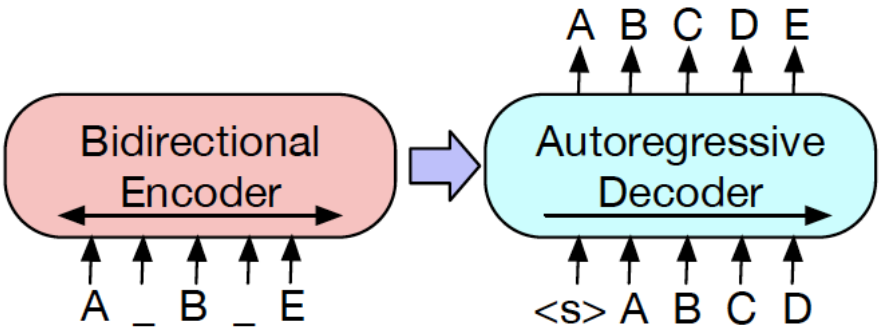


Figure 2.11: The architecture of Bart composed of Bert encoder and GPT decoder.(Lewis et al., 2019)

noising function to corrupt input text. Bart designs five kinds of transformations to corrupt the documents, including Token Masking, Token Deletion, Token infilling, Sentence Permutation and Document Rotation, shown in the Fig 2.12. (Lewis et al., 2019) wants to find the best noising function for the encoder-decoder framework. The second step is to force the model to reconstruct the original text. Using the noise information in the encoder and previous tokens generated in the decoder, the model learns to re-organize the documents by reducing Loss.

It turns out that in the experiment, the best performance can be achieved by shuffling the order of sentences and applying token infilling, where the spans of text are replaced with one single special token

The downstream tasks are generally divided into discriminative and generation tasks, such as Sequence Classification, Token Classification, Sequence Generation Tasks and Machine Translation.

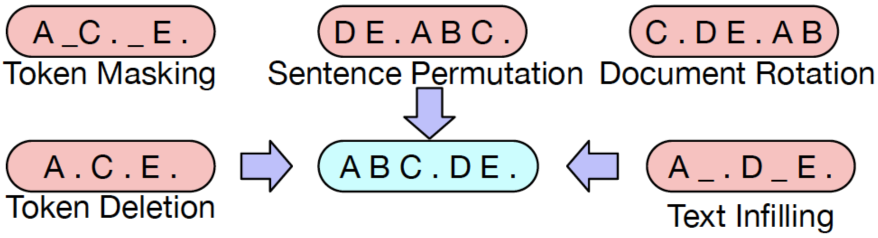


Figure 2.12: The noising transformations of Bart. (Lewis et al., 2019)

Moreover, in order to study the differences between different objectives, (Lewis et al., 2019) compares denoising objectives with many other pretraining objectives, including Language Model, Masked Language Model, Permuted Language Model and Masked Seq-to-Seq in the both discriminative and generation tasks. The result shows that Bart achieves the most consistent and powerful performance, especially using text infilling noising function.

In addition, some other findings rise from the experiments. For instances, the performance of the noising function is extremely affected by the tasks; Token masking is significant; Left-to-right pretraining improves generation.

Bart achieves state-of-the-art on a large number of text generation tasks and shows outstanding performance in the discriminative tasks as well. It explores various possibilities in the encoder-decoder frameworks.

2.1.10 Pegasus

In the abstractive summarisation area, (Zhang, Zhao, Saleh and Liu, 2019) proposes a new model called Pegasus, which uses Transformer combined with self-supervised objective to generate accurate and precious summaries. Pegasus uses the traditional Transformer encoder-decoder architecture, taking at least three sentences from the original document as input text. One of the sentences in the middle is masked as target generation text with a specific mask token and feed into decoder as well. At the same time, some other tokens in the rest sentences are randomly masked as well like Bert. To predict the masked sentences and tokens are the two pre-training objectives called Gap sentence generation (GSG) and Masked Language Model (MLM), see Fig 2.13.

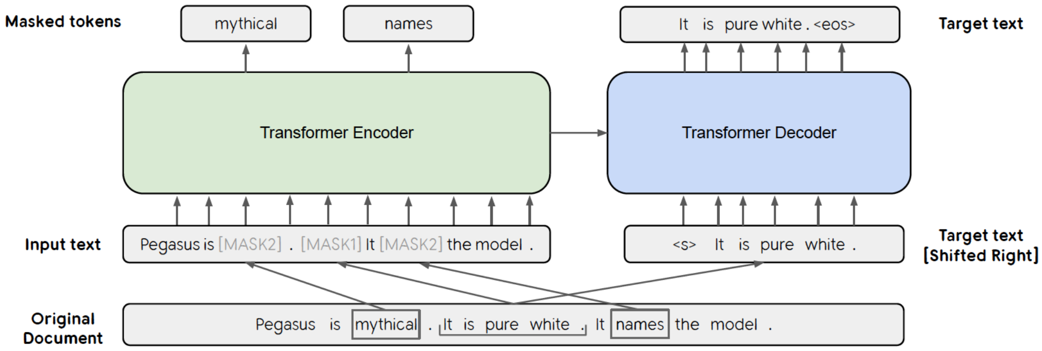


Figure 2.13: The overview of Pegasus encoder and decoder, following the GSG and MLM. (Zhang, Zhao, Saleh and Liu, 2019)

GSG designs a self-supervised objective in the absence of abstractive summary, which allows data to provide the supervision of each other. Therefore, both the preceding sentence and the following sentence have weak supervision to the gap sentences. The author proposes several strategies for selecting gap sentences in the original document without replacement. For example, select m optimal sentences based on their importance, computed by Rouge1 F1 between the selected sentence and the remaining document.
MLM is a standard method that has introduced in the Bert. 15% tokens of the input text are selected. Among these tokens, 80% of them are masked by a mask token, 10% of them are randomly replaced with other tokens, and the rest remains unchanged.
Like the Bert, Pegasus has the two version as well, Pegasus-Base and Pegasus-Large. The size of Pegasus-Large is about twice larger than the Pegasus-Base.
Two large corpora are used in the pretraining, C4 and HugeNews, and 12 summarization task are evaluated in the downstream tasks like XSum, arXiv, PubMed. The paper uses the performance of the downstream tasks to assess the effect of Pegasus models, pre-training objectives and vocabulary. For the two large pre-training corpus, the paper finds that HugeNews are more effectively on the news downstream tasks and C4 is more commonly used.
As far as the GSG selection strategy is concerned, choosing the principal sentences works best for downstream summarization tasks. Gap Sentence Ratio (GSR) is the hyperparameter to provide challenges to pre-training. The GSR is set up to 30% to Pegasus large.

Pegasus large pre-trained on C4 and hugeNew achieves state of the art for all 12 abstraction summarization tasks.
In addition, the paper finds that Pegasus large shows great performance in the zero

and low resource summarization. Pegasus large achieves the best results on 6 out of 12 datasets with only 1000 fine-tuning examples.

2.1.11 Text to Text Transfer Transformer (T5)

Transfer learning is considered as a widely-used technique, which pretrains the model on the abundant unlabedled data and transfers its learned knowledge and ability to the downstream tasks. However, With the emergence of more and more diverse pretraining approaches and model architectures, it becomes difficult to make a comparison and find out the limit of existing methods for using transfer learning. Therefore, instead of proposing a new approach, (Raffel et al., 2019) intends to provide a comprehensive perspective and pushes the limits of current study in the NLP field.

The primary idea is simple, which treats all the NLP problems, like document summarization, question answering and sentiment classification as text-to-text task, taking text as input and generating new text as output.

Model

(Raffel et al., 2019) sets up an encoder-decoder transformer, called Text-to-Text Transfer Transformer (T5), closely follow to the standard Transformer. The encoder consists of two sub-components, a self-attention layer followed by a small feed-forward network. Layer normalization and drop out is applied after the sub-components. The decoder has similar structure with encoder except a new sub-component performing multi-head attention over the output of the encoder. Instead of using the fixed positional embedding like Bert, T5 implements sinusoidal relative positional encoding. It shares the positional embedding parameters across all the layers for efficiency.

The baseline model (T5-base) has 220M parameters, about twice of the Bert-base. 12 blocks for each encoder and decoder, 768 dimension embedding for input and output, 12 heads for attention mechanism, 3072 inner-layer dimensionality for Feed-forward network and 0.1 dropout for every layer. Other Scaled up models are all derived from baseline, including T5-small (60M), T5-Large (770M), T5-3B (3B) and T5-11B (11B).

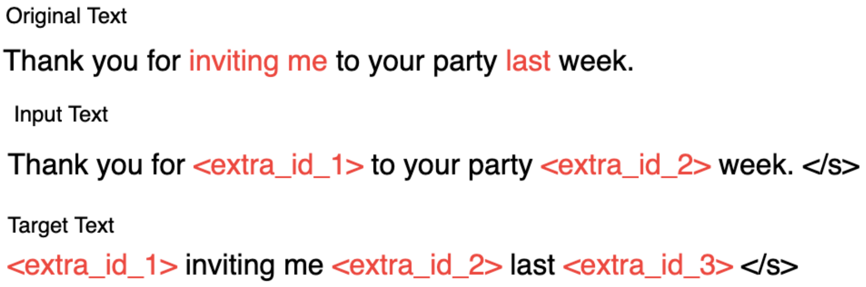
Dataset

A new and massive dataset is released along with the T5, called Colossal Clean Crawled Corpus (C4), where T5 pretrained on. It is derived from a public-available website Common Crawl. There are lots of tasks left for the scrapted HTML file. It removes sequences like script, bad words, placeholder ”lorem ipsum” text, non-english pages and only keep lines that ended with a terminal punctuation mark. The size of the filtered and unlabeled dataset C4 is about 750GB and it develops some other domain-specific subsets, like RealNews-like, WebText-like for dataset comparison.

Input and target

This model provides a consistent training objective both for pretraining and fine-

tuning, which is trained with a maximum likelihood objective (using teacher forcing on all tasks). In order to distinguish which task should perform, it adds a task-specific prefix to the input sequences before feeding into the model.
A denoising objective is used in the T5, in which the model predicts the missing or masked tokens in the input.
First, it randomly samples tokens and drops 15% of corrupted tokens per input sequence. Then, all consecutive corrupted tokens or isolated tokens are replaced by an unique sentinel token. The target sequences corresponds to all of the dropped tokens, which forms a mirror of the processed input sequence with the final sentinel token at the end, see Fig 2.14
Since the prediction of the entire uncorrupted span text requires self attention over long sequences in the decoder. It is more efficient to use a single sentinel token to represent an entire span of text in the prediction.



Original Text

Thank you for inviting me to your party last week.

Input Text

Thank you for <extra_id_1> to your party <extra_id_2> week. </s>

Target Text

<extra_id_1> inviting me <extra_id_2> last <extra_id_3> </s>

Figure 2.14: An instance of pretraining input and target for T5.

Pretraining

It takes 2^{19} steps for pretraining on C4 before fine-tuning on multi-tasks, with sequence length 512 and batch size 128. Besides, there are 2^{35} tokens used in the pretraining, where 2^{35} tokens only covers a fraction of the entire C4 dataset. So T5 never repeats any data during pretraining. Learning scheduler is set as inverse square root, with 10^4 for warm step. Vocabulary 32K with SentencePiece, shared in both input and output.

Downstream

The downstream aims to measure the learning ability of the general language. A various benchmarks, including machine translation, question answering, abstractive summarization, and text classification are tested in the downstream. Specifically, (Raffel et al., 2019) measures the performance of text classification on the GLUE and SuperGLUE benchmark; abstractive summarization on the CNN/Daily Mail dataset; question answering on the SQuAD; and WMT English to French, German and Romanian translation.

Method

As described above, the contribution of this (Raffel et al., 2019)is not to design new

architecture or method, but to broadly analyse the effect of different techniques and reveal the capability of transfer learning in the meantime.
With regard to the comparison of pretraining approaches, it covers topics, including model architectures, unsupervised objectives, corruption strategies, corruption rate, corrupt span length, pretraining datasets and dataset size. Here are some experiment conclusions:

1. Encoder-decoder with denoising objective is superior than many other text to text architectures, such as Language Model, Prefix Language Model and some other architectural variants

2. Bert-style objective has a distinct advantage over other Prefix language modeling and deshuffling objectives. Furthermore, replace corrupted span, a variant of Bert-style objective is preferred among Mass-style, Bert-style (Masking token) and Drop corrupted token method.

3. Keeping the 15% corrupted tokens of the original sequence is slightly better even though corrupted rate has a limited effect on the performance.

4. Setting the average corrupted span length as 3 mildly outperforms than span lengths of 2, 5 and 10, shown in the Fig 2.15.

5. C4 and its variants distinctly improve the performance in comparison to dataset like Wikipida and Toronto Books Corpus.

6. The effect of artificially shrinking the C4 dataset shows that pretraining with full dataset (without repeat) is the best and the performance degrades along with the shrinking of dataset size. Using the large pretraining datasets whenever possible.

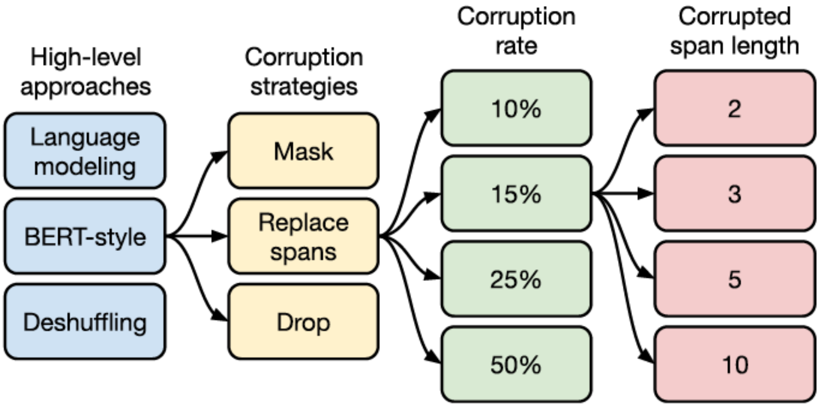


Figure 2.15: The exploration of unsupervised objectives.(Raffel et al., 2019)

With respect to the comparison of downstream approaches, it contains aspects, such as fine-tuning methods, Multi-task learning methods, Combining multi-task learning with fine-tuning and Model Scaling. Here are some experiment findings:

1. Directly fine-tuning on the all of the model's parameters can surpass other approaches, such as adapter layers and gradual unfreezing.

2. In general, multi-task training underperforms pretraining followed by fine-tuning on most tasks. Mixture strategies, like Equal mixing shows dramatically degrade performance and temperature-scaled mixing with temperature equal to 2 performs the best among multi-task training strategies.

3. Standard multi-task learning (examples-proportional mixture strategy with K = 2^{19}) with fine-tuning afterwards is comparable with the performance of baseline (pretrain then fine-tune), which indicates using fine-tuning after multi-task learning is helpful.

4. Model Scaling indicates that different scaling methods varies the trade-offs that are separate from model's performance. The eventual use of the model is crucial when choosing different scaling methods.

Generally speaking, through the systematic study, T5 clearly summarizes the advantages of different approaches and puts all the pieces together for further progressing the NLP field. Some high-level perspective and many significant findings are provided to make sure the NLP research is promising. The results from T5 becomes a reference, which also contributes to this project. Additionally, T5-base and its pretraining is used in the experiment sections afterwards.

2.2 Data Processing

This section focuses on data processing methods, particularly for MIMIC-III dataset. By introducing these methods, the structure of pretraining and finetuning data will be better presented. In addition, some evaluation methods is introduced as well.

