MedTem2.0: Prompt-based Temporal Classification of Treatment Events from Discharge Summaries

Yang Cui, Lifeng Han, and **Goran Nenadic**
Department of Computer Science
The University of Manchester
Oxford Rd, Manchester M13 9PL, UK
yang.cui-2@student.manchester.ac.uk
lifeng.han, g.nenadic@manchester.ac.uk

Abstract

Discharge summaries are comprehensive medical records that encompass vital information about a patient's hospital stay. A crucial aspect of discharge summaries is the temporal information of treatments administered throughout the patient's illness. With an extensive volume of clinical documents, manually extracting and compiling a patient's medication list can be laborious, time-consuming, and susceptible to errors. The objective of this paper is to build upon the recent development on clinical NLP by temporally classifying treatments in clinical texts, specifically determining whether a treatment was administered between the time of admission and discharge from the hospital. State-of-the-art NLP methods including prompt-based learning on Generative Pre-trained Transformers (GPTs) models and fine-tuning on pre-trained language models (PLMs) such as BERT were used to classify temporal relations between treatments and hospitalisation periods in discharge summaries. Fine-tuning with the BERT model achieved an F1 score of 92.45% and a balanced accuracy of 77.56%, while prompt learning using the T5 model and mixed templates resulted in an F1 score of 90.89% and a balanced accuracy of 72.07%. Our codes and data are available at https://github.com/HECTA-UoM/MedTem.

1 Introduction

Clinical texts contain important temporal information, such as medication start and end dates, appointment dates, and diagnosis dates. Extracting this information can provide insights into a patient's medical history and allow doctors to make more informed decisions about their treatment. However, this process requires a significant amount of time and effort. To help healthcare professionals make informed decisions more efficiently, leading to better patient outcomes, we designed the project **MedTem**, medication and treatment event extraction and their relation modelling with temporal information. By using natural language processing (NLP) methods to extract temporal information from clinical texts, doctors can spend less time deciphering medical records and more time focusing on providing the best care possible to their patients. This study reports findings from MedTem2.0, a follow-up work from our previous investigation MedTem (Tu, 2022).

Clinical texts can be challenging to process due to their unstructured nature and the use of medical jargon. Thus, developing effective NLP techniques for extracting temporal information from clinical texts is crucial for improving healthcare outcomes. The primary goal of this work is to classify temporal information related to medication, surgeries, and other treatments within Electronic Health Records (EHRs) to determine if these treatments occurred during the hospitalisation period. This work aims to develop a system capable of classifying temporal information using prompt-based learning (PBL) from texts, which could aid healthcare professionals in understanding patients' medical histories and facilitate research in clinical text mining.

As an example, in Table 1, given the admission and discharge dates, we aim to determine if the *a left carotid endarterectomy* and *vein patch angioplasty* were used during the hospitalisation period. The note indicates that those treatments were administered on 3/3/92, which is during the admission and discharge dates, suggesting that it was used during hospitalisation. We assume that all treatment information is provided and only need to analyse the temporal information.

To the best of our knowledge, this is the first attempt at using prompt-based learning for the temporal classification of treatments in the clinical domain, with the following outcomes: 1) we established a high baseline score with 90.89% F1 measurement and 72.07% balanced accuracy by using prompt-based learning, demonstrating the

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics - Student Research Workshop, pages 160–183
July 10-12, 2023 ©2023 Association for Computational Linguistics

clinical free text		
Admission Date	**Discharge Date**	**Doctor's Note**
02/22/92	*03/08/92*	*She was, therefore, cleared for the operating room, and on 3/3/92, she underwent* ***a left carotid endarterectomy****, with continuous electroencephalogram monitoring and* ***vein patch angioplasty****, which was uneventful .*

Table 1: Task Example

effectiveness of the developed system for classifying temporal relationships between treatments and hospitalisation times; 2) we achieved improved performance using fine-tuning with the BERT model, resulting in a 92.45% F1 score and 77.56% balanced accuracy.

2 Methodologies

2.1 Task Overview

The pipeline shown in Figure 1 presents the methodology. The key approaches entail deriving gold labels from annotated datasets, following several pre-processing steps such as few-shot learning and sentence segmentation, among others. To evaluate the efficacy of prompt-based learning in temporally classifying treatment entities, two widely-adopted paradigms were used for comparison: pre-trained fine-tuning and prompt-based learning. Within these paradigms, three state-of-the-art pre-trained language models were used to perform the task: the Masked Language Model BERT, Seq2seq model T5 and Auto-regressive Language Model GPT-2 (Devlin et al., 2018; Raffel et al., 2020; Radford et al., 2019). All these models are based on Transformer structures but with different architecture/components, BERT for the encoder, GPT for the decoder, and T5 for both the encoder and decoder. We used BERT-base instead of BERT-large because the latter one costs too much power that the Colab platform we used could not afford.

2.2 Data Pre-processing

Step I: Generation of Gold Standard The i2b2 temporal relations corpus we used contains pre-existing layers of gold standard annotations, such as clinical concepts (problems, tests, treatments) and coreference relations (Uzuner et al., 2012, 2011), which can facilitate temporal reasoning.

In each discharge note, there are three types of annotations: events, temporal expressions, and temporal relations. Event annotations (EVENTs) encompass three distinct clinical concepts (i.e. PROBLEMs, TESTs, and TREATMENTs), clinical departments, EVIDENTIALs (words or phrases patients use to describe their symptoms), and OCCURRENCEs (other events, such as admission, that indicate the patient's timeline). Each EVENT possesses three attributes: TYPE, MODALITY, and POLARITY. For this specific task, we only need to identify the TYPE of EVENT as TREATMENT and OCCURRENCE among all the TYPE attributes (PROBLEM, TEST, TREATMENT, CLINICAL_DEPT, EVIDENTIAL, or OCCURRENCE). Figure 2 shows the discharge summary paragraph; the EVENTs in this record are shown in Table 2.

In clinical records, the temporal expression annotations use the TIMEX3 tag, which includes four categories: time, date, duration, and frequency. Each TIMEX3 value (VAL) is standardised to a unified format, such as time and date being represented as [YYYY-MM-DD]T[HH:MM]. Additionally, the MOD attribute indicates the characteristics of the temporal expression. Table 3 shows the TIMEX3 in the sample clinical record snippet. Once we have acquired all the EVENT and TIMEX3 information, we can map the temporal relations (TLINKs) between time and events, or between events themselves (Table 4). The TLINK categories include BEFORE, AFTER, BEGUN_BY, ENDED_BY, DURING, SIMULTANEOUS, OVERLAP, and BEFORE_OVERLAP.

Upon identifying all the treatment EVENTs and their relationships with admission and discharge times, we assign a label of "ON" to those entities where treatment occurs after or overlaps with the admission time and is also before or overlaps with the discharge time, indicating that the treatment was administered during hospitalisation. Conversely, we assign a label of "OFF" to the remaining treatments, signifying that they were not used during hospitalisation. Figure 3 illustrates the application of this rule-based approach for generating

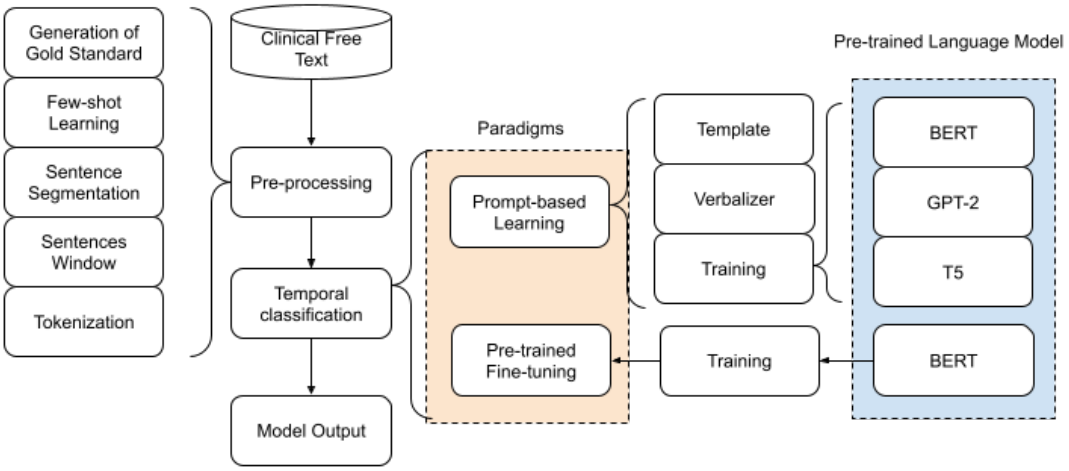

Figure 1: System Pipeline

Admission Date :
06/11/1991
Discharge Date :
06/22/1991
HISTORY OF PRESENT ILLNESS :
Patient is a 28 year old gravida IV , para 2 with metastatic cervical cancer admitted with a question of malignant pericardial effusion . Patient underwent a total abdominal hysterectomy in 02/90 for a 4x3.6x2 cm cervical mass felt to be a fibroid at Vanor .

Figure 2: Sample Clinical Record Snippet (Underscored: EVENTs, Italics: TIMEX3s)

the necessary gold labels. These gold labels comprise the document name, discharge note, treatment entity, and the label. In this study, the provided dataset consists of a training dataset and a testing dataset. After processing the data using the gold label generator as above, we obtained 3,075 ON-labelled training samples (indicating treatments used during hospital stays) and 762 OFF-labelled samples (indicating treatments not used during hospital stays). This results in an imbalanced label set on the dataset.

Step II: Few-shot Learning to Balance Labels To address the label-imbalance issue, we used a few-shot learning approach to create a balanced training dataset. This involved randomly selecting an equal number of samples from each label and combining them to form the few-shot training dataset.

Furthermore, most notes contain numerous abbreviations, such as "mcg subq q.d.", which stands for "micrograms subcutaneously once daily". However, since our objective is to analyse temporal information related to treatments, addressing dosage and frequency abbreviations is not necessary.

Step III: Sentence Segmentation Due to the nature of the dataset, which consists of clinical discharge notes, doctors frequently use brief sentences or even short phrases to describe various treatments, tests, or other patient-related information. This characteristic simplifies the process of *Sentence Segmentation*, which can be achieved by splitting the text based on newline characters ("/n") and periods (".") . The rationale behind sentence segmentation is to preserve and enhance the extraction of contextual information within the text, as distinct sentences often address different topics or aspects.

