Methods and Applications for Summarising Free-Text Narratives in Electronic Health Records

Searle, Tom

Awarding institution:
King's College London

Methods and Applications for Summarising Free-Text Narratives in Electronic Health Records

Thomas Searle

Supervisor: Prof. Richard Dobson

Dr. Zina Ibrahim

The Department of Biostatistics and Health Informatics

King's College London

This dissertation is submitted for the degree of

Doctor of Philosophy

September 2023

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Thomas Searle
September 2023

Acknowledgements

I would like to acknowledge firstly my supervisors Professor Richard Dobson and Dr Zina Ibrahim who have supported my work throughout the thesis, provided a unqiuely collaborative environment encouraging cross academic / health provider collaborations to work on problems that are guided by domain experts. This has provided wealth of collaborations and opportunities to work with data and groups nationally and internationally resulting in well-rounded PhD experience allowing for a broad experience of where the state-of-the-art is and where it is headed.

I would like to acknowledge my department, college and cross-college colleagues of which there are too many to name. For all of the collaborative efforts over research projects, experiments and manuscript writing. The core outputs of this thesis would not have been possible without you all. Special mentions to Professor James Teo who has offered expert clinical guidance throughout the thesis day and night, and to my frequent collaborators and friends Zeljko Kraljevic, Anthony Shek and Aurelie Mascio who have provided support, research ideas and above all laughter throughout the years.

Finally and most importantly - thank you to my family and wife Ellie for supporting me every step of the way, in my decision to undertake a PhD, through the weeks and months of intense focus and always believing I would get there. Very lastly our dog Noodle - who obviously can’t read - but thank you for the mandatory walk breaks and cuddles on the sofa after a long day.

Abstract

As medical services move towards electronic health record (EHR) systems the breadth and depth of data stored at each patient encounter has increased. This growing wealth of data and investment in care systems has arguably put greater strain on services, as those at the forefront are pushed towards greater time spent in front of computers over their patients. To minimise the use of EHR systems clinicians often revert to using free-text data entry to circumvent the structured input fields. It has been estimated that approximately 80% of EHR data is within the free-text portion. Outside of their primary use, that is facilitating the direct care of the patient, secondary use of EHR data includes clinical research, clinical audits, service improvement research, population health analysis, disease and patient phenotyping, clinical trial recruitment to name but a few.

This thesis presents a number of projects, previously published and original work in the development, assessment and application of summarisation methods for EHR free-text. Firstly, I introduce, define and motivate EHR free-text analysis and summarisation methods of open-domain text and how this compares to EHR free-text. I then introduce a sub-problem in natural language processing (NLP) that is the recognition of named entities and linking of the entities to pre-existing clinical knowledge bases (NER+L). This leads to the first novel contribution the Medical Concept Annotation Toolkit (MedCAT) that provides a software library workflow for clinical NER+L problems. I frame the outputs of MedCAT as a form of summarisation by showing the tools contributing to published clinical research and the application of this to another clinical summarisation use-case 'clinical coding'. I then consider methods for the textual summarisation of portions of clinical free-text. I show

how redundancy in clinical text is empirically different to open-domain text discussing how this impacts text-to-text summarisation. I then compare methods to generate discharge summary sections from previous clinical notes using methods presented in prior chapters via a novel ‘guidance’ approach.

I close the thesis by discussing my contributions in the context of state-of-the-art and how my work fits into the wider body of clinical NLP research. I briefly describe the challenges encountered throughout, offer my perspectives on the key enablers of clinical informatics research, and finally the potential future work that will go towards translating research impact to real-world benefits to healthcare systems, workers and patients alike.

Table of contents

List of figures

Chapter 1

Introduction

Modern healthcare systems produce healthcare records for every patient encounter. Synonymous terms for such data include medical record or medical/health *charts*. Records often begin at birth and end in death, and contain sensitive, personal information covering physical and mental health, social histories of the patient and potentially relatives and even prognostic data. The documentation of healthcare encounters is crucial to the effective and safe delivery of care across settings [101, 89]. Care settings can be classified into: *primary* care - the first point of contact with a healthcare service, *secondary* - the local hospital visit often via a referral from primary care, *tertiary* care - a specialist hospital where patients are referred usually from secondary care, and *social* or *community* care - that occurs directly at home or outside of a clinician surgery such as a pharmacy, optician or dental practice. The importance of effective communication through clear and thorough documentation is highlighted as patients move between care settings, e.g. primary to secondary [156], or care teams shift between: inpatient / outpatient, day / night, or between hospital departments [26, 58, 79]. Often, care providers 'silo' their patient data so the burden of data sharing is on the patient themselves [139]. Patients often must recount basic clinical information: medical / social history, allergies, current and past diseases, medications etc. making for an inefficient, frustrating experience for patients and care providers [34].

The *primary* use of medical records directly concerns patient care, while the *secondary* use includes use cases such as clinical research and population health analysis. Secondary use of clinical records holds great potential allowing care systems to continually tailor, improve and learn from past experience. Currently, clinicians rely on their training and prior experience combined with clinical practice guidelines (CPGs) [88], such as the NICE guidelines[1] in the United Kingdom (UK). CPGs are the result of a feedback loop, where care is evidence and data-driven. However, CPGs have to be integrated and consistently adhered to in care [91] and they can be slow in responding to research outcomes [113]. Moreover, they are not tailored to individual patients or patient sub-groups [45] and their initial development can be expensive, time-consuming [111] and even contradictory [110]. The burden of guidelines usage also falls on clinical staff, so changes must be interpreted and applied by staff in clinical settings as guidelines are introduced or updated, leading to inconsistencies between care offered between hospitals.

Global healthcare systems have recently moved from paper-based systems to electronic health record (EHR) systems [14]. This is a step towards enabling scalable secondary usage of routinely collected data to inform research, population analysis and ultimately improve direct patient care [66, 124]. Routinely collected data include vital signs, laboratory tests (i.e. 'structured' tabular data), diagnostic imaging (x-ray, MRI, CT, echo etc.), and clinical narratives (i.e. 'unstructured' free-text) where clinicians provide free-text expressions that interpret the current patient state, document current hospital course and next steps.

Multi-modal data sources present a variety of problems with analysis at scale [153]. Imaging and structured data items are often accompanied by clinical narratives that explain findings in an image or a laboratory measure. Despite ongoing efforts to improve EHR system user interfaces to support structured data entry, inputting free-text clinical narratives is the easiest, most natural input method afforded to clinicians. Murdoch and Detsky [99] estimates 80% of EHRs are unstructured data. Structured data can easily be queried, aggregated and analysed as it is well formed and typed, whereas unstructured data is often

[1] https://www.nice.org.uk/guidance

messy, incomplete and highly variable. Due to the difficulties working with unstructured data most uses of secondary uses of EHR data i.e. reporting and research, focus on the minority share of structured data [66].

The widespread usage of EHRs has lead to a deluge of data [124] with the potential uses in areas including: drug discovery [164], precision healthcare [112] and real-time population health analysis [27] to name a few. This potential is for the most part unrealised, as EHR adoption globally has often been driven by top-down regulatory forces, deployments have often catered to the administrative processes in care delivery rather than prioritising patient and care provider perspectives. This has contributed to the current scenario of increased spending on systems and healthcare broadly, but conversely declining outcomes [147]. Artificial Intelligence (AI), interpreted as the convergence of technology and data to create adaptable algorithms that are not explicitly coded to perform a task, has been considered as one route to draw down the already considerable costs incurred with the move to EHRs. AI methods for healthcare include: natural language processing (NLP) algorithms to read, structure and process clinical text [84, 73], computer vision algorithms to segment, classify and triage radiology scans [3, 120], multi-modal decision support algorithms in surgery [18], multi-modal algorithms for pathology diagnosis and prognosis [28], diagnosis prediction in specialties - e.g. emergency medicine [67], cardiology [83], neurology [103], pulmonology [65], gastroenterology [74].

Throughout the thesis I will present methodological developments and example applications for the *summarisation* of EHR free-text data. I consider *summarisation* of unstructured data to include methods that either identify and extract structured data from the text as well as the generation of further free-text that captures the salient and most informativeness parts of unstructured data. The applicability of the methods is strengthened by considering large real-world EHR text corpora covering disparate clinical specialities and healthcare provider geographies. This uniquely places my work across multiple boundaries of leading-edge clinical AI development, empirical testing across wide ranging real-world datasets and translation of initial research results into clinical impact providing insights and a perspective for how AI systems can be utilised within healthcare research, healthcare delivery and healthcare administration.

1.1 Research Questions and Hypotheses

Overall, my guiding research question asks how NLP methods and approaches can be used to improve current performance in real-world clinical text summarisation tasks.

Firstly, I ask if novel methods can improve current baseline performance for a common task in clinical text summarisation - the identification and extraction of structured clinical events. These events are the mentions of symptoms, findings, disorders, medications and procedures in text.

Secondly, I investigate if this novel method can be applied to the task of 'clinical coding' - the task of summarising patient admissions / episodes / spells into structured codes summarising the patient 'primary condition', comorbidities and corresponding interventions administered. Importantly, I ask if current datasets used to train clinical coding NLP methods are fit for purpose.

Finally, I focus on clinical text-to-text summarisation and ask how this is different from open-domain text summarisation tasks and what methods can be used to detect redundancy in text. I then ask if discharge summary generation can be automated through a novel NLP model. Discharge summary generation is another common example of clinical text-to-text summarisation that occurs during each in-patient hospital stay.

1.2 Ethical Considerations

Clinical domain data, especially the free-text portion of the EHR, is highly sensitive and often contains sensitive personally identifiable information (PII) requiring careful, controlled access. Examples of PII in EHR text could be patient names, addresses, phone numbers, email addresses, health and social care histories, personal family details and further 3rd party details relating to healthcare providers, family, friends etc. that should be kept private.

All research within this thesis has leveraged real-world closed clinical free-text data as well as openly available data such as MIMIC-III [64] – a large real-world, clinical dataset that is available after an ethics training course. Closed data is leveraged from data platforms deployed on-site at specific partner hospitals allowing for controlled access to data. King's College Hospital NHS Foundation Trust data was made approved by the London South East Research Ethics Committee approval (reference 18/LO/2048), South London and Maudsley NHS Foundation Trust CRIS data was approved for research by the National Research Ethics Service, South Central – Oxford C (08/H06060/71). University College London Hospitals Foundation Trust provided limited data and was under COVID-19 ethics for Chapter 3 research.

1.3 Thesis Statement / Contributions

This thesis is a combination of work published as part of the thesis and extensions to those works. Where prior published work is included in the thesis, it is clearly marked within the section name. Thomas Searle is the first, or joint first author (this is marked on the publication title page where this is the case) for all work included in the thesis. Each published paper includes its own specific background and methods sections directly applicable to the work presented.

1.4 Chapter Summaries

A summary of each chapter follows:

Chapter 2: Methods and Experimental Data

The thesis focuses on the development and application of Natural Language Processing (NLP), a subset of AI methods, applied to clinical text corpora for the purposes of summarisation. This chapter introduces text-analysis, NLP and the broad range of existing methods that are used throughout the thesis to encode text, infer structure, discern meaning and finally build summaries. I also introduce CogStack [57], an ecosystem of technologies used throughout the thesis that provides the means to 'unlock' clinical free-text that would otherwise be locked away within a given hospital site EHR system. The CogStack ecosystem is a crucial enabler in the development and evaluation of the NLP methods presented throughout the thesis, and has also enabled unique collaborations and opportunities cross-functionally from clinical research to service audit and improvement work that would otherwise not be possible. This chapter provides sufficient context for future chapters that present novel methodological contributions, their applications and the wider impact my work has had.

Chapter 3: Summarising EHRs through Named Entity Recognition and Linking

This chapter focuses on the development, deployment testing and continued usage of a novel Named Entity Recognition and Linking (NER+L) framework for effective and efficient structuring of EHR free-text narratives using any predefined clinical terminology. Importantly, the algorithm and output AI models are integrated into an associated workflow allowing clinicians and expert users to validate refine and extend models if and when the need arises. Two published papers are contained within this chapter demonstrating the core algorithm, workflow and associated annotation tool. The first paper [72] is published in the journal *Artificial Intelligence in Medicine (2021) vol. 117* and encompasses a large body of work across multiple hospitals testing the method performance across clinical terminologies. The second paper [136] is published *In Proc. Empirical Methods of Natural*

Language Processing (2020) as a system demo. We make all software source code, pre-trained models and tutorials available to the research community for replication, usage and further development of the work. Links to source code are listed in Section 6.2. These contributions are used throughout the following chapters.

Chapter 4: Clinical Coding as Summarisation

Clinical coding is an important administrative process in care delivery. Clinical coders are tasked with the assignment of diagnosis and procedure codes to patient episodes effectively *summarising* complexity of the patient condition and care provided. This chapter describes the coding process, how AI-assisted tools could be used in the coding workflow, a review of recent work from the clinical NLP communities working in the area of automated coding and importantly the translational gap between these contributions and real clinical coding practise. I include a published paper [134] appearing in *In Proc. of the 19th SIGBioMed Workshop on Biomedical Language Processing* describing the development of a silver-standard dataset to enrich a clinical coding dataset frequently used to develop and assess AI models, providing empirical support for why the translational gap remains. Experimental setup code and output silver-standard dataset are made available to the research community as listed in Section 6.2.

Chapter 5: Textual Summarisation of EHR Text

Generating textual summaries from *source* texts are a daily task performed by all healthcare professionals. This chapter first compares this continual textual summarisation process with open-domain tasks such as news summarisation. I then present an analysis of existing EHR text data clearly differentiating the open-domain textual summarisation problem from the clinical space estimating the redundancy in open-domain and clinical text. This paper [135] appears in the *Journal of Biomedical Informatics (2021) vol. 124*. Secondly, I present the development of a novel ensemble method for the textual generation of *Brief*

Hospital Course (BHC) sections from source notes. This summarisation process occurs at every inpatient discharge and therefore presents a varied, challenging and impactful potential use case for the deployment of textual automated summarisation models. I draw on methods presented in earlier chapters to develop the novel methodology and present a range of empirical results that explore the problem of BHC summarisation. This paper has been published in *Journal of Biomedical Informatics vol. 141*. Code and models are made available as listed in Section 6.2.

Chapter 6: Discussion, Conclusions, Future Work

I close the thesis by discussing the impacts of my work, bringing together the viewpoints of *EHR summarisation*. The future impact of my work, not only on the research community but for the wider communities of clinical information technology.

Chapter 2

Methods

This chapter firstly introduces vectorisation of natural language text and the Word2Vec method. This is the transformation of text into a computer interpretable set of semantic features that enable modern Machine Learning (ML) based Natural Language Processing (NLP) methods that we use throughout the thesis. We briefly review ML concepts and methods that are used throughout the thesis, especially highlighting where these methods have been previously applied to the analysis of clinical free-text and for summarisation. We then review the Transformer neural network architecture. We close this chapter by introducing CogStack [57], an ecosystem of technologies that 'unlocks' health record data for data discovery, clinical analytics and research. We describe how this core technology has supported and offered a framework to enable the work carried out in this thesis and many other research projects.

A full and detailed explanation of language, text-mining and text analysis and the basis for vector representations of text is provided in Appendix A.1.

2.1 Dense Vector Representations

Before we discuss these methods, we introduce some linear algebra notation for ease of reading. Scalar values will be represented by a lowercase letter, i.e. a, b. Vectors will be $\vec{a}$, $\vec{b}$ and matrices in capitals such as A, B.

We can reuse the idea of a co-occurrence matrix to find dense, distributed semantics vector representations of each token in vocabulary V for some chosen dimension size d and context window size s. Importantly, $d \ll |V|$ forcing both the representations to be efficient and being more usable downstream. We will now review Word2Vec in detail, as the ideas are fundamental to the development of modern methods in NLP and of our methods presented in the following chapter.

2.1.1 Word2Vec

The objective of Word2Vec is to compute the likelihood of a model θ, a vector of real numbers, given all possible sliding windows over all texts $\hat{T}$. There are two alternative algorithms in Word2Vec as illustrated in Figure 2.1 for token vector optimisation. Continuous-Bag-Of-Words (CBOW) predicts the centre token via all context words, whereas Skip-Gram predicts the target token via the target's context vectors.

Initially, each $w_i \in V$ token is assigned two randomly initialised vectors $\vec{v}_{i_{ctx}}$ and $v_{\vec{i}_{cntr}}$ both of size d. A sliding window of size $2s$ considers each token of a text t_j, with s tokens to the left and right of token w_i.

The likelihood is formally:

$$L(\theta;T) = \prod_{j=0}^{\hat{T}} \prod_{i=-s,i\neq 0}^{s} P(w_{i+j}|w_i;\theta) \tag{2.1}$$

Taking the negative, log-likelihood provides numerical stability:

$$J(\theta) = \sum_{j=0}^{\hat{T}} \sum_{i=-s,i\neq 0}^{s} logP(w_{i+j}|w_i;\theta) \tag{2.2}$$

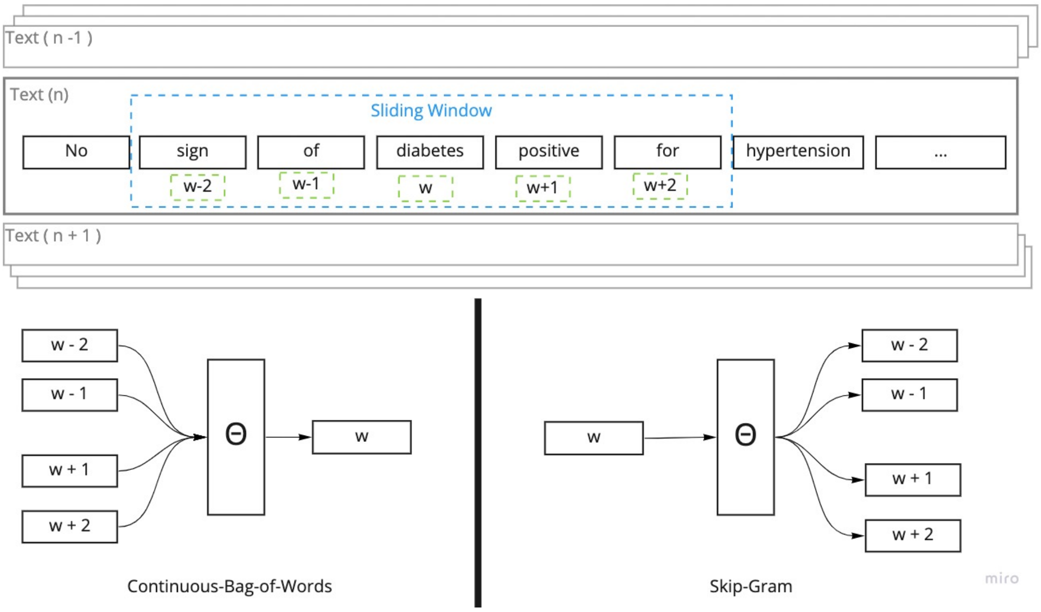


Fig. 2.1 The Word2Vec optimisation approaches for finding dense, distributed semantics token vectors from a text corpus - reproduced from [93]

θ is the stack of token vectors: $(v\vec{0}_{cntr}, \ldots v\vec{n}_{cntr}, v\vec{0}_{ctx}, \ldots v\vec{n}_{ctx})$. The probability term of a token w_i and each surrounding context token w_j with $i - s \leq j \leq i + s, i \neq j$ can be written as:

$$P(w_i|w_j) = \frac{\exp(v_{i_{cntr}}^{\intercal} v_{j_{ctx}})}{\sum_k^V \exp(v_{k_{cntr}}^{\intercal} v_{k_{ctx}})} \tag{2.3}$$

We can now see that the vector product of any v_{cntr} and v_{ctx} vectors that are similar will increase, the numerator term, and conversely for vectors that are dissimilar. The denominator scales the numerator so the probabilities of all tokens given their contexts sum to 1. Given equations 2.2 and 2.3, we can use a generic optimisation method, namely gradient descent, to iteratively modify the model parameters θ. In Section 2.2, we will cover gradient descent and optimisation methods that are broadly used in machine learning for NLP and training of neural networks.

For now, the current state of the model at step s, θ_s can be updated to θ_{s+1} by:

$$\theta_{s+1} = \theta_s - \alpha \nabla J(\theta_s) \tag{2.4}$$

Where α is some small real number that scales the amount we update θ from the gradient of the negative log likelihood, $J(\theta)$. The partial derivatives of $J(\theta)$ can be formulated wrt. $v_{\vec{i}_{cntr}}$ and $v_{\vec{j}_{ctx}}$ for each i and j pair providing:

$$\frac{\partial J(\theta)}{\partial v_{i_{cntr}}} = - v_{j_{ctx}} + \sum_{k}^{V} P(w_k|w_j) v_{j_{ctx}} \tag{2.5a}$$

$$\frac{\partial J(\theta)}{\partial v_{j_{ctx}}} = \begin{cases} -v_{i_{cntr}} + \sum_k^V P(w_k|w_j) \cdot v_{i_{cntr}} & , j \in (i-s, \ldots, i+s) \\ \sum_k^V P(w_k|w_j) \cdot v_{i_{cntr}} & , j \notin (i-s, \ldots, i+s) \end{cases} \tag{2.5b}$$

These partial derivatives update all parameters of θ at each sliding window. This is computationally expensive as for the majority of time most tokens w_j and their corresponding v_j will not be within the small context window given real text documents. An optimisation method named negative-sampling can be used, where a pre-built unigram table updates a random number of words not in a given context window instead of looping over all vocabulary words each sliding window. Prior work empirically found between $5-20$ samples at each sliding window was sufficient for model optimisation [92].

The original algorithm also includes further optimisations and parameters to adjust how these optimisations function. This includes a sampling rate, to adjust if a given token w_i is to be included in the vocabulary at all, depending on its number of occurrences in T or its marginal probability $P(w = \text{<some token>})$. This is important for common tokens that appear in many contexts such as ‘the’ or ‘a’ as there is very little information to be gained either from using these common words as a context token i.e. w_j or a center token, i.e. w_i.

A further optimisation method allows for common multi-token phrases to be detected according to the bigram, i.e. two token pair, counts of tokens plus a fixed discount factor δ, adjusted by their unigram i.e. single token, counts. This means common bigrams that appear often but above the parameter threshold δ are merged together. This process was repeated 2-4 times before token vector training [92].

The resulting vectors produced by Word2Vec were empirically shown to have embedded a notion of similarity. This similarity can be observed by taking the cosine-similarity of a given pair of vectors:

$$cos(v_i, v_j) = \frac{v_i \cdot v_j}{||v_i||\,||v_j||} \tag{2.6}$$

Vectors with all components equivalent will produce a similarity score of 1, with opposing components i.e. negative, will score a -1. The Word2Vec work [92] showed that over numerous tasks that the vectors could capture semantic and syntactic features. Word2Vec also demonstrated the compositionality of learnt vectors through simple component-wise addition and subtraction. A famous example being: $v_{king} + v_{women} - v_{man} = v_{queen}$.

2.1.2 Beyond Word2Vec

From 2013-2019, Word2Vec and its derivatives were the de-facto standard of producing distributed semantic vector spaces for words. Its performance, speed, low compute requirements and abilities to be run over any corpus with easy re-use and sharing to further tasks allowed a wealth of exploration [47].

Throughout that time a flurry of research provided enhancements such as: Global Vectors (GloVe) [104]: a method that used the global features of a non-zero token co-occurrence matrix alongside Word2Vec style localised token to token contexts, FastText: that included improved performance through the use of subword embeddings [19], sentence and entire document embeddings [78], low-resource languages embeddings [62], and even special character - emoji embeddings [35].

Despite these contributions, a couple of major hurdles were still present. Tokens that did not appear in the Vocabulary during training could not be modelled. This is known as out-of-vocabulary (OOV) occurrences, which were often simply removed from any texts during processing. FastText [19] proposed an approach, via the addition of sub-word vectors to form OOV tokens, but as the sub-word tokens did not take into account the context during inference, the constructed vectors still left room for improvement.

An associated and broader issue with these embeddings were that a token w_i and its vector v_i were represented by one ‘fixed’ or ‘static’ vector irrespective of the context under which w_i is used. This is significant as many languages, including English, frequently have equivalent words with multiple meanings derived from how they are used in context. These are known as homographs, e.g.: ‘the patient was admitted to hospital’, and ‘they were asked to be patient’. Ideally, two distinct vectors would have been learnt for *patient* referring to a hospital attendee, and the adjective for tolerance.

Contextualised word embedding models such as ELMO [105] and BERT [33] and GPT [115] are now state-of-the-art and have essentially superseded Word2Vec and the static embeddings that are calculated on a per token basis. Instead these approaches use the surrounding context of each word to generate a context aware vector representation. These NLP models, the underlying architectures, and their downstream use-cases will be discussed in detail throughout Section 2.5.

2.2 Machine Learning for NLP

We will now review machine learning (ML) methods relevant to the processing of text. ML methods are general purpose methods for the optimisation of a model’s parameters θ wrt. to data X, to perform a task without explicit programming for each input within the domain of X. Figure 2.2 shows how ML methods fit within wider methodological fields of natural language processing (NLP) and artificial intelligence (AI).

Artificial Intelligence (AI) is the field of computer science that aims to emulate or even create human-level intelligence across input modalities and associated tasks [40]. Throughout this work we assume AI and artificial narrow intelligence (ANI) are equivalent. The methods we will discuss are all *narrow* as they generally function with one data

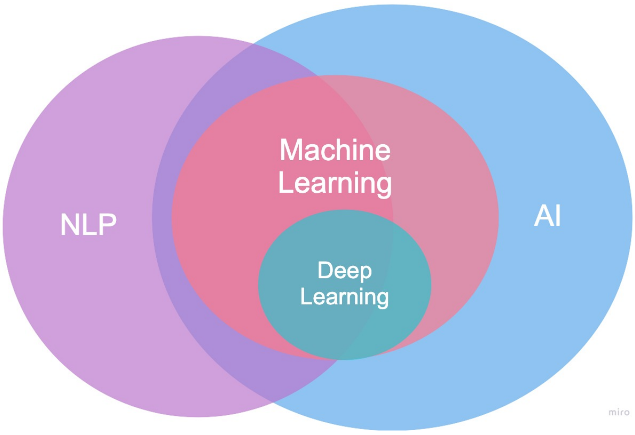


Fig. 2.2 Venn diagram for the classifications of method groups.

modality, i.e. text, and for one task, i.e. classification or clustering or regression. The debate regarding the very recent methods potentially providing a path towards artificial general intelligence (AGI) is outside of scope of this methods review.

NLP is broader than just the application of ML to text. For example, corpus linguistics the study of language via the use of computational methods using whole text corpora could be considered NLP but not ML [148]. The application of rule-based systems in NLP has a long history [9, 73] and could be considered NLP and AI but not ML. ML methods exist outside of NLP and the converse is also true. Finally, deep learning (DL) methods are a subset of ML and are again are applicable to both in and outside of NLP.

Machine learning methods can largely be split into three algorithm groups. The first, supervised learning, requires the use of a 'labelled' dataset to learn given the optimal mapping function from input data to the labels. These learning methods includes problems as regression, classification, and specific to NLP problems include language modelling and textual summarisation generation. The second, unsupervised learning, requires only the input data i.e. without labels, and learns the optimal function to recognise latent structure within the data. These methods tackle problems such as clustering and anomaly detection and data generation. The third, reinforcement learning includes methods to learn the optimal action given a current world view, sensory inputs and goals. These methods are used in game playing, robotics and autonomous vehicles.

ML methods may be referred to prediction modelling in the literature as they can can generate predictions on potentially previously unseen data X', for some target variable (i.e. supervised models), identify some latent structure or clusters (unsupervised models), or model some state/action reward space for goal directed behaviours (reinforcement learning).

The majority of the thesis focuses on the development and application of supervised learning methods for EHR summarisation. This includes *self-supervised* methods, a recent definition to describe approaches that infer the label directly from dataset being used but still use standard optimisation techniques found in supervised learning. These may also be referred to as *unsupervised* in the literature [33, 114], as the label or output signal is not manually provided alongside the input data.

2.3 Supervised Learning

Binary classification problems can be modelled as a supervised learning task. Within healthcare, this could be a prediction of a patient to attend a follow-up appointment, are to be discharged over some set time period, or to be readmitted over some time period.

Formally, given input data X and labels $Y \in \{0,1\}$, a supervised learning method finds some function $f(X) = Y$. θ are the parameters of f. A common method couple fit logistic regression model using a tf-idf matrix (reviewed in Section A.5.1) as input, θ is optimised to find the *important* words for classifying an input example into their respective classes and $\theta \in \mathbb{R}^{|V|}$. Our aim then is to find the optimal θ that brings us as close to Y given inputs X. The logistic or sigmoid function $s_\theta(x)$ squashes all values into the range [0,1] and therefore is a well formed probability. The probability of our label set is then:

$$P(Y=1|X) = s_\theta(x) = \frac{1}{1+\exp(-\theta^\intercal \cdot \vec{x})} \tag{2.7a}$$

$$P(Y=0|X) = 1 - P(y=1|x) \tag{2.7b}$$

Initially, we choose a values for all i in θ, either random or via some other means, and compute how 'far' we are from the optimal solution. This is also known as the 'loss':

$$J(\theta) = -\sum_{i}^{|X|}[y_i \cdot log(s_\theta(\vec{x_i})) + (1-y_i)\cdot log(1-s_\theta(\vec{x_i}))] \tag{2.8}$$

Similar to Equation 2.2, our definition of the Word2Vec loss, we take the negative log-likelihood for numerical stability for our logistic regression loss. The first half of the term can be interpreted as for when $y_i = 1$ and the second half for when $y_i = 0$. The opposite sides are then cancelled out by the multiplication of 0.

2.3.1 Minimization via Gradient Descent

To minimise $J(\theta)$ we use gradient descent, an iterative algorithm briefly discussed in our discussion of Word2Vec and in Equations (2.5) and (2.4). Gradient descent uses the partial derivative wrt. each item within θ.

$$\nabla J(\theta) = (\frac{\partial J(\theta_1)}{\partial \theta_1}, \ldots, \frac{\partial J(\theta_n)}{\partial \theta_n}) \tag{2.9}$$

Each partial derivative is the rate of change of the loss wrt. the parameter. Alternatively, the effect each parameter has on the loss and therefore how well θ fits as a set of parameters for f given our data X. With our example logistic regression model and an input tf-idf feature matrix we have $|V|$ partial derivatives that we compute. Each θ_j partial derivative is:

$$\frac{\partial J(\theta_j)}{\partial \theta_j} = \sum_{i}^{|X|} x_i^j \cdot (s_{\theta_j}(x_i^j) - y_j) \tag{2.10}$$

Each derivative can be interpreted as the weighted sum over all sigmoid normalised inputs X for scalar features j compared with the corresponding outputs from Y. Computing the change in θ at step t can now be computed using Equation 2.4, where a hyperparameter α scales the effect the derivative has on the next state θ_{t+1}.

Gradient descent converges when $\nabla J(\theta) \simeq 0$. Gradient descent will converge to a local or global minimum of f depending on the convexity of $J(\theta)$. A simple curve, e.g. $f(x) = x^2$ is an example of convex functions where gradient descent will find the global minimum. Local minimum for a non-convex loss function are 'pockets' of a loss function that approach 0 but are not globally the best values for θ. Larger α allows gradient descent to reduce the chance of becoming stuck in local minima but may also prevent the convergence completely as the gradient 'bounces' out of the minima.

Computing the derivatives wrt. each parameter requires computing over the entire dataset of X at each iteration of the algorithm. For large X and large θ, this can be time consuming especially as the $\nabla J(\theta)$ may already be informative after a subset of X. Stochastic gradient descent (SGD) uses a single sample for each new θ_{t+1}. This can produce an erratic loss function that moves in different directions due to some training samples in X indicating where components in θ should be changed. Mini-batch gradient descent is a middle ground method that uses a 'batch', some defined size of samples, e.g. 10 or 20, in the calculation of $\nabla J(\theta)$. This balances the noise from individual samples that may move θ into the wrong direction in the SGD case, but aims to converge quicker as θ_{t+1} is computed after each small mini-batch.

Once converged our θ parameters can be used for any new dataset X' to make predictions y'. Methods to choose the previously discussed hyperparameters, feature-set size i.e. the size of θ, and measuring performance are discussed in Section 2.4.

2.3.2 Multi-Class Classification and Softmax

Our logistic regression model can be generalised for a multi-class classification problem. For example, the risk of readmission or mortality could be split into 3 classes of days 0-2, 3-6 or 7+. In a multi-class problem with C classes, Θ is a matrix of size $n \times C$. Softmax regression and the binary classification form of logistic regression are *linear* models, as they assume the function $f(X)$ can be expressed with some linear combination of the feature space. With softmax, this linear combination is at the class level for each column vector in Θ.

Many problems however cannot be expressed as a linear combination of their input features. This inability for a model to adequately express relationships between inputs and outputs can be seen during model training and validation and testing. These stages are discussed in more detail in Section 2.4.

2.4 Standard ML Experimental Methodology

In any supervised learning, and most ML modelling exercise, the primary aim is often to output a high performing, generalisable model. During model development a dataset X is often split into $X_{train}, X_{val}, X_{test}$ with ratios in the region of 80/10/10. This provides the majority (80%) of the original input data for model training, and optimisation of actual model parameters θ. Then 10% for model, and hyperparameter improvement using the validation set, and the final 10% for final testing of the model. Practitioners should only report results of their models using the test set and never perform hyperparameter optimisation with the test set.

A model that has converged, i.e. the loss is no longer decreasing after a number of update iterations, but still outputs poor train, validation and even test set performance can be said to *underfit* the data. In our logistic regression example, this could mean that the tf-idf matrix does not sufficiently capture the relationships between words or phrases for the task and / or the model itself lacks the parameters to capture those relationships.

Overfitting occurs if a model performs well on the train and validation sets and the training loss either is or approaches 0, but the test set performance is not good. This suggests the model has too closely fit to the specific idiosyncrasies of the train and validation data and therefore is not generalisable to the unseen test data.

Regularisation methods can be used to limit and reduce the likelihood of overfitting occurring. For a logistic regression this could include techniques such as LASSO [128], Ridge [53] regression that add penalty terms to loss function $J(\theta)$, applying the intuition of Occam's Razor, that a simpler i.e. a smaller number of parameters, are preferred over complex so large complex θ increase the loss and smaller θ the reverse.

2.4.1 ML Evaluation Metrics

A high performing model has the appropriate number of parameters configured correctly to achieve *good* performance on the test set data. However, model performance can be optimised with specific measures in mind. For example, our scenario of hospital mortality prediction from patient notes may favour true-positive results (i.e. the model makes a prediction of death that is correct in the test set) so that patients at high risk are not missed. This may then raise the false-positive (i.e. the model predicts death but this is incorrect), creating false alarms, but is still favourable compared to missing a true-positive sample.

Balancing these metrics is a model development and testing choice, and is often context specific. ML model research often reports a variety of performance metrics. These are calculated via the test set and the counts of true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN) of each test set item and the associated model prediction. Metrics include:

- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$: a common metric, that is the proportion of both positive and negative predictions normalised by the sum of all possible prediction states.
- Specificity: $\frac{TN}{TN+FP}$: the proportion of correct negative predictions of total predictions. E.g. if we have 10 negative samples and the model predicts 5 negatives correctly we have 50% specificity.
- Sensitivity or Recall: $\frac{TP}{TP+FN}$: the proportion of positive predictions of total predictions. E.g. if we have 10 positive samples out of 20 in a dataset and the model only predict 5 out of the 10 correct, we have 50% recall.
- Precision: $\frac{TP}{TP+FP}$: the proportion of positive predictions that are correct. E.g. if the model only makes 2 positive predictions and 1 is correct we have 50% precision.
- F1: $\frac{2 \cdot (Precision \cdot Recall)}{Precision+Recall}$: The harmonic mean between precision and recall. F1 scores are often reported as they include both precision and recall, and therefore offer a good view of classifier performance irrespective of data specific issues that could skew performance in favour of recall or precision. E.g. if a class only appears once or a few times, achieving high recall might be trivial for a classifier that has many false positives.

Multi-class classification problems can compute a precision, recall etc. score per class. Often two alternative averaging scenarios are used to report a single score to compare model performance. *Macro* averaging averages the performance across each individual class score. *Micro* averaging gathers all predictions across all classes and averages via the above definitions. For problems with large class imbalances micro and macro averages can tell varying stories of model performance.

Textual generation methods within NLP are particularly difficult to reliably automatically evaluate. A human being can compare a reference and generated text to assessing equivalence using a variety of measures. For example, are the same topics, entities or words being used in the texts? Are texts factually equivalent? Are the style and tone equivalent? Embedding these elements into an evaluation metric are active research areas [150]. A simple and often used metric is ROUGE [81], the Recall-Oriented Understudy for Gisting Evaluation, is an n-gram based metric that directly compares tokens. ROUGE-1, ROUGE-2 are the unigram and bigrams that overlap between the reference text and model generated texts. ROUGE-L is the longest common substrings between the texts. All ROUGE scores are calculated and reported with using the precision, recall and F1 calculations defined above. Text generation is more difficult to automatically evaluate than for binary or even multi-class classification, as unigram, bigrams or common sub-sequences are only proxy measures to a 'correct' prediction by a text generation model.

It is accepted that ROUGE has limitations for assessing effective text generation models [131], and recent work has introduced a variety parameterised evaluation metrics [166, 150] that suggest better alignment between model prediction score and manual human evaluated scores. These are presented and discussed in more detail in Chapter 5.

An introduction to artificial neural networks, first principle architectures (i.e. feed-forward networks) and sequence dependency architectures are provided in Appendix B

2.5 Transformer Models

The Transformer architecture [149], from the problem of natural language translation, uses *attention* for the majority of its representation learning, replacing the explicit sequential nature of the recurrent neural network entirely (described in Appendix B. Figure 2.3 shows the familiar encoder-decoder architecture where each encoder and decoder model is comprised of stacked Transformer blocks.

Transformer blocks are comprised of *multi-headed self-attention* computations. For the input sequence $X = (x_1, \cdots, x_n)$ the self-attention score is calculated via:

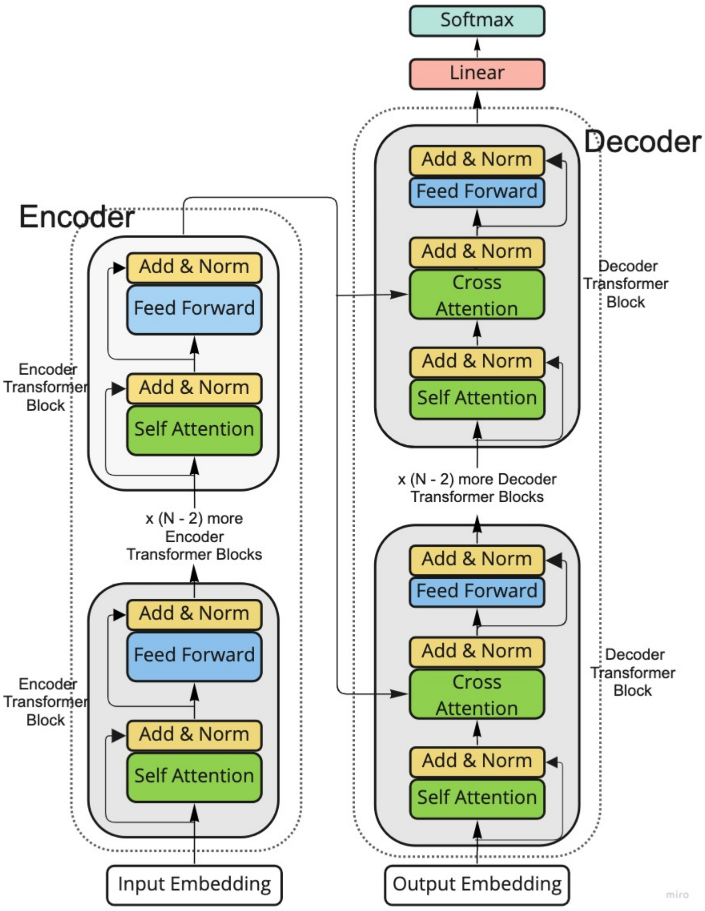


Fig. 2.3 Transformer encoder decoder architecture as presented in the original work [149] for the open-domain translation use case.

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V \tag{2.11}$$

where each Q,K,V are the projections of X via the weight matrices W_K,W_Q,W_V respectively. This calculation is called *dot-product attention* compared to *additive attention* discussed in the previous section. Dot-product attention allows for improved space and time efficiency as all computations can be carried out in parallel using optimised linear algebra software. The Transformer authors also extend this calculation to *multiple-heads*, so there are h many weight matrices $W^K_{1\cdots h}, W^Q_{1\cdots h}, W^V_{1\cdots h}$ allowing for the self-attention heads to focus on varying relevant relationships between inputs.

Figure 2.3 shows that the encoder model is built of stacked multi-headed-self-attention calculations followed by feed-forward layers. All inputs can be *attended* to by all other inputs, and all inputs to the self-attention calculation come from the previous block. The decoder model has some slight variations however. Firstly, the encoder output is fed into each decoder Transformer block, allowing any encoder inputs to be attended to giving the name *cross-attention*. Secondly, the decoder should only be able attend to all positions up to an including the position it is currently decoding. This is referred to an *auto-regressive* model in the literature and is only relevant during training as tokens to the right have yet to be decoded during inference. This is implemented by *masking* all decoder inputs to the right of the current position, i.e. setting to $-\infty$.

Layer normalisation [7] is applied after each self-attention and feed-forward layer providing regularisation at each sub-block of computation. These layers normalise the incoming inputs activations to have zero mean and unit variance, improving training convergence time.

Residual connections [50], or skip connections, allow the model to be much deeper, i.e. comprised of many more layers. These network connection structure were initially shown to be effective in the training of deep neural networks for computer vision models but were also subsequently shown to be beneficial with deep sequence models [155]. The intuition with residual connections are that activations in the lower layers can be used by successive deeper layers without having to explicitly *survive* the computations of layers between, therefore allowing for larger numbers of parameters in any given model whilst maintaining the flexibility of learning a range of relationships.

With the Transformer, all computations of a sequence are performed in a parallel with no explicit definition of sequence order. *Positional embeddings* are used alongside the input to provide the model knowledge of where tokens are in relation to the other tokens. These embeddings allow the projected inputs i.e. K, Q, V to maintain data related to relative positions in the sequence.

2.5.1 Transformer models: BERT

Arguably one of the most important NLP models recently, Bi-directional encoder representations from Transformers (BERT) [33] offered a multi-purpose model demonstrating state-of-the-art results across a range of benchmarks datasets such as SQuAD [117, 116], open-domain question answering, and GLUE [151], a multi-task dataset including single-sentence tasks, similarity / paraphrase tasks and inference tasks.

BERT consists of only the encoder as shown in Figure B.4. The work achieves these results through a large, deep configuration of Transformer encoder blocks, a large pre-training dataset, and carefully designed pre-training task. $BERT_{BASE}$ and $BERT_{LARGE}$ comprise of 110M and 340M parameters respectively. They are pre-trained using a corpus of circa 3B tokens then fine-tuned on each downstream task from SQuAD and GLUE. The authors demonstrate empirically that deep bidirectional pre-trained models can perform across a range of settings.

The experimental setup introduces the pre-training task of *masked language modelling*, where a token within a fixed window is masked and the model is tasked with predicting the token, allowing the model to use bi-directional representations from the left and right of the masked token. This is contrasted with *causal language modelling* where only the left hand side of a given input are used.

BERT is an important example of *Transfer Learning* that allows, the same base model to be re-used across a range of tasks demonstrating that the parameters have gained some *understanding* of text and natural language. BERT's impressive performance across a wide range of tasks prompted numerous derivative models and studies of how the architecture and in particular multi-headed self-attention learns the intricacies of language [123].

BERT's remarkable performance is also shown when used purely as an embedding layer, i.e. as a stage 3 Vectorisation stage of a text analysis pipeline (Section A.5). BERT and its derivatives provide context relevant embeddings for each input, allowing models to be aware of differences in context for words that would otherwise be represented equivalently using static embedding methods such as Word2Vec.

Empirically, BERT and derivative models, such as DistilBERT [127] and RoBERTa [82], have been tested in diverse areas such as clinical text normalisation [61] and disease prediction [118]. Prior work has also used the contextualised embeddings from BERT for Alzheimer's detection [134] and radiology report analysis [96].

2.5.2 Transformer models: GPT

Transformer encoder models such as BERT are pre-trained on masked language modelling task. Allowing the model to view inputs the left and right of the masked token. For text generation tasks, including summarisation, decoder only Transformer stacks that mask all inputs to the right of the current token are more suitable. Generative Pre-trained Transformer (GPT) [115] and its successors GPT-2 [114] and GPT-3 [21] are pre-trained on this causal language model task and have successively shown improved performance surpassing previous work each time. Remarkably, the neural architecture is the most part the same between each successive iteration, only the depth and amount of data available for pre-training is increased.

The latest iteration even showed impressive performance in few-shot and zero-shot learning scenarios where the dataset for pre-training contains either a very small number or no specific cases of the task being asked of the model. Both masked and causal language modelling are also referred to as self-supervised learning as the supervision signal, i.e. the label for a given input training data item is inferred allowing for the model to optimise itself across potentially billions of examples. These abilities for models to provide basis across a wealth of tasks has recently been coined as foundation models having widespread potential applications across diverse domains such as healthcare, law and education [20].

2.6 Enablers for Neural Model Success

The success of neural models can be attributed to a handful of technical advancements on both the hardware and software side. Large datasets to potentially train neural models have been accessible even in the clinical space since the early 90's [37]. However, neural model calculations both the forward and backward pass heavily rely on linear algebra calculations. Even optimised linear algebra code is limited by the fundamental architecture of the CPU.

Graphics processing unit (GPU) hardware originally designed for rendering graphics are now heavily used and configured for ML workloads. This enabled cheaper and improved massively parallelisable linear algebra calculations through the use of heterogeneous hardware CPU + GPU configurations supporting training of large models in hours instead of weeks. Further ML specialised hardware dedicated to neural network training and inference predominately optimised for the linear algebra heavy calculations present in neural models. Examples of these include the tensor processing unit (TPUs[1]) and the intelligence processing unit IPU[2].

Alongside the hardware to support experimentation, software improvements have allowed for quick and easy experimentation without specialised knowledge of the underlying

[1] https://cloud.google.com/tpu

[2] https://www.graphcore.ai/products/ipu

hardware. Software frameworks such as TensorFlow [87], PyTorch [102] and differentiable programming allowed for experimentation at an unprecedented rate and scale.

The culmination of hardware, software and the availability of mass data across many domains have lead to ongoing and expanding interest and success of predominately neural models in AI and clinical AI research.

2.7 Natural Language Understanding

Despite this enormous progress across the range of NLP tasks such as text classification, natural language inference, question answering and summarisation, we are still arguably far from natural language understanding where a model truly understands the text it has been trained upon or interrogating at inference time. Recent work suggests no model can truly understand language as the training data only provides form i.e. the text alone, and not the meaning behind what the text is referring to, or the meaning of the words and relationships between the language and the external manifestation of what the language is referring to [13]. Further work describes how generalised AI cannot be realised due to the lack of embodiment and in the language sense this refers to the model not being able to relate words to their reality, i.e. a person, a hospital or biological process as things that exist and the words that refer to them [40].

2.8 Challenges with Clinical Free Text

Working with clinical free-text is challenging due to a number of issues:

- Data Sensitivity: the data is highly sensitive, often describing in great detail personal information that must be kept private. These requirements often make working with clinical free-text difficult as large corpora are often inaccessible for substantial training of models and compute power is limited in constrained hospital settings.

- Technical language: the clinical domain is knowledge-rich, with many taxonomoies and ontologies for varying specialties, some of which are covered in future chapters. This highly specialised domain means working with text or summarisation or other use cases requires specialist domain-specific knowledge.

- Non-Standardization: EHR data can be collected at source via different source systems and authors. A single clinical note for a radiology scan is written by a different author, with a different task to accomplish compared to a nursing progress note with potentially 2 separate systems for data entry.

- Inconsistency of Available Data: clinical data often only covers a period of time in which a patient was admitted or attended an appointment, and is heavily reliant on stretched workforce staff to accurately remember and enter all relevant data. A patients may have no available data large periods of time if for example only secondary care EHR records are available and cannot be linked with primary care data.

- Incomplete / Error Prone: Unfortunately EHR systems have done little to assist with the recording of higher quality, less error-prone data. Instead they may have perpetuated errors through the use of functions such as copy-paste [122]

- Multi-modal: Clinical free-text often accompanies other modalities of data. This could be a radiological image, a structured tabular dataset from a lab, or a genome test result. Building a truly effective system, one that is remotely comparable to a current level of care provided by a modern day clinical multi-disciplinary team will require machine learning and AI models to be multi-modal in their accepted input.

2.9 The CogStack Ecosystem

Secondary and tertiary care providers offer certain specialties and areas of focus, which involves varying digital systems to support those activities. The large scale migration from pen and paper to electronic systems is relatively recent [14] and is often not consistent even within the same hospital, especially if a hospital is large and spread over multiple sites [37]. EHR systems and the wider footprint of systems that hold patient data relevant for primary i.e. direct patient care, and secondary (i.e. uses such as research) are often comprised of multiple distinct databases [57]. This presents difficulties even for the direct patient care scenario where clinicians must access multiple systems simultaneously for accurate patient information leading to frustrations, decreased productivity and even worse outcomes [90]. This problem is compounded by the more disparate systems that comprise a hospitals EHR system, and further still if the use case involves multiple patients. Even the initial hurdle of locating and extracting the raw patient data can make a research question infeasible.