2.2.1 Electronic Health Records and MIMIC-III Data Processing

In order to reduce time spent on patients' Electronic Health Records systems and contributing to clinician burnout, (Liu, 2018) proposes a generative model to predict the content of notes based on the patient's medical notes on the past.
The MIMIC-III dataset was the only source the model used in this paper. Therefore, The paper illustrates how to represent both the structured and unstructured data from the clinical notes in detail. It splits the unstructured context data in the 24 hours into 4 classes, demographic data (D), Medications (M), Lab tests (L) and past notes (Np). They construct the task dataset by converting to a string representation, see Fig 2.16 with unique delimiting tokens between classes. In order to examine the effect of each class, these classes are added to structured context data in order, see Fig 2.17.
There are two kinds of model architectures used in the paper. One is the standard transformer encoder-decoder, and another one is the Transformer with memory-compressed attention. It shows that standard Transformer performs well with short

```
<Context> ::= <Hint><NoteType><Demographic><MedList><LabList><NoteList>
<Demographic> ::= <Gender><Age>
<Hint> ::= first-10-tokens-of-note "<H>"
<NoteType> ::= note-type "<T>"
<Gender> ::= "M" | "F" "<G>"
<Age> ::= age-in-years "<A>"
<MedList> ::= <Medication> "<M>" | <Medication> <Delim> <MedList>
<Medication> ::= drug-name
<Delim> ::= "|"
<LabList> ::= <Lab> "<L>" | <Lab> <Delim> <LabList>
<Lab> ::= lab-name "," lab-value "," unit-of-measurement <LabFlag>
<LabFlag> ::= "abnormal" | ""
<NoteList> ::= <Note> | <Note> "<N>" <NoteList>
<Note> ::= raw-note-text
```

Figure 2.16: The structured MIMIC-III discharge summaries. (Liu, 2018)

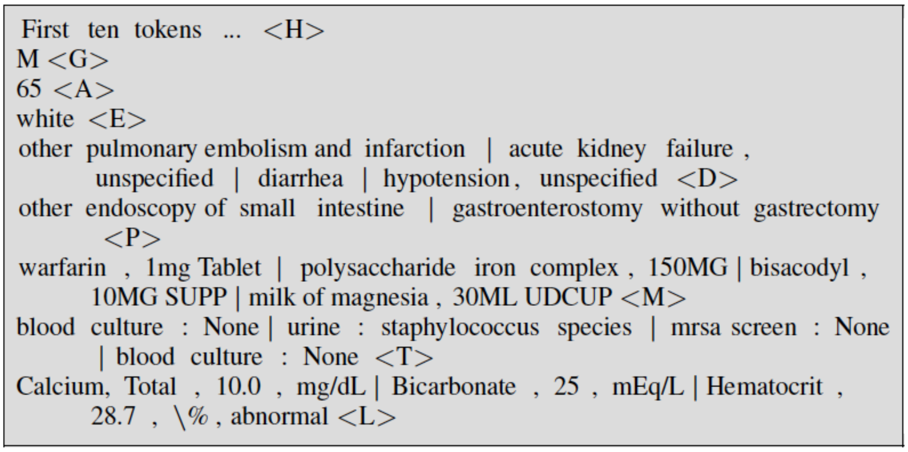
```
Start of note <H>Nursing/other<T>F<G>46<A>Phenylephrine|Heparin<0>
Potassium,4.1,mEq/L,|Nitrogen,4,mg/dL,abnormal<1>Progress note<N>
Another progress note
```

Figure 2.17: The input sequence comes from the structured MIMIC-III. (Liu, 2018)

context while with memory compressed attention is effectively used for large scale text. The metrics evaluating for the model includes perplexity per token(PPL), Rouge1/2 and Sex/Age accuracy. The results show that the Transformer with memory compressed learned much more than standard Transformer and performs better. Apart from these, paper (Liu, 2018) also shows the method of error-detection and note auto-compete. The author corrupted some selected notes by randomly replacing certain drug names to simulate the errors. Then, training the models to compute the probability of each word labelled as an error in Medication class.

2.2.2 Medical Dataset Augmentation

Due to patient confidentiality that prohibits the sharing of personal information, the data access to the medical dataset is restricted. However, many models require a large volume of medical data to find the meaningful and accurate pattern. (Amin-Nejad et al., 2020) proposes a methodology to augment dataset so that the structured medical data can be useful on the downstream clinically tasks.
EHR Discharge summaries in MIMIC-III dataset are used, and they are split into high resource and low resources scenario. High resources dataset contains all discharge summaries while the low resources dataset keep the same validation set and only shrink the size of the training and test datasets. Follows the procedure of generating WikiText-9, MIMIC-Text 9 (low resource dataset) tries not to affect the downstream task as much as possible.
It splits the Discharge summaries into 6 classes, Demographic data (G.A.E), Diagnoise (D), Procedure(P), Medication (M), Microbiology Tests (T) and Laboratory Tests (L). Re-constructing the unstructured summary into a string of input context following 6 classes in order, see Fig 2.18 ScispaCy tokenizer is used due to the ability of processing biomedical texts.
The model architectures used in this paper are vanilla Transformer and GPT-2-small.



First ten tokens ... <H>
M <G>
65 <A>
white <E>
other pulmonary embolism and infarction | acute kidney failure , unspecified | diarrhea | hypotension, unspecified <D>
other endoscopy of small intestine | gastroenterostomy without gastrectomy <P>
warfarin , 1mg Tablet | polysaccharide iron complex , 150MG | bisacodyl , 10MG SUPP | milk of magnesia , 30ML UDCUP <M>
blood culture : None | urine : staphylococcus species | mrsa screen : None | blood culture : None <T>
Calcium, Total , 10.0 , mg/dL | Bicarbonate , 25 , mEq/L | Hematocrit , 28.7 , \% , abnormal <L>

Figure 2.18: Structured MIMIC-III discharge summaries. (Amin-Nejad et al., 2020)

Simply apply MIMIC-Text9 and 98 to model and evaluated by negative perplexity, BLEU, ROUGE-2 and ROUGE-L.
In order to assess the quality of the synthetic data, two downstream tasks: Readmission Prediction and Phenotype Classification are implemented. Both tasks use Bert and BioBert to evaluate the original and synthetic medical context in term of Auc and F1 score.
The results show that while the synthetic data is generally of more reduced quality, it can yield results significantly better than our baselines augment (word replacement EDA) on the readmission prediction task.

2.2.3 Rouge

As a standard metric for the summarization, rouge is widely-used in many researches, especially for generative tasks.
Rouge (Lin, 2004), stands for Recall-Oriented Understudy for Gisting Evaluation, is thought of as the granularity of texts being compared between the system and reference summaries. There are many different kinds of rouge. For example, rouge-1 refers to overlap of unigrams between the system and reference summaries. rouge-2 refers to the overlap of bigrams between the system and reference summaries.
Same with the accuracy, rouge also has its precision, recall and F1-score.
For instance, rouge-n with 50% recall means that 50% of the n-grams in the reference summary are present in the generated summary. rouge-n with 50% precision means that 50% of the n-grams in the generated summary can be found in the reference summary.
Normally, Rouge N is regard as a recall-related measurement, which refers the recall of rouge N.
Apart from Rouge-N, Rouge-L is used to compute the longest common subsequences (LCS) between generated text and reference summaries. It has the precision, recall and F1-score as well. The advantage of using rouge-L is that it doesn't require to conectively matches the subseqences but in-sequence matches. It reflects the word

order in the sentence-level.
In real world, the most popular rouge score are Rouge1, Rouge2 and RougeL, which are also used in our evaluation.

2.2.4 BLEU

BLEU (bilingual evaluation understudy)(Kishore Papineni, 2002), is a metric closely related to rouge score. It is universally used in the evaluation of machine translation and summarization.
Instead of measuring how much words in the references summaries appears in the generated summaries (Rouge), BLEU measures the precision, which how much the words in the generated summaries appeared in the reference summaries.
As the precision-based measurement, BLEU can take multiple reference summaries and counts the percentage of N-grams in the translation text overlap with the human references.
In addition, BLEU also performs brevity penality, which penalizes sentences that are shorter than any of the reference summaries. It can be achieved by comparing it to the length of the reference sentence that it the closest in length.

2.2.5 Perplexity

Perplexity (PPL) is a common metric to access NLP models. It depends on the probability distribution of each word in the sentences to find how accurately the NLP model predicts the words.
The perplexity can be calculated by using cross entropy. Cross entropy indicates the average number of information needed to predict a token, and perplexity is the number of tokens that can be predicted with those information.
In another words, perplexity can be computed by using the exponential function to the average loss, which is estimated by cross entropy. Both of them are related to the loss of each token.
Unlike BLEU and Rouge, language models generally aim to minimize perplexity. In general, the smaller perplexity is, the better the performance of the model is.

Chapter 3

Legal and Ethical Considerations

The main purpose of Ethical Considerations is to protect participants from the harm in any conducted research. Considering there is no participant directly involved into the project, the risk of harm can be minimised.

However, there are three aspects worthy of consideration, in case the potential illegal or ethical issue exists.

First, the research has to protect anonymity and keep confidentiality. Clinical data, to a large extent, refers to privacy of patients, especially EHR. It records the personal information of patients, such as name, age, and medical history. In this project, the data used comes from MIMIC-III and PubMed. For these two sources , the sensitive information has been cleaned up before release. For instances, MIMIC-III uses PHI technique to replace the real name, location, age and time. Therefore, it doesn't need to worry about the privacy disclosure.

Secondly, it has to ensure that information is obtained in a legitimate manner . The application for accessing MIMIC-III dataset is approved before experiment and the articles extracted from the PubMed is free accessible. Other model architectures, like T5 is publicly accessible for research as well.

Last but not least, this project avoids deceptive practices and aims to assist the medical staff to summarise more reliable medical information.

Chapter 4

Methodology & Design

This chapter introduces many theory and approaches, such as data pre-processing, ontology extraction, model architecture and the design for this project. It describes the methodology used in the project in a general form and expresses the flow of the whole project.

4.1 Data Pre-Processing

As the goal of our project is to generate text from the biomedical and clinical fields, the data used should contain sufficient clinical or biomedical information. Data, such as Electric Health Record (EHR), Biomedical literature, Medical journal becomes the right direction to look for.

As the medical data resource for many models, EHR system plays a vital role in the safeguard of patient information as well. Due to the confidentially of patient information, the majority of the EHRs are prohibited public access. Only a few de-identified datasets are available and MIMIC-III [1] is one of the largest clinical databases supporting a diverse variety of medical studies.

Given the limited availability of public clinical data, we used biomedical data as an alternative because they all contain a certain amount of medical vocabulary. One of the most well-known biomedical databases is PubMed[2], which contains massive biomedical articles. In the PubMed, abstract of these articles, as well as part of full-text articles (PubMed Central [3]), are freely accessed.

Both datasets need to be cleaned up and extracted before they can be used for pre-training and downstream task. The detail of these dataset and pre-processing methods are presented in the following sections.

[1] MIMIC-III: https://mimic.physionet.org/

[2] PubMed: https://pubmed.ncbi.nlm.nih.gov/

[3] PMC: https://www.ncbi.nlm.nih.gov/pmc/

4.1.1 MIMIC-III

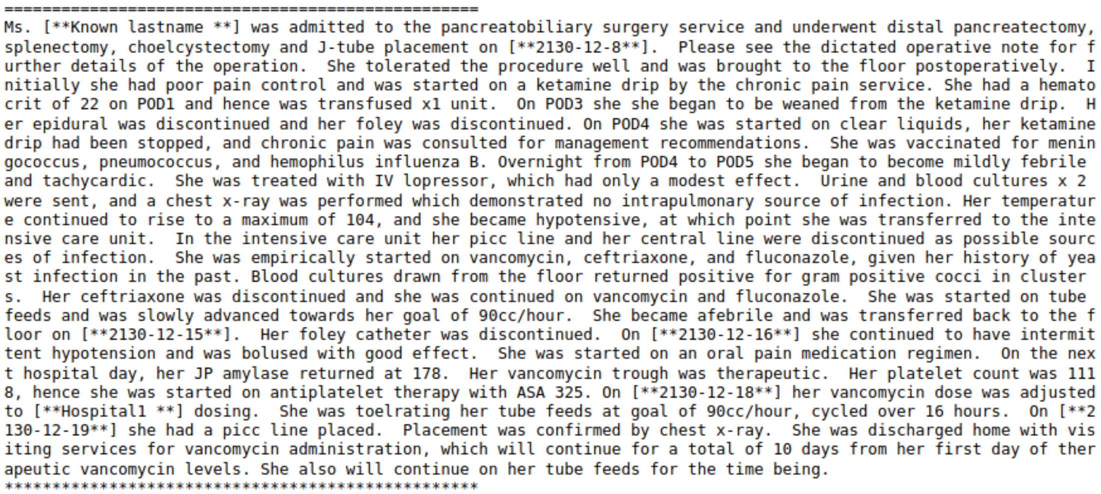
Medical Information Mart for Intensive Care (MIMIC) is an openly available dataset developed by the MIT Lab. It contains patient information, such as demographics, prescriptions, diagnostics, laboratory tests, medications from 60,000 intensive care unit (ICU) admissions.

As the third version of MIMIC, MIMIC-III registers de-identified critical care data from approximately 40,000 patients. It contains 26 relational tables, such as admission, patients, prescriptions, labEvent, NoteEvent and more. NoteEvent is the largest table, which reserves the discharge summaries and other reports, mainly those that are not easily digitized.

As a form of EHR, the discharge summary in NoteEvent table details the ICU patient's condition, and describes their illness history and the usage of medication. It contains sufficient medical information to be the core data of our experiment. There are only 55177 discharge summaries collected over 2 million event notes.

The discharge summary includes many sections, such as allergies, history of present illness, past medical history, social history, medications on admission, discharge condition and so on.

Among these sections, two of them are particularly important, Brief Hospital Course (BHC) and Discharge Instruction (DIN). Brief Hospital Course is considered as a short summary of other sections. It registers patient's condition before admission, ICU health status, medication dosage, discharge conditions and more. While Discharge instruction contains doctor's advice, medication home dose, precautions, etc.



```
=================================================
Ms. [**Known lastname **] was admitted to the pancreatobiliary surgery service and underwent distal pancreatectomy,
splenectomy, choelcystectomy and J-tube placement on [**2130-12-8**].  Please see the dictated operative note for f
urther details of the operation.  She tolerated the procedure well and was brought to the floor postoperatively.  I
nitially she had poor pain control and was started on a ketamine drip by the chronic pain service. She had a hemato
crit of 22 on POD1 and hence was transfused x1 unit.  On POD3 she she began to be weaned from the ketamine drip.  H
er epidural was discontinued and her foley was discontinued. On POD4 she was started on clear liquids, her ketamine
drip had been stopped, and chronic pain was consulted for management recommendations.  She was vaccinated for menin
gococcus, pneumococcus, and hemophilus influenza B. Overnight from POD4 to POD5 she began to become mildly febrile
and tachycardic.  She was treated with IV lopressor, which had only a modest effect.  Urine and blood cultures x 2
were sent, and a chest x-ray was performed which demonstrated no intrapulmonary source of infection. Her temperatur
e continued to rise to a maximum of 104, and she became hypotensive, at which point she was transferred to the inte
nsive care unit.  In the intensive care unit her picc line and her central line were discontinued as possible sourc
es of infection.  She was empirically started on vancomycin, ceftriaxone, and fluconazole, given her history of yea
st infection in the past. Blood cultures drawn from the floor returned positive for gram positive cocci in cluster
s.  Her ceftriaxone was discontinued and she was continued on vancomycin and fluconazole.  She was started on tube
feeds and was slowly advanced towards her goal of 90cc/hour.  She became afebrile and was transferred back to the f
loor on [**2130-12-15**].  Her foley catheter was discontinued.  On [**2130-12-16**] she continued to have intermit
tent hypotension and was bolused with good effect.  She was started on an oral pain medication regimen.  On the nex
t hospital day, her JP amylase returned at 178.  Her vancomycin trough was therapeutic.  Her platelet count was 111
8, hence she was started on antiplatelet therapy with ASA 325. On [**2130-12-18**] her vancomycin dose was adjusted
to [**Hospital1 **] dosing.  She was toelrating her tube feeds at goal of 90cc/hour, cycled over 16 hours.  On [**2
130-12-19**] she had a picc line placed.  Placement was confirmed by chest x-ray.  She was discharged home with vis
iting services for vancomycin administration, which will continue for a total of 10 days from her first day of ther
apeutic vancomycin levels. She also will continue on her tube feeds for the time being.
*************************************************
```

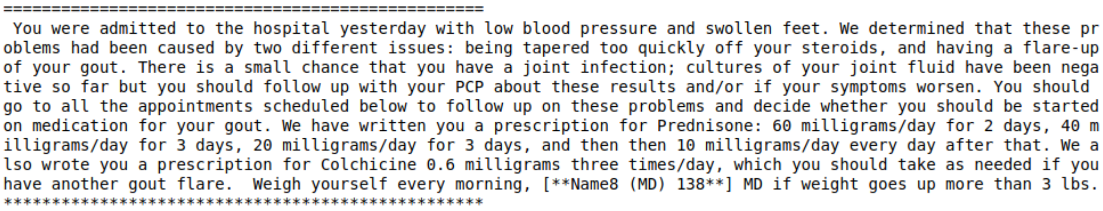
Figure 4.1: An instance of raw Brief Hospital Course.

In the experiment, Brief Hospital Course was taken as the input text and Discharge Instruction was considered to be the target sequence in the downstream. The model needs to derive information from short hospital courses and automatically generate reliable discharge guidance for patients.