Step IV: Sentence Window An interesting aspect is that a single treatment may be mentioned multiple times in one clinical note, each referring to different events with distinct time sequences. Providing the entire text as input data would be imprecise and inaccurate. Additionally, clinical notes predominantly consist of factual statements and clinical declarations, with sentences generally

Event	Type	Modality	Polarity
[Admission]	OCCURRENCE	FACTUAL	POS
[Discharge]	OCCURRENCE	FACTUAL	POS
[gravida IV]	OCCURRENCE	FACTUAL	POS
[metastatic cervical cancer]	PROBLEM	FACTUAL	POS
[malignant pericardial effusion]	PROBLEM	POSSIBLE	POS
[a total abdominal hysterectomy]	TREATMENT	FACTUAL	POS
[a fibroid]	PROBLEM	POSSIBLE	POS
[Vanor]	CLINICAL_DEPT	FACTUAL	POS

Table 2: EVENT Annotation Examples

	document name	discharge note	treatment entity	label(1-ON,0-OFF)
0	472.xml.tlink	Admission Date : 2017-06-16 Discharge Date : 2...	specific interventions	1
1	626.xml.tlink	Admission Date : 06/25/1990 Discharge Date : 0...	crutches	1
2	167.xml.tlink	Admission Date : 2019-06-25 Discharge Date : 2...	extubated	1
3	236.xml.tlink	ADMISSION DATE : 08/15/1998 DISCHARGE DATE : 0...	a left internal mammary artery graft to the le...	0
4	387.xml.tlink	Admission Date : 2013-08-24 Discharge Date : 2...	monitored very closely	1

Figure 3: Example of Generated Gold Label

TIMEX3	Type	VAL	Mod
[06/11/1991]	DATE	1991-06-11	NA
[06/22/1991]	DATE	1991-06-22	NA

Table 3: TIMEX3 Annotation Examples

being independent. As a result, we used a **Sentence Window** approach to extract valuable information. For instance, if the target treatment entity is in the target sentence, and the sentence window size is set to 4, the model selects two sentences before and after the target sentence. The input data consists of the target sentence, its surrounding sentences, and the key temporal information of admission and discharge times, which appear at the beginning of every clinical note. Thus, this approach ensures that the model incorporates relevant temporal information and context.

Step V: Tokenization Tokenization is a crucial step in the natural language processing pipeline, wherein paragraphs are segmented into sentences, and sentences are further broken down into individual tokens or words (Koehn, 2009). This process enables the conversion of unstructured textual data into a structured, word-based data format, facilitating subsequent processing and analysis. By transforming unstructured data into structured data, we can represent textual information as vectors, and tokenization serves as the foundational step in this transformation.

In prompt-based learning, designing a template that includes an input sequence and prompting sentence is essential. However, creating a tokenizer for this purpose can be time-consuming and prone to errors. This is due to the presence of specific information, such as masked tokens or auto-generated tokens, embedded in the template, which requires careful handling during tokenization. Any mismatches in masked tokens can result in serious consequences. Furthermore, different PLMs may have distinct architectures, leading to varying tokenization strategies, necessitating consistency in context processing.

2.3 Prompt-based Learning vs Fine-Tuning

In conventional supervised learning for NLP, the objective is to predict an **output y** based on an **input x** utilising the model $P(y|x;\theta)$ (Manning and Schutze, 1999). In classification tasks, **y** denotes the class label corresponding to **input x**. To train the model's parameters θ, a dataset consisting of input-output pairs is required for predicting this conditional probability (Goodfellow et al., 2016). However, obtaining adequately annotated (labelled) data for certain domains can be challenging. Prompt learning methods address this limitation by learning a language model (LM) that estimates the probability $P(x;\theta)$ of the text **x** itself. Consequently, this probability is used to predict **y**, thereby bypassing the need for extensive labelled datasets (Liu et al., 2023; Ding et al., 2021). There

From extent	Type	To extent
[Admission]	SIMULTANEOUS	[06/11/1991]
[Discharge]	SIMULTANEOUS	[06/22/1991]
[gravida IV]	BEFORE	[SECTIME: 06/11/1991]
[para 2]	BEFORE	[SECTIME: 06/11/1991]
[para 2]	OVERLAP	[gravida IV]
[...]	...	[...]
[a total abdominal hysterectomy]	BEFORE	[SECTIME: 06/11/1991]

Table 4: TLINK Annotation Examples

will be three main steps of doing that including prompt construction, answer selection, and answer mapping (refer to Appendix C.1).

We used **OpenPrompt**, a toolkit for implementing prompt learning in downstream tasks (Ding et al., 2021). It offers a function for loading PLMs, tokenizers, and other required configurations, which function accommodates the choice of PLMs (MLM, LM, and Seq2seq) and conducts tokenization accordingly. Designed with encapsulated data processing APIs, users can apply a human-readable style to create templates and conveniently operate on both the input and template simultaneously.

To identify the optimal prompt format for this task, we examine various components in the prompt-based construction. We explore different large langauge model (LLM) architectures, and adjust the template's structure and format within the prompt construction. We modify the answer's form in answer selection to correspond with the chosen template.

In this context, we will first define the templates and verbalizers used within the framework and our experiments. We refer to the traditional prompt-based learning approach that uses human designed templates and verbalizers as *manual templates* and *manual verbalizers* respectively. This strategy was initially introduced as Pattern-Exploiting Training (PET) by Schick and Schütze (Schick and Schütze, 2020).

Manual Template Creating manual components in prompt learning can be quite intricate, as slight modifications to the tokens can lead to significant changes in performance. Domain expertise is typically required for effective engineering of these components. Examples of manual template can be a statement or question-answering format.

The **Soft Template** (Example 1) approach shares similarities with the manual method but replaces fixed manual components with soft (trainable) tokens or embeddings, denoted as <[soft]>. Combining some fixed manual components with soft tokens leads to the **Mixed Template** approach (Example 2), which uses both fixed and trainable elements in the template construction.

Listing 1: Example of Soft Template

```
text = '<[clinical_record]> <[soft]>
    <[treatment]> <[soft]> <[soft]>
    <[mask]> <[soft]>.'
```

Listing 2: Example of Mixed Template

```
text = '<[clinical_record]> Question:
    <[treatment]> <[soft]> <[soft]>
    <[soft]> <[soft]> <[soft]>. Is it
    correct? <[mask]>'
```

Leveraging the T5 model's encoder-decoder architecture, we can generate variable-length output sequences based on the input sequence. With this advantage, the PLM can generate part of the prompt within the manual template. Choosing to sacrifice human interpretability, one can create soft prompt components instead. A typical mixed template takes the form $x_0 = [P_0, P_1, \ldots, P_j], x, [P_{j+1}, P_{j+2}, \ldots, P_k], [\text{MASK}]$, where for $i \in 0, 1, \ldots, k$, P_i represents the token of the template.

Verbalizer The verbalizer functions as a mechanism that maps single or multiple distinct tokens to well-defined class labels. The embedding or hidden state associated with the $<$ [MASK] $>$ position, generated by the PLM, is subsequently processed through a standard language model head or classifier. This step computes the probabilities connected to the class label tokens derived from the verbalizer. In this task, a **Manual Verbalizer** was used, which entailed manually constructing a list of answers. These answers can be either token-based or span-based, depending on the specific template

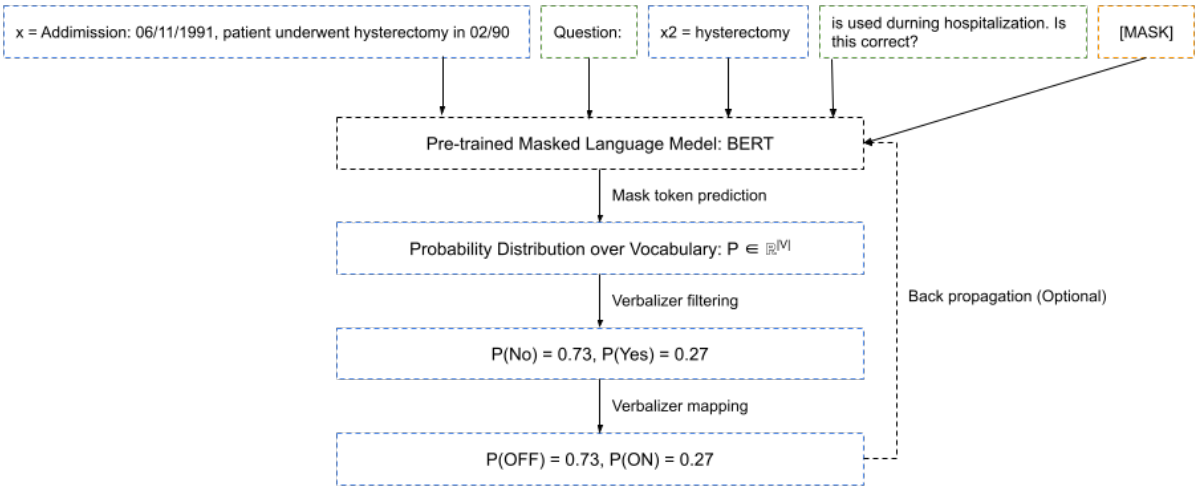

Figure 4: Illustration of Manual Template and Verbalizer in Prompt Learning

used.

In a similar fashion to the soft template, a **Soft Verbalizer** can be conceptualised as replaced words in the verbalizer with trainable embeddings for each class. As a result, when useing a soft verbalizer, there is no necessary to establish a mapping from vocabulary V to class C, as the trainable vectors lack semantic meaning.

2.4 Traditional Fine-tuning

In traditional fine-tuning methodology, the downstream task uses a multilayer perceptron (MLP) denoted as $f_{MLP}(\cdot)$. This MLP takes the pooled sequence embedding generated by the PLM as input and delivers an n-dimensional vector, where n represents the numeral of classes (Kowsari et al., 2019). Given an input text x, the PLM first processes the raw input to obtain the m-dimensional embedding for each token. Next, a pooling process, such as the mean, is involved in all the token's embeddings to generate a single sequence embedding $h(x)$ with the same m-dimensional size. The sentence embedding $h(x)$ is then fed into the MLP block through a typical feed-forward process to obtain the likelihood distribution across n classes using a softmax operator.

Figure 4, 5, and 6 illustrate the examples of PBL and PLM fine-tuning on our task, adapted from (Taylor et al., 2022).

