Imperial College London

INTERIM REPORT

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

Integrating Concept Bottleneck Models and Large Language Models for Enhanced Patient Insight

Author:
Anish Narain

Supervisor:
Dr. Sonali Parbhoo

May 3, 2024

Contents

Chapter 1

Project Specification

1.1 Motivation

The digitisation of medical data offers an excellent opportunity to apply AI technologies like Machine Learning (ML) and Natural Language Processing (NLP) to improve healthcare delivery. For example, the Medical Information Mart for Intensive Care (MIMIC) database has been an invaluable tool for ML research. The database contains health-related information about patients who were admitted to the intensive care units (ICUs) of the Beth Israel Deaconess Medical Center [1]. Researchers have utilised this database to train models on clinical observations and patient-specific information for predicting timely and useful treatments for ICU patients.

In January 2023, a collection of free-text notes was added to the MIMIC-IV database. These are notes written by healthcare providers working in the ICU and it is an important source of information for understanding a patient's clinical course [2]. Large Language Models (LLMs) have shown promising results in extracting useful insights from unstructured data, like clinical notes, and could provide an efficient method to process the MIMIC-IV notes. This project aims to use the additional insights extracted from the unstructured notes by an LLM to improve the performance of a model which has been trained on the structured data available in MIMIC-IV.

1.2 Project Work

Implementing ML models on healthcare data comes with its own challenges. Firstly, medical data often contain confounding factors which cause a model to form an incorrect interpretation of the relationships in the data. Secondly, state-of-the-art models tend to be end-to-end: they go directly from raw input to target output which makes it difficult for medical professionals to interpret the underlying reasoning of the models. Concept bottleneck models (CBMs) offer a remedy to these problems by incorporating natural language "concepts" in their training. Medical professionals can understand and alter these concepts. By specifying desired concepts, they can also try to avoid some of the incorrect relationships that the model can form.

In this project, the aim is to combine the capabilities of CBMs and LLMs to produce useful insights about patients in the MIMIC database for downstream tasks. There are two ways to implement this combination.

1. Start with a concept bottleneck model with known concepts from the structured data in MIMIC-IV. Then, use an LLM on the MIMIC-IV clinical notes and highlight key phrases. The phrases can be augmented as concepts on top of the original concept bottleneck model and then the model can be trained.
2. Train an LLM on MIMIC-IV and then in the final or second last layer, rather than having a feed-forward or NLP-based architecture, insert a concept layer.

The project consists of several stages. Firstly, relevant literature on CBMs, LLMs for medical information extraction, and ML research on the MIMIC database will be reviewed. Although considerable material has already been explored, additional papers will continue to be read to inform design choices needed to meet the project's aim. Concurrently, a basic large language model and concept bottleneck model will be trained to understand the fundamentals of developing these tools. Then these models will be trained using MIMIC-IV data. A dataset of ICU patients will be extracted from the MIMIC-IV database, formatted to serial daily analyses of relevant vital signs, lab values and interventions. Following recognised medical definitions, labels of clinical concepts (gold standard labels) will be created to train the concept bottleneck model. In parallel, another set of clinical labels will be derived from MIMIC-IV clinical notes using a Large Language Model. Finally, having implemented both models and reviewed relevant literature, an informed decision will be made on how to combine the two models and their performance in downstream prediction tasks will be evaluated.

1.3 Project Deliverables

1. Gold standard labels produced from a CBM trained on structured data from the MIMIC database.
2. Clinical labels generated from MIMIC-IV clinical notes by an LLM.
3. Comparison and analysis of the quality and overlap between the labels generated by the LLM and gold standard labels.
4. Integration of the two models and an evaluation of this new model's performance.

Chapter 2

Background

2.1 Interpretability

Interpretability in machine learning is the ability of a system to explain or present its decisions in understandable terms to humans. In [3], the authors discuss when a ML system requires interpretability, and their reasoning emphasises the importance of interpretability for ML models operating in the medical domain. Firstly, and most importantly, models need to interpretable to ensure safety. Interpretability allows medical professionals to understand and verify a system's decisions and reasoning which is important when stakes are high and consequence of errors can be severe. The paper also considers ethical considerations. In healthcare, it is important to guard against certain kinds of discrimination. But without interpretability, the concept of fairness is too abstract to incorporate into the system. The authors also discuss the role of interpretability for scientific understanding and avoiding mismatched objectives between user and model.

Many state-of-art models, like deep neural networks, are inherently black-box. They consist of multiple layers and complex structures that transform raw input data into target outputs without revealing the decision-making process. This lack of interpretability makes them difficult to adopt for healthcare. However, work has been done to meet this demand for transparency, with concept bottleneck models being a notable example.

2.2 Concept Bottleneck Models

Supervised learning is a type of machine learning where the model is trained using data that is labelled. This means the data contains examples of both inputs and correct outputs [4]. The model learns from these examples to become better at predicting the desired output when provided new data. Concept bottleneck models are a specific type of supervised learning ML model which not only train on raw inputs and target outputs, but also pre-defined human-understandable concepts. The approach involves mapping

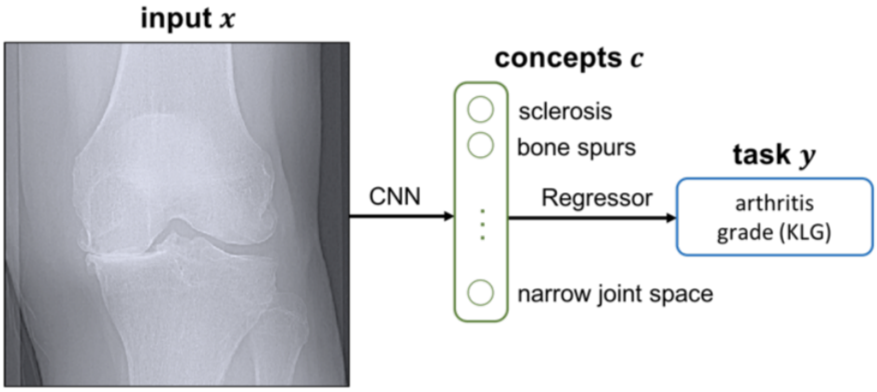


Figure 2.1: Illustration from [6]. The concept bottleneck model first predicts "an intermediate set of human-specified concepts c, then uses c to predict final output y"

raw inputs x to concepts c and then mapping c to target labels y, [5].

The approach of models initially predicting concepts and then using those concepts to make further predictions was not originally proposed by [6]. However, this paper's formulation of the CBM architecture, demonstration of strong task accuracy and subsequent discussion of future work has inspired a wave of papers exploring the nuances of concept bottleneck models. Hence this paper serves as a strong basis for understanding CBMs.

In the paper, the authors utilise an example of a radiologist using a machine learning model to examine the knee x-ray of a patient for arthritis. A traditional model would take in the raw pixels input x and produce the target output y, which is the severity of arthritis. Concept bottleneck models, on the other hand, first predict "an intermediate set of concepts c like joint space narrowing and bone spurs." These concepts c would then be used to predict target y. Figure 2.1 from the paper illustrates the process.

By providing users with such concepts, they get an insight into the model's reasoning meaning it would be easier for healthcare providers, who are using CBMs, to understand the rationale behind clinical predictions. In [7], researchers created a concept bottleneck model which predicted how long, after administration, it would take vasopressors (blood medication) to start having effects on a patient. They presented the concepts generated by the model to an expert and reported that the concepts allowed for meaningful clinical insights about the prediction. The paper stated that even for concepts that did not show immediate medical significance, having this concept framework made it much easier to evaluate the model's decision making process.