However, not all the discharge summaries include a Brief Hospital Course and a Discharge Instruction. We retain only those summaries that contain both Brief Hospital

Course and Discharge Instructions as valid samples. In addition, many discharge instructions contain nothing meaningful (e.g. None or Nan) or are too short (less than 50 characters). These instructions are invalid and will be discarded in the pre-processing. Eventually, nearly 25,000 samples are kept as qualified downstream samples.

The content of other sections in the discharge summaries was used as pretraining materials since there is not overlap between other sections and BHC, DIN.



```
==================================================
 You were admitted to the hospital yesterday with low blood pressure and swollen feet. We determined that these pr
oblems had been caused by two different issues: being tapered too quickly off your steroids, and having a flare-up
of your gout. There is a small chance that you have a joint infection; cultures of your joint fluid have been nega
tive so far but you should follow up with your PCP about these results and/or if your symptoms worsen. You should
go to all the appointments scheduled below to follow up on these problems and decide whether you should be started
on medication for your gout. We have written you a prescription for Prednisone: 60 milligrams/day for 2 days, 40 m
illigrams/day for 3 days, 20 milligrams/day for 3 days, and then then 10 milligrams/day every day after that. We a
lso wrote you a prescription for Colchicine 0.6 milligrams three times/day, which you should take as needed if you
have another gout flare.  Weigh yourself every morning, [**Name8 (MD) 138**] MD if weight goes up more than 3 lbs.
**************************************************
```

Figure 4.2: An instance of raw Discharge Instructions.

The raw discharge summaries contain plenty of PHI markers, see Fig 4.1 and 4.2, which could impact the model to learn sentence structure. Therefore, we replaced these markers with general categories, such as alter the name as ”XXX”, Hospital as ”location”.

Beyond that, there are also many separators and special symbols that are difficult to identify in other sections, and we removed those consecutive separators. (e.g. ####, ====)

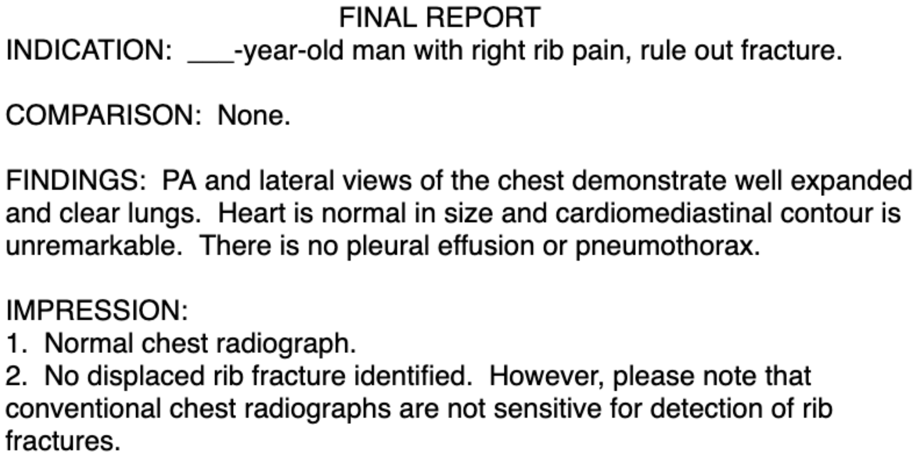
4.1.2 MIMIC-CXR

MIMIC-CXR (Chest X-ray) is a derivative from MIMIC, which contains more than 300,000 de-identification chest radiographs with free-text radiology reports. Slightly different with MIMIC-III, CXR has already removed all PHI markers and replace them with ”___” symbol, see Fig 4.3. The dataset intends to support a wide range of study in radiographs, including computer vision, natural language processing, and decision support.

Radiological reports, rather than chest radiographs, are what we are interested in. Majority of the patients in the CXR are also registered in the MIMIC-III. Some discharge summaries in MIMIC-III contains information in the CXR report, so CXR reports can also be regarded as a detailed subset of MIMIC-III.

There are about 227,835 reports in the CXR dataset in total. Compared to discharge summaries in MIMIC-III, CXR reports are relatively short. The impression and findings, see Fig 4.3, in the reports are the valuable information, which can be used as pretraining data.

Findings describes the condition of the patients’ chest and other organs based on the radiographs while impression determine if the X-ray is normal and if there are other issue exists. The relationship between findings and impression is similar to



FINAL REPORT
INDICATION: ___-year-old man with right rib pain, rule out fracture.

COMPARISON: None.

FINDINGS: PA and lateral views of the chest demonstrate well expanded and clear lungs. Heart is normal in size and cardiomediastinal contour is unremarkable. There is no pleural effusion or pneumothorax.

IMPRESSION:
1. Normal chest radiograph.
2. No displaced rib fracture identified. However, please note that conventional chest radiographs are not sensitive for detection of rib fractures.

Figure 4.3: A sample of radiological report.

Brief hospital course and Discharge instructions. For instance, (Zhang, Merck, Tsai, Manning and Langlotz, 2019) utilises the findings and impression to training a summarization model.

Instead of working as a downstream task, MIMIC-CXR is used as a data source for our pretraining, because clinical medical data is much less than biomedical information and we would like to keep the balance between these two fields. Both MIMIC-III and MIMIC-CXR are real world data sources rather than academic and both findings and impression can be input into as the pretraining model as unlabeled text.

As the same with discharge summaries, not all the reports contain a finding and a impression. However, as pretraining data, all reports can be utilized regardless of whether there is a finding and impression or not.

4.1.3 PubMed Central

PubMed Central (PMC) is a free full-text archive of biomedical and life sciences journal literature. More than 5 million full-text records can be found in the PMC. Majority articles in PMC can be queried on the PubMed. The unique PMC id is associated to the PubMed id. PMC has contributed to many literature and models such as BioBert (Lee et al., 2020).

The PMC Open Access (OA) subset is a part of the total collection of articles in PMC, which contains about 2.8M full-text articles. The articles in the OA subset are made available under legal license and free to use. There are two collections in the OA bulk package for different purposes, commercial use (1.7M articles) and non-commercial use collection (1.1M articles).

Non-commercial use collection is the appropriate choice for our project. There are two formats of the article in the collection, XML and TXT. Rather than TXT format, I choose the XML format for two reasons:

1. XML has a clear structure. Each paragraph is clearly distinguished from the

section to which it belongs. While TXT simply puts everything together.

2. PubMed Parser (Radford et al., 2020) was used to retrieve the body of the articles and it takes XML format only. Other content, such as abstractions, appendices, and tables, is automatically discarded through the parser.

Non-commercial use collection contains more than 1M literature and is roughly about 65GB in XML format. It is further divided into four subsets based on the starting letter of the journal title abbreviation, A-B (182K articles), C-H (251K articles), I-N (452K articles) and O-Z (205K articles).

These subsets contain all retrievable articles, and then the body sections and paragraphs of the article were extracted and linked together using the PubMed parser.

In addition, the PubMed parser can also extract abstracts of articles simultaneously. However, in order to avoid the model learning any summarized content, PubMed abstracts are not included in the pretraining.

As for text cleaning, the articles extracted by parser are highly readable, so we don't do much cleaning to the original text. The original text will be used for pretraining of the model.

4.2 Ontology Extraction

As a common technique, ontology extraction is universally used in many study, such as Ontology-Aware model MacAvaney et al. (2019). Like the domain-specific NER, ontology extraction relies on a standard vocabulary system to collect all possible specialized terminologies and phrases in the document. However, these extracted ontologies are difficult to be accurately expressed through general embedding. Because general embedding is not trained in professional knowledge and it is not possible to understand the relationship between terms directly through the finetuning. Therefore, the purpose of extracting specialized words is to obtain an ontology embedding.

In this project, MIMIC and PMC articles contain a lot of medical terminology. Nevertheless, there is a slightly difference between these two datasets. PMC is a collection of biomedicine, and the content is derived from journals and literatures. Compared to the real-world discharge summaries of MIMIC-III, PMC is more academic. The sentence structure in the MIMIC-III is more realistic, including a large number of abbreviations and acronym, sometimes even not a completed sentence structure.

Therefore, although directly pretraining of more specialized data can improve performance at last, it is not the optimal choice for learning ontology embedding. The model also learns the relationship between common terms simultaneously, which is not computational efficiency. However, if we extract similar medical terms from both MIMIC and PMC, and force the model to learn these terms, we will effectively achieve more accurate ontology embedding.

Logically, if the ontology extraction is not employed on the pretraining data, the model can still learn the relationship between ontologies at last, but with much longer pretraining steps. While learning the relationship between specialized terms,

the model without ontology extraction also needs to understand the relationship between common terms, which may reduce the computational efficiency.
However, if we extract medical terms from MIMIC and PMC and force the model to learn only those terms, we will achieve accurate ontology embedding more quickly. In addition, many model architectures, such as T5, SpanBert and Mass, use random text spans for pre-training. It would be more effective to learn ontology embedding by replacing the random text span with a targeted text span, which containing medical terminology.

Again, the reasons for applying ontology extraction can be summarized as the following two points:

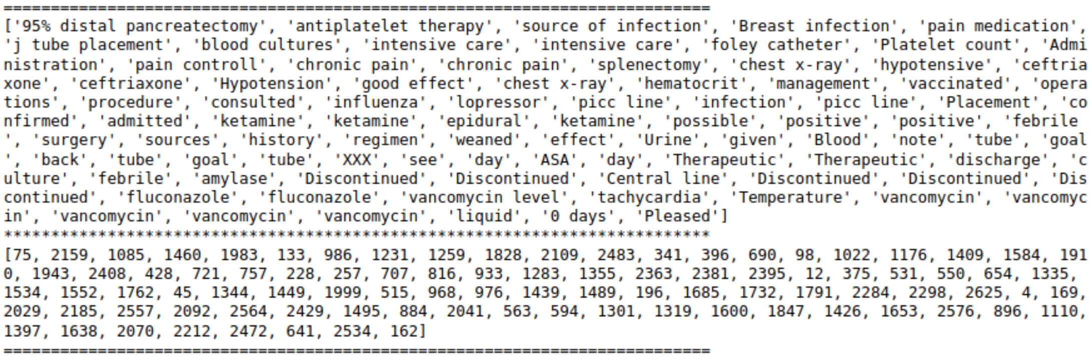
1. Learning ontology embedding more effectively with less data.
2. The random sampled span of the token in some model architectures can be replaced by the ontology span of the token.

4.2.1 Unified Medical Language System (UMLS)

UMLS (Unified Medical Language System)[4] is a giant medical term system developed by U.S. national library of medicine for more than 20 years, which covers clinical pharmacy, biology, medicine and many medical related subjects. This unprecedented medical system contains more than 2 million medical concepts and 500 million medical words, and still growing.
As the largest knowledge source of UMLS system, Meta-thesaurus is responsible for vocabulary control. At the same time, Meta-thesaurus can be seen as a conceptual terminology repository storing terms extracted from many different dictionaries or tables in the biomedical domain, such as ICD-10, MeSH, SNOMED CT, etc. These dictionaries have a large amount of ontology overlap. The same term may appear in different vocabularies, which need to be filtered through the extraction tool later. The Meta-thesaurus is about 40GB in total.
In addition, In order to reduces the complexity of the Metathesaurus, UMLS groups concepts according to the semantic types that have been assigned to them. Currently, there are in total 15 semantic groups and 127 semantic types. Each semantic type can be mapped to a semantics group. For instance, Body Substance (T031) is a type of Anatomy (ANAT) semantic group. Clinical drug (T200) is a type of Chemicals (CHEM) semantic group.
Ontology extraction is a time-consuming task, especially for matching such a large system. Therefore, we need a fast and accurate extraction tool.
As a public extraction tool designed for the UMLS system, QuickUMLS (Luca Soldaini, 2020) uses unsupervised approach and approximate string matching for medical concept extraction. Given a sequence, QuickUMLS extract spans of text and approximately matches the string sets in the UMLS, returning the relevant medical concepts and scores.
Compared to other extraction tools such as CTake and MetaMap, (Luca Soldaini,

[4] UMLS: https://www.nlm.nih.gov/research/umls/index.html

2020) shows that QuickUMLS has an absolute superiority in terms of speed and provides reliable accuracy in the meantime.
As the Fig 4.4 shown, a extraction result of BHC sample (Fig 4.1), QuickUMLS finds the terminology that exist in the documents and the index where the terms locate.
Due to the approximate matching method, there are some errors in the extracted ontology. Strictly speaking, terms, such as ”day”, ”see”, ”goal”, ”note”, are not medical terms. However, I think it is better to extract as many words as possible rather than to extract a small number of accurate words, because a small amount of error will not have a great impact on the final result in the experiment. Moreover, if the number of extracted ontology exceeds the demand for the next phase, which is quite common, we can make random selection to reduce the impact.



```
=========================================================
['95% distal pancreatectomy', 'antiplatelet therapy', 'source of infection', 'Breast infection', 'pain medication',
'j tube placement', 'blood cultures', 'intensive care', 'intensive care', 'foley catheter', 'Platelet count', 'Admi
nistration', 'pain controll', 'chronic pain', 'chronic pain', 'splenectomy', 'chest x-ray', 'hypotensive', 'ceftria
xone', 'ceftriaxone', 'Hypotension', 'good effect', 'chest x-ray', 'hematocrit', 'management', 'vaccinated', 'opera
tions', 'procedure', 'consulted', 'influenza', 'lopressor', 'picc line', 'infection', 'picc line', 'Placement', 'co
nfirmed', 'admitted', 'ketamine', 'ketamine', 'epidural', 'ketamine', 'possible', 'positive', 'positive', 'febrile
', 'surgery', 'sources', 'history', 'regimen', 'weaned', 'effect', 'Urine', 'given', 'Blood', 'note', 'tube', 'goal
', 'back', 'tube', 'goal', 'tube', 'XXX', 'see', 'day', 'ASA', 'day', 'Therapeutic', 'Therapeutic', 'discharge', 'c
ulture', 'febrile', 'amylase', 'Discontinued', 'Discontinued', 'Central line', 'Discontinued', 'Discontinued', 'Dis
continued', 'fluconazole', 'fluconazole', 'vancomycin level', 'tachycardia', 'Temperature', 'vancomycin', 'vancomyc
in', 'vancomycin', 'vancomycin', 'vancomycin', 'liquid', '0 days', 'Pleased']
*********************************************************
[75, 2159, 1085, 1460, 1983, 133, 986, 1231, 1259, 1828, 2109, 2483, 341, 396, 690, 98, 1022, 1176, 1409, 1584, 191
0, 1943, 2408, 428, 721, 757, 228, 257, 707, 816, 933, 1283, 1355, 2363, 2381, 2395, 12, 375, 531, 550, 654, 1335,
1534, 1552, 1762, 45, 1344, 1449, 1999, 515, 968, 976, 1439, 1489, 196, 1685, 1732, 1791, 2284, 2298, 2625, 4, 169,
2029, 2185, 2557, 2092, 2564, 2429, 1495, 884, 2041, 563, 594, 1301, 1319, 1600, 1847, 1426, 1653, 2576, 896, 1110,
1397, 1638, 2070, 2212, 2472, 641, 2534, 162]
=========================================================
```

Figure 4.4: A sample of ontology extraction.

4.3 Model Architectures

The model architecture used in this project is T5-Base. According to the description of (Raffel et al., 2019), T5-Base has the same structure with the standard Transformer.
There are a few reasons why I use T5-Base as my experiment model.

1. T5 model achieves the state of the art in many tasks and can be used as a reliable reference.

2. Ontology extraction requires models that support span text processing, which applies to T5. Compared to Mass, T5 achieves better performance.

3. Certainly, there are some larger models such as T5-Large and T5-3B. However, due to the limited computing power, T5-base (220M parameters) is the proper model for this project.

A real example show in the Fig 4.5. The medical terms (red) in the input text are replaced by unique sentinel tokens. However, for the target text, only these medical terms is unmasked. Other tokens are masked by the sentinel tokens.
The sentinel tokens in the input text are the same with that in the target sequence.

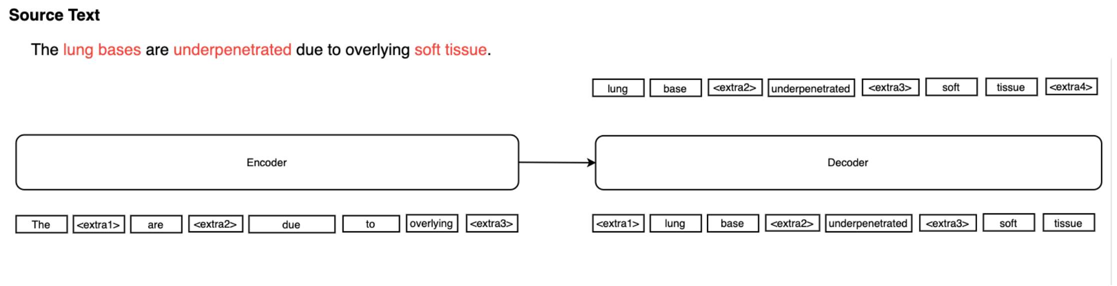


Figure 4.5: An real example for our model architecture.