2.5 Evaluation Methods

We take the label "ON" as the positive class and label "OFF" as the negative class. In addition to F1 score, we used balanced accuracy as a performance measure for our model, which calculates the average recall across all classes. The decision to use balanced accuracy instead of overall accuracy stems from the imbalanced distribution of class labels in the test dataset, with 3164 instances of label "ON" and 921 instances of the label "OFF". Balanced accuracy considers the performance of the model on each class individually, thus avoiding potential misinterpretations that can arise from using overall accuracy when one class is substantially more prevalent than the other.

3 Experimental Work

3.1 Dataset

In this project, we use electronic health records (EHRs) from the National NLP Clinical Challenges (n2c2, formerly known as i2b2) dataset, which is part of an annual challenge workshop [1]. We primarily focus on the 2012 n2c2 challenge (Sun et al., 2013b), which is centred around temporal relations. The dataset consists of 310 patient clinical history records and hospital course sections from Partners Healthcare and Beth Israel Deaconess Medical Center, along with clinical events, time expressions, and temporal relationship annotations (Sun et al., 2013a). For ethical reasons and to protect patient privacy, the data has been de-identified and abstracted, including the obfuscation or alteration of names, addresses, and other personal information. Additionally, accurate time information has been randomly shifted.

[1] https://n2c2.dbmi.hms.harvard.edu/about-n2c2

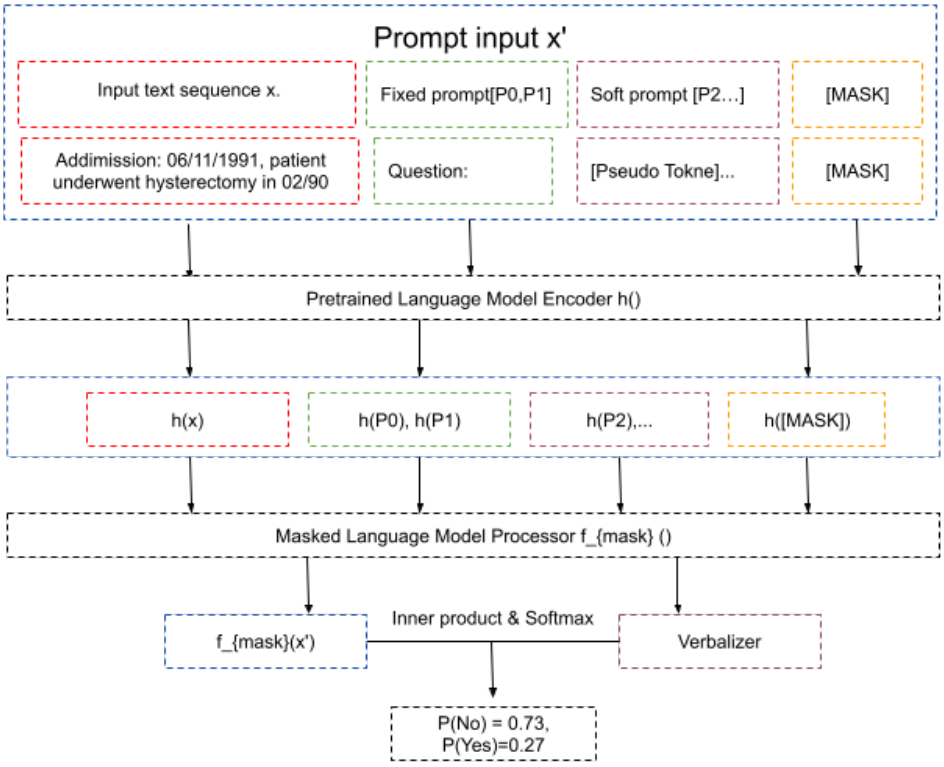

Figure 5: Illustration of Mixed Template and Verbalizer in Prompt Learning.

3.2 Output from Prompt-based Learning

We adopt a systematic approach to optimise the performance of different PLMs. Initially, we use various PLMs by the full training dataset, basic manual templates, and verbalizers, while fixing the sentence window for input text and adjusting the learning rate to identify the optimal performance for each model. Comparing the results, we will determine the best-performing PLM at this stage.

Next, with the best PLM and fixed sentence window, we will train the model using the full dataset while varying templates and verbalizers to identify the most effective template. Furthermore, we will maintain the best PLM and template while altering the sentence window to assess the impact of input text on performance.

Upon completing the hyperparameter selection for prompt-based learning, we will obtain the best-performing model. Finally, we will use few-shot learning to compare this model with the fine-tuning paradigm.

3.2.1 Different Language Models

To evaluate the performance of various models, we use a combination of admission and discharge information along with three sentences that include the target sentence and the sentences immediately preceding and following it, where the target sentence contains the target treatment entity. Moreover, we use manual templates and verbalizers, with the template following a question-answering format. The verbalizer is set to a collection of words, specifically "Yes", "No". The entire training process spans 5 epochs.

	L.R.	F1.on	B.Accy.
BERT	1E-4	87.29	50
	2E-4	**90.75**	**69.72**
	5E-6	90.14	69.57
GPT-2	6E-5	90.57	70.24
	2E-5	**90.79**	**71.19**
	5E-6	90.28	65.58
T5	6E-5	90.69	70.43
	4E-5	**91.24**	**71.43**
	2E-5	90.12	68.36

Table 5: Performance of Different PLM. L.R.: learning rate; F1.on: score of ON class; B.A.:Balanced Accuracy

Upon adjusting the learning rate for the various PLMs, several examples of results were obtained in table 5. The bold font indicates the highest score for each PLM. In fact, there was not a big difference between them. T5 is 1.71 and 0.24 higher than BERT and GPT-2 under balanced accuracy

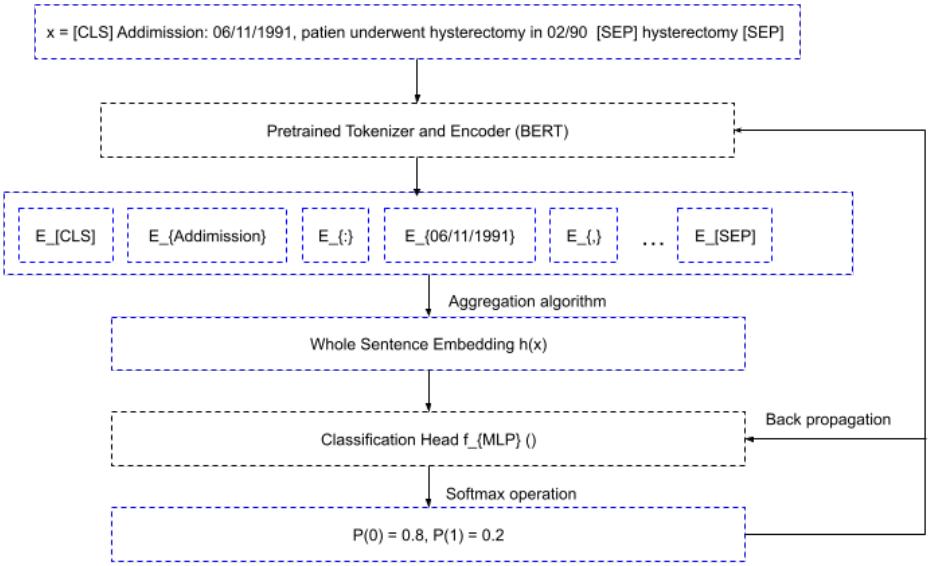

Figure 6: Illustration of Conventional Fine-tuning Method. (Here [CLS] and [SEP] tokens are special tokens for BERT-based models that are added to the beginning and end of sequences.)

respectively and held a 0.49 and 0.45 advantage in F1 score.

During the training process, we observed that all the results demonstrated a higher recall than precision, indicating that the model correctly identifies most of the true positive cases (with few false negatives). This situation can be attributed to the training data having a significantly larger number of positive examples compared to negative ones, which is also reflected in the testing dataset. Additionally, when examining the negative class accuracy, the models only achieve approximately 50%. This suggests that they are not proficient in detecting negative classes. However, when using a balanced training dataset, the negative class accuracy increases to 61%.

3.2.2 Different Prompt Learning Setups

In order to assess the effectiveness of different combinations of templates and verbalizers, we used a variety of templates in conjunction with both manual and soft verbalizers. For the manual template, we used a question-answering format, combined with a yes, no manual verbalizer and a soft verbalizer. Additionally, the soft template used Example 1 for prompting, with fixed and predefined positions and lengths for the soft tokens, and was combined with the same manual and soft verbalizers as the manual template. For the mixed template, we used Example 2 along with the same verbalizers as before. During the comparison of different prompt engineering approaches, we also experimented with various text lengths for each template category.

Template	Verbalizer	F1.on	B.Accy.
Manual	Manual	**91.24**	**71.43**
	Soft	90.85	70.52
Soft	Manual	**90.68**	68.33
	Soft	89.8	**72.48**
Mixed	Manual	**90.89**	**72.07**
	Soft	90.7	69.01

Table 6: Performance of Different Prompt Learning. F1.on: score of ON class; B.Accy.: Balanced Accuracy

The evaluation results presented in Table 6 reveal that the (Manual, Manual) combination, with the format (Template, Verbalizer), achieves the highest F1 score of 91.24. This indicates its strong capability to classify "ON" class samples. Additionally, the (Soft, Soft) setup demonstrates the best balanced accuracy of 72.42, which is more suitable when the "OFF" class is as important as the positive class. We list error analysis examples and comparisons of different input text in Appendix (F). The (Mixed, Manual) configuration showcases comparatively good results for both evaluation metrics and will be used as the standard for the next section of comparisons.

3.3 PBL vs Traditional Fine-Tuning

The Hyperparameters-optimised outputs from PBL and traditional fine-tuning are displayed in Table

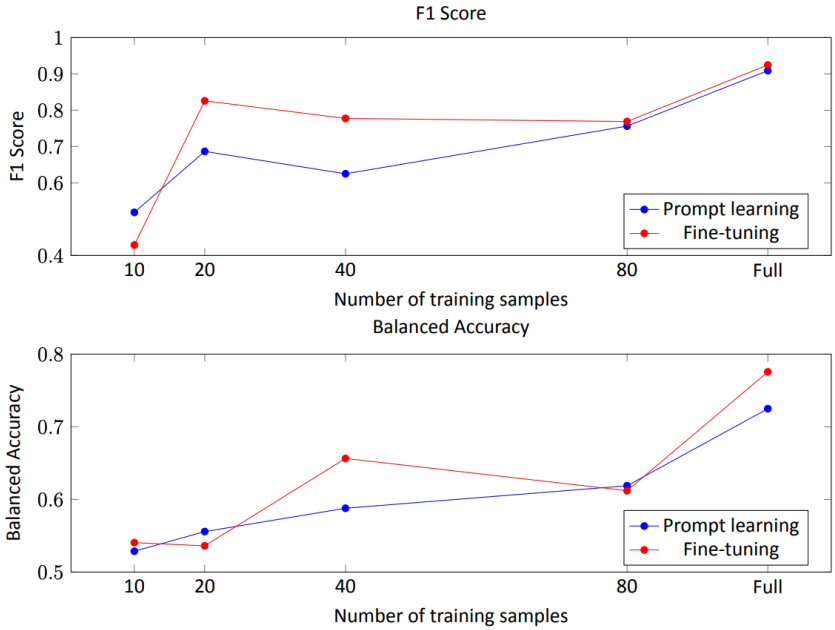

Figure 7: Balanced Accuracy and F1 Score for Prompt Learning and Traditional Fine-tuning Frameworks Across the Temporal Classification Task. ("Full" refers to a full training dataset size.)