CogStack is an open-source ecosystem of tools designed to support clinical informatics use-cases with EHR data data[3]. CogStack is positioned to be deployed alongside existing heterogeneous EHR databases and systems providing a single ‘data-lake’ store for downstream tasks.

Figure 2.4 shows a high level view of the areas of the ecosystem. CogStack is designed to be EHR, data type and data format agnostic. This allows all EHR data, (aside from images), to be ingested, harmonized and indexed into a single ‘data-lake’ source. This is the CogStack Data layer. Structured tabular data such as laboratory test results, observations and patient demographics can be ingested and harmonized alongside unstructured free-text data such as clinician admission, progress and discharge notes, radiology reports and clinic letters.

It is estimated that 80% of any given data [99] source is unstructured, but making such data searchable, structured and relevant for downstream use cases is non-trivial.

[3] https://github.com/CogStack/

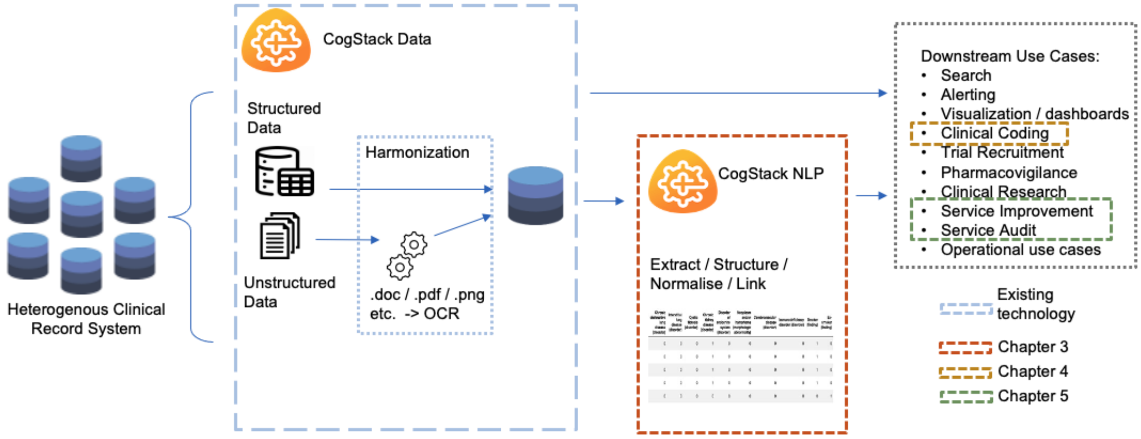


Fig. 2.4 A high-level diagram of the CogStack ecosystem of technologies and areas of focus for this thesis.

For the first use case of simply searching through unstructured data, CogStack provides a pipeline to harmonize and ingest unstructured data in a number of file formats, and data types. Once harmonized CogStack provides a real-time search capability via open-source ElasticSearch[4] technology to ingest and provide real-time free-text searching across potentially millions of documents.

A further open-source tool, kibana[5], is used for interactive, real-time visualisations and dashboards of clinical data from the CogStack data-lake. This thesis will focus on development of the ‘CogStack NLP’ section that aims to extract relevant clinical phenomena from the unstructured data, and link the extracted terms to a standardised clinical vocabulary, therefore normalising and structuring the data for downstream use.

CogStack is uniquely positioned as both an end-2-end ecosystem of tools supporting downstream use cases listed on the right of Figure 2.4, whilst also not strongly enforcing the use of all pipeline stages if existing systems already fulfill certain stages. This is enabled through the use of simple, decoupled web APIs, allowing for components to be deployed separately if required. This has been beneficial for end users that initially only want deploy sub components of either CogStack Data or CogStack NLP for small scale experiments.

[4] https://github.com/elastic/elasticsearch
[5] https://github.com/elastic/kibana

2.9.1 CogStack Deployments: Sources of Real-World Clinical Data

Mature CogStack deployments are in 4 large London secondary care hospitals, King's College London NHS Foundation Trust(KCH), Guy's and St Thomas' NHS Foundation Trust, University College London NHS Foundation Trust (UCLH) and South London and Maudsely NHS Foundation Trust (SLaM).

KCH, GSTT and SLaM CogStack deployments have ingested the entirety of their respective EPR systems across decades of administered care. This has unlocked the data from the previous disparate silos of specialist systems, databases, file formats and data types. UCLH use their CogStack deployment as a clinical research platform for a number of studies across departments and specialties. The CogStack deployment at SLaM, one of the largest mental health service providers in the UK, extensively use alerting, visualisation and dashboarding capabilities for population and case load management. All 4 deployments use different EPR systems, have varying areas of focus, speciality and targeted downstream use. This demonstrates the versatility of the ecosystem and the need for such tools to unlock the data within hospital settings that are often inconsistent between sites.

Further CogStack deployments can be found both nationally, here in the UK, and abroad in the Netherlands (UMC Utrecht) and Australia (Monash Health Partners). Further users of specific components of the ecosystem can also be found in the United States and India.

We will reference KCH, UCLH and SLaM CogStack deployments within later chapters as our primary sources of clinical data. Specific details for each dataset are available within the experimental setup section of each paper referenced within the thesis. Our work also makes extensive use of MIMIC-III [64]. A large, freely available, US based, real world ICU dataset collected between 2001-2012 covering circa. 53k admissions.

Figure 2.4 shows where the following chapters of this thesis will focus. Chapter 3 focuses on the development of NLP methods for the extraction, linking and normalisation of unstructured clinical data to existing clinical terminology. A form of summarisation that is frequently performed by users of free-text clinical data even during direct patient care, for example understanding what chronic diagnoses a given patient has from historical notes. I will introduce MedCAT, a toolkit for developing these NLP models explaining technical details and the wider impact and research outputs this toolkit has supported within the CogStack ecosystem.

Chapter 4 then considers how these methods can be used specifically for an important downstream use case of 'clinical coding', an administrative process that assigns specific codes patient *episodes* identifying the diagnoses and interventions received by the patient. A globally used process for the administration, planning and reimbursement of care, it is often performed manually and is therefore error-prone and difficult to scale.

Finally, Chapter 5 considers a service audit use case, again using CogStack Data and CogStack NLP models, to consider how automated text-summarisation could be used to one-day mitigate the negative effects of excessive EPR usage.

Chapter 3

CogStack NLP: Extract, Normalise, Structure Clinical Notes

During direct patient care clinicians do not read a patients prior medical notes ‘cover-to-cover’ as one would a book. Electronic patient record (EPR) systems that house electronic health records (EHRs) are organised into sections that allow for easier access to *relevant information* [49]. Clinical notes also have metadata specifying the note type to identify the clinical speciality e.g. radiology, emergency, neurological etc, the stage in care, e.g. admission, progress, discharge or if a note is a referral or intended for another care setting e.g. GP letter. The *relevant information* for direct patient care can vary depending upon the scenario, but often includes identifying current diagnoses, assessing symptoms or findings and determining the next best course of action such as a procedure, test or scan or assessing how the prior intervention was received. For even relatively simple cases patient free-text notes are rich sources of data offering detail not available in the structured data.

For secondary use cases, such as clinical research, identifying this *relevant information* manually would be extremely time-consuming and error prone. Using automated methods to identify entities within the text is well studied and is often an important step in unlocking further use cases of the unstructured portion of the clinical record [6, 130].

3.1 Clinical Terminologies

Within the healthcare domain there has been considerable efforts in creating standardised terminologies supporting knowledge management, data consistency and decision support [16].

The agreed standardised clinical terminology for the NHS is the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT). It is the most comprehensive, multilingual clinical healthcare terminology globally [125]. SNOMED CT provides a base international release translated into multiple end-user languages with additional country specific additions overlaid. At its core the terminology consists of *concepts* and *relations*. Concepts encapsulate a clinical relevant entity, such as a diagnosis, a finding or symptom. Concepts are organised into hierarchies so the concept *Myocardial Infaction* is uniquely identified by the SCTID (22298006) and has multiple parent concepts including Myocardial Disease (57809008) and multiple children concepts including Cardiomyopathy (85898001). Concepts also include further textual alternative names or synonyms alongside the full concept name. Relations provide a mechanism to link concepts with one-another according to specific relation types. These include relations such as *is a* (116680003), *causative agent* (246075003) and *associated finding* (246090004) etc.

SNOMED CT is designed to model clinical scenarios, and therefore is often not directly used in the administrative part of healthcare delivery. Further terminologies such as International Classification of Diseases (ICD-9, ICD-10 and the imminent ICD-11) and OPCS-4 classification for procedures and interventions are used in the UK for the task of clinical coding. This secondary use of clinical data is discussed further in Chapter 4. The Huaman Phenotype Ontology (HPO) aims to model phenotypic properties of human disease including genomic phenotypes [69]. RxNorm is another terminology that provides a hierarchical set of clinical drugs, often used and integrated into pharmacy and drug management systems [16]. Specialist terminologies for specific disciplines such as Radiology Lexicon (RadLex) for radiology [77], and the National Cancer Institute thesaurus (NCIt) ontology[1] for cancer.

Larger ontologies such as SNOMED-CT include mappings between alternative ontologies, allowing concepts to be mapped to SNOMED CT for example from ICD and OPCS. However, as these terminologies are designed with different aims some mappings can result in one-to-many or many-to-one mappings. This is discussed more in Chapter 4.

3.2 Medical Concept Annotation Toolkit (MedCAT)

Given the considerable efforts in creating comprehensive biomedical ontologies, methods to automatically identify, extract and link (NER+L) free-text spans to one or more of these ontologies is a well established area of research [6, 130].

I now introduce published novel work of an open-source toolkit for the automated NER+L task on clinical free-text. This published work details the technical contributions, demonstrating the state-of-the-art empirical performance of our methodology and associated workflow. Initial algorithm development and testing was performed by Z. Kraljevic. Together with ZK, I was responsible for further development of the toolkit and surrounding workflow, gathering of supervised training data, testing of the toolkit and manuscript writing.

[1] https://github.com/NCI-Thesaurus/thesaurus-obo-edition

Artificial Intelligence In Medicine 117 (2021) 102083

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Multi-domain clinical natural language processing with MedCAT: The Medical Concept Annotation Toolkit

Zeljko Kraljevic [a,1], Thomas Searle [a,f,1], Anthony Shek [c], Lukasz Roguski [b,d,h], Kawsar Noor [b,d,h], Daniel Bean [a,b], Aurelie Mascio [a,f], Leilei Zhu [d,h], Amos A. Folarin [a,d,f], Angus Roberts [a,b,f], Rebecca Bendayan [a,f], Mark P. Richardson [c], Robert Stewart [e,f], Anoop D. Shah [b,d,h], Wai Keong Wong [d,h], Zina Ibrahim [a], James T. Teo [c,g], Richard J.B. Dobson [a,b,d,f,*]

[a] *Department of Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK*
[b] *Health Data Research UK London, University College London, London, UK*
[c] *Department of Clinical Neuroscience, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK*
[d] *Institute of Health Informatics, University College London, London, UK*
[e] *Department of Psychological Medicine, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK*
[f] *NIHR Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London, London, UK*
[g] *Department of Neurology, King's College Hospital NHS Foundation Trust, London, UK*
[h] *NIHR BRC Clinical Research Informatics Unit, University College London Hospitals, NHS Foundation Trust, London, UK*

ARTICLE INFO

Keywords:
Electronic health record information extraction
Clinical natural language processing
Clinical concept embeddings
Clinical ontology embeddings

ABSTRACT

Electronic health records (EHR) contain large volumes of unstructured text, requiring the application of information extraction (IE) technologies to enable clinical analysis. We present the open source Medical Concept Annotation Toolkit (MedCAT) that provides: (a) a novel self-supervised machine learning algorithm for extracting concepts using any concept vocabulary including UMLS/SNOMED-CT; (b) a feature-rich annotation interface for customizing and training IE models; and (c) integrations to the broader CogStack ecosystem for vendor-agnostic health system deployment. We show improved performance in extracting UMLS concepts from open datasets (F1:0.448–0.738 vs 0.429–0.650). Further real-world validation demonstrates SNOMED-CT extraction at 3 large London hospitals with self-supervised training over ~8.8B words from ~17M clinical records and further fine-tuning with ~6K clinician annotated examples. We show strong transferability (F1 > 0.94) between hospitals, datasets and concept types indicating cross-domain EHR-agnostic utility for accelerated clinical and research use cases.

1. Introduction

Electronic health records (EHR) are large repositories of clinical and operational data that have a variety of use cases from population health, clinical decision support, risk factor stratification and clinical research. However, health record systems store large portions of clinical information in unstructured format or proprietary structured formats, resulting in data that is hard to manipulate, extract and analyse. There is a need for a platform to accurately extract information from freeform health text in a scalable manner that is agnostic to underlying health informatics architectures.

We present the Medical Concept Annotation Toolkit (MedCAT): an open-source Named Entity Recognition + Linking (NER+L) and contextualization library, an annotation tool and online learning training interface, and integration service for broader CogStack [1] ecosystem integration for easy deployment into health systems. The MedCAT library can learn to extract concepts (e.g. disease, symptoms, medications) from free-text and link them to any biomedical ontology such as SNOMED-CT [2] and UMLS [3]. MedCATtrainer [4], the annotation tool, enables clinicians to inspect, improve and customize the extracted concepts via a web interface built for training MedCAT information extraction pipelines. This work outlines the technical contributions of

[*] Corresponding author at: Department of Biostatistics and Health Informatics Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK.

E-mail address: richard.j.dobson@kcl.ac.uk (R.J.B. Dobson).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.artmed.2021.102083
Received 7 October 2020; Received in revised form 24 March 2021; Accepted 28 April 2021
Available online 1 May 2021
0933-3657/© 2021 Elsevier B.V. All rights reserved.

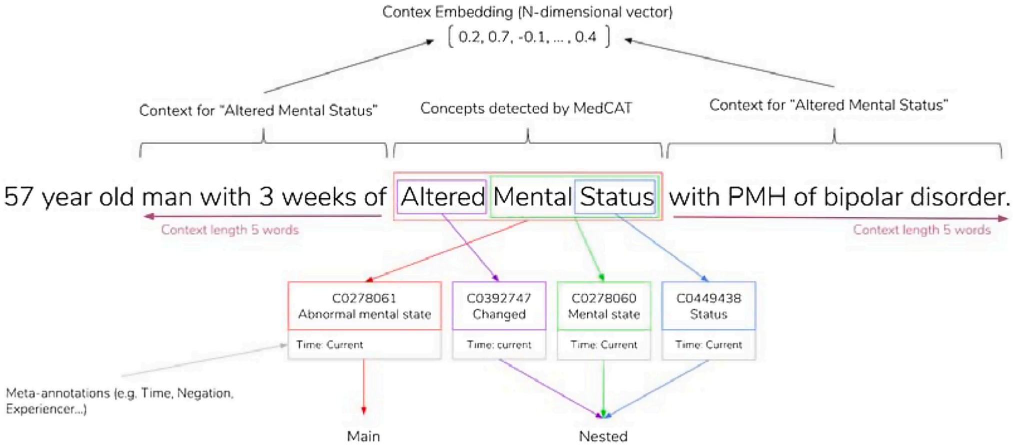

Contex Embedding (N-dimensional vector)
[0.2, 0.7, -0.1, ... , 0.4]
Context for "Altered Mental Status"
Concepts detected by MedCAT
Context for "Altered Mental Status"
57 year old man with 3 weeks of Altered Mental Status with PMH of bipolar disorder.
Context length 5 words
Context length 5 words
C0278061
Abnormal mental state
Time: Current
C0392747
Changed
Time: current
C0278060
Mental state
Time: Current
C0449438
Status
Time: Current
Meta-annotations (e.g. Time, Negation, Experiencer...)
Main
Nested

Fig. 1. A fictitious example of biomedical NER+L with nested entities and further 'meta-annotations'; a further classification or 'context' applied to an already extracted concept, e.g. 'time current' indicates extracted concepts are mentioned in a temporally present context. This context may also be referred to as an attribute of a recognized entity. Each one of the detected boxes (nested) has multiple candidates in the Unified Medical Language System (UMLS). The goal is to detect the entity and annotate it with the most appropriate concept ID, e.g. for the span Status, we have at least three candidates in UMLS, namely C0449438, C1444752, C1546481.

MedCAT and compares the effectiveness of these technologies with existing biomedical NER+L tools. We further present real clinical usage of our work in the analysis of multiple EHRs across various NHS hospital sites including running the system over ~20 years of collected data pre-dating even the usage of modern EHRs at one site. MedCAT has been deployed and contributed to clinical research findings in multiple NHS trusts throughout England [5,6].

1.1. Problem definition

Recently NER models based on deep learning (DL), notably transformers [7] and long-short term memory (LSTM) networks [8] have achieved considerable improvements in accuracy [9]. However, both approaches require explicit supervised training. In the case of biomedical concept extraction, there is little publicly available labelled data due to the personal and sensitive nature of the text. Building such a corpus can be onerous and expensive due to the need for direct EHR access and domain expert annotators. In addition, medical vocabularies can contain millions of different named entities with overlaps (see Fig. 1). Extracted entities will also often require further classification to ensure they are contextually relevant; for example extracted concepts may need to be ignored if they occurred in the past or are negated. We denote this further classification as meta-annotation or a 'contextualization' of a recognized entity. Overall, using data-intensive methods such as DL can be extremely challenging in real clinical settings.

This work is positioned to improve on current tools such as the Open Biomedical Annotator (OBA) service [10] that have been used in tools such as DeepPatient [11] and ConvAE [12] to structure and infer clinically meaningful outputs from EHRs. MedCAT allows for continual improvement of annotated concepts through a novel self-supervised machine learning algorithm, customization of concept vocabularies, and downstream contextualization of extracted concepts. All of which are either partially or not addressed by current tools.

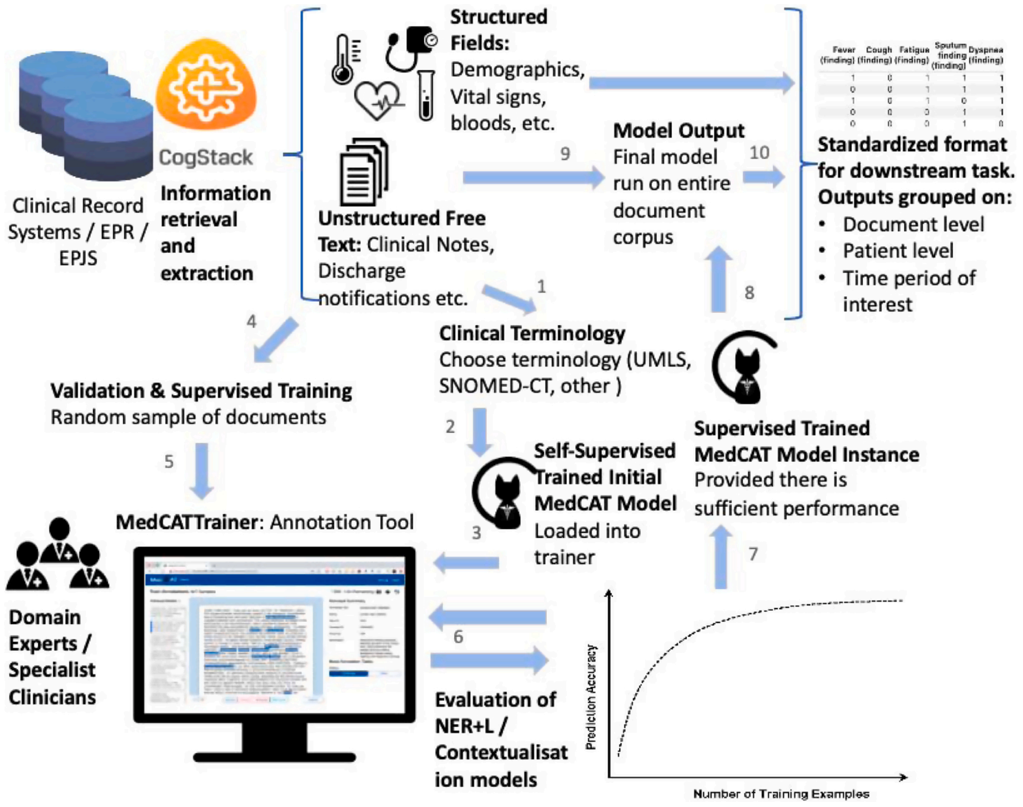
1.2. NER+L in a biomedical context

Due to the limited availability of training data in biomedical NER+L, existing tools often employ a dictionary-based approach. This involves the usage of a vocabulary of all possible terms of interest and the associated linked concept as specified in the clinical database, e.g. UMLS or SNOMED-CT. This approach allows the detection of concepts without providing manual annotations. However, it poses several challenges that occur frequently in EHR text. These include: spelling mistakes, form variability (e.g. kidney failure vs failure of kidneys), recognition and disambiguation (e.g. does 'hr' refer to the concept for 'hour' or 'heart rate' or neither).

1.3. Existing biomedical NER+L tools

We compare prior NER+L tools for biomedical documents that are capable of handling extremely large concept databases (completely and not a small subset). MetaMap [13] was developed to map biomedical text to the UMLS Metathesaurus. MetaMap cannot handle spelling mistakes and has limited capabilities to handle ambiguous concepts. It offers an opaque additional 'Word-Sense-Disambiguation' system that attempts to disambiguate candidate concepts that consequently slows extraction. Bio-YODIE [14] improves upon the speed of extraction compared to MetaMap and includes improved disambiguation capabilities, but requires an annotated corpus or supervised training. SemEHR [15] builds upon Bio-YODIE to somewhat address these shortcomings by applying manual rules to the output of Bio-YODIE to improve the results. Manual rules can be labour-intensive, brittle and time-consuming, but they can produce good results [16]. cTAKES [17], builds on existing open-source technologies-the Unstructured Information Management Architecture [18] framework and OpenNLP [19] the natural language processing toolkit. The core cTAKES library does not handle any of the previously mentioned challenges without additional plugins. ScispaCy [20] is a practical biomedical/scientific text processing tool, which heavily leverages the spaCy[2] library. In contrast to other tools mentioned, ScispaCy is primarily a supervised model for NER with limited linking capabilities. CLAMP [21] is a comprehensive clinical NLP software that enables recognition and automatic encoding of clinical information in narrative patient reports. Similar to ScispaCy it is a supervised approach and not directly comparable to other tools mentioned here. MetaMap, BioYODIE, SemEHR, cTakes and ScispaCy only support extraction of UMLS concepts. BioPortal [22] offers a web hosted annotation API for 880 distinct ontologies. This is important for use cases that are not well supported by only the UMLS concept vocabulary [23] or are better suited to alternative terminologies [24].

[2] https://github.com/explosion/spaCy.


Clinical Record Systems / EPR / EPJS
CogStack
Information retrieval and extraction
Structured Fields: Demographics, Vital signs, bloods, etc.
Unstructured Free Text: Clinical Notes, Discharge notifications etc.
Model Output Final model run on entire document corpus
9
10
Standardized format for downstream task. Outputs grouped on:
• Document level
• Patient level
• Time period of interest
1
8
4
Clinical Terminology Choose terminology (UMLS, SNOMED-CT, other)
Validation & Supervised Training Random sample of documents
2
Self-Supervised Trained Initial MedCAT Model Loaded into trainer
Supervised Trained MedCAT Model Instance Provided there is sufficient performance
5
3
MedCATTrainer: Annotation Tool
Domain Experts / Specialist Clinicians
7
6
Evaluation of NER+L / Contextualisat ion models
Prediction Accuracy
Number of Training Examples

Fever (finding)	Cough (finding)	Fatigue (finding)	Sputum finding (finding)	Dyspnea (finding)
1	0	1	1	1
0	0	1	1	1
1	0	1	0	1
0	0	0	1	1
0	0	0	1	0

Fig. 2. An example MedCAT workflow using the MedCAT core library and MedCATtrainer technologies to support clinical research.

However, transmitting sensitive hospital data to an externally hosted annotation web API may be prohibited under data protection legislation [25]. The BioPortal annotator is a 'fixed' algorithm so does not allow customization or improvements through machine learning or support of non-English language corpora [26].

CLAMP, and in a limited capacity cTakes and SemEHR, support further contextualization of extracted concepts. MetaMap, BioYODIE and scispaCy treat this as a downstream task although it is often required before extracted concepts can be used in clinical research. MedCAT addresses these shortcomings of prior tools allowing for flexibly clinician driven definition of concept contextualization, supporting modern information extraction requirements for biomedical text.

2. Methods

MedCAT presents a set of decoupled technologies for developing IE pipelines for varied health informatics use cases. Fig. 2 shows a typical MedCAT workflow within a wider typical CogStack deployment. CogStack queries selectively extract relevant documents from the EHR including the structured and unstructured (freetext) notes. With MedCAT we firstly agree with clinical partners the relevant terms within a clinical terminology(1) and train MedCAT self-supervised(2). We load the model into the MedCATtrainer annotation tool(3) alongside a random sample of the extracted EHR documents(4). Clinical domain experts validate and improve the model using supervised online learning (5). Metrics demonstrate the quality of a fine-tuned MedCAT model(6) and once desired performance is reached the fine-tuned model is exported(7) and run upon the wider free-text EHR dataset(8,9), facilitating downstream clinical research through the newly structured data (10).

This section presents the MedCAT platform technologies, its method for learning to extract and contextualize biomedical concepts through self-supervised and supervised learning. Integrations with the broader CogStack ecosystem are presented alongside source code.[3] Finally, we present our experimental methodology for assessing MedCAT in real clinical scenarios.

2.1. The MedCAT Core Library

We now outline the technical details of the NER+L algorithm, the self-supervised and supervised training procedures and methods for flexibly contextualizing linked entities.

2.1.1. Vocabulary and concept database

MedCAT NER+L relies on two core components:

- **Vocabulary (VCB)**: the list of all possible words that can appear in the documents to be annotated. It is primarily used for the spell checking features of the algorithm. We have compiled our own VCB by scraping Wikipedia and enriching it with words from UMLS. Only the Wikipedia VCB is made public, but the full VCB can be built with scripts provided in the MedCAT repository (https://github.com/CogStack/MedCAT). The scripts require access to the UMLS Metathesaurus (https://www.nlm.nih.gov/research/umls).
- **Concept database (CDB)**: a table representing a biomedical concept dictionary (e.g. UMLS, SNOMED-CT). Each new concept added to the CDB is represented by an ID and Name. A concept ID can be referred

[3] https://cogstack.atlassian.net/wiki/spaces/COGDOC/pages/733380653/Natural+Language+Processing.

to through multiple names with identical conceptual meanings such as heart failure, myocardial failure, weak heart and cardiac failure.

2.1.2. The NER+L algorithm

With a prepared CDB and VCB, we perform a first pass NER+L pipeline then run a trainable disambiguation algorithm. The initial NER+L pipeline starts with cleaning and spell-checking the input text. We employ a fast and lightweight spell checker (http://www.norvig.com/spell-correct.html) that uses word frequency and edit distance between misspelled and correct words to fix mistakes. We use the following rules:

- A word is spelled against the VCB, but corrected only against the CDB.
- The spelling is never corrected in the case of abbreviations.
- An increase in the word length corresponds to an increase in character correction allowance.

Next, the document is tokenized and lemmatized to ensure a broader coverage of all the different forms of a concept name. We used SciSpaCy [20], a tool tuned for these tasks in the biomedical domain. Finally, to detect entity candidates we use a dictionary-based approach with a moving expanding window:

1. Given a document d_1
2. Set window_length = 1 and word_position = 0
3. There are three possible cases:
   (a) The text in the current window is a concept in our CDB (the concept dictionary), mark it and go to 4. Note that MedCAT can ignore token order, but only for up-to two tokens (stopwords are not counted in the two token limit).
   (b) The text is a substring of a longer concept name, if so go to 4.
   (c) Otherwise reset window_length to 1, increase word_position by 1 and repeat step 3
4. Expand the window size by 1 and repeat 3.

Steps 3 and 4 help us solve the problem of overlapping entities shown in Fig. 1.

2.2. Self-supervised training procedure

For concept recognition and disambiguation, we use context similarity. Initially, we find and annotate mentions of concepts that are unambiguous, (e.g. step 3.a. in the previous expanding window algorithm) then we learn the context of marked text spans. For new documents, when a concept candidate is detected and is ambiguous its context is compared to the currently learned one, if the similarity is above a threshold the candidate is annotated and linked. The similarity between the context embeddings also serves as a confidence score of the annotation and can be later used for filtering and further analysis. The self-supervised training procedure is defined as follows:

1. Given a corpus of biomedical documents and a CDB.
2. For each concept in the CDB ignore all names that are not unique (ambiguous) or that are known abbreviations.
3. Iterate over the documents and annotate all of the concepts using the approach described earlier. The filtering applied in the previous steps guarantee the entity can be annotated.
4. For each annotated entity calculate the context embedding V_{cntx}.
5. Update the concept embedding V_{concept} with the context embedding V_{cntx}.

The self-supervised training relies upon one of the names assigned to each concept to be unique in the CDB. The unique name is a reference point for training to learn concept context, so when an ambiguous name appears (a name that is used for more than one concept in the CDB) it can be disambiguated. For example, the UMLS concept id:*C0024117* has the unique name Chronic Obstructive Airway Disease. This name is unique in UMLS. If we find a text span with this name we can use the surrounding text of this span for training, because it uniquely links to C0024117. ~95% of the concepts in UMLS have at least one unique name.

The context of a concept is represented by vector embeddings. Given a document d_1 where C_x is a detected concept candidate (Eq. (1)) we calculate the context embedding. This is a vector representation of the context for that concept candidate (Eq. (2)). That includes a pre-set (s) number of words to the left and right of the concept candidate words. Importantly, the concept candidate words are also included in context embedding calculation as the model is assisted by knowing what words the surrounding context words relate to.

$$d_1 = w_1 \quad w_2 \quad \cdots \quad \overbrace{w_k \quad w_{k+1}}^{Cx} \quad \cdots \quad w_n \tag{1}$$

where

d_1 – an example of a document
$w_{1..n}$ – words in the document, or to be more specific tokens
C_x – the detected concept candidate that matches the words w_k and w_{k+1}

$$V_{\text{cntx}} = \frac{1}{2s}\left[\sum_{i=1}^{s} V_{w_{k-i}} + \sum_{i=1}^{s} V_{w_{k+1+i}}\right] \tag{2}$$

where

V_{cntx} – calculated context embedding
V_{w_k} – word embedding
s – words from left and right that are included in the context of a detected concept candidate. Typically, s is set to 9 for *long* context and 2 for *short* context.

To calculate context embeddings we use the word embedding method Word2Vec [27]. Contextualized embedding approaches such as BERT [28] were also tested alongside fastText [29] and GloVe [30]. Results presented in Section 3.1 show the BERT embeddings (the MedCAT U/MI/B configuration) perform worse on average compared to the simpler Word2Vec embeddings. FastText and GloVe perform similarly to Word2Vec, therefore our default implementation uses Word2Vec for ease of implementation. We trained 300 dimensional Word2Vec embeddings using the entire MIMIC-III [31] dataset of 53,423 admissions.

Once a correct annotation is found (a word uniquely links to a CDB name), a context embedding V_{cntx} is calculated, and the corresponding V_{concept} is updated using the following formula:

$$\text{sim} = \max\left(0, \frac{V_{\text{concept}}}{\|V_{\text{concept}}\|} \cdot \frac{V_{\text{cntx}}}{\|V_{\text{cntx}}\|}\right) \tag{3}$$

$$\text{lr} = \frac{1}{C_{\text{concept}}} \tag{4}$$

$$V_{\text{concept}} = V_{\text{concept}} + \text{lr}\cdot(1 - \text{sim})\cdot V_{\text{cntx}} \tag{5}$$

where

C_{concept} – number of times this concept appeared during training
sim – similarity between V_{concept} and V_{cntx}
lr – learning rate.

The update rule is based on the Word2Vec model and aims to make the concept embedding V_{concept} similar to the context in which the concept was presently found V_{cntx}. The scaling which is achieved via the cosine similarity is used to favour new contexts in which a concept appears over contexts that frequently appeared in the past.

To prevent the context embedding for each concept being dominated by most frequent words, we used negative sampling as defined in [27]. Whenever we update the V_{concept} with V_{cntx} we also generate a negative context by randomly choosing K words from the vocabulary consisting of all words in our dataset. Here K is equal to $2s$, i.e. twice the window size for the context (s is the context size from one side of the detected concept, meaning in the positive cycle we will have s words from the left and s words from the right). The probability of choosing each word and the update function for vector embeddings is defined as:

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_j^n f(w_j)^{3/4}} \qquad (6)$$

$$f(w_i) = \frac{C_{w_i}}{\sum_j^n C_{w_j}} \qquad (7)$$

$$V_{\text{ncntx}} = \frac{1}{K}\sum_i^K V_{w_i} \qquad (8)$$

$$\text{sim} = \max\left(0, \frac{V_{\text{concept}}}{\left\|V_{\text{concept}}\right\|} \cdot \frac{V_{\text{ncntx}}}{\left\|V_{\text{ncntx}}\right\|}\right) \qquad (9)$$

$$V_{\text{concept}} = V_{\text{concept}} - \text{lr}\cdot\text{sim}\cdot V_{\text{ncntx}} \qquad (10)$$

where

n – size of the vocabulary
$P(w_i)$ – probability of choosing the word w_i
K – number of randomly chosen words for the negative context
V_{ncntx} – negative context

2.2.1. Supervised training procedure

The supervised training process is similar to the self-supervised process but given the correct concept for the extracted term we update the V_{concept} using the calculated V_{ctx} as defined in Eqs. (3)–(10). This no longer relies upon the self-supervised constraint that at least one name in the set of possible names for a concept is unique as the correct term is provided by human annotators.

2.2.2. Contextualization of identified and linked concepts: meta-annotations

Once a span of text is recognized and linked to a concept, further contextualization or meta-annotation is often required. For example, a simple task of identifying all patients with a fever can entail classifying the located fever text spans that are current mentions (e.g. the patient reports a fever vs the patient reported a fever but, etc.), are positive mentions (e.g. patient has a high fever vs patient has no sign of fever), are actual mentions (e.g. patient is feverish vs monitoring needed if fever reappears), or are experienced by the patient (e.g. pts family all had high fevers). We treat each of these contextualization tasks as distinct binary or multiclass classification tasks Meta-annotations are equivalent to 'attributes' in cTakes parlance.

The MedCAT library provides a 'MetaCAT' component that wraps a Bidirectional-Long-Short-Term-Memory (Bi-LSTM) model trainable directly from MedCATtrainer project exports. Bi-LSTM models have consistently demonstrated strong performance in biomedical text classification task [32–34] and our own recent work [35] demonstrated a Bi-LSTM based model outperforms all other assessed approaches, including Transformer models. MetaCAT models replace the specific concept of interest for example 'diabetes mellitus' with a generic parent term of the concept '[concept]'. The forward/backward pass of the model then learns a concept agnostic context representation of the concept allowing MetaCAT models to be used across concepts as observed in our results (Section 3.3.3). The MetaCAT API follows standard neural network training methods but are abstracted away from end users whilst still maintaining enough visibility for users to understand when MetaCAT models have been trained effectively. Each training epoch displays training and test set loss and metrics such as precision, recall and F1. An open-source tutorial showcasing the MetaCAT features are available as part of the series of wider MedCAT tutorials.[4] Once trained, MetaCAT models can be exported and reused for further usage outside of initial classification tasks similarly to the MedCAT NER+L models.

2.3. MedCATTrainer: annotation tool

MedCATtrainer allows domain experts to inspect, modify and improve a configured MedCAT NER+L model. The tool either actively trains the underlying model after each reviewed document (facilitating live model improvements as feedback is provided by human users) or simply collects and validates concepts extracted by a static MedCAT model. The active learning is done on a concept level and MedCATtrainer will automatically mark some concepts as correct/incorrect and ask for user input for others where it is not confident enough. Version 0.1 [4] presented a proof-of-concept annotation tool that has been rewritten and tightly integrated with the MedCAT library, whilst providing a wealth of new features supporting clinical informatics workflows. We also provide extensive documentation[5] and pre-built containers[6] updated with each new release facilitating easy setup by informatics teams.

2.4. Datasets and experimental setup

2.4.1. Named entity recognition and linking open datasets

MedCAT concept recognition and linking was validated on the following publicly datasets:

1. MedMentions [36] – consists of 4392 titles and abstracts randomly selected from papers released on PubMed in 2016 in the biomedical field, published in the English language, and with both a Title and Abstract. The text was manually annotated for UMLS concepts resulting in 352,496 mentions. We calculate that ~40% of concepts in MedMentions require disambiguation, suggesting a detected span of text can be linked to multiple UMLS concepts if only the span of text is considered.
2. ShARe/CLEF 2014 Task 2 [37] – we used the development set containing 300 documents of 4 types – discharge summaries, radiology, electrocardiograms, and echocardiograms. We have used the UMLS annotations and ignored the attribute annotations.
3. MIMIC-III [31] – consists of ~58,000 de-identified EHRs from critical care patients collected between 2001 and 2012. MIMIC-III includes demographic, vital sign, and laboratory test data alongside unstructured free-text notes.

We attempted to use the SemEval 2019 shared task for the evaluation of the NER+L task,[7] but dataset access is currently under review for all requests to i2b2.

2.4.2. Clinical use case datasets

Our further experiments used real world EHR data from the following UK NHS hospital Trusts:

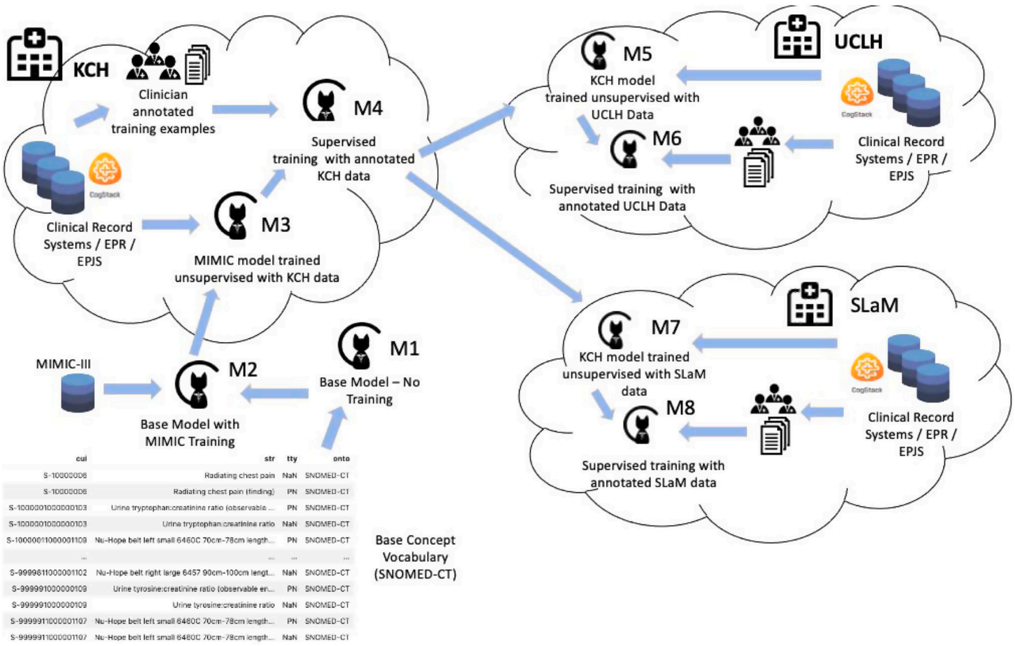
- King's College Hospital Foundation Trust (KCH) Dataset:

[4] https://colab.research.google.com/drive/1zzV3XzFJ9ihhCJ680DaQV2QZ5XnHa06X.

[5] https://github.com/CogStack/MedCATtrainer/blob/master/README.md

[6] https://hub.docker.com/r/cogstacksystems/medcat-trainer

[7] https://competitions.codalab.org/competitions/19350.


KCH
Clinician annotated training examples
M4
Supervised training with annotated KCH data
CogStack
Clinical Record Systems / EPR / EPJS
M3
MIMIC model trained unsupervised with KCH data
MIMIC-III
M2
Base Model with MIMIC Training
M1
Base Model – No Training
Base Concept Vocabulary (SNOMED-CT)
UCLH
M5
KCH model trained unsupervised with UCLH Data
M6
Supervised training with annotated UCLH Data
CogStack
Clinical Record Systems / EPR / EPJS
SLaM
M7
KCH model trained unsupervised with SLaM data
M8
Supervised training with annotated SLaM data
CogStack
Clinical Record Systems / EPR / EPJS

cui	str	tty	onto
S-10000006	Radiating chest pain	NaN	SNOMED-CT
S-10000006	Radiating chest pain (finding)	PN	SNOMED-CT
S-1000001000000103	Urine tryptophan:creatinine ratio (observable ...	PN	SNOMED-CT
S-1000001000000103	Urine tryptophan:creatinine ratio	NaN	SNOMED-CT
S-10000011000001109	Nu-Hope belt left small 6460C 70cm-78cm length...	PN	SNOMED-CT
...	...	...	...
S-9999611000001102	Nu-Hope belt right large 6457 90cm-100cm lengt...	NaN	SNOMED-CT
S-999991000000109	Urine tyrosine:creatinine ratio (observable en...	PN	SNOMED-CT
S-999991000000109	Urine tyrosine:creatinine ratio	NaN	SNOMED-CT
S-9999911000001107	Nu-Hope belt left small 6480C 70cm-78cm length...	PN	SNOMED-CT
S-9999911000001107	Nu-Hope belt left small 6480C 70cm-78cm length...	NaN	SNOMED-CT

Fig. 3. Model provenance for NER+L clinical use case results between datasets and sites. M1-8, showing the MedCAT model instances, the data and method of training and base model used across all sites.

–300 free text inpatient notes for Covid-19 positive patients, 121 Epilepsy clinic letters 2018–2019, 100 Cardiac Clinic letters, 200 echocardiographic reports, 100 CT pulmonary angiograms, 700 10k character chunks of clinical notes of patients with Diabetes Mellitus/ Gastroenteritis/ Inflammatory bowel disease/Crohn's disease/ulcerative colitis for supervised training.

–∼17 million documents with ∼8.8 billion tokens (entire KCH electronic health record from 1999 to 2020 consisting documents from 'multi-era', multi-vendor electronic health records (including iSoft iCM, EMIS Symphony and AllScripts) and multiple geographically-distributed hospital sites (Kings College Hospital, Princess Royal University Hospital and Orpington Hospital) were processed for self-supervised training.

- South London and Maudsley Foundation Trust (SLaM): 2200 free text notes for patients with a primary or secondary diagnosis of severe mental illness between 2007 and 2018 with each document reviewed for only a specific physical health comorbidity that may or may not appear in the note.
- University College London Hospitals Foundation Trust (UCLH) Covid-19 Datasets: 300 Free text clinical notes for Covid-19 positive or suspected patients from January to April 2020 from single-vendor electronic health record (Epic).

We used two large biomedical concept databases and prepared them as described in our source-code repository,[8] the databases are:

- UMLS 2018AB: 3.82 million concepts and 14 million unique concept names from 207 source vocabularies.
- SNOMED CT UK edition: >659K concepts. The UK SNOMED CT clinical extension 20200401 and UK Drug Extension 20200325 with ICD-10 and OPCS-4 mappings.

2.4.3. Named entity recognition and linking experimental setup

We use MedMentions [36], ShARe/CLEF [37] and MIMIC-III [31] datasets in our experiments. We denote the 'MedMentions' dataset (i.e. all concepts) and 'MedMentions Disorders Only' (i.e. only concepts grouped under the Disorder group as shown in [38]). We train MedCAT self-supervised on MIMIC-III configured with the UMLS database. We denote the version using Word2Vec embeddings as 'MedCAT' and the one using Bio_ClinicalBERT [39] embeddings as 'MedCAT BERT'.

An annotation by MedCAT is considered correct only if the exact text value was found and the annotation was linked to the correct concept in the CDB. We contrast our performance with the performance of tools presented in Section 1.3. Appendix C provides self-supervised training configuration details.

2.4.4. Clinical use case NER+L experimental setup

For our clinical use cases we extracted SNOMED-CT terms, the official terminology across primary and secondary care for the UK National Health service, as this was preferred by our clinical teams over UMLS.

Fig. 3 shows our process of model training and distribution to partner hospital Trusts. Initially, we built our untrained MedCAT model using the SNOMED-CT concept vocabulary (M1), we then trained it self-supervised on the MIMIC-III dataset (M2). Next, the entire KCH EPR (17M documents with 8.8B tokens) is used for self-supervised training (M3). We collect annotations with clinician experts at KCH and train supervised (M4). We share this model with each partner hospital site where further self-supervised training (M5, M7) and specific supervised training with their respective annotation datasets (M6, M8).

Site-specific models (M3, M5, M7) are loaded into deployed instances of MedCATtrainer and configured with annotation projects to collect SNOMED-CT annotations for a range of site specific disorders, findings, symptoms, procedures and medications that our clinical teams are interested in for further research (i.e. already published work on Covid-19[5,6]). These included chronic (i.e. diabetes mellitus, ischaemic heart disease, heart failure) and acute (cerebrovascular accident, transient ischemic attack) disorders. For comparison between sites we find 14 common extracted concept groups (Appendix A) and calculate

[8] https://github.com/CogStack/MedCAT#building-concept-databases-from-scratch.

Table 1
Meta annotation tasks defined per site, KCH = King's College Hospital NHS Foundation Trust, UCLH = University College London Hospitals NHS Foundation Trust, SLaM = South London and Maudsley NHS Foundation Trust.

Site	Task	Values
KCH	Presence	Affirmed/Negated/Hypothetical
	Experiencer	Patient/Family/Other
	Temporality	Past/Present/Future
UCLH	Negation	Yes/No
	Experiencer	Yes/No
	Problem Temporality	Past Medical Issue/Current Problem
	Certainty	Confirmed/Suspected
	Irrelevant	Yes/No
SLaM	Status	Patient/Other/NA
	Diagnosis	Yes/No

F1 scores for each concept group and reporting average, standard deviation (SD), and interquartile-range (IQR).

We shared fine-tuned MedCAT models between KCH and 2 NHS partner Trusts UCLH and SLaM. This was a collaborative effort with each hospital team only having access to their respective hospital EHR/ CogStack instance. Each site collected annotated data using MedCAT-trainer, tested the original base model, a self-supervised only trained model and a final supervised trained model with the MedCATtrainer collected annotations.

2.4.5. *Clinical use case contextualization model experimental setup*

From ongoing and published work [5,6] we configured and collected meta-annotation training examples and trained a variety of contextualization models per site as defined in Table 1.

Our experiments test the effectiveness of our meta annotation modelling approach to flexibly learn contextual cues by assessing cross-disorder and cross-site transferability (Section 3.3.3. To assess cross-disorder transferability of each of the 11 disorder groups (as specified in Appendix A) we use the SLaM collected 'Diagnosis' dataset that consists of ~100 annotations for each disorder group. We stratify our train/test sets by disorder, placing all examples for one disorder group in the test set and use the remaining disorder examples as a train set. We run this procedure 11 times so that each disorder group is tested once. We average all scores of each fold and report results.

To demonstrate cross-site transferability we derive an equivalent meta-annotation dataset from the 'Presence' (KCH) and 'Status' (SLaM) datasets as they are semantically equivalent despite having different possible annotation values. We merge 'Presence' annotations from Affirmed/Hypothetical/False to Affirmed/Other to match classes available in SLaM. We then train and test new meta annotation models between sites and datasets report average results.

3. Results

We firstly present our concept recognition and linking results, comparing performance across previously described tools in Section 1.3 using the UMLS concept database and openly available datasets presented in Section 2.4 We then present a qualitative analysis of learnt concept embeddings demonstrating the captured semantics of MedCAT concepts. Finally, we show real world clinical usage of the deployed platform to extract, link and contextualize SNOMED-CT concepts across multiple NHS hospital trusts in the UK.

3.1. *Entity extraction and linking*

Table 2 presents our results for self-supervised training of MedCAT and NER+L performance compared with prior tools using openly available datasets. Metrics for all the tools were calculated consistently. Bold indicates best performance. For each manual annotation we check whether it was detected and linked to the correct Unified Medical Language System (UMLS) concept. The metrics are precision (P), recall (R) and the harmonic mean of precision and recall (F1). MedCAT models were configured with UMLS concepts and trained (self-supervised) on MIMIC-III: the base version (MedCAT) uses Word2Vec embeddings (trained on MIMIC-III), while (MedCAT BERT) uses static word embeddings from Bio_ClinicalBERT [39]. For the BERT version of MedCAT we do not use the full BERT model to calculate context representations, but only the pre-trained static word embeddings.

Our results show MedCAT improves performance compared to all prior tools across all tested metrics (excluding precision when compared to ScispaCy/CLAMP – which are supervised models). We observe that the best performance across all tools is achieved on the ShARe/CLEF dataset. However, MedCAT still improves F1 performance by ~9 percentage points over the next best system. We note the simpler Word2Vec embedding (base MedCAT) on average performs better than the more expressive Bio_ClinicalBERT (BERT) embeddings. We provide a further breakdown of the range of performances by MedCAT across MedMentions and ShARe/CLEF split by UMLS semantic type in Table 3.

Table 3
MedCAT performance for different UMLS semantic types on MedMentions and ShARe/CLEF.