Furthermore, [6] underscores the interpretability of CBMs because they allow interventions (Figure 2.2). A CBM gets trained on data points (x, c, y). Then at test time, given an input x, it would predict concepts $\hat{c}$ and use those concepts to predict the target label $\hat{y}$. This offers the opportunity for users to edit $\hat{c}$ and propagate those changes to $\hat{y}$. Going back to the radiologist examining the knee x-ray example, if let's say the radiologist recognises the model is misidentifying bone spurs, they can update

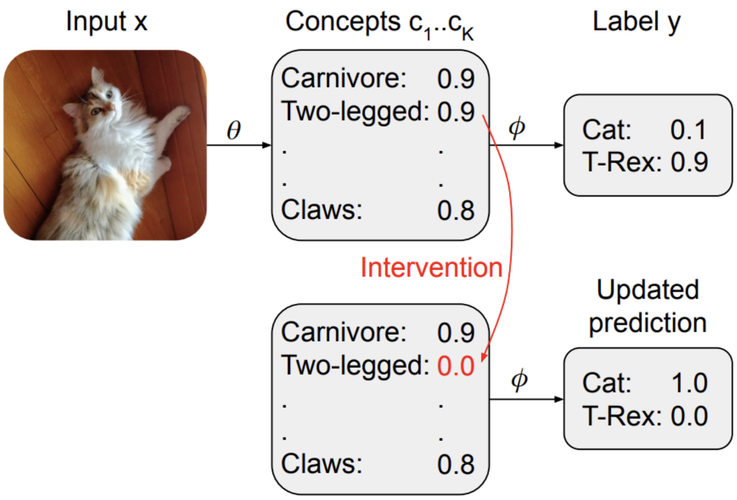


Figure 2.2: Illustration from [8] which explains: "when the model mispredicts a concept, one can intervene by setting the concept probability to 0 or 1 and obtain an updated prediction".

the model's prediction by changing $\hat{c}$. Testing this on actual applications allowed the paper to conclude that partially correcting concept mistakes at test time improved the accuracy of their CBM significantly compared to the standard baseline model.

2.3 CBM Training Methods

Concept bottleneck models can be formulated in the following manner [5]:

1. The training data points are $\{(x^{(i)}, y^{(i)}, c^{(i)})\}_{i=1}^{n}$. In this case, the input is $x \in \mathbb{R}^d$, the target label is $y \in \mathbb{R}$, and $c \in \mathbb{R}^k$ is a vector of k concepts.

2. Concept bottleneck models are of form $f(g(x))$, where g maps an input x into the concept space and f maps concepts into a final prediction.

3. At actual test time (execution time) CBMs produce prediction $\hat{y}$ where $\hat{y} = f(g(x))$. The test-time concepts are $\hat{c}$ where $\hat{c} = g(x)$.

4. *Task accuracy* is how accurately $f(g(x))$ predicts y. *Concept accuracy* is how accurately $g(x)$ predicts c.

There are three well-defined training methods for CBMs: independent, sequential, and joint training. In machine learning, the loss function is a useful metric because it evaluates how well the model predicts the actual values. During training, the goal is to adjust the parameters to minimise a model's loss function. Let L_{C_j} be a loss function that measures the difference between the predicted and true j-th concept [6]. Let L_Y be the loss function that measures the difference between the predicted and true target label y values. Finally, let the learning concept bottleneck model $f(g(x))$ be represented using $\hat{f}$ and $\hat{g}$.

Independent

An independent bottleneck CBM learns $\hat{g}$ and $\hat{f}$ independently. This means at training time, $\hat{f}$ is optimised to predict the labels from the true concepts. At test time, $\hat{g}$ is used to generate the concepts from new inputs which are fed into $\hat{f}$.

Concept predictor:

$$\hat{g} = \arg\min_{g} \sum_{i,j} L_{C_j}(g_j(x^{(i)}); c_j^{(i)}) \tag{2.1}$$

Label predictor:

$$\hat{f} = \arg\min_{f} \sum_{i} L_Y(f(c^{(i)}); y^{(i)}) \tag{2.2}$$

Sequential

In a sequential bottleneck CBM, $\hat{g}$ is first trained to predict concepts and then $\hat{f}$ is trained to predict the target from the concepts predicted by $\hat{g}$. In this case $\hat{f}$ is not trained with the true concepts but with the concepts predicted by $\hat{g}$.

Concept predictor (learns it in the same way as independent bottleneck):

$$\hat{g} = \arg\min_{g} \sum_{i,j} L_{C_j}(g_j(x^{(i)}); c_j^{(i)}) \tag{2.3}$$

Label predictor:

$$\hat{f} = \arg\min_{f} \sum_{i} L_Y(f(\hat{g}(x^{(i)})); y^{(i)}) \tag{2.4}$$

Joint

The joint bottleneck CBM combines aspects of both the independent and sequential approaches. It trains $\hat{f}$ and $\hat{g}$ simultaneously and optimises a weighted sum of the losses for predicting concepts and the target label.

Concept and label predictor:

$$\hat{g}, \hat{f} = \arg\min_{f,g} \sum_{i} \Big[L_Y(f(g(x^{(i)})); y^{(i)}) + \sum_{j} \lambda L_{C_j}(g(x^{(i)}); c^{(i)}) \Big] \tag{2.5}$$

2.4 Large Language Models

Large Language Models or LLMs are artificial intelligence programs which can understand and generate texts like humans because they are trained on large amounts of data [9]. Rather than training them on a domain specific set of data, these models are fed millions of gigabytes worth of text from the internet. At the fundamental level, LLMs are based on a transformer model, which processes data by converting it into tokens and applies mathematical equations to it to identify the relationships between the tokens [10]. This allows the model to recognise patterns like how humans see them, and they can understand context, reply to human prompts, summarise text, generate code, etc.

Existing general LLMs have three main types of architectures: encoder-only, decoder-only, encoder-decoder [11]. Figure 2.3 offers a comprehensive list of existing general LLMs.

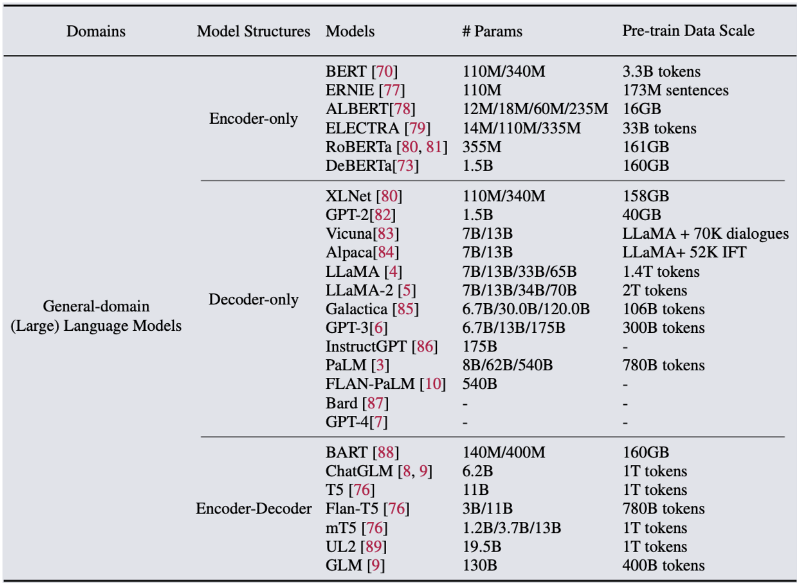
1. *Encoder-only*: Utilise stacks of transformer encoder layers which analyse input text bidirectionally, meaning they consider context from both before and after the token being processed. These LLMs are suited for getting a deep understanding of text which is useful for tasks like document classification and sentiment analysis. Examples include BERT and DeBERTa.