7 and Figure 7, with the hyper-parameter sets in Appendix (G).

Paradigm	F1 score	B.Accy.
Traditional fine-tuning	92.45	77.56
Prompt-based learning	91.79	75.08

Table 7: Hyperparameter Optimised Model for Temporal Classification. B.Accy.: Balanced accuracy

4 Related Work

Early research in temporal relation classification focused on extracting and representing temporal information from clinical text. Hripcsak et al. (2002) proposed a method for representing clinical events and their temporal relationships using an interval-based temporal model, laying the groundwork for understanding temporal dependencies in clinical text.

Inspired by the TimeML standard (Pustejovsky et al., 2003) for annotating temporal expressions and relations in text, the THYME (Temporal Histories of Your Medical Events) annotation guidelines were developed by Styler IV et al. (2014) to adapt TimeML for clinical narratives. These guidelines provided a foundation for temporal relation classification research in the clinical domain. However, achieving temporal understanding in clinical narratives is challenging due to the complexity of determining implicit temporal relations, handling temporal granularity, and dealing with diverse temporal expressions.

5 Conclusion and Future Work

In this work, two state-of-the-art approaches were developed to classify the relative timing of treatments in hospital discharge summaries, focusing on determining whether a treatment was administered during hospitalisation or not. These approaches used cutting-edge pre-trained language models, BERT, GPT-2, and T5, in conjunction with prompt-based learning and fine-tuning paradigms. Both approaches achieved F1 scores of 91.79% and 92.45%, and balanced accuracy of 75.08% and 77.56%, respectively, on the n2c2 2012 Temporal Relations dataset. The primary challenge was accurately classifying the "OFF" class due to data imbalance and complex semantic meanings that made it difficult for the models to make correct decisions. Future work could investigate the impact of fixed tokens on mixed template performance or the role of longer sequence lengths in soft templates for improved understanding. Additionally, a more comprehensive comparison of prompt learning and traditional fine-tuning can be conducted across various clinical domain tasks, using frozen PLMs in conjunction with few-shot learning methods.

Limitations

There are several limitations to the experiments conducted in this project that should be acknowledged:

- Selection of the best pre-trained language model (PLM) for prompt-based learning: The evaluation method used to compare the performance of BERT, GPT-2, and T5 in the context of manual templates and manual verbalizers may not be entirely accurate. The performance of these models did not show significant differences, making it difficult to determine the best model for prompt-based learning. Furthermore, other domain-specific PLMs, such as Bio-BERT, which may be better suited for handling clinical data, were not considered in this project.

- Limited exploration of templates: The experiments utilized a limited number of templates, particularly for soft and mixed templates. These templates were primarily based on prompts derived from manual templates. Further experimentation is needed to explore different patterns, such as varying the position and length of soft token sequences or using soft tokens in mixed templates to replace manual tokens (e.g., "Question:").

- Comparison with frozen PLMs: The experiments did not include a comparison between fine-tuned and frozen PLMs, as done in Taylor's study (Taylor et al., 2022). This comparison could provide valuable insights into the performance trade-offs between these two approaches.

- Addressing the effects of imbalanced datasets, several strategies have gained popularity. 1) Re-sampling techniques, for example, Monte Carlo Simulation Analysis, can be used to balance class distribution by oversampling the minority class, undersampling the majority class, or the combination of these two (Gladkoff et al., 2021). 2) Data augmentation techniques, such as the use of Generative Adversarial Networks (GANs), can generate new examples for the minority class by applying transformations to existing data. 3) Furthermore, machine learning approaches like bagging and bootstrapping can reduce variances by implementing a "voting system" that enables models to make better decisions.

- Finally, it would be advantageous to develop a post-processing step that generates a table displaying all treatments along with their corresponding temporal information. This would create an end-to-end system that physicians could use as a practical tool.

Future research should address these limitations by exploring a broader range of PLMs, templates, and experimental setups to provide a more comprehensive understanding of the performance characteristics of prompt-based learning methods in the clinical domain. Application to some more powerful computational resources will also extend this work.

Ethical Discussion

The n2b2 (formerly i2b2) 2012 Temporal Relations dataset was used for the development of the approach in this project. This dataset comprises patient-level data in the form of discharge summaries. These documents have been de-identified in accordance with the Health Insurance Portability and Accountability Act of 1996 privacy regulations by the organizers of the n2c2 2012 NLP challenge (Act, 1996). The dataset was obtained with permission for academic use only after signing a Data Use and Confidentiality Agreement with the n2c2 National Center for Biomedical Computing. So no further ethical approval forms were required to gain access to the dataset.

Acknowledgements

We thank the reviewers for their precious comments on making our paper better. The work was partially supported by Grant EP/V047949/1 "Integrating hospital outpatient letters into the healthcare data space" (funder: UKRI/EPSRC).

References

Accountability Act. 1996. Health insurance portability and accountability act of 1996. *Public law*, 104:191.

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

William A Chren. 1998. One-hot residue coding for low delay-power product cmos design. *IEEE Transactions on circuits and systems II: Analog and Digital Signal Processing*, 45(3):303–313.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Richard S Dick, Elaine B Steen, Don E Detmer, et al. 1997. The computer-based patient record: an essential technology for health care.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Serge Gladkoff, Irina Sorokina, Lifeng Han, and Alexandra Alekseeva. 2021. Measuring uncertainty in translation quality evaluation (tqe). *arXiv preprint arXiv:2111.07699*.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Aaron Li-Feng Han, Xiaodong Zeng, Derek F Wong, and Lidia S Chao. 2015. Chinese named entity recognition with graph-based semi-supervised learning model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 15–20.

Lifeng Han, Gleb Erofeev, Irina Sorokina, Serge Gladkoff, and Goran Nenadic. 2022. Examining large pre-trained language models for machine translation: What you don't know about it. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 908–919, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jerry R Hobbs, Douglas Appelt, David Is Bear, and Mabry Tyson. 1997. Extracting information from natural-language text. *Finite-state language processing*, page 383.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

George Hripcsak, John HM Austin, Philip O Alderson, and Carol Friedman. 2002. Use of natural language processing to translate clinical information from a database of 889,921 chest radiographic reports. *Radiology*, 224(1):157–163.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Erwan Moreau, Ashjan Alsulaimani, Alfredo Maldonado, Lifeng Han, Carl Vogel, and Koel Dutta Chowdhury. 2018. Semantic reranking of crf label sequences for verbal multiword expression identification.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Chaitanya Shivade, Preethi Raghavan, Eric Fosler-Lussier, Peter J Embi, Noemie Elhadad, Stephen B Johnson, and Albert M Lai. 2014. A review of approaches to identifying patient phenotype cohorts using electronic health records. *Journal of the American Medical Informatics Association*, 21(2):221–230.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C De Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the association for computational linguistics*, 2:143–154.

Weiyi Sun, Anna Rumshisky, and Özlem Uzuner. 2013a. Annotating temporal information in clinical narratives. *Journal of biomedical informatics*, 46:S5–S12.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013b. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Niall Taylor, Yi Zhang, Dan Joyce, Alejo Nevado-Holgado, and Andrey Kormilitzin. 2022. Clinical prompt learning with frozen language models. *arXiv preprint arXiv:2205.05535*.

Hangyu Tu. 2022. *Extraction of Temporal Information from Clinical Free Text*. MSc. Thesis, The University of Manchester.

Özlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett R South. 2012. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association*, 19(5):786–791.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.

Yuping Wu, Lifeng Han, Valerio Antonini, and Goran Nenadic. 2022. On cross-domain pre-trained language models for clinical text mining: How do they perform on data-constrained fine-tuning? *arXiv preprint arXiv:2210.12770*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

A Background and More Literature

In this section, We introduce some key concepts and then explore the methods and techniques used in clinical text mining, with a particular focus on temporal classification (Tu, 2022). We will begin by examining the fundamentals of clinical text mining and its applications in healthcare, followed by an in-depth discussion on the challenges associated with temporal event extraction and classification. Next, we will delve into the recent developments in prompt-based learning and its potential to revolutionise the field of clinical text mining, including its ability to handle diverse NLP tasks with a unified framework.

Our objective is to provide a comprehensive overview of the current landscape of clinical text mining in the context of temporal classification, emphasising the emerging role of prompt-based learning and its potential to drive further innovation and improvement in healthcare research and practice.

A.1 Temporal Classification from EHRs

Electronic Health Records (EHRs) have evolved from the concept of Computer Patient Records (CPR) proposed by the Institute of Medicine in 1991 (Dick et al., 1997). Temporal relation classification of clinical events is crucial in understanding the chronological sequence and dependencies of events within electronic health records (EHRs). Extracting and analysing temporal information from EHRs can enhance our comprehension of disease progression, treatment efficacy, and patient risk factors, ultimately leading to improved healthcare outcomes.

A.2 Related NLP Applications

Rule-based methods in NLP involve using a predefined set of linguistic rules, patterns, or heuristics to process and analyse text. These rules are often developed by domain experts or linguists, reflecting the inherent structure and patterns present in the language. For instance, in Named Entity Recognition (NER) tasks, rule-based approaches can identify proper names, organisations, and locations using regular expressions (Hobbs et al., 1997), which often target words starting with a capital letter. And Chapman (Chapman et al., 2001) proposes a rule-based algorithm designed for detecting negated concepts in clinical text. The advantages of rule-based methods include their speed and the lack of requirement for extensive computational resources.