Semantic type	Dataset	MedMentions			ShARe/CLEF		
		P	R	F1	P	R	F1
T047	Disease or Syndrome	0.59	0.59	0.59	0.87	0.75	0.80
T121	Therapeutic or Preventive Procedure	0.52	0.52	0.52		NO DATA	
T061	Pharmacologic Substance	0.49	0.38	0.43		NO DATA	
T184	Sign or Symptom	0.58	0.70	0.64	0.86	0.75	0.80
T048	Mental or Behavioural Dysfunction	0.63	0.55	0.58	0.71	0.63	0.66

Table 2
Comparison of NER+L tools for the extraction of UMLS concepts. * The results for ScispaCy/CLAMP are not directly comparable to other tools as they are supervised models.

Model\dataset	MedMentions			MedMentions (disorders only)			ShARe/CLEF		
	P	R	F1	P	R	F1	P	R	F1
SemEHR	0.252	0.165	0.200	0.295	0.499	0.371	0.680	0.623	0.650
Bio-YODIE	0.316	0.143	0.197	0.445	0.366	0.402	0.700	0.607	0.650
cTAKES	0.284	0.129	0.178	0.313	0.375	0.342	0.567	0.640	0.601
MetaMap	0.305	0.465	0.368	0.358	0.460	0.403	0.755	0.540	0.630
ScispaCy*	**0.451**	0.408	0.429	0.487	0.443	0.464	0.711	0.463	0.561
CLAMP*	0.324	0.067	0.110	**0.533**	0.236	0.327	0.772	0.447	0.566
MedCAT BERT	0.386	0.475	0.426	0.459	0.513	0.485	0.788	0.678	0.729
MedCAT	0.406	**0.500**	**0.448**	0.470	**0.523**	**0.495**	**0.796**	**0.688**	**0.738**
+δ(MedCAT-Best)	− 0.045	0.035	0.019	− 0.063	0.024	0.031	0.041	0.048	0.088

Table 4
Qualitative analysis of learnt concept embeddings. UMLS concepts that have highest cosine similarity between learnt vector embeddings of concepts in **bold**. The first row defines the chosen concept and the target concept type. We have randomly chosen the most frequent concepts and presented the 8 most similar concepts for each target concept type. For example, Neoplastic Process (C0006826) and the following rows show the top 8 most similar Procedure concepts.

Disease → Medication	Disease → Procedure	Symptom → Medication
Hypertensive disease	**Neoplastic process**	**Fever**
Metoprolol 50 MG	Chemotherapy	Levofloxacin
Metoprolol 25 MG	Radiosurgery	Vancomycin
Valsartan 320 MG	FOLFOX Regimen	Vancomycin 750 MG
Nadolol 20 MG	Chemotherapy Regimen	Azithromycin
Atenolol 100 MG	Preoperative Therapy	Levofloxacin 750 MG
Enalapril 10 MG	Anticancer Therapy	Dexamethasone
Oral form diltiazem	Parotidectomy	Lorazepam
nimodipine 30 MG	Resection of Ileum	Acetaminophen

3.2. Qualitative analysis

For concept disambiguation the MedCAT core library learns vector embeddings from the contexts in which a concept appears. This is similar to prior work [40], although we also present a novel self-supervised training algorithm, annotation system and wider workflow. Using our learnt concept embeddings we perform a qualitative analysis by inspecting concept similarities, with the expectation that similar concepts have similar embeddings. Table 4 shows the learnt context embeddings capture medical knowledge including relations between diseases, medications and symptoms. We train MedCAT self-supervised over MIMIC-III [31] using the entirety of UMLS, 3.82 Million concepts from 207 separate vocabularies. Training configuration details are provided in C.

3.3. Clinical use cases across multiple hospitals

The MedCAT platform was used in a number of clinical use cases providing evidence for its applicability to answer relevant, data intensive research questions. For example, we extracted relevant comorbid health conditions in individuals with severe mental illness and patients hospitalized after Covid-19 infection [5,6,41]. These use cases analysed data sources from 2 acute secondary/tertiary care services at King's College Hospital (KCH), University College London Hospitals (UCLH) and mental health care services South London and Maudsley (SLaM) NHS Foundation Trusts in London, UK.

The following results focus on providing an aggregate view of MedCAT performance over real NER+L clinical use-cases, meta-annotation or context classification tasks and model transferability across clinical domains (physical health vs mental health), EHR systems and concepts.

3.3.1. Entity extraction and linking

Table 5 shows our results for NER+L across hospital sites, model and training configurations as described in Section 2.4.2 Our KCH annotations were collected across a range of clinicians, clinical research questions and therefore MedCATtrainer projects. This unfortunately led to a lack of resourcing to enable double annotations and calculation of inter-annotator-agreement (IIA) scores. SLaM annotations were collected by clinician/non-clinician pairs with average inter-annotator agreement (IIA) at 0.88, disagreements were discarded before results were calculated to ensure a gold-standard. UCLH IIA was at 0.85 between two medical students with annotation disagreements arbitrated by an experienced clinician providing the final gold-standard dataset. For our KCH results we use all annotations collected across various MedCATtrainer projects within our 14 concept groups as described in Section 2.4.4 Both KCH and UCLH annotations contained occurrences of all 14 concept groups, SLaM annotated notes did not contain any occurrences of Dyspnea (SCTID:267036007), Pulmonary embolism (SCTID:59282003) and Chest pain (SCTID:29857009).

3.3.2. Entity extraction and linking model transferability

Table 5 demonstrates the improved NER+L performance that arises from using domain specific data first self-supervised in MIMIC-III, then KCH. We observe further improvements with clinician expertise with supervised training using the KCH data. With model sharing to UCLH we observe a 0.044 average drop in F1 performance compared to KCH. Further self-supervised training directly on UCLH data offers minimal average performance gains but does reduce the F1 SD and IQR suggesting there is less variability in performance across concepts. Supervised training on a small (499) annotations from UCLH delivers comparable performance to our KCH trained model. For our experiments at SLaM we see average F1 performance drop initially by 0.062 using the KCH model directly on SLaM data. SLaM is a large mental health service provider where EHRs are markedly different to acute care hospitals KCH and UCLH. Interestingly, successive self-supervised (M7) and supervised training (M8) show benefits across all measures with final performance largely similar to final KCH performance.

Importantly, this suggests performance is transferred to the different hospital sites and initially only drops by ~0.04. With self-supervised training and further supervised training we are able to reach KCH performance with ~7× fewer manually collected examples at UCLH or ~2× fewer examples at SLaM.

3.3.3. Contextualization model performance

Contextualization of extracted and linked concepts is, by design, bespoke per project. Due to this, reporting and comparing results across studies/sites is difficult as the definitions of tasks and concepts collected are different and therefore output trained models are bespoke. Table 6a shows aggregate performance at each site, and Table 6b and c shows further experiments for cross-site and cross-concept model transferability.

Table 5
NER+L results across hospitals. MedCAT NER+L performance for common disorder concepts defined in Appendix A by clinical teams. Annotations for supervised learning are used as test sets for models M1, M2, M3, M5, M7. Average performance on a 10 fold cross-validation with a held out test set is reported for models M4, M6, M8. KCH: Kings College Hospital; UCLH: University College Hospital; SLaM: South London and The Maudsley NHS Foundation Trusts.

Model	Training configuration	Hospital test site	# Annotated examples	Avg. F1	F1 SD±	F1 IQR
M1	Base – No Training	KCH	3358	0.638	0.297	0.333
M2	Base + Self-Supervised MIMIC-III	KCH	3358	0.840	0.109	0.150
M3	Base + Self-Supervised KCH	KCH	3358	0.889	0.078	0.103
M4	KCH Self-Supervised + KCH Supervised	KCH	3358	0.947	0.044	0.051
M4	KCH Self-Supervised + KCH Supervised	UCLH	499	0.903	0.103	0.112
M5	KCH Self-Supervised + KCH Supervised + UCLH Self-Supervised	UCLH	499	0.905	0.079	0.034
M6	KCH Self-Supervised + KCH Supervised + UCLH Self-Supervised + UCLH Supervised	UCLH	499	0.926	0.060	0.086
M4	KCH Self-Supervised + KCH Supervised	SLaM	1425	0.885	0.095	0.088
M7	KCH Self-Supervised + KCH Supervised + SLaM Self-Supervised	SLaM	1425	0.907	0.047	0.082
M8	KCH Self-Supervised + KCH Supervised + SLaM Self-Supervised + SLaM Supervised	SLaM	1425	0.945	0.029	0.025

Table 6
Contextualization model results.

(a) Site specific contextualization model performance. Weighted/Macro average F1 Meta annotation model performance custom defined and trained per site – detailed definitions are provided in Appendix D. Task definitions are uniquely defined at each site, e.g. Experiencer at KCH considers the values patient/family/other whereas Experiencer at UCLH only considers the value patient/other. Status at SLaM considers the values affirmed/other and Certainty at UCLH considers the values confirmed/suspected. We include all concepts of interest as defined under clinician guidance at each site, therefore site-to-site comparison in performance cannot be made.

Site	Task	# Annotated examples	Macro F1	Weighted F1
KCH	Presence	37,310	0.846	0.929
	Temporality	18,670	0.803	0.943
	Experiencer	18,670	0.867	0.959
SLaM	Patient diagnosis	1152	0.904	0.913
	Status	1152	0.775	0.812
UCLH	negation	4400	0.836	0.970
	Experiencer	4400	0.940	0.996
	Problem temporality	4350	0.848	0.970
	Certainty	4160	0.836	0.970
	Irrelevant	4390	0.835	0.969

(b) Cross site transferability performance. 11 fold concept stratified CV vs randomized CV for SLaM 'Diagnosis' contextualization task performance. The 11 concepts were selected from NER+L experiment concepts available at SLaM (Table A.1). The 'Diagnosis' task at SLaM was used as this was our most balanced dataset between all tasks and concepts collected.

Site	Task	Train/test split	Macro F1	Weighted F1
SLaM	Diagnosis	Concept stratified	0.82	0.85
SLaM	Diagnosis	Random	0.90	0.91

(c) Cross-site transferability of the MetaCAT model for Presence (at KCH)/status (at SLaM converted to values of Affirmed/Other) – as that was the only task that existed across sites. Results show 10 fold CV where applicable – e.g. row 2 is direct testing of the KCH model on SLaM data, so no training is performed on the SLaM side.

Site	Trained on	# Annotated examples	Macro F1	Weighted F1
KCH	KCH	37,310	0.89	0.93
SLaM	KCH	37,310	0.71	0.91
SLaM	SLaM	1152	0.77	0.87
SLaM	KCH + SLaM	38,462	0.85	0.96

We achieve strong weighted (0.892–0.977)/macro (0.841–0.860) F1 performance across all tasks and sites, with breakdown of each metric per site/task available in Appendix D. We report average macro and weighted F1 score demonstrating the variation in performance due to unbalanced datasets across most tasks.

For cross-concept transferability, Table 6b shows a decrease in performance when stratifying by concept. However, we still observe a relatively high 0.82–0.85 score suggesting the model is capable of learning disorder independent representations that distinguish the classification boundary for the 'Diagnosis' task, not just the disorder specific contexts.

Our cross-site transferability results, Table 6c, suggest the 'Status' context model that is trained on cross site (i.e. KCH) data then fine-tuned on site specific data (i.e. SLaM) performs better (+0.08 Macro/+0.09 Weighted F1) compared with training on only the SLaM site specific training only (i.e. comparing row 3 and 4).

4. Discussion

4.1. Named entity recognition and linking

Our evaluation of MedCAT's NER+L method using self-supervised training was bench-marked against existing tools that are able to work with large biomedical databases and are not use-case specific. Our datasets and methods are publicly available making the experiments transparent, replicable, and extendable. With the MedMentions dataset, using only self-supervised learning, our results in 3.1, demonstrate an improvement on the prior tools for both disorder detection (F1 = 0.495 vs 0.464) and general concept detection (F1 = 0.448 vs 0.429). We observe all tools perform best with the ShARe/CLEF dataset. We suggest this broadly due to the lack of ambiguity and the more clinical setting allowing alternative systems to also perform reasonably well.

We now discuss the result between our BERT and regular (Word2-Vec) configured MedCAT models. Generally BERT, a deep neural embedding model, performs well for a range of downstream tasks [28] better than older approaches such as Word2Vec, i.e. a shallow neural embedding. We believe this due to our use of pre-trained static BERT embeddings that: (1) are not specifically trained to produce similar values for words appearing in a similar context, (2) sub-word tokenization might be problematic if the tokenizer was trained on a non-medical dataset (no matter whether it was fine-tuned later on MIMIC-III, pubmed or similar).

The general concept detection task with MedMentions is difficult due to: the larger number of entities to be extracted, the rarity of certain concepts and the often highly context dependent nature of some occurrences. Recent work [42] highlights examples of ambiguous texts within the MedMentions dataset such as 'probe' with 7 possible labels ('medical device', 'indicator reagent or diagnostic aid' etc.) Further work[40] also showed a deep learning approach (BioBERT+) that achieved F1 = 0.56. When MedCAT is provided with the same supervised training data we achieve F1 = 0.71. We find our improved performance is due to the long tail of entities in MedMentions that lack sufficient training data for methods such BioBERT to perform well.

Our qualitative inspection of the learnt concept embeddings, 3.2 indicate learnt semantics of the target medical domain. This result mirrors similar findings reported in fields such as materials science [43]. Recent work has suggested an approach to quantity the effectiveness of learnt embeddings[38] in representing the source ontology. However, this relies on concept relationships to be curated before assessment requiring clinical guidance that may be subjective in the clinical domain. We leave a full quantitative assessment of the learnt embeddings to future work for this reason.

As more concepts are extracted the likelihood of concepts requiring disambiguation increases, particularly in biomedical text [44]. Estimating the number of training samples for successful disambiguation is difficult but based on our experiments we need at least 30 occurrences of a concept in the free text to perform disambiguation. We provide more details in Appendix B.

Finally, we note that there are no limitations algorithmically for MedCAT to support languages other than our tested language, English. As MedCAT uses a concept dictionary/vocab for NER+L, if there are existing resources (e.g. SNOMED-CT has already been translated into Spanish, Dutch, Swedish and Danish) they can be used directly for these languages with likely similar results. Alternatively, users could build their own custom concept dictionary (CDB) for their language of choice. Meta-annotation or contextualization models also do not have language specific features, i.e. English, and would also likely perform well as they only rely on bi-directional context from supervised examples to make predictions.

4.2. Clinical use cases

MedCAT models and annotated training data have been implemented to be easily shared and reused, facilitating a federated learning approach to model improvement and specialization with models brought to sensitive data silos. Our results in Section 3.3 demonstrate that we are able directly apply models trained at one hospital site (KCH) to multiple other sites, and clinical domains (physical vs mental health datasets) with only a small drop in average F1 (0.044 at UCLH, 0.062 at SLaM), and after small amount of additional site specific training, we observe comparable performance (− 0.021 at UCLH, − 0.002 at SLaM).

We also highlight that separate teams were able to deploy, extract and analyse real clinical data using the tools as is by following provided examples, documentation and integrations with the wider CogStack ecosystem. Academic engineering projects are often built to support a single research project, however MedCAT and the CogStack ecosystem are scalable fit-for-purpose locally-tunable solutions for teams to derive value from their data instead of being stalled by poor quality code or lack of documentation. This means the model is broadly useful with top-up tuning also available for specific scenarios, domains and hospitals.

Each hospital site and clinical team freely defined the set of contextualization tasks and associated values for each task. On aggregate our results show performance is consistently strong across all sites and tasks (Macro F1: 0.841–0.860, Weighted F1: 0.892–0.977). With many of the tasks the annotated datasets are highly unbalanced. For example, the ‘Presence’ task at KCH, disorders are often only mentioned in the EHR if they are affirmed (e.g. “…pmhx: TIA…”), and only rarely are hypothetical (e.g. “…patient had possible TIA…”) or negated terms (e.g. “…no sign of TIA…”) encountered. This explains the differences in performance when reporting macro vs weighted average F1 score. We would expect generalization performance to lie between these reported metrics.

4.3. Limitations

MedCAT is able to employ a self-supervised training method as the initial pass of the algorithm uses a given unique name to learn and improve an initial concept embedding. However, if the input vocabulary linked to the concepts inadequately specifies possible names or the given names of a concept rarely appear in the text then improvements can only occur during standard supervised learning. The main limitation of our approach is that it greatly depends on the quality of the concept database. Large biomedical concept databases (e.g. UMLS) however have a well specified vocabulary offering many synonyms, acronyms and differing forms of a given concept.

A limitation of our concept embedding approach is if different concepts appear in similar contexts disambiguation and linking to the correct concept can be difficult. For example, ‘OD’ can link to ‘overdose’ or ‘once daily’, both referring to medications with very different implications. We have rarely seen this problem during real-world corpus. Our approach can also struggle if concepts appear in many varying contexts that are rarely seen or annotated for. With each new context updating the underlying concept embedding this may decrease performance of the embedding.

Supervised learning requires training data to be consistently labelled. This is a problem in the clinical domain that consists of specialized language that can be open to interpretation. We recommend using detailed annotation guidelines that enumerate ambiguous scenarios for annotators.

4.4. Future work

MedCAT uses a vocabulary based approach to detect entity candidates. Future work could investigate the expansion of such an approach with a supervised learning model like BERT [28]. The supervised learning model would then be used for detection of entity candidates that have enough training data and to overcome the challenge of detecting new unseen forms of concept names. The vocabulary based approach would cover cases with insufficient annotated training data or concepts that have few different names (forms). The linking process for both approaches would remain the same self-supervised.

Our self-supervised training over the ~20 year KCH EHR, as described in Section 2.4, took over two weeks to complete. Future work could improve the training speed by parallelizing this process since concepts in a CDB are mostly independent of one another. Further work could address effective model sharing, allowing subsequent users/sites to benefit from prior work, where only model validation and fine-tuning is required instead of training from scratch.

Finally, ongoing work aims to extend the MedCAT library to address relation identification and extraction. For example, linking the extracted drug dosage/frequency with the associated drug concept, or identifying relations between administered procedures and following clinical events.

5. Conclusions

This paper presents MedCAT a multi-domain clinical natural language processing toolkit within a wider ecosystem of open-source technologies namely CogStack.

The biomedical community is unique in that considerable efforts have produced comprehensive concept databases such as UMLS and SNOMED-CT amongst many others. MedCAT flexibly leverages these efforts in the extraction of relevant data from a corpus of biomedical documents (e.g. EHRs). Each concept can have one or more equivalent names, such as abbreviations or synonyms. Many of these names are ambiguous between concepts. The MedCAT library is based upon a simple idea: at least one of the names for each concept is unique and given a large enough corpus that name will be used in a number of contexts. As the context is learned from the unique name, when an ambiguous name is later detected, its context is compared to the learnt context, allowing us to find the correct concept to link. By comparing the context similarity we can also calculate confidence scores for a provided linked concept.

With MedCAT we have built an effective, high performance IE algorithm demonstrating improved performance over prior solutions on open access datasets. We have commoditized the development, deployment and implementation of IE pipelines with supporting technologies MedCATtrainer/MedCATservice supporting the transfer, validation, re-use and fine-tuning of MedCAT models across sites, clinical domains and concept vocabularies. MedCAT deployments are enabled by extensive documentation, examples, APIs and supporting real world clinical use cases outlined in prior published work.

Overall, MedCAT is built to enable clinical research and potential improvements of care delivery by leveraging data in existing clinical text. Currently, MedCAT is deployed in a number of hospitals in the UK in silo or as part of the wider CogStack ecosystem, with wide-ranging use cases to inform clinical decisions with real-time alerting, patient stratification, clinical trial recruitment and clinical coding. The large volume of medical information that is captured solely in free text is now accessible using state-of-the-art healthcare specific NLP.

Data availability

Data for reproduction of experiments for the assessment for the core NER+L in comparison with are available from prior work (MedMentions, ShARe/CLEF 2014 Task 2, MIMIC-III). Due to the confidential nature of free-text data, we are unable to make patient-level data available. Interested readers should contact the authors to discuss feasibility of access of de-identified aggregate data consistent with legal permissions.

Code availability

All code for running the experiments, the toolkit and integration with wider CogStack deployments are available here:

MedCAT: https://github.com/CogStack/MedCAT
MedCAT Tutorials/Example Code: https://github.com/CogStack/MedCAT/tree/master/tutorial
MedCATtrainer: https://github.com/CogStack/MedCATtrainer
MedCATtrainer Examples: https://github.com/CogStack/MedCATtrainer/tree/master/docs
MedCATservice: https://github.com/CogStack/MedCATservice
CogStack: https://github.com/CogStack/CogStack-Pipeline

Data access ethics

NER+L experiments use freely available open-access datasets accessible by data owners. SNOMED-CT and UMLS licences were obtained by all users at all hospital sites. Site specific ethics is listed below. KCH: This project operated under London South East Research Ethics Committee approval (reference 18/LO/2048) granted to the King's Electronic Records Research Interface (KERRI); specific work on research on natural language processing for clinical coding was reviewed with expert patient input on the KERRI committee with Caldicott Guardian oversight. Direct access to patient-level data is not possible due to risk of re-identification, but aggregated de-identified data may be available subject to legal permissions. UCLH: UCLH is deploying CogStack within its records management infrastructure and is growing its capacity to annotate its clinical records as part of wider work for routine curation. The work at UCLH described here is a service evaluation that represents MedCAT's annotation of the records. Access to the medical records will not be possible given their confidential nature. SLaM: This project was approved by the CRIS Oversight Committee which is responsible for ensuring all research applications comply with ethical and legal guidelines. The CRIS system enables access to anonymized electronic patient records for secondary analysis from SLaM and has full ethical approvals. CRIS was developed with extensive involvement from service users and adheres to strict governance frameworks managed by service users. It has passed a robust ethics approval process acutely attentive to the use of patient data. Specifically, this system was approved as a dataset for secondary data analysis on this basis by Oxfordshire Research Ethics Committee C (08/H06060/71). The data is de-identified and used in a data-secure format and all patients have the choice to opt-out of their anonymized data being used. Approval for data access can only be provided from the CRIS Oversight Committee at SLaM.

Authors' contribution

ZK, TS, JT, RD, AS, AF conceived the study design. ZK, TS, AS, LR, KN performed data processing and software development. ZK, TS, JT, AS, AM, LZ, ADS performed data validation. RD, JT, RS, ZI, AR, DB, ZI, RB, MPR, ADS, AM performed critical review. TS, ZK, AS, LR, ZI, RB, DB, AM, RD wrote the manuscript

Conflict of interest

JTHT received research support and funding from InnovateUK, Bristol-Myers-Squibb, iRhythm Technologies, and holds shares < £5000 in Glaxo Smithkline and Biogen.

Acknowledgements

We would like to thank all the clinicians who provided annotation training for MedAT; this includes Rosita Zakeri, Kevin O'Gallagher, Rosemary Barker, David Nicholson Thomas, Rhian Raftopoulos, Pedro Viana, Elisa Bruno, Eugenio Abela, Mark Richardson, Naoko Skiada, Luwaiza Mirza, Natalia Chance, Jaya Chaturvedi, Tao Wang, Matt Solomon, Charlotte Ramsey and James Teo.

RD's work is supported by (1) National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London. (2) Health Data Research UK, which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, Economic and Social Research Council, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), British Heart Foundation and Wellcome Trust. (3) The National Institute for Health Research University College London Hospitals Biomedical Research Centre.

DMB is funded by a UKRI Innovation Fellowship as part of Health Data Research UK MR/S00310X/1 (https://www.hdruk.ac.uk).

RB is funded in part by grant MR/R016372/1 for the King's College London MRC Skills Development Fellowship programme funded by the UK Medical Research Council (MRC, https://mrc.ukri.org) and by grant IS-BRC-1215-20018 for the National Institute for Health Research (NIHR, https://www.nihr.ac.uk) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London.

ADS is supported by a postdoctoral fellowship from THIS Institute. AS is supported by a King's Medical Research Trust studentship. RS is part-funded by: (i) the National Institute for Health Research (NIHR) Biomedical Research Centre at the South London and Maudsley NHS Foundation Trust and King's College London; (ii) a Medical Research Council (MRC) Mental Health Data Pathfinder Award to King's College London; (iii) an NIHR Senior Investigator Award; (iv) the National Institute for Health Research (NIHR) Applied Research Collaboration South London (NIHR ARC South London) at King's College Hospital NHS Foundation Trust.

This paper represents independent research part funded by the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust, The UK Research and Innovation London Medical Imaging & Artificial Intelligence Centre for Value Based Healthcare (AI4VBH); the National Institute for Health Research (NIHR) Applied Research Collaboration South London (NIHR ARC South London) and King's College London. The views expressed are those of the author(s) and not necessarily those of the NHS, MRC, NIHR or the Department of Health and Social Care. We thank the patient experts of the KERRI committee, Professor Irene Higginson, Professor Alastair Baker, Professor Jules Wendon, Professor Ajay Shah, Dan Persson and Damian Lewsley for their support.

Appendix A. SNOMED-CT groupings

Each group was defined with expert clinical guidance. S-267036007 – dyspnea (finding), S-59282003 – pulmonary embolism, (disorder) S-29857009 – chest pain (finding) do not appear in the SLaM annotations for supervised training.

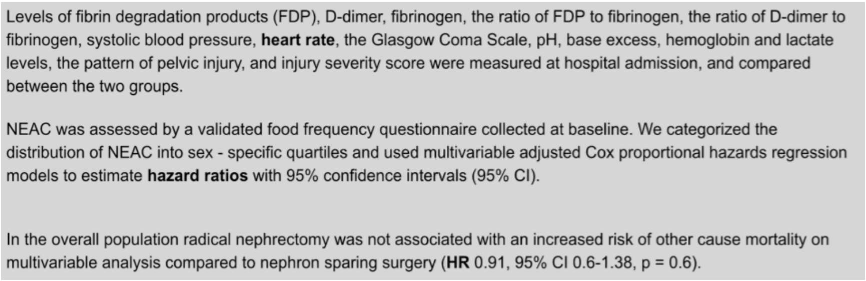
Table A.1
SNOMED-CT concept level groupings for clinical use cases.

Container concept	Concepts
S-73211009 – Diabetes mellitus(disorder)	S-44054006 – Diabetes mellitus type 2 (disorder) S-46635009 – Diabetes mellitus type 1 (disorder) S-422088007 – Disorder of nervous system co-occurrent and due to diabetes mellitus (disorder) S-25093002 – Disorder of eye co-occurrent and due to diabetes mellitus (disorder) S-73211009 – Diabetes mellitus (disorder)
S-84114007 -Heart failure (disorder)	S-128404006 – Right heart failure (disorder) S-48447003 – Chronic heart failure (disorder) S-56675007 – Acute heart failure (disorder) S-85232009 – Left heart failure (disorder) S-42343007 – Congestive heart failure (disorder) S-84114007 – Heart failure (disorder)
S-414545008 – Ischemic heart disease (disorder)	S-413439005 – Acute ischemic heart disease (disorder) S-413838009 – Chronic ischemic heart disease (disorder) S-194828000 – Angina (disorder) S-22298006 – Myocardial infarction (disorder) S-414545008 – Ischemic heart disease (disorder)
S-38341003 – Hypertensive disorder, systemic arterial (disorder)	S-31992008 – Secondary hypertension (disorder) S-48146000 – Diastolic hypertension (disorder) S-56218007 – Systolic hypertension (disorder) S-59621000 – Essential hypertension (disorder) S-38341003 – Hypertensive disorder systemic arterial (disorder)
S-13645005 – Chronic obstructive lung disease (disorder)	S-195951007 – Acute exacerbation of chronic obstructive airways disease (disorder) S-87433001 – Pulmonary emphysema (disorder) S-13645005 – Chronic obstructive lung disease (disorder)
S-195967001 – Asthma (disorder)	S-195967001 – Asthma (disorder)
S-709044004 – Chronic kidney disease (disorder)	S-723190009 – Chronic renal insufficiency (disorder) S-709044004 – Chronic kidney disease (disorder)
S-230690007 – Cerebrovascular accident (disorder)	S-25133001 – Completed stroke (disorder) S-371040005 – Thrombotic stroke (disorder) S-371041009 – Embolic stroke (disorder) S-413102000 – Infarction of basal ganglia (disorder) S-422504002 – Ischemic stroke (disorder) S-723082006 – Silent cerebral infarct (disorder) S-1078001000000105 – Haemorrhagic stroke (disorder) S-230690007 – Cerebrovascular accident (disorder)
S-266257000 – Transient ischemic attack (disorder)	S-266257000 – Transient ischemic attack (disorder)
S-84757009 – Epilepsy (disorder)	S-352818000 – Tonic-clonic epilepsy (disorder) S-19598007 – Generalized epilepsy (disorder) S-230456007 – Status epilepticus (disorder) S-509341000000107 – Petit-mal epilepsy (disorder) S-84757009 – Epilepsy (disorder)
S-49436004 – Atrial fibrillation (disorder)	S-49436004 – Atrial fibrillation (disorder)
S-267036007 – Dyspnea (finding)	S-267036007 – Dyspnea (finding)
S-59282003 – Pulmonary embolism (disorder)	S-59282003 – Pulmonary embolism (disorder)
S-29857009 – Chest pain (finding)	S-29857009 – Chest pain (finding)

Appendix B. Estimating example counts for sufficient F1 score

To test the required number of examples to achieve a high enough F1 score, we created a mini-dataset from MedMentions. It contains two concepts: C0018810 (heart rate) and C2985465 (hazard ratio). Figure B.4 shows an example texts for both concepts. Both concepts have a unique name and the ambiguous abbreviation HR that can link to either one. We chose these two concepts, as the abbreviation HR is the most frequent ambiguous concept in MedMentions, given the requirement that it must be ambiguous. Our dataset consists of:

- 60 training examples (30 per concept). In each example the full name of the concept was used, see below MedMentions Text Extracts.
- 174 test examples, each document contains the ambiguous abbreviation HR, see below MedMentions Text Extracts.



Levels of fibrin degradation products (FDP), D-dimer, fibrinogen, the ratio of FDP to fibrinogen, the ratio of D-dimer to fibrinogen, systolic blood pressure, **heart rate**, the Glasgow Coma Scale, pH, base excess, hemoglobin and lactate levels, the pattern of pelvic injury, and injury severity score were measured at hospital admission, and compared between the two groups.

NEAC was assessed by a validated food frequency questionnaire collected at baseline. We categorized the distribution of NEAC into sex - specific quartiles and used multivariable adjusted Cox proportional hazards regression models to estimate **hazard ratios** with 95% confidence intervals (95% CI).

In the overall population radical nephrectomy was not associated with an increased risk of other cause mortality on multivariable analysis compared to nephron sparing surgery (**HR** 0.91, 95% CI 0.6-1.38, p = 0.6).

Fig. B.4. MedMentions text extracts: three samples from the dataset used to test the amount of training samples needed for disambiguation to work. First example is a training case for the concept C0018810, second for C2985465 and third is used to test the disambiguation performance.

Table B.2
Relation between the number of training examples and performance of MedCAT concept disambiguation.

Number of examples per concept	F1 on Test
1	0.74
5	0.81
10	0.82
30	0.86

We have tested the performance for different sizes of the training set: 1, 5, 10 and 30. If we set the training set size to, e.g. 5, we split the full training set into 6 parts (in total the training set has 30 examples per concept), each containing 5 examples per concept. Then we check the performance for each part and report the average over the 6 parts, see Table B.2.

Appendix C. Self-supervised training configuration

C.1 Self-supervised training configuration

MedCAT was configured for self-supervised training across experiments presented in Section 2.1 as follows:

- Misspelled words were fixed only when 1 change away from the correct word for words under 6 characters, and 2 changes away for words above 6 characters.
- For each concept we calculate long and short embeddings and take the average of both. The long embedding takes into account $s = 9$ words from left and right (as shown in Eq. (2)). The short embedding takes into account $s = 2$ words from left and right. The exact numbers for s were calculated by testing the performance of all possible combinations for s in the range $[0, 10]$.
- The context similarity threshold used for recognition is 0.3 unless otherwise specified. This means for a given concept candidate, or sequence of words, to be recognized and linked to the given concept the concept similarity provided by Eq. (2) would be greater than 0.3.

C.2 Qualitative analysis training configuration

We train MedCAT self-supervised over MIMIC-III using the entirety of UMLS, 3.82 Million concepts from 207 separate vocabularies. We use $\sim$2.4M clinical notes (nursing notes, notes by clinicians, discharge reports etc.) on a small one-core server taking approximately 30 hours to complete.

Appendix D. Contextualization task results per site

D.1 Contextualization results breakdown for KCH

Table D.3 shows aggregate results for each defined meta-annotation at KCH. Performance is aggregated over all extracted concepts listed in Appendix A. We defined the following meta-annotation tasks:

- Presence: is the concept affirmed, negated or hypothetical, values: [Affirmed, Negated, Hypothetical]

Table D.3
Meta annotation results at KCH.

(a) Presence average 10 fold CV 90/10 ratio

CLS	F	P	R	Support test (10% of total)
Hypothetical	0.756	0.797	0.72	360
Negated	0.865	0.878	0.852	440
Affirmed	0.955	0.961	0.951	2930
Macro	0.86	0.875	0.846	3731
Weighted	0.927	0.927	0.929	3731

(b) Experiencer average 10 fold CV 90/10 ratio

CLS	F1	P	R	Support test (10% of total)
Family	0.801	0.865	0.751	13
Other	0.823	0.838	0.809	205
Patient	0.977	0.975	0.98	1649
macro	0.867	0.893	0.847	1867
weighted	0.959	0.959	0.959	1867

(c) Temporality average 10 fold CV 90/10 ratio

CLS	F	P	R	Support test (10% of total)
Recent	0.969	0.964	0.94	1655
Past	0.771	0.807	0.74	162
Future	0.667	0.706	0.74	50
macro	0.803	0.825	0.783	1867
weighted	0.943	0.943	0.945	1867

- Experiencer: is the concept experienced by the patient or other, values: [Patient/Family/Other]
- Temporality: is the concept in the past, present or future, values: [Past, Recent, Future]

Table D.4
Meta annotation results at SLaM.

CLS	F	P	R	Support test (10% of total)
(a) Status average 10 fold CV 90/10 ratio				
NA	0.873	0.869	0.878	43
Other	0.544	0.663	0.475	7
Affirmed	0.908	0.893	0.924	60
Macro	0.775	0.812	0.757	109
Weighted	0.873	0.874	0.873	109
(b) Diagnosis average 10 fold CV 90/10 ratio				
Yes	0.931	0.935	0.926	68
No	0.872	0.889	0.880	39
Macro	0.904	0.908	0.905	109
Weighted	0.913	0.912	0.913	109

D.2 Meta annotation results breakdown for SLaM

Table D.4 shows aggregate results for each defined meta-annotation at SLaM. Performance is aggregated over all extracted concepts listed in Appendix A. We defined the following meta-annotation tasks:

- Status: is the concept affirmed to be affecting the patient or not, values: [Patient/Other/NA]
- Diagnosis: is the concept a diagnosis related to the patient, or not, values: [Yes, No]

D.3 Meta annotation results breakdown for UCLH

Table D.5 shows aggregate results for each defined meta-annotation at UCLH. Performance is aggregated over all extracted concepts listed in Appendix A. We defined the following meta-annotation tasks:

- Negation: is the concept negated or not, values: [Yes/No]
- Experiencer: is the concept experienced by the patient or not, values: [Patient, Other]
- Problem Temporality: is the concept referring to a historical mention, values [Past Medical Issue, Current Problem]
- Certainty: is the concept confirmed to be present, values: [Confirmed, Suspected]

Table D.5
Meta annotation results at UCLH.

CLS	F	P	R	Support test (10% of total)
(a) Negation: average 10 fold CV 90/10 ratio				
Yes	0.896	0.895	0.900	46
No	0.688	0.767	0.631	394
Macro	0.836	0.767	0.631	440
Weighted	0.970	0.969	0.971	440
(b) Experiencer: average 10 fold CV 90/10 ratio				
Other	0.681	0.883	0.65	3
Patient	0.998	0.997	0.999	437
Macro	0.940	0.940	0.825	440
Weighted	0.996	0.996	0.996	440
(c) Problem Temporality: average 10 fold CV 90/10 ratio				
Past Medical Issue	0.710	0.758	0.676	23
Current Problem	0.985	0.981	0.988	412
Macro	0.848	0.870	0.832	435
Weighted	0.970	0.969	0.971	435
(d) Certainty: average 10 fold CV 90/10 ratio				
Confirmed	0.985	0.980	0.989	395
Suspected	0.688	0.767	0.631	21
Macro	0.836	0.874	0.810	416
Weighted	0.970	0.970	0.971	416
(e) Irrelevant: average 10 fold CV 90/10 ratio				
Yes	0.685	0.846	0.579	24
No	0.986	0.976	0.994	415
Macro	0.835	0.911	0.787	439
Weighted	0.969	0.970	0.972	439

- Irrelevant: is the concept relevant, values: [Yes, No]

References

[1] Jackson R, Kartoglu I, Stringer C, Gorrell G, Roberts A, Song X, et al. CogStack – experiences of deploying integrated information retrieval and extraction services in a large national health service foundation trust hospital. BMC Med Inform Decis Mak 2018;18(1):47. https://doi.org/10.1186/s12911-018-0623-9.

[2] Stearns MQ, Price C, Spackman KA, Wang AY. SNOMED clinical terms: overview of the development process and project status. Proc AMIA Symp 2001:662–6.

[3] Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res 2004;32(Database issue):D267–70. https://doi.org/10.1093/nar/gkh061.

[4] Searle T, Kraljevic Z, Bendayan R, Bean D, Dobson R. MedCATTrainer: a biomedical free text annotation interface with active learning and research use case specific customisation. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP): system demonstrations, association for computational linguistics; 2019. p. 139–44. https://doi.org/10.18653/v1/D19-3024.

[5] Bean DM, Kraljevic Z, Searle T, Bendayan R, Kevin O G, Pickles A, et al. ACE-inhibitors and angiotensin-2 receptor blockers are not associated with severe SARS-COVID19 infection in a multi-site UK acute hospital trust. Eur J Heart Fail 2020. https://doi.org/10.1002/ejhf.1924.

[6] Carr E, Bendayan R, Bean D, Stammers M, Wang W, Zhang H, et al. Evaluation and improvement of the national early warning score (NEWS2) for COVID-19: a multi-hospital study. 2020. https://doi.org/10.1101/2020.04.24.20078006.

[7] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, editors. Advances in neural information processing systems 30. Curran Associates, Inc; 2017. p. 5998–6008.

[8] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8): 1735–80.

[9] Howard J, Ruder S. Universal language model fine-tuning for text classification. 2018. arXiv:1801.06146.

[10] Jonquet C, Shah NH, Musen MA. The open biomedical annotator. Summit Transl Bioinform 2009;2009:56–60.

[11] Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Sci Rep 2016;6: 26094. https://doi.org/10.1038/srep26094.

[12] Landi I, Glicksberg BS, Lee H-C, Cherng S, Landi G, Danieletto M, et al. Deep representation learning of electronic health records to unlock patient stratification at scale. NPJ Digit Med 2020;3:96. https://doi.org/10.1038/s41746-020-0301-z.

[13] Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc 2010;17(3):229–36. https://doi.org/10.1136/jamia.2009.002733.

[14] Gorrell G, Song X, Roberts A. Bio-YODIE: a named entity linking system for biomedical text. 2018. arXiv:1811.04860.

[15] Wu H, Toti G, Morley KI, Ibrahim ZM, Folarin A, Jackson R, et al. SemEHR: a general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research. J Am Med Inform Assoc 2018;25(5):530–7. https://doi.org/10.1093/jamia/ocx160.

[16] Gorinski PJ, Wu H, Grover C, Tobin R, Talbot C, Whalley H, et al. Named entity recognition for electronic health records: a comparison of rule-based and machine learning approaches. 2019. arXiv:1903.03985.

[17] Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc 2010;17(5): 507–13. https://doi.org/10.1136/jamia.2009.001560.

[18] Ferrucci D, Lally A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Nat Lang Eng 2004:1–26.

[19] Morton T, Kottmann J, Baldridge J, Bierner G. Opennlp: a java-based nlp toolkit. Proc EACL 2005.

[20] Neumann M, King D, Beltagy I, Ammar W. ScispaCy: fast and robust models for biomedical natural language processing. 2019. arXiv:1902.07669.

[21] Soysal E, Wang J, Jiang M, Wu Y, Pakhomov S, Liu H, et al. Clamp – a toolkit for efficiently building customized clinical natural language processing pipelines. J Am Med Inform Assoc 2017. https://doi.org/10.1093/jamia/ocx132. ocx132.

[22] Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, et al. BioPortal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Res 2011;39(Web Server issue):W541–5. https://doi.org/10.1093/nar/gkr469.

[23] Keselman A, Smith CA, Divita G, Kim H, Browne AC, Leroy G, et al. Consumer health concepts that do not map to the UMLS: where do they fit? J Am Med Inform Assoc 2008;15(4):496–505. https://doi.org/10.1197/jamia.M2599.

[24] Wang KC. Standard lexicons, coding systems and ontologies for interoperability and semantic computation in imaging. J Digit Imaging 2018;31(3):353–60. https://doi.org/10.1007/s10278-018-0069-8.

[25] Data protection and information governance. https://www.hra.nhs.uk/planning-and-improving-research/policies-standards-legislation/data-protection-and-information-governance/ [accessed 31 July 2020].

[26] Hellrich J, Hahn U. Fostering multilinguality in the UMLS: a computational approach to terminology expansion for multiple languages. AMIA Annu Symp Proc 2014;2014. 655-660d.

[27] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in neural information processing systems 26. Curran Associates, Inc; 2013. p. 3111–9.

[28] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. 2018. arXiv:1810.04805.

[29] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Trans Assoc Comput Linguist 2017;5:135–46. https://doi.org/10.1162/tacl_a_00051.

[30] Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014:1532–43.

[31] Johnson AEW, Pollard TJ, Shen L, Lehman L-WH, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Sci Data 2016;3:160035. https://doi.org/10.1038/sdata.2016.35.

[32] Luo L, Yang Z, Yang P, Zhang Y, Wang L, Lin H, et al. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. Bioinformatics 2018;34(8):1381–8. https://doi.org/10.1093/bioinformatics/btx761.

[33] Wang X, Zhang Y, Ren X, Zhang Y, Zitnik M, Shang J, et al. Cross-type biomedical named entity recognition with deep multi-task learning. Bioinformatics 2019;35 (10):1745–52. https://doi.org/10.1093/bioinformatics/bty869.

[34] Xu B, Shi X, Zhao Z, Zheng W. Leveraging biomedical resources in Bi-LSTM for drug–drug interaction extraction. IEEE Access 2018;6:33432–9. https://doi.org/10.1109/ACCESS.2018.2845840.

[35] Mascio A, Kraljevic Z, Bean D, Dobson R, Stewart R, Bendayan R, et al. Comparative analysis of text classification approaches in electronic health records. Proceedings of the 19th SIGBioMed workshop on biomedical language processing. Stroudsburg, PA, USA: Association for Computational Linguistics; 2020. p. 86–94. https://doi.org/10.18653/v1/2020.bionlp-1.9.

[36] Mohan S, Li D. MedMentions: a large biomedical corpus annotated with UMLS concepts. 2019. arXiv:1902.09476.

[37] Mowery DL, Velupillai S, South BR, Christensen L, Martinez D, Kelly L, et al. Task 2: ShARe/CLEF ehealth evaluation lab 2014. 2014.

[38] Bodenreider O, McCray AT. Exploring semantic groups through visual approaches. J Biomed Inform 2003;36(6):414–32.

[39] Alsentzer E, Murphy JR, Boag W, Weng W-H, Jin D, Naumann T, et al. Publicly available clinical BERT embeddings. 2019. arXiv:1904.03323.

[40] Beam AL, Kompa B, Schmaltz A, Fried I, Weber G, Palmer NP, et al. Clinical concept embeddings learned from massive sources of multimodal medical data. 2018. arXiv:1804.01486.

[41] Zakeri R, Bendayan R, Ashworth M, Bean DM, Dodhia H, Durbaba S, et al. A case–control and cohort study to determine the relationship between ethnic background and severe COVID-19. 2020. https://doi.org/10.1101/2020.07.08.20148965.

[42] Fraser KC, Nejadgholi I, De Bruijn B, Li M, LaPlante A, El Abidine KZ. Extracting UMLS concepts from medical text using general and domain-specific deep learning models. 2019. arXiv:1910.01274.

[43] Tshitoyan V, Dagdelen J, Weston L, Dunn A, Rong Z, Kononova O, et al. Unsupervised word embeddings capture latent knowledge from materials science literature. Nature 2019;571(7763):95–8. https://doi.org/10.1038/s41586-019-1335-8.

[44] Krauthammer M, Nenadic G. Term identification in the biomedical literature. J Biomed Inform 2004;37(6):512–26. https://doi.org/10.1016/j.jbi.2004.08.004.

3.2.1 Discussion

A MedCAT model is comprised of two parts. A Concept database *(CDB)* is a collection of biomedical terms and their representative fixed-length vector embedding for each concept. A Vocabulary *(VCB)* is a collection words that represent all possible words in a language, and each word's associated fixed-length vector embedding. The VCB is pretrained and does not change according to any training that occurs during normal MedCAT operation, training and/or inference. The specific concept embedding within the CDB is updated during training according to the algorithm described in Section 2.2.

In Section 2.1.2 of the paper presented in Section 3.2, the spell-checking process is described. The concept of an abbreviation, however, is not fully described. An abbreviation is an alternative shortened version of a given concept as specified during the building of the source CDB data. Source terminology data used to build a CDB often contains the biomedical concept name and any 'known' abbreviations or synonyms of that term. If any span of text matches a given concept's synonym spelling then this is not 'corrected' by the spell-checker.

In Section 4.3 of the paper the limitations are described. I further expand this section to describe algorithm errors in various forms. Firstly, MedCAT can miss spans of text that should be a recognised entity but are not. These errors are due to spans of text that are not listed within the MedCAT CDB, either because the original terminology does not include this missing span, or during training this span has never been encountered and added by an annotator. Common short hands for some clinical events (disorders and a measure) are presented in Table 3.1

A second type of error is the incorrect linking of a span of text to a CDB concept.

MedCAT annotations that are predictions of the model can be fixed and linked to the 'correct' concept through the MedCATtrainer interface. For example, a span of text could link the text "DM" to "diabetes mellitus", but this should be linked to more specific diabetic retinopathy a common complication for diabetes patients as retinopathy is mentioned elsewhere in the context of the current document. This specialisation of models to specific clinical areas and certain acronyms holding multiple meanings across clinical domains further suggests the need for models to be continually validated.

Both errors are addressed through supervised training, manual annotations and running of the training process over this collected data.

3.3 MedCATtrainer - The MedCAT Annotation Tool

MedCAT is a toolkit for the development, validation and ongoing fine-tuning of named entity recognition and linking (NER+L) models for clinical concept extraction and the further contextualisation of extracted concepts. An important part of of this workflow is the partnership with clinical collaborators who are using the downstream output. Building effective applied AI in a clinical setting requires an interdisciplinary approach, that engages domain experts early and throughout the process of model development and ongoing maintenance [142, 76].

This next published work provides further details, and empirical evidence suggesting the MedCATtrainer interface supports this previously described workflow of MedCAT model validation and fine-tuning. Importantly, it closely integrates with MedCAT models and supports a seamless means to both validate and fine-tune MedCAT models.

MedCATTrainer: A Biomedical Free Text Annotation Interface with Active Learning and Research Use Case Specific Customisation

Thomas Searle[1], **Zeljko Kraljevic**[1], **Rebecca Bendayan**[1],
Daniel Bean[1], **Richard Dobson**[1,2]
[1]Department of Biostatistics and Health Informatics,
Kings College London, London, U.K.
[2]Institute of Health Informatics, University College London,
222 Euston Road, London NW1 2DA, U.K.
{firstname.lastname}@kcl.ac.uk

Abstract

We present MedCATTrainer[1] an interface for building, improving and customising a given Named Entity Recognition and Linking (NER+L) model for biomedical domain text. NER+L is often used as a first step in deriving value from clinical text. Collecting labelled data for training models is difficult due to the need for specialist domain knowledge. MedCATTrainer offers an interactive web-interface to inspect and improve recognised entities from an underlying NER+L model via active learning. Secondary use of data for clinical research often has task and context specific criteria. MedCATTrainer provides a further interface to define and collect supervised learning training data for researcher specific use cases. Initial results suggest our approach allows for efficient and accurate collection of research use case specific training data.

1 Introduction

We present a flexible web-based open-source use-case configurable interface and workflow for biomedical text concept annotation - MedCATTrainer[2].

Murdoch and Detsky (2013) estimates that 80% of biomedical data is stored in unstructured text such as Electronic health records (EHRs). Although EHRs have seen widespread global adoption, effective secondary use of the data remains difficult (Elkin et al., 2010). However, significant progress has been made on agreement and usage of standardised terminologies such as the Systematized Nomenclature of Medical Clinical Terms (SNOMED-CT) (Stearns et al., 2001) and the Unified Medical Language System (UMLS)(Bodenreider, 2004). Annotating EHR text with these concept databases is often seen as a first step in delivering data driven applications such as precision medicine, clinical decision support or real time disease surveillance (Assale et al., 2019).

EHR text annotation is challenging due to the use of domain specific terms, abbreviations, misspellings and terseness. Text can also be 'copy-pasted' from prior notes, structured tables entered into unstructured form, content with varying temporality and scanned images of physical documents (Botsis et al., 2010). Annotation is further complicated as researchers have task and context specific parameters. For example, whether family history or suspected diagnoses are considered relevant to the task.