2. *Decoder-only*: Comprised of transformer decoder layers, these models process text unidirectionally (left-to-right), focusing on generating text by predicting the next token based on the preceding ones. These LLMs are suited for text generation tasks. Examples include the GPT-series developed by OpenAI, the LLaMA-series developed by Meta, PaLM and Bard (Gemini) developed by Google.

3. *Encoder-decoder*: Feature a combination of bidirectional encoder layers that understand input sequences and unidirectional decoder layers that produce output sequences. These LLMs are suited for both the comprehension and generation of text. Examples include Flan-T5 and ChatGLM.

Many leading companies have released LLMs which are open to use by the public [12]. However, the consensus about these models is that they have strong global performance but are weaker with specific task-oriented problems [13]. Hence, it is important to fine-tune these LLMs to have a better performance for specific tasks. There are several different types of fine-tuning [14]:

- *Transfer learning*: The goal of this fine-tuning technique is to transfer knowledge from the source task to the target task. This requires taking the model and fine-tuning it on a smaller, task-specific dataset. Domain-specific fine-tuning is a type of transfer learning approach where a model is fine-tuned on a dataset composed of text from a target domain to adapt its ability to understand domain-specific context.



Domains	Model Structures	Models	# Params	Pre-train Data Scale
General-domain (Large) Language Models	Encoder-only	BERT [70]	110M/340M	3.3B tokens
		ERNIE [77]	110M	173M sentences
		ALBERT[78]	12M/18M/60M/235M	16GB
		ELECTRA [79]	14M/110M/335M	33B tokens
		RoBERTa [80, 81]	355M	161GB
		DeBERTa[73]	1.5B	160GB
	Decoder-only	XLNet [80]	110M/340M	158GB
		GPT-2[82]	1.5B	40GB
		Vicuna[83]	7B/13B	LLaMA + 70K dialogues
		Alpaca[84]	7B/13B	LLaMA+ 52K IFT
		LLaMA [4]	7B/13B/33B/65B	1.4T tokens
		LLaMA-2 [5]	7B/13B/34B/70B	2T tokens
		Galactica [85]	6.7B/30.0B/120.0B	106B tokens
		GPT-3[6]	6.7B/13B/175B	300B tokens
		InstructGPT [86]	175B	-
		PaLM [3]	8B/62B/540B	780B tokens
		FLAN-PaLM [10]	540B	-
		Bard [87]	-	-
		GPT-4[7]	-	-
	Encoder-Decoder	BART [88]	140M/400M	160GB
		ChatGLM [8, 9]	6.2B	1T tokens
		T5 [76]	11B	1T tokens
		Flan-T5 [76]	3B/11B	780B tokens
		mT5 [76]	1.2B/3.7B/13B	1T tokens
		UL2 [89]	19.5B	1T tokens
		GLM [9]	130B	400B tokens

Figure 2.3: Table from [11]. It categorises the current state-of-the-art general LLMs based on the three different types of architectures. It also provides a quantitative summary of the number of parameters and the size of the datasets used for model training.

- *Few-shot learning*: Shots are examples of the required task. In this method of learning, the model is presented with numerous sets of input-output demonstrations to produce the intended response.

- *Alignment tuning*: This involves aligning the model with human intentions and values by using human feedback to guide the model's behaviour. For example, generating unexpected responses and updating the model's parameters accordingly. To do alignment tuning, researchers use reinforcement learning with human feedback (RLHF).

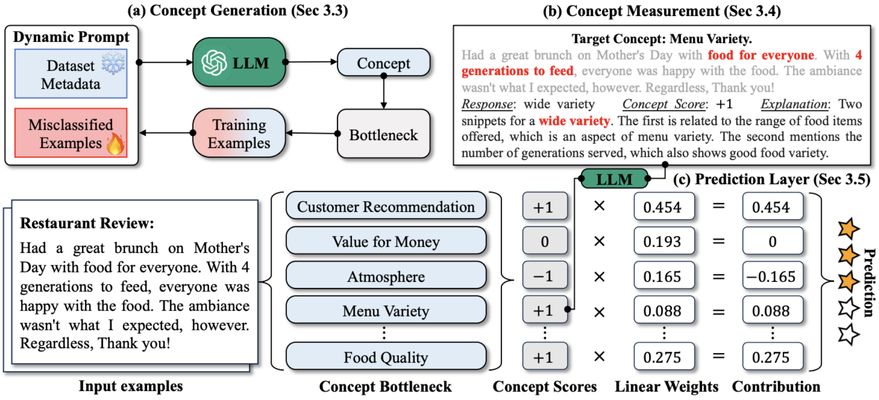
Chapter 3

Related Work

3.1 Automating Concept Generation

In the seminal CBM paper [6], one of the final points of discussion was the big limitation of concept bottleneck models (Section 2.2): they require annotated concepts at training time (not only do they need the x and y but also c). Traditional approaches of CBMs involve manually annotating the data to try and construct a set of concepts which is informative for the model's task. This requires a significant amount of time to be spent by individuals who are domain experts. This restricts the scalability of CBMs, especially in domains such as medicine which would require licensed physicians taking time out to do annotations.

Some studies have been conducted to try and address this issue. Work has been done to automate the task of generating concepts using LLMs. In [15], the authors proposed a Textual Bottleneck Model (TBM). This extends a CBM and uses concepts to make text classification interpretable with 3 steps: concept generation, concept measurement and final prediction, see Figure 3.1. Firstly, the model generates a set of concepts from the text by prompting an LLM to discover new concepts from the data that



Had a great brunch on Mother's Day with **food for everyone**. With **4 generations to feed**, everyone was happy with the food. The ambiance wasn't what I expected, however. Regardless, Thank you!

Response: wide variety *Concept Score*: +1 *Explanation*: Two snippets for a **wide variety**. The first is related to the range of food items offered, which is an aspect of menu variety. The second mentions the number of generations served, which also shows good food variety.

Had a great brunch on Mother's Day with food for everyone. With 4 generations to feed, everyone was happy with the food. The ambiance wasn't what I expected, however. Regardless, Thank you!

Figure 3.1: Demonstration of the three modules in Textual Bottleneck Models from [15]: Concept Generation, Concept Measurement, and Prediction Layer.

help tell the difference between examples that have been classified incorrectly. Each concept it generates contains 3 key pieces of information: the concept name, concept description and a question which has a predefined set of possible responses such "Positive", "Negative", "Uncertain" or "Not applicable". There is also a numerical score associated with each response which means in the TBM's concept measurement stage, an LLM is provided the text data and the concept question. Its response to the question allows the mapping of the concept to a score which assigns the value of the concept to the model. Finally, the concept scores are put together to make a final prediction.

The paper lays the groundwork for using LLMs to reduce the cost of CBMs. However, a human evaluation of the concept sets produced in the concept generation step revealed some key limitations. The authors found that the module had almost no problem discovering concepts relevant to the task label, but it occasionally accepted unnecessary concepts that were too similar to the previously generated ones. Furthermore, they observed that some concepts were directly leaking the task label. For certain datasets, instances of redundancy and leakage were especially high. Redundancy unnecessarily increases the size of the concept space and leaky concepts can undermine the faithfulness of the provided explanations. Since this project aims to generate concepts using an LLM from clinical notes, it is important to consider how the two issues will be tackled. Section 3.2 contains a detailed discussion on how to address leakage. The rest of this section looks at how to ensure concept quality and the work done in the "Concept bottleneck model for predicting vasopressor onset" project (Section 3.4) provides further ideas.

The concept measurement method discussed in [15] is an interesting approach for evaluating the usefulness of generated concepts. Additionally, in [16] the authors were developing a medical image classification CBM and they used a "visual activation score" for filtering LLM generated concepts. Their qualitative and quantitative evaluation revealed that the concepts selected using their score consistently boosted the performance of their CBM. Thus, the idea merits further exploration for this project.