However, rule-based methods have many limitations such as low recall (Riloff, 1996). In certain domains, only experts can develop effective rules. Changes in the data source might render existing rules ineffective. Moreover, rule-based methods can be challenging to apply in temporal classification tasks involving free text, due to the absence of a standard format and the diverse and varied language expressions.

Statistical sequence models are particularly well-suited for language processing tasks due to their ability to handle variable-length sequences, such as sentences. CRFs have been widely used in sequence labelling tasks such as part-of-speech tagging, information extraction, and named entity recognition (NER) (Moreau et al., 2018; Han et al., 2015). In clinical domain, Shivade et al. (2014) used a combination of HMMs and CRFs for clinical named entity recognition (NER) tasks. They used these methods to identify medical concepts such as medications, dosages, and durations from clinical text. Their results demonstrated that HMMs and CRFs could effectively recognize medical concepts, with CRFs outperforming HMMs in most cases.

Before the advent of word embeddings, researchers primarily used statistical techniques like one-hot encoding (Chren, 1998) and TF-IDF (Aizawa, 2003) to represent words based on their frequency of occurrence in the text. This led to the creation of large, sparse vectors for word representation. The introduction of Word2Vec (Goldberg and Levy, 2014) offered several advantages, including lower-dimensional, dense, and continuous vectors that captured semantic similarity between words based on their co-occurrence with other words.

With the development of hardware capabilities, large neural networks have become feasible, which allows the exploration of deep learning architectures that can discover hidden features and automatically learn representations from the input in an end-to-end structure, mostly via the encoder-decoder style (Goodfellow et al., 2016). Collobert and Weston (2008) first introduced temporal convolutional neural networks (CNNs) for named entity recognition (NER) tasks. To model long sequences, Hochreiter and Schmidhuber (1997) proposed the long short-term memory (LSTM) model based on the architecture of recurrent neural networks (RNNs), addressing the challenge of capturing long-distance historical information and mitigating the vanishing gradient problem faced by RNNs.

Tu (2022) used a combination of Bidirectional Long Short-Term Memory (BiLSTM) and Conditional Random Fields (CRF) to perform Named Entity Recognition (NER) tasks on a clinical dataset. The model achieved a weighted average accuracy of 0.98 and a macro-averaging score of 0.69. Additionally, they explored the use of a Convolutional Neural Network (CNN) with BiLSTM, resulting in improved performance compared to the BiLSTM+CRF model. This hybrid model demonstrated a precision of 85.67%, recall of 87.83%,

and an F1-score of 88.17%.

A.3 Recent Large Language Models

A.3.1 Pre-trained Language Models

The development of the Transformer architecture by Vaswani et al. (2017) brought NLP to a new stage with its self-attention mechanism, which enhances the model's ability to capture long-range dependencies among words in the input sequence. Pre-trained language models like BERT, GPT, and T5, which are based on the Transformer architecture, have achieved state-of-the-art performance on numerous tasks. These models learn contextualised word representations, different from traditional word representations (e.g., Word2Vec, GloVe), which map words to fixed-length vectors and assume words in similar contexts have similar meanings. In contrast, pre-trained models learn context-dependent representations, capturing contextual information more effectively (Qiu et al., 2020). This process allows models to better "understand" language, context, and words.

A.3.2 Fine-tuning Paradigm

Fine-tuning has been the traditional approach for adopting pre-trained language models (PLMs) to specific tasks. This is usually done by task-specific layers or heads on top of the pre-trained model and adjusting the model's weights through back-propagation (Wu et al., 2022). It has achieved state-of-the-art results in many NLP tasks, such as sentiment analysis (Socher et al., 2013), named entity recognition (Wadden et al., 2019) and machine translation (Vaswani et al., 2018; Han et al., 2022). However, it requires lots of training data, which may not be available in certain scenarios, and to fine-tuning a model can be computationally expensive.

Fine-tuning From 2017 to 2019, there was a paradigm shift in NLP model learning, with researchers moving away from fully supervised methods and increasingly adopting the pre-training and fine-tuning paradigm. This approach uses a fixed architecture pre-trained language model (PLM) to predict the probability of observed textual data. The PLM is adapted to different downstream tasks by fine-tuning additional parameters using objective functions specific to each task. For instance, Zhang et al. (Zhang et al., 2020) introduced a loss function for predicting salient sentences, and when combined with PLMs and fine-tuning, it resulted in state-of-the-art performance on various popular datasets and tasks (Devlin et al., 2018). However, the fine-tuning approach is most suitable when large-scale text data is available for optimising the objective function, which is not always feasible in certain domains. In the case of clinical records, data privacy issues and the need for clinical experts to annotate data for training make it difficult to produce large open clinical datasets. For example, BERT models trained on non-medical text tend to perform poorly when applied to medical domain tasks (Lee et al., 2020; Wu et al., 2022). Additionally, each specific task requires its own fine-tuning process, and as the NLP field continues to increase model sizes to improve performance (e.g., Microsoft's Megatron (Shoeybi et al., 2019) with 530 billion parameters), full or partial fine-tuning of these massive models demands considerable computational, financial resources, and time (Han et al., 2022). These concerns have led to the emergence of a new paradigm called prompt-based learning, which aims to achieve strong performance across a wide range of applications without the need for extensive fine-tuning.

A.3.3 Few-shot Learning

Few-shot learning is an area of machine learning that focuses on training models to recognize or generalize new concepts with very limited labelled examples. This approach aims to alleviate the need for large amounts of labelled data, which can be costly and time-consuming to obtain. The few-shot learning problem is typically framed in terms of episodes, where each episode consists of a small support set and a query set. The support set contains a few labelled examples of each class, while the query set comprises unlabelled examples from the same classes. The goal is to learn a model that can accurately classify the query set instances based on the limited information provided in the support set. Finn et al. (Finn et al., 2017) proposed MAML, a meta-learning algorithm that learns an optimal initialisation of model parameters, enabling rapid adaptation to new tasks with few gradient updates.

A.3.4 Prompt-based Learning Paradigm

Prompt-based learning is a recent paradigm in NLP that leverages pre-trained language models (PLMs) like GPT-3 (Brown et al., 2020) to perform various tasks without the need for fine-tuning. This approach involves using carefully designed prompts

or templates that guide the PLM to generate desired outputs based on the input context. Moreover, this approach is especially useful in situations with limited task-specific training data, as it does not require retraining the entire model, however, crafting effective prompts for specific tasks can be challenging and may require manual engineering or iterative search procedures. It gives me the inspiration to construct a fine prompt learning and challenge with more traditional fine-tuning methods.

Prompt-based learning emerged with the advent of models like T5 and GPT-3, as researchers discovered that pre-trained language models (PLMs) could be effectively guided by textual prompts in low-data scenarios. The T5 model innovation suggested that PLMs possess strong language understanding capabilities, and by providing appropriate instructions or prompts, they can adapt to various tasks (Liu et al., 2023). This approach, dubbed "pretrain, prompt, and predict" or prompt-based learning, revolves around prompt engineering, which tailors prompts to suit different downstream tasks.

For instance, given the sentence “Patient is complaining of a stomachache” an emotion recognition task can be framed by adding a prompt like “Patient felt so ___”, prompting the language model to fill in the blank with an emotion-laden word. Similarly, for translation tasks, a prompt like “English: Patient is complaining of a stomachache, Chinese: ___” can be used. ChatGPT’s ability to understand and answer questions in natural language can also be considered a form of prompting, influencing the quality of responses.

OpenPrompt Ding et al. (Ding et al., 2021) introduced a unified, user-friendly toolkit called OpenPrompt to facilitate prompt-based learning with PLMs. OpenPrompt’s modular and combinable research-friendly framework enables the integration of various tasks, prompting techniques, and PLMs while accommodating different template formats, verbalizer formats, and initialization strategies. Taylor et al. (Taylor et al., 2022) applied prompt learning to the clinical domain using frozen language models by using the OpenPrompt framework. Their research compared prompt-based learning and fine-tuning in clinical classification tasks, finding that prompt learning typically matched traditional fine-tuning performance on full datasets and outperformed it in few-shot settings which means prompt learning is more adopted training with smaller datasets. Additionally, prompt learning excelled when working with frozen PLMs, showcasing its potential with fewer trainable parameters.

A.4 Summary

In this section, we delve into prior work concerning temporal classification and examine the fundamental concepts and methods used in constructing our model. Given the absence of previous studies utilising prompt-based learning for temporal classification in the clinical domain, there are no established guidelines or approaches for this task. In the following section, we will provide a detailed explanation of the methodology used to develop our model, outlining each step of the process.

B On Dataset Used

Figure 8 presents the format used for training the model, where the discharge note column contains clinical text information, and the treatment entity column comprises treatment entities. The training dataset consists of 3,836 samples, with 3,075 having the label "ON" (treatment used during hospitalisation) and 762 having the label "OFF" (treatment not used during hospitalisation), resulting in an imbalanced distribution with label "ON" being four times more prevalent than label "OFF".

To gain a deeper understanding of the dataset, various statistical analyses were conducted. As depicted in Figure 9, the word count distribution for clinical notes, excluding the first five lines, is displayed. The first five lines of each note, which contain admission and discharge dates, are not considered beneficial for statistical analysis. The figure illustrates that most sentences have fewer than 20 words, and no sentences in the training dataset exceed 80 words. Based on this information, the maximum input sequence length can be determined.

C Learning Models

C.0.1 State-of-the-Art PLMs

A pre-trained language model is a neural network model that has already been trained on a large corpus of text data before being fine-tuned for specific tasks (Han et al., 2022). These models are designed to learn the structure and nuances of a language by predicting the next word in a sentence or reconstructing a sentence with masked words. By learning the complex patterns and relationships

	document name	discharge note	treatment entity	label
0	422.xml.tlink	Admission Date : 2017–07–12 Discharge Date : 2...	oxycodone	1
1	631.xml.tlink	ADMISSION DATE : 10/10/97 DISCHARGE DATE : 10/...	diabetes control	1
2	272.xml.tlink	Admission Date : 2011–09–24 Discharge Date : 2...	extubated	1
3	96.xml.tlink	Admission Date : 11/17/2003 Discharge Date : 1...	pain control	0
4	422.xml.tlink	Admission Date : 2017–07–12 Discharge Date : 2...	a standing IVF order	1
...	...	...	...	...
3832	422.xml.tlink	Admission Date : 2017–07–12 Discharge Date : 2...	repletion	1
3833	736.xml.tlink	Admission Date : 03/17/1998 Discharge Date : 0...	Gentamicin	1
3834	577.xml.tlink	Admission Date : 2009–06–23 Discharge Date : 2...	levofloxacin	1
3835	177.xml.tlink	Admission Date : 2012–11–21 Discharge Date : 2...	CellCept	1
3836	26.xml.tlink	Admission Date : 12/11/2005 Discharge Date : 1...	oral analgesics	0

Figure 8: Training Dataset Format

within the language, these models can generate contextually relevant embeddings or representations of words and phrases.