MedCAT[3], manuscript in preparation (Zeljko and Lucasz, 2019), is a **Med**ical **C**oncept **A**nnotation **T**ool that uses unsupervised machine learning to recognise and link medical concepts with clinical terminologies such as UMLS. MedCAT, like similar tools, uses a concept database to find and link concept mentions inside of biomedical documents. In addition it has disambiguation, spell-checking and the option for supervised learning for improved disambiguation.

We introduce a novel web based application that supplements usage of a biomedical NER+L models, such as MedCAT. Our contributions are as follows:

1. **Concept Inspection and Addition:** an interface that to inspect the identified concepts from free text, and add missing concepts to an existing NER+L model. This interface aligns with MedCAT, but could also be used with other models that have similar capabilities.

2. **Active Learning:** an interface for active learning, enabling users to provide minimal

[1] https://www.youtube.com/watch?v=lM914DQjvSo
[2] https://github.com/CogStack/MedCATtrainer
[3] https://github.com/CogStack/MedCAT

Proceedings of the 2019 EMNLP and the 9th IJCNLP (System Demonstrations), pages 139–144
Hong Kong, China, November 3 – 7, 2019. ©2019 Association for Computational Linguistics

training data to assist in improving and correcting the NER+L. This interface requires that the backing NER+L system supports active learning.

3. **Clinical Research Question Specific Annotation:** a further interface for configurable use case specific annotation of identified concepts. Allowing for the collection of research question specific training data. For example, annotating specific temporal features of a concept.

2 Related Work

Outside of the biomedical domain general purpose annotation interfaces have been developed for most popular NLP tasks such as NER, NEL, relation extraction, entity normalisation, dependency parsing, chunking etc. Popular choices include open-source tools such as BRAT (Stenetorp et al., 2012) that also allows for managing the distribution, monitoring and collection of annotated corpora. General purpose tools with active learning include the commercial product Prodigy[4]. Although these tools are mature and offer advanced features they can be complex to setup and do not offer integration with existing biomedical domain NER+L systems.

Prior work on biomedical NER+L includes MetaMAP (Aronson, 2001) and CTakes (Savova et al., 2010). Both have provided interfaces to inspect recognised entities but they have not provided means to correct and amend concepts or specify further annotations for specific research questions.

Another tool for biomedical NER+L, SemEHR Wu et al. (2018), offers features to add custom pre and post processing steps and research specific use cases, but does not directly improve the NER+L model via an interface. Instead it treats the provided NER+L model as a black-box model.

3 MedCATTrainer

MedCATTrainer is a web-based interface for inspecting, adding and correcting biomedical NER+L models through active learning. An additional interface allows for research specific annotations to be defined and collected for training of supervised learning models.

The interfaces are built with Vue.js[5] for the front-end and the python[6] web framework Django[7] for the web API and integration with NER+L models such as MedCAT. We use the Django admin features to allow administrators to configure research question specific supervised learning tasks.

MedCATTrainer is deployed via a Docker[8] container. This ensures users can build, deploy and run MedCATTrainer cross-platform without lengthy build and run processes, advanced infrastructure knowledge or root access to systems. This is especially important in health informatics as hospital infrastructure is often restrictive. MedCATTrainer allows researchers to build on top of existing biomedical domain ontologies, such as UMLS, for two use cases. Firstly, improving the underlying NER+L model by adding synonyms, abbreviations, multi-token concepts and misspellings directly from the interface. Secondly, by allowing research use case specific annotations to be defined and collected for training of supervised learning models.

3.1 Concept Inspection and Addition

Figure 1a shows the ‘Train Annotations’ interface. Users can inspect and correct the concepts identified by the underlying NER+L model. Entities that have not been recognised can also be added to the NER+L model concept database. This allows researchers to test the learnt entity recognition/linking capabilities of the model whilst tailoring it to recognise sub-domain specific lexicon. This can include abbreviations or misspellings common to specific corpora. Figure 1b shows the form entry to add new concepts to the underlying concept database. Semantically equivalent texts can be added under the same Concept Unique Identifier along with synonyms. Advanced NER+L tools (e.g. MedCAT) learn from the contextual embeddings of words to disambiguate future occurrences. MedCATTrainer provides a textbox for entering the surrounding context tokens to assist with concept disambiguation.

3.2 Active Learning

Annotating biomedical domain text for NER+L requires expert knowledge and therefore cannot be

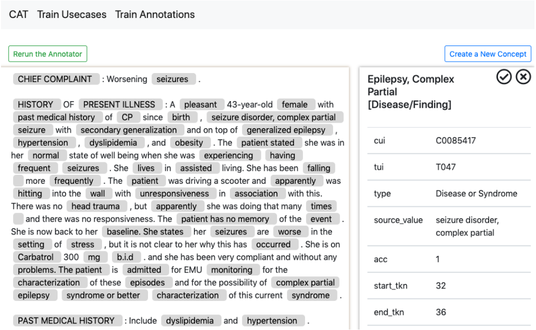
[4] https://explosion.ai/blog/prodigy-annotation-tool-active-learning

[5] https://vuejs.org/

[6] https://www.python.org/

[7] https://www.djangoproject.com/

[8] https://www.docker.com



CHIEF COMPLAINT : Worsening seizures .

HISTORY OF PRESENT ILLNESS : A pleasant 43-year-old female with past medical history of CP since birth , seizure disorder, complex partial seizure with secondary generalization and on top of generalized epilepsy , hypertension , dyslipidemia , and obesity . The patient stated she was in her normal state of well being when she was experiencing having frequent seizures . She lives in assisted living. She has been falling more frequently . The patient was driving a scooter and apparently was hitting into the wall with unresponsiveness in association with this. There was no head trauma , but apparently she was doing that many times and there was no responsiveness. The patient has no memory of the event . She is now back to her baseline. She states her seizures are worse in the setting of stress , but it is not clear to her why this has occurred . She is on Carbatrol 300 mg b.i.d . and she has been very compliant and without any problems. The patient is admitted for EMU monitoring for the characterization of these episodes and for the possibility of complex partial epilepsy syndrome or better characterization of this current syndrome .

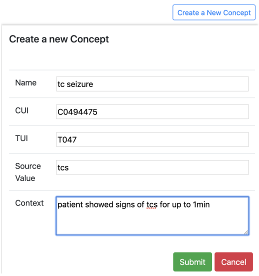
PAST MEDICAL HISTORY : Include dyslipidemia and hypertension .

Epilepsy, Complex Partial [Disease/Finding] ✓ ☒

cui	C0085417
tui	T047
type	Disease or Syndrome
source_value	seizure disorder, complex partial
acc	1
start_tkn	32
end_tkn	36

(a) The MedCATTrainer interface for viewing identified concepts by the underlying NER+L model of a publicly available[a] neurological consultation summary showing the concept metadata and active learning feedback input controls.

[a] https://bit.ly/2RLcdJx



Create a new Concept

Name	tc seizure
CUI	C0494475
TUI	T047
Source Value	tcs
Context	patient showed signs of tcs for up to 1min

(b) Side panel for the addition of new concepts.

Figure 1: The interfaces for inspecting annotations and the addition of concepts.

easily crowd sourced. Active learning is a common approach to provide a minimal set of high value training examples for manual annotation. Examples are valued with respect to expected improvement in classification performance once labelled and the model retrained (Settles, 2009).

We use a simple strategy of certainty based selective sampling (Lewis and Catlett, 1994) to display low confidence examples. Concretely, given a trained model $\mathbf{M}$, and the total set of annotations predicted on a new document d by model $\mathbf{M}$ is $\mathbf{L} = \{l_1, l_2, \ldots l_n\}$ where the model labelled the document with n annotations. An annotation l_i has an associated confidence c_{l_i} probability in the annotation. An annotation manager defines δ, a confidence cutoff score. The set of annotations $\mathbf{A}$ shown to an annotator is therefore $\Phi(\mathbf{L})$ where $\Phi(l_i) = c_{l_i} > \delta$.

Each human annotator is instructed to review each identified concept and provide feedback on correctness. Feedback is provided through the action of clicking the 'tick' for correct or 'cross' for incorrect as shown in the top right of Figure 1a.

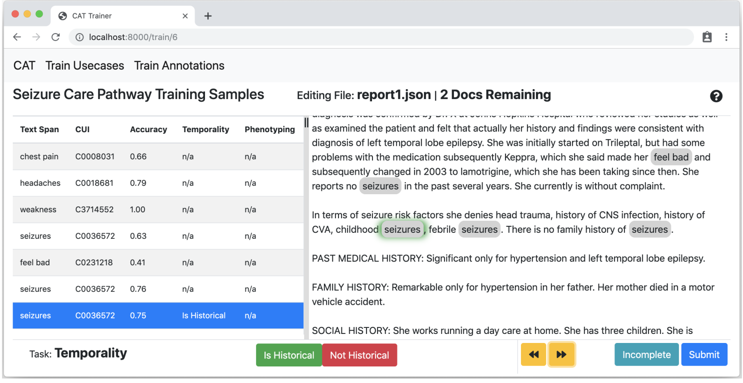
If an identified concept is incorrect human annotators are asked to provide feedback, rerun the NER+L model (top left 'Rerun the Annotator'), and then confirm if the misidentified concept has been corrected. More feedback can be provided if needed. Our pilot test users found this quickly resulted in the correctly identified and linked concept as text spans often only have one or two alternative concepts.

3.3 Clinical Research Question Specific Annotation

It would be infeasible to have a clinical terminology to define every possible contextual representation of a concept. For example, disambiguation of 'seizure' for a symptom of epilepsy and 'first seizure clinic' for a clinic that provides epilepsy care or 'history of seizures' for a historical case of epilepsy.

Our second interface solves this problem by allowing clinical researchers to define use case orientated tasks and associated annotations for previously identified and linked concepts. Custom classifiers are then trained and layered over the existing NER+L model for context specific concept disambiguation. An example configured screen for 'Temporality' and 'Phenotyping' tasks for an ongoing clinical research project is shown in Figure 2 - using replacement publicly available data. The top bar lists the overall task name followed by the number of documents to be annotated. The top right corner opens the current task help document, listing annotation guidelines for this use-case.

The left panel itemises each text span, the associated Concept Unique Identifier (CUI) - that the NER+L model has identified and linked with the text, and the current value of each task specific annotation. The value 'n/a' indicates the task has not been completed for that span. Users can choose any order of the text spans to annotate. The currently selected text span is highlighted in the table and within the central text area showing the entirety of the document. Clinical notes can be



Seizure Care Pathway Training Samples

Editing File: **report1.json | 2 Docs Remaining**

Text Span	CUI	Accuracy	Temporality	Phenotyping
chest pain	C0008031	0.66	n/a	n/a
headaches	C0018681	0.79	n/a	n/a
weakness	C3714552	1.00	n/a	n/a
seizures	C0036572	0.63	n/a	n/a
feel bad	C0231218	0.41	n/a	n/a
seizures	C0036572	0.76	n/a	n/a
seizures	C0036572	0.75	Is Historical	n/a

as examined the patient and felt that actually her history and findings were consistent with diagnosis of left temporal lobe epilepsy. She was initially started on Trileptal, but had some problems with the medication subsequently Keppra, which she said made her feel bad and subsequently changed in 2003 to lamotrigine, which she has been taking since then. She reports no seizures in the past several years. She currently is without complaint.

In terms of seizure risk factors she denies head trauma, history of CNS infection, history of CVA, childhood seizures, febrile seizures. There is no family history of seizures.

PAST MEDICAL HISTORY: Significant only for hypertension and left temporal lobe epilepsy.

FAMILY HISTORY: Remarkable only for hypertension in her father. Her mother died in a motor vehicle accident.

SOCIAL HISTORY: She works running a day care at home. She has three children. She is

Task: **Temporality**

Figure 2: Task and context specific annotation interface configured for 'Temporality' and 'Phenotype' tasks

long in length. Clicking a text span from the sidebar scrolls the central text area to the corresponding span assisting human annotators in locating the span to annotate. The text area also highlights each spans current annotated value for the current task.

The bottom bottom bar lists the current task and the possible annotation values. Figure 2 shows the 'Temporality' task and the associated annotation values 'Is Historical' and 'Not Historical'. The values are in context to a seizure care pathway use case and are defined as any currently experienced mention of seizure symptoms in present clinical encounter. Use cases and associated tasks values are configurable via the admin interface.

The bottom right corner provides navigation between text spans and tasks via the arrow buttons. Navigating between spans highlights the current span to be annotated in the main left sidebar and auto scrolls to the next span in the main text area. The navigation controls here, the sidebar and the main text area allow human annotators to complete the task in any order they are comfortable.

The 'Incomplete' button marks the current document to be revisited at a later date. Samples are marked incomplete if the NER+L model has misidentified the concept or there is a genuine ambiguity. The 'Submit' button marks the document as complete. Both actions store and retrieve the next document if there is one available. If there are no more files to annotate a dialog prompts the user to return to the home screen.

Corpora are currently directly uploaded via a use case management screen. Future deployments will directly ingest documents via an elasticsearch[9] connector to hospital EHR deployments of CogStack (Jackson et al., 2018) an EHR ingestion, transformation and search service deployed at King's College Hospital (KCH) and South London and Maudsley(SLaM) NHS Foundation Trusts, UK.

4 Results

We ran an initial small scale pilot experiment to test the suitability of our use case specific tool to quickly and accurately collect training data labelling the temporal features of seizure symptoms. This is similar to the task shown in Figure 2. We used MIMIC3 (Johnson et al., 2016), a de-identified publicly available database of ICU admission data that includes observations, consultation and discharge summary reports. We randomly sampled 127 discharge summaries that contained one or more token occurrences that match the regular expression 'seizure|seizre|seizur|siezure', where | is an OR operator between the text tested to be present. We intentionally rely on a rule-based NER mode (i.e. the regex) here to demonstrate our tools flexibility to use possible alternatives to MedCAT if desired.

We asked 2 human non-clinical annotators to label temporal features of each occurrence in relation to a 'present', i.e. 'chief complaint: seizure' or 'historical', i.e. 'family history of seizures', mention of the term. Both took approximately 35 minutes to review all 127 documents. We achieve an percent agreement of 89% and a Cohen's Kappa $\kappa = 0.695$, Table 1. Both annotators marked some records as incomplete as they either mostly referred to non symptomatic mentions

[9] https://www.elastic.co/

	R1*	R2*	R1	R2
# Documents	107	117	100	100
# Concepts	351	344	317	317
# Historical	67	80	79	65
# Not Historical	276	264	238	252

Table 1: Total labelled ‘seizure’ symptom concepts and for each human annotator (R1, R2) for the ‘temporality’ task of labelling concepts that have occurred the past relative to the hospital episode. * indicates raw numbers before taking into account the intersection of notes between annotators

of seizure, i.e. ‘anti-seizure meds prophylaxis’ or the prevention of future seizures. This resulted in each rater having differing total documents ‘submitted’ as there are some document with mixes of the above occurrences. We took the intersection of submitted documents from both raters to compute the final agreement scores.

Using the collected data we fit a simple Sckit-learn[10] Random Forest (RF) classifier model demonstrating the effectiveness of the data collection in being able to easily fit a well performing model for the task of recognising temporality of seizure symptoms. We took a random 70/30 train test split, took 100 characters either side of the labelled ‘seizure’ occurrence, tokenized the plain text on whitespace then used a TF-IDF vectoriser with the default English stop-words list. We ran a grid search across TF-IDF and random forest classifier parameters, with a 3 fold cross validation and found the best fitting parameters: TF-IDF features 500 (range:500, 1000, 10000), RF maximum number trees of 100 range(100, 300, 500, 1000) and maximum tree depth 20 (range: 5, 20, 50, 75). We achieve an accuracy of this binary classification task of 92% and f1 score .79.

5 Discussion and Future Work

From our labelling exercise we demonstrate the speed and accuracy of our configurable use case specific interface. Strong scores across % agreement, Cohen’s Kappa and trained model accuracy indicate good agreement between annotators, interpretations of the task and reasonable signal captured even with this small data set. Although, it is likely the model is over-fitting due to the size of the data set. Given the prior experiment - across two raters - gathering enough accurate data to, for example, fine-tune a pretrained language model based classifier would be of the order of hours of manual labelling for approx 2k samples. We see this rapid labelling ability as a key strength of our interface.

We foresee that trained classifiers will likely generalise to additional research questions. For example language used to express temporality of seizures is likely to be similar to temporality of stroke or myocardial infarction.

Generally, training models across use cases will likely capture shared semantics. This suggests particular use cases would require less examples to train as annotated data or the model itself could be reused, therefore jump-starting clinical research. If a model is not performing for a new use case, further data could be collected to fine tune the model to a specific task, context or sub-domain corpora.

Clinically, domain experts in the neurology department of KCH, with varying levels of expertise (medical student to practising consultant) are scheduled to participate in the use case shown in Figure 2 in the coming months.

Our initial testing, not shown above due to space, of the active learning approach for improving the bound NER+L model suggests we can improve performance with minimal training data.

6 Conclusions

We have presented a lightweight, flexible, web-based, open-source annotation interface for biomedical domain text. MedCATTrainer is integrated with a biomedical NER+L model and allows for addition of missing concepts, improvements to the underlying NER+L model through active learning, and a configurable interface for clinical researchers to define annotations specific for their research questions. Preliminary results show promise for our interface and our approach to biomedical NER+L, which is often seen as a first step in deriving value from data sources such as electronic health records.

Acknowledgments

DMB is funded by a UKRI Innovation Fellowship as part of Health Data Research UK (MR/S00310X/1). RB is funded in part by grant MR/R016372/1 for the Kings College London MRC Skills Development Fellowship programme funded by the UK Medical Research Council

[10] https://scikit-learn.org/stable/index.html

(MRC) and by grant IS-BRC-1215-20018 for the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and Kings College London. RD's work is supported by 1.National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and Kings College London. 2. Health Data Research UK, which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, Economic and Social Research Council, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), British Heart Foundation and Wellcome Trust. 3. The National Institute for Health Research University College London Hospitals Biomedical Research Centre. This paper represents independent research part funded by the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and Kings College London. The views expressed are those of the author(s) and not necessarily those of the NHS, MRC, NIHR or the Department of Health and Social Care.

References

A R Aronson. 2001. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. *Proc. AMIA Symp.*, pages 17–21.

Michela Assale, Linda Greta Dui, Andrea Cina, Andrea Seveso, and Federico Cabitza. 2019. The revival of the notes field: Leveraging the unstructured content in electronic health records. *Front. Med.*, 6:66.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database issue):D267–70.

Taxiarchis Botsis, Gunnar Hartvigsen, Fei Chen, and Chunhua Weng. 2010. Secondary use of EHR: Data quality issues and informatics opportunities. *Summit Transl Bioinform*, 2010:1–5.

Peter L Elkin, Brett E Trusko, Ross Koppel, Ted Speroff, Daniel Mohrer, Saoussen Sakji, Inna Gurewitz, Mark Tuttle, and Steven H Brown. 2010. Secondary use of clinical data. *Stud. Health Technol. Inform.*, 155:14–29.

Richard Jackson, Ismail Kartoglu, Clive Stringer, Genevieve Gorrell, Angus Roberts, Xingyi Song, Honghan Wu, Asha Agrawal, Kenneth Lui, Tudor Groza, Damian Lewsley, Doug Northwood, Amos Folarin, Robert Stewart, and Richard Dobson. 2018. CogStack - experiences of deploying integrated information retrieval and extraction services in a large national health service foundation trust hospital. *BMC Med. Inform. Decis. Mak.*, 18(1):47.

Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Sci Data*, 3:160035.

David D Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In William W Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, San Francisco (CA).

Travis B Murdoch and Allan S Detsky. 2013. The inevitable application of big data to health care. *JAMA*, 309(13):1351–1352.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *J. Am. Med. Inform. Assoc.*, 17(5):507–513.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

M Q Stearns, C Price, K A Spackman, and A Y Wang. 2001. SNOMED clinical terms: overview of the development process and project status. *Proc. AMIA Symp.*, pages 662–666.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: A web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 102–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Honghan Wu, Giulia Toti, Katherine I Morley, Zina M Ibrahim, Amos Folarin, Richard Jackson, Ismail Kartoglu, Asha Agrawal, Clive Stringer, Darren Gale, Genevieve Gorrell, Angus Roberts, Matthew Broadbent, Robert Stewart, and Richard J B Dobson. 2018. SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research. *J. Am. Med. Inform. Assoc.*, 25(5):530–537.

Kraljevic Zeljko and Roguski Lucasz. 2019. Cogstack/medcat: First release of medcat.

Condition	Common Shorthand
Methicillin-resistant Staphylococcus aureus	MRSA
chronic obstructive pulmonary disease	COPD
Diabetes Mellitus Type 2	DM2, DMII
Heart Failure	HF
Heart Failure with Preserved Ejection Fraction	HFpEF
Left Ventricle Ejection Fraction	LVEF

Table 3.1 Clinical disorders and measures that appear in clinical text but are often referred to in their shorthand form. Representative of MedCAT errors, where the CDB has not encountered these text spans during training

3.3.1 Discussion

Overall, MedCATtrainer has been completely rewritten since the above paper to better support the aims of validating, fine-tuning and collecting high quality training data for MedCAT models. However, the application still retains all of the features described in the original paper. Both the screens for concept recognition and task specific annotation collections have been combined so concepts can be recognised - marked correct or incorrect and missing concepts can be added. Task specific annotations can also be quickly added for each recognised concept if needed. In summary, the latest features allow for:

1. Adding missing concepts by selecting text and looking up a missing concept within a linked MedCAT concept database.

2. A fast concept database lookup configurable for each MedCAT concept database.

3. A flexible and configurable relation annotation collection screen.

4. A flexible, multi-user, multi-annotation project system allowing for a single deployment to manage hundreds of individual projects with precise access control of annotation data.

MedCATtrainer is still in active development and I continue to make each new release opensource.

3.4 Clinical Research with MedCAT

This section will present and discuss the MedCAT and MedCATtrainer tools as enablers for clinical research. I will first critically demonstrate our analysis of fine-tuning models across various clinical research projects at KCH via the MedCATtrainer tool. Then I will briefly review clinical research papers that have used a MedCAT trained model to extract ad contextualise comorbidity terms.

3.4.1 MedCATtrainer Annotation Analysis

As of September 2022, the KCH deployment of MedCATtrainer has been used for 44 separate projects by 31 distinct users collecting >36k annotations for >4100 distinct clinical concepts (mostly SNOMED-CT), and >60k annotations for 3 meta annotation tasks.

The total number of unique concept forms evolve throughout annotation projects and the ratio of correct (blue area) / incorrect (orange area) annotations changes as our clinicians annotate documents. The number of word forms that appear within a single document can vary, hence the plot the ratio of correct / incorrect annotations per document relative to the number of word forms that have been seen during the annotation session (marked by the black line in each curve). A word form is a synonym for a concept. For example, the SNOMED CT concept myocardial infarction (SCTID: 22298006) can be referred to as 'heart attack', 'MI' or myocardial infarction'. This is an example of 3 word forms or unique concept forms for this one concept. Figure 3.1 shows multiple projects that initially use the KCH self-supervised trained SNOMED-CT model to annotate a range of concepts and how they uniquely appear in the texts. Visually, this is the area under the cumulative word forms line in each plot taken by either the correct or incorrect ratios. We observe that in the Covid_COPD project we annotated 5 different concepts, and 12 word forms for those concepts. This converges to 100% correctness after 60 documents even with further forms added on two separate occasions. However, larger annotation projects such as Covid_CTPA_Reports saw over 400 word forms of 194 concepts where the model

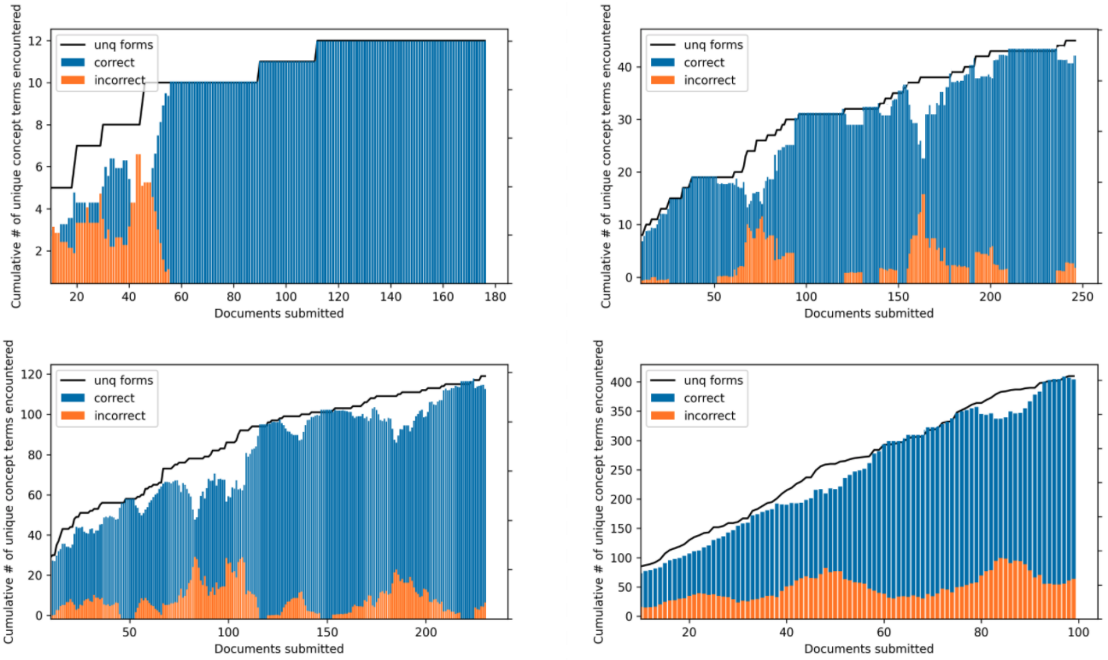


Fig. 3.1 Top left to bottom right: MedCATtrainer annotation projects number of concepts seen during human annotation vs number of configured concepts that could have appeared: Covid_COPD (5/2012), Covid_Gastro (8/679), Diabetes_Covid (15/864), Covid_CTPA_Reports (194/297280)

was still converging to an optimal model. A further observation is the ratio of correct to incorrect annotations that dip (orange area vs blue area under the black unique forms line) where MedCATtrainer is presented with a high volume of new word concept forms. This performance drop is quickly rectified (blue area increases) by subsequent training. Model performance should be understood by examining the progressive increase in the ratio of correct/incorrect annotations, rather than the absolute number of incorrect annotations. For example in Covid_CTPA_Reports, the incorrect ratio area (orange) looks largely flat, however, performance is slowly improving as correct annotation ratios per document are improving (blue area is larger than orange).

Subsequent error analysis found the causes of the performance drops shown i.e. the sharp rises in incorrect orange area vs blue correct area under the word forms line. In the Covid_Gastro project, our clinical annotators marked "ileal Crohn's" incorrect for Crohn's disease despite it being a sub-type of the more general Crohn's disease concept, and marked “Previous medical history: UC” as incorrect for ulcerative colitis. Both of these could arguably be marked correct. In our Diabetes_Covid project we see annotators that mark examples such as “episode in diabetic clinic” and “referred to medics by diabetic reg” where the condition is being used as an adjective so was likely marked incorrect as it is not directly describing a condition experienced by the patient. These are confusing for the MedCAT model, and should actually be marked as correct and left to a meta annotation to determine patient experience.

3.4.2 MedCAT Downstream Clinical Research

CogStack and the wider research group has utilised both the data availability and newly developed and trained MedCAT models and associated workflow provided by MedCAT-trainer to support clinical research before and during the Covid-19 pandemic. CogStack alerts provided early indications of upsurges ahead of lab results and trending terms could be identified such as ‘anosmia’ - now a common symptom alongside Covid-19.

As the pandemic was beginning to impact the King’s College Hospital emergency ward we were able to support important, time-critical research to answer priority questions such as whether angiotensin-converting enzyme inhibitors (ACE-i) were still safe and did not lead to increased Covid-19 severity risk [11]. ACE-Is are common drugs used within the treatment plan of common conditions such as high blood pressure, certain chronic kidney conditions, coronary artery disease and heart failure and so are frequently prescribed and taken by groups that are known to be particularly susceptible to severe Covid-19 cases. It was well understood that SARS-CoV and SARS-Cov-2 i.e. Covid-19, enter host cells via the ACE-2 receptor. Therefore, it was hypothesised that ACE-Is may lead to increased Covid-19 viral load therefore to greater severity risk. CogStack and MedCAT enabled rapid analysis of patient data incoming to King’s College Hospital. CogStack supported searching and data extraction before Covid-19 even had a consistent name within the EHR and before laboratory testing was consistently available. MedCAT allowed for rapid identification, extraction and contextualisation of comorbidity concepts allowing us to come to the conclusion there was no evidence of increased severity of Covid-19 outcomes for those on ACE-Is.

A further study used again data extracted from the KCH CogStack and MedCAT extracted comorbidities for the assessment of the early warning scoring system of NEWS - a system for risk stratification of Covid-19 patients recommended in the UK at the time [71]. We found that predictive risk could be improved by including readily available blood and physiological parameters such as supplemental oxygen flow rate, urea, age, oxygen saturation) and MedCAT extracted commodities such as hypertension, diabetes cardiovascular, respiratory and kidney disorders [25].

Another study characterised the various biological responses of admitted Covid-19 patients at KCH again extracting common relevant comorbidities. We found 5 distinct classes of patient, detailing specific biological responses at each class, the rate at which each were admitted to the ICU and the prevalence of comorbidities across each class. We concluded further research would be required for identifying potential early interventions for classes of patients to improve in-hospital outcomes [165].

A final study at South London and Maudsley NHS Foundation Trust (SLaM) used MedCAT extracted physical health conditions across a large cohort of patients (n=17,500) diagnosed with serious mental illness between 2007 and 2018. We extracted 21 common physical health conditions with F1 score at or above 0.9 for all conditions. This study found the 40% of the cohort had at least one physical health i.e. multi-morbidity, with 20% having complex multimorbidity, i.e. two or more physical health conditions alongside their SMI diagnosis [12].

3.5 Summarisation via MedCAT Models

Throughout this chapter I have shown our novel toolkit and associated workflow for the development, validation, fine-tuning and application of clinical NLP models. These models extract clinical terms from any configured terminology and provided free-text document. I have often found that the precise location of the clinical concept, i.e. the exact span of text that MedCAT has identified, linked and contexualised is not important for downstream use. Users of these models are typically only interested at a patient admission or the patient level entirely if a condition is chronic for example. I consider this usage of the model as a form of primitive summary often centered around summarising patient diagnosis, symptoms, findings, or medications.

The next chapter describes a downstream use case of such a summarisation system to improve current administrative processes that is currently performed manually.

Chapter 4

Existing Clinical Summarisation Tasks: Clinical Coding

EHR free-text's primary purpose is for direct patient care. A secondary purpose uses records for hospital administration and billing for remuneration of care provided. This involves extracting and summarising a patient *episode* where an episode is defined by one or more encounters with a service in a hospital care environment. For an inpatient multi-day stay this could involve multiple visits from various clinical teams: clinical specialists, surgery, nursing, radiology etc. A single episode can span an inpatient stay over multiple days generating many documents for each healthcare worker encounter or simply an outpatient could simply be one or two encounters and the associated documentation. There is potentially a huge variety in complexity from one episode to the next.

4.1 The Clinical Coding Process

In the UK entities from taxonomies such as ICD-10 and OPCS-4, as first discussed in Section 3.1, are assigned by *clinical coders* to patient episodes. This process effectively summarises the set of diagnoses and procedures / interventions for each episode. Clinical coding (CC) requires clinical knowledge of the words and phrases describing diagnoses and procedures, specific training of the *coding rules*, and of specific national and local priorities, but the role is not technically *clinical*. Meaning coders cannot make clinical inferences or judgements and assign a code for a diagnosis or intervention unless it is explicitly written within the notes and confirmed by clinical staff. For example, if a diagnosis is written ‘likely to be pulmonary edema’, or ‘possible pulmonary edema’, then this cannot be assigned a clinical code [157].

Specialist knowledge and growing demand for coding of administered care has resulted in a staff shortage [109] suggesting a need for improvements to the coding function from multiple perspectives [119]. One such perspective is the integration and usage of technology for computer assisted coding (CAC), and to move away from the fully manual, labour-intensive, error prone processes currently used. A recent literature review suggests CAC could improve data quality, streamline the process, and further develop the careers of clinical coders [23]. This review also suggested significant hurdles for CAC integration, including ongoing monitoring of the systems and retraining of existing staff.

Figure 4.1 shows the current coding process for a single admission to discharge, the aggregation of the care notes and coding, then the downstream uses of coded data for provider remuneration alongside collection for local and national clinical research and care planning databases. As suggested the coding function sits within its own function and does not impact patient care directly and similar to other data collected for local or national purposes the incentives for collection of high quality, clean and complete data are often not top-of-mind [109].

Clinical coding is therefore a task of document or multi-document level extraction of codes from clinical text. The NER+L problem and MedCAT methodology posed in Chapter 3 provides mention-level extraction of clinical codes from text, that is a code is specifically assigned to a contiguous block of text within specific document. To convert this approach to a clinical coding required output, mention-level codes can be aggregated and deduplicated. A specifically designed document or multi-document classification

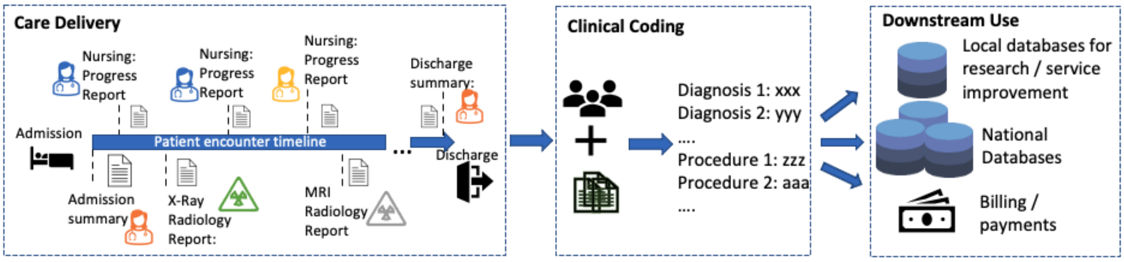


Fig. 4.1 A visual of the clinical coding process and downstream use of assigned codes applicable to large proportion of global healthcare.

system might be required where clinical codes are combinations of multiple spans of text within or across documents.

4.2 Clinical Coding and NLP Models

Clinical coding has recently attracted increased attention from NLP researchers. I believe this is motivated by: 1) the easy formulation of clinical coding as challenging multi-document, multi-label classification problem, 2) the wealth of 'labelled' data that is readily available through current clinical coding processes, allowing researchers to develop, test and benchmark models against consistent data, and 3) the problem is a real-world application of NLP that could assist a currently manual and labour-intensive process.

However, upon review of the literature the majority of the AI / NLP approaches proposed have consistent shortcomings [162, 10, 97, 24, 59]. Firstly, the majority of studies only report results on MIMIC-III. This is problematic as the dataset is only from a single US based site, only covers the ICU department between 2001-2012, and relies on the older ICD-9 taxonomy that has now been decommissioned in favour of version 10. There is a lack of empirical evidence to suggest these models will be successful across other sites, clinical specialties or geographical locations.

Secondly, diagnosis and procedure codes are often treated as group of equally plausible labels. This is contrary to what happens in reality. For example, a procedure or intervention code that is relevant for a neonate would only be applied to a patient that is neonatal. Most modern deep learning approaches rely on data intensive approaches to learn these ‘hard’ rules. Whereas, clinical coders can use both rule-based and ‘data‘ or experience driven approaches to assign codes.

Thirdly, prior NLP approaches treat each clinical code as equally important. In reality clinical coders are tasked with coding the primary diagnosis and primary intervention / procedure then secondary diagnoses and procedures / interventions are coded. The importance of some codes could be included within a loss function of a given model to weight towards certain classes, but this importance is actually often more dynamic in clinical coding practise. For long and complex episodes coding all of the secondary diagnoses is less important than coding secondary conditions in an inpatient day-case for example.

Finally, there is little to no consideration of how the NLP model will be deployed, maintained and integrated into a clinical coding workflow. I accept that a research paper cannot contain a complete plan for roll-out and maintenance, however, the NLP models discussed often present predictions without explicit reference to where in the text the predicted assignment came from. An important constraint in the coding process is that clinical coders cannot infer the diagnosis or procedure, it must be within the text.

A recent comprehensive literature review of automated coding systems and CACs shows the breadth and depth of work carried out within the clinical NLP community [60]. The authors conclude with similar thoughts as I have listed above.

4.3 Exploring the Suitability of MIMIC-III for Clinical Coding

The next published work investigated focused on the usage of MIMIC-III as a data source for training and evaluating CAC systems. I critically analyse the ICD-9 dataset arguing the lack of a ‘gold-standard’, double annotated and agreed upon set of codes makes the dataset a ‘silver standard’ dataset with potential issues for generalisability particularly for those methods that only use this dataset. This work fits alongside an important recent observation, suggesting the focus of AI / NLP researchers is skewed towards model development rather than on high quality, well understood datasets [126]. This observation is especially important in healthcare as this is a high stakes environment where algorithmic predictions even if not directly impacting patient care, such as clinical coding, can have downstream impact through the planning or reprioritization of care at local and national levels.

Experimental Evaluation and Development of a Silver-Standard for the MIMIC-III Clinical Coding Dataset

Thomas Searle[1], **Zina Ibrahim**[1], **Richard JB Dobson**[1,2]
[1]Department of Biostatistics and Health Informatics,
Institute of Psychiatry, Psychology and Neuroscience,
King's College London, London, U.K.
[2]Institute of Health Informatics, University College London,
London, London, U.K.
{firstname.lastname}@kcl.ac.uk

Abstract

Clinical coding is currently a labour-intensive, error-prone, but critical administrative process whereby hospital patient episodes are manually assigned codes by qualified staff from large, standardised taxonomic hierarchies of codes. Automating clinical coding has a long history in NLP research and has recently seen novel developments setting new state of the art results. A popular dataset used in this task is MIMIC-III, a large intensive care database that includes clinical free text notes and associated codes. We argue for the reconsideration of the validity MIMIC-III's assigned codes that are often treated as gold-standard, especially when MIMIC-III has not undergone secondary validation. This work presents an open-source, reproducible experimental methodology for assessing the validity of codes derived from EHR discharge summaries. We exemplify the methodology with MIMIC-III discharge summaries and show the most frequently assigned codes in MIMIC-III are under-coded up to 35%.

1 Introduction

Clinical coding is the process of translating statements written by clinicians in natural language to describe a patient's complaint, problem, diagnosis and treatment, into an internationally-recognised coded format (World Health Organisation, 2011). Coding is an integral component of healthcare and provides standardised means for reimbursement, care administration, and for enabling epidemiological studies using electronic health record (EHR) data (Henderson et al., 2006).

Manual clinical coding is a complex, labour-intensive, and specialised process. It is also error-prone due to the subtleties and ambiguities common in clinical text and often strict timelines imposed on coding encounters. The annual cost of clinical coding is estimated to be $25 billion in the US alone (Farkas and Szarvas, 2008).

To alleviate the burden of the status quo of manual coding, several Machine learning (ML) automated coding models have been developed (Larkey and Croft, 1996; Aronson et al., 2007; Farkas and Szarvas, 2008; Perotte et al., 2014; Ayyar et al., 2016; Baumel et al., 2018; Mullenbach et al., 2018; Falis et al., 2019). However, despite continued interest, translation of ML systems into real-world deployments has been limited. An important factor contributing to the limited translation is the fluctuating quality of the manually-coded real hospital data used to train and evaluate such systems, where large margins of error are a direct consequence of the difficulty and error-prone nature of manual coding. To our knowledge, the literature contains only two systematic evaluations of the quality of clinically-coded data, both based on UK trusts and showing accuracy to range between 50 to 98% Burns et al. (2012) and error rates between 1%-45.8% CHKS Ltd (2014) respectively. In Burns et al. (2012), the actual accuracy is likely to be lower because the reviewed trusts used varying statistical evaluation methods, validation sources (clinical text vs clinical registries), sampling modes for accuracy estimation (random vs non-random), and the quality of validators (qualified clinical coders vs lay people). CHKS Ltd (2014) highlight that 48% of the reviewed trusts used discharge summaries alone or as the primary source for coding an encounter, to minimise the amount of raw text used for code assignment. However, further portions of the documented encounter are often needed to assign codes accurately.

The Medical Information Mart for Intensive Care (MIMIC-III) database (Johnson et al., 2016) is the largest free resource of hospital data and constitutes a substantial portion of the training of automated coding models. Nevertheless, MIMIC-III is significantly under-coded for specific conditions (Kokotailo and Hill, 2005), and has been shown to exhibit reproducibility issues in the problem of

Proceedings of the BioNLP 2020 workshop, pages 76–85
Online, July 9, 2020 ©2020 Association for Computational Linguistics

mortality prediction (Johnson et al., 2017). Therefore, serious consideration is needed when using MIMIC-III to train automated coding solutions.

In this work, we seek to understand the limitations of using MIMIC-III to train automated coding systems. To our knowledge, no work has attempted to validate the MIMIC-III clinical coding dataset for all admissions and codes, due to the time-consuming and costly nature of the endeavour. To illustrate the burden, having two clinical coders, working 38 hours a week re-coding all 52,726 admission notes at a rate of 5 minutes and $3 per document, would amount to ∼$316,000 and ∼115 weeks work for a 'gold standard' dataset. Even then, documents with a low inter-annotator agreement would undergo a final coding round by a third coder, further raising the approximate cost to ∼$316,000 and stretching the 70 weeks.

In this work, we present an experimental evaluation of coding coverage in the MIMIC-III discharge summaries. The evaluation uses text extraction rules and a validated biomedical named entity recognition and linking (NER+L) tool, MedCAT (Kraljevic et al., 2019) to extract ICD-9 codes, reconciling them with those already assigned in MIMIC-III. The training and experimental setup yield a reproducible open-source procedure for building silver-standard coding datasets from clinical notes. Using the approach, we produce a silver-standard dataset for ICD-9 coding based on MIMIC-III discharge summaries.

This paper is structured as follows: Section 2 reviews essential background and related work in automated clinical coding, with a particular focus on MIMIC-III. Section 3 presents our experimental setup and the semi-supervised development of a silver standard dataset of clinical codes derived from unstructured EHR data. The results are presented in Section 4, while Section 5 discusses the wider impact of the results and future work.

2 Background

2.1 Clinical Coding Overview

The International Statistical Classification of Diseases and Health Related Problems (ICD) provides a hierarchical taxonomic structure of clinical terminology to classify morbidity data (World Health Organisation, 2011). The framework provides consistent definitions across global health care services to describe adverse health events including illness, injury and disability. Broadly, patient encounters with health services result in a set of clinical codes that directly correlate to the care provided.

Top-level ICD codes represent the highest level of the hierarchy, with ICD-9/10 (ICD-10 being the later version) listing 19 and 21 chapters respectively. Clinically meaningful hierarchical subdivisions of each chapter provide further specialisation of a given condition.

Coding clinical text results in the assignment of a single primary diagnosis and further secondary diagnosis codes (World Health Organisation, 2011). The complexity of coding encounters largely stems from the substantial number of available codes. For example, ICD-10-CM is the US-specific extension to the standard ICD-10 and includes 72,000 codes. Although a significant portion of the hierarchy corresponds to rare conditions, 'common' conditions to code are still in the order of thousands.

Moreover, clinical text often contains specialist terminology, spelling mistakes, implicit mentions, abbreviations and bespoke grammatical rules. However, even qualified clinical coders are not permitted to infer codes that are not explicitly mentioned within the text. For example, a diagnostic test result that indicates a condition (with the condition not explicitly written), or a diagnosis that is written as 'questioned' or 'possible' cannot be coded.

Another factor contributing to the laborious nature of coding is the large amount of duplication present in EHRs, as a result of features such as copy & paste being made available to clinical staff. It has been reported that 20-78% of clinicians duplicate sections of records between notes (Bowman, 2013), subsequently producing an average data redundancy of 75% (Zhang et al., 2017).

2.2 MIMIC-III - a Clinical Coding Database

MIMIC-III (Johnson et al., 2016) is a de-identified database containing data from the intensive care unit of the Beth Israel Medical Deaconess Center, Boston, Massachusetts, USA, collected 2001-12. MIMIC-III is the world's largest resource of freely-accessible hospital data and contains demographics, laboratory test results, procedures, medications, caregiver notes, imaging reports, admission and discharge summaries, as well as mortality (both in and out of the hospital) data of 52,726 critical care patients. MIMIC provides an open-source platform for researchers to work on real patient data. At the time of writing, MIMIC-III has over 900 citations.

2.3 Automated Clinical Coding

Early ML work on automated clinical coding considered ensembles of simple text classifiers to predict codes from discharge summaries (Larkey and Croft, 1996). Rule-based models have also been formulated, by directly replicating coding manuals. A prominent example of rule-based models is the BioNLP 2007 shared task (Aronson et al., 2007), which supplied a gold standard labelled dataset of radiology reports. The dataset continues to be used to train and validate ML coding. For example, Kavuluru et al. (2015) used the dataset in addition to two US-based hospital EHRs. Although the two additional datasets used by Kavuluru et al. (2015) were not validated to a gold standard, they are reflective of the diversity found in clinical text. Their largest dataset contained 71,463 records, 60,238 distinct code combinations and had an average document length of 5303 words.

The majority of automated coding systems are trained and tested Using MIMIC-III. Perotte et al. (2014) trained hierarchical support vector machine models on the MIMIC-II EHR (Saeed et al., 2011), the earlier version of MIMIC. The models were trained using the full ICD-9-CM terminology, creating baseline results for subsequent models of 0.395 F1-micro score. Ayyar et al. (2016) used a long-short-term-memory (LSTM) neural network to predict ICD-9 codes in MIMIC-III. However, Ayyar et al. (2016) cannot be directly compared to former methods as the model only predicts the top nineteen level codes.

Methodological developments continued to use MIMIC-III with Tree-of-sequence LSTMs (Xie and Xing, 2018), hierarchical attention gated recurrent unit (HA-GRU) neural networks (Baumel et al., 2018) and convolutional neural networks with attention (CAML) (Mullenbach et al., 2018). The HA-GRU and CAML models were directly compared with (Perotte et al., 2014), achieving 0.405 and 0.539 F1-micro respectively. A recent empirical evaluation of ICD-9 coding methods predicted the top fifty ICD-9 codes from MIMIC-III, suggesting condensed memory networks as a superior network topology (Huang et al., 2018).

3 Semi-Supervised Extraction of Clinical Codes

In this section, we describe the data preprocessing, methodology and experimental design for evaluating the coding quality of MIMIC-III discharge summaries. We also describe the semi-supervised creation of a silver-standard dataset of clinical codes from unstructured EHR text based on MIMIC-III discharge summaries.

3.1 Data Preparation

Discharge summary reports are used to provide an overview for the given hospital episode. Automated coding systems often only use discharge reports as they contain the salient diagnostic text (Perotte et al., 2014; Baumel et al., 2018; Mullenbach et al., 2018) without over burdening the model. MIMIC-III discharge summaries are categorised distinctly from other clinical text. The text is often structured with section headings and content section delimiters such as line breaks. We identify Discharge Diagnosis (DD) sections in the majority of discharge summary reports 92% (n=48,898) using a simple rule based approach. These sections are lists of diagnoses assigned to the patient during admission. Xie and Xing (2018) previously used these sections to develop a matching algorithm from discharge diagnosis to ICD code descriptions with moderate success demonstrating state-of-the-art sensitivity (0.29) and specificity (0.33) scores. For the 8% (n=3,828) that are missing these sections we manually inspect a handful of examples and observe instances of patient death and administration errors. The SQL procedures used to extract the raw data from a locally built replica of the MIMIC-III database and the extraction logic for DDs are available open-source as part of this wider analysis[1].

Table 1 lists example extracted DDs. There is a large variation in structure, use of abbreviations and extensive use of clinical terms. Some DDs list the primary diagnosis alongside secondary diagnosis, whereas others simply list a set of conditions.

3.2 Semi-Supervised Named Entity Recognition and Linkage Tool

We use MedCAT (Kraljevic et al., 2019), a pre-trained named entity recognition and linking (NER+L) model, to identify and extract the corresponding ICD codes in a discharge summary note. MedCAT utilises a fast dictionary-based algorithm for direct text matches and a shallow neural network concept to learn fixed length distributed semantic vectors for ambiguous text spans. The method is conceptually similar to Word2Vec

[1] https://tinyurl.com/t7dxn3j

Extracted Discharge Diagnosis	Admission ID
CAD now s/p CABG HTN, DM, Osteoarthritis, Dyslipidemia	102894
Left convexity, tentorial, parafalcine Subdural hematoma	161919
Primary Diagnoses: 1. Acute ST segment Elevation Myocardial Infarction Secondary Diagnoses: 1. Hypertension 2. Hyperlipidemia	152382
Seizures.	132065

Table 1: Example discharge diagnosis subsections extracted from MIMIC-III discharge summaries

(Mikolov et al., 2013) in that word representations are learnt by detecting correlations of context words, and learnt vectors exhibit the semantics of the underlying words. The tool can be trained in a unsupervised or a supervised manner. However, unlike Word2Vec that learns a single representation for each word, MedCAT enables the learning of 'concept' representations by accommodating synonymous terms, abbreviations or alternative spellings.

We use a MedCAT model pre-loaded with the Unified Medical Language System (Bodenreider, 2004) (UMLS). UMLS is a meta-thesaurus of medical ontologies that provides rich synonym lists that can be used for recognition and disambiguation of concepts. Mappings from UMLS to the ICD-9[2] taxonomy are then used to extract UMLS concept to ICD codes. Our large pre-trained MedCAT UMLS model contains ∼1.6 million concepts. This model cannot be made publicly available due to constraints on the UMLS license, but can be trained in an an unsupervised method in ∼1 week on MIMIC-III with standard CPU only hardware[3].