In [17], the authors also developed a medical image classifier which incorporates natural language concepts to improve interpretability. They used GPT-4 to generate medical concepts. Rather than applying GPT-4 to any particular data, they explored the efficacy of the general model by asking a series of questions about radiology which became increasingly specific until eventually they were able to generate a list of concepts. The study found collecting ideal concepts which aligned with experts required "a few rounds of revision of the prompts and interaction with the model." Furthermore, "the generation results of GPT-4 come with randomness and are not perfectly controllable." The quality of the generated concepts heavily relied on the clarity and effectiveness of the questions posed to the model, indicating that some input and guidance from medical experts was still necessary. The paper reveals the need for fine-tuning a general-purpose LLM on the medical domain in order to generate useful concepts which this project aims to benefit from. It is also important to note that the LLM used in this project would have clinical notes as reference for producing concepts and would not be dependent on the quality of user prompts.

"Label-free" CBMs were proposed by [18], which used GPT-3 for producing the initial concept set. This paper formed the basis of the work done by [15], [16], and [17]. It provides some useful guidelines for filtering concepts generated by LLMs.

1. Concept length: Delete concepts longer than 30 characters to keep them simple.
2. Remove concepts similar to classes: Delete concepts too similar to target label class names using cosine similarity (> 0.85).
3. Remove similar concepts: Delete duplicate or synonymous concepts with cosine similarity > 0.9.
4. Remove concepts not present in training data: Delete concepts with low activation in CLIP (this is a cut-off specific to the dataset the paper was using).
5. Remove inaccurately projectable concepts: Remove neurons from the Concept Bottleneck Layer (CBL) that are not interpretable.

3.2 CBMs and Leakage

There are two ways of passing concepts to the label predictor $\hat{f}$ (Section 2.3) in concept bottleneck models: hard concepts (binary numbers 0 or 1) and soft concepts (any number between 0 and 1). Individual, sequential, and joint bottlenecks are all training methods for soft CBMs.

While evaluating concept bottleneck models, it was found that soft CBMs consistently outperform hard CBMs in predictive performance. Furthermore, when the concept predictor $\hat{g}$ and label predictor $\hat{f}$ are trained independently, they do not perform as well as a black-box predictor [8]. The best performing CBMs do not use the independent bottleneck training method. Instead, the inputs to the label predictor f are concept probabilities produced from the concept predictor.

However, soft CBMs which are trained sequentially or jointly suffer from information ‘leakage’, as discussed in [8]. Hard CBMs, on the other hand, are unaffected by this issue. Leakage is a phenomenon where the concept predictor unintentionally conveys extra information about the target label that should not be available to the label predictor. The paper uses the example of an animal classification task where the target class labels are ‘cat’ or ‘dog’. The model might identify subtle differences in the predicted probabilities for concepts like ‘fur’ and ‘tail’ which could allow the model to distinguish between cats and dogs, even though these features alone should not actually be enough for a reliable classification. This leakage makes it difficult to trust the model’s predictions and undermines its interpretability because it is unclear whether predictions are based on genuine concepts present in the data or just hidden encodings to guess the class label.

One of the first discussions on leakage in CBMs was in [5]. In the paper, leakage was mainly associated with joint training of CBMs, where the concept and label predictors are trained together, potentially leading the concept predictor to encode additional label information. The authors suggested that techniques like task-blind training or Concept Whitening, which decorrelates concept representations, could mitigate leakage. However, research by [19] suggests that leakage is a more widespread issue that persists even with natural mitigation strategies, particularly in models that use soft concept representations. This indicates a deeper, systemic problem within high-performing current CBMs. Having considered this research, the authors of [8] conducted studies to address the performance gap between hard and soft CBMs. By doing so, they could attain the robustness of hard CBMs against leakage while still achieving the higher accuracy of soft CBMs. They found that the gaps observed can largely be attributed to inadequate concept sets and inflexible concept predictors.

Looking more closely at the first reason, the paper suggests that hard CBMs will under-perform compared to a soft CBM predictor when the Markovian assumption is not met. The Markovian assumption is that the concept set c, which the CBM will be trained on, contains all the information about y present in x. To avoid leakage, this project can train a hard CBM and ensure the Markovian assumption is met. A completeness score can be used to determine when the assumption is fulfilled. Additionally, the project stands to gain by augmenting the existing concept set with additional concept labels extracted by an LLM.

3.3 LLMs Trained on Medical Data

Continuing the discussion on LLMs from Section 2.4, clinical LLMs are considered. These are general LLMs which have been fine-tuned for the medical domain [11]. Clinical LLMs have been trained on a high-quality medical corpus to conduct fine tuning through transfer learning. The leading models were obtained from fine-tuning Meta's LLaMA models and studies have demonstrated that these models are not only getting better at understanding and generating medical text, but they are improving their ability to make clinical decisions. Furthermore, researchers at Google have combined different types of prompting methods like few-shot learning and prompt turning to create a new technique which they call "Instruction Prompt Tuning", [20]. This was used to develop MedPaLM-2, an improved version of their previous model MedPaLM. The researchers reported "a competitive score of 86.5% compared to human experts 87.0% on the US Medical Licensing Examination" (USMLE) [11].

According to [11], although existing clinical LLMs are showing promising results, there are some key issues which are limiting their ability to be widely adopted in healthcare.

Most existing medical LLMs report their performance on answering medical questions like USMLE, however these examinations "do not capture the complexity of realistic clinical cases." When the authors of [21] constructed new datasets on more challenging clinical cases, LLMs like MedPaLM-2 and Llama-2 did not perform nearly as well. Furthermore, the paper revealed an inconsistency between automatic and human-evaluations of model generated explanation suggesting that expert clinicians still need to be in the loop to fully evaluate the performance of these LLMs. Another limitation of clinical LLMs discussed in [11] is that it is expensive and inefficient to inject new knowledge into an LLM through re-training. However, in the medical domain, it is necessary to update LLMs with things like a new adverse effect of a medication or a novel disease. Additionally, LLMs are black-box and their lack of interpretability (Section 2.1) makes it challenging to align their behaviours with the ethics and objectives of a medical task.

This reveals a gap in LLMs which can be filled using a concept layer. Clinical LLMs thrive when they have been fine-tuned for specialised tasks and using concepts is an excellent method for incorporating domain expertise into LLMs to encode clinical knowledge. Using concepts, clinicians can intervene and update the knowledge of a model and more easily evaluate the decision-making of an LLM. Concepts can also be used to tune the LLM to demonstrate the required behaviour for healthcare delivery.

3.4 MIMIC-Based Projects

Below are summaries of previous work done on the MIMIC dataset which is related to this project.

1. Predicting intervention onset in the ICU, [22].

This study worked on training "unsupervised switching state autoregressive models" to identify patterns in ICU patients' vital signs and predict one of five treatments for the patients based on their data. This model was trained on the MIMIC-III dataset and the study offers an insight into how to effectively select a cohort of patients for training a model. It also highlights the useful data types for predicting treatments:

- Clinical Observations: This is a time series of different clinical measurements taken at various times during the patient's stay in the ICU.
- Clinical Intervention Labels: This array indicates when and which medical interventions were given to the patient.
- Static Observations: These are observations that do not change over time, such as demographic information or initial clinical status.

2. Concept bottleneck model for predicting vasopressor onset, [7].

This paper is a valuable reference for developing a CBM for MIMIC. The authors created a concept bottleneck model which predicted how long, after administration, it would take vasopressors (blood medication) to start having effects on a patient. Their description of cohort selection, feature choices and evaluation approach is very insightful for this project. For data-preprocessing, pre-existing MIMIC-III data pipelines were used. This inspired a search for a similar MIMIC-IV data pipeline for this project, and the following was found [23].