This is because by learning from the encoder, the sentinel token can relate to all other tokens and extract their information. If the same sentinel token is used in the decoder, which indirectly input these compressed information from all other tokens in the encoder to predict target tokens.

If the target sequence is not ended with sentinel token, it means the end of input sequence is sampled as a span text. Hence, a individual sentinel token is required to indicate the end of target sequence.

Encoder extracts all the background information (Non-medical terms), which decoder uses to predict the medical terms. Therefore, the model learns these medical terminology.

The unique sentinel token is used instead of unified masked symbol [M], because it lets the model know that each sentinel represents different information and multiple tokens in the same sequence.

For the downstream task, there is no sentinel token to mask the input and target. As usual, the entire source text are placed through the encoder and the decoder learns and predicts the target sequence using teacher forcing.(Williams and Zipser, 1989)

4.4 Data Processing Flowchart

In this section, the pretraining data processing flow is shown in the following diagram 4.6:

I'll briefly explain the whole process and key points.

1. Data needs to be cleaned up before pretraining. Cleanup measures include replacing the PHI labels, removing nonsensical symbols, and formatting sentences. At the same time, we need to concatenate the training data to reach max sequence length 512 as much as possible, in which the model can receive more training data within the same epochs. The 512 is the default input size of T5 encoder.

2. Extract the medical ontology with QuickUMLS and return a set of sorted indexes for each sample. The extraction process for 500K pretraining samples is time-consuming. These indexes represent the starting point for each ontology terms in the document. As the example illurstrates in the Figure 4.4, 4 and 19

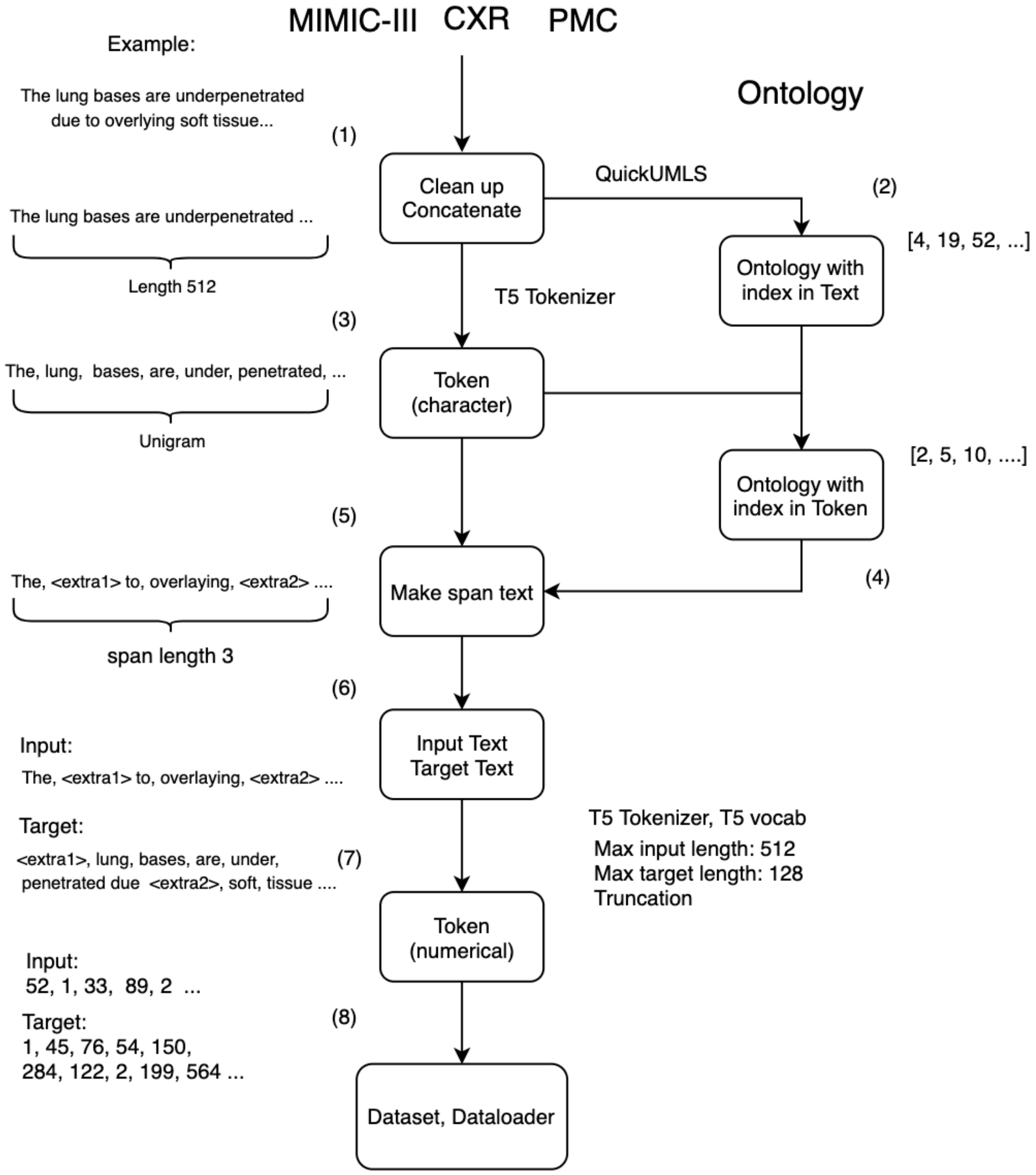


Figure 4.6: Flowchart of pretraining data processing.

are the index number of substring ”lung bases” and ”underpenertrated”, which are our medical terminologies.

3. On the other hand, the cleaned data is tokenized into unigram format by the T5 tokenizer. T5 tokenizer uses sentencepiece to encode sequence with a vocabulary size of 32,000. For example, the ”underpenertrated” does not exist in the vocabulary, but it can be composed by token ”under” and ”penertrated”.

4. Convert the indexes of the original text to the indexes of tokens. The new index tells us which token the ontology is located on. For instance, according

to the ”4”and ”19” in step2, the ”2” and ”5” can be mapped as the token index of ”lung” and ”under”.

5. With the tokens and ontology indexes, we can mask ontology span text with average length 3. Considering the possible overlap of span text, I create a ontology mask for input text, like padding mask, to indicate whether the token in the span text or not. Then, the inverse of the ontology mask can be used for target text. From the example shown in the Figure 4.4, token ”lung” and other two connected tokens are masked as ”F”, token ”under” and other two connected are masked as ”F” and so on. The consecutive False token are replaced by a single sentinel token ”extra1”.

6. Replacing all the consecutive span text with a sentinel token, then, the input and target are formed. The input and target text are nearly complementary, see example in the Figure 4.4.

7. T5 tokenizer and T5 vocabulary are used to numericate the processed input and target. The max input length is 512 tokens, and the max target length is 128 tokens. The tokens are convert to the number based on the index of the vocabulary.

8. Define a custom pretraining dataset and feed relevant dataset into the dataloader.

In addtion, the tokenizer and vocabulary used in this project come from the original T5-base for the following reasons:

1. The compatibility of our model with T5 allows the weight of pretrained T5-base to be reused.Same with tokenizer, it provides a more reliable comparison in the evaluation.

2. The new words and medical terminology can still be represented using the original T5-base vocabulary set.

The pretraining data processing flowchart details the relationship between the source text and terminologies, especially the steps I took to process the pretraining data.

For downstream data, since there is no need to extract medical terms in advance, the ontology extraction steps (step 2 and step 4) are ignored. We can save a lot of time by processing the data directly through the T5 tokenizer from Step 3 through Step 7. BHC-DIN is the only data source for the downstream task.

4.5 Low-resource summarization

In the real world, large-scale document-summary datasets are rare to find and many downstream tasks have only a small amount of samples. Therefore, training from the scratch with limited samples is not sufficent. It is a good idea to solve this problem through pretraining.

In the pretraining, the model trains a large number of unlabeled documents in advance and learns these data more from the sentence-level. Therefore, pre-training is helpful for the model to understand sentence structure.

For example, in the Pegasus(Zhang, Zhao, Saleh and Liu, 2019), instead of learning the span text, the model learns the most important sentence using the GSP method in the pretraining, which leads to the remarkable performance in the low-resource evaluation.

At the same time, the model should learn more information from the token-level and make fine-grained adjustments to the sentence structure in the downstream.

These adjustments should be highly dependent on the characteristics of the downstream data. To some extent, pretraining helps the model to save the time and data needed to relearn sentence structures in the downstream.

Zero-shot is a special case of summarization where performance is directly tested through a pretrained model without giving any sample of the downstream tasks. Generally speaking, as a baseline, the better the zero shot performs, the better the model will perform with more data.

In our project, we would like to test the performance of the model when the downstream training samples are available in different level.

In our project, we wanted to test the model's performance when training samples are limited available.

For example, after Shuffing, we extracted the first 0, 100, 1000, 10,000 and 20,000 BHC-DIN samples. The sampled data were used to verify the effectiveness of the pretraining with limited resources available.

Chapter 5

Experiment

This section describes the detailed setup of experiments related to the methodology & design section. The experiment is divided into four parts, data, pretraining, fine-tuning and evaluation. The configuration and results of four parts are revealed in more detail.

5.1 Data

As mentioned in the section 4.1, there are three data sources for this projects, MIMIC-III, MIMIC-CXR and PubMed Central. These three resources forms the data for both pretraining and fine-tuning.

5.1.1 Pretraining Data

The pretraining data comes from all three data sources. Each pretraining sample should reach up to max length 512 as much as possible. Considering that the unlabeled data was sufficient for pretraining and multiple tokens were replaced by a single sentinel token, we attempted to concatenate them together in terms of the entire article or discharge summary. Therefore, the length of sample is larger than 512. When dataset was processed, the tokenizer truncate the sample to max length. The total number of pretraining samples in this experiment is 500,000, which I think can preliminarily reflect the quality of the model.
The data distribution of 500K samples are illustrates in the following:

1. All other sections of discharge summary except BHC and DIN. There are about 68K samples and the average length of the samples is 674. It accounts for 13.6% of all samples.

2. All the findings and impression content in CXR. There are about 22K samples, accounting for 4.44% of the pre-training data, and the average sequence length is 574.

3. A subset (A-Z) of the PMC database. There are about 410K PMC samples, the average sequence length is 603 with a 82% share of pretraining data.

Clearly, PMC is the main source for pretraining, and the other two sources only help to diversify the data. Besides, more data can be extracted from PMC as I only use one of the four PMC subsets, and other three subsets can be used for the further research. In other words, we could try to abandon CXR and MIMIC sources for pretraining and rely on PMC only for future work.
I believes that 500K samples are good enough to reveal the superiority of ontology extraction.

QuickUMLS

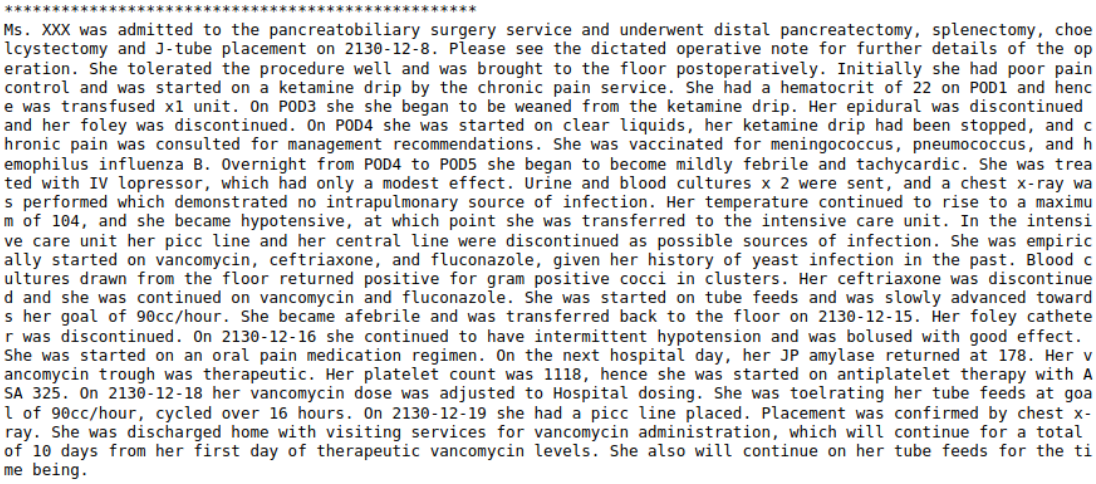
QuickUMLS requires a predefined subset of the meta-thesaurus. So I downloaded the meta-thesaurus first and selected the default subset, which is specifically designed for medical extraction, including the vocabulary from HPO, ICD-9, ICD-10, MeSH, SNOMED CT, etc.
QuickUMLS has a few parameters, such as window size, threshold, semantics and overlapping criteria.
Window size is set to be 5, which means the maximum token to match once from the span of document to UMLS is 5 tokens. With a threshold of 0.8, any similarity between the span text and the target greater than 0.8 is considered a medical term. Because not all medical terms are exactly the same, and they may have close derivatives. Therefore, a threshold is given to accept more possible medical terms within the allowance.
Score is set up to be the overlapping criteria, which measure the priority of different vocabularies. Because if the number of ontologies is too small, a part of the text will be randomly sampled in the later period to meet the number of similarity terms. The higher score is placed at the first.
Semantics are the default setting, following the UMLS semantic groups and types, such as clinical drugs (T200), Antibiotic (T195), Laboratory test(T034), Diagnose Procedure (T060), etc. By default, there are 27 semantic types associated with medical extraction. Deleting or adding some semantics will have an effect on the number of ontologies extracted but will not have much effect on the experimental results. Because if the number of ontologies is too small, additional span of text will be randomly sampled in the later to meet the requirement that 15% of tokens are masked per sequence.

QuickUMLS extracted the index of each ontology in the sample. On average, about 80 indexes were extracted on 512 maximum length.
According to the methodology demonstrated in T5, 15% tokens are masked with a span length of 3. Meanwhile, I made sure that each span text ends up with the whole word. Therefore, approximately 25 ontology indexes are required per sample. If the number of ontology indexes is less than 25, then additional random indexes were selected from the sequence; if greater than 25, then only 25 indexes are randomly selected from these ontology indexes.
Next, mapping these ontology indexes to token indexes and generating the ontology mask. Then, based on the ontology mask, replaced the span text with the sentinel token.

Eventually, half a million medical samples were used for pretraining and another 40,000 samples were used for validation. There is no test data in the pretraining.