Masked Language Model: BERT BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model developed by Google researchers in 2018 (Devlin et al., 2018). As its name suggests, it uses the encoder architecture from the Transformer model but with a deeper structure, as shown in Figure 14. The BERT-base language model comprises 12 encoder blocks, which is twice the size of a standard Transformer Encoder.

In contrast to OpenAI's GPT (Generative Pre-trained Transformer), BERT uses a bidirectional Transformer block connection layer (Figure 15), allowing it to access information from both preceding and following content, while GPT only considers the preceding content during training. Although the concept of "bi-directionality" is not new. For example, ELMo uses two individual objective functions $P(w_i|w_1, ...w_{i-1})$,$P(w_i|w_{i+1}, ..., w_n)$ to train the language model. However, BERT uses a single objective function:

$$P(w_i|w_1, ...w_{i-1}, w_{i+1}, ..., w_n) \quad (1)$$

to train the language model, integrating both preceding and following context.

The Masked Language Model (MLM) serves as one of BERT's pre-training tasks, wherein it randomly masks certain words in a sentence with the [mask] token. By leveraging the bidirectional Encoder Representations, BERT predicts the masked words based on both preceding and following context, resulting in a more comprehensive understanding of word meanings. Additionally, the Next Sentence Prediction (NSP) pre-training task trains the model to discern the relationship between sentences by determining whether sentence B follows sentence A in the original text (Devlin et al., 2018).

The input for BERT consists of Token Embeddings, Segment Embeddings, and Position Embeddings, as illustrated in Figure 16. Each input sentence is treated as a sequence of tokens, with every sequence starting with a special classification token, [CLS]. BERT uses another special token, [SEP], to separate sentences and assigns segment embeddings to each token to indicate whether it belongs to sentence A or B. This enables BERT to handle various downstream tasks, such as separating question and answer sequences (Devlin et al., 2018). By incorporating position embeddings, the model generates distinct word vector outputs for the same word based on its contextual environment, thereby enhancing the model's accuracy.

Fine-tuning enables BERT to accommodate various downstream tasks by adjusting the corresponding inputs and outputs (Figure 17). The same pre-trained model parameters are used to initialise models for different downstream tasks, and all parameters are fine-tuned end-to-end to adapt the model to the specific task. In comparison to pre-training, fine-tuning is relatively cost-effective and computationally efficient.

Auto-regressive Language Model: GPT-2 The Generative Pre-trained Transformer 2 (GPT-2) is

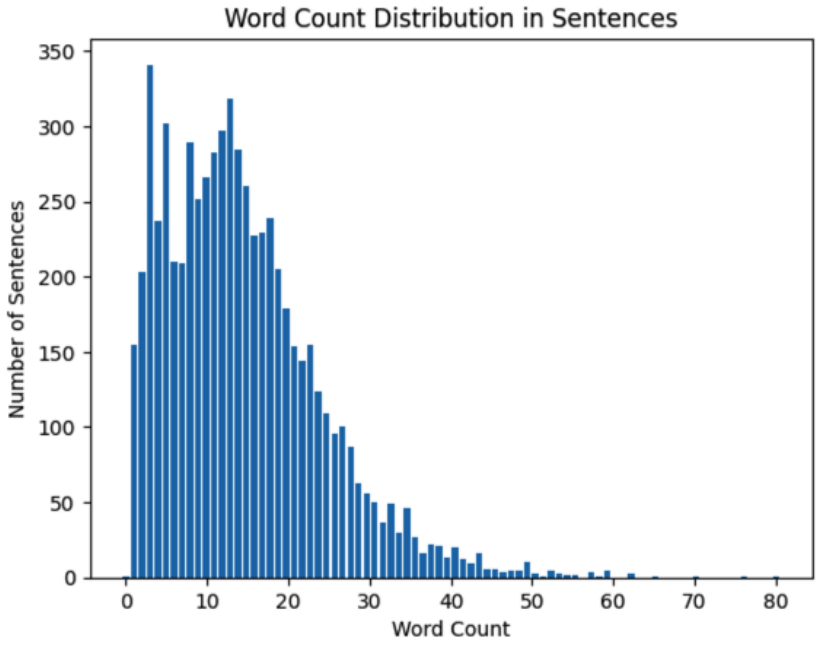

Figure 9: Word Count Distribution in Sentences

an advanced language model introduced by OpenAI in 2019, building upon the foundation of the original GPT (Radford et al., 2019). GPT-2 uses a transformer-based decoder architecture with multi-layer, multi-head self-attention mechanisms, as shown in Figure 18. This design allows GPT-2 to generate sequences of arbitrary length, making it particularly adept at producing highly coherent and contextually relevant text, often used for question-answering and summarization tasks.

GPT-2 differs from BERT in several ways. As an autoregressive model, GPT-2 predicts one token at a time, using previously generated tokens as context for subsequent predictions based on the equation of $p(s_{s-k}, ..., s_n|s_1, ..., s_{n-k-1})$. This process continues until the desired output length is achieved or an end-of-sequence token is generated. By modelling a sequence of outputs as a product of conditional probabilities, GPT-2 leverages the natural sequence of symbols inherent in language. Unlike BERT's bidirectional approach, GPT-2 uses masked self-attention, processing input sequences in a unidirectional manner, resulting in more contextually relevant text generation (Radford et al., 2018).

One innovative aspect of GPT-2 is its ability to perform supervised learning tasks using an unsupervised pre-training model. While traditional supervised learning aims to estimate $p(output|input)$, GPT-2 seeks to model $p(output|input, task)$, allowing for a more generalised model across various tasks. This approach has been used in multitask and meta-learning settings. For instance, a translation training example could be presented as a sequence (translate to French, English text, French text), enabling the model to understand the translation task and the relationship between input and output (McCann et al., 2018).

Seq2Seq: T5 T5, an abbreviation for Text-To-Text Transfer Transformer, proposes the idea that fine-tuning models for specific tasks may no longer be necessary (Raffel et al., 2020). Instead, a large pre-trained model can be used for any task, with the main focus on adapting the task into appropriate textual inputs and outputs (Raffel et al., 2020). For example, refer to Figure 19, in translation tasks, inputting "translate English to German" followed by a [sequence] results in the model producing the translated [sequence]. Similarly, for summarization tasks, inputting "summarise" along with the [sequence] generates a summary of the [sequence]. This method establishes a unified Text-to-Text format for NLP tasks, expressed as $[Prefix + SequenceA] \rightarrow [SequenceB]$, enabling the use of the same model, loss function, training process, and decoding process across all NLP tasks with different prefix information.

To accomplish this, a powerful language model that genuinely comprehends language is required.

The Google team developed a strategy to determine the optimal model architecture and parameters, ultimately creating a robust baseline. First, they examined three popular model architectures. The encoder-decoder Transformer (Vaswani et al., 2017), also known as a seq2seq model (left panel of Figure 20), comprises two layer stacks: the encoder processes the input sequence and encodes each token, while the decoder generates a new output sequence with each token based on the decoding input and previous output sequences. The language model architecture (middle one of Figure 20), akin to the decoder in an encoder-decoder Transformer, predicts output at each time-step based on previous time-step predictions, with GPT-2 being a typical Example The Prefix LM (language model) incorporates fully-visible masking applied to the prefix, rendering the architecture more effective for a wide range of text-to-text tasks shown in the left panel of Figure 20. Following experimentation, the Google team determined that the encoder-decoder architecture is the most suitable for the text-to-text framework, thus adopting it for T5 (Raffel et al., 2020).

Subsequently, they used masked language modelling (BERT-style) as an unsupervised pre-training method. Similar to BERT, but using masks to replace spans surrounding the original masked tokens as corruption strategies, with a 15% corruption rate and 3 corrupted span length according to experimental results.

After utilising multi-task learning to train with the C4 (Colossal Clean Crawled Corpus) dataset, which comprises hundreds of gigabytes of clean English text extracted from the web, the Google team acquired the best pre-trained language model, T5, among numerous combinations of model architecture, training methods, and various parameters.

C.1 Prompt-Based Learning

Prompt Construction The first step involves creating a *prompting function* $f_{prompt}(\cdot)$, which transforms the input **x** into a prompted $x' = f_{prompt}(x)$ (Liu et al., 2023). This function entails two stages: (1) Designing a *template*, a string containing an *input slot* [X] for the input **x** and an *answer slot* [Z] for the generated answer, which is mapped to the output **y**. (2) Filling the slot [X] with the input **x**.

In the case of temporal classification for treatment "a total abdominal hysterectomy," the template could be structured as "[Input] Here is the clinical record, treatment a total abdominal hysterectomy [Z] during the hospitalisation." Additionally, templates can be categorised based on the position of the empty slot, such as close (prompts with slots in the middle of the text) or prefix prompts (slots appearing before the entity) z (Liu et al., 2023).

Answer Selection Subsequently, the language model (LM) is used to identify the highest-probability text $\hat{z}$. Liu et al. (Liu et al., 2023) characterises Z as a collection of acceptable values for z, indicating that the LM determines the most probable answer z from the set of answers Z. This process is also referred to as answer engineering or verbalisation (we will consistently use the terms verbalizer[2] and verbalization).

The verbalizer can be regarded as a mapping between one or many distinct tokens and unique class labels. The embedding generated at the <[MASK]> position by using PLM is through a large language model head or classifier, and prediction of the tokens from verbalizer class labeled are obtained. In the previous temporal classification example, Z = "is", "is not" corresponds to class labels Y = ON, OFF.

The function $f_{fill}(x', z)$ fills the slot [Z] in prompt x' with a potential answer z. Lastly, the probability of the corresponding filled prompt is calculated using a PLM $P(\cdot; \theta)$, as shown in Eq. 2:

$$\hat{z} = search_{z \in Z} P(f_{fill}(x', z); \theta) \qquad (2)$$

The search function could use argmax for the highest-scoring output or sampling to randomly generate outputs according to the LM's probability distribution (Liu et al., 2023).