Additionally, the researchers designed a procedure selecting the most interpretable and predictive concept definitions by using a "greedy optimisation" method. In Section 3.1, the idea of a concept measurement method was explored as a means to filter out unnecessary concepts. Furthermore, in Section 3.2, the importance of fulfilling the Markovian Assumption by achieving a high completeness score for a hard CBM was discussed. These examples demonstrate the value of having an algorithm to ensure the quality of concepts in this project. It certainly merits further investigation.

3. Google study on using LLMs for extracting insights from clinical notes, [24].

To train NLP models for understanding and interpreting clinical notes in EHRs, large amounts of labelled data are required. Relying only on human annotators is time consuming and expensive. This paper proposes using LLMs along with human annotators. They use Google's medical LLM Med-PaLM to do "base annotations" and then bring in actual domain experts to refine them.

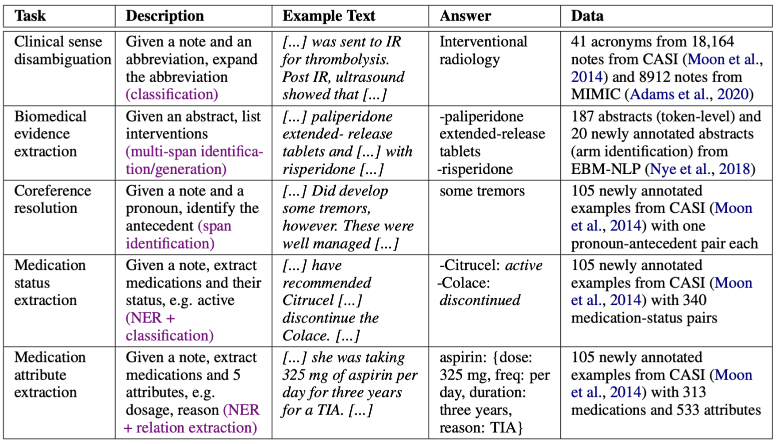
The authors propose a few-shot learning approach (Section 2.4) and demonstrate promising results from quantitative evaluation. They lacked a comprehensive qualitative study, which this project aims to fulfil. The paper has two major contributions for this project; firstly, it provides an insight into how to refine the LLM's prompt structure and schema. Secondly, the authors used their medical information extraction pipeline on the MIMIC-IV clinical notes (specifically the discharge data), and they have produced annotations for the text within the CSV files (things like REASON, MEDICATION, MODE, DURATION), available on PhysioNet [25]. This is the first publicly available set of annotated labels for MIMIC-IV clinical notes and forms a useful baseline of comparison for one of this project's deliverables.

4. Studying 5 different LLM tasks for extracting insights from clinical notes, [26].

In this paper, the authors set a benchmark for how LLMs such as GPT-3 perform at clinical NLP tasks using different medical data including MIMIC. They documented their evaluation on 5 diverse medical information extraction tasks, which are well-defined and will inform the prompting structure for this project, Figure 3.2.

5. LLMs built using MIMIC, [27].

T5 stands for "Text-To-Text Transfer Transformer," a type of encoder-decoder large language model architecture (Section 2.4). The authors trained four different T5 models in their study. Two of these models were initialised from previous T5 models, while the



Task	Description	Example Text	Answer	Data
Clinical sense disambiguation	Given a note and an abbreviation, expand the abbreviation (classification)	*[...] was sent to IR for thrombolysis. Post IR, ultrasound showed that [...]*	Interventional radiology	41 acronyms from 18,164 notes from CASI (Moon et al., 2014) and 8912 notes from MIMIC (Adams et al., 2020)
Biomedical evidence extraction	Given an abstract, list interventions (multi-span identification/generation)	*[...] paliperidone extended- release tablets and [...] with risperidone [...]*	-paliperidone extended-release tablets -risperidone	187 abstracts (token-level) and 20 newly annotated abstracts (arm identification) from EBM-NLP (Nye et al., 2018)
Coreference resolution	Given a note and a pronoun, identify the antecedent (span identification)	*[...] Did develop some tremors, however. These were well managed [...]*	some tremors	105 newly annotated examples from CASI (Moon et al., 2014) with one pronoun-antecedent pair each
Medication status extraction	Given a note, extract medications and their status, e.g. active (NER + classification)	*[...] have recommended Citrucel [...] discontinue the Colace. [...]*	-Citrucel: *active* -Colace: *discontinued*	105 newly annotated examples from CASI (Moon et al., 2014) with 340 medication-status pairs
Medication attribute extraction	Given a note, extract medications and 5 attributes, e.g. dosage, reason (NER + relation extraction)	*[...] she was taking 325 mg of aspirin per day for three years for a TIA. [...]*	aspirin: {dose: 325 mg, freq: per day, duration: three years, reason: TIA}	105 newly annotated examples from CASI (Moon et al., 2014) with 313 medications and 533 attributes

Figure 3.2: Overview of the five tasks studied in [26].

other two were trained from scratch. Below is a more detailed description from [27]:

1. Clinical-T5-Base: This model was initialised from the T5-Base architecture, which was previously trained on a variety of general text using the masked language modelling (MLM) training scheme. The authors then fine-tuned this model on clinical text from MIMIC-III and MIMIC-IV notes.

2. Clinical-T5-Sci: This model was initialised from SciFive, which uses the T5-Base architecture as its starting point. SciFive further pretrains the model on PubMed abstracts and PubMed Central (these are a large collection of biomedical literature). The authors then fine-tuned this model on clinical text from MIMIC-III and MIMIC-IV notes.

3. Clinical-T5-Scratch: This model uses the same architecture as T5-Base but with randomly initialised weights. The authors constructed a vocabulary for the model based on MIMIC notes and then trained it from scratch using the MLM task with chunks of text from MIMIC.

4. Clinical-T5-Large: This model uses the same architecture as T5-Large but with randomly initialised weights. Similarly, the authors constructed a vocabulary for the model based on MIMIC notes and trained it from scratch using the MLM task with chunks of text from MIMIC.

The models have been made available of PhysioNet and will be very useful sources of comparison for the models produced in this project.

Chapter 4

Implementation

Having reflected on relevant literature, the proposed contributions of this project are defined as follows:

- Investigate the performance of a fine-tuned LLM for generating concepts from clinical notes. Furthermore, propose a scoring system for selecting useful concepts. This will contribute to the efforts of improving the scalability of CBMs by automating the concept annotation process.
- Add concepts to a hard CBM's existing concept set from additional clinical notes and build new concepts from this to minimise leakage.
- Produce more complete medical concept definitions by using an LLM on new clinical data.

4.1 MIMIC Database

In order to access MIMIC-IV, certain credentials had to be obtained (Chapter 7). Then, background reading was completed about the database. According to the MIMIC Docs [28], "MIMIC-IV contains data from 2008-2019. The data was collected from Metavision bedside monitors. MIMIC-III contains data from 2001-2012. The data was collected from Metavision and CareVue bedside monitors." Most of the works discussed in Section 3.4 use MIMIC-III which has some subtle differences from MIMIC-IV. According to the docs, MIMIC-IV includes some of the patient data from MIMIC-III during the years after the CareVue system stopped being used. Additionally, "a number of improvements have been made [to the MIMIC-IV database], including simplifying the structure, adding new data elements, and improving the usability of previous data elements."