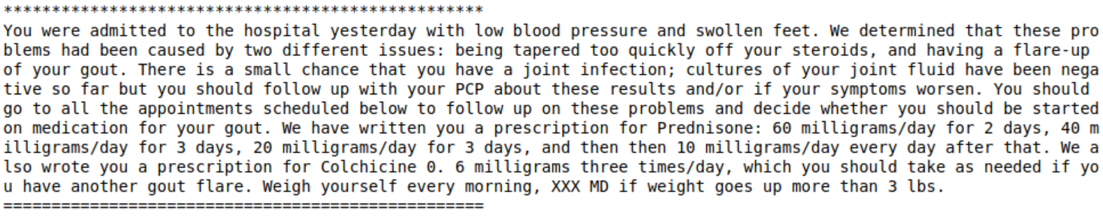
5.1.2 Fine-tuning Data

Brief hospital course and Discharge Instructions in MIMIC-III are the only data used for fine-tuning. The data has already been prepared during the MIMIC-III pretraining data processing, demonstrated in the section 4.4. There is no sentinel token and ontology extraction in the downstream.
With BHC as input text and DIN as target text, the maximum input length is 512 and the maximum target length is 128 respectively. Figure 5.1 and Figure 2 5.2 are the processed brief hospital course and discharge instructions. Compared with the original text Figure 4.1 and Figure 4.2, the sentence structure is more complete and the readability is much higher.



```
**************************************************
Ms. XXX was admitted to the pancreatobiliary surgery service and underwent distal pancreatectomy, splenectomy, choe
lcystectomy and J-tube placement on 2130-12-8. Please see the dictated operative note for further details of the op
eration. She tolerated the procedure well and was brought to the floor postoperatively. Initially she had poor pain
control and was started on a ketamine drip by the chronic pain service. She had a hematocrit of 22 on POD1 and henc
e was transfused x1 unit. On POD3 she she began to be weaned from the ketamine drip. Her epidural was discontinued
and her foley was discontinued. On POD4 she was started on clear liquids, her ketamine drip had been stopped, and c
hronic pain was consulted for management recommendations. She was vaccinated for meningococcus, pneumococcus, and h
emophilus influenza B. Overnight from POD4 to POD5 she began to become mildly febrile and tachycardic. She was trea
ted with IV lopressor, which had only a modest effect. Urine and blood cultures x 2 were sent, and a chest x-ray wa
s performed which demonstrated no intrapulmonary source of infection. Her temperature continued to rise to a maximu
m of 104, and she became hypotensive, at which point she was transferred to the intensive care unit. In the intensi
ve care unit her picc line and her central line were discontinued as possible sources of infection. She was empiric
ally started on vancomycin, ceftriaxone, and fluconazole, given her history of yeast infection in the past. Blood c
ultures drawn from the floor returned positive for gram positive cocci in clusters. Her ceftriaxone was discontinue
d and she was continued on vancomycin and fluconazole. She was started on tube feeds and was slowly advanced toward
s her goal of 90cc/hour. She became afebrile and was transferred back to the floor on 2130-12-15. Her foley cathete
r was discontinued. On 2130-12-16 she continued to have intermittent hypotension and was bolused with good effect.
She was started on an oral pain medication regimen. On the next hospital day, her JP amylase returned at 178. Her v
ancomycin trough was therapeutic. Her platelet count was 1118, hence she was started on antiplatelet therapy with A
SA 325. On 2130-12-18 her vancomycin dose was adjusted to Hospital dosing. She was toelrating her tube feeds at goa
l of 90cc/hour, cycled over 16 hours. On 2130-12-19 she had a picc line placed. Placement was confirmed by chest x-
ray. She was discharged home with visiting services for vancomycin administration, which will continue for a total
of 10 days from her first day of therapeutic vancomycin levels. She also will continue on her tube feeds for the ti
me being.
```

Figure 5.1: An instance of processed Brief Hospital Course.



```
**************************************************
You were admitted to the hospital yesterday with low blood pressure and swollen feet. We determined that these pro
blems had been caused by two different issues: being tapered too quickly off your steroids, and having a flare-up
of your gout. There is a small chance that you have a joint infection; cultures of your joint fluid have been nega
tive so far but you should follow up with your PCP about these results and/or if your symptoms worsen. You should
go to all the appointments scheduled below to follow up on these problems and decide whether you should be started
on medication for your gout. We have written you a prescription for Prednisone: 60 milligrams/day for 2 days, 40 m
illigrams/day for 3 days, 20 milligrams/day for 3 days, and then then 10 milligrams/day every day after that. We a
lso wrote you a prescription for Colchicine 0. 6 milligrams three times/day, which you should take as needed if yo
u have another gout flare. Weigh yourself every morning, XXX MD if weight goes up more than 3 lbs.
==================================================
```

Figure 5.2: An instance of processed Discharge Instructions.

Meanwhile, as required by T5 multi-tasking, added the prefix ”summarize:” to the beginning of all input text to remind the model to perform summarization task.

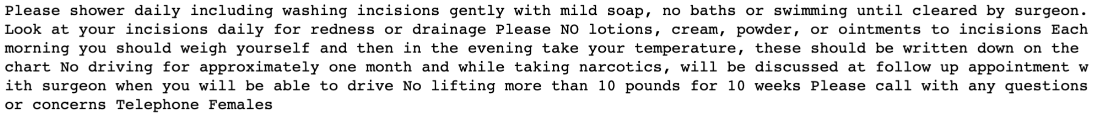
In total, there are about 25,000 samples in the downstream task. According to the ratio of 8:1:1, I used a maximum of 20,000 samples for finetuning, 2000 samples for validation and 2000 samples left for testing. For low-resource summarization, the training samples varied with the availability of resources while the validation samples and test samples remain the same.

Features

Discharge instructions in the MIMIC have two characteristics that are different from other summaries.

First, although the hospital course of patients are different, their discharge instructions may be the same. This makes sense for a real-world dataset. Those patients who have already recovered are more likely given the similar instructions. For example, due to the truncation, 452 identical discharge instructions over 25,000 samples, see Fig 5.3, exist throughout the entire dataset.

Another feature is that many discharge instructions have a formatted prefix, such as ”You were admitted to the hospital”, ”1) Monitor wounds for signs of infection”, ”Please shower daily including washing incisions”, ”Please call your doctor or return to the ER for any of the following”. The prefix ”You were admitted to the hospital” is used at the beginning of 2393 discharge instructions over 25,000 samples. These prefixes are not short and could impact the final results.



```
Please shower daily including washing incisions gently with mild soap, no baths or swimming until cleared by surgeon.
Look at your incisions daily for redness or drainage Please NO lotions, cream, powder, or ointments to incisions Each
morning you should weigh yourself and then in the evening take your temperature, these should be written down on the
chart No driving for approximately one month and while taking narcotics, will be discussed at follow up appointment w
ith surgeon when you will be able to drive No lifting more than 10 pounds for 10 weeks Please call with any questions
or concerns Telephone Females
```

Figure 5.3: One of the most common discharge instructions in the BIH-DIN task.

In order to avoid overfitting, a repetition penalty (Nitish Shirish Keskar, 2019) is introduced in the generation. The control code in the repetition penalty allows for predictable variation in generation even for the identical prompts. It provides more explicit control over the entire text generation. The experiments showed that the rouge score is dropped if repetition penalty is not used. The penalty is set up to 2.0 for all the tasks and the range start from the 1.0 to infinite.

In addition, I observed that the rouge F1 score between BHC and DIN was around 11.894, suggesting that the correlation between inputs and targets is limited.

5.2 Pretraining

The pretraining experiment is divided into three parts. Each part has a pretraining to be generated. These three types of pretraining support all fine-tuning tasks and were used to validate experimental assumptions.

T5 with 500K

The first pretraining is T5 with 500K. T5 with 500K is initialized with pretraining weights from T5-Base and then directly pretrained on the 500K medical samples. Ontology extraction is applied to the pretraining of 500K samples.

Considering that the computational cost of pretraining at both C4 dataset and 500K is relatively large, I chose to directly utilize the pretraining weights from T5-Base. Compared to pretraining samples for T5-Base (C4 has about 67M samples), 500K medical pretraining samples only account for a small proportion. Therefore, if the additional 500K medical samples can make an improvement on the T5-Base's result, it will become a direct evidence that this domain-specific pretraining is effective.

About the configuration, T5 with 500K took 5 epochs (About 300,000 steps) in total. Batch size is set up to 8, which is the maximum I can set up due to the limited GPU memory.
Optimizer is AdamW and learning rate is start with 0.00001.
The linear warmup and square root decay learning rate scheduler were used. Given the warm-up steps, increase linearly to the initial learning rate through these steps. The square root learning rate scheduler is the same with that used in T5. $1/\sqrt{max(n,k)}$ is used in the pretraining, where n is the current training step and k is the number of warm-up steps. I set up a linear warm-up for the first 10K steps and then reduced the learning rate accordingly until the end of the pretraining.
Beam search for generation with the size 4.
In addition, due to limited memory, 500K samples were loaded twice per epoch by data loader, with samples of 250K each time.
Loss is computed through the cross entropy of these non-padding position. Instead of sum up the loss, I chose the average loss of each token.
The model saved a checkpoint every 10,000 steps and got the performance on the situation where loss is minimal.
Again, the purpose of T5 with 500K is to make a comparison to the performance of T5-Base and show the superiority of domain-specific pretraining.

500K with ontology extraction
The second pretraining is 500K with ontology extraction. This pretraining starts from the scratch and it only pretrains on the 500K medical samples.
It applied the ontology extraction approach in the pretraining as well, which means that the embedding is specifically designed for the medical field and that only medical terms were learned by model.
The purpose of this pretraining is to obtain a pure medical embedding and make a comparison to the the third embedding.
The configuration is similar to the first one, except the learning rate is changed to 0.0001. The checkpoint is saved after each epoch.

500K without ontology extraction
The third pretraining is 500K without ontology extraction. 500K without ontology is similar with the second pretraining. However, there is no ontology extraction in this pretraining. In other words, it directly applied the T5 approaches and randomly sampled 15% tokens. It does not guarantee that the each span text contains medical terminology.
The goal of this pretraining is to make a comparison to the second pretraining and find whether the ontology extraction is a appropriate method for this downstream task. The configuration is same with to the second pretaining.

These are the three pretrainings designed for this experiment. The success of this project can be basically determined through these three experiments. Meanwhile, the experimental results was used for further experiments and adjustment.

5.3 Downstream

Downstream task (**BHC-DIN task**) accepted the Brief Hospital Course as the input and generated the Discharge instructions as target. Our model is an abstractive summarization model for extracting the useful information in the BHC, summarizing discharge condition and providing the relevant discharge orders. The BHC-DIN task contains around 25,000 pair of BHC and DIN.

According to the three pretrainings proposed in section 5.2, MIMIC-III downstream task can be divided into following five types, see Fig 5.4:
T5 pretraining: Directly implemented the pretraining of T5-Base and finetuning on the BHC-DIN task.
T5 pretraining with 500K: 500K processed medical samples were taken and the pretraining was initialized with the weight of T5-Base.
500K with ontology: Pretraining from the scratch with 500K medical samples and ontology extraction approach was employed to the pretraining data.
500K without ontology: 500K medical samples were pretrained from scratch, and the pretraining data were randomly sampled. (Exactly the same as T5 pretraining, only with different data sources)
No pretraining: Trained from the scratch.

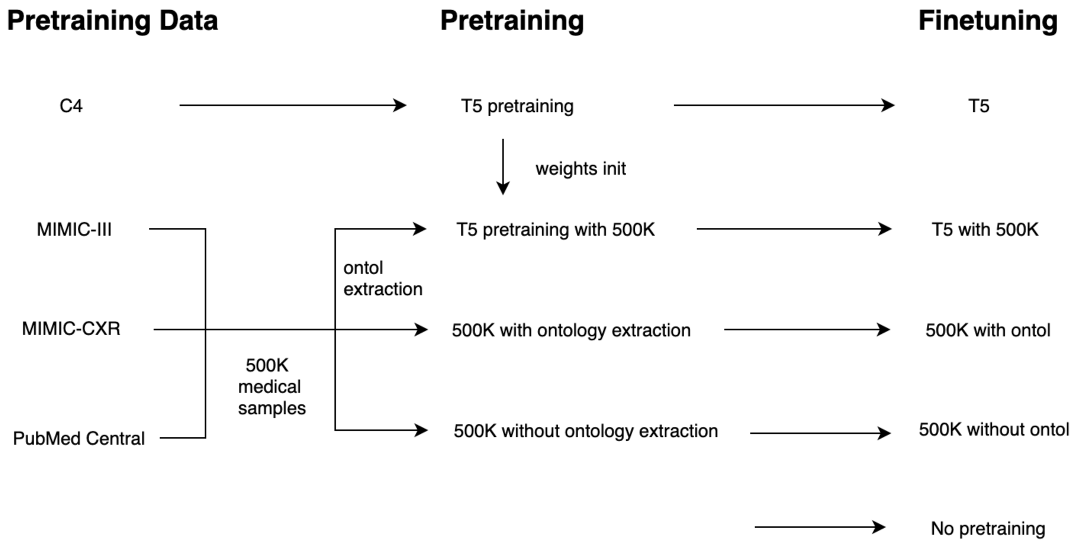


Figure 5.4: The relationship between experiments for pretraining and finetuning.

By comparing the performance of these five tasks, we can verify the following assumptions:

1. By comparing T5 pretraining with non-pretraining, we can easily find out whether pretraining contributes to this medical task.
2. Based on the first assumption, we can determine whether adding 500K medical samples to the task is meaningful through T5 pretraining and 500K of T5 pretraining.
3. Once pretraining is proven to be effective, we shall to further verify if the ontology extraction method is useful through 500K with ontology and 500K without ontology.

Some other conclusions can be drawn along with the progress of experiment, such as the performance of low-resourced summarization.

The downstream task accepted 75K steps (30 Epochs for 20,000 samples). It used the same batch size with the pretraining.
The optimizer is Adam and the learning rate is 1e-4 for all tasks.
Same learning rate scheduler with pretraining but only takes 1000 warm up steps.
For generation, beam search is used with beam size 2.
Early stop when the loss is minimum.
The repetition penalty is 2.
The checkpoint is saved after each epoch as usual.

In addition, the loss will fluctuates greatly and difficult to converge because the batch size is relatively small. As we know, rouge score is the most important criterion to evaluate the performance of a summarization model. Although the rouge score inversely proportional to the loss, it does not mean that the minimum loss matches the maximum rouge score. Also, there is a fluctuation between loss and rouge score. Therefore, I used the early stop and chose the checkpoint, which has the lowest loss.

5.3.1 Low resource summarization

To verify the ability of low-resourced summaries or zero shot, I draw 0, 100, 1000, 10,000, and 20,000 samples from the BHC-DIN dataset for comparison.

Each types of downstream task has these five kinds of sub-tasks. Among so many sets of sub-tasks, I want to make two comparisons in particular:

1. Compared with non-pre-training, I want to show that T5 pre-training performs better with low resources available
2. Compared with T5 pre-training, I would like to show that T5 pre-training with 500K performs better with the limited resources.

The configuration are the same for all the sub-tasks.

5.4 Model

Majority configuration for T5-Base model comes from the huggingFace[1], including model architecture, pretraining weights, T5 Tokenizer and T5-Base configuration. As a standard reference, it greatly speeds up the entire project and ensures the accuracy of the T5 model.

At the same time, considering the extensibility of the project, other scales of T5 can be obtained directly from huggingFace for further study. Some other summarization models from huggingFace, such as Bart and Mass can also serve as a reference in the future work.

5.5 Metrics

For evaluation, there are three kinds of important metrics used to measure the performance of each model, Rouge, BLEU and Perplexity.

Compared with BLUE and Perplexity, Rouge is the most important metric used in the majority summarization models. Our evaluation focus on the Rouge1, Rouge2, and RougeL as well.

The implementation of Rouge comes from rouge-score pypi[2]. The BLEU employed in this project comes from NLTK package and gives the average score through 1,2,3 and 4 grams with equal weights. Perplexity is the exponential of the testing loss.

5.5.1 Ontology evalution

Apart from the three standard metrics mentioned above, another special evaluation used is the ontology evaluation of the generated text.

Ontology extraction can be implemented to the generated text to count the number of terminology exists. It works as the NER to calculate whether the generated text contains enough medical terms.

This is also mentioned by (MacAvaney et al., 2019), which use the RadLex, radiology ontology, to evaluate the performance of ontology PG.

In this project, we employed the same QuickUMLS used in pretraining to directly count and compare the number of medical ontologies of each model. There are 2000 generated instructions and we evaluated the number of medical terms per instructions on average.

[1] https://huggingface.co/transformers/model_doc/t5.html

[2] https://pypi.org/project/rouge-score/

Chapter 6

Evaluation and Results

The evaluation results for the experiments are shown in this section. For each task, there are five criteria to evaluate the performance, including Rouge-f1, Rouge-f2, Rouge-L, BLEU, Perplexity (PPL). As I demonstrates in the section 5.3, there are three assumptions we need to verify based on the experiment results.

According to these assumptions, we have to

1. Find out whether pretraining contributes to this medical task.
2. Find out whether additional 500K medical samples are helpful.
3. Find out whether the ontology extraction method is useful.

6.1 T5 pretraining and non-pretraining

To find out if pretraining contributes to this medical task, we need to compare T5 pretraining with non-pretraining.
The Table 6.1 is the result of using the default T5-Base pretraining and fine-tuning in various available resources.

Num of Samples	Rouge1	Rouge2	RougeL	BLEU (%)	PPL
20000	33.213	13.597	22.664	9.413	4.110
10000	32.663	11.642	20.903	8.690	4.644
1000	31.897	10.324	20.190	7.582	6.744
100	24.009	7.285	17.956	3.644	14.922
0	14.482	1.336	8.576	0.061	277.599

Table 6.1: T5 pretraining performance

T5-Base pretraining achieves rouge1 33.213, rouge2 13.597 and rougeL 22.664 in the test set. Considering the excellent performance of T5-Base in the general dataset (CNN/DM) of 42.05/20.34/39.40, this real-world clinical task is a challenge for T5 Pretraining. This is also why we want to design a domain-specific summarization model.