Answer Mapping The final step maps the highest-scoring answer $\hat{z}$ to the highest-scored output $\hat{y}$. While this step might not be crucial in binary classification, it is necessary for tasks like translation or sentiment analysis with multiple words (e.g., "good", "wonderful", "perfect") mapped to the same class (e.g., "++"). Thus, a mapping process between the answer and the true output value is required (Ding et al., 2021).

D Parameters and Settings

The code below shows how to load the PLM of T5 and tokenizer in OpenPrompt: " plm, tokenizer, model_config, WrapperClass = load_plm ("t5", "t5-base") "

[2]

E More Discussion on PLM Outputs

The dataset we used is derived from clinical notes, implying that in real life, there are indeed more positive labels than negative ones. In some cases, having a high recall may be more important than having high precision. For instance, in medical diagnosis, it could be crucial to identify all patients with a specific disease (high recall) to ensure they receive appropriate treatment, even if some healthy patients are misclassified as having the disease (low precision). It is unclear whether recall is more important than precision in the context of temporal information of treatment. However, doctors can adjust the model's preference based on their specific situations.

It is not surprising that T5 outperforms the other models in the comparison. Firstly, T5 is the most recent model among the three and has been extensively tested by Raffel et al. (Raffel et al., 2020) to evaluate its advantages and disadvantages relative to the other architectures. Their results suggest that T5's encoder-decoder architecture performs better than BERT and GPT-2 in certain tasks. our experiment also demonstrates that T5 has a slight advantage over BERT and, more notably, GPT-2, which exhibit comparable performance.

Secondly, although it is not universally true that "bigger models are better" in the NLP field, OpenAI has made significant strides in showcasing the effectiveness of larger models in recent years. The development of models such as GPT-2, GPT-3, and, more recently, Megatron-Turing, has demonstrated that models with more parameters can improve performance on a variety of natural language processing tasks, as illustrated in Figure 10. In our experiment, we used *bert-base-uncased*, which has 110M parameters, and the *gpt-2* model with 117M parameters. However, *T5-base* model has 220M parameters, twice as many as *bert-base-uncased*. Therefore, T5 is the best model for temporal classification in the clinical domain when compared to the other two models.

F PBL with Differed Input Text

One intuitive method to create prompts is to manually craft templates based on human understanding. For instance, we can create a cloze-style manual template using Code 3, where the $<$ [MASK] $>$ token appears in the middle of the template. According to the code example, the $<$ [MASK] $>$ token can be filled with "is" or "is not".

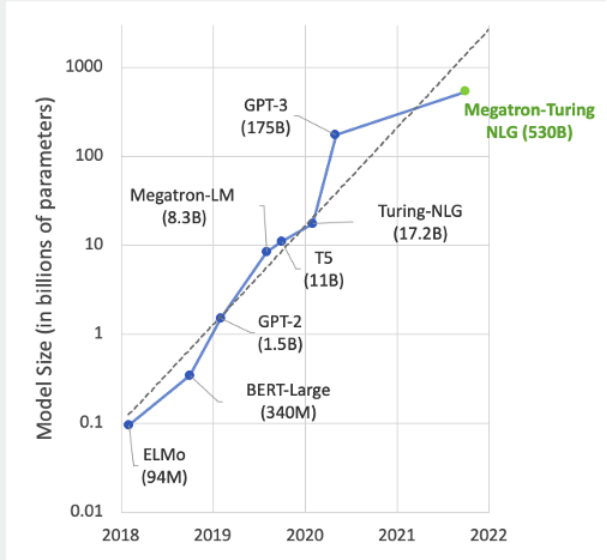

Figure 10: Development of Model in NLP Recently (from COMP34312 week5 slides)

Listing 3: Example of cloze manual template

```
text = '<[clinical_record]> In this
    paragraph of the note,
    <[treatment]> <[mask]> used between
    admission and discharge time.'
```

Another popular manual template approach is the question prompt shown in Figure 4, in which the $<$ [MASK] $>$ token is placed at the end. In this template, a discriminative statement or question is presented, such as "Question: this treatment was used between admission and discharge time. Is it correct?" Combined with the clinical context input, the PLM decides whether the statement is correct. Therefore, the possible answers for $<$ [MASK] $>$ can be "yes" or "no".

Listing 4: Example of manual template with question

```
text = '<[clinical_record]> Question:
    <[treatment]> were used between
    admission and discharge time. Is it
    correct? <[mask]>'
```

In the previous work, Gu et al. (2021) report a mixed template tokens and soft tokens in some yields better than manual and soft template, and Taylor et al. (Taylor et al., 2022) propose that soft template working with soft verbalizer perform the best on ICD9 Triage task in clinical domain.

During manual template engineering, some interesting findings were made. Initially, the manual template was designed as "<clinical note>. Question: <treatment> was used during hospitalisation. Is it correct?". While this appeared sufficient, upon

analysing errors in the testing data, a particular example revealed that the treatment in question was used during the patient's last hospitalisation but not the current one. Consequently, the template was modified to specify "between admission and discharge time", which better emphasised the temporal aspect.

Furthermore, certain errors were identified due to complex language logic. During this period, chatGPT was a popular topic in NLP domain, and the GPT-3.5 model demonstrated remarkable question-answering abilities. we input a template (shown in Figure 11) to the chatGPT and the chatGPT model provided an incorrect response, despite giving an accurate explanation, which is not self coherent. This indicates that GPT-3.5 and the T5 model have difficulty capturing information from words such as "attempt" and "but".

By comparing the results of the cloze (Example 3) and question prompt (Example 4) in the manual template, it was found that the question prompt performed better. This suggests that the PLM may be more proficient in judging discriminative statements or providing answers after processing the entire input sentence. The (Mixed, Manual) pair also performed well, possibly because the generated soft tokens, based on the input sentence and fixed template tokens, provided guidance for the model to better select an answer from the set of possible responses.

F.0.1 Different Input Text

Experiments of Different Input Text In this experiment, the input length for clinical records was modified by controlling the number of sentences in the input text using a sentence window size, as well as the number of sentences before and after the target sentence.

Discussion and Summary of Different Input Text The results displayed in Table.8 indicate that as the number of input sentences increases, both the F1 score and balanced accuracy improve. However, when the input text becomes too long, such as the entire clinical text, the performance slightly declines. It was found that a window size of 6, comprising 3 sentences before the target sentence, the target sentence itself, and 2 sentences after, yielded the best F1 score and balanced accuracy of 91.79 and 75.08, respectively.

G PBL vs Traditional Fine-tuning

G.0.1 Summary of Prompt-based Learning Evaluation

In conclusion, the prompt-based learning paradigm experiments led to the establishment of a benchmark for the best-performing prompt model. The hyperparameter details are provided in Table.9. In the following section, this model will be compared to the traditional fine-tuning paradigm using a few-shot learning approach.

G.1 Prompt Learning versus Traditional Fine-tuning

In this section, we present a benchmark comparison between Prompt-based Learning (PBL) and Traditional Fine-tuning (FT) under few-shot settings. Table 10 displays the selected hyperparameters for Fine-tuning. we chose to focus on a mixed template approach, which combines a manually designed template for the task with soft and trainable tokens. Since few-shot scenarios can introduce bias and variance that significantly affect performance, we aggregated the results from 10 trials and averaged them, providing a more accurate assessment.

The results (Table 7 and Figure 7) indicate that in the temporal classification task, the traditional fine-tuning model outperforms the prompt learning model. The prompt learning model performs better than the fine-tuning model only when the training set size is 10 in terms of F1 score, and when the dataset size is 20, the prompt learning model's balanced accuracy is slightly higher. This finding is consistent with Taylor's work (Taylor et al., 2022), which showed that prompt learning did not outperform fine-tuning in various clinical domain classification tasks, such as ICD-9 50, ICD-9 Triage, and In-hospital mortality. However, in specific classification tasks under Frozen PLM conditions, prompt learning exhibited better performance. In this context, "frozen" refers to the absence of updates to the model's weights and parameters during the fine-tuning process.

These results were surprising, as prompt learning has been frequently reported to be more effective in few-shot settings in numerous publications. There could be several reasons for this discrepancy. First, the soft and trainable tokens in the mixed template were not trained using a separate optimizer, which may have resulted in suboptimal tokens for the given task. Second, the benchmark for prompt learning might not be accurate due to computa-

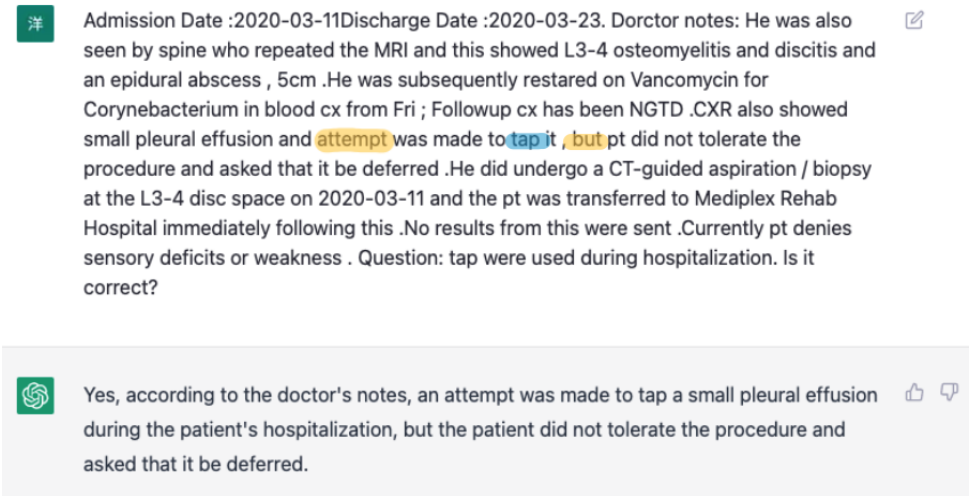

Figure 11: Example of error analysis with ChatGPT.("tap" is the treatment)

Sentences window size	(sentences before, sentences after)	F1 score of ON class	B.Accy.
1	(0,0)	88.13	64.24
2	(0,1)	89.58	65.89
3	(1,1)	90.89	72.07
4	(1,2)	91.60	73.29
5	(2,2)	91.93	73.00
6	(3,2)	**91.79**	**75.08**
7	(3,3)	91.44	71.71
Whole text		84.86	63.95

Table 8: Performance of Different Input Text (B.Accy.: Balanced Accuracy)

Parameter	**Value**
PLM	T5
learning rate	4E-5
batch size	4
epochs	5
optimizer	AdamW
template	mixed template
verbalizer	manual verbalizer
input sentences window size	6 (3,2)

Table 9: Hyperparameter Selection for Prompt-based Learning

Parameter	**Value**
PLM	BERT
learning rate	2E-5
batch size	4
epochs	5
optimizer	AdamW
input sentences window size	6 (3,2)

Table 10: Hyperparameter Selection for Fine-tuning

tional resource and time limitations. For instance, the best PLM and learning rate were determined based on a manual template and manual verbalizer, but these selections may not be ideal for mixed and soft templates. Third, potential biases in the training process could have impacted the results, as no validation set was used for prompt learning, possibly preventing the selection of the best model during training. Furthermore, averaging the results of 10 trials might not provide a sufficiently accurate assessment, and more trials could be necessary. Fourth, in a few-shot learning scenario, useing a language model pre-trained on medication and clinical domain data might be more beneficial for clinical classification tasks. Finally, prompt-based learning is a relatively new paradigm with much-untapped potential, whereas traditional fine-tuning has a well-developed training and tuning process.