With this in mind, it was crucial to pinpoint the relevant aspects of MIMIC for the project. Initially the clinical notes pertinent to the LLM were considered. These notes are stored in CSV files within MIMIC, and basic scripts were developed to query the

MIMIC-IV Clinical Notes files. This exercise aimed to grasp the structure of the notes, yielding the following insights:

In the Clinical Notes, there are four files, and the two mains ones are the discharge and radiology notes. The additional two files, discharge detail and radiology detail, provide auxiliary information about the primary notes. Below is information from [28]:

- Discharge notes: These narratives detail a patient's reason for admission, hospital course, and relevant discharge instructions. *Key columns include note_id, subject_id, hadm_id, note_type, note_seq, charttime, storetime, and text.*
- Radiology notes: These encompass free-text reports associated with radiography imaging. There are two types: RR (radiology report) and AR (radiology report addendum). *Key columns mirror those in discharge notes.*
- The addendum typically includes further observations, clarifications, or additional findings that were not included in the initial report but are deemed relevant or important.

The following was observed from running scripts to query the files.

Discharge Notes: These typically contain more detailed information compared to radiology notes. While some subheadings, like Oncological history, may vary depending on the patient, many subheadings and structures are repeated. The amount of data captured also varies.

Radiology RR Notes: The subheadings (data captured) vary based on the examination type, resulting in different structures. The amount of data captured varies across different types of examinations.

Radiology AR Notes: These tend to be shorter and accompany corresponding RR texts. Unlike RR notes, there are no subheadings; AR notes consist of short additional notes.

From this preliminary investigation, it became evident that querying the Clinical Notes would be a straightforward process. However, some considerations need to be made regarding cohort selection for the actual prediction task. Additionally, for longer discharge notes, preprocessing may be necessary to adhere to the token limit of LLMs as described in [29].

This concluded the exploration of MIMIC for Phase 1. Moving forward, the chosen downstream task is predicting patient mortality. This entails identifying relevant patient information as x inputs for the concept bottleneck model from the structured data in MIMIC. Other considerations include cohort selection and strategies for handling missing data and abnormalities. Papers addressing patient mortality prediction using MIMIC-IV, like [30][31][32], will help with this. Furthermore [23] offers pipeline tools for more complex MIMIC data extraction and pre-processing.

4.2 LLM Basic Implementation

For an initial LLM, a pretrained Llama 2 model was fine-tuned on a dataset from Hugging Face (a company that offers free tools for AI and ML). The dataset was made up of a question as an input and the correct corresponding answer as output. It was designed to enable supervised fine-tuning. The code was run on the single GPU option of Google Colab. Some key aspects of this fine-tuning approach involved using Quantisation and Parameter Efficient Fine Tuning (PEFT). Quantisation is the process of reducing the precision of the model's weights and activations. It is a way to reduce the model's computation requirements and memory footprint without having a significant impact on performance. PEFT is a technique designed to address the inefficiency and resource-intensive nature of fine-tuning every single parameter of an LLM. Instead of fine-tuning the entire model, PEFT selectively identifies and fine-tunes only a small subset of parameters. This focused approach significantly reduces the computational burden, making the fine-tuning process more efficient and practical.

In later phases of this project, an LLM needs to be implemented on the MIMIC-IV Clinical Notes. It may need to be fine-tuned on a specific medical dataset, and would build on the approach used for the initial LLM.

4.3 CBM Basic Implementation

At this stage of the project, a very simple CBM has been implemented to understand how the model gets trained. The "PyTorch, Explain!" library was utilised to abstract aspects of the CBM implementation [33]. The library provided a dataset of x, c, and y values, which required no pre-processing. In the future, work will need to be done to produce useful training concepts for mortality prediction from MIMIC. The CBM's architecture had two main components: a concept encoder and a task predictor. The concept encoder processes input features to produce both raw concept embeddings and predicted concept labels. These embeddings are then fed into the task predictor, which uses them to predict the final output. During training, the model was optimised using a combination of concept loss, which ensures the predicted concept labels accurately match the true labels, and task loss, which aligns the final predictions with the desired outputs. The training involved adjusting the model parameters through backpropagation, using an Adam optimiser over multiple epochs to minimise the overall loss.

Chapter 5

Project Plan

The plan for this project is illustrated in the Gantt Chart in Figure 5.1. For reference, the deadline for this Interim Report is on the Friday of Week 5.

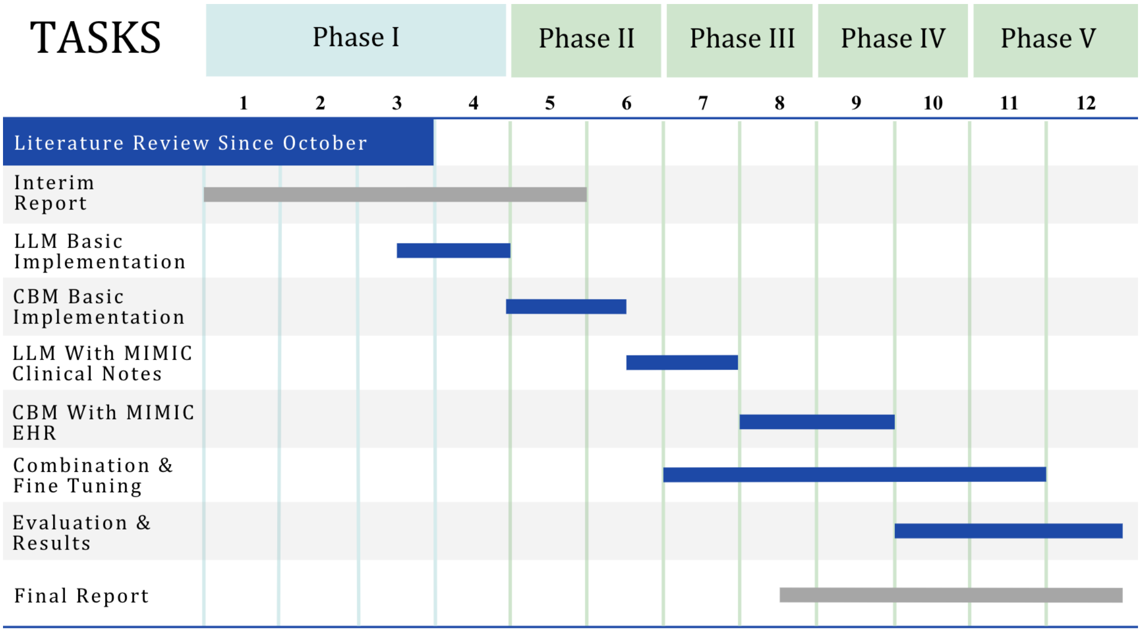


Figure 5.1: Weeks numbered 1-12 enumerate the time dedicated solely to the Final Year Project during Easter Holidays and Summer Term. The timeline is broken down into 5 distinct phases which reflect the levels of progress through the project.

Phase I: Previous Terms and Weeks 1-4

During Autumn and Spring Term of this academic year, regular meetings with the Project Supervisor were attended which helped inform the literature review for this project. From Week 1 onwards, this report was written and a basic LLM was implemented, Section 4.

Phase II: Weeks 5-6

The aim for this phase is to implement a basic end-to-end concept bottleneck model on an arbitrary dataset. Like with the LLM implementation, the goal is to set up the develop environment and understand the steps required to create a model to inform the implementation of a CBM on the MIMIC EHR data. During the second half of Week 6, the groundwork for setting up a pre-trained LLM to extract textual concepts from the MIMIC Clinical Notes will be established.

Phase III: Weeks 7-8

The aim of this phase is to complete the LLM task and get started with running a CBM on the MIMIC EHR data. Fine-tuning of the LLM can also commence. At this point of the project, it will be possible to begin writing the Implementation section of the Final Report. Additionally, work on the Front Matter, Acknowledgement, Introduction, and Background sections can begin to be finalised for the draft report deadline.

Phase IV: Weeks 9-10

During this phase, the aim to is to complete the CBM task and begin combining the two models. A decision will be reached on which approach for combination should be adopted. Fine-tuning of the LLM and CBM parameters will take place and test results will be logged. In Week 9, the Testing, Results and Evaluation sections of the Final Report can be started. The Abstract and Draft Report deadline is on the Monday of Week 10.