Num of Samples	Rouge1	Rouge2	Rouge L	BLEU%	PPL
20000	30.462	11.157	19.618	9.014	6.221
10000	29.446	10.411	19.617	6.624	8.605
1000	27.357	7.142	16.456	2.138	80.012
100	20.461	4.243	12.518	1.009	689.49
0	0.372	0.000	0.3513	0.000	51418.844

Table 6.2: Non pretraining performance

If there is no pretraining prepared for this medical task, the model also achieves rouge1 30.462, rouge2 11.157 and rougeL 19.618 as the best results.
Non-pretraining is about 2.751 in rouge1, 2.440 in rouge2 and 3.046 in rougeL lower than T5 pretraining. Therefore, the model with T5 pretraining performs better than without pretraining. It turns out that even the general pretraining is helpful for the model on the domain-specific task.
Generally, the higher rouge score is, the higher BLEU score achieves, which is what we can observe from the Fig 6.1 and 6.2. Compared with the Rouge score, the BLEU of the T5 pretraining is superior to that of non pretraining.
The testing loss of T5 pretraining is much lower than that of non pretraining, which is reflected from the PPL value. T5 pretraining is more likely to get a smaller PPL due to the pretraining.

The reason why pretraining is helpful for the generative task is because that pretraining forces the model to pay more attention to the sentence-level. The sentence structure and relationship between tokens can be learned from the pretraining. Therefore, the model knows how to construct a sentence.
For the downstream, model focus on the token-level, such as how to accurately predict a token, how to use multiple tokens to form a correct word. The downstream, to some extent, helps the model to refine the performance.
In fact, (He et al., 2019) demonstrates that rather than improving accuracy a lot, pretraining speeds up the convergence when a large downstream dataset is given. If the target dataset is small, pretraining is useful for greatly improving the performance.

In the case of limited resources, see Fig 6.1, T5 pretraining is also completely superior to non-pretraining, especially for zero-shot. By learning sentence structure and relationship in pretraining only, the model can obtain the 14.482 in the rouge1 and 1.336 in rouge2.

In addition, a observation is that Rouge score is higher than BLEU score for all the tasks. This is highly because the generated instructions contains many words from the references (High Rouge), but also many words which the references do not include (Lower BLEU). Since the abstractive summarization model does not direct copy the content from the references, it is more likely to extract meaning and be able to use new vocabularies. Besides, the generated text could be longer than the

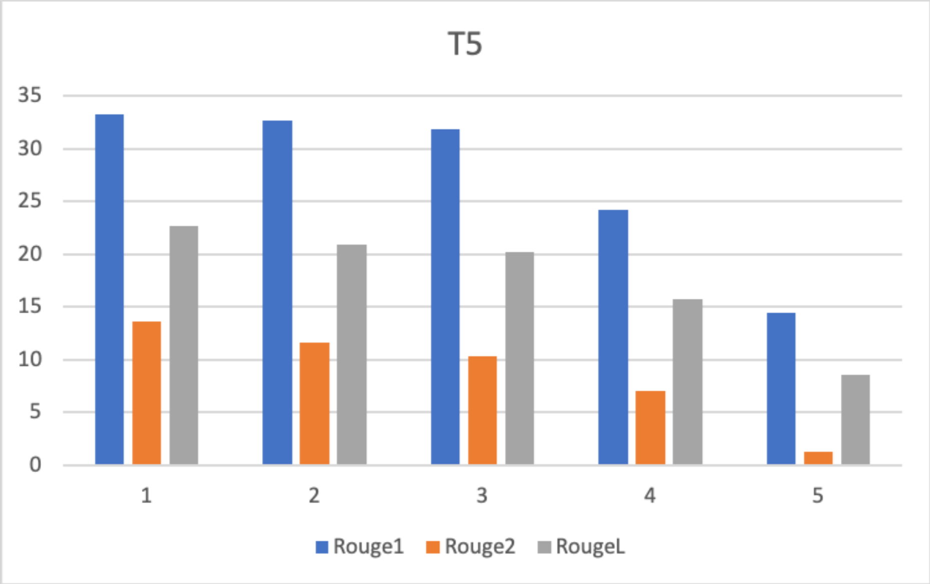


Figure 6.1: A bar chart of T5's performance in low resources.

actual references on average.

In short, this experiment proved that pretraining is helpful to this medical task and it is worth further pretraining and improving.

6.2 T5 pretraining with 500K medical samples

Involving 500K medical data for pretraining, through the pretraining, the model can understand the relationship between medical terms. Meanwhile, applying ontology extraction on the 500K medical samples in order to force the model learn the medical terminology as much as possible.
The following is my experiment results for the T5 with 500K.

Num of Samples	Rouge1	Rouge2	Rouge L	BLEU%	PPL
20000	36.819	15.496	25.087	11.404	4.801
10000	35.331	15.326	24.291	9.118	5.269
1000	32.740	12.843	22.025	6.783	7.740
100	25.862	10.778	19.888	4.027	10.162
0	16.520	1.520	9.204	0.133	121.856

Table 6.3: T5 with 500K medical sample's performance

From the Table 6.3, we can observe that the loss of T5 with 500K (PPL) is slight higher than that of T5, but the performance in the Rouge and BLEU are higher in the 20000 samples, especially the Rouge1 score shows that T5 with 500K completely outperforms than that of T5. This is probabily because T5 with 500K has better understanding about the relationship between common words and medical terms. Compared with T5 pretraining in the Table 6.1, the results in the Table 6.3 are more superior. With the same model size, T5 with 500K achieved the SOTA for this task, which is Rouge1 36.819, Rouge2 15.496, RougeL 25.087 and BLEU 11.404%, and the improvements on Rouge1, Rouge2, RougeL and BLEU are 3.606, 1.899, 2.423 and 2.39, respectively.

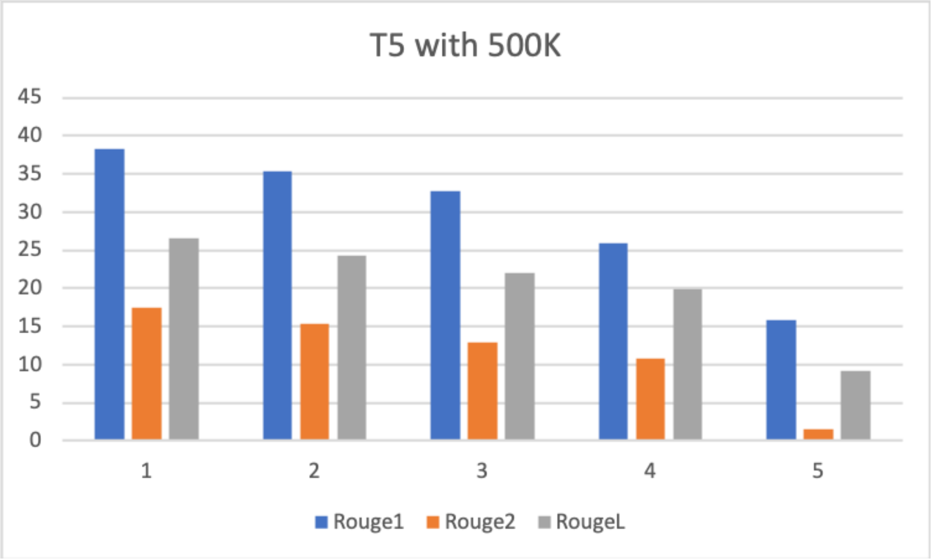


Figure 6.2: A bar chart of T5 with 500K performance in low resources.

The bar chart, see Fig 6.2, illustrates the difference between each low level resources. As expected, the value of all Rouge scores continues to grow between zero and 20,000 samples, with the fastest growth around zero and the slowest growth around 20,000. Because the more training samples given, the better performance the model get. But model training still has a limit, so it grows slowly as more data becomes available.

The finetuning for the T5 and T5 with 500K took about 30 epochs and the testing loss can be seen from the Figure 6.3. The loss of T5-Base is the lowest at 27 epoch and then loss increases and continuous slight fluctuations. However, the loss of the T5 with 500K continues to decline in the 30 Epochs. The lowest point is at 30 epoch. It has a continuous downward trend and is gradually approaching the bottom of T5-Base.
Actually, the epoch for the T5 with 500K should increase until there is no improvement for the loss. Given the time constraints and the uniformity of all training setup, I left the 75,000 steps (30 epochs for 20,000 samples) for all the resources level.

From the Figure 6.4, it shows the rouges score of each epoch. Initially , the rouge scores for T5 with 500K medical samples is lower than that of T5-Base. However, it catch up with the T5-Base around 10 epoch and transcend T5-Base at last.
I think the reason T5 performs better with a 500K medical sample is that it explores

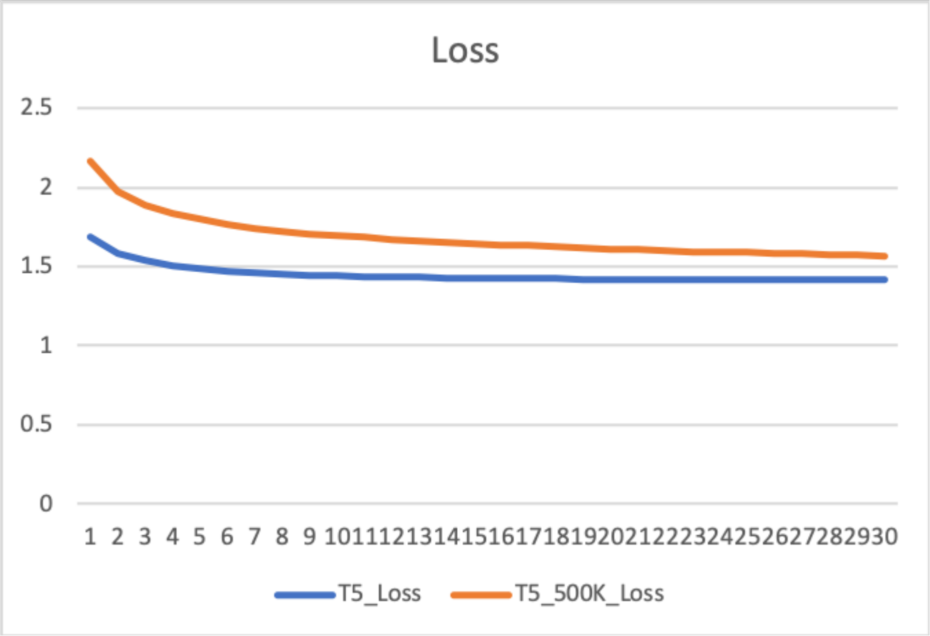


Figure 6.3: The validation loss of T5 and T5 with 500K in 20,000 samples.

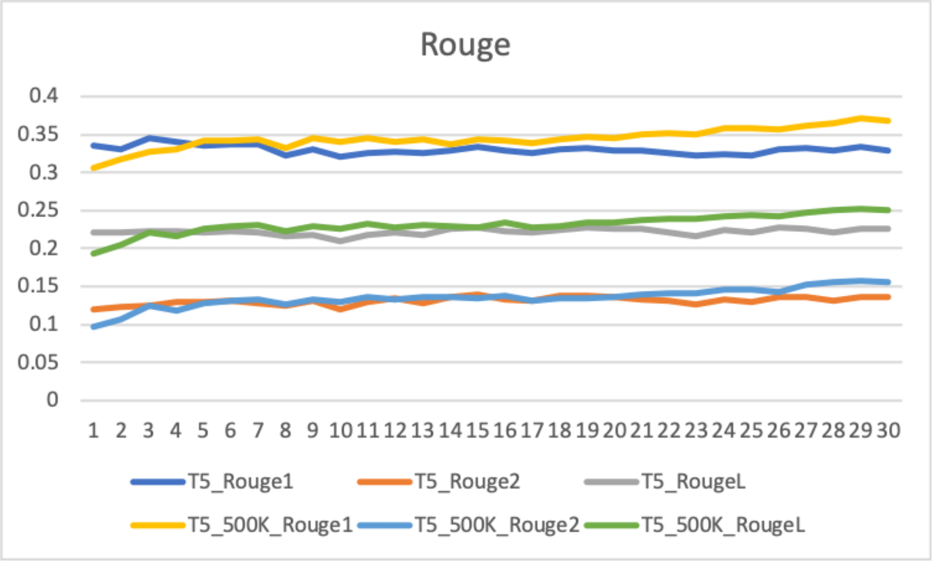


Figure 6.4: The rouge of T5 and T5 with 500K in 20,000 samples.

the relationship between common words and medical terms in the pretraining. At the beginning, the loss of T5 pretraining was lower and the performance should better, but with the emergence of more medical terms found in the training samples, T5 with 500K medical samples had advantages.
In addition to learning the relationship between general words through T5-Base weights, T5 with 500K medical samples enables the use of the same vocabulary and tokenizer to extend the relationship between the general words into the medical field.
To sum up, this experiment confirmed our second assumption that additional 500K

medical samples are conducive to the pretraining of the model and its prediction of medical terms in downstream.

6.3 500K medical samples with or without ontology extraction

In this part of the experiment, we further studied whether the directly applying ontology extraction for pretraining is better than random extraction.
For the pretraining of T5, it randomly samples the span of text. However, we would like to try ontology extraction for domain-specific summarization models to see if this approach improves performance.

Num of Samples	Rouge1	Rouge2	Rouge L	BLEU%	PPL
20000	29.625	10.342	18.906	6.317	5.126
10000	28.956	9.113	18.306	4.343	8.005
1000	27.648	7.762	16.167	3.182	11.050
100	23.306	5.149	14.024	2.322	34.696
0	12.482	0.875	7.474	0.000	2269.055

Table 6.4: The performance of 500k with ontology extraction.

Num of Samples	Rouge1	Rouge2	Rouge L	BLEU%	PPL
20000	31.421	12.013	20.681	7.729	5.051
10000	29.886	11.129	19.795	6.007	5.888
1000	28.124	8.286	16.852	4.112	11.165
100	23.540	6.010	15.157	2.740	31.271
0	13.757	0.907	8.501	0.000	2291.835

Table 6.5: The performance of 500K without ontology extraction.

The Table 6.4 and Table 6.4 represented the performance of 500K samples with and without ontology extraction pretraining. It showed that random extraction performs better than ontology extraction in general.
The performance of random extraction with 20000 training samples is slightly higher than that of ontology extraction. There is a gap about 1.796, 1.671 and 1.775 in three rouge scores between random extraction and ontology extraction.

The relationship among ontology extraction, random extraction of 500K and other three pretrainings was analyzed in the following:

Compare to the performance of T5 and T5 with 500K, 500K with and without ontology extraction have worse performances in the both Rouge scores and BLEU. This phenomenon is reasonable, and is exactly what I expected.
Considering the pretraining configuration in T5, this may be caused by:

1. T5 pretraining takes more steps and longer training time.

2. The size of C4 dataset is much larger than that of 500K medical samples. Besides, T5 pretraining samples never repeated during pretraining.

These two reasons directly lead to the superiority of T5 pretraining or using of T5 weights.

Compare to the performance of non pretraining, 500K without ontology extraction performs slightly better whereas 500K with ontology extraction is a little bit worse. However, there is no big gap between the three situations. This may be caused by:

1. 500K pretraining do not take sufficient training steps. The batch size is too small to converge, so the result may fluctuate around the final result.

2. Generation configuration may be different for the 500K pretraining and training from scratch. It needs to re-explore the other possible configurations.

According the conclusion draw from the first assumption, pretraining contributes to this downstream task. However, the performance of 500K with ontology extraction is not as good as that of no pretraining when more samples are given, which is most likely due to the reasons explained in the following comparison.

Compare to the performance of 500K with ontology extraction, 500K without ontology extraction performs better.
The reason why the performance of 500K ontology extraction is worse than that of no ontology extraction is mainly because the sequences generated from ontology extraction do not support the common words. Common words are inevitable to form a sentence. The relationship between common words are not learned in the pretraining, only the medical terms are learned. Therefore, the model cannot correlate the relationship between common words and specialized words.
However, for T5 with 500K pretraining, it already understands the relationship between common words in the T5 weights. And in the middle tasks, it relates the common words to the medical terminology. Hence, directly pretraining from scratch with ontology extraction is not as good as that of random sampling.

The Table 6.6 also showed that the number of terminologies extracted from 500K with ontology extraction is much more than that of without ontology extraction in the same generated text. It also more than the number of terms in the actual text, which indicates that 500K with ontology extraction is over-predict the medical terms. It considers many common words as the medical terms.
A better idea is whether a ratio can be set to determine how many ontologies need to be extracted, which could be part of future research.

6.4 Low resource evaluation

In this part, we compared the performance gap between each model for Rouge1, Rouge2, and RougeL. The low resource evaluation is also analysed below.