In an effort to keep our analysis tractable we limit our MedCAT model to only extract the 400 ICD-9 codes that occur most frequently in the dataset. This equates to 76% (n=48,2379) of total assigned codes (n=634,709). We exclude the other 6,441 codes that occur less frequently. Future work could consider including more of these codes.

3.3 Code Prediction Datasets

We run our MedCAT model over each extracted DD subsection. The model assigns each token or sequence of tokens a UMLS code and therefore an associated ICD code. In our comparison of the MedCAT produced annotations with the MIMIC-III assigned codes we have 3 distinct datasets:

1. MedCAT does not identify a concept and the code has been assigned in MIMIC-III. Denoted **A_NP** for 'Assigned, Not Predicted'.

2. MedCAT identifies a concept and this matches with an assigned code in MIMIC-III. Denoted **P_A** for 'Predicted, Assigned'.

3. MedCAT identifies a concept and this does not match with an assigned code in MIMIC-III dataset. Denoted **P_NA** for 'Predicted, Not Assigned'.

We do not consider the case where both MedCAT and the existing MIMIC-III assigned codes have missed an assignable code as this would involve manual validation of all notes, and as previously discussed, is infeasible for a dataset of this size.

3.3.1 Producing the Silver Standard

Given the above initial datasets we produce our final silver-standard clinical coding dataset by:

1. Sampling from the missing predictions dataset (A_NP) to manually collect annotations where our out-of-the-box MedCAT model fails to recognise diagnoses.

2. Fine-tuning our MedCAT model with the collected annotations and re-running on the entire DD subsection dataset producing updated A_NP, P_A, P_NA datasets.

3. Sampling from P_NA and P_A and annotating predicted diagnoses to validate correctness of the MedCAT predicted codes.

4. Exclusion of any codes that fail manual validation step as they are not trustworthy predictions made by MedCAT.

We use the MedCATTrainer annotator (Searle et al., 2019) to both collect annotations (stage 1) and to validate predictions from MedCAT (stage 3). To collect annotations, we manually inspect 10 randomly sampled predictions for each of the 400 unique codes from A_NP and add further acronyms,

[2] https://bioportal.bioontology.org/ontologies/ICD9CM
[3] https://tinyurl.com/yadtnz3w

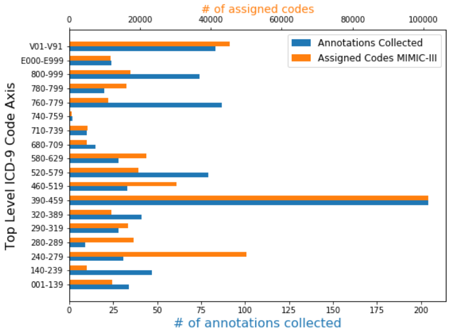


Figure 1: The distributions of manually annotated ICD-9 codes and the assigned codes in MIMIC-III grouped by top-level ICD-9 axis.

abbreviations, synonyms etc for diagnoses if they are present in the DD subsection to improve the underlying MedCAT model. To validate predictions from P_A and P_NA, we use the MedCAT-Trainer annotator to inspect 10 randomly sampled predictions for each of the 179 & 182 unique codes respectively found. We mark each prediction made by MedCAT as correct or incorrect and report results in Section 4.1.

4 Results

The following section presents the distribution of manually collected annotations from sampling A_NP, our validation of updated P_A and P_NA post MedCAT fine-tuning, and the final distribution of codes found in our produced silver standard dataset.

Adding annotations to selected text spans directly adds the spans to the MedCAT dictionary, thereby ensuring further text spans of the same content are annotated by the model - if the text span is unique. We collect 864 annotations after reviewing 4000 randomly sampled DD notes from the A_NP (Assigned, Not Predicted) dataset. 21.6% of DDs provide further annotations suggesting that the majority of missed codes lie outside the DD subsection, or are incorrectly assigned.

Figure 1 shows the distributions of manually collected code annotations and the current MIMIC-III set of clinical codes, grouped by their top-level axis as specified by ICD-9-CM hierarchy.

We collect proportionally consistent annotations for most groups, including the 390-459 chapter (Diseases Of The Circulatory System), which is the top occurring group in both scales. However, for groups such as 240-279 (endocrine, nutritional and metabolic diseases) and 460-519 (diseases of the respiratory system) we see proportionally fewer manually collected examples despite the high number of occurrence of codes assigned within MIMIC-III. We explain this by the DD subsection lacking appropriate detail to assign the specific code. For example codes under 250.* for diabetes mellitus and the various forms of complications are assigned frequently but often lack the appropriate level of detail specifying the type, control status and the manifestation of complication.

Using the manual amendments made on the 864 new annotations, we re-run the MedCAT model on the entire DD subsection dataset, producing updated P_NA, P_A and A_NP datasets. We acknowledge A_NP likely still includes cases of abbreviations, synonyms as we only subsampled 10 documents per code allowing for further improvements to the model.

The MedCAT fine-tuning process was run until convergence as measured by precision, recall and F1 achieving scores 0.90, 0.92 and 0.91 respectively on a held out a test-set with train/test splits 80/20. The fine-tuning code is made available[4]. Annotations are available upon request given the appropriate MIMIC-III licenses.

4.1 P_A & P_NA Validation

We use the MedCATTrainer interface to validate our MedCAT model predictions in the 'Predicted, Assigned' (P_A) and 'Predicted, Not Assigned' (P_NA) datasets. We sample (a maximum of) 10 unique predictions for each ICD-code resulting in 179 & 182 ICD-9 codes and 1588 & 1580 manually validated predictions from P_A and P_NA respectively. The validation of code assignment is performed by a medical informatics PhD student with no professional clinical coding experience and a qualified clinical coder, marking each term as correct or incorrect. We achieve good agreement with a Cohen's Kappa of 0.85 and 0.8 resulting in 95.51% and 87.91% marked correct for P_A and P_NA respectively. We exclude from further experiments all codes that fail this validation step as they are not trustworthy predictions made by MedCAT.

4.2 Aggregate Assigned Codes & Codes Silver Standard

We proportionally predict ∼10% (n=42,837) of total assigned codes (n=432,770). We predict ∼16%

[4] https://github.com/tomolopolis/MIMIC-III-Discharge-Diagnosis-Analysis/blob/master/Run_MedCAT.ipynb

of total assigned codes (n=258,953) if we only consider the 182 codes that resulted in at least one matched assignment to those present in the MIMIC-III assigned codes.

We label and gather our three datasets into a single table, with an extra column called 'validated', with values: 'yes' for codes that have matched with an assigned code (P_A), 'new_code' for newly discovered codes (P_NA), and 'no' for codes that we were not able to validate (A_NP). We have made this silver-standard dataset available alongside our analysis code[5].

4.3 Undercoding in MIMIC-III

This work aims to identify inconsistencies and variability in coding accuracy in the current MIMIC-III dataset. Ultimately to rigorously identify undercoding of clinical text full, double blind manual coding would be performed. However, as previously discussed, this is prohibitively expensive.

Comparing the codes predicted by MedCAT to the existing assigned codes enables the development of an understanding of specific groups of codes that exhibit possible undercoding. In this section we firstly show the effectiveness of our method in terms of DD subsection prediction coverage. We then present our predicted code distributions against the MIMIC-III assigned codes at the ICD code chapter level, highlighting the most prevalent missing codes and showing correlations between document length and prevalence.

4.3.1 Prediction Coverage

MedCAT provides predictions at the text span level, with only one concept prediction per span. We can therefore calculate the breadth of coverage of our predictions across all DD subsections. Figure 2 shows the proportion of DD subsection text that are included in code predictions. We note the 100% proportion (n=2105) is 75% larger than the next largest indicating that we are often utilising the entire DD subsection to suggest codes although the majority of the coverage distribution is around the 40-50% range.

We find a token length distribution of DD subsections with μ =14.54, σ =15.9, $Med = 10$ and $IQR = 14$ and a code extraction distribution with $\mu = 3.6$ and $\sigma = 3.1$, $Med = 3$ and $IQR = 4$ suggesting the DD subsections are complex and often list multiple conditions of which we identify, on average, 3 to 4 conditions.

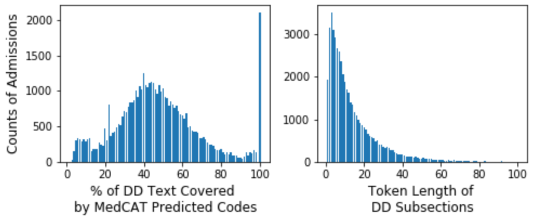


Figure 2: Left: Counts of admissions and the associated % of characters covered by MedCAT code predictions. Right:Distribution of DD token lengths

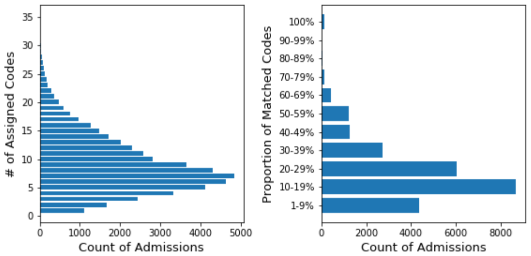


Figure 3: Proportions of matching predictions against total number of assigned codes per admission.

4.3.2 Predicted & Assigned

Figure 3 shows the distributions of the number of assigned codes and the proportion of matches grouped into buckets of 10% intervals. We see a high proportion of matches in assigned codes in the 1-40% range, indicating that although the DD subsection does contribute to the assigned ICD codes, many of the assigned codes are still missed. We exclude the admissions that had 0 matched codes and discuss this result further in Section 4.3.4.

If we order codes by the number of predicted and assigned we find the three highest occurring codes (4019, 41404, 4280) in MIMIC-III also rank highest in our predictions. However, we note that these three common codes only yield 25-39% of their total assigned occurrence, which could be explained by these chronic conditions not being listed in the DD subsection and referred elsewhere in the note. If we normalise predictions by their prevalence, we are most successful in matching specific conditions applicable to preterm newborns (7470, 7766), pulmonary embolism (41519) and liver cancer (1550), all of which we match between 69-55% but rank 114-305 in total prevalence. We suggest these diagnoses are either acute, or the primary cause of an ICU admission so will be specified in the DD subsection.

[5] https://tinyurl.com/u8yae8n

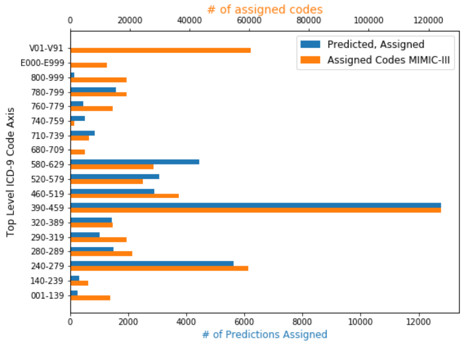


Figure 4: **Predicted, Assigned Codes** grouped by top-level code group vs total assigned codes

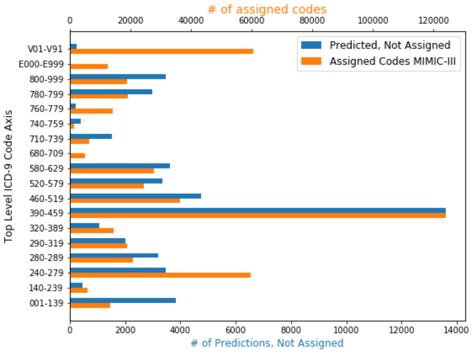


Figure 5: **Predicted, Not Assigned Codes** grouped by top-level code group vs total assigned codes

We also group the predicted codes into their respective top-level ICD-9 groups in Figure 4 and observe that predicted assigned codes display a similar distribution to total assigned codes. We quantify the difference in distributions via the Wasserstein metric or 'Earth Movers Distance'(Ramdas et al., 2015). This metric provides a single measure to compare the difference in our 3 datasets distributions when compared with the current assigned code distribution. We compute a small 2.7×10^{-3} distance between both distributions, suggesting our method proportionally identifies previously assigned codes from the DD subsection alone.

4.3.3 Predicted & Not Assigned

This dataset highlights codes that may have been missed from the current assigned codes.

Figure 5 shows that the distribution of predicted but not assigned codes is minimally different for most codes, supporting our belief that the MIMIC-III assigned codes are not wholly untrustworthy, but are likely under-coded in specific areas.

From this dataset we calculate how many examples of each code that has potentially been missed, or potentially under-coded. For the 10 most frequently assigned codes we see 0-35% missing occurrences. We also identify the most frequent code 4019 (Unspecified Essential Hypertension) has 16% or 3312 potentially missing occurrences.

To understand if DD subsection length impacts the occurrence of 'missed' codes we first calculate a Pearson-Correlation coefficient of 0.17 for DD subsection line length and counts of assigned codes over all admissions. This suggests a weak positive correlation between admission complexity and number of existing assigned codes.

In contrast we find a stronger positive correlation of 0.504 for predicted and not assigned codes and DD subsection line length. This implies that where an episode has a greater number of diagnoses or the complexity of an admission is greater, there is a likelihood to result in more codes being missed during routine collection.

We compute the Wasserstein metric between these two distributions at 1.6×10^{-2}. This demonstrates a degree of similarity between distributions albeit is 8x further from the Predicted and Assigned dataset distance presented in Section 4.3.2. We expect to see a larger distance here as we are detecting codes that are indicated in the text but have been missed during routine code assignment.

4.3.4 Assigned & Not Predicted

We observe that the distribution of assigned and not predicted codes largely mirrors the distribution of total codes assigned in MIMIC-III with a Wasserstein distance of 2.7×10^{-3} that is similar to the distance observed in in our Predicted and Assigned Section 4.3.2) dataset. This suggests that our method is proportionally consistent at not annotating codes that have likely been assigned from elsewhere in the admission, but may also be incorrectly assigned.

5 Discussion

On aggregate, the predicted codes by our MedCAT model suggest that the discharge diagnosis sections listed in 92% of all discharge notifications are not sufficient for full coding of an episode. Unsurprisingly, this confirms that clinicians infrequently document all codeable diagnoses within the discharge summary. Although, as previously stated, coders are not permitted to make clinical inferences. Therefore, to correctly assign a code, the diagnoses must be present within the documented patient episode within the structured or unstructured data.

However, the positive correlation between document length and number of predicted codes indicates that missed codes are more prevalent in highly complex cases with many diagnoses. From a coding workflow perspective, coders operate under strict time schedules and are required to code a minimum number of episodes each day. Therefore, it logically follows that the complexity of a case directly correlates to the number of codes missed during routine collection.

Looking at individual code groups we find 240-279 is not predicted proportionally with assigned codes both in P_A and P_NA. We explain this as follows. Firstly, DD subsections generally convey clinically important diagnoses for follow-up care. Certain codes such as (250.*) describe diabetes mellitus with specific complications, but the DD subsection will often only describe the diagnoses 'DMI' or 'DMII'. Secondly, ICU admissions are for individuals with severe illness and therefore are likely to have a high degree of co-morbidity. This is implied by the majority of patients (74%) are assigned between 4 and 16 codes.

We also observe E000-E999 and V01-V99 codes are disproportionately not predicted. However, this is expected given that both groups are supplementary codes that describe conditions or factors that contribute to an admission but would likely not be relevant for the DD subsection.

In contrast, we observe a disproportionately large number of predictions for 001-139 (Infectious and Parasitic Diseases). This is primarily driven by 0389 (Unspecified septicemia). A proportion of these predictions may be in error as the specific form of septicemia is likely described in more detail elsewhere in the note and therefore coded as the more specific form.

5.1 Method Reproducibility & Wider Utility

Inline with the suggestions of Johnson et al. (2017), the original authors of MIMIC-III, we have attempted to provide the research community all available materials to reproduce and build upon our experiments and method for the development of silver standard datasets. Specifically, we have made the following available as open-source: the SQL scripts to extract the raw data from a replica of the MIMIC-III database, the script required to parse DD subsections, an example script to build a pre-trained MedCAT model, the script required to run MedCAT on the DD subsections, load into the annotator and finally re-run MedCAT and perform experimental analysis alongside outputting the silver standard dataset[6].

Given these materials it is possible for researchers to replicate and build upon our method, or directly use the silver standard dataset in future work that investigates automated clinical coding using MIMIC-III. The silver standard dataset clearly marks if each assigned code has been validated or not, or if it is a new code according to our method.

6 Conclusions & Future Work

This work highlighted specific problems with using MIMIC-III as a dataset for training and testing an automated clinical coding system that would limit model performance within a real deployment.

We identified and deterministically extracted the discharge diagnosis (DD) subsections from discharge summaries. We subsequently trained an NER+L model (MedCAT) to extract ICD-9 codes from the DD subsections, comparing the results across the full set of assigned codes. We find our method covers 47% of all tokens, considering we only take 400 of the ∼7k unique codes and perform minimal data cleaning of the DD subsection. We have shown in Section 4.3.2 and 4.3.3 that the MedCAT predicted codes are proportionally inline with assigned codes in MIMIC-III.

Interestingly, we found a 0.504 positive correlation between DD length and the number of codes predicted by MedCAT, but not assigned in MIMIC-III. This result can be understood by observing that the ICU admissions in MIMIC-III can be extremely complex, with up to 30 clinical codes assigned to a single episode. The DD subsections alone can contain up to 50 line items indicating highly complex cases where codes could easily be missed.

We found that the code group 390-459 (Diseases of the Circulatory System) is both the most assigned group and the group of codes where there are the most missing predictions from our model. Furthermore, codes such as Hypertension (4019), Sepsis and Septicemia (0389, 99591), Gastrointestinal hemorrhage (95789), Chronic Kidney disease (5859), anemia (2859) and Chronic obstructive asthma (49320) are all frequently assigned but

[6] https://github.com/tomolopolis/MIMIC-III-Discharge-Diagnosis-Analysis

are also the highest occurring conditions that appear in the DD diagnosis subsection but are not assigned in the MIMIC-III dataset. This suggests that MIMIC-III exhibits specific cases of undercoding, especially with codes that are frequently occurring in patients but are not likely to be the primary diagnosis for an admission to the ICU.

As we only use the DD section, there are many codes which likely appear elsewhere in the note that we cannot assign. Although 92% of discharge summaries contain DD subsections we only match $\sim 16\%$ of assigned codes. We suggest this is due to: our NER+L model lacking the ability to identify more synonyms and abbreviations for conditions, the DD subsections lacking enough detail to assign codes and in some occasions, little evidence to suggest a code assignment. Our textual span coverage, presented in Section 4.3.1 demonstrates that we often cover all available discharge diagnosis, although there is still room for improvement as the majority of the coverage distribution is around the 50% mark.

For future work we foresee applying the same method to either the entire discharge summary or more specific sections such as 'previous medical history' to surface chronic codeable diagnoses that could be validated against the current assigned code set. Researchers would however likely need to address false positive code predictions as clinical coding requires assigned codes to be from current conditions associated with an admission.

In conclusion, this work has found that frequently assigned codes in MIMIC-III display signs of undercoding up to 35% for some codes. With this finding we urge researchers to continue to develop automated clinical coding systems using MIMIC-III, but to also consider using our silver standard dataset or build on our method to further improve the dataset.

Acknowledgments

RD's work is supported by 1.National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London. 2. Health Data Research UK, which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, Economic and Social Research Council, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), British Heart Foundation and Wellcome Trust. 3. The National Institute for Health Research University College London Hospitals Biomedical Research Centre. This paper represents independent research part funded by the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London. The views expressed are those of the author(s) and not necessarily those of the NHS, MRC, NIHR or the Department of Health and Social Care.

References

Alan R Aronson, Olivier Bodenreider, Dina Demner-Fushman, Kin Wah Fung, Vivian K Lee, James G Mork, Aurélie Névéol, Lee Peters, and Willie J Rogers. 2007. From indexing the biomedical literature to coding clinical text: Experience with MTI and machine learning approaches. In Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, BioNLP '07, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sandeep Ayyar, O B Don, and W Iv. 2016. Tagging patient notes with icd-9 codes. In Proceedings of the 29th Conference on Neural Information Processing Systems.

Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Nóemie Elhadad. 2018. Multi-Label classification of patient notes: Case study on ICD code assignment. In Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res., 32(Database issue):D267–70.

Sue Bowman. 2013. Impact of electronic health record systems on information integrity: quality and safety implications. Perspect. Health Inf. Manag., 10:1c.

E M Burns, E Rigby, R Mamidanna, A Bottle, P Aylin, P Ziprin, and O D Faiz. 2012. Systematic review of discharge coding accuracy.

CHKS Ltd. 2014. CHKS - insight for better healthcare: The quality of clinical coding in the NHS. `https://bit.ly/34eU5g3`. Accessed: 2019-5-10.

Matus Falis, Maciej Pajak, Aneta Lisowska, Patrick Schrempf, Lucas Deckers, Shadia Mikhael, Sotirios Tsaftaris, and Alison O'Neil. 2019. Ontological attention ensembles for capturing semantic concepts in ICD code prediction from clinical text.

Richárd Farkas and György Szarvas. 2008. Automatic construction of rule-based ICD-9-CM coding systems. BMC Bioinformatics, 9 Suppl 3:S10.

Toni Henderson, Jennie Shepheard, and Vijaya Sundararajan. 2006. Quality of diagnosis and procedure coding in ICD-10 administrative data. Med. Care, 44(11):1011–1019.

Jinmiao Huang, Cesar Osorio, and Luke Wicent Sy. 2018. An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes.

Alistair E W Johnson, Tom J Pollard, and Roger G Mark. 2017. Reproducibility in critical care: a mortality prediction case study. In Proceedings of the 2nd Machine Learning for Healthcare Conference, volume 68 of Proceedings of Machine Learning Research, pages 361–376, Boston, Massachusetts. PMLR.

Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. Sci Data, 3:160035.

Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. Artif. Intell. Med., 65(2):155–166.

Rae A Kokotailo and Michael D Hill. 2005. Coding of stroke and stroke risk factors using international classification of diseases, revisions 9 and 10. Stroke, 36(8):1776–1781.

Zeljko Kraljevic, Daniel Bean, Aurelie Mascio, Lukasz Roguski, Amos Folarin, Angus Roberts, Rebecca Bendayan, and Richard Dobson. 2019. MedCAT – medical concept annotation tool.

Leah S Larkey and W Bruce Croft. 1996. Combining classifiers in text categorization. In SIGIR, volume 96, pages 289–297.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C J C Burges, L Bottou, M Welling, Z Ghahramani, and K Q Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. J. Am. Med. Inform. Assoc., 21(2):231–237.

Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. 2015. On wasserstein two sample testing and related families of nonparametric tests.

Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. 2011. Multiparameter intelligent monitoring in intensive care II: a public-access intensive care unit database. Crit. Care Med., 39(5):952–960.

Thomas Searle, Zeljko Kraljevic, Rebecca Bendayan, Daniel Bean, and Richard Dobson. 2019. MedCATTrainer: A biomedical free text annotation interface with active learning and research use case specific customisation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pages 139–144, Stroudsburg, PA, USA. Association for Computational Linguistics.

World Health Organisation. 2011. International Statistical Classification of Diseases and Related Health Problems.

Pengtao Xie and Eric Xing. 2018. A neural architecture for automated ICD coding. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1066–1076.

Rui Zhang, Serguei V S Pakhomov, Elliot G Arsoniadis, Janet T Lee, Yan Wang, and Genevieve B Melton. 2017. Detecting clinically relevant new information in clinical notes across specialties and settings. BMC Med. Inform. Decis. Mak., 17(Suppl 2):68.

4.3.1 Discussion

This paper is limited by only applying the MedCAT model to the discharge diagnosis (DD) section of the discharge summary note. Early feasibility work suggested using MedCAT on the entirety of the note would result in many more suggested codes, many of which would be ‘noise’, that should not be counted as an assignable clinical code as they are not current / or chronic conditions and should not be coded according to clinical coding rules.

We acknowledge that the DD section can often be a small portion of the the discharge summary and overall admission notes, and is insufficient to fully code an admission. The DD section is a good place to start with automated analysis as this is one of the first sections a human clinical coder will check for diagnosis as it should be a summarised and prioritised list of all presenting diagnoses. Analysing the DD section alone is useful in demonstrating where if a code is not assigned to the admission, but is clearly written in DD section, this is strong evidence of a missed code.

For the other groups of predicted assigned / and assigned but not predicted codes - these two groups demonstrate that the DD is only sometimes sufficient for complete and full coding. More analysis could be done to investigate if there are correlations between patient types, authors, admission complexity etc.

A full gold-standard coding of MIMIC-III would be unfeasibly expensive to complete. We attempt to compare relative distributions of codes across ICD-9 chapters, whilst also identifying the specific admissions that are missing these assigned codes, providing the ‘silver standard’ back to the community.

4.4 Computer Assisted Clinical Coding Conclusions

Tools such as MedCAT can support data quality goals and the often associated audit processes that follow not only in clinical coding but for clinical, operational or administrative data efforts. I foresee tools such as MedCAT holding potential for broadly improving accuracy e.g. suggesting where codes or clinical data can be automatically verified, or suggesting a potential false-positive to be flagged for review, completeness e.g. suggesting additional codes or clinical data that may have been missed, consistency e.g. ensuring chronic conditions that should consistently appear between care episodes.

A recent study for a stroke clinical audit has already demonstrated this potential. Researchers used a MedCAT fine-tuned model for common comorbidity concepts and compared the identification of these against current curation methods [141]. They observed improvements across all F1 performance scores across all extracted comorbidities when compared with existing curation methods, suggesting merit in the approach.

So far I have only discussed MedCAT, the toolkit in providing means to extract structured representations, i.e. SNOMED CT codes, from free-text narratives. I have also covered the meta-annotation approach that allows for contextualisation of these extracted terms, such as the diagnosis terms that are further classified to be relevant to the patient, or perhaps terms that a positive mentions and not negated. I have presented how this method has broad applications across clinical research and also for the downstream summarisation use case of clinical coding.

As discussed in Appendix Section A.1, free-text narratives are a rich efficient means to store and communicate nuanced and detailed information. This is in part why free-text inputs are often preferred by clinicians where there is no appropriate structured input field. The next chapter will revisit free-text generation to build further methods to understand the differences between open-domain and clinical text generation. I also present a final study on methods to summarise inpatient hospital stays to generate the ‘brief hospital course’ section of a discharge summary.

Chapter 5

Automated text summarisation of EHR text

Previous chapters have considered methods and their applications for structuring clinical free-text, a common task performed both during direct patient care and for secondary purposes such as care audit, clinical research or the administration of healthcare i.e. clinical coding. However, free-text in the clinical setting is often summarised and enriched during successive note entries. During a patient episode for direct patient care, often only the most recent note will be used. For example clinicians might only use the very latest ‘clinical progress note’ ignoring older notes of the same type. If a patients care transitions from one department i.e. emergency medicine to a ward, or is discharged from secondary to primary care, then receiving clinicians will likely read through the ‘discharge summary’ note. This summarises the entire admission outlining the presenting problem, associated investigations, interventions and finally future follow-up care to be provided elsewhere [49].

Overall, patient records evolve over time. They are updated periodically, at varying cadence according to speciality, context and clinical events. Different parts of a patient’s record are often more useful than others. For example, in the scenario of a patient being discharged from a secondary care setting to their primary care provider (PCP), the clinician (a GP in the UK) will likely only read the discharge report or discharge letter ignoring the progress, admission or other specialist reports. They would expect that the discharge summary has already summarised relevant clinical information from previous reports [58]. However, this may not always be the case as a discharge summary could be incomplete, or the patient condition particularly complex etc. [121] requiring usage of the other notes.

Recent systematic reviews of summarisation methods for EHR text [107, 94] found proposed systems often did not generate text narratives as a summary. The reviews note that systems often use a NER+L (e.g. MedCAT) approach of first extracting clinically relevant terms then use a secondary step to present these extracted terms in either a visualisation [51, 22] or to complete a predefined template [159, 2].

Summarisation models that output a text narrative as the summary, can either be *extractive* - a subset of words, phrases or sentences are selected and combined from the source text or *abstractive* - sentences are generated by sampling from a vocabulary via an autoregressive process. This is discussed more fully in Section 5.3, and have only recently been applied to clinical texts in limited scenarios.

Abstractive models are called sequence-to-sequence (seq-to-seq) models as they receive a sequence of source note texts and outputs sequences of output summary text. Seq-to-seq models have been being trained for radiology report summarisation of the *impression* section of a report, from the *background* and *findings* sections [167]. This is still early research, with a long road to be useful from a clinical perspective. System produced summaries are susceptible to factual errors [168], and *best* performing systems are still a long way off to being equivalent to a real radiologist performance [31].

5.1 Summarisation from an Information Theory Perspective

Information theory provides a theoretical basis to calculate the *information* contained within text S, via an optimal encoding scheme $H(S)$. Further formal definitions are provided in the following Section 5.2, but information theory underpins many areas in ML and NLP. For example, cross-entropy allows the calculation of a numerical loss given a loss function, such as those discussed in Section 2.1.1 and B.1.1, and single numerical value for the difference between distribution of predictions and the intended label or *reference* distribution.

Perplexity is another measure from information theory that is used in language modelling - the task to predict the next token given a sequence of tokens initially discussed in Section 2.5.2. Perplexity quantifies language model performance over a given corpus and experimental setup. Perplexity can also be interpreted as the average *surprise* or branching factor to encode this information.

Irrespective of the output summary format i.e. free-text or visualisation, the initial step in building a summary is the identification of the set of relevant entities, their interactions and their relative *importance* to one another. Recent work has presented a theoretical framework for *importance* in the context of summarisation [106]. This framework presents an information theoretic framework to assess importance through definitions of *redundancy*, *relevance* and *informativeness*.

For clinical free-text summarisation, *relevance* and *informativeness* require the summary users' context, e.g. the equivalent summary for a radiologist and a nurse will vary in relevance and informativeness. However, *redundancy* is irrespective of the user. The framework suggests an H_{max}, or a theoretical maximum entropy that could be achieved with a 'perfect' summary and entropy produced from the current summary $H(S)$. The redundancy of summary $Red(S)$ is formulated:

$$Red(S) = H_{max} - H(S) \tag{5.1}$$

Considering Equation 5.1, we do not directly have H_{max} or a means to collect an objective H_{max} for clinical texts. Prior work, reports that consistency in clinical text is difficult due to subjective data interpretation and user dependent preferences [56].

The amount of redundancy in a text is the volume of text that is repeating one or more of the semantic units in the text. Redundant text will not inform a reader anymore than they already know. These parts of the texts should be removed in summarisation generation, given that the aim of the summary is to reduce the total volume of text whilst maintaining the meaning.

5.2 Estimating Redundancy in Clinical Text

However, we can still compare entropy / perplexity across domains to understand relative redundancy. In the below paper I perform these experiments to quantify redundancy within clinical texts. I compare perplexity through the use a previously state-of-the-art language model over open-domain and clinical text corpora and find consistent differences in perplexity across two real world clinical data sets compared to open-domain corpora.

I also carry our further experiments comparing redundancy at the token level between successive clinical notes of the equivalent note types. This method quantifies redundancy within note sequences by assuming note sequences are continual summaries of prior notes.

Journal of Biomedical Informatics 124 (2021) 103938

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Estimating redundancy in clinical text

Thomas Searle [a,*], Zina Ibrahim [a], James Teo [b], Richard Dobson [a,c]

[a] Department of Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK
[b] King's College Hospital NHS Foundation Trust, London, UK
[c] Institute of Health Informatics, University College London, London, UK

ARTICLE INFO

Keywords:
Natural language processing methods to estimate redundancy of clinical text
Deep transfer learning for language modelling of clinical text

ABSTRACT

The current mode of use of Electronic Health Records (EHR) elicits text redundancy. Clinicians often populate new documents by duplicating existing notes, then updating accordingly. Data duplication can lead to propagation of errors, inconsistencies and misreporting of care. Therefore, measures to quantify information redundancy play an essential role in evaluating innovations that operate on clinical narratives.

This work is a quantitative examination of information redundancy in EHR notes. We present and evaluate two methods to measure redundancy: an information-theoretic approach and a lexicosyntactic and semantic model. Our first measure trains large Transformer-based language models using clinical text from a large openly available US-based ICU dataset and a large multi-site UK based Hospital. By comparing the information-theoretic *efficient encoding* of clinical text against open-domain corpora, we find that clinical text is ∼ 1.5× to ∼ 3× less efficient than open-domain corpora at conveying *information*. Our second measure, evaluates automated summarisation metrics Rouge and BERTScore to evaluate successive note pairs demonstrating lexicosyntactic and semantic redundancy, with averages from ∼43 to ∼65%.

1. Introduction

Electronic Health Record (EHR) text details patient history, findings, symptoms, diagnoses, procedures and plans for future care. A single inpatient hospital stay can result in multiple document types (e.g. GP letters, inpatient admission/ discharge notes) created by the different specialisms involved in the patient's care (e.g. nursing, A&E, cardiology, neurology, radiology etc.) as well as progress documents to address previous questions and introducing follow-up actions or queries. As a result, a patient's records can contain different perspectives accumulated through time, by various specialities documenting the patient's 'progress' throughout the care pathway [26]. Therefore, it naturally follows that EHR text and the design of systems induces redundancy. This is not necessarily a negative as repeated mentions could be used to indicate importance, corroboration or confirmation of a prior finding, diagnosis etc. However, using the clinical narratives for direct patient care can be difficult [20], as clinicians must navigate through potentially redundant, out-of-date or erroneous information to come to the *current* state of a patient, although this problem of navigation and data consumption is not exclusive to unstructured portion of EHRs. For secondary research purposes [3,31] this requires significant time cleaning and pre-processing data [30,21].

Using clinical narratives in EHRs is unavoidable. For direct patient care, forcing EHR users to specify patient state in only structured fields thereby avoiding free-text input is both impractical and insufficient [11,1] and also does not consider existing free-text patient data. Outside of direct patient care, prior work has shown EHR text analysis offers insights in diverse areas such as disease classification [37], trajectory modelling [36], patient stratification [21], therapeutic development [27] and personalised medicine [52]. Yet, the free-text content of EHRs forces researchers to spend considerable time manually exploring datasets attempting to identify the most *informative* portions of notes to inform predictive models [34].

Current EHR system designs have focused on the administrative side of care delivery forcing clinical users to spend more of their time performing data entry [51,42,13]. Systems do not allow users to refer to, append, or amend prior notes whilst keeping the original document as recorded [7]. To overcome this limitation free text is often copied from prior notes, duplicating data that could otherwise be referenced [35].

This work aims to highlight and quantify an often acknowledged but neglected area of study - the scale of redundancy in EHR text. As redundancy is so prevalent in clinical text the research community must

* Corresponding author.
E-mail addresses: thomas.searle@kcl.ac.uk (T. Searle), zina.ibrahim@kcl.ac.uk (Z. Ibrahim), jamesteo@nhs.net (J. Teo), richard.j.dobson@kcl.ac.uk (R. Dobson).

https://doi.org/10.1016/j.jbi.2021.103938
Received 10 June 2021; Received in revised form 19 August 2021; Accepted 17 October 2021
Available online 23 October 2021

do more to understand where and why this redundancy exists in an effort to minimise and mitigate its effects, allowing for further progress in the diverse use cases of clinical text as previously discussed.

Understanding where the most meaningful data is within a record will enable researchers to better understand where time should be spent preparing data, as well as potentially informing EHR system designers where changes can be made to improve data entry design or other data redundancy reduction mechanisms for future implementations.

We present two approaches to measure redundancy in clinical texts:

- **Information-theoretic redundancy:** We show language models trained and tested on public and private clinical texts consistently show higher levels of redundancy in comparison to open-domain text as demonstrated by information-theory measures of perplexity and cross-entropy [48].
- **Syntactic and semantic redundancy of successive note pairs:**we show average token level redundancy across various clinical note types, through calculation of summarisation metrics of temporally successive note pairs. This measure assumes that successive notes from the same admission and of the same type are 'summaries' of former notes within the same clinical admission. We discuss the implications of recall and precision of these metrics and perform a manual analysis of randomly selected notes.

2. Background

2.1. Prior work

Despite information redundancy in clinical text being widely reported, work to develop methods or measures of redundancy and applying these to clinical text have been limited. Early work investigated lexical matching to measure redundancy [58], presenting a modified Levenshtein edit-distance based algorithm that aligned and measured redundancy of 100 randomly selected admissions [58] reporting an average 78% and 54% redundancy for sign-out and progress notes respectively. Further work applied lexical normalisation, stop word removal followed by a sliding window alignment algorithm over multiple sentences [63], showing a 82% correlation with human annotated expert judgements of redundancy for randomly selected sentences in outpatient notes.

Assessing the semantic similarity of documents provides a more robust method to detect redundancy, as lexical and syntactic variations that may arise when a prior note is summarised or copy/pasted then edited can still be marked redundant. Prior work has used statistical modelling techniques to recognise new relevant information for various note types [64,65].

Automated summarisation systems perform a similar process to redundancy identification. Intuitively, an effective summary will identify the most 'important' sections of a document, highlighting the informative, relevant parts of a document whilst ignoring the redundant sections [38]. An extractive summary of text can be seen as an inverse ranking of redundancy, selecting the least redundant sections of a source text, and an abstractive summarisation performs the same ranking followed by a natural language generation step [32]. Outside of the clinical domain, there is strong interest in models for open-domain free text summarisation [47,40,22,62]. Many of these methods use deep neural network based methods to learn representations that capture lexical, syntactic and semantic meaning of texts to produce coherent and informative summaries. Most methods are *knowledge-free*, having no reliance on external modelled knowledge graphs or databases and learn to write summaries only from input text and the associated reference summary.

The clinical domain is uniquely rich with modelled knowledge graphs such as the UMLS [6] and SNOMED-CT [50]. Applying Named Entity Recognition and Linking systems such as cTakes [44], MetaMap [2] or MedCAT [19] over EHRs and aggregating extracted concepts over groups of documents per admission could determine documents with equivalent extracted concepts as redundant. However, solving such an NER + L task is an ongoing research problem due to the scale of modelled knowledge (i.e. hundreds of thousands of possible concepts) and the variability of clinical text [60,59].

Recently, corpora of synthetic [41] and manually annotated [55] semantic similarity sentence pairs have been used in shared tasks to promote further research and system development in this area [56]. Deep neural models such as BERT [9] and S-BERT [43] achieved high scores from multiple challenge submissions achieving 0.88 correlation in ranking sentences with a similarity scale of 0–5.

To our knowledge there is no prior work that estimates information theoretic content of clinical text and compares such estimates to open-domain text. Prior work has estimated redundancy using sequence alignment algorithms for estimating token-level redundancy, largely not considering semantic redundancy, i.e. the tokens differ across texts but the meaning is equivalent, or they have considered sentence to sentence semantic similarity, training models to predict similarity between sentences.

2.2. Measuring redundancy of text through informativeness

The following sections provide the information theoretic basis for empirically estimating redundancy of clinical text. We initially introduce relevant notation and information theory concepts, then describe how language modelling can be used to estimate redundancy.

Given a language L with a vocabulary V comprised of the number of n symbols $w_1 \ldots w_n \in V$ where w_i is a character, word or *word piece* produced by some tokeniser function Z over text $t, Z(t)$ provides some sequence of w symbols. Given that P is a probability distribution over all symbols in V we can define the average *information* conveyed by a language L via Shannon's Entropy [49]. $H(P)$ is defined as:

$$H(P) = E[I_2(P)] = -\sum_{i=1}^{n} p(w_i)\log_2 p(w_i) \tag{1}$$

Entropy is the negative sum of proportional $\log_2$ probabilities of each symbol w_i with information units represented as bits (i.e. log_2). Intuitively, entropy provides the average number of bits used to convey a symbol from set V for the most efficient coding of L. A maximum bound for the entropy of L is the uniform distribution for P over all symbols in V. Given Eq. 1 this provides:

$$\begin{aligned} H(P) &= \sum_{i=1}^{n} p(w_i)\log_2 p(w_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \log_2 n = \frac{1}{n} n\log_2 n \\ &= \log_2 n \end{aligned} \tag{2}$$

A theoretical lower bound of $H(P) \approx 1$ is if the probability of a single symbol W is $P(W = w_i) \approx 1$ as the probability mass is focused on w_i, i.e. L effectively only has 1 symbol. Eq. 1 holds in the limit of all possible texts that can be produced for L. As we cannot produce all possible texts from L we empirically estimate $H(P)$ with a distribution Q over the same vocabulary V for some, usually large, defined set of texts from L. The cross entropy between distributions P and Q is:

$$H(P, Q) = H(P) + D_{KL}(P \| Q) \tag{3}$$

where $D_{KL}(P \| Q)$ is the Kullback-Leibler(KL) divergence or relative entropy of Q from P. These are the extra bits needed to encode symbols from distribution P through the use of the optimal encoding scheme found through the distribution Q.

2.3. Causal language modelling

Causal Language modelling (LM) is the task to predict the next symbol conditioned on previous symbols. Given a defined set texts from L fitting such a model minimises the $D_{KL}(P||Q)$ term of Eq. 3 therefore providing an estimate of entropy for L. A language model estimates the joint probability of a sentence by conditioning the current symbol w_i on all previous $w_1 \ldots w_{i-1}$:

$$P(w_1, \ldots, w_i) = p(w_1) \ldots p(w_i|w_1, \ldots, w_{i-1}) \tag{4}$$

2.4. *Perplexity and cross-entropy to compare redundancy across texts*

Perplexity (PPL) is the 'surprise' a language model finds having encountered w_n given $w_1, \ldots w_{n-1}$, and is the $2^{H(P,Q)}$ of entropy [18]. Language models are often evaluated using PPL where the lower the score the better the model generalises to unseen texts from language L. Given a language model trained on general purpose text L_{en}, and another language model with the same available vocabulary V trained on clinical text L_{clinic} then comparing PPL/ i.e. cross-entropy by taking log_2(PPL), provides a reflection of the level of *information* and therefore redundancy present in texts across the two languages.

It is however important to highlight that this information theoretic measure of redundancy, i.e. estimating the efficiency of encoding of a given a language given the same language model, does not capture a human level measure of informativeness as clinical texts are subject to a context in which they are written. For example, clinical text progress reports have represent a time series of clinical information and therefore repetitions in text could indicate a continuation or confirmation of prior clinical information and may not necessarily be redundant.

2.5. *Re-purposing summarisation evaluation metrics for sequential note sequences*

The primary purpose of clinical narratives are to document new clinical information. However, EHR data entry often is often poorly designed [5] or users lack sufficient training, time or incentives for clean data entry. This results in frequent use of the copy-paste function with prior data copied into the current note with additions and amendments for the new clinical information [12,35,54]. Therefore, our second set of experiments frame a set of clinical notes of the same type for a given admission as successive summaries of one another and seeks to measure the prevalence of copy-pasted notes from successive note pairs.

We apply n-gram and semantic embedding summarisation metrics to successive pairs of clinical notes. In this context 'recall' captures the proportion of the previous note that is contained in the current note, whereas 'precision' is more ambiguous as successive notes with high precision and high recall indicate a note is redundant (i.e. the content is equivalent), whereas high recall, low precision indicates a summary of the previous note with additional new information. Low recall and low precision indicates a successive note does not summarise prior events at all, we expect this to be the case for procedure and investigative notes such as radiology reports as these events are often standalone, even if they take place during the same admission. There are no clear aims for high precision/ recall such as the case for comparing predictive model performance.

3. Methods

3.1. *Datasets*

Descriptive statistics for datasets and splits are provided in Table 1. We consider two clinical datasets in our analysis, we take a 'stroke' specific subset to compare results to our other clinical dataset:

Table 1
Descriptive statistics for clinical and open domain datasets. Average document length is in characters and a single note type for MIMIC-III is the combined *category* and *description* fields. KCH uses a single field for note type. M-III is the MIMIC-III 'full' dataset and (S) is the stroke (I63.*) primary diagnosis subset. WebText & WikiText-2 do not have # 'Note Types' and WebText is only available as sentences only.

Dataset	# Docs	Avg. Length	# Note Types	Test Set Vocab Size
M-III	1,172,433	2,201	3,127	31,017
M-III (S)	8,213	2,232	241	12,167
KCH	26,348	5,217	1310	27,722
WebText	5000	n/a	n/a	48,105
WikiText2	4358	579	n/a	19,037

- MIMIC-III: [17] A large, freely-available US based ICU dataset collected between 2001 and 2012 containing 53,423 distinct admissions. We consider MIMIC-FULL (∼1.17 M documents) that contains all free text notes for primary coded conditions that appeared at least 20 times (∼41 k admissions), and MIMIC-Stroke (337 admissions) with a primary diagnosis of ICD10 code:I63.*.
- KCH: clinical records for patients diagnosed with Cerebral infarction (ICD10 code:I63.*) from the King's College Hospital (KCH) NHS Foundation Trust, London, UK, EHR. This includes 9,892 distinct admissions and ∼26 K documents. We extract data via the internal CogStack [15] system, an Elasticsearch based ingestion and harmonization pipeline for EHR data. This patient cohort is driven by permitted ethical approval and our ability to compare to a similar patient cohort in MIMIC-Stroke.

Our two open domain English language datasets are available via the HuggingFace Datasets[1] library, and are used to demonstrate the entropy/ PPL of non-clinical open-domain datasets. We use:

- OpenWebText [10]: a recreated openly available version of the original data used to train GPT-2. There is no defined 'test' split so we randomly sample 5000 texts. It is worth noting our base pre-trained language model (GPT-2 [39]) has likely seen some if not all of the samples in this random sample during pre-training. Vocabulary size is 48,105.
- WikiText2 [28]: the test data split of WikiText2, a corpus of 4358 Wikipedia articles often used to assess language models. This data is unseen by all LMs and is used to assess open-domain text language modelling performance.

3.2. *Experimental setup*

3.2.1. *Data preparation*

To exclude very rare conditions or cases that may not represent typical clinical language found in EHRs we extract all MIMIC-III notes and filter the admissions that have a primary diagnosis that appeared ⩾20 times in the dataset. We decided upon this threshold after initial small-scale experimentation. We do not clean the notes from MIMIC-III or KCH in any way, although the MIMIC-III notes have already undergone a de-identification process to remove sensitive information such as dates and names.

3.2.2. *Pre-trained language models*

We estimate the entropy of clinical language using GPT-2 [39] a previous state-of-the-art auto-regressive causal language model, based upon the Transformer [53] architecture that has been pre-trained with the 'WebText' corpus, ∼40 Gb of text data collected from the Web. Model/ tokenizer weights, configurations and model implementations are via the HuggingFace 'transformers' [57] library. We use the base GPT-2 model with 124 M parameters, 12 Transformer block layers with model dimensionality of 768, and vocabulary size 50,257.

[1] https://huggingface.co/docs/datasets/master/.

3.2.3. *Language model fine tuning and PPL calculations*

We fine-tune GPT-2 in a self-supervised manner, i.e. after tokenizing the clinical text we feed each token sequentially into the model, conditioning on previous symbols, we produce the distribution over V via the forward pass of the model, compute the loss and back-propagate the error gradient back through the model to update parameters. Code for tokenizing, training, validating and testing the fine-tuned model for the openly available datasets are made available.[2] We calculate perplexity by concatenating all test set texts and applying a strided sliding window half the size of the model dimension (384) to condition the model and make a token prediction. This method ignores inconsistent sentence breaks, a common problem in EHR text. Importantly, this produces results inline with original GPT-2 [39] work, allowing us to focus on the impact the datasets have on PPL calculations.

3.2.4. *Internote type summary evaluation*

Our second method of estimating levels of redundancy in clincal text applies summarisation evaluation metrics to ordered note pairs as demonstrated in Fig. 1. We firstly group each admission’s note types and order by update time. We apply a sliding window of pairwise evaluations over each note sequence then average over the sequence and admissions. Our output is a table for MIMIC and KCH with the average token level summarisation score per note type. This method measures the level of redundancy between successive clinical notes within the same admission of the same type.

We use a Gestalt Pattern matching algorithm [4] as a baseline that computes the ratio of matching sub-sequences of ‘tokens’, (i.e. whitespace separated words) between each successive note. We then report precision/recall for ROUGE [23] another lexical/syntactic token metric and BERTScore [66] a recent deep-learning model based metric that embeds texts using pre-trained semantic vector space, cosine similarity between the embedded texts produces a similarity score between them. BERTScore was shown to correlate higher with human level judgements of generated summary quality than token based metrics such as ROUGE, somewhat addressing the documented failings of ROUGE [45]. Our clinical texts are longer than the maximum dimension supported by the default and highest performing model configured with BERTScore. Therefore, We use the xlnet-base-cased [61] embeddings due to increased maximum permitted input length. Our scores are normalised to the model baseline to produce an improved uniformity in similarity scores as discussed in the original work [66].

4. Results

We present results for both clinical datasets presented in Section 3 and open datasets originally used to train/test LMs.

4.1. *Estimating entropy of clinical text*

Table 2 reports PPL scores across datasets used to pre-train and further fine-tune GPT-2 models. We report our test set results for the pre-trained GPT-2 and the model fine-tuned to clinical datasets presented in Section 3.1. ‘Test’ values for each dataset provide empirical estimates of entropy for languages L_{en} i.e. OpenWebText, and L_{clinic}, i.e. MIMIC (Stroke/ Full) and KCH.

We show LM performance on validation and test sets, observing that test set PPLs are largely consistent with validation set scores indicating the models are not over-fitting to idiosyncrasies only present in the validation set. We are potentially underfitting the data as we did not especially experiment with techniques such early stopping, learning rate optimisation and architecture optimisation. As the model performance is not the valuable contribution of this work we only used a small number of fixed epochs (i.e. 8) with a scheduled weight decay within the AdamW [24] optimizer (i.e. 0.01).