Phase V: Weeks 11-12

In this phase, the aim is to finalise the fine-tuning parameters of the combined models, collect the results that are needed for the Final Report, and finish writing the Final Report. Its deadline is on the Friday of Week 12.

Chapter 6

Evaluation Plan

The exact details for the evaluation of this project's deliverables are not possible to define while in Phase I of the timeline. However, upon reviewing the literature (Chapter 3), potential evaluation methods can be discussed.

6.1 Quantitative Evaluation

To numerically evaluate the predictive performance of concept bottleneck models, the consensus was to use accuracy of the concept predictor and label predictor, [8][17][18][34]. In some papers where the output of the model was binary, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used [7][34]. "The ROC curve is a graphical representation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) for different threshold values", [35]. The AUC-ROC summarizes the performance of the classifier across all possible threshold values, ranging from 0 to 1. A classifier with an AUC-ROC of 1 indicates perfect discrimination between the positive and negative classes, while a classifier with an AUC-ROC of 0.5 suggests random performance (no discrimination). Typically, higher AUC-ROC values indicate better classifier performance.

Another important quantitative metric to include in evaluating CBMs is the assessment of concept quality. [8] proposes a completeness score to evaluate whether concept set meets the Markovian assumption, although this requires implementing side-channels. [36] discusses a more general score for quantifying "how sufficient a particular set of concepts is in explaining a model's prediction behavior." Other papers have context specific methods for evaluating concept quality based on the dataset [15][16]. To enhance this project exploring metrics beyond accuracy for assessing concept quality by drawing inspiration from these works would be beneficial.

For the LLM aspect of this project, the evaluation should be on the insights extracted from the MIMIC Clinical Notes. The quality of insights is best suited to qualitative evaluation as discussed below. However, when assessing which pretrained model to

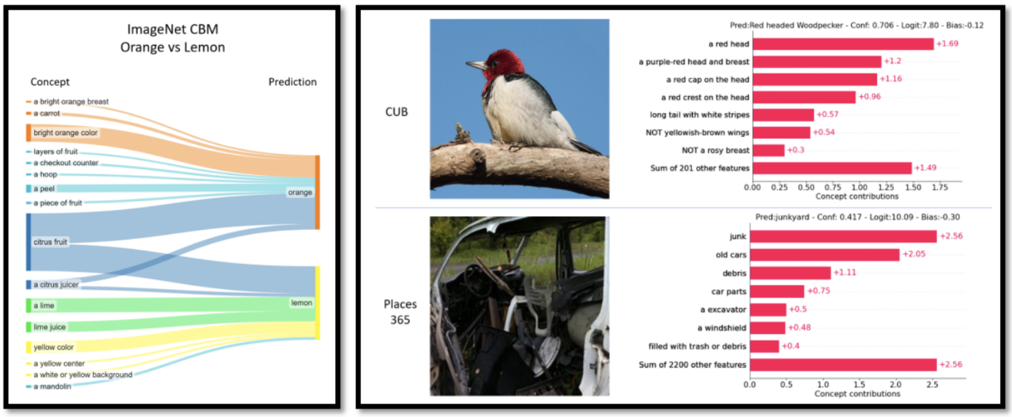


Figure 6.1: Sankey diagram and bar plots from [18] to visualise the concept set

utilise for processing the clinical notes, precision, recall, and F1-score are the most commonly used metrics. [25] broke down their concept annotation task into two parts: Named Entity Recognition (NER) (for identifying fields) and Relationship Extraction (RE) (for linking the fields together to form a medication entry). Using an algorithm inspired from [37], they obtained vertical and horizontal metrics for NER and RE, respectively. These metrics are used to produce precision, recall, and F1-score. [26] and [38] also comprehensively evaluated LLMs for 5 different medical tasks. Precision, recall, and F1-score were used for the tasks that involved concept extraction.

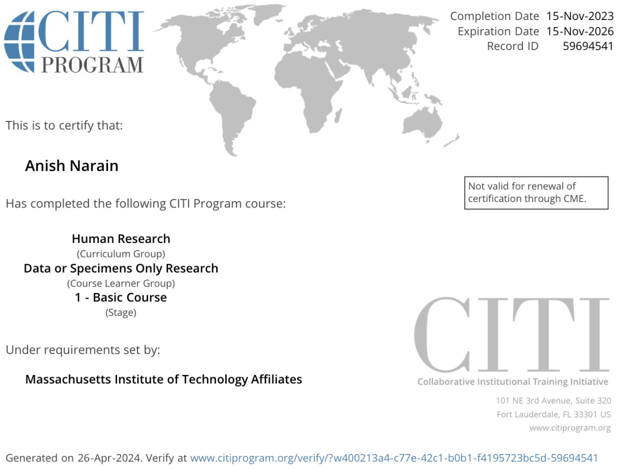
6.2 Qualitative Evaluation

The qualitative evaluation for this project should focus on the concept set produced from the combination of LLM and CBM. This would help identify whether the representations learned by the model actually capture meaningful clinical information [22]. By incorporating textual information alongside numerical values from the Electronic Health Record (EHR), these concepts are more interpretable and this advantage can be utilised. In [18], a Sankey diagram was used to provide a global understanding of how the model behaves. Furthermore, the contribution of the most important concept features to the overall prediction of the model was visualised with a bar plot. Having visualisation methods like Figure 6.1 would make it much easier to present the concepts learned by the model to medical experts. In [17], the authors brought in a "board-certified radiologist to manually evaluate the quality of interpretations from our models." Similarly, a licensed physician will be invited to evaluate the concept set in this project.

Chapter 7

Ethical, Legal, and Safety Plan

The work done in this project involves using real medical data from patients, which is highly sensitive, so complying with all safety guidelines to work with the dataset is a top priority. To obtain the credentials for accessing MIMIC, the "CITI Data or Specimens Only Research" course was completed (7.1). This course covers important aspects of research with human participant data including ethical principles, privacy and HIPAA (Health Insurance Portability and Accountability Act) compliance.



CITI PROGRAM

Completion Date 15-Nov-2023
Expiration Date 15-Nov-2026
Record ID 59694541

This is to certify that:

Anish Narain

Not valid for renewal of certification through CME.

Has completed the following CITI Program course:

Human Research
(Curriculum Group)
Data or Specimens Only Research
(Course Learner Group)
1 - Basic Course
(Stage)

Under requirements set by:

Massachusetts Institute of Technology Affiliates

CITI
Collaborative Institutional Training Initiative
101 NE 3rd Avenue, Suite 320
Fort Lauderdale, FL 33301 US
www.citiprogram.org

Generated on 26-Apr-2024. Verify at www.citiprogram.org/verify/?w400213a4-c77e-42c1-b0b1-f4195723bc5d-59694541

Figure 7.1: Completion certificate for Data or Specimens Only Research Course

Furthermore, the terms of use for the MIMIC dataset [39] will be diligently observed. This includes exercising care to prevent disclosure of identities in any communication, not sharing access with others, using the data solely for lawful scientific research, contributing code if publishing results, and acknowledging that these obligations persist even after termination of the agreement.

Finally, research has been shown that large language models have the potential to leak sensitive information [40]. Hence, following the example from [27], any models released to the public will be done so under PhysioNet credentialed access.

Bibliography

[1] A. Johnson, L. Bulgarelli, T. J. Pollard, S. Horng, L. A. Celi, and R. G. Mark, *MIMIC-IV (version 2.2)*, `https://doi.org/10.13026/6mm1-ek67`, 2023.

[2] A. Johnson, T. J. Pollard, S. Horng, L. A. Celi, and R. G. Mark, *MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2)*, `https://doi.org/10.13026/1n74-ne17`, 2023.