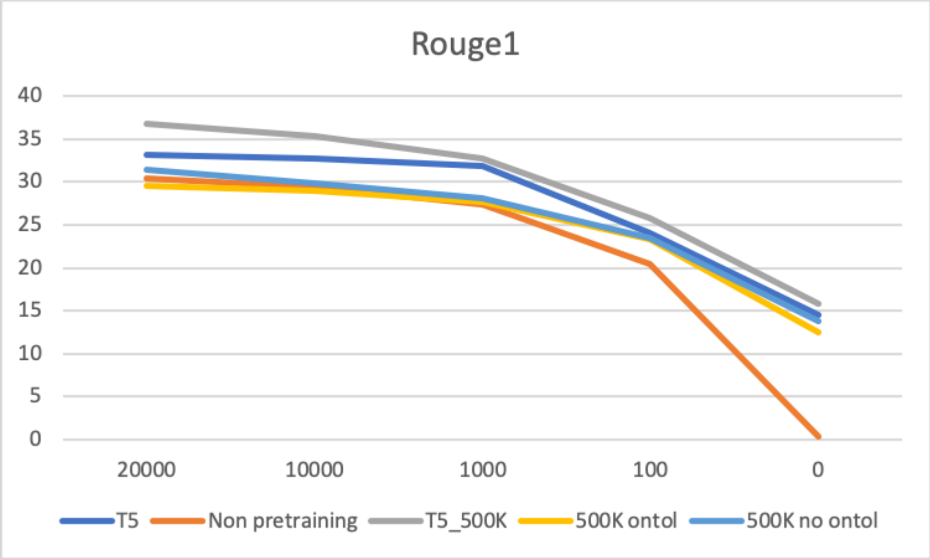


Figure 6.5: The rouge1 for all source levels.

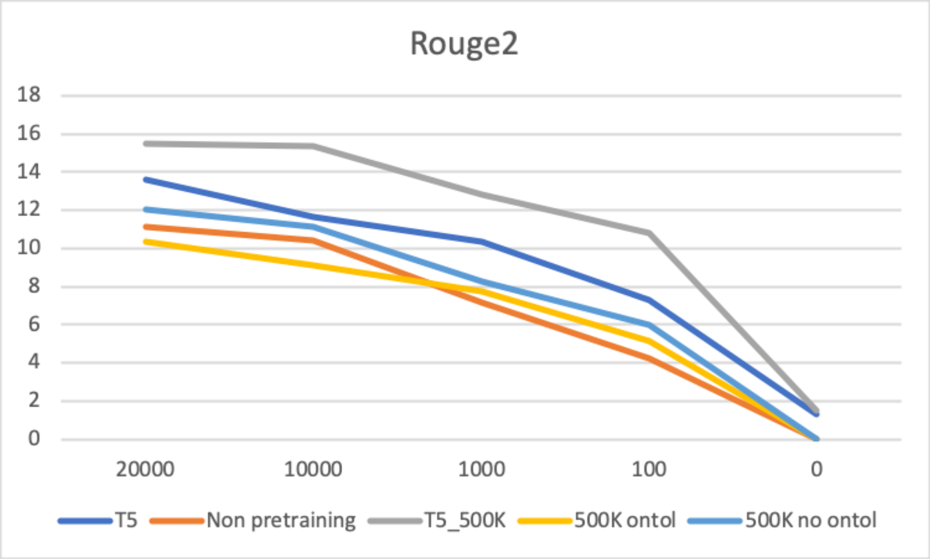


Figure 6.6: The rouge2 for all source levels.

The low resource performance for each tasks are shown in the Fig 6.5, Fig 6.6 and Fig 6.7. The best performance of low resource is T5 with 500K medical samples, the worst is the no pretraining.
Here are some analyses and observations of low resource summarization:

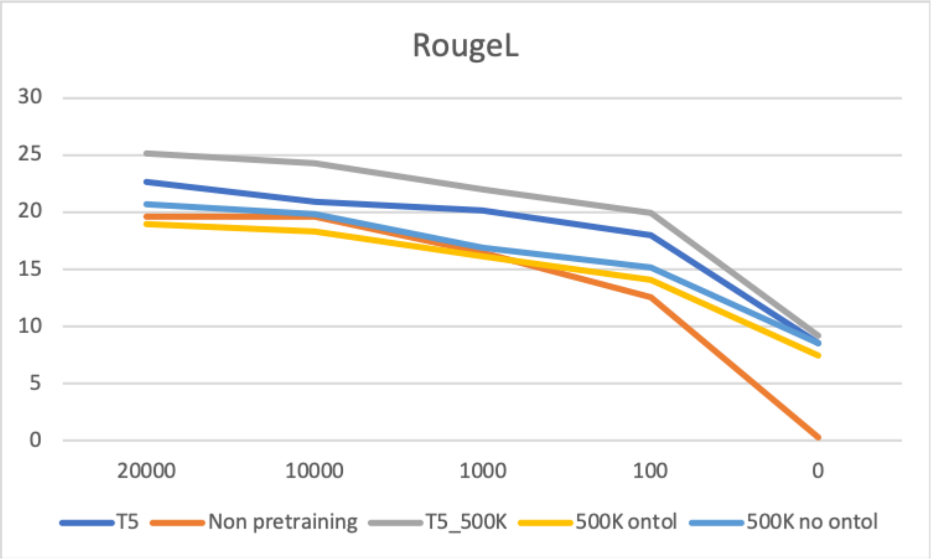


Figure 6.7: The rougeL for all source levels.

1. For all the models, the more training samples given, the better performance it achieves.

2. For all the rouge scores, T5 with 500K medical samples is superior to T5-Base, indicating that our pretraining outperforms than T5 in all source level for this task.

3. For all the rouge scores, the model with pretraining is superior to non pretraining, indicating that pretraining contributes to this medical task.

4. An exception is 500K with ontology extraction, which performs poorly when the sample size exceeds 1000, even worse than non pretraining. This is because the relationship between common words are not taken into account in pretraining and only the medical terms are pretrained. This becomes a mislead for the downstream task and ultimately worse when more samples are given.

6.5 Ontology Evaluation

Ontology evaluation measured how many medical terms can be found in the 2000 testing generated samples. The Table 6.6 showed the number of medical terms extracted from each generated discharge instruction using the same QuickUMLS in the pretraining. As a reference, there are about 17.854 medical terms extracted from each target reference on average.

In the Table6.6, it does not mean that the more ontologies extracted from prediction, the better the model is. This is because some models only learn the relationship between ontologies, such as 500K with ontology extraction, so the tokens predicted

Training Samples	T5	T5_500K	500K_ontol	500K_no_ontol	No_pretrain	Actual
20000	17.068	18.849	19.435	17.850	16.905	17.854
10000	16.715	18.313	18.761	16.250	14.742	—
1000	14.113	17.139	17.514	13.876	12.465	—

Table 6.6: The number of medical terms extracted from the generated text.

are more likely to be medical terms, which is a overfitting between the terminologies and common words.
Basically, we can observe that the generated text using the 500K medical samples in the pretraining contain more medical terms than those do not use. For example, the ontologies from T5 with 500K is more than that of T5, the terminologies found from 500K with ontology extraction and without extraction are more than that of no pretraining.
In addition, we can observe that the T5 with 500K can generate more medical terms than direct using T5 pretraining. This is because, in the middle task, T5 with 500K focused on the relationship between medical terms that T5 had never access to before.
For 500K models with and without ontology extraction, the number of ontologies predicted through ontology extraction is significantly higher than that without ontology extraction. This is because 500K only learns terms through ontology extraction, while 500K without ontology extraction is able to access common words through random sampling. On the other hand, 500K with ontology extraction is overfitting on the predicting the common words and medical terms.

In general, Table 6.6 meets our expectations:

1. The generated medical terms from those pretrainings that used ontology extraction is more than that are not used.

2. The generated medical terms from those pretrainings that used 500K medical samples is more than that are not used.

6.6 Pretraining performance

In this section, I presented some observations during the pretraining and compare the performance of each pretraining.

The Figure 6.8 showed the training and validation loss of the T5 pretraining with 500K medical samples. The interval of validation is 31250 steps. The best performance is where the lowest loss locates, which is about the 100,000 steps, see from the Figure 6.8. However, we can observe there is a overfitting occurs around 110K steps for both training and validation loss. The pretraining loss is dramatically increased, which is unusual. According to finding from the (Gururangan et al., 2020), pretraining should never stop and it will not overfitting even for the domain adaptive

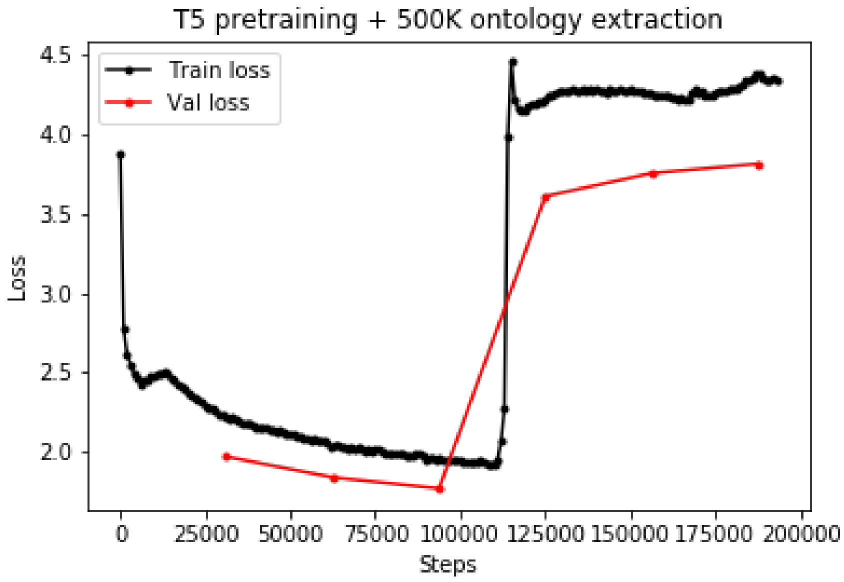


Figure 6.8: The losses of T5 with 500K ontology extraction.

transfer learning. In other words, we should do pretraining for as long as possible.

However, this does not apply to my results. I think there are several reasons for this phenomenon:

1. The amount of pretraining data is limited. 500K medical samples was not compatible with the massive general dataset C4. Therefore, it is possible to result in overfitting.

2. The middle task can be viewed as the a fine-tuning of the pretraining. Due to the ontology extraction, the predicted terms were not random. They were in the same domain and many terminologies were repeated in the same document, which leaded to the overfitting eventually.

3. This model only predicted the medical terms in pretraining by using the general vocabulary, in which the connection between common words and medical terms were limited. If some common words and medical terms were introduced together during the pretraining, the vocabulary can be more compatible with the relationship between the two.

To verify my ideas above, I added a new pretraining, which is similar to T5 with 500K pretraining. Instead of using the ontology extraction, the extraction method was replaced with random extraction, which had the possibly to sample both common words and medical terms.

Compared to Fig 6.8, Fig 6.9 presented a pretraining without overfitting, which is what I expected. This means that changing the ratio of medical terms and common words can avoid overfitting in the pretraining.

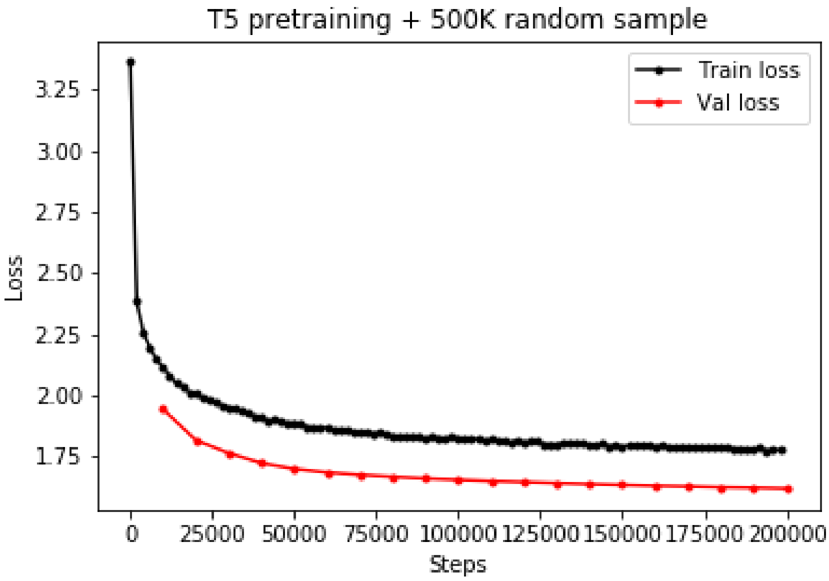


Figure 6.9: The losses of T5 with 500K random sampling.

Although both losses reached about 1.6 in the same step, if there is no overfitting, the pretraining is able to take longer and may achieve better results. Different with ontology extraction, the interval of validation loss is about 10,000 steps in the random sampling.

The following Fig 6.10 and 6.11 are two pretrainings started from the scratch.

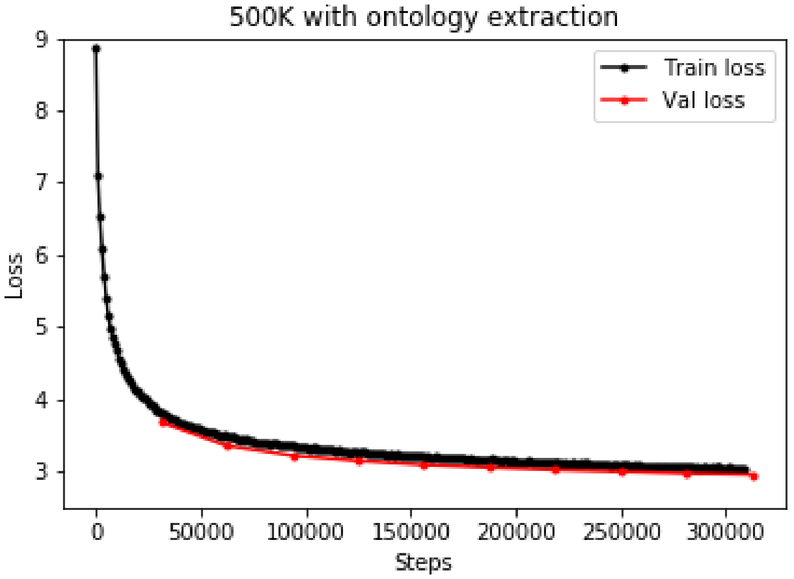


Figure 6.10: The losses of 500K with ontology extraction.

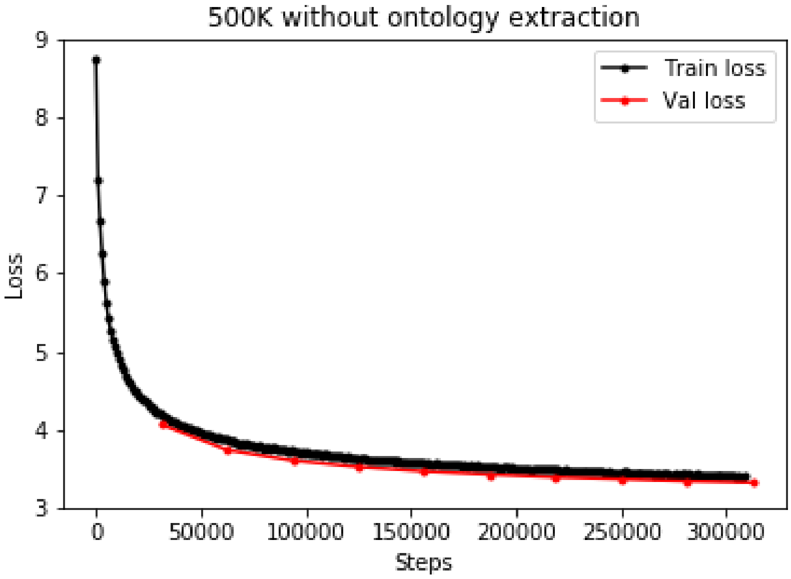


Figure 6.11: The losses of 500K without ontology extraction.

Since the pretraining was learned from scratch and was not a middle task, the pretraining itself did not overfitting. In other words, due to the large number of unlabeled samples provided, the longer pretraining took, the better it performed in the downstream.

The loss of these two 500K pretrainings (Fig 6.10 and 6.11) is completely higher than these using T5 weights (Fig 6.8 and 6.9). This is because T5 pretraining had a much better understanding of the relationships between general words than others. Therefore, in the middle task, two pretrainings (Fig 6.8 and 6.9) also had advantage to understand the relationship between medical terms.

The loss of with ontology extraction is lower than that without ontology extraction. This may be because without ontology extraction is more likely to extract common words, which is difficult to predict the relationship in comparison with the medical terms extracted from ontology.

Medical terms are more likely to overlap in comparison with common words. In the same document, the medical terms extracted appear more than once. However, the common words has different possibilities, including but not limited to nourn, verb, adverb, and many other part of speech. Therefore, the loss for random sampling from the scratch is slightly higher than that of ontology extraction.