Our results demonstrate the PPL of clinical texts to be smaller than open domain text. Using Eqs. 2, 3 and computing $log_2(PPL)$ we estimate the *information content* of our open-domain text language $L_{en} = 5.16$ and our clinical language $L_{clinic} = 1.66 - 3.26$. This suggest that clinical text is $\sim 1.5\times$ to $\sim 3\times$ less efficient in encoding *information* than regular open domain text. It is important however to note this *efficiency* is with the respect to the definition of an optimal encoding of a language L. Predictability of texts within L_{clinic} does not necessarily measure the informativeness from a human perspective in comparison to L_{en}.

We further test our models on WikiText-2 dataset to observe open-domain performance after clinical text training. We find that once GPT-2 is further trained with clinical text it loses the ability to accurately model open-domain text resulting in large PPLs. This is seen to a greater extent in MIMIC (Full) compared to MIMIC (Stroke)/ KCH, which is likely due to the MIMIC (Full) model having seen the highest volume of clinical text.

4.1.1. *Perplexity across clinical datasets*

We compare our models trained and tested on available alternative clinical datasets as shown in Table 3. As our MIMIC (Stroke)/ KCH trained models share the common stroke diagnosis we would expect clinical language and the description of symptoms, findings, clinical events, procedures to be similar. Our KCH trained and MIMIC (Stroke) tested model performs modestly, i.e. PPL is still 6–13 points less than open domain PPLs, whereas the MIMIC trained and KCH tested model performs poorly. Surprisingly, the similarity in disorder seems to offer little or no benefit, as KCH trained and testing on both MIMIC test sets produces similar PPLs. MIMIC trained and KCH tested also performs better with *Full* compared with *Stroke*. We believe the poor performance with MIMIC trained models is due to heterogeneity of the KCH dataset, including out patient notes, patient letters, procedure reports etc. whereas MIMIC only contains inpatient ICU notes albeit notes from across specialisms such as physician, nursing, radiology, etc.

4.2. *Token level redundancy*

Fig. 2 shows our results computing summarisation metrics described in Section 3.2.4 for the MIMIC (Full) and KCH datasets. Broadly, our baseline (difflib), ROUGE and BERTScore metrics display similar trends, as seen by coloured gradients consistently decreasing across all metrics for similar types of documents. There are some exceptions in the MIMIC dataset such as *Respiratory: Respiratory Care Shift Note* where our baseline method reports a lower similarity ratio as compared to the summarisation metrics.

We report the micro-averaged median scores for each note type to reduce skew from extremes of either side of the distribution of scores. Recall and precision for ROUGE and BERTScore at each note type are largely equivalent, indicating each note type has on average proportionally equivalent amounts of redundant, i.e. duplicated text, from previous notes (the recall score), and ‘new’ text (the precision score. We observe that this varies substantially according to note type with almost no redundant text with some types, i.e. *Nursing/other:Report* and in contrast the majority of text being redundant, i.e. *Physician:Physician Resident Admission Note.*

Table 4 shows a final average across each metric weighted by total number of tokens within each document and type. Interestingly, recall and precision are equivalent for ROUGE and BERTScore. Intuitively, this indicates that successive notes often have a ‘core’ section which is static throughout an admission and updates are provided by editing certain sections only. This reflects a typical workflow for providing status updates on patient condition or progress.

4.3. *Manual analysis*

We perform a manual analysis of 70 randomly selected note pairs,

[2] https://github.com/tomolopolis/clinical_sum.

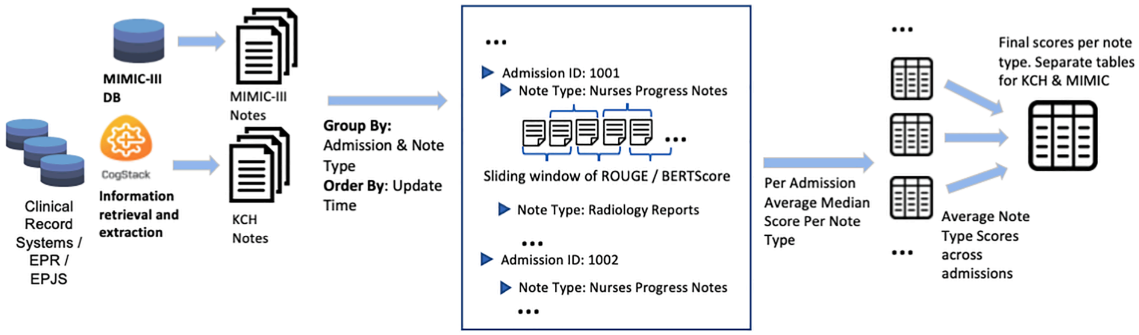

MIMIC-III DB
MIMIC-III Notes
Clinical Record Systems / EPR / EPJS
CogStack
Information retrieval and extraction
KCH Notes
Group By: Admission & Note Type
Order By: Update Time
...
Admission ID: 1001
Note Type: Nurses Progress Notes
...
Sliding window of ROUGE / BERTScore
Note Type: Radiology Reports
...
Admission ID: 1002
Note Type: Nurses Progress Notes
...
Per Admission Average Median Score Per Note Type
...
...
Final scores per note type. Separate tables for KCH & MIMIC
Average Note Type Scores across admissions

Fig. 1. Internote type summarisation evaluation process.

Table 2
Perplexity scores for GPT-2 trained on (Open) WebText (i.e. the model is not trained in this work at all), further training on the MIMIC (Stroke), KCH, and MIMIC (Full) datasets. WikiText2 test split results are also provided for an unseen test set of open-domain text for all models.

Dataset	Val	Test	WikiText2
OpenWebText	–	29.57	35.56
MIMIC (Stroke)	6.14	5.38	144.4
MIMIC (Full)	3.12	3.15	204.9
KCH	8.78	9.58	74.51

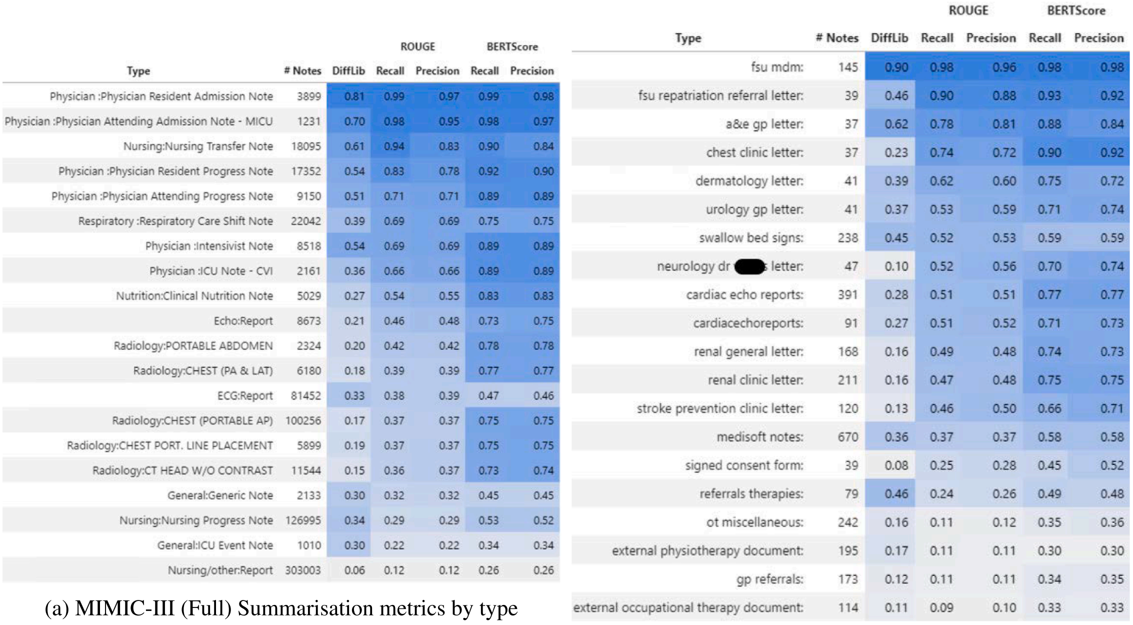
Table 3
GPT-2 trained and tested across our clinical datasets.

Training	Test	PPL
KCH	MIMIC (Stroke)	23.05
KCH	MIMIC (Full)	23.98
MIMIC (Stroke)	KCH	119.66
MIMIC (Full)	KCH	94.19

(35 each from MIMIC-III and KCH). We group, order and split the notes as shown in Fig. 1 and visually highlight the token level differences between successive pairs to assist with determining similarity/ differences. We use a Likert scale of 1–5 to rate redundancy between note pairs and compute a correlation with F1 score. Table 5 shows that ROUGE scores correlate better with our human annotated measure of redundancy than BERTScore.

Table 4
Weighted average by token length of sequential token level redundancy. Rec = Recall, Prec = Precision.

Dataset	DiffLib	ROUGE		BERTScore	
		Rec	Prec	Rec	Prec
MIMIC	0.26	0.43	0.42	0.58	0.58
KCH	0.32	0.49	0.49	0.65	0.65



Type	# Notes	DiffLib	ROUGE		BERTScore	
			Recall	Precision	Recall	Precision
Physician :Physician Resident Admission Note	3899	0.81	0.99	0.97	0.99	0.98
Physician :Physician Attending Admission Note - MICU	1231	0.70	0.98	0.95	0.98	0.97
Nursing:Nursing Transfer Note	18095	0.61	0.94	0.83	0.90	0.84
Physician :Physician Resident Progress Note	17352	0.54	0.83	0.78	0.92	0.90
Physician :Physician Attending Progress Note	9150	0.51	0.71	0.71	0.89	0.89
Respiratory :Respiratory Care Shift Note	22042	0.39	0.69	0.69	0.75	0.75
Physician :Intensivist Note	8518	0.54	0.69	0.69	0.89	0.89
Physician :ICU Note - CVI	2161	0.36	0.66	0.66	0.89	0.89
Nutrition:Clinical Nutrition Note	5029	0.27	0.54	0.55	0.83	0.83
Echo:Report	8673	0.21	0.46	0.48	0.73	0.75
Radiology:PORTABLE ABDOMEN	2324	0.20	0.42	0.42	0.78	0.78
Radiology:CHEST (PA & LAT)	6180	0.18	0.39	0.39	0.77	0.77
ECG:Report	81452	0.33	0.38	0.39	0.47	0.46
Radiology:CHEST (PORTABLE AP)	100256	0.17	0.37	0.37	0.75	0.75
Radiology:CHEST PORT. LINE PLACEMENT	5899	0.19	0.37	0.37	0.75	0.75
Radiology:CT HEAD W/O CONTRAST	11544	0.15	0.36	0.37	0.73	0.74
General:Generic Note	2133	0.30	0.32	0.32	0.45	0.45
Nursing:Nursing Progress Note	126995	0.34	0.29	0.29	0.53	0.52
General:ICU Event Note	1010	0.30	0.22	0.22	0.34	0.34
Nursing/other:Report	303003	0.06	0.12	0.12	0.26	0.26

(a) MIMIC-III (Full) Summarisation metrics by type

Type	# Notes	DiffLib	ROUGE		BERTScore	
			Recall	Precision	Recall	Precision
fsu mdm:	145	0.90	0.98	0.96	0.98	0.98
fsu repatriation referral letter:	39	0.46	0.90	0.88	0.93	0.92
a&e gp letter:	37	0.62	0.78	0.81	0.88	0.84
chest clinic letter:	37	0.23	0.74	0.72	0.90	0.92
dermatology letter:	41	0.39	0.62	0.60	0.75	0.72
urology gp letter:	41	0.37	0.53	0.59	0.71	0.74
swallow bed signs:	238	0.45	0.52	0.53	0.59	0.59
neurology dr [redacted] letter:	47	0.10	0.52	0.56	0.70	0.74
cardiac echo reports:	391	0.28	0.51	0.51	0.77	0.77
cardiacechoreports:	91	0.27	0.51	0.52	0.71	0.73
renal general letter:	168	0.16	0.49	0.48	0.74	0.73
renal clinic letter:	211	0.16	0.47	0.48	0.75	0.75
stroke prevention clinic letter:	120	0.13	0.46	0.50	0.66	0.71
medisoft notes:	670	0.36	0.37	0.37	0.58	0.58
signed consent form:	39	0.08	0.25	0.28	0.45	0.52
referrals therapies:	79	0.46	0.24	0.26	0.49	0.48
ot miscellaneous:	242	0.16	0.11	0.12	0.35	0.36
external physiotherapy document:	195	0.17	0.11	0.11	0.30	0.30
gp referrals:	173	0.12	0.11	0.11	0.34	0.35
external occupational therapy document:	114	0.11	0.09	0.10	0.33	0.33

(b) KCH Summarisation metrics by type

Fig. 2. Summarisation metrics calculated over a sliding window of *generated* and *reference* summaries for admission texts grouped by admission then by note type and ordered by time. We only show the first 20 note types of each dataset ordered by ROUGE score.

Table 5
F1 score correlation of redundancy of ROUGE and BERTScore with manual annotations on a 1–5 Likert scale of a random sample of note pairs.

Dataset	ROUGE	BERTScore
KCH	0.83	0.77
MIMIC	0.77	0.63

5. Discussion

5.1. Language modelling for clinical text

Our PPL scores suggest that clinical text is $\sim 1.5\times$ to $\sim 3\times$ less efficient in encoding information than regular open domain text, or $\sim 1.5\times$ to $\sim 3\times$ more text is used to communicate the same volume of *information* in comparison to open domain text.

To our knowledge this is the first work to estimate in information theoretic terms the entropy of clinical language L_{clinic} and compare against open-domain language L_{en}. These estimates are dependent upon the text and models used, but we believe they are representative as both datasets are large, from varied geographies, hospital sites, specialisms and patient types (outpatient vs inpatient). Our L_{en} corpora are built from curated texts (i.e. Wikipedia and positive karma Reddit posts) that cover a wide array of topics. However, our results may be highly dependent upon these text sources. Future work could compare other easily available datasets such as news or academic papers to provide further clarity on our findings.

Language modelling performance is dependent upon the size of vocabulary of the model and the test set. Model vocab size is static as the same model (GPT-2) and tokenizer configurations are used throughout all experiments. Despite the narrower focus of clinical text, the vocabulary sizes in Table 1 indicate MIMIC-III (Full) and KCH are in fact larger than the WikiText2 corpus although we observe substantially lower PPLs for clinical text. This suggests clinical text is overall less *informative* and therefore more redundant when compared to open-domain corpora. However, this interpretation must be further clarified, as EHRs are written with a clear task in mind to communicate health status, and record clinical events. This is in contrast to open-domain text that has a far wider array of possible tasks for the text.

We compute PPL scores inline with the original GPT-2 authors [39], as this work is an assessment of the data rather than the specific model. A reduced sliding window stride length during PPL calculation would decrease scores further, although relative difference would remain similar. However, we acknowledge that our results are dependent on model architecture, i.e. GPT-2 has higher performing model variants 'GPT-2(large)' even newer variants, 'GPT-3', with an even larger parameter space [8] We propose our results show the trend that clinical domain text is redundant by some multiple compared to open-domain text.

The drop in open-domain text performance after clinical text finetuning suggests the model is incapable of modelling clinical and open-domain text simultaneously. The difference in lexicon and syntax forces the model to minimise a loss landscape substantially different from that found in open-domain text. Further work, could experiment with larger models or with a training process that jointly attempts to model open-domain and clinical text, in an effort to maintain high performance on both. Multiple works [39,40] have already highlighted the effect and importance of data quality, pre-processing and training configuration in LM training.

5.2. Sequential inter-note type redundancy

We used BERTScore configured with *xlnet-base-cased*, due to the size of input texts. The *xlnet-base-cased* embeddings in the BERTScore framework report worse correlation with human annotations of summarisation quality than the default settings that otherwise do not support long input texts. Our manual evaluation of notes against the computed scores indicate ROUGE more accurately captures redundancy than the current BERTScore configuration. During the manual review we noticed BERTScore often scored notes highly that had small token-level differences. As BERTScore projects note pairs into a learnt semantic vector space it is difficult to compare scores with the n-gram based ROUGE. One explanation is that note pairs are likely by the same clinician, are the same clinical specialism and about the same patient and therefore score highly, although n-gram differences are larger. A model such as ClinicalXLNET [14] would likely assist in capturing differences in clinical language thereby producing more appropriate embeddings compared to the open domain variety currently used. We leave this experiment to future work.

This work only considers sequences of notes labelled as the same type. Analysis of intra-note type redundancy, where notes of one type refer to clinical events documented in other note types is another potential avenue of future work. Future work could also order note sequences by clinician, or compare only first and last note for example.

Overall, the interpretation of recall and precision of the summarisation metrics and their relationship to redundant text is nuanced. For example, repeated mentions of an acute condition may simply indicate the continued presence of a condition or symptom, and may not be redundant text after all. These measures do not account for the time series nature of clinical information present in the record. Future work could investigate information extraction, normalisation and linking methods that leverage clinical knowledge bases such as cTakes [44], MetaMap [2] or MedCAT [19]. Extracted concepts could then be compared across notes whilst being grounded in clinical knowledge. This would allow for redundant clinical events to be identified alongside how they present in the text.

6. Conclusions

We have presented two empirical approaches for an often acknowledged [34] but neglected area of clinical natural language processing research, to measure redundancy in clinical text. We have trained large language models on multiple clinical datasets resulting in perplexity and therefore cross-entropy estimates for a clinical language L_{clinic}. We observe a $\sim 1.5\times$ to $\sim 3\times$ reduction in entropy when comparing the same model trained on open domain text. Our approach shows the token level redundancy between different note types with the usage of automated summarisation evaluation metrics. We observe variable scores across different types with some results indicating clinical notes can be 97–98% redundant (i.e. the text is largely duplicated across documents *MIMIC: Physician Resident Admission Note*), or only 0.12% redundant (*MIMIC: Nursing/other:Report*).

Overall, our results support prior work suggesting clinical text contains redundant text [34,58,63]. In information theory terms we show that clinical text is less *efficient* than open domain text meaning on average more text is required to express the same volume of *information* in comparison to general purpose texts. However, this *efficiency* measure does not take into account the context in which EHR records are written, that is a time series of clinical events, where repetition may not necessarily be redundant but indicative of an ongoing condition or clinical event.

With more stressors on our healthcare system than ever before [29] and despite increasing investment [16] we continue to see increased clinician burn-out [33]. A contributing factor is the often enforced usage of EHR systems, increasing doctor-computer time [20], forcing clinicians to overcome poor usability of systems [5]. Improving EHR entry to allow easy updating, cross referencing and versioning of notes could alleviate an extra burden on clinical staff. To this aim we would urge EHR providers to adapt their systems to improve data entry and maintenance, potentially considering features similar to source code management version control allowing for a *living* document to improve data quality, minimise redundancy and errors that are propagated through the usage of copy/paste. We acknowledge this would however require substantial non-trivial changes to systems and user workflow [25,46]. Until EHR providers address these shortcomings researchers will have to rely on ad-ssssshoc pre-processing logic to clean datasets before carrying out analysis.

Data Availability Statement

Open-domain text (OpenWebText and WikiText2) data is openly available as described in Section 3.1. MIMIC-III [17] is freely available but users must obtain permission and a license from dataset owners. KCH data is a highly sensitive dataset and is not easily available. Interested researchers are encouraged to discuss potential projects with the authors to discuss how data access can be granted.

CRediT authorship contribution statement

Thomas Searle: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Zina Ibrahim:** Writing – review & editing, Supervision, Formal analysis. **James Teo:** Resources, Data Curation, Writing – review & editing. **Richard J.B. Dobson:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: JT received research support and funding from InnovateUK, Bristol-Myers-Squibb, iRhythm Technologies, and holds shares <£5,000 in Glaxo Smithkline and Biogen.

Acknowledgements

RD’s work is supported by 1.National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King’s College London. 2. Health Data Research UK, which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, Economic and Social Research Council, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), British Heart Foundation and Wellcome Trust. 3. The National Institute for Health Research University College London Hospitals Biomedical Research Centre. This paper represents independent research part funded by the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King’s College London. The views expressed are those of the author(s) and not necessarily those of the NHS, MRC, NIHR or the Department of Health and Social Care.

References

[1] Amy P. Abernethy, James Gippetti, Rohit Parulkar, Cindy Revol, Use of electronic health record data for quality reporting, J. Oncol. Pract. 13 (8) (2017) 530–534.

[2] A.R. Aronson, Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program, Proc. AMIA Symp. (2001) 17–21.

[3] K. Bruce Bayley, Tom Belnap, Lucy Savitz, Andrew L. Masica, Nilay Shah, Neil S. Fleming, Challenges in using electronic health record data for CER: experience of 4 learning organizations and solutions applied, Med. Care 51 (8 Suppl 3) (2013) S80–6.

[4] Paul E. Black, Ratcliff/Obershelp pattern recognition, in: Dictionary Algorithms Data Struct., 2004, p. 17.

[5] Benjamin Michael Bloom, Jason Pott, Stephen Thomas, David Ramon Gaunt, Thomas C. Hughes, Usability of electronic health record systems in UK EDs, Emerg. Med. J. (2021).

[6] Olivier Bodenreider, The unified medical language system (UMLS): integrating biomedical terminology, Nucleic Acids Res. 32 (Database Issue) (2004) D267–D270.

[7] Sue Bowman, Impact of electronic health record systems on information integrity: quality and safety implications, Perspect. Health Inf. Manag. 10 (2013) 1c.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei, Language models are Few-Shot learners, in: Advances in Neural Information Processing Systems, vol. 33, Curran Associates Inc, 2020, pp. 1877–1901.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[10] Aaron Gokaslan*, Vanya Cohen*, Ellie Pavlick, Stefanie Tellex, OpenWebText corpus, 2019. http://Skylion007.github.io/OpenWebTextCorpus.

[11] William Goossen, Representing knowledge, data and concepts for EHRS using DCM, Stud. Health Technol. Inform. 169 (2011) 774–778.

[12] Robert E. Hirschtick, A piece of my mind. copy-and-paste, JAMA 295 (20) (2006) 2335–2336.

[13] A. Jay Holmgren, N. Lance Downing, David W. Bates, Tait D. Shanafelt, Arnold Milstein, Christopher D. Sharp, David M. Cutler, Robert S. Huckman, Kevin A. Schulman, Assessment of Electronic Health Record Use Between US and Non-US Health Systems, JAMA Intern. Med. 181 (2) (2021) 251–259.

[14] Kexin Huang, Abhishek Singh, Sitong Chen, Edward Moseley, Chih-Ying Deng, Naomi George, Charolotta Lindvall, Clinical XLNet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation, in: Proceedings of the 3rd Clinical Natural Language Processing Workshop, Online, Association for Computational Linguistics, 2020, pp. 94–100.

[15] Richard Jackson, Ismail Kartoglu, Clive Stringer, Genevieve Gorrell, Angus Roberts, Xingyi Song, Wu. Honghan, Asha Agrawal, Kenneth Lui, Tudor Groza, Damian Lewsley, Doug Northwood, Amos Folarin, Robert Stewart, Richard Dobson, CogStack - experiences of deploying integrated information retrieval and extraction services in a large national health service foundation trust hospital, BMC Med. Inform. Decis. Mak. 18 (1) (2018) 47.

[16] Mihajlo Jakovljevic, Yuriy Timofeyev, Chhabi Lal Ranabhat, Paula Odete Fernandes, Jo ao Paulo Teixeira, Nemanja Rancic, Vladimir Reshetnikov, Real GDP growth rates and healthcare spending - comparison between the G7 and the EM7 countries, Global. Health 16(1) (2020) 64.

[17] Alistair E.W. Johnson, Tom J. Pollard, Lu. Shen, Li-Wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, Roger G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (2016) 160035.

[18] Dan Jurafsky, James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, 2009.

[19] Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A. Folarin, Angus Roberts, Rebecca Bendayan, Mark P. Richardson, Robert Stewart, Anoop D. Shah, Wai Keong Wong, Zina Ibrahim, James T. Teo, Richard J.B. Dobson, Multi-domain clinical natural language processing with MedCAT: The medical concept annotation toolkit, Artif. Intell. Med. 117 (2021) 102083.

[20] Philip J. Kroth, Nancy Morioka-Douglas, Sharry Veres, Katherine Pollock, Stewart Babbott, Sara Poplau, Katherine Corrigan, Mark Linzer, The electronic elephant in the room: Physicians and the electronic health record, JAMIA Open 1 (1) (2018) 49–56.

[21] Isotta Landi, Benjamin S. Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T. Dudley, Cesare Furlanello, Riccardo Miotto, Deep representation learning of electronic health records to unlock patient stratification at scale, NPJ Digit. Med. 3 (2020) 96.

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer, BART: Denoising Sequence-to-Sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, Association for Computational Linguistics, 2020, pp. 7871–7880.

[23] Chin-Yew Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.

[24] Ilya Loshchilov, Frank Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2019.

[25] Joseph P. Lyons, Stephen Klasko, Introduction of an electronic medical record system into physician practice offices: why is it so #%!&-ing hard for everybody?-part II, J. Med. Pract. Manage. 26 (6) (2011) 342–345.

[26] Alexander Mathioudakis, Ilona Rousalova, Ane Aamli Gagnat, Neil Saad, Georgia Hardavella, How to keep good clinical records, Breathe (Sheff) 12 (4) (2016) 369–373.

[27] Stuart Maudsley, Viswanath Devanarayan, Bronwen Martin, Hugo Geerts, Intelligent and effective informatic deconvolution of "big data" and its future impact on the quantitative nature of neurodegenerative disease therapy, Alzheimers. Dement. 14 (7) (2018) 961–975.

[28] Stephen Merity, Caiming Xiong, James Bradbury, Richard Socher, Pointer sentinel mixture models, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net, 2017.

[29] Cristina Mesa Vieira, Oscar H. Franco, Carlos Gómez Restrepo, Thomas Abel, COVID-19: The forgotten priorities of the pandemic, Maturitas 136 (2020) 38–41.

[30] Riccardo Miotto, Li Li, Brian A. Kidd, Joel T. Dudley, Deep patient: An unsupervised representation to predict the future of patients from the electronic health records, Sci. Rep. 6 (2016) 26094.

[31] Benjamin J. Miriovsky, Lawrence N. Shulman, Amy P. Abernethy, Importance of health information technology, electronic health records, and continuously aggregating data to comparative effectiveness research and learning health care, J. Clin. Oncol. 30 (34) (2012) 4243–4248.

[32] Hans Moen, Laura-Maria Peltonen, Juho Heimonen, Antti Airola, Tapio Pahikkala, Tapio Salakoski, Sanna Salanterä, Comparison of automatic summarisation methods for clinical free text notes, Artif. Intell. Med. 67 (2016) 25–37.

[33] A. Montgomery, E. Panagopoulou, A. Esmail, T. Richards, C. Maslach, Burnout in healthcare: the case for organisational change, BMJ 366 (2019) l4774.

[34] Travis B. Murdoch, Allan S. Detsky, The inevitable application of big data to health care, JAMA 309 (13) (2013) 1351–1352.

[35] Heather C. O'Donnell, Rainu Kaushal, Yolanda Barrón, Mark A. Callahan, Ronald D. Adelman, Eugenia L. Siegler, Physicians' attitudes towards copy and pasting in electronic note writing, J. Gen. Intern. Med. 24 (1) (2009) 63–68.

[36] Hyojung Paik, Matthew J. Kan, Nadav Rappoport, Dexter Hadley, Marina Sirota, Bin Chen, Udi Manber, Seong Beom Cho, Atul J. Butte, Tracing diagnosis trajectories over millions of patients reveal an unexpected risk in schizophrenia, Sci. Data 6 (1) (2019) 201.

[37] R.H. Perlis, D.V. Iosifescu, V.M. Castro, S.N. Murphy, V.S. Gainer, J. Minnier, T. Cai, S. Goryachev, Q. Zeng, P.J. Gallagher, M. Fava, J.B. Weilburg, S. E. Churchill, I.S. Kohane, J.W. Smoller, Using electronic medical records to enable large-scale studies in psychiatry: treatment resistant depression as a model, Psychol. Med. 42 (1) (2012) 41–50.

[38] Maxime Peyrard, A simple theoretical model of importance for summarization, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019, pp. 1059–1073.

[39] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, Ilya Sutskever, Language models are unsupervised multitask learners, 2019.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, Exploring the limits of transfer learning with a unified Text-to-Text transformer, J. Mach. Learn. Res. 21 (140) (2020) 1–67.

[41] Majid Rastegar-Mojarad, Sijia Liu, Yanshan Wang, Naveed Afzal, Liwei Wang, Feichen Shen, Sunyang Fu, Hongfang Liu, BioCreative/OHNLP challenge 2018, in: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 575.

[42] Raj M. Ratwani, Jacob Reider, Hardeep Singh, A decade of health information technology usability challenges and the path forward, JAMA 321 (8) (2019) 743–744.

[43] Nils Reimers, Iryna Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992.

[44] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, Christopher G. Chute, Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications, J. Am. Med. Inform. Assoc. 17 (5) (2010) 507–513.

[45] Natalie Schluter, The limits of automatic summarisation according to ROUGE, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Short Papers, vol. 2, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017.

[46] Deeann Schmucker, Change management with the electronic health record, J. Med. Pract. Manage. 25 (2) (2009) 93–95.

[47] Abigail See, Peter J. Liu, Christopher D. Manning, Get to the point: Summarization with Pointer-Generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1073–1083.

[48] C.E. Shannon, Prediction and entropy of printed english. Bell Syst. Tech. J. (1951).

[49] C.E. Shannon, The mathematical theory of communication. 1963, MD Comput. 14 (4) (1997) 306–317.

[50] M.Q. Stearns, C. Price, K.A. Spackman, A.Y. Wang, SNOMED clinical terms: overview of the development process and project status, Proc. AMIA Symp. (2001) 662–666.

[51] Ming Tai-Seale, Cliff W. Olson, Jinnan Li, Albert S. Chan, Criss Morikawa, Meg Durbin, Wei Wang, Harold S. Luft, Electronic health record logs indicate that physicians split time evenly between seeing patients and desktop medicine, Health Aff. 36 (4) (2017) 655–662.

[52] Eric J. Topol, High-performance medicine: the convergence of human and artificial intelligence, Nat. Med. 25 (1) (2019) 44–56.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł. Ukasz Kaiser, Illia Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc, 2017, pp. 5998–6008.

[54] Lokesh K. Venkateshaiah, John D. Thornton, The power of the electronic medical record: How often do residents and attendings copy information in critical care progress notes? Chest 138 (4) (2010) 790A.

[55] Yanshan Wang, Naveed Afzal, Fu. Sunyang, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, Hongfang Liu, MedSTS: a resource for clinical semantic textual similarity, Lang. Resour. Eval. 54 (1) (2020) 57–72.

[56] Yanshan Wang, Sunyang Fu, Feichen Shen, Sam Henry, Ozlem Uzuner, Hongfang Liu, The 2019 n2c2/OHNLP track on clinical semantic textual similarity: Overview, 2020.

[57] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander Rush, Transformers: State-of-the-Art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, Association for Computational Linguistics, 2020, pp. 38–45.

[58] Jesse O. Wrenn, Daniel M. Stein, Suzanne Bakken, Peter D. Stetson, Quantifying clinical narrative redundancy in an electronic health record, J. Am. Med. Inform. Assoc. 17 (1) (2010) 49–53.

[59] Yonghui Wu, Min Jiang, Xu. Jun, Degui Zhi, Xu. Hua, Clinical named entity recognition using deep learning models, AMIA Annu. Symp. Proc. 2017 (2017) 1812–1819.

[60] Yonghui Wu, Xu Jun, Min Jiang, Yaoyun Zhang, Xu. Hua, A study of neural word embeddings for named entity recognition in clinical text, AMIA Annu. Symp. Proc. 2015 (2015) 1326–1333.

[61] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, Quoc V. Le, XLNet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc, 2019.

[62] Jingqing Zhang, Yao Zhao, Mohammad Saleh, Peter Liu, PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 119, PMLR, 2020a, pp. 11328–11339.

[63] Rui Zhang, Serguei Pakhomov, Bridget T. McInnes, Genevieve B. Melton, Evaluating measures of redundancy in clinical texts, AMIA Annu. Symp. Proc. 2011 (2011) 1612–1620.

[64] Rui Zhang, Serguei V. Pakhomov, Janet T. Lee, Genevieve B. Melton, Using language models to identify relevant new information in inpatient clinical notes, AMIA Annu. Symp. Proc. 2014 (2014) 1268–1276.

[65] Rui Zhang, Serguei V.S. Pakhomov, Elliot G. Arsoniadis, Janet T. Lee, Yan Wang, Genevieve B. Melton, Detecting clinically relevant new information in clinical notes across specialties and settings, BMC Med. Inform. Decis. Mak. 17 (Suppl 2) (2017) 68.

[66] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi, BERTScore: Evaluating text generation with BERT, in: Eighth International Conference on Learning Representations, 2020b.

5.2.1 Paper Clarifications

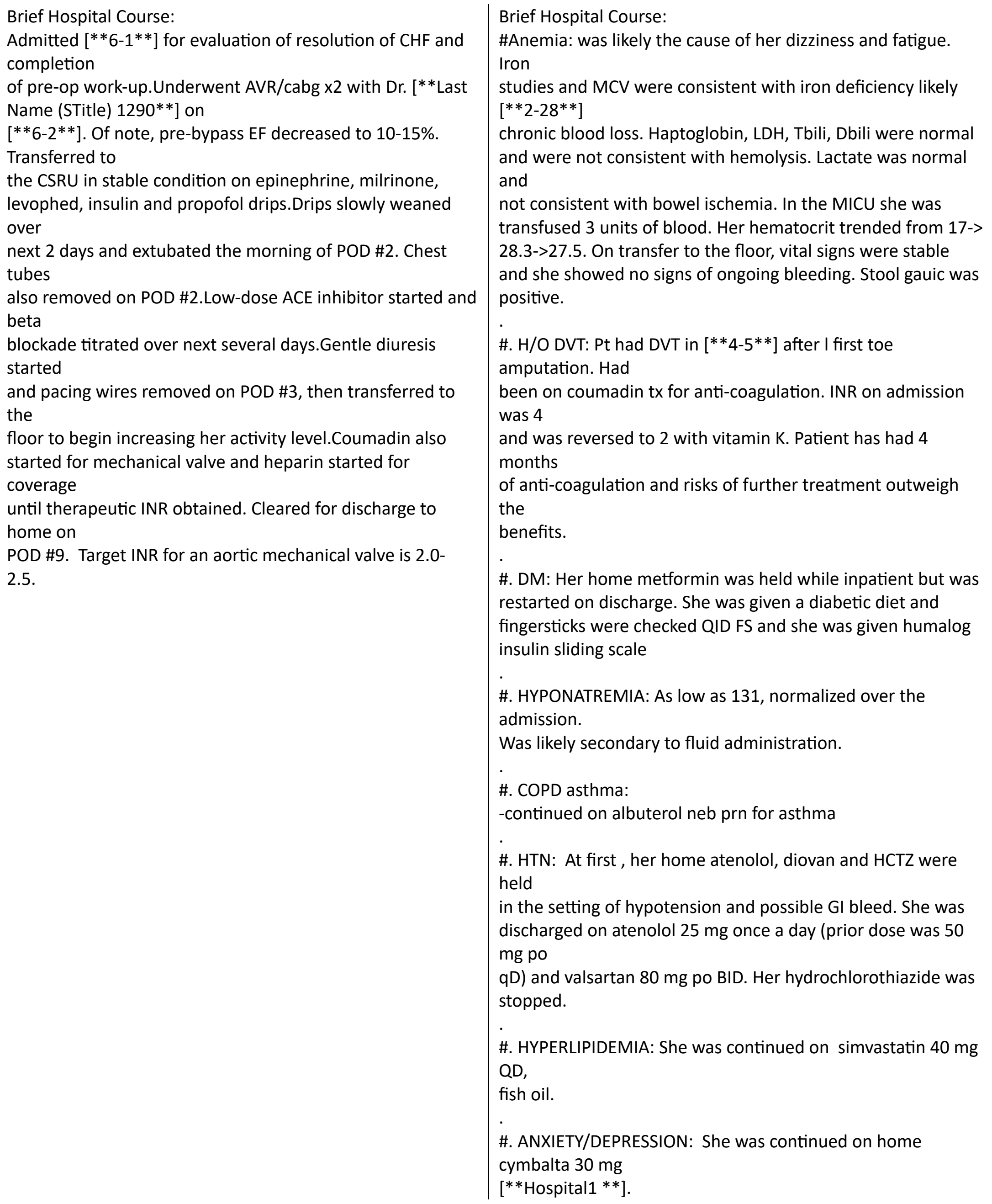
This work has reviewed multiple datasets and presented methodologies for measuring redundancy at scale. Another possible solution to remove these seemingly high measures of redundancy would be to improve user guidance during note creation. This could include the use of specific guidance tailored to users workflow and the EHR system, to allow for referencing portions of the record, rather than copy-pasting or overly relying on template text.

The results in this work support the view that clinical text is *more* redundant than open-domain text, and quantifies this via multiple methods. Overall, this difference in redundancy leads to the following work investigating automated text summarisation, and frames why clinical text summarisation is distinctly different to open-domain text summarisation. As a key task for summarisation is the identification and removal, or alternatively, the selection of non-redundant information, understanding the scale of this problem in clinical text will allow the research community to build more performant and tailored summarisation systems.

5.3 Summarisation of EHRs for Discharge Summaries

In the next below work I experiment with a range of summarisation methods for in-patient clinical text summarisation. The work focuses on summarising the ‘Brief Hospital Course’ section within the ‘Discharge Summary’ note. An example of these sections are shown in Figure 5.1.

Section B.2.1 has provided sufficient background to the methods used. In-patient stay summarisation is already performed manually by senior clinicians who document an admission as part of the discharge process. The complexity of in-patient stays can vary hugely, so I assess the performance of existing pre-trained abstractive models with this task, comparing performance to the open-domain task of multi-document, time-series news



Brief Hospital Course:
Admitted [**6-1**] for evaluation of resolution of CHF and completion
of pre-op work-up.Underwent AVR/cabg x2 with Dr. [**Last Name (STitle) 1290**] on
[**6-2**]. Of note, pre-bypass EF decreased to 10-15%. Transferred to
the CSRU in stable condition on epinephrine, milrinone, levophed, insulin and propofol drips.Drips slowly weaned over
next 2 days and extubated the morning of POD #2. Chest tubes
also removed on POD #2.Low-dose ACE inhibitor started and beta
blockade titrated over next several days.Gentle diuresis started
and pacing wires removed on POD #3, then transferred to the
floor to begin increasing her activity level.Coumadin also started for mechanical valve and heparin started for coverage
until therapeutic INR obtained. Cleared for discharge to home on
POD #9. Target INR for an aortic mechanical valve is 2.0-2.5.

Brief Hospital Course:
#Anemia: was likely the cause of her dizziness and fatigue. Iron
studies and MCV were consistent with iron deficiency likely [**2-28**]
chronic blood loss. Haptoglobin, LDH, Tbili, Dbili were normal and were not consistent with hemolysis. Lactate was normal and
not consistent with bowel ischemia. In the MICU she was transfused 3 units of blood. Her hematocrit trended from 17->28.3->27.5. On transfer to the floor, vital signs were stable and she showed no signs of ongoing bleeding. Stool gauic was positive.
.
#. H/O DVT: Pt had DVT in [**4-5**] after l first toe amputation. Had
been on coumadin tx for anti-coagulation. INR on admission was 4
and was reversed to 2 with vitamin K. Patient has had 4 months
of anti-coagulation and risks of further treatment outweigh the
benefits.
.
#. DM: Her home metformin was held while inpatient but was restarted on discharge. She was given a diabetic diet and fingersticks were checked QID FS and she was given humalog insulin sliding scale
.
#. HYPONATREMIA: As low as 131, normalized over the admission.
Was likely secondary to fluid administration.
.
#. COPD asthma:
-continued on albuterol neb prn for asthma
.
#. HTN: At first , her home atenolol, diovan and HCTZ were held
in the setting of hypotension and possible GI bleed. She was discharged on atenolol 25 mg once a day (prior dose was 50 mg po
qD) and valsartan 80 mg po BID. Her hydrochlorothiazide was stopped.
.
#. HYPERLIPIDEMIA: She was continued on simvastatin 40 mg QD,
fish oil.
.
#. ANXIETY/DEPRESSION: She was continued on home cymbalta 30 mg
[**Hospital1 **].

Fig. 5.1 Example BHC sections from the publicly available dataset MIMIC-III

summarisation [140, 38]. I also experiment with extractive models for initial sentence extraction as prior work observed that initial in patient stay summaries began with extractive summaries [1]. I then adapt the abstractive model to use ‘clinical guidance’ extracted via a pre-trained MedCAT(Chapter 3) model, closing with a simple ensemble model to perform extractive and abstractive summarisation of inpatient summary text.

Journal of Biomedical Informatics 141 (2023) 104358

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Original Research

Discharge summary hospital course summarisation of in patient Electronic Health Record text with clinical concept guided deep pre-trained Transformer models

Thomas Searle [a,*], Zina Ibrahim [a], James Teo [b], Richard J.B. Dobson [a,c]

[a] Department of Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK
[b] King's College Hospital NHS Foundation Trust, London, UK
[c] Institute of Health Informatics, University College London, London, UK

ARTICLE INFO

Keywords:
Clinical natural language processing
Clinical text summarisation
Pre-trained, deep learning, fine-tuned models for clinical summarisation

ABSTRACT

Brief Hospital Course (BHC) summaries are succinct summaries of an entire hospital encounter, embedded within *discharge summaries*, written by senior clinicians responsible for the overall care of a patient. Methods to automatically produce summaries from inpatient documentation would be invaluable in reducing clinician manual burden of summarising documents under high time-pressure to admit and discharge patients. Automatically producing these summaries from the inpatient course, is a complex, multi-document summarisation task, as source notes are written from various perspectives (e.g. nursing, doctor, radiology), during the course of the hospitalisation. We demonstrate a range of methods for BHC summarisation demonstrating the performance of deep learning summarisation models across extractive and abstractive summarisation scenarios. We also test a novel ensemble extractive and abstractive summarisation model that incorporates a medical concept ontology (SNOMED) as a clinical guidance signal and shows superior performance in 2 real-world clinical data sets.

1. Introduction

A patient's clinical journey is documented in rich free-text narratives stored in time-ordered linked documents in Electronic Health Records (EHRs). Narratives include commentary from multiple care teams, specialisms and perspectives with varying scope, detail, structure and time-span covered. Content broadly presents the patient experience, symptoms, findings and diagnosis alongside resulting procedures and interventions. Clinical and social histories and future prognoses are often referenced to provide further context and any potentially follow up actions to occur in some defined time period. Single notes also often mention or refer to previous notes. An encounter such as a simple routine outpatient procedure could generate only a few sentences, whereas a complex admission may result in hundreds of distinct documents. When a patient is discharged from an inpatient encounter, the discharging clinician *summarises* the entirety of the visit often within a section of the *Discharge Summary* note known as the *Brief Hospital Course* (BHC) section. For short, i.e. day case admissions BHC sections are likely to be short and potentially not clearly defined. For longer, multi-day, complex admissions where a patient is being discharged to a primary, community or even tertiary care service this section is more likely to be present as its vital for continuity of care [1]. However, with most free-text clinical narrative, there can be large variability with how this data is presented [2]. Overall, it is generally accepted that an effective discharge summary should document the clinical events of an admission [3].

Manually generating this summary is laborious, time-consuming and potentially error prone [4]. Fig. 1 shows a fictitious, multi-day inpatient encounter. This single admission produces 6 distinct documents from a range of perspectives (Nursing, Doctors, Radiology) in the first 18 h. The first 2 *Nursing - Progress Notes* are by the same author, the differing radiology scans (X-ray vs. MRI) have different authors and the discharge summary is the same author that wrote the initial admission. Discharge occurs ~2 days after admission with more notes taken than those shown. Each document can inform the BHC section. However, not all notes are treated equally, notes are categorised into care provider categories, and further by admission, progress, discharge amongst other types. Due to the volume of text and the time-constraints for doctors to produce these summaries, it is improbable that a clinician author reads the entirety of the record and certainly not thoroughly.

In computational linguistics, this problem can be framed as a challenging multi-document summarisation task, with the model required to adapt to varying numbers of documents (simple vs. complex cases),

[*] Corresponding author.
E-mail address: thomas.searle@kcl.ac.uk (T. Searle).

https://doi.org/10.1016/j.jbi.2023.104358
Received 21 November 2022; Received in revised form 29 March 2023; Accepted 2 April 2023
Available online 5 April 2023

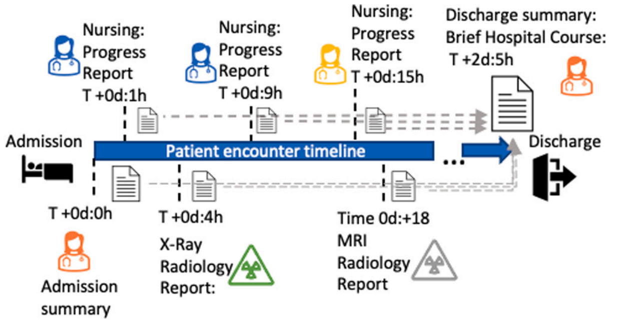

Nursing: Progress Report T +0d:1h
Nursing: Progress Report T +0d:9h
Nursing: Progress Report T +0d:15h
Discharge summary: Brief Hospital Course: T +2d:5h
Admission
Patient encounter timeline
Discharge
T +0d:0h
Admission summary
T +0d:4h
X-Ray Radiology Report:
Time 0d:+18
MRI Radiology Report

Fig. 1. An example patient admission timeline where a patient is admitted with an admission summary note, nursing progress notes, radiology reports and a discharge summary note. Each is written by potentially different authors (colour coding), as the admission progresses. Each note potentially informs the BHC section within the discharge summary. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

large time variances between notes, differences between note types, varying source document authors aims and focus areas.

A recent detailed analysis of BHC sections [5], found BHC summaries to: (1) be information dense, (2) switch quickly between extractive and abstractive summarisation styles, beginning with top-level extractive summaries of an admission followed by *problem orientated* abstractive summary of the admission, (3) be only a *silver-standard* and can lack important information.

To the authors' knowledge, this is the first work to offer a range of summarisation models for BHC summarisation trained and tested on multiple real-world sources of clinical text. The contributions of this work are:

- A baseline evaluation of existing pre-trained Transformer models for abstractive summarisation fine-tuned on the BHC summarisation task.
- An evaluation of extractive top-k sentence extractive summarisation models. Using unsupervised and supervised methods to analyse the *extractiveness* of the opening BHC sentences.
- An adapted abstractive summarisation model (BART) [6] to include a clinical ontology aware guidance signal of relevant terms to produce *problem-list* orientated abstractive summaries.
- An evaluation of an ensemble model for extractive and abstractive summarisation combining the extractive and abstractive models.

2. Background

2.1. Automatic summarisation

Automatic summarisation of text aims to provide a concise, fluent representation of the source material, retaining 'important' information whilst ignoring redundant or irrelevant information. Formally, with single document summarisation, a set of documents $T = \{t_1, t_2, \dots, t_n\}$ we aim to find some function $f(T) = T'$ where $T' = \{t'_1, t'_2, \dots, t'_n\}$ the set of texts that maximise some parameters of an effective summary. These parameters can include: *maximum length* that could vary according to use case, *correctness* if the generated summaries are factually inline with source texts, *completeness*, if the generated summary captures all important information from source texts, and *fluency*, a often subjective measure of the writing quality of the generated summary [7]. In multi-document summarisation we have multiple texts for each sample $T = \{t_{1_{1\dots i}}, t_{2_{1\dots j}}, \dots, t_{n_{1\dots k}}\}$. With BHC summarisation each t_i has one or more documents.

2.2. Extractive & abstractive summarisation

Research interest in automatic summarisation has a long history with empirical data-driven methods divisible into two main groups [8].

1. Extractive summarisation is the selection and combination of important words, phrases, or sentences i.e. some syntactic unit, of source texts to form the summary text. Consider some document text t of syntactic units $S = \{s_1, s_2, \dots, s_3\}$, $f(t) = t'$ where $t' = S'$ and $S' \subset S$.
   Some extractive summarisation methods can be considered Information Extraction (IE) [9] methods that identify important information and simply use s_j where the information is found, or possibly surrounding syntactic units s_{j-1} and s_{j+1}. Information is extracted until desired summary length is reached or there is no more information to extract. Further extractive approaches search/rank a document's S according to some *importance* metric and select the top-n many sentences for the desired summary length [10].
2. Abstractive summarisation methods do not enforce generated summaries to be directly drawn from source texts. Instead, abstractive methods allow f to generate any syntactic unit, i.e. $S' \not\subset S_i$. This means a 'generation' step is used once a latent *importance* model of source texts T is found. Models are often equipped with a suitable vocabulary V and are tasked with generating fluent, informative summary text, whist being guided by the latent *importance* model.

Prior work has combined extractive and abstractive approaches, allowing f to balance abstractive and extractive summarisation, most notably the pointer-generator model [11].

Recently, large pre-trained Transformer [12] models have been shown to perform well across a range of tasks such as machine translation, question answering and abstractive summarisation [13]. The Transformer architecture supports learning of deep latent representations of input data by layering *encoder* and *decoder* blocks, the model learns deep contextual representations of input, and how to decode these representations for a range of sequence-to-sequence tasks.