[3] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[4] Google Cloud, *What is supervised learning?* Accessed: 2024-04-21, 2024. [Online]. Available: `https://cloud.google.com/discover/what-is-supervised-learning`.

[5] A. Margeloiu, M. Ashman, U. Bhatt, Y. Chen, M. Jamnik, and A. Weller, "Do concept bottleneck models learn as intended?" *arXiv preprint arXiv:2105.04289*, 2021.

[6] P. W. Koh, T. Nguyen, Y. S. Tang, *et al.*, "Concept bottleneck models," in *International conference on machine learning*, PMLR, 2020, pp. 5338–5348.

[7] C. Wu, S. Parbhoo, M. Havasi, and F. Doshi-Velez, "Learning optimal summaries of clinical time-series with concept bottleneck models," in *Machine Learning for Healthcare Conference*, PMLR, 2022, pp. 648–672.

[8] M. Havasi, S. Parbhoo, and F. Doshi-Velez, "Addressing leakage in concept bottleneck models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 386–23 397, 2022.

[9] "What is a large language model?" Cloudflare. (2024), [Online]. Available: `https://www.cloudflare.com/en-gb/learning/ai/what-is-large-language-model/`.

[10] "Large language models - what is a large language model?" Elastic. (2023), [Online]. Available: `https://www.elastic.co/what-is/large-language-models`.

[11] H. Zhou, B. Gu, X. Zou, *et al.*, "A survey of large language models in medicine: Progress, application, and challenge," *arXiv preprint arXiv:2311.05112*, 2023.

[12] "Hugging face models," Hugging Face. (2024), [Online]. Available: `https://huggingface.co/models`.

[13] J. Ferrer. “An introductory guide to fine-tuning llms,” DataCamp. (Feb. 2024), [Online]. Available: `https://www.datacamp.com/tutorial/fine-tuning-large-language-models`.

[14] H. Naveed, A. U. Khan, S. Qiu, *et al.*, “A comprehensive overview of large language models,” *arXiv preprint arXiv:2307.06435*, 2023.

[15] J. M. Ludan, Q. Lyu, Y. Yang, L. Dugan, M. Yatskar, and C. Callison-Burch, “Interpretable-by-design text classification with iteratively generated concept bottleneck,” *arXiv preprint arXiv:2310.19660*, 2023.

[16] I. Kim, J. Kim, J. Choi, and H. J. Kim, “Concept bottleneck with visual concept filtering for explainable medical image classification,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2023, pp. 225–233.

[17] A. Yan, Y. Wang, Y. Zhong, *et al.*, “Robust and interpretable medical image classifiers via concept bottleneck models,” *arXiv preprint arXiv:2310.03182*, 2023.

[18] T. Oikarinen, S. Das, L. M. Nguyen, and T.-W. Weng, “Label-free concept bottleneck models,” *arXiv preprint arXiv:2304.06129*, 2023.

[19] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and W. Pan, “Promises and pitfalls of black-box concept learning models,” *arXiv preprint arXiv:2106.13314*, 2021.

[20] K. Singhal, T. Tu, J. Gottweis, *et al.*, “Towards expert-level medical question answering with large language models,” *arXiv preprint arXiv:2305.09617*, 2023.

[21] H. Chen, Z. Fang, Y. Singla, and M. Dredze, “Benchmarking large language models on answering and explaining challenging medical questions,” *arXiv preprint arXiv:2402.18060*, 2024.

[22] M. Ghassemi, M. Wu, M. C. Hughes, P. Szolovits, and F. Doshi-Velez, “Predicting intervention onset in the icu with switching state space models,” *AMIA Summits on Translational Science Proceedings*, vol. 2017, p. 82, 2017.

[23] M. Gupta, B. Gallamoza, N. Cutrona, P. Dhakal, R. Poulain, and R. Beheshti, “An Extensive Data Processing Pipeline for MIMIC-IV,” in *Proceedings of the 2nd Machine Learning for Health symposium*, ser. Proceedings of Machine Learning Research, vol. 193, PMLR, 28 Nov 2022, pp. 311–325. [Online]. Available: `https://proceedings.mlr.press/v193/gupta22a.html`.

[24] A. Goel, A. Gueta, O. Gilon, *et al.*, “Llms accelerate annotation for medical information extraction,” in *Machine Learning for Health (ML4H)*, PMLR, 2023, pp. 82–100.

[25] A. Goel, A. Gueta, O. Gilon, S. Erell, and A. Feder, *Medication extraction labels for mimic-iv-note clinical database (version 1.0.0)*, PhysioNet, `https://doi.org/10.13026/ps1s-ab29`, 2023.

[26] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, “Large language models are few-shot clinical information extractors,” *arXiv preprint arXiv:2205.12689*, 2022.

[27] E. Lehman and A. Johnson, *Clinical-T5: Large Language Models Built Using MIMIC Clinical Text (version 1.0.0)*, PhysioNet, 2023. DOI: `10.13026/rj8x-v335`. [Online]. Available: `https://doi.org/10.13026/rj8x-v335`.

[28] MIT Laboratory for Computational Physiology. "MIMIC documentation." (2021), [Online]. Available: `https://mimic.mit.edu/docs/about/`.

[29] S. Kweon, J. Kim, H. Kwak, *et al.*, "Ehrnoteqa: A patient-specific question answering benchmark for evaluating large language models in clinical settings," *arXiv preprint arXiv:2402.16040*, 2024.

[30] Z. Nowroozilarki, A. Pakbin, J. Royalty, D. K. Lee, and B. J. Mortazavi, "Real-time mortality prediction using mimic-iv icu data via boosted nonparametric hazards," in *2021 IEEE EMBS international conference on biomedical and health informatics (BHI)*, IEEE, 2021, pp. 1–4.

[31] T. Huang, D. Le, L. Yuan, S. Xu, and X. Peng, "Machine learning for prediction of in-hospital mortality in lung cancer patients admitted to intensive care unit," *PLOS one*, vol. 18, no. 1, e0280606, 2023.

[32] K. Pang, L. Li, W. Ouyang, X. Liu, and Y. Tang, "Establishment of icu mortality risk prediction models with machine learning algorithm using mimic-iv database," *Diagnostics*, vol. 12, no. 5, p. 1068, 2022.

[33] P. Barbiero, *Version 0.5.4*, `https://doi.org/10.5281/zenodo.4903521`, Accessed on May 2, 2024, 2021.

[34] U. Klimiene, R. Marcinkevičs, P. Reis Wolfertstetter, *et al.*, "Multiview concept bottleneck models applied to diagnosing pediatric appendicitis," in *2nd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, ETH Zurich, Institute for Machine Learning, 2022.

[35] N. Krishna and K. Nagamani, "Multi-modal imaging-based feature fusion for accurate glaucoma diagnosis with deep learning," 2023.

[36] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," *Advances in neural information processing systems*, vol. 33, pp. 20 554–20 565, 2020.

[37] Ö. Uzuner, I. Solti, and E. Cadag, "Extracting medication information from clinical text," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 514–518, 2010.

[38] X. Yang, A. Chen, N. PourNejatian, *et al.*, "A large language model for electronic health records," *NPJ digital medicine*, vol. 5, no. 1, p. 194, 2022.

[39] PhysioNet. "MIMIC-IV Data Use Agreement." Supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) under NIH grant number R01EB030362., MIT Laboratory for Computational Physiology. (n.d.), [Online]. Available: `https://physionet.org/content/mimiciv/view-dua/2.2/`.

[40] J. G. Wang, J. Wang, M. Li, and S. Neel, "Pandora's white-box: Increased training data leakage in open llms," *arXiv preprint arXiv:2402.17012*, 2024.