Chapter 7

Conclusions

In this work, a summary model designed for the clinical domain was presented. The model can generate discharge instructions for patients by learning their hospital course. Automatically generating discharge instructions can save clinicians considerable time and provide patients with personalized rehabilitation and precautions.
Through three data sources, MIMIC-III, MIMIC-CXR and PMC, we extracted 500K unlabeled medical samples for pretraining and a real-world downstream task, in which brief hospital courses were the input text and discharge instructions were the target sequences.
Secondly, ontology extraction was employed to the pretraining data in order to force the model familiar with the relationship between medical terms more quickly.
Compared with ontology extraction, we also tried the model without ontology extraction, which is random sampling, and made a comparison. Our findings showed that compared with ontology extraction, the pretraining from scratch is more reliable and effective by using random sampling.
We conducted three pretrainings, T5 with 500K, 500K with ontology and 500K without ontology, and analysed their performance based on the BHC-DIN downstream task.
The experimental results showed that pretraining on the 500K medical samples initialized with the weight of T5-Base performs the best and achieves the best in this task.
In addition, we conducted low-resourced assessments for diffenent pretraining, demonstrating that our pretraining can achieve better performance in the limited samples.
In summary, I produced a clinical domain text generator that can help clinicians write reliable discharge instructions to better serve patients. Moreover, I explored the application of pretraining in general fields and specialized fields. Based on additional 500K medical data samples, I generated a new pretraining of the T5-Base model for the medical domain. The effectiveness of ontology extraction was also studied under different pretrainings.

7.0.1 Future work

1. Instead of just using DHC-DIN task, we should find some other clinical and biomedical summary tasks to test the performance of our pretraining, such as impressions and findings in the MIMIC-CXR.

2. As mentioned in the section 6.3, instead of using ontology to extract medical terms only or using the random sampling for common words, a trade-off should be explored about how many medical terms and common words are used in the pretraining.

3. More pretraining data should be prepared. For example, our experiment used only part of the PMC articles, and more articles can be extracted for further pretraining. Meanwhile, considering MIMIC-CXR only accounts for 4.4% of 500K samples, it seems to be a downstream task better than the pretraining data.

Bibliography

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T. and McDermott, M. (2019), 'Publicly available clinical bert embeddings', *arXiv preprint arXiv:1904.03323* . pages 9

Amin-Nejad, A., Ive, J. and Velupillai, S. (2020), Exploring transformer text generation for medical dataset augmentation, *in* 'Proceedings of The 12th Language Resources and Evaluation Conference', pp. 4699–4708. pages 21, 22

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* . pages 6, 7

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. and Smith, N. A. (2020), 'Don't stop pretraining: Adapt language models to domains and tasks', *arXiv preprint arXiv:2004.10964* . pages 53

He, K., Girshick, R. and Dollár, P. (2019), Rethinking imagenet pre-training, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 4918–4927. pages 45

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J. et al. (2001), 'Gradient flow in recurrent nets: the difficulty of learning long-term dependencies'. pages 4

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L. and Levy, O. (2020), 'Spanbert: Improving pre-training by representing and predicting spans', *Transactions of the Association for Computational Linguistics* **8**, 64–77. pages 10, 11

Kishore Papineni, Salim Roukos, T. W. W.-J. (2002), 'Bleu: a method for automatic evaluation of machine translation'. pages 23

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H. and Kang, J. (2020), 'Biobert: a pre-trained biomedical language representation model for biomedical text mining', *Bioinformatics* **36**(4), 1234–1240. pages 8, 28

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. (2019), 'Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension', *arXiv preprint arXiv:1910.13461* . pages 14, 15

Lin, C.-Y. (2004), ‘Rouge: A package for automatic evaluation of summaries’. pages 22

Liu, P. J. (2018), ‘Learning to write notes in electronic health records’, *arXiv preprint arXiv:1808.02622* . pages 20, 21

Luca Soldaini, N. G. (2020), ‘Quickumls: a fast, unsupervised approachfor medical concept extraction’, https://github.com/Georgetown-IR-Lab/QuickUMLS. pages 3, 30

MacAvaney, S., Sotudeh, S., Cohan, A., Goharian, N., Talati, I. and Filice, R. W. (2019), Ontology-aware clinical abstractive summarization, *in* ‘Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval’, pp. 1013–1016. pages 11, 29, 43

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. et al. (2016), ‘Abstractive text summarization using sequence-to-sequence rnns and beyond’, *arXiv preprint arXiv:1602.06023* . pages 4

Nitish Shirish Keskar, Bryan McCann, L. R. V. C. X. R. S. (2019), ‘Ctrl: A conditional transformer language model for controllable generation’. pages 39

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018), ‘Deep contextualized word representations’, *arXiv preprint arXiv:1802.05365* . pages 6

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2018), ‘Improving language understanding by generative pre-training’. pages 6

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2020), ‘Pubmed parser: A python parser for pubmedopen-access xml subset and medline xml datasetxml dataset’, https://github.com/titipata/pubmed_parser. pages 29

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J. (2019), ‘Exploring the limits of transfer learning with a unified text-to-text transformer’, *arXiv preprint arXiv:1910.10683* . pages 1, 17, 18, 19, 31

See, A., Liu, P. J. and Manning, C. D. (2017), ‘Get to the point: Summarization with pointer-generator networks’, *arXiv preprint arXiv:1704.04368* . pages 4, 11, 12

Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.-Y. (2019), ‘Mass: Masked sequence to sequence pre-training for language generation’, *arXiv preprint arXiv:1905.02450* . pages 12, 13

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017), Attention is all you need, *in* ‘Advances in neural information processing systems’, pp. 5998–6008. pages 1, 4, 5

Williams, R. J. and Zipser, D. (1989), 'A learning algorithm for continually running fully recurrent neural networks', *Neural computation* **1**(2), 270–280. pages 32

Zhang, J., Zhao, Y., Saleh, M. and Liu, P. J. (2019), 'Pegasus: Pre-training with extracted gap-sentences for abstractive summarization', *arXiv preprint arXiv:1912.08777* . pages 15, 16, 35

Zhang, Y., Merck, D., Tsai, E. B., Manning, C. D. and Langlotz, C. P. (2019), 'Optimizing the factual correctness of a summary: A study of summarizing radiology reports', *arXiv preprint arXiv:1911.02541* . pages 28

Appendix

Code

There are several files in my code files, including

Pretraining: The main pretraining file to pretrains the model in Epoch and Steps. It has training, validation, pretraining methods. The dataset loader function only loads a given number of datasets. Considering that the training sample is too large to be loaded by a single loader, there are two loaders, each loader containing 250K samples. Three pretrainings, T5 with 500K, 500K with ontol and 500K without ontol are produced from this file.

Finetuning: The main file for finetuning the downstream tasks in Epoch and Steps. Similar to Pretraining file, but the dataset load function is different. The data loader loads training, validation and testing loader together. Five types of downstream task, T5, T5 with 500K, 500K with ontol, 500K without ontol and non-pretraining are performed in this file.

PretrainDataProcess: A file to process all the pretraining samples. It has processing methods about MIMIC-III, MIMIC-CXR, PMC and ontology extraction methods.

DownstreamDataProcess: A file to process the downstream samples. It has processing methods for brief hospital course and discharge instructions.

Evaluation: A file to evaluate the generate text in the Rouge, BLUE and PPL.

Dataset: A file contains custom datasets for the pretraining and downstreams.

Utils: A file contains many functions such as span masking, replace sentinel, train test splits, etc.

Configuration

For pretraining:

Types	**epochs**	**lr**	**Batch size**	**Beam size**	**warm up**
T5 with 500K ontol	5	1e-5	8	4	10K
500K with ontol	5	1e-4	8	4	10K
500K without ontol	5	1e-4	8	4	10K

Table 7.1: Configuration for pretraining

For finetuning:

Types	**steps**	**lr**	**batch size**	**beam size**	**repet penalty**	**warm up**
T5	75K	le-4	8	2	2	1K
T5 with 500K	75K	le-4	8	2	2	1K
500K with ontol	75K	le-4	8	2	2	1K
500K without ontol	75K	le-4	8	2	2	1K
no pretraining	75K	1e-4	8	2	2	1K

Table 7.2: 500k without ontology performance

Examples

Actuals

1. 1) Please shower daily including washing incisions gently with mild soap, no baths or swimming until cleared by surgeon. Look at your incisions daily for redness or drainage. 2) Please NO lotions, cream, powder, or ointments to incisions. 3) Each morning you should weigh yourself and then in the evening take your temperature, these should be written down on the chart provided. 4) No driving for approximately one month and while taking narcotics. Driving will be discussed at follow up appointment with surgeon when you will likely be cleared to drive. 5) No lifting more than 10

2. Please call your doctor or return to the ER for any of the following: * You experience new chest pain, pressure, squeezing or tightness. * New or worsening cough or wheezing. * If you are vomiting and cannot keep in fluids or your medications. * You are getting dehydrated due to continued vomiting, diarrhea or other reasons. Signs of dehydration include dry mouth, rapid heartbeat or feeling dizzy or faint when standing. * You see blood or dark/black material when you vomit or have a bowel movement. * Your pain is not improving within

3. Dear Mrs. XXX, You were hospitalized because you have a bacterial infection in your urine. Additionally, you were found to be hypotensive (low blood pressure), and you were initially admitted to the ICU. Your blood pressure improved after getting IV fluids, and you were started on antibiotics for treatment of your urinary tract infection. You were transfered out of the ICU, and your symptoms continued to improve. It was a pleasure taking care of you. Please

note that the following changes have been made to your medications: 1. Please take ciprofloxacin 500

4. Please call Dr. XXX office Telephone if you develop: * Fever (¿101 F) or chills * Nausea or vomiting * Abdominal or chest pain * Shortness of breath * Any other concerns You may shower. pat your chest incision dry after showering. The steri-strips will curl up and fall off over time. Do not remove them. The VNA will change your chest dressing daily- call if the drainage increases, becomes yellow/green or smells foul. take a mild laxative and stool softner while you are taking pain medication to

5. You were transferred to Hospital on 2200-3-8 for evaluation for a TIPS procedure after your second variceal bleed in a few weeks. You were evaluated by our liver specialists who felt you would benefit from the procedure. Unfortunately, this procedure was unsuccessful on first attempt. It is very important that you refrain from lifting heavy objects. You remained comfortable and without repeat episodes of bleeding. You will have the repeat-procedure done next week. The following follow-up visits are extremely important: - Please arrive at the liver clinic this coming Tuesday promptly at 8:00am. You will need bloodwork checked

6. * You were admitted to the hospital for lung surgery and you've recovered well. You are now ready for discharge. * Continue to use your incentive spirometer 10 times an hour while awake. * Check your incisions daily and report any increased redness or drainage. Cover the area with a gauze pad if it is draining. * Your chest tube dressing may be removed in 48 hours. If it starts to drain, cover it with a clean dry dressing and change it as needed to keep site clean and dry. * You will continue to need pain medication once you are home but you can wean

7. You were admitted with a heart attack. Several of your medications have been changed.. Please seek medical attention if you develop a fever ¿ 101 F, chest pain, shortness of breath, groin swelling or any other concerning symptoms.

8. You were admitted with cholangitis and had a biliary stent changed while hospitalized. You should continue all antibiotics prescribed. You will need to return to have this stent removed in 3 months. The GI department will be contacting you.

==
T5

1. 1) Please shower daily including washing incisions gently with mild soap, no baths or swimming until cleared by surgeon. Look at your incision sites daily for redness or drainage 2) Please NO lotions, cream, powder, and ointments to incision areas

2. Please call your doctor or return to the ER for any of these: * You experience new chest pain, pressures, squeezing or tightness. * New or worsened cough

or wheezing; difficulty breathing – If you are vomiting and cannot keep in fluids or your medications. * Your wound was closed with staple removal by endotracheal cauterization 2. Do not drive while taking this medication 3. No heavy lifting (¿ 10lbs) 4. Resume all home mediciation as prescribed

3. Dear Ms XXx, It was our pleasure taking care of you during your hospitalization at Hospital. You were admitted for syncope and found to have an air embolism (blood clot in the lung). We treated this with blood thinner medication called Lovenox which helps prevent future heart attacks as well that may be contributing factors behind these episodes but we did not make any changes on discharge from rehab until after they are resolved by physical therapy/rehabilitation

4. Please call your doctor or return to the ER for any of these: * You experience new chest pain, pressures, squeezing or tightness. * New or worsening cough or wheezing; Increased redness and/or drainage from incision sites – If you are vomiting and cannot keep in fluids or your medications. Signicant signs such as fever greater than 101 (F) degrees or 38(C) degrees F* Any serious change in mental status including but not limited only by severe headache or dizziness

5. You were admitted to the hospital with a variceal blood loss. Your liver tests have been stable, and you are being discharged for another TIPS procedure on 3-19 (to be done in 5 days). Please follow-up as below

6. You were admitted to the hospital with pneumonia. We treated you in our ICU and then transferred your outpatient medications for treatment of this infection into a general medicine unit where we continued antibiotics through IV'S (Vancomycin, Ceftriamethoxacin) until 2169-5-6 when they will be administered at home by DrXXX clinic as an outpatient

7. You had a heart attack and needed to have cardiac catheterization. The procedure was complicated by disection of your left anterior coronary vessel, which caused the blockage in one part or another that is still open today (the LAD). Your chest pain has now resolved but you will need close follow-up with DrsXXx as an outpatient for further management

8. You were admitted with cholangitis, an infection of your gallbladder. This was treated by antibiotics and you will need to continue taking these for 7 days total (to complete the course). Please follow-up as below

==

T5-500K

1. Please shower daily including washing incisions gently with mild soap, no baths or swimming until cleared by surgeon. Look at your incisions daily for redness or drainage Please NO lotions, cream, powder, or ointments to incisions Each morning you should weigh yourself and then in the evening take your temperature, these should be written down on the chart No driving for

approximately one month and while taking narcotics, will be discussed at follow up appointment with surgeon when you shall be able to drive No lifting more than 10 pounds for 12 weeks Please call with any questions or concerns Telephone Female

2. Please call your doctor or nurse practitioner or return to the Emergency Department for any of the following: *You experience new chest pain, pressure, squeezing or tightness. *New or worsening cough, shortness of breath, or wheeze. *If you are vomiting and cannot keep down fluids or your medications. *You see blood or dark/black material when you vomit or have a bowel movement. *Your skin, or the whites of your eyes become yellow or tarry when standing. * You are getting dehydrated due to continued vomiting, diarrhea, or other reasons

3. Dear Ms. XXX, It was a pleasure taking care of you during your hospitalization at Hospital. You were admitted to the hospital because you had a urinary tract infection. We treated you with IV fluids and antibiotics for this infection. Your kidney function improved, and you are now ready for discharge home. Please take all medications as prescribed.

4. Call Dr. XXX office Telephone if you experience: -Fevers greater than 101 or chills -Increased shortness of breath, cough or chest pain -Incision develops drainage -Chest tube site cover with a bandaid until healed -No driving while taking narcotics -Take stool softners with narcotics -Take stool softeners with narcotics -Take stool softners with narcotics -Take stool softners with narcotics -Take stool

5. You were admitted to the hospital with a bleed in your liver. This bleed was treated with a TIPS procedure, but it did not show any clear cause for this bleed. Your blood counts have been stable and you will need to follow-up with Dr. XXX on 3-19 for further evaluation of your liver disease. Please take all medications as prescribed. If you experience any fevers, chills, nausea, vomiting, abdominal pain, or any other concerning symptoms please call your doctor or go to the emergency room.

6. You were admitted to the hospital with shortness of breath and found to have a pneumonia. Your infection was treated with antibiotics and your breathing improved. You will need to continue to take Flagyl and Levaquin every day for at least one year. Please call Dr. XXX office or return to the emergency room if you experience any of the following: * You experience new chest pain, pressure, squeezing or tightness. * New or worsening cough or wheezing. * If you are vomiting and cannot keep in fluids or your medications until instructed otherwise by

7. You had a heart attack and a cardiac catheterization was performed. Your heart attack was complicated by a blockage in one of your coronary arteries. This blockages the artery that supplies blood to your heart. It is very important for you to take all medications as prescribed. Please follow-up with Dr. XXX and

Dr. XXX. Please call your doctor or return to the hospital if you experience chest pain, shortness of breath, fevers, chills, nausea, vomiting, diarrhea, or any other concerning symptoms.

8. You were admitted to the hospital with cholangitis. This was treated with antibiotics and you will need to complete 7 days of antibiotics for this. Please take all medications as prescribed. If you experience any fevers, chills, nausea, vomiting, chest pain, shortness of breath, or any other concerning symptoms please call your doctor or go to the emergency room.