Upon examining errors from the test dataset of prompt-based learning, specifically for both "ON"

```
  "label": "OFF",
  "meta": "727.xml.tlink",
  "clinical_record": "Admission Date: 2014-03-31 Discharge Date :
2014-04-01 . Dorctor notes: No maternal fever . No prolonged rupture of
membranes . Clear amniotic fluid . Anesthesia by epidural . Vaginal
delivery . Apgars were 8 and 9 . ",
  "treatment": "Anesthesia",
```


Figure 12: Example of an error in OFF class


```
  "label": "ON",
  "meta": "208.xml.tlink",
  "clinical record": "Admission Date : 2018-05-26 Discharge Date :
2018-05-31 . Dorctor notes: WBC s since admission were as high as
14,000 but normalized . She also had 2 echocardiograms which revealed
persistent pericardial effusions . She has been gently diuresed but has
worsening ARF . Her O2 requirement has increased despite diuresis . She
denies any CP / cough / fever , abdominal pain / diarrhea , black or
bloody stools or headache . Her urine output decreased to nearly zero .
",
  "treatment": "diuresis",
```


Figure 13: Example of an error in ON class

and "OFF" classes as shown in Figures 12 and 13, it becomes evident that determining whether a treatment was administered during hospitalisation can be challenging. The input content often lacks sufficient temporal information to clearly indicate the treatment status. Furthermore, there are instances of ambiguity in the dataset annotations, which complicates the classification task. The sentence tense and specific temporal expressions might be the only cues for understanding the event timeline, even for human readers, without considering the broader context of the document. It is also worth noting that discharge summaries are typically prepared at the end of a patient's hospital stay, and as such, they do not describe the hospitalisation period as the present. These observations highlight the complexities involved in classifying temporal relationships in clinical texts and the need for further improvements in methods to effectively address such challenges.

H Learning Structures

Figure 21 illustrates the general architecture of OpenPrompt, which allows for modifications to the PLM-related class (purple block) and the prompt-related class (blue block).

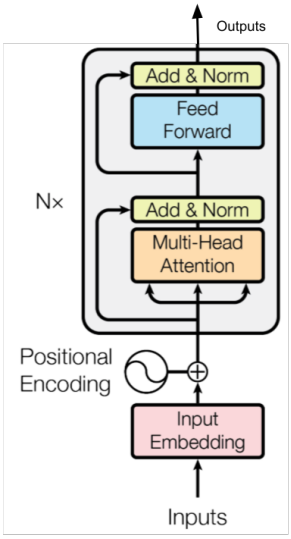

Figure 14: BERT Architecture (Vaswani et al., 2017)

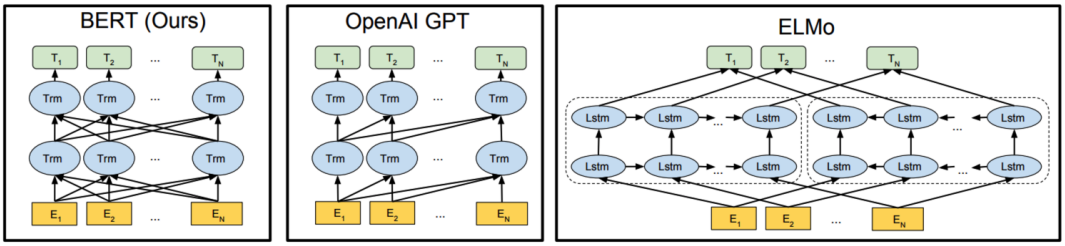

Figure 15: Differences in Rre-training Model Architectures (Devlin et al., 2018)

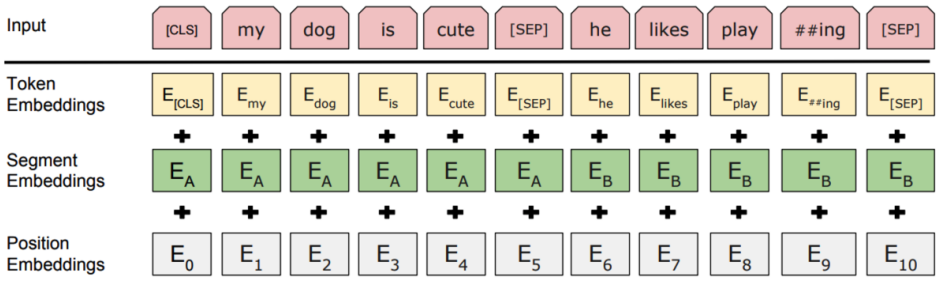

Figure 16: BERT Input Representation (Devlin et al., 2018)

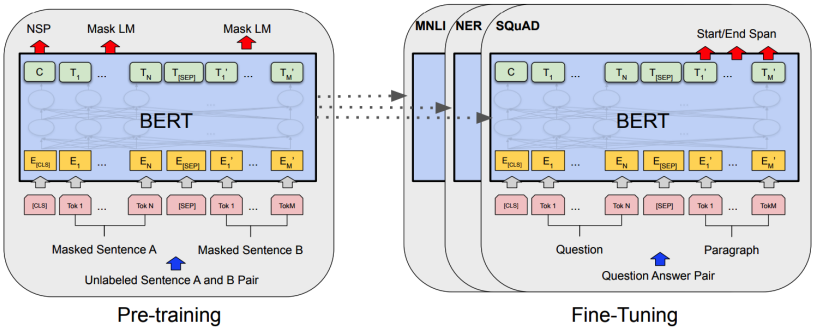

Figure 17: Overall Pre-training and Fine-tuning Procedures for BERT (Devlin et al., 2018)

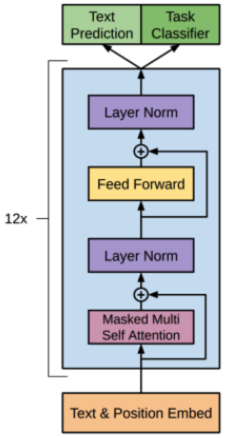

Figure 18: Architecture of GPT2 (Radford et al., 2018)

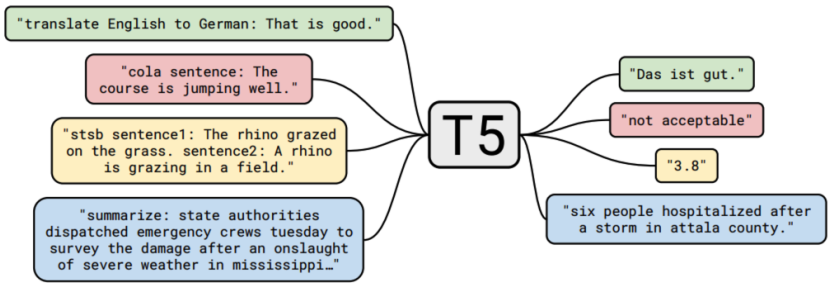

Figure 19: Text-to-text Framework (Raffel et al., 2020)

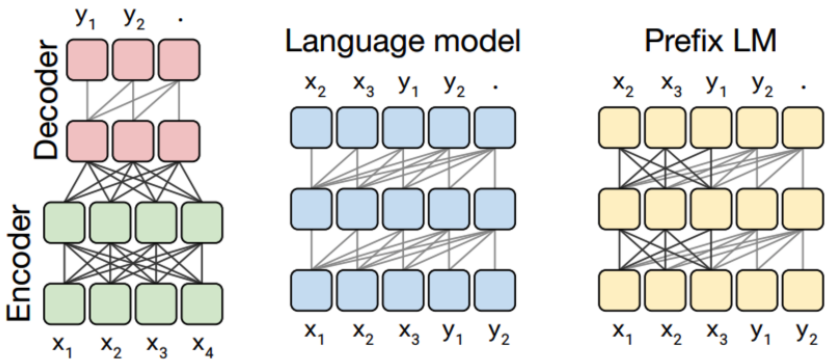

Figure 20: Different Transformer Architecture (Raffel et al., 2020)

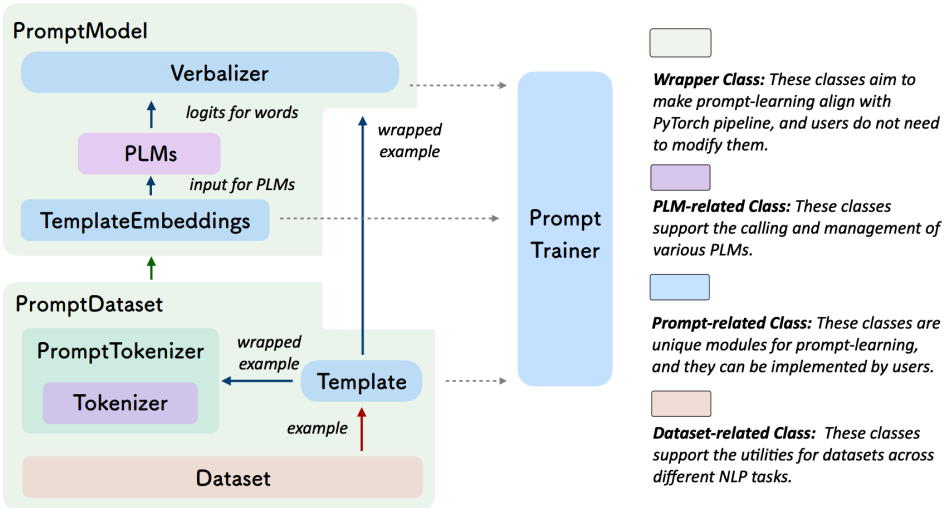

Figure 21: OpenPrompt Overall Architecutre (Ding et al., 2021)