2.3. Clinical text summarisation

Clinical narratives are estimated to comprise 80% of EHR data [14]. However, the development and application of text summarisation methods is progressing slowly [15] when compared with areas such as disease prediction [16], mortality prediction [17], and clinical information extraction [18]. Contributing factors include: (1) the difficulty in collecting reference summaries [5], *Gold standard* reference summary collection is difficult as the language is complex and highly specialised, (2) produced summaries present a *high stakes* AI scenario that has potential to cause negative downstream effects [19] if the model makes errors, (3) assessing summarisation model performance using automated metrics such as ROUGE [20] is difficult, as high scoring models can still perform poorly when assessed by human evaluators [21].

Prior work has initially focused on extractive approaches [22]. Approaches focused on modelling semantic similarity, and methods to optimally pick representative sentences, i.e. s_i units, from latent topics discovered during model fitting. Recent work, has focused on single document summarisation of radiology reports [23–25]. Radiology reports generally consist of three sections, the *background* of patient, the *findings* – the visible phenomena within the scan and finally the *impression* – an often abstractive summary of the background and findings used during the clinical followup. The *impression* sections are treated as the target reference summaries for model development. Radiology report summarisation is similar to a single document open-domain task, where modelling sentence salience and sentence compression are the primary aims.

2.4. Consistency of discharge summary content

There is currently no standard for discharge summary format or content although a majority of surveyed clinicians agree there should be a standard [2]. There is ongoing work to improve the consistency of education and training in effective discharge summary writing [3,26] and in some fields such as surgical pathology, synoptic reporting is an agreed upon standard form for reporting clinical events [27]. Unfortunately, there is no cross specialty, consistent method of writing a BHC, or even a consistent section header to define the BHC section of a discharge summary. We presume this is due to the variability in clinical encounters that are documented, where it would be very difficult to define rigid structure to cover all eventualities without being overly onerous.

2.5. BHC section analysis

Prior work [5] has shown BHC sections are: (1) dense with clinical terms, (2) can vary widely in complexity and quality, (3) quickly switch between extractive and abstractive styles. These make the BHC summarisation task a difficult task. In this work we attempt to find an effective method that can be consistently used across these varied datasets and that takes advantage of the density of clinical terms. It is outside the scope of this work to address the issue of the variability in discharge summary and BHC sections themselves.

3. Datasets & methods

3.1. Datasets

We extensively pre-process and clean the admission's discharge summaries to extract only the BHC section. We discard the rest of the discharge summary so as to not bias the source texts.

Our datasets are:

- MIMIC-III: Johnson et al. [28] A large, US based ICU dataset collected between 2001–2012 containing 47,591 unique patient admissions. We extract BHC sections from discharge summaries with regular expressions and clean all other notes of headers/footers resulting in 1,441,109 unique documents.
- KCH: clinical records for inpatients diagnosed with cerebral infarction (ICD10 code:I63.*) from the King's College Hospital (KCH) NHS Foundation Trust, London, UK, EHR. We extract data via the Trust deployed CogStack [29] system, an Elasticsearch based ingestion and harmonisation pipeline for EHR data. We extract BHC sections with regular expressions and clean source notes of common headers/footers resulting in 34,179 unique documents.

Table 1 shows that the average case includes many documents, over a multi-day stay. The MIMIC-III dataset of USA based ICU admissions, are skewed towards complex multi-day stays generating many small EHR notes. The KCH dataset are UK-derived clinical records containing only patients diagnosed with cerebral infarction requiring inpatient rehabilitation for associated disability and therefore covers substantially longer time periods.

Concatenating entire patient episode free-text narratives can create very long sequences of text. For encounters that are over 1000 sentences we pick the top and bottom 500 sentences, based on the intuition that patient notes often begin with an important admission note describing the patient history, initial diagnosis and finish with the most recent summary of the patient state. Our source-code for cleaning and preparing the data, and the following model code is made available to the research community.[1]

Table 1
Descriptive statistics for our MIMIC-III (M-III) and KCH clinical text data. From left to right, the number of admissions, the average admission length in days, the average number of notes per admission, the average sequence length of a document excl. the discharge summary, and the average sequence length of the BHC section within the discharge summary.

Dataset	# Adm	Adm length	# Docs	Src Seq	BHC Seq
M-III	47,591	7	26	206	731
KCH	1586	49	21	441	274

3.2. Assessing model performance

We assess performance of our models using ROUGE [20]. The ROUGE authors describe ROUGE-recall to measure the generated summaries 'coverage' of the reference summary or how much of the reference summary is included with the generated summary. ROUGE-precision measures relevancy, or how much of the generated summary is relevant to reference summary. An ideal summary will balance both coverage and relevancy, which can be expressed as the ROUGE-F1 score. A higher ROUGE score correlates with higher human levels of satisfaction with a generated summary but there are still notable issues with the score interpretation [30] . For context, in open-domain summarisation tasks ROUGE often still used and reported. Current state-of-the-art performance is 37–41 points [6].

3.3. Extractive baseline BHC approaches

Our initial experiments test a recent finding that BHC sections are often extractive summaries initially before moving to more abstractive summaries as the BHC section progresses [5]. We compare a range of unsupervised and supervised extractive summarisation models to predict the initial sentences of the BHC sections.

All methods first concatenate each document text in chronological order, split into sentences via Spacy,[2] then embed sentences by averaging GloVe [31] or directly using S-BERT [32] embeddings, finally feeding these to a ranking model, an unsupervised TextRank [33] or supervised Bi-LSTM [34] model. We train multiple models to select top 1 to 15 ranked sentences. Our final baseline model, the Oracle model, uses the reference summary to rank source sentences via Gestalt Pattern matching [35] computing the ratio of matching 'tokens' (i.e. white-space separated words), for each reference summary sentence and source sentence pair. The top k ranked source sentences are used in the oracle. The Oracle model provides an estimate of the performance ceiling of sentence based extractive summarisation for both datasets.

3.4. Pre-trained transformer based models

We consider end-to-end abstractive summarisation models as further baselines. Large pre-trained Transformer [12] models have been successful across a variety of tasks including textual summarisation. Models such as BERT [36], T5 [13] and BART [6] have demonstrated state-of-the-art performance across classification, summarisation, translation, language comprehension and question answering with for the most part a single model architecture. Transformer models for sequence-to-sequence (seq2seq) tasks such as machine translation and summarisation consist of layers of Transformer blocks configured either as *encoders* or *decoders*. Models such as T5 and BART are end-to-end trained for a range of tasks, whereas BERT in its original configuration consisted of encoder only Transformer blocks. Further work has showed BERT models can be repurposed in encoder–decoder configurations for summarisation [37].

Once trained on large, open-domain datasets these models can be re-used on further specialised domains, transferring base knowledge to

[1] https://github.com/tomolopolis/BHC-Summarisation.

[2] https://spacy.io/.

a narrower domain and problem [38]. Transfer learning has recently been shown to be effective for biomedical use cases [39]. However, to our knowledge BHC summarisation has not been considered to date, and our baseline experiments initially establish if large pre-trained models can be fine-tuned to produce high quality BHC sections from source notes directly.

All abstractive models have been pre-trained on large corpora of open-domain text prior to fine-tuning with clinical text. We use existing pre-trained model parameter and configurations from the publicly available huggingface model hub.[3] Then, Fine-tuning is performed using 3 Nvidia Titan X GPUs (M-III experiments) and 8 Nvidia DGX V100 GPUs (KCH experiments). We use different compute for each dataset due to the difference in availability of access, the Nvidia DGX machine is shared, and restricted infrastructure co-located on the KCH site whereas the other hardware is openly accessible on the university network. We split datasets 80/10/10 for training, validation and test. We fine-tune for 20 epochs assessing validation set performance after each epoch. Initial experiments determined learning rate schedule and optimiser parameters and a suitable number of epochs for convergence. We report our results in Section 4 on the test set only.

As discussed in Section 3.1 clinical notes and BHC sections are highly variable in length and complexity. One limitation of recent models are the limited source and target text sequence lengths that can be produced due to the self-attention mechanism requiring all input representations to attend to all others. For example, BERT scales quadratically limiting the max input sequence length to 512. BHC summarisation is difficult as both input source notes are (far) greater than this maximum, as shown in Table 1. Recent models such as the Reformer [40], and LongFormer [41] use various optimisations for the attention calculations to enable longer sequences to be encoded/decoded.

Abstractive summarisation models use source text saliency to focus the summary on only the important parts of the source text. Models must also learn how to faithfully produce source texts alongside ensuring the correct content. Prior work has shown models can be prone to *hallucinations*, producing text that is not representative of the source text [42]. This is problematic for high risk settings such as healthcare but to our knowledge this problem has only been studied for radiology report summarisation [43].

3.5. Clinical concept guided summarisation

Guiding summarisation models using a variety of guidance stimulus forcing the model to focus on specific inputs has recently been shown to be beneficial for open-domain summarisation [44].

We guide our abstractive summariser to focus on summarising the clinical problems and associated interventions of each admission, as is often the method used by clinicians when writing the BHC section [5]. We perform named entity recognition (NER) and entity linking to extract and link SNOMED-CT [45] terms, a standardised clinical terminology via a pre-trained MedCAT [46] model that has been unsupervised trained on both MIMIC-III and KCH datasets for SNOMED-CT *problems* i.e. clinical findings, disease, disorders, and *interventions* i.e. procedures and drugs. Specific top level SNOMED-CT terms provided in Tables 6 and 7. We use *MetaCAT* models configured within MedCAT to contextualise extracted terms. Therefore, all extracted terms are patient-relevant (i.e. not familial history mentions), positive (i.e. not negated), and are classified as a diagnosis (i.e. not mentions of department name or clinical specialisms e.g. "patient attended the stroke clinic" would not annotate stroke as a diagnosis).

Table 8 provides full descriptive statistics of extracted terms across source and BHC notes. An interesting measure 'term density' the average number of all word tokens per each extracted concept. This follows analysis in prior work that showed differences in the density of clinically relevant information between note types [5]. For example, if a sentence were to contain 20 words describing a patient diagnosis, of which our MedCAT model extracts 5 clinical terms, this would provide a term density of 4, as there are 4 word tokens for each clinical term. If of the 5 clinical terms there are only 2 unique clinical terms this provides a unique term density of 10. We observe that for MIMIC-III and KCH datasets the Notes and BHC sections have similar SNOMED-CT term density (56 vs. 52) and word token densities (26 vs. 29), but when considering unique terms the BHC sections have almost double the density of unique clinical terms (63 Notes vs. 118 BHC) for M-III whereas for the KCH notes it is circa equivalent (at 32 Notes vs. 35 BHC), indicating in the M-III dataset BHC sections quickly change from one clinical topic to another when compared to source notes. Redundancy within these datasets have been described in prior work [47].

We use the huggingface[4] BART [6] architecture pretrained on open-domain texts and additionally pretrained on a summarisation corpus PubMed [48]. We choose this architecture as it is specifically tuned for natural language generation (NLG) including summarisation. We follow the architecture modifications outlined in recent work [44]. This includes using dual Transformer based encoders, one for the raw text input and another for the MedCAT extracted guidance input. Importantly, the guidance input is aligned to the text input by padding guidance input, so the dual encoders receive the text and MedCAT extracted concept term at the same sequence step. The model fails to converge without this alignment. These pre-trained parameters are shared for the first 3 encoder Transformer blocks reducing number of model parameters. The rest of the encoder Transformer blocks only see either the original text input or the associated guidance signal. The decoder Transformer blocks is implemented to use an extra cross-attention layer that uses the encoded guidance aware signal to the regular cross-attention layer from the text input encoder representation. Fig. 2 shows our clinical concept guided abstractive summarisation architecture that uses MedCAT-extracted concept sequences to guide the text summariser. We use teacher-forcing for the MedCAT-extracted concept encoder input, and the decoder output embedding signal. Code is made available for the adapted BART model and the input preparation.[5]

We configure the guidance signals to include only *problem* (disease, disorder, finding), and *problem & intervention (drug, procedure)* extracted concepts. This aims to explore the effect of varying the guidance signal across datasets. The original work indicates the guidance signal choice can affect the resulting summary performance [44].

3.6. Extractive and abstractive ensemble model

Our final experiments ensemble the above clinically guided abstractive model with our extractive top-level summary models, therefore utilising both the extractive and abstractive models simultaneously. We predict the initial *n* lines of the BHC section using the extractive model then use the abstractive model with the guidance signal to predict the following sentences. Importantly, the ensemble predictions are fed into the abstractive model—to replicate the scenario of the summarisation model having already produced these sentences.

4. Results

Our results can be interpreted as the balance of generated summary relevancy, i.e. including only content found in the reference summary, and coverage, i.e. content available in the reference summary is in the generated summary. Prior work has shown a positive correlation between the higher the ROUGE scores the high performing summary when manually judged by a human [20].

[3] https://huggingface.co/models.

[4] https://huggingface.co/.

[5] https://github.com/tomolopolis/BHC-Summarisation/blob/master/guidance_models/.

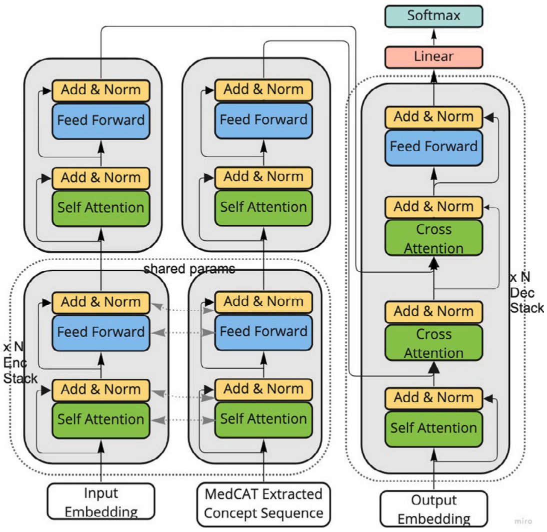


Fig. 2. The encoder–decoder architecture using clinical relevant guidance signal during the encoding, decoding process.

Table 2
ROUGE-LSum F1 scores for the extractive summarisation via sentence ranking for varying sentence limits. **WV** is the Word vector embedding method, and **SB** the sent-BERT embedding method used as input to our modelling approaches TextRank or Bi-LSTM. **Bold** indicates the best score across each sentence limit experiment. The Oracle model results are the performance ceiling for each configuration. Further results available in Appendix B.

Sentence	MIMIC-III					KCH				
limit	TextRank		Bi-LSTM		Oracle	TextRank		Bi-LSTM		Oracle
	WV	SB	WV	SB		WV	SB	WV	SB	
1	0.0	0.0	18.3	**21.8**	30.2	4.09	3.6	4.3	**14.7**	23.3
2	5.6	5.0	17.2	**18.8**	31.1	5.56	5.2	8.3	**10.1**	29.1
3	7.6	5.1	16.6	**17.5**	31.8	6.63	6.4	**10.8**	9.9	31.7
5	18.8	11.3	22.1	**23.5**	32.8	7.61	7.5	**16.1**	12.4	34.2
10	17.9	17.7	27.5	**28.7**	34.3	9.12	9.2	13.4	**20.59**	35.6
15	24.1	28.3	30.1	**31.1**	35.3	13.0	12.9	15.8	**16.0**	35.3

4.1. Extractive models

Our extractive models rank all sentences within the source text to find the top-k salient sentences that comprise the summary. Table 2 show our results across varying initial BHC section sentence limits for the various model embedding and ranking model configurations. Prior work found BHC sections are initially extractive then quickly move to abstractive problem focused narratives [5]. The Oracle model that has access to the target BHC section to rank candidate sentences against, shows the performance ceiling on both datasets is between 5 and 10 of the initial BHC sentences. This is more clearly shown in the KCH dataset with only a very small improvement between 5 and 15 sentences.

Our best performing ranker models use the semantic contextual sentence embeddings from S-BERT and the LSTM ranker across the majority of the sentence limits for both datasets. It is noteworthy that the improvements of using sentence specific embeddings S-BERT vs. average word vectors are minor in comparison to performance improvements from the unsupervised TextRank ranker to the supervised LSTM model. This suggests that relying on relative importance of words and sentences within the documents is an ineffective model, and domain knowledge is needed to build BHCs.

4.2. Abstractive models

Table 3 shows our pre-trained Transformer based models fine-tuned on our datasets. We observe that the performance of these deep pre-trained models is not comparable with open-domain summarisation, even when these models are further pre-trained on biomedical corpora such as PubMed or even MIMIC-III itself. Prior work reports ROUGE-L F1 scores between 37–41 points for BART, BERT, T5 with the open domain summarisation datasets, namely the CNN/Daily Mail [49] and XSum [50] datasets, whereas our results show a range between 7–32 points. The ROUGE-2 performance gap is even larger with open-domain summarisation for these models varying between 19–22 and our results showing a range 1–11 points on our clinical datasets. The BART model pre-trained on PubMed is our best performing model by a substantial margin for both MIMIC-III and KCH BHC summarisation.

Table 3
ROUGE-LSum and ROUGE-2 F1 scores for pre-trained transformer models fine-tuned on the entirety of the BHC Summarisation task.

Model	M-III	KCH
T5-base	7.3/1.3	11.0/6.3
T5-small	14.4/5.6	10.8/4.1
BERT-2-BERT	22.4/4.6	7.4/2.1
BERT-2-BERT (PubMed)	23.8/4.2	6.2/1.6
BERT-2-BERT (M-III)	–	8.6/2.2
BART	26.9/**11.1**	17.1/8.0
BART (PubMed)	**32.7**/**11.1**	**22.1**/**8.6**

Table 4
ROUGE-LSum/ROUGE-2 F1 scores for our clinically guided abstractive summarisation models. BART is pre-trained on the open-domain XSUM [50] datasets, and BART (PubMed) is pre-trained on PubMed [48]. **Bold** indicates the best performance for the metric and dataset.

Model	M-III	KCH
BART	26.9/11.1	17.1/8.0
BART + Prb	26.0/9.1	23.4/12.0
BART + (Prb & Inv)	26.2/8.5	23.4/12.2
BART(PubMed)	32.7/11.1	22.1/8.6
BART(PubMed) + Prb	**34.7**/10.6	**26.6**/**13.7**
BART(PubMed) + (Prb & Inv)	33.6/**11.5**	24.0/12.8

4.3. Clinically guided abstractive summarisation

Table 4 shows our guidance aware abstractive summarisation results. We use 2 different guidance signals extracted by our pretrained MedCAT model. The first signal *Prb* includes only the *problem* extracted concepts. The second *Prb* + *Inv* includes MedCAT extracted problems and interventions.

The M-III BART shows a small drop in performance, 1 and 3 points with both guidance signals, whereas the KCH model improves by 6 and 4 points for ROUGE-LSum and ROUGE-2 respectively. For BART(PubMed) we observe improved ROUGE-LSum performance with both guidance signal types *Prb* and *Prb* + *Inv*. We observe a small gain with ROUGE-2 in MIMIC-III but more noticeable in KCH(4 points). BART(PubMed) experiments show both guidance signals are comparable, with *Prb* offering a marginal improvements when compared to the *Prb* + *Inv* signal, despite there being less *guidance* offered.

4.4. Ensemble extractive/abstractive summarisation

Table 5 shows ablation results for our baseline and ensemble models. *Abs* is the abstractive only model BART with PubMed pre-training. *Ext* + *Abs* is the extractive and abstractive model - S-BERT into Bi-LSTM sentence ranker and BART with PubMed fine-tuning. *Ext* + *Abs* + *Prb* is our final model that is extractive and abstractive with *Problem* extracted clinical term guidance.

We only observe small improved performance through either ensembling with or without guidance. Only the KCH ROUGE-2 score is worse with the ensemble model.

Table 5
ROUGE-LSum and ROUGE-2 F1 score results for our baseline abstractive and ensemble summariser configurations. *Bold* indicates the best performance for the respective metric/dataset pair.

Model	M-III	KCH
Abs	32.7/**11.1**	22.1/**8.6**
Ext + Abs	**34.9**/10.6	**23.6**/7.5
Ext + Abs + Prb	**34.9**/10.6	22.4/6.7

4.5. *Summarisation extracted concept analysis*

Alongside ROUGE scores, we analyse the clinical terms output by our summarisation models. As our guidance signal should push the model to generate more clinically relevant information. We run our pre-trained NER+L model (MedCAT), the same model used to produce the guidance signals, over the generated summaries from the models in Table 5 comparing the proportion of terms in the generated vs. reference summary.

Table 9 provides full results. There are small improvement with both datasets using the guidance model, with summaries having 0%–4% more clinical terms in the guidance models compared to the baseline abstractive models, indicating the guidance signal is assisting the model produce more clinically relevant terms. The guidance assists the generation of problems more so than interventions unsurprisingly as this guidance only includes problem extracted terms. Overall, there is still a majority of concepts (>50%) that are missed entirely by all generated summaries, suggesting there is plenty of room for improvement.

4.6. *Qualitative analysis*

We manually review 40 random summaries from the set of model configurations presented in Table 5 with two clinicians. We compare the generated BHC, the reference summary and the original source notes for only the MIMIC-III dataset due to the sensitivity of the KCH data. Examples of these comparisons can be found in Appendix C. We use a Likert scale 1–5, to measure: (1) coherence - the overall quality of all sentences of the BHC, (2) fluency - the quality of each individual sentence, (3) consistency - the correct facts are in the BHC compared to source notes, (4) relevance - the BHC only contains the relevant facts from the source notes and is not overly verbose. These measures have been used and defined in prior work for large scale qualitative assessment of summarisation texts [51]. Our reviewers - review BHCs blind as to not bias the ratings towards either the reference or generated summaries. We record an average Cohen's Kappa of 0.65 across the 4 metrics. We take the mid point score if there are disagreements. From all ratings there are no disagreements larger than 2 points.

Now we discuss the % of samples with scores ≥2.5 for each metric. For *coherence* we observe that all our models achieve 70% (28/40) vs. 75% (30/40) for the reference summary. For *fluency* our models achieve 60% (24/40) vs. 70%(28/40) for reference summaries. For *consistency* we see a small difference in favour of our guidance model 58% (23/40) vs. abstractive baseline 55% (27/40). Reference summary consistency was at 90% (36/40). Finally, *relevance* showed another small improvement with the guidance models 73%(29/40) vs. 70%(28/40) with the reference summary at 80%(32/40).

We find that the majority of the same summaries are rated ≥2.5 over the 4 metrics. Indicating the variability in difficulty the models encountered with the task. Overall, from this small scale manual analysis it is positive to see that the models, including the baseline abstractive model, performing well across all metrics. However, there is still much room for improvements, with between 30%–40% of produced summaries without acceptable outputs. This poor performing text resulted in common abstractive summarisation issues such as repeated phrases or words within and across sentences, and most worryingly are the occurrences of inconsistencies between source and generated summary facts. For example, 'No documented hypoxia at this hospital' is correctly within the reference but is generated in the BHC summaries: 'Hypoxia: The patient was initially hypoxic on admission to the ICU.'. This inconsistent fact is between multiple consistent facts in the generated summaries. A high performing summary must be near perfect in its consistency to be usable in a real scenario.

A promising result here is that these bad performing summaries were often easy to pick out, and could potentially be systematically excluded if the model were to be included within a real production system. For example the system could decline to 'auto-complete' a summary given a set of admission notes, if the produced summary excessively repeated a phrase or sentence.

We notice that our ensembling strategy to first sample extractive sentences then from the abstractive model do not read as coherently as the abstractive only models. This indicates that summaries move between extractive and abstractive generation at the sub-sentence level, and require a more sophisticated model to balance extractive selection of representative words or phrases alongside abstractive generation, e.g. the Pointer Generator model [11].

5. Discussion and future work

We first discuss our initial baseline methods—our extractive models and our pre-trained fine-tuned abstractive models. We then discuss our guidance signal enhanced model and final ensemble approach. We then discuss a range of issues of our approaches and the problem more broadly. This includes common problems with abstractive models, reference summary quality and the difficulties around real-world clinical text, summarisation metrics and possible future directions for real-world usage of such systems.

Our sentence ranking extractive summarisation experiments suggest the amount of 'extractiveness' for a BHC section depends largely on the dataset. Prior work showed that BHC sections often rely on extractive summarisation initially, i.e. direct copy and paste from source notes into the BHC for the first few sentences, but then quickly switch to abstractive summarisation in later sentences [5]. Our work supports the finding that both extractive and abstractive techniques are used. The M-III dataset shows the opening sentences of the BHCs are more consistently 'extractive' than KCH, as seen by the differences in Oracle model performance as the sentence limit increases. Our best performing model uses a pre-trained contextual sentence embedding model (S-BERT) alongside a Bi-LSTM. Future work could consider further ranking models i.e. a Transformer model to rank sentences, or an appropriate embedding boundary to build sub-sentence, or phrase level embeddings extractive summaries from these. Moreover, we would expect to see different results in sub-sentence level extractions over the whole sentence extractions that we report.

Our fine-tuning of pre-trained abstractive summarisation systems suggest BART, the only model specifically trained for NLG tasks such as summarisation, offers the best performance across datasets and metrics for BHC summarisation. Models such as T5, a general seq-to-seq Transformer model and the BERT-2-BERT models perform substantially worse than BART. For BART we find that further pre-training on a relevant corpus i.e. PubMed [48] compared to only open-domain pre-training, offers improvements inline with prior research [38].

We find that guidance signals for BHC abstractive summarisation offers improvements compared to our best model without guidance. We observe best performance once an existing pre-trained model has already been fine-tuned with biomedical data. We observe that guidance signal improvements are dataset dependent. All experiments use the equivalent hyperparameters, e.g. learning rate, learning rate scheduler, epoch number etc. as the baseline abstractive models. It is likely that further performance gains are possible with further hyperparameter tuning. The guidance models share the parameters for the initial 3 encoder layers. Further work could explore the effect of increasing or decreasing the number of shared parameters.

5.1. Guidance signal

The guidance signal uses a pre-trained MedCAT [46] model. This model has not been validated across the entirety of clinical terms that could be extracted. It has been configured to favour precision over recall, and so likely misses clinical terms that otherwise should be identified and included within the guidance signal. Further work could fine-tune and improve the model performance to improve the guidance signal offered to the summarisation model using the MedCAT annotation tool and workflow [52].

This guidance signal used in our experiments is produced using the MedCAT [46] NER+L approach that is trained unsupervised on the same MIMIC-III and KCH text data. This approach could be replaced with a rules-based, ML or otherwise approach to extract clinical terms. The effectiveness of the clinical term extraction and subsequent usage as a guidance signal will impact the effectiveness of the adapted model. If the NER+L sufficiently under performs and relevant terms are missed, it is very likely the guidance assisted model will perform the same or worse than the standard abstractive model as the decoder stack needs to learn to ignore cross-attention from the encoder.

Moreover, it must be highlighted that our NER+L model has seen the MIMIC-III/KCH data during unsupervised training although it has not received supervised training on any of this data. MedCAT models are based upon a concept dictionary lookup, alongside a concept vector disambiguation algorithm that adapts concept vectors according to the context in which they are found. We have configured the model to highly favour high confidence predictions (i.e. high precision) predictions so it is likely that the majority of predictions are simply dictionary matches. However, the impact of pre-training this guidance model has on our results is unclear and should be considered alongside our results. The guidance signal could be biased and higher performing than signal output by a model that has not seen the input summarisation data. Overall, As reported in prior work [44], further work on guidance signal generation is needed.

For successful model fine-tuning the guidance signal must be aligned with the raw text input. We align the signal by padding the signal with the white space token, but further experiments could investigate aligning the signal with syntactic hints such as punctuation, i.e. full stops, commas, new lines, colons etc. Further work could also experiment with replacing identified guidance terms directly with clinical concept embeddings. During our experiments we attempted to replace the raw text with the standardised terminology name but this lead to model failures and only keeping the original source text allowed for model convergence.

5.2. Ensemble models

We use a very simple ensembling strategy, sampling the extractive model and feeding into the abstractive summariser. Prior work suggests that BHC sections are initially extractive then become abstractive [5]. We find this to be partly true – we reach an Oracle performance limit for both datasets between 10 and 15 sentences – but it is probably at the sub-sentence/phrase level rather than full sentences where summaries are extractive. Further work could explore a PG [11] network architecture, with a mechanism to favour extractiveness initially then abstractive generation afterwards.

5.3. Problems with abstractive summarisation models

Repetition is a known problem with Abstractive summarisation models [53]. Prior work have studied numerous methods to reduce repetition and therefore improve summarisation quality. These include a specific training regime that improves the models ability to sample previously unselected n-grams [54], and a coverage model that adjusts the loss to include words and phrases that sufficiently cover the source text [11]. Repetition is highly unlikely to occur in human generated summaries. Utilising the above techniques would likely improve performance, as observed in open-domain settings [53], although we argue this would still not guide the model to ‘focus’ on the problem-list during summary generation as our method allows.

Factual correctness is an important problem in summarisation and especially important applying these models to clinical scenarios, a high stakes use case that lead to large downstream impacts for model errors. An incorrect statement within a generated BHC summary could miss a diagnosis, follow-up or report a result incorrectly. Our own manual analysis identified various examples especially within long BHCs, of occurrences of inconsistent facts, detecting these and ensuring the model is consistent with the source text is arguably the most important metric in BHC generation. A real deployment of a BHC summarisation system would likely require a ‘human-in-the-loop’ to monitor, similar to most medical AI [55]. The human user would correct, further edit and sign-off on any produced summaries. Even if a system were only able to provide a basic BHC summary, this would still beneficially reduce the administrative burden of completing the BHC section from scratch.

5.4. Reference summary quality

The reference summary BHC sections in both datasets were collected as part of routine care. They have not been reviewed and validated so represent a silver-standard BHCs. Real-world clinical data often does not undergo secondary validation, and even MIMIC-III a heavily studied clinical dataset has data quality concerns [56,57]. It is likely that there are mistakes and omissions in this dataset but given the complexity of clinical text, developing a gold-standard double annotated corpora would be prohibitively expensive. If for example we used two clinicians and each took on average 30 min per admission, one to generate a new summary of each admission, and another to compare both the existing reference and newly written summary this would still take circa. 7.4 years of manual work for both clinicians, of 8 h work days, 5 days a week and 40 weeks a year. This is clearly not going to be possible, across multiple datasets.

However, we argue in line with prior analysis that BHC writing is context and author specific so it is likely another domain expert clinician with different training, geography etc. would result in a different summary [5]. Future work could seek to better understand the variability between BHC sections, or even validate BHC sections creating a gold-standard.

5.5. Summarisation metrics

The ROUGE score shows our guidance assisted and ensemble models offer some but limited improvement. However, in context current top performing ROUGE-LSum scores in open-domain summarisation are at 37–41 points [6] and improvements needed for achieving a few points above the current state-of-the-art is difficult. Analysis using MedCAT extracted concepts and from manual review indicates the addition of guidance helps to produce longer more ‘clinically complete’ summaries despite the similarity in ROUGE score.

ROUGE has been criticised in the literature as summarisation quality can score highly whilst perform poorly during manual evaluation [30]. Alternative metrics such as BERTScore [58] or the recently introduced question answering metrics [59,60] rely on manually generating questions for reference/generated summary pairs or a pre-trained answer conditional question generation model. Assessing our experimental scenarios with these metrics is left to future work, but would likely assist in higher quality, more factually correct summaries. Factual accuracy is critical in BHC generation, as this section is both a legal record and likely to be used by followup care upon discharge.

5.6. Downstream summary use

Automatic generation of BHC sections from source notes is still a long way off. Embedding an automatic summarisation model in *high stakes scenarios* such as healthcare would involve engineering a solution well beyond a research project. Aside from initial validation, ML operations tasks such as detecting model drift or bias would be essential.

In any real-use scenario—a generated summary would likely only be used with explicit clinician supervision and ultimate responsibility for the produced summary, ensuring factual correctness and coherence. Pivovarov and Elhadad [61] provides a further categorisation of generated summaries and how the output is integrated into a workflow. They explain that *indicative* summaries highlight significant or important parts of source texts, whereas *informative* summaries are intended to replace the original text and used in place of it.

6. Conclusions

Our work has demonstrated a range of possible models using both extractive, abstractive summarisation approaches, pre-trained and fine-tuned to specific data and a pre-trained guidance signal generation model (MedCAT) to push the summarisation models to focus on clinically relevant terms. We train a state-of-the-art abstractive model guided by clinically relevant *problem* terms outperforming all baselines across 2 real-world clinical dataset.

Overall, we have shown BHC automated summarisation to be a challenging task supporting prior work [5] suggesting that BHCs are both extractive and abstractive. We hope this work motivates further work in this area that could one day improve the overall healthcare experience for patient and clinician alike through the minimisation of *screen time*. A well documented contributing factor for clinician burn-out [4,62].

CRediT authorship contribution statement

Thomas Searle: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Zina Ibrahim:** Writing – review & editing, Supervision, Formal analysis. **James Teo:** Resources, Data curation, Writing – review & editing. **Richard J.B. Dobson:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: James Teo reports a relationship with Bristol Myers Squibb Co that includes: funding grants. James Teo reports a relationship with Innovate UK that includes: funding grants. James Teo reports a relationship with iRhythm Technologies Inc that includes: funding grants. Richard Dobson reports a relationship with National Institute for Health Research Maudsley Biomedical Research Centre that includes: funding grants. Richard Dobson reports a relationship with Health Data Research UK that includes: funding grants. Richard Dobson reports a relationship with NIHR University College London Hospitals Biomedical Research Centre that includes: funding grants.

Acknowledgements

RD’s work is supported by 1. National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King’s College London. 2. Health Data Research UK, which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, United Kingdom, Economic and Social Research Council, United Kingdom, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), British Heart Foundation and Wellcome Trust. 3. The National Institute for Health Research University College London Hospitals Biomedical Research Centre. This paper represents independent research part funded by the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King’s College London. The views expressed are those of the author(s) and not necessarily those of the NHS, MRC, NIHR or the Department of Health and Social Care.

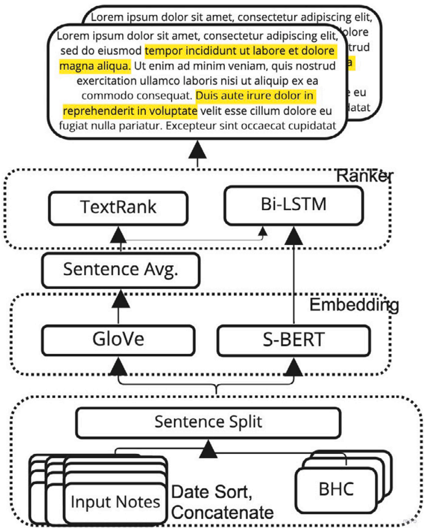

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore
magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat
Ranker
TextRank
Bi-LSTM
Sentence Avg.
Embedding
GloVe
S-BERT
Sentence Split
Input Notes
Date Sort,
Concatenate
BHC

Fig. 3. Baseline Top-N sentence extractive model architectures.

Table 6
The set of ‘Problem’ semantic tags from SNOMED-CT configured within MedCAT, and extracted from source texts and BHC summaries.

Type ID	SCTID root term	Description	# Concepts available
T-11	64572001	Disorder	77,284
T-18	404684003	Clinical finding	44,201
T-29	49755003	Morphologic abnormality	4897
T-35	410607006	Organism	34,778
T-38	260787004	Physical object	198,890

Table 7
The set of ‘Intervention’ semantic tags from SNOMED-CT configured within MedCAT. All SNOMED-CT terms with these semantic terms are extracted from source texts and BHC summaries and treated as ‘Intervention’ terms.

Type ID	SCTID root term	Description	# Concepts available
T-9	373873005	Clinical drug	6247
T-26	373873005	Medicinal product	7715
T-27	373873005	Medicinal product Form	6203
T-39	71388002	Procedure	6,4291
T-40	373873005	Product	17,3894
T-55	105590001	Substance	27,626

Appendix A. MedCAT extracted terms

We configure MedCAT [46] to extract ‘problem’ and intervention terms. Tables 6 and 7 are provided.

Table 8 shows descriptive statistics of the extracted terms of both datasets MIMIC-III and KCH.

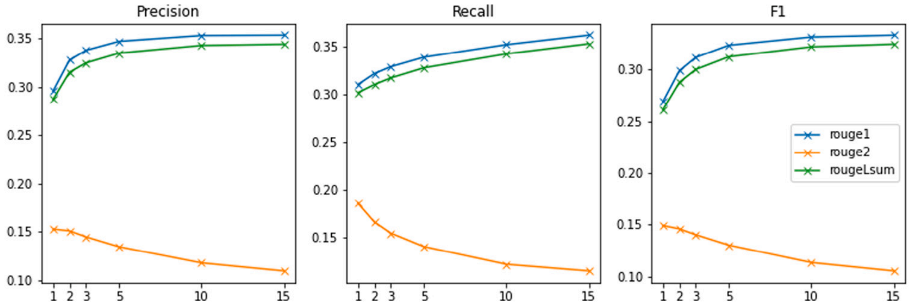


Fig. 4. Extractive score max.

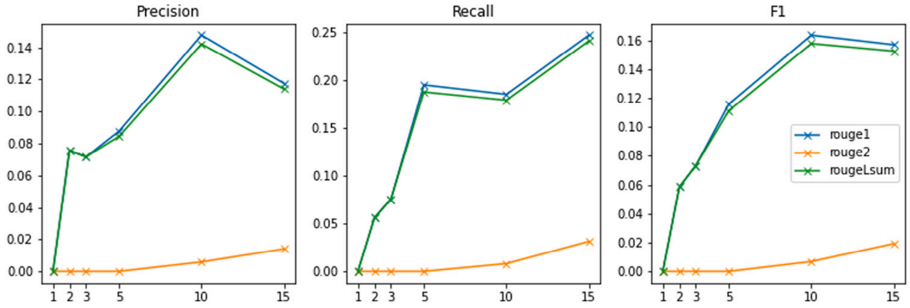


Fig. 5. GloVe Embeddings: TextRank.

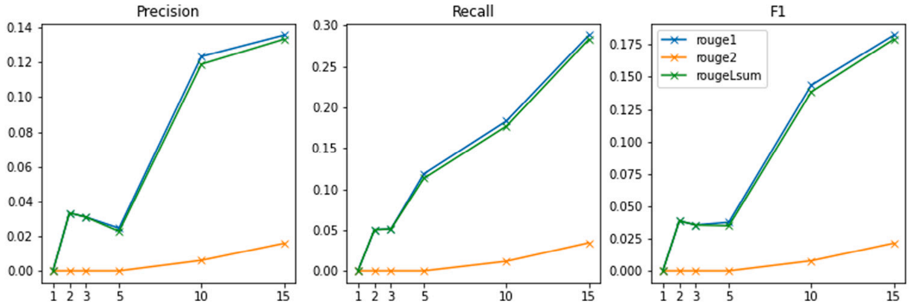


Fig. 6. S-BERT embeddings: TextRank.

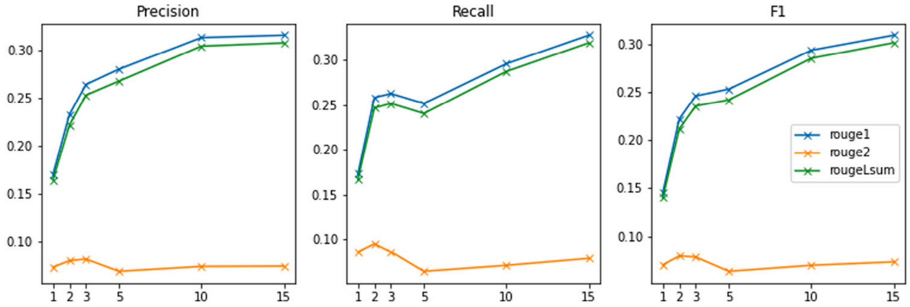


Fig. 7. S-BERT embeddings: Bi-LSTM.

Appendix B. Extractive baseline architectures

Fig. 3, shows our baseline extractive model architectures (see Fig. 4).

Appendix C. Extractive summarisation plots precision, recall, F1 plots appendix

Extractive Summarisation Methods for Top-N-Line BHC Summarisation (see Figs. 5–7).

Table 8
Extracted and linked average: term counts, unique term counts, and their respective densities with regards to the number tokens per clinical term.

		Dataset	
		M-III	KCH
Notes	# Terms	156	110
	Term density	55	26
	# Uniq terms	56	43
	Uniq term density	118	32
BHC	# Terms	19	10
	Term density	52	29
	# Uniq terms	15	8
	Uniq term density	63	35

Table 9
MedCAT Extracted Term comparisons vs. reference summary. Average % of problem only, intervention only and both problem & intervention terms in the generated vs. the reference summary. **Bold** indicates model with highest proportion of clinical terms generated compared with reference summary.

Dataset	Model	% Prob	% Inv	% Total
M-III	Abs	31	32	34
	Ext + Abs	33	33	34
	Ext + Abs + Prb	**35**	**35**	34
KCH	Abs	40	30	38
	Ext + Abs	41	**34**	41
	Ext + Abs + Prb	**43**	**34**	**42**

Appendix D. Measures of clinically relevant information across summarisation models

Table 9 shows the proportion of concepts that we successfully generate in the predicted summaries vs. the reference summaries.

References

[1] Aaron M Silver, Leigh Anne Goodman, Romil Chadha, Jason Higdon, Michael Burton, Venkataraman Palabindala, Nageshwar Jonnalagadda, Abey Thomas, Christopher O'Donnell, Optimizing discharge summaries: A multispecialty, multicenter survey of primary care clinicians, J. Patient Saf. 18 (1) (2022) 58–63.

[2] Atsushi Sorita, Paul M Robelia, Sharma B Kattel, Christopher P McCoy, Allan Scott Keller, Jehad Almasri, Mohammad Hassan Murad, James S Newman, Deanne T Kashiwagi, The ideal hospital discharge summary: A survey of U.S. Physicians, J. Patient Saf. 17 (7) (2021) e637–e644.

[3] David Ming, Kahli Zietlow, Yao Song, Hui-Jie Lee, Alison Clay, Discharge summary training curriculum: a novel approach to training medical students how to write effective discharge summaries, Clin. Teach. 16 (5) (2019) 507–512.

[4] Heather C O'Donnell, Rainu Kaushal, Yolanda Barrón, Mark A Callahan, Ronald D Adelman, Eugenia L Siegler, Physicians' attitudes towards copy and pasting in electronic note writing, J. Gen. Intern. Med. 24 (1) (2009) 63–68.

[5] Griffin Adams, Emily Alsentzer, Mert Ketenci, Jason Zucker, Noémie Elhadad, What's in a summary? Laying the groundwork for advances in Hospital-Course summarization, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2021, pp. 4794–4811, Online.

[6] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer, BART: Denoising Sequence-to-Sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 7871–7880, Online.

[7] Philippe Laban, Andrew Hsi, John Canny, Marti A. Hearst, The summary loop: Learning to write abstractive summaries without examples, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Vol. 1, 2020.

[8] Constantin Orăsan, Automatic summarisation: 25 years On, Nat. Lang. Eng. 25 (6) (2019) 735–751.

[9] Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, Kiri Wagstaff, Multidocument summarization via information extraction, in: Proceedings of the First International Conference on Human Language Technology Research - HLT '01, Association for Computational Linguistics, Morristown, NJ, USA, 2001.

[10] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, Xuanjing Huang, Searching for effective neural extractive summarization: What works and what's next, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 1049–1058.

[11] Abigail See, Peter J. Liu, Christopher D. Manning, Get to the point: Summarization with Pointer-Generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1073–1083.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł Ukasz Kaiser, Illia Polosukhin, Attention is all you need, in: I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, R Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5998–6008.

[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, Exploring the limits of transfer learning with a unified Text-to-Text transformer, J. Mach. Learn. Res. 21 (140) (2020) 1–67.

[14] Travis B. Murdoch, Allan S. Detsky, The inevitable application of big data to health care, JAMA 309 (13) (2013) 1351–1352.

[15] Rashmi Mishra, Jiantao Bian, Marcelo Fiszman, Charlene R Weir, Siddhartha Jonnalagadda, Javed Mostafa, Guilherme Del Fiol, Text summarization in the biomedical domain: a systematic review of recent research, J. Biomed. Inform. 52 (2014) 457–467.

[16] Laure Wynants, Ben Van Calster, Marc M J Bonten, Gary S Collins, Thomas P A Debray, Maarten De Vos, Maria C Haller, Georg Heinze, Karel G M Moons, Richard D Riley, Ewoud Schuit, Luc J M Smits, Kym I E Snell, Ewout W Steyerberg, Christine Wallisch, Maarten van Smeden, Prediction models for diagnosis and prognosis of covid-19 infection: systematic review and critical appraisal, BMJ 369 (2020) m1328.

[17] Alistair E.W. Johnson, Tom J. Pollard, Roger G. Mark, Reproducibility in critical care: a mortality prediction case study, in: Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, Jenna Wiens (Eds.), Proceedings of the 2nd Machine Learning for Healthcare Conference, in: Proceedings of Machine Learning Research, vol. 68, PMLR, Boston, Massachusetts, 2017, pp. 361–376.

[18] Kory Kreimeyer, Matthew Foster, Abhishek Pandey, Nina Arya, Gwendolyn Halford, Sandra F Jones, Richard Forshee, Mark Walderhaug, Taxiarchis Botsis, Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review, J. Biomed. Inform. 73 (2017) 14–29.

[19] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, Lora M Aroyo, "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1–15.

[20] Chin-Yew Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, 2004, pp. 74–81.

[21] Ananya B. Sai, Akash Kumar Mohankumar, Mitesh M. Khapra, A survey of evaluation metrics used for NLG systems, 2020, arXiv [cs.CL].

[22] Hans Moen, Laura-Maria Peltonen, Juho Heimonen, Antti Airola, Tapio Pahikkala, Tapio Salakoski, Sanna Salanterä, Comparison of automatic summarisation methods for clinical free text notes, Artif. Intell. Med. 67 (2016) 25–37.

[23] Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D Manning, Curtis P Langlotz, Learning to summarize radiology findings, in: Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 204–213.

[24] Ravi Kondadadi, Sahil Manchanda, Jason Ngo, Ronan McCormack, Optum at MEDIQA 2021: Abstractive Summarization of Radiology Reports using simple BART finetuning, in: Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, 2021, pp. 280–284, Online.

[25] Songtai Dai, Quan Wang, Yajuan Lyu, Yong Zhu, BDKG at MEDIQA 2021: System report for the radiology report summarization task, in: Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, 2021, pp. 103–111, Online.

[26] Elizabeth Stopford, Sean Ninan, Nick Spencer, How to write a discharge summary, BMJ 351 (2015).

[27] Andrew A Renshaw, Mercy Mena-Allauca, Edwin W Gould, S Joseph Sirintrapun, Synoptic reporting: Evidence-based review and future directions, JCO Clin. Cancer Inform. 2 (2018) 1–9.

[28] Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, Roger G Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (2016) 160035.

[29] Richard Jackson, Ismail Kartoglu, Clive Stringer, Genevieve Gorrell, Angus Roberts, Xingyi Song, Honghan Wu, Asha Agrawal, Kenneth Lui, Tudor Groza, Damian Lewsley, Doug Northwood, Amos Folarin, Robert Stewart, Richard Dobson, CogStack - experiences of deploying integrated information retrieval and extraction services in a large National Health Service Foundation Trust hospital, BMC Med. Inform. Decis. Mak. 18 (1) (2018) 47.

[30] Natalie Schluter, The limits of automatic summarisation according to ROUGE, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017.

[31] Jeffrey Pennington, Richard Socher, Christopher D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[32] Nils Reimers, Iryna Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992.

[33] Rada Mihalcea, Paul Tarau, TextRank: Bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411.

[34] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[35] Paul E. Black, Ratcliff/Obershelp pattern recognition, in: Dictionary of Algorithms and Data Structures, Vol. 17, 2004.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[37] Sascha Rothe, Shashi Narayan, Aliaksei Severyn, Leveraging pre-trained checkpoints for Sequence Generation tasks, Trans. Assoc. Comput. Linguist. 8 (2020) 264–280.

[38] Anna Rogers, Olga Kovaleva, Anna Rumshisky, A primer in BERTology: What we know about how BERT works, Trans. Assoc. Comput. Linguist. 8 (2020) 842–866.

[39] Yifan Peng, Shankai Yan, Zhiyong Lu, Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets, in: Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy, 2019, pp. 58–65.

[40] Nikita Kitaev, Lukasz Kaiser, Anselm Levskaya, Reformer: The efficient transformer, in: International Conference on Learning Representations, 2020.

[41] Iz Beltagy, Matthew E. Peters, Arman Cohan, Longformer: The Long-Document transformer, 2020, arXiv:2004.05150.

[42] Zheng Zhao, Shay B. Cohen, Bonnie Webber, Reducing quantity hallucinations in abstractive summarization, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, 2020, pp. 2237–2249, Online.

[43] Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D Manning, Curtis Langlotz, Optimizing the factual correctness of a summary: A study of summarizing radiology reports, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 5108–5120, Online.

[44] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, Graham Neubig, GSum: A general framework for guided neural abstractive summarization, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2021, pp. 4830–4842, Online.

[45] M. Q. Stearns, C. Price, K. A. Spackman, A. Y. Wang, SNOMED clinical terms: overview of the development process and project status, in: Proc. AMIA Symp., 2001, pp. 662–666.

[46] Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A Folarin, Angus Roberts, Rebecca Bendayan, Mark P Richardson, Robert Stewart, Anoop D Shah, Wai Keong Wong, Zina Ibrahim, James T Teo, Richard J B Dobson, Multi-domain clinical natural language processing with MedCAT: The Medical Concept Annotation Toolkit, Artif. Intell. Med. 117 (2021) 102083.

[47] Thomas Searle, Zina Ibrahim, James Teo, Richard Dobson, Estimating redundancy in clinical text, J. Biomed. Inform. 124 (2021) 103938.

[48] Vivek Gupta, Prerna Bharti, Pegah Nokhiz, Harish Karnick, SumPubMed: Summarization dataset of PubMed scientific articles, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, Association for Computational Linguistics, Stroudsburg, PA, USA, 2021.

[49] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulcehre, Bing Xiang, Abstractive text summarization using sequence-to-sequence RNNs and beyond, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 280–290.

[50] Shashi Narayan, Shay B. Cohen, Mirella Lapata, Don't give me the details, just the summary! Topic-Aware convolutional neural networks for extreme summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1797–1807.

[51] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, Dragomir Radev, SummEval: Re-evaluating summarization evaluation, Trans. Assoc. Comput. Linguist. 9 (2021) 391–409.

[52] Thomas Searle, Zeljko Kraljevic, Rebecca Bendayan, Daniel Bean, Richard Dobson, MedCATTrainer: A biomedical free text annotation interface with active learning and research use case specific customisation, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 139–144.

[53] Pranav Ajit Nair, Indian Institute of Technology (BHU), India Varanasi, Anil Kumar Singh, Indian Institute of Technology (BHU), India Varanasi, On reducing repetition in abstractive summarization, in: Proceedings of the Student Research Workshop Associated with RANLP 2021, INCOMA Ltd. Shoumen, BULGARIA, 2021.

[54] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, Jason Weston, Neural text generation with unlikelihood training, in: Eighth International Conference on Learning Representations, 2020.

[55] Fabrice Jotterand, Clara Bosco, Keeping the "human in the loop" in the age of artificial intelligence : Accompanying commentary for "correcting the brain?" by rainey and erden, Sci. Eng. Ethics 26 (5) (2020) 2455–2460.

[56] Thomas Searle, Zina Ibrahim, Richard Dobson, Experimental evaluation and development of a Silver-Standard for the MIMIC-III clinical coding dataset, in: Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, Association for Computational Linguistics, 2020, pp. 76–85, Online.

[57] Ali S Afshar, Yijun Li, Zixu Chen, Yuxuan Chen, Jae Hun Lee, Darius Irani, Aidan Crank, Digvijay Singh, Michael Kanter, Nauder Faraday, Hadi Kharrazi, An exploratory data quality analysis of time series physiologic signals using a large-scale intensive care unit database, JAMIA Open 4 (3) (2021) ooab057.

[58] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, Yoav Artzi, BERTScore: Evaluating text generation with BERT, in: Eighth International Conference on Learning Representations, 2020.

[59] Matan Eyal, Tal Baumel, Michael Elhadad, Question answering as an automatic evaluation metric for news article summarization, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3938–3948.

[60] Alex Wang, Kyunghyun Cho, Mike Lewis, Asking and answering questions to evaluate the factual consistency of summaries, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 5008–5020, Online.

[61] Rimma Pivovarov, Noémie Elhadad, Automated methods for the summarization of electronic health records, J. Am. Med. Inform. Assoc. 22 (5) (2015) 938–947.

[62] Eugenia McPeek-Hinz, Mina Boazak, J Bryan Sexton, Kathryn C Adair, Vivian West, Benjamin A Goldstein, Robert S Alphin, Sherif Idris, W Ed Hammond, Shelley E Hwang, Jonathan Bae, Clinician burnout associated with sex, clinician type, work culture, and use of electronic health records, JAMA Netw. Open 4 (4) (2021) e215686.

5.3.1 Discussion

This work covers only BHC text summarisation. This is only a section, albeit an important one, of the discharge summary of an admission. Text within the discharge summary, that requires external world knowledge or knowledge base, as all information is not necessarily within the prior notes, is also needed for a comprehensive EHR summarisation system. Prior work has presented comprehensive EHR summarisation as longitudinal summarisation, with systems such CliniText [43] utilising expert knowledge to influence summary generation. Effective summarisation could also include a multi-modal system capable of summarising using both the structured and unstructured portion of the record [80]. Further work could look to extend or integrate our BHC generation method into a larger multi-modal longitudinal summarisation system.

A key finding from this work has been the difficulty in assessment of effective summarisation, and the reliance on manual review still being necessary. Recent work has attempted to address the shortfalls of metrics such as ROUGE, BERTScore [166] or QAGS [150]. However, both these scores rely on deep Transformer based models to produce scores or data for an assessment protocol.

5.4 Chapter Summary

This chapter has focused on the analysis of clinical free-text in the context of textual summary generation. I firstly analysed and compared clinical text to open-domain text, and secondly developed and tested a range of textual summarisation models and scenarios for inpatient summary generation. I again used the NER+L method presented in Chapter 3, to assist with the summarisation task, leveraging a pre-modelled clinical knowledge graph.

The next and final chapter will summarise the contributions of the thesis, challenges and lessons I have learned and potential future directions for the work.

Chapter 6

Wider Discussion, Future Work and Conclusions

This final chapter will summarise the main findings of this thesis. I will summarise the contributions of the work, how they fit into the wider context of clinical data research and why they are important. I will also highlight future directions of the work alongside current and future challenges to further progress.

I will conclude this chapter and thesis by reflecting firstly on the key enablers of the research, how the contributions can be replicated and scaled across datasets, clinical areas and settings. I will then conclude with how I hope this and future work can in some small way provide the means to improve delivery of clinical care.

6.1 Summary of Research Findings

6.1.1 Named Entity Recognition and Linking for Clinical Concepts

The first published work of the thesis presents the Medical Concept Annotation Toolkit (MedCAT). This focuses on the problem of named entity recognition and linking (NER+L) for clinical text. This is an important problem as the majority (~80%) of EHR data is unstructured [99] and asking even seemingly simple questions can be difficult. For example, finding all patients with a certain condition or the number of a patients prescribed a certain medication is non-trivial. MedCAT importantly leverages significant previous investments in the creation of clinical ontologies (e.g. SNOMED-CT [144] and UMLS [17]) that define clinical concepts and the relationships between them. Identifying the portions of text and linking them to these ontological terms is a first step in making use of the unstructured data 'unlocking' the data for specific successive use cases. MedCAT also provides further models to classify extracted concepts for any number of 'Meta Annotations' or properties. The underlying model uses a Bidirectional-LSTM [52] and can be configured to classify across any number of classification tasks. For example - temporality - a concept being referred to in the past, present or future, negations - a concept is present, or explicitly stated as not present or diagnosis - a term used in the diagnostic sense, or some other reference.

Our novel methods empirically demonstrate improved performance over existing methods such as cTakes [130] and MetaMap [6]. Importantly, our work acknowledges and relies upon domain expert human validation of trained models via the MedCATtrainer tool. Fine-tuning via annotation collection is also available allowing the model to specialise in clinical areas guided by domain expertise or even be localised and modified if the terminological system does not adequately represent requirements.

MedCAT is an open-source software library written to support the addition of further clinical informatics models. Recent developments from colleagues have included:

1. Relation Annotations: further modules in MedCAT and MedCATtrainer are being developed to annotate relations across two clinical terms. For example, the relation *symptom of* for a *finding* term and a *disorder* term, or *dose*, *frequency* relations for medication terms.

2. Document Annotations: MedCATtrainer modules have been prototyped to support the collection of document level annotations. These models would be configurable to

support any definable classification task, similar to 'Meta Annotations'. The backing model would likely be a deep pre-trained Transformer model i.e. BERT [33].

3. De-Identification Annotations: Colleagues within my department have recently completed experiments on a de-identification pipeline using a BERT based model to identify and replace sensitive data. A custom ontology for sensitive data has been constructed, trained and validated. Early data is promising, and the pipeline step has been included open-source within the library. A high performing De-ID model could alleviate the difficulties in data sharing providing more health data researchers access to data.

These new features are being developed with the same 'workflow' as the Meta Annotation models. This means a model's training data is defined and collected through the MedCATtrainer interface. Then model training, validation and testing occur directly through the MedCAT API operating directly on the trainer exported data. Finally, once a model has been trained the library allows for easy exporting and re-use.

All of the methods presented here operate on text. Structuring the data at point of entry could alleviate the need for these methods, but despite large continued investment in EHR systems offering tailored forms and workflows, data entry through natural language is still preferred [54]. Preliminary work at University College London Hospitals aims to combine the flexibility of natural language entry with MedCAT models to extract and contextualise clinical terms in real-time at point of entry[1].

For historical text data the CogStack group within my wider research lab are working towards secure and scalable model sharing for further CogStack deployments to use and contribute to model fine-tuning. These models aim to be continually improved by running them upon new data as it is being generated, i.e. for self-supervised training, and further annotated data collected for downstream clinical research projects, i.e. for supervised training.

[1] https://www.uclhospitals.brc.nihr.ac.uk/criu/research-impact/medical-information-ai-data-extractor

Future work also aims to ensure manually collected annotations are utilised beyond their initial clinical projects. I aim to build and integrate machine learning operations (MLOps) tooling into MedCATtrainer, such as MLFlow[2], to enable storage, serving, discovering, and traceability of models.

6.1.2 Summarisation Methods for Clinical Coding

Chapter 4 focused on the application of our methods to an existing summarisation task within healthcare delivery - namely Clinical Coding. This is the administrative process of extracting *clinical codes* from patient *episodes*. Episodes are variable periods of care ranging from a single encounter such as an inpatient *day-case* encounter, or a large complex admission with many encounters across multiple specialties and teams. Clinical codes represent the diagnoses and associated interventions for an episode and are represented in clinical coding taxonomies such as ICD-10 or OPCS-4 in the UK. Clinical coding is a skilled but mostly manual, potentially error-prone process where clinical coders identify and aggregate the correct codes given a set of episodes notes. Clinical coders do not make clinical judgements so a given code must be explicitly written within the notes for a code to be extracted.

Recently, the latest NLP methods have been applied to clinical coding [97, 10, 162, 24], as the problem can be modelled as a large multi-class classification problem. A recent review found steady improvements as demonstrated by the automated metric performance of various models [60], but there are consistent shortcomings. Firstly, most systems used end-to-end code prediction. Methods often used deep learning approaches to learn the latent relationship between episode text and associated code predictions making prediction interpretation difficult. This often then does not allow specific terms to be attributed to a predicted code, a requirement for real-world clinical coding. Secondly, there is inconsistent individual code performance reported by most publications. Clinical coding has a priority

[2] https://mlflow.org/

order, and priority codes are more important to assign correctly rather than secondary comorbidities or interventions. Thirdly, clinical coding is context specific. Human clinical coders review notes in the context of the patient episode, the patient demographics, the specific hospital department and even clinician. This informs their coding process where to look for codes and the specific coding rules to apply. This can be especially helpful for the detection of false negatives, where a code should be assigned, but notes are missing or not fully documented by clinical teams, An issue that current modelling approaches do not account for. Finally, most of the research focuses on a single ICU dataset MIMIC-III [64]. This is problematic as the clinical coding dataset has not undergone a secondary audit. In published work I argue this dataset can only be treated as 'silver standard'. I perform a detailed analysis by fine-tuning a MedCAT model for specific codes outlining areas where I believe there is undercoding. Importantly, MedCAT identities specific words or phrases from the text and links the appropriate ICD code, providing an interpretable set of predicted codes.

6.1.3 Text Summarisation in the Delivery of Care

Chapter 5 focuses on the application of summarisation methods to new areas within healthcare. I include two research papers. The first explores techniques to quantify the redundancy in clinical text, presenting methods to better understand information theoretic differences between clinical and open-domain text, and a pipeline to estimate duplication between concurrent notes of the same type. Both methods quantify sources of redundancy. Identifying and mitigating the effects of redundant text is a key area in the development of effective textual summaries.

The second paper builds extractive / abstractive text summarisation models for *brief hospital course* (BHC) summarisation. BHC sections are within discharge summary clinical notes and they summarise the entirety of an admission. This work compares extractive vs abstractive summarisation methods, evaluates a novel model that uses a clinically aware *guidance* signal within an abstractive summarisation model and how an ensemble (extractive & abstractive) model can be used to produce BHC sections. This work demonstrated consistently improved results for a ‘clinically aware’ summarisation model for two real-world clinical datasets.

6.2 Available Software and Wider Research Community Impact

All software that has contributed to the thesis has been released open-source and can be found on github. The wider CogStack ‘organisation’ houses the source-code that supports the ingestion and search capability used to train and evaluate the NER+L methods:

- CogStack-Nifi: Data ingestion pipelines for EHR structured and unstructured data using ElasticSearch as a data sink. https://github.com/CogStack/CogStack-NiFi
- MedCAT: https://github.com/CogStack/MedCAT
- MedCATtrainer: https://github.com/CogStack/MedCATtrainer

Experiment or study specific software is also made available:

- MIMIC-III Coding Analysis: https://github.com/tomolopolis/MIMIC-III-Discharge-Diagnosis-Analysis
- Redundancy Exploration: https://github.com/tomolopolis/clinical_sum
- Guided summarisation: https://github.com/tomolopolis/BHC-Summarisation

Users of the software have been far and wide. There are active users of CogStack, MedCAT and MedCATtrainer extensively within the UK at the time of writing. This data and digital infrastructure provides a path for ongoing research output.

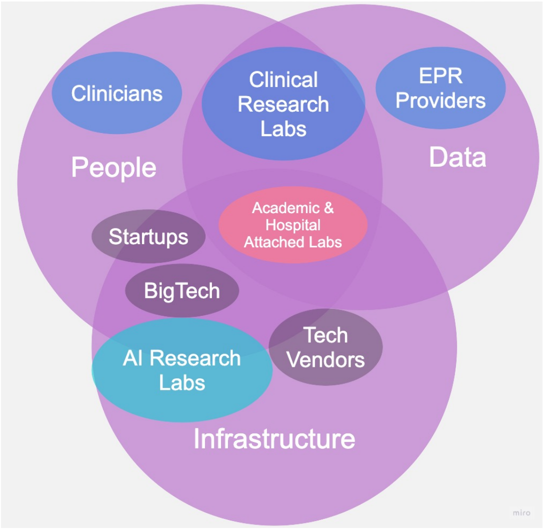


Fig. 6.1 A venn diagram of the enablers for successful research in clinical informatics

6.3 Enablers of Clinical Informatics Research

In Section 2.6, I briefly discuss the enablers of open-domain ML research. This includes the arrival of large curated datasets, software frameworks enabling accessible, iterative experimentation, and innovative use of hardware (i.e. GPUs, TPUs) to drastically speed up model training.

Reflecting on the range of research projects for this thesis and in my contributions to the wider lab's work, I can outline 3 further enablers of effective clinical informatics research with a focus on the application machine learning and often deep learning methods.

Figure 6.1 shows these components and compares the PHI data lab with alternative research and industry efforts within clinical informatics. These components are:

1. **Data**: Clinical data is highly sensitive as it describes our private health status over the course of our lives. With each encounter we leave a digital footprint about our clinical condition, the interventions and importantly the objective (lab results / observations) and subjective (how we felt) care experience. The richness of this data and its possible utility is largely unrealised, but will certainly play a role across the spectrum of possible decisions be it clinical, operational or administrative. Within the UK, health care data is decentralised, stored and maintained within each care

provider's EHR system e.g. Unfortunately, there is currently no central store of national records, that includes both structured and unstructured data. Navigating the bureaucracy to gain data access is difficult and often only available through academic partnerships between provider and attached academic partners.

2. **People**: Effective clinical informatics research requires multidisciplinary teams. This includes: clinicians and domain experts to direct and guide the research, validate findings and provide human annotations for supervised learning. Engineers or computational scientists to process and prepare data, setup and run experiments. Lastly, information governance, patient public involvement & engagement (PPIE) groups to ensure data is being used responsibly and in the interest of the patients and public.

3. **Infrastructure**: Alongside the physical hardware and software to collect and analyse data, setup and run experiments there is significant process and organisational infrastructure that enables clinical informatics research. There are often local, specific administrative processes that can bring together teams of clinical specialists on the hospital side, and engineers or data scientist researchers on the university side. Having a clear well defined process that enables team with different specialisms to come together and execute on a project within a timely manner is critical to success.

The research in this thesis has been fortunate in all 3 areas. All projects have had access to real-world clinical data, often from multiple large hospital sites with diverse populations adding sufficient weight to findings and analysis. This is largely due to the successful and ongoing deployments of the CogStack ecosystem at various NHS Hospital Trusts in London and beyond. The key development that CogStack provides is the ingestion of all EPR data in a single, real-time updated and searchable index. Without CogStack, gathering the necessary data to run large scale model training, with data potentially spread across multiple SQL databases and document management systems housing .pdf, .doc or .xlsx docs could take months of custom scripts and manual efforts.

6.4 Challenges

This section will summarise the challenges encountered throughout projects for the thesis.

6.4.1 Clinical Data Quality

Clinical data is heterogeneous with presenting clinical phenotypes hugely variable for even a single condition. Data can be scarce and shallow for patients that have uncomplicated complaints and receive routine care, whereas other patients can have broad and deep data for chronic complex issues involving many encounters. There is further heterogeneity across data formats such as pdf, word docs, and images. The OCR pipeline supplied within CogStack is able to handle most formats and extract data, however there is still a word error rate (WER) depending largely on the quality of source data, i.e. a handwritten note vs a typed pdf. This inter-patient difference was seen throughout the Covid-19 research where some patients had hundreds of pages of notes whereas as others had only couple of paragraphs.

6.4.2 CogStack Deployment Differences

CogStack is designed as a loosely coupled ecosystem of technologies to flexibly support clinical research. Hospital IT is often heterogeneous within and across Trusts. Each CogStack deployment requires a data ‘ingestion’ script to replicate data from the source system into CogStack. Setting this up can vary in difficulty depending on the volume and complexity of data to be ingested. If CogStack is yet to ingest a dataset then it must be retrieved directly from the source system, which can be difficult to locate, access and retrieve. Fortunately, in my case the majority of data used throughout this thesis was already ingested by deployed CogStack instances, or was in the process of being ingested.

CogStack is flexible and has been designed to cope with issues of missing, incomplete and messy data. However this flexibility does not enforce or even suggest a schema for ingested data. This can result in inconsistencies in deployments within and between sites. For example, equivalent data such as ‘document description’, could be called ‘doc desc’, ‘document desc’ across 3 different deployments or indices. Schema differences are simple to fix, but as analysis pipelines are often long and complex, modifying even simple things such as parameter names can be time-consuming to implement and run. Overall, CogStack deployment differences often resulted in delays in running replications of analysis across sites as subtle variations in code or edge cases needed to be fixed before results could be gathered and analysed.

6.4.3 Data Access

Throughout this thesis methods have been built to enable clinical research (e.g. MedCAT) and supplemented my own research interest of clinical text summarisation. Healthcare providers that store and maintain the data have strict information governance (IG) processes to ensure the data is used responsibly and ethically. For example, CogStack requests that supply data for clinical research questions have to undergo independent internal committee review, be sponsored by a clinical supervisor within the Trust and ensure appropriate levels of patient and public involvement and engagement (PPIE) are considered. Each hospital has their own broadly similar process with small but often significant differences. SLaM has the CRIS process, KCH the KERRI and GSTT the GERRI process. Unfortunately, despite there being a ‘National’ health service in the UK, there is not single national level agreed upon process that each Trust follows to provide access for research. This can be understood by each provider operating independently, with their own unique patient populations, clinical specialisms, technology stacks and data requirements and therefore IG processes. Navigating through IG and research administrator processes was difficult and only possible with assistance of staff placed within those Trusts.

6.4.4 Data Discovery with Clinical Datasets

A challenge acutely felt within the early part of the thesis was the lack of data model visibility at NHS Trusts. The expectation for clinical research is to specify the data requirements first then to submit a proposed project. This was difficult initially as the development and testing of novel methods required large scale data access that project review committees had not previously come across. Once various projects were framed in particular way, project review committees became easier to navigate. Systems such as CogStack allow large scale analysis that previously have not been possible.

6.4.5 Compute Restrictions

Both the PHI DataLab at KCL and the KCH NHS Trust have access to GPU compute to accelerate model training, fine-tuning and inference. For experiments involving MIMIC-III [64], the departmental local GPU compute was available for fast, interactive access. The KCH compute is a large 80 core, 8 V-100 GPU NVIDIA-DGX machine, located on the KCH network behind the Trust firewall. This machine has KCH CogStack access but has a restrictive firewall so all experiments require pre-built docker containers. This has been time consuming as the time to test a hypothesis, fix bugs and explore further ideas requires rebuilding large docker containers that contain models, software dependencies and external datasets. Software and models have to be configured or modified to operate in an 'offline' manner, allowing for container images to be built, and moved to isolated compute. As the KCH CogStack data is highly sensitive, identifiable health data there are few alternative options.

6.4.6 Annotation Collection

Modern AI methods are extremely capable pattern recognition systems. However, no model is perfect. Understanding where a model makes errors is an important part in building trust with model users. MedCAT uses human domain expert collected annotations for model validation, and model fine-tuning via supervised training. Collecting manual annotations is labour intensive and potentially error prone. Finding motivated domain experts who are prepared to manually annotate data can be difficult. MedCAT and MedCATtrainer attempt to provide easy to use interfaces, suggesting potential annotations, keyboard shortcuts, and fast concept look-ups but there is still a learning curve to use the tools effectively. Ensuring annotators are able to consistently annotate to the same standard is crucial, as I have found to my detriment, with previous project annotation exercises yielding little value despite annotators spending hours annotating. Although annotation guidelines are provided[3] and strongly recommended ensuring human annotators adhere to them is difficult.

Overall, the challenges encountered were overcome, or mitigated through perseverance and the expert guidance from clinical, technical and NHS Trust process experts.

6.5 Limitations

Overall, the impact of this work is limited to healthcare settings where firstly the data is digitally available. Paper records could not be used in the methods shown throughout this thesis. Fortunately, converting paper records to electronic form can be fairly easily done at a reasonable expense, although there are errors associated with the conversion, particularly in converting handwritten text. Secondly, all methods have focused on English language text only. Through the open-source code bases there has been engagement across Europe and Asia where the clinical text is not English, but we have only seen initial results to suggest the results are precisely replicable across languages [36]. Another limitation is that our methods are unimodal, and only use the unstructured portion of the EHR. EHRs are rich multi-modal sources of data, containing radiology scan images and structured laboratory test results / demographic data that could be used for better model predictive power. Beyond the EHR, there is a wealth of relevant data that could be incorporated into

[3] http://shorturl.at/hmy78

learning systems integrated into specific healthcare scenarios, these include wearable and mobile device, social media and genomic data.

6.6 Future Work and the Challenges Ahead

6.6.1 The CogStack Ecosystem

At the time of writing CogStack is an open-source ecosystem of solutions that includes the MedCAT and MedCATtrainer tools developed as part of this thesis. CogStack is owned and maintained by the core group in the PHIDL, with further contributions from individuals attached to hospitals or universities based in UK, Australia and the Netherlands. This 'organic' growth has resulted in deployments of various part of the ecosystem across˜15 UK based hospitals and international healthcare providers / research organisations. Future work in this area will look to build further clinical informatics research outputs but also address shortcomings of the academic software into a well documented and supported industrial solution. A reason for its successes so far are:

- The components are loosely coupled. As previously mentioned, NHS Trusts can greatly vary in EPR deployments, digital maturity etc. CogStack provides an end-to-end solution for indexing, searching, visualising and structuring EHR data but each component can also be used independently. So for example, if a site already has a single data-source for all clinical data MedCAT can be used in isolation.

- The software is open-source. Open-source software has steadily grown in popularity over the last decade with many businesses small and large choosing to 'download' rather than 'build' or 'buy'. Open-source has allowed potential users of CogStack to investigate, deploy and trial the technology for their own use cases without securing funding to purchase the 'product'. Future work will continue to push for open-source to continue to maximise its availability and potential for impact.

- CogStack has grown organically within academia and NHS Trusts. This growth has meant that progress has been slower (6+ years) than a dedicated, built in silo product. However, CogStack is now a trusted, identifiable brand with brand equity in the close-knit clinical informatics community. This has been built by continual delivery of peer-reviewed published research and delivery of service improvement / audit projects at many healthcare providers across the UK and internationally.

I have shown that NER+L developed in Chapter 3 has uses in downstream clinical research (Section 3.4.2) and various summarisation use cases (Chapter 4 and 5). The continued investment in the CogStack ecosystem will productionise the models used for these studies, enabling re-use, model discovery and sharing. This is largely the discipline of ML Operations. An active research and industry area that supports how a machine learning model can be continually monitored and controlled to ensure safety, ongoing validation and governance.

6.6.2 CogStack to ‘Unlock’ EHR Data

Overall, CogStack aims to ‘unlock’ data already held within an EHR. Healthcare providers are continuing to invest in large expensive EPR deployments to manage and directly administer care. However, EPR providers often lack the capabilities that CogStack provides and are often not incentivised to build out these capabilities. Namely, to search and visualise data in real-time, or structure and prepare data rapidly for downstream uses such as clinical research, decision support and predictive modelling. This unlocking of the data will be a huge leap forward in what is capable, but it will only be the start. CogStack deployments at a given site ideally will allow any timely, relevant, structured data (regardless of source system) ready for any conceivable downstream use.

6.7 Conclusions

To conclude, this thesis has presented, applied and evaluated methods to summarise data within EHRs. The methods for extracting and contextualising clinical terms from any clinical terminology has been evaluated across multiple healthcare settings, fine-tuned for specific subsets of terms and contributed to research studies across specialties, clinical areas and even use cases i.e. clinical, operational and administrative. The thesis has highlighted the relevance of summarisation of EHR data across the range of healthcare roles. For example, front line staff summarising patient narratives for care team or setting transitions. For administrative staff that summarise episodes and assign clinical codes for billing and reporting, and for researchers and innovators looking to build decision support tools and predictive models that summarise the raw clinical notes by extracting patient or episode level terms for further processing.

Healthcare is a data rich domain, and this thesis has only considered EHR data, focusing on the often untouched unstructured portion of the record. The work has reinforced and furthered our abilities to realise value locked away within the unstructured or patient narrative part of the EHR. I have critically looked at current research and built tools that have utility across areas of healthcare delivery. The open-source release of tools has enabled dissemination as users have already by using the tools in downstream healthcare use cases. However, I have only scratched the surface of what is possible. Structuring decades of historical data could drastically improve clinical, operational and administrative use cases in the short to medium term. Many research questions that have simply not been possible, due to a lack of data and expertise in analysis could be answered by enabling researchers to rapidly ask questions of the data.

Looking ahead, I foresee systems using multi-modal approaches integrating multiple data sources, supporting healthcare workers and even patients themselves to make timely, well-informed decisions. These multi-modal methods would still include data from the patient narrative and NLP methods, but also speech from both clinicians and patients, image data, e.g. radiology scans, multi-omics data, e.g. genomics, proteomics, transcriptomics etc., and newer data sources such as mobile health sensors providing continual monitoring of patients outside of hospital settings.

Overall, I believe increased data and precise, careful, considered application of AI methods in healthcare will lead to improvements across many aspects of healthcare delivery. Allowing clinicians, researchers, operations and administrative personal to augment their current capabilities beyond what is currently possible. Importantly, this will not replace any of these roles but allow the human user of the AI to perform tasks at greater speed and efficiency applying expert human guidance where necessary. I hope one day future PhD and academic research projects will continue to push the envelope, using the contributions in this work and beyond, building upon our methodologies and experiments, implementing the currently experimental solutions into real-world practise; using the wealth of EHR data work for the patient, clinician and system as a whole. For example, our work could be used to improve patient care by identifying and extracting structured data allowing computable guidelines to be used and advise clinicians through electronic clinical decision support, or building a system where the clinician's experience is improved by reducing the friction in EHR system usage through the automated generation of text, or even improving clinical coding depth and breadth of coding improving care planning and financial remuneration for a provider. Ultimately, I hope this work in some small way moves global healthcare to a more equitable, efficient and available service.

References

[1] Griffin Adams et al. “What’s in a Summary? Laying the Groundwork for Advances in Hospital-Course Summarization”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 4794–4811. DOI: 10.18653/v1/2021.naacl-main.382.

[2] Stergos Afantenos, Vangelis Karkaletsis, and Panagiotis Stamatopoulos. “Summarization from medical documents: a survey”. en. In: *Artif. Intell. Med.* 33.2 (Feb. 2005), pp. 157–177. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2004.07.017.

[3] Tarun Agrawal and Prakash Choudhary. “Segmentation and classification on chest radiography: a systematic survey”. en. In: *Vis. Comput.* (Jan. 2022), pp. 1–39. ISSN: 0178-2789. DOI: 10.1007/s00371-021-02352-7.

[4] Alan Akbik, Duncan Blythe, and Roland Vollgraf. “Contextual String Embeddings for Sequence Labeling”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649.

[5] Wael Abdulrahman Almurashi. “An Introduction to Halliday’s Systemic Functional Linguistics”. In: *Journal for the Study of English Linguistics* 4.1 (May 2016), pp. 70–80. ISSN: 2329-7034, 2329-7034. DOI: 10.5296/jsel.v4i1.9423.

[6] Alan R Aronson and François-Michel Lang. “An overview of MetaMap: historical perspective and recent advances”. In: *J. Am. Med. Inform. Assoc.* 17.3 (2010), pp. 229–236. ISSN: 1067-5027. DOI: 10.1136/jamia.2009.002733.

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer Normalization”. In: (July 2016). arXiv: 1607.06450 `[stat.ML]`.

[8] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *3rd International Conference on Learning Representations, ICLR 2015*. Jan. 2015.

[9] Caroline Bassett. “The computational therapeutic: exploring Weizenbaum’s ELIZA as a history of the present”. In: *AI Soc.* 34.4 (Dec. 2019), pp. 803–812. ISSN: 0951-5666, 1435-5655. DOI: 10.1007/s00146-018-0825-9.

[10] Tal Baumel et al. “Multi-Label Classification of Patient Notes: Case Study on ICD Code Assignment”. en. In: *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*. June 2018.

[11] Daniel M Bean et al. “ACE-inhibitors and Angiotensin-2 Receptor Blockers are not associated with severe SARS-COVID19 infection in a multi-site UK acute Hospital Trust”. In: *Eur. J. Heart Fail.* (June 2020). ISSN: 1388-9842. DOI: 10.1002/ejhf.1924.

[12] Rebecca Bendayan et al. “Mapping multimorbidity in individuals with schizophrenia and bipolar disorders: evidence from the South London and Maudsley NHS Foundation Trust Biomedical Research Centre (SLAM BRC) case register”. en. In: *BMJ Open* 12.1 (Jan. 2022), e054414. ISSN: 2044-6055. DOI: 10.1136/bmjopen-2021-054414.

[13] Emily M Bender and Alexander Koller. “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5185–5198. DOI: 10.18653/v1/2020.acl-main.463.

[14] Fredric Evan Blavin and Melinda Beeuwkes Buntin. “Forecasting the use of electronic health records: an expert opinion approach”. en. In: *Medicare Medicaid Res. Rev.* 3.2 (Apr. 2013). ISSN: 2159-0354. DOI: 10.5600/mmrr.003.02.a02.

[15] Richard A Blythe and William Croft. “How individuals change language”. en. In: *PLoS One* 16.6 (June 2021), e0252582. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0252582.

[16] Oliver Bodenreider, Ronald Cornet, and Daniel J Vreeman. “Recent Developments in Clinical Terminologies - SNOMED CT, LOINC, and RxNorm”. en. In: *Yearb. Med. Inform.* 27.1 (Aug. 2018), pp. 129–139. ISSN: 0943-4747, 2364-0502. DOI: 10.1055/s-0038-1667077.

[17] Olivier Bodenreider. “The Unified Medical Language System (UMLS): integrating biomedical terminology”. In: *Nucleic Acids Res.* 32.Database issue (Jan. 2004), pp. D267–70. ISSN: 0305-1048. DOI: 10.1093/nar/gkh061.

[18] Sebastian Bodenstedt et al. “Artificial Intelligence-Assisted Surgery: Potential and Challenges”. en. In: *Visc Med* 36.6 (Dec. 2020), pp. 450–455. ISSN: 2297-4725. DOI: 10.1159/000511351.

[19] Piotr Bojanowski et al. “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5 (Dec. 2017), pp. 135–146. DOI: 10.1162/tacl_a_00051.

[20] Rishi Bommasani et al. “On the Opportunities and Risks of Foundation Models”. In: (Aug. 2021). arXiv: 2108.07258 [cs.LG].

[21] Tom Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[22] Alex A T Bui, Denise R Aberle, and Hooshang Kangarloo. “TimeLine: visualizing integrated patient records”. en. In: *IEEE Trans. Inf. Technol. Biomed.* 11.4 (July 2007), pp. 462–473. ISSN: 1089-7771. DOI: 10.1109/titb.2006.884365.

[23] Sharon Campbell and Katrina Giadresco. “Computer-assisted clinical coding: A narrative review of the literature on its benefits, limitations, implementation and impact on clinical coding professionals”. en. In: *Health Inf. Manag.* 49.1 (Jan. 2020), pp. 5–18. ISSN: 1322-4913. DOI: 10.1177/1833358319851305.

[24] Pengfei Cao et al. “HyperCore: Hyperbolic and Co-graph Representation for Automatic ICD Coding”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 3105–3114. DOI: 10.18653/v1/2020.acl-main.282.

[25] Ewan Carr et al. “Evaluation and improvement of the National Early Warning Score (NEWS2) for COVID-19: a multi-hospital study”. en. In: *BMC Med.* 19.1 (Jan. 2021), p. 23. ISSN: 1741-7015. DOI: 10.1186/s12916-020-01893-3.

[26] Jane M Carrington and Judith A Effken. “Strengths and limitations of the electronic health record for documenting clinical events”. en. In: *Comput. Inform. Nurs.* 29.6 (June 2011), pp. 360–367. ISSN: 1538-2931, 1538-9774. DOI: 10.1097/NCN.0b013e3181fc4139.

[27] Joan A Casey et al. “Using Electronic Health Records for Population Health Research: A Review of Methods and Applications”. en. In: *Annu. Rev. Public Health* 37 (2016), pp. 61–81. ISSN: 0163-7525, 1545-2093. DOI: 10.1146/annurev-publhealth-032315-021353.

[28] Hye Yoon Chang et al. “Artificial Intelligence in Pathology”. en. In: *J Pathol Transl Med* 53.1 (Jan. 2019), pp. 1–12. ISSN: 2383-7837. DOI: 10.4132/jptm.2018.12.16.

[29] Kyunghyun Cho et al. “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012.

[30] Morten H Christiansen and Simon Kirby. *Language Evolution*. en. OUP Oxford, July 2003. ISBN: 9780191581663.

[31] Songtai Dai et al. “BDKG at MEDIQA 2021: System Report for the Radiology Report Summarization Task”. In: *Proceedings of the 20th Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, June 2021, pp. 103–111. DOI: 10.18653/v1/2021.bionlp-1.11.

[32] Ashwin Devaraj et al. “Paragraph-level Simplification of Medical Texts”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 4972–4984. DOI: 10.18653/v1/2021.naacl-main.395.

[33] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[34] C Drummond and A Simpson. “’Who’s actually gonna read this?’ An evaluation of staff experiences of the value of information contained in written care plans in supporting care in three different dementia care settings”. en. In: *J. Psychiatr. Ment. Health Nurs.* 24.6 (Aug. 2017), pp. 377–386. ISSN: 1351-0126, 1365-2850. DOI: 10.1111/jpm.12380.

[35] Ben Eisner et al. “emoji2vec: Learning Emoji Representations from their Description”. In: *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*. Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 48–54. DOI: 10.18653/v1/W16-6208.

[36] Bram van Es et al. "Negation detection in Dutch clinical texts: an evaluation of rule-based and machine learning methods". en. In: *BMC Bioinformatics* 24.1 (Jan. 2023), p. 10. ISSN: 1471-2105. DOI: 10.1186/s12859-022-05130-X.

[37] R S Evans. "Electronic Health Records: Then, Now, and in the Future". en. In: *Yearb. Med. Inform.* Suppl 1 (May 2016), S48–61. ISSN: 0943-4747, 2364-0502. DOI: 10.15265/IYS-2016-s006.

[38] Alexander Fabbri et al. "Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1074–1084. DOI: 10.18653/v1/P19-1102.

[39] J R Firth. "A synopsis of linguistic theory, 1930-1955". In: *Studies in Linguistic Analysis* (1957).

[40] Ragnar Fjelland. "Why general artificial intelligence will not be realized". en. In: *Humanities and Social Sciences Communications* 7.1 (June 2020), pp. 1–9. ISSN: 2662-9992, 2662-9992. DOI: 10.1057/s41599-020-0494-4.

[41] Marta Garnelo and Murray Shanahan. "Reconciling deep learning with symbolic artificial intelligence: representing objects and relations". In: *Current Opinion in Behavioral Sciences* 29 (Oct. 2019), pp. 17–23. ISSN: 2352-1546. DOI: 10.1016/j.cobeha.2018.12.010.

[42] Reza Ghaeini, Xiaoli Fern, and Prasad Tadepalli. "Interpreting Recurrent and Attention-Based Neural Models: a Case Study on Natural Language Inference". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4952–4957. DOI: 10.18653/v1/D18-1537.

[43] Ayelet Goldstein and Yuval Shahar. "Generation of Natural-Language Textual Summaries from Longitudinal Clinical Records". en. In: *Stud. Health Technol. Inform.* 216 (2015), pp. 594–598. ISSN: 0926-9630, 1879-8365.

[44] S Gorn, R W Bemer, and J Green. *American standard code for information interchange*. 1963. DOI: 10.1145/366707.367524.

[45] Richard Grol and Jeremy Grimshaw. "From best evidence to best practice: effective implementation of change in patients' care". en. In: *Lancet* 362.9391 (Oct. 2003), pp. 1225–1230. ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(03)14546-1.

[46] Yohan Bonescki Gumiel et al. "Temporal Relation Extraction in Clinical Texts: A Systematic Review". In: *ACM Comput. Surv.* 54.7 (Sept. 2021), pp. 1–36. ISSN: 0360-0300. DOI: 10.1145/3462475.

[47] Luis Gutiérrez and Brian Keith. "A Systematic Literature Review on Word Embeddings". In: *Trends and Applications in Software Engineering*. Springer International Publishing, 2019, pp. 132–141. DOI: 10.1007/978-3-030-01171-0_12.

[48] Zellig S Harris. "Distributional Structure". In: *Word World* 10.2-3 (Aug. 1954), pp. 146–162. ISSN: 0275-5270, 0043-7956. DOI: 10.1080/00437956.1954.11659520.

[49] Kristiina Häyrinen, Kaija Saranto, and Pirkko Nykänen. “Definition, structure, content, use and impacts of electronic health records: a review of the research literature”. en. In: *Int. J. Med. Inform.* 77.5 (May 2008), pp. 291–304. ISSN: 1386-5056. DOI: 10.1016/j.ijmedinf.2007.09.001.

[50] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[51] Jamie S Hirsch et al. “HARVEST, a longitudinal patient record summarizer”. In: *J. Am. Med. Inform. Assoc.* 22.2 (2014), pp. 263–274. ISSN: 1067-5027.

[52] S Hochreiter and J Schmidhuber. “Long short-term memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667.

[53] Arthur E Hoerl and Robert W Kennard. “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12.1 (Feb. 1970), pp. 55–67. ISSN: 0040-1706. DOI: 10.1080/00401706.1970.10488634.

[54] A Jay Holmgren et al. “Assessment of Electronic Health Record Use Between US and Non-US Health Systems”. en. In: *JAMA Intern. Med.* 181.2 (Feb. 2021), pp. 251–259. ISSN: 2168-6106, 2168-6114. DOI: 10.1001/jamainternmed.2020.7071.

[55] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural Netw.* 2.5 (Jan. 1989), pp. 359–366. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8.

[56] Gretchen M Hultman et al. “Challenges and Opportunities to Improve the Clinician Experience Reviewing Electronic Progress Notes”. en. In: *Appl. Clin. Inform.* 10.3 (May 2019), pp. 446–453. ISSN: 1869-0327. DOI: 10.1055/s-0039-1692164.

[57] Richard Jackson et al. “CogStack - experiences of deploying integrated information retrieval and extraction services in a large National Health Service Foundation Trust hospital”. In: *BMC Med. Inform. Decis. Mak.* 18.1 (June 2018), p. 47. ISSN: 1472-6947. DOI: 10.1186/s12911-018-0623-9.

[58] Marijn Janssen et al. “Competencies to promote collaboration between primary and secondary care doctors: an integrative review”. en. In: *BMC Fam. Pract.* 21.1 (Sept. 2020), p. 179. ISSN: 1471-2296. DOI: 10.1186/s12875-020-01234-6.

[59] Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. “Does the magic of BERT apply to medical code assignment? A quantitative study”. en. In: *Comput. Biol. Med.* 139 (Dec. 2021), p. 104998. ISSN: 0010-4825, 1879-0534. DOI: 10.1016/j.compbiomed.2021.104998.

[60] Shaoxiong Ji et al. “A Unified Review of Deep Learning for Automated Medical Coding”. In: (Jan. 2022). arXiv: 2201.02797 [cs.CL].

[61] Zongcheng Ji, Qiang Wei, and Hua Xu. “BERT-based Ranking for Biomedical Entity Normalization”. en. In: *AMIA Jt Summits Transl Sci Proc* 2020 (May 2020), pp. 269–277. ISSN: 2153-4063.

[62] Chao Jiang et al. “Learning Word Embeddings for Low-Resource Languages by PU Learning”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1024–1034. DOI: 10.18653/v1/N18-1093.

[63] Johnson. “Lexical facts”. en. In: *The Economist* (May 2013). ISSN: 0013-0613.

[64] Alistair E W Johnson et al. “MIMIC-III, a freely accessible critical care database”. In: *Sci Data* 3 (May 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35.

[65] Danai Khemasuwan, Jeffrey S Sorensen, and Henri G Colt. “Artificial intelligence in pulmonary medicine: computer vision, predictive model and COVID-19”. en. In: *Eur. Respir. Rev.* 29.157 (Sept. 2020). ISSN: 0905-9180, 1600-0617. DOI: 10.1183/16000617.0181-2020.

[66] Ellen Kim et al. “The Evolving Use of Electronic Health Records (EHR) for Research”. en. In: *Semin. Radiat. Oncol.* 29.4 (Oct. 2019), pp. 354–361. ISSN: 1053-4296, 1532-9461. DOI: 10.1016/j.semradonc.2019.05.010.

[67] Abirami Kirubarajan et al. “Artificial intelligence in emergency medicine: A scoping review”. en. In: *J Am Coll Emerg Physicians Open* 1.6 (Dec. 2020), pp. 1691–1702. ISSN: 2688-1152. DOI: 10.1002/emp2.12277.

[68] Kleene. “Representation of events in nerve nets and finite automata”. In: *Automata studies* ().

[69] Sebastian Köhler et al. “The Human Phenotype Ontology in 2021”. en. In: *Nucleic Acids Res.* 49.D1 (Jan. 2021), pp. D1207–D1217. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkaa1043.

[70] John F Kolen and Stefan C Kremer. “Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies”. In: *A Field Guide to Dynamical Recurrent Networks*. IEEE, 2001, pp. 237–243. ISBN: 9780470544037. DOI: 10.1109/9780470544037.ch14.

[71] Ina Kostakis et al. “The performance of the National Early Warning Score and National Early Warning Score 2 in hospitalised patients infected by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)”. en. In: *Resuscitation* 159 (Feb. 2021), pp. 150–157. ISSN: 0300-9572, 1873-1570. DOI: 10.1016/j.resuscitation.2020.10.039.

[72] Zeljko Kraljevic et al. “Multi-domain clinical natural language processing with MedCAT: The Medical Concept Annotation Toolkit”. In: *Artif. Intell. Med.* 117 (July 2021), p. 102083. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2021.102083.

[73] Kory Kreimeyer et al. “Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review”. en. In: *J. Biomed. Inform.* 73 (Sept. 2017), pp. 14–29. ISSN: 1532-0464, 1532-0480. DOI: 10.1016/j.jbi.2017.07.012.

[74] Paul T Kröner et al. “Artificial intelligence in gastroenterology: A state-of-the-art review”. en. In: *World J. Gastroenterol.* 27.40 (Oct. 2021), pp. 6794–6824. ISSN: 1007-9327, 2219-2840. DOI: 10.3748/wjg.v27.i40.6794.

[75] Taku Kudo and John Richardson. “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012.

[76] Jacqueline K Kueper et al. “Artificial Intelligence and Primary Care Research: A Scoping Review”. en. In: *Ann. Fam. Med.* 18.3 (May 2020), pp. 250–258. ISSN: 1544-1709, 1544-1717. DOI: 10.1370/afm.2518.

[77] Curtis P Langlotz. “RadLex: a new method for indexing online educational materials”. en. In: *Radiographics* 26.6 (Nov. 2006), pp. 1595–1597. ISSN: 0271-5333, 1527-1323. DOI: 10.1148/rg.266065168.

[78] Quoc Le and Tomas Mikolov. “Distributed Representations of Sentences and Documents”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research. Bejing, China: PMLR, 2014, pp. 1188–1196.

[79] Myeong-Seon Lee and Seonah Lee. “Implementation of an Electronic Nursing Record for Nursing Documentation and Communication of Patient Care Information in a Tertiary Teaching Hospital”. en. In: *Comput. Inform. Nurs.* 39.3 (July 2020), pp. 136–144. ISSN: 1538-2931, 1538-9774. DOI: 10.1097/CIN.0000000000000642.

[80] Jennifer Liang, Ching-Huei Tsou, and Ananya Poddar. “A Novel System for Extractive Clinical Note Summarization using EHR Data”. In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 46–54. DOI: 10.18653/v1/W19-1906.

[81] Chin-Yew Lin. “Rouge: A package for automatic evaluation of summaries”. In: *Text summarization branches out*. 2004, pp. 74–81.

[82] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: (July 2019). arXiv: 1907.11692 `[cs.CL]`.

[83] Francisco Lopez-Jimenez et al. “Artificial Intelligence in Cardiology: Present and Future”. en. In: *Mayo Clin. Proc.* 95.5 (May 2020), pp. 1015–1039. ISSN: 0025-6196, 1942-5546. DOI: 10.1016/j.mayocp.2020.01.038.

[84] Pilar López-Úbeda et al. “Natural Language Processing in Radiology: Update on Clinical Applications”. en. In: *J. Am. Coll. Radiol.* 19.11 (Nov. 2022), pp. 1271–1285. ISSN: 1546-1440, 1558-349X. DOI: 10.1016/j.jacr.2022.06.016.

[85] H P Luhn. “A Statistical Approach to Mechanized Encoding and Searching of Literary Information”. In: *IBM J. Res. Dev.* 1.4 (Oct. 1957), pp. 309–317. ISSN: 0018-8646. DOI: 10.1147/rd.14.0309.

[86] Marcinkiewicz. “Building a large annotated corpus of English: The Penn Treebank”. In: *Using Large Corpora* ().

[87] Martın Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[88] Josephine Mayer, Christopher Kipps, and Hannah R Cock. “Implementing clinical guidelines”. en. In: *Pract. Neurol.* 19.6 (Dec. 2019), pp. 529–535. ISSN: 1474-7758, 1474-7766. DOI: 10.1136/practneurol-2017-001814.

[89] Brian McMillan et al. “Primary Care Patient Records in the United Kingdom: Past, Present, and Future Research Priorities”. en. In: *J. Med. Internet Res.* 20.12 (Dec. 2018), e11293. ISSN: 1439-4456, 1438-8871. DOI: 10.2196/11293.

[90] Eugenia McPeek-Hinz et al. “Clinician Burnout Associated With Sex, Clinician Type, Work Culture, and Use of Electronic Health Records”. en. In: *JAMA Netw Open* 4.4 (Apr. 2021), e215686. ISSN: 2574-3805. DOI: 10.1001/jamanetworkopen.2021.5686.

[91] Sharon Mickan, Amanda Burls, and Paul Glasziou. “Patterns of ’leakage’ in the utilisation of clinical guidelines: a systematic review”. en. In: *Postgrad. Med. J.* 87.1032 (Oct. 2011), pp. 670–679. ISSN: 0032-5473, 1469-0756. DOI: 10.1136/pgmj.2010.116012.

[92] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’13. Lake Tahoe, Nevada: Curran Associates Inc., Dec. 2013, pp. 3111–3119.

[93] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *In Proc. ICLR*. 2013.

[94] Rashmi Mishra et al. “Text summarization in the biomedical domain: a systematic review of recent research”. en. In: *J. Biomed. Inform.* 52 (Dec. 2014), pp. 457–467. ISSN: 1532-0464, 1532-0480. DOI: 10.1016/j.jbi.2014.06.009.

[95] Isabel Moreno-Sánchez, Francesc Font-Clos, and Álvaro Corral. “Large-Scale Analysis of Zipf’s Law in English Texts”. en. In: *PLoS One* 11.1 (Jan. 2016), e0147073. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0147073.

[96] Ali Mozayan et al. “Practical Guide to Natural Language Processing for Radiology”. en. In: *Radiographics* 41.5 (Sept. 2021), pp. 1446–1453. ISSN: 0271-5333, 1527-1323. DOI: 10.1148/rg.2021200113.

[97] James Mullenbach et al. *Explainable Prediction of Medical Codes from Clinical Text*. 2018. DOI: 10.18653/v1/n18-1100.

[98] Thomas Müller et al. “Joint Lemmatization and Morphological Tagging with Lemming”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 2268–2274. DOI: 10.18653/v1/D15-1272.

[99] Travis B Murdoch and Allan S Detsky. “The inevitable application of big data to health care”. en. In: *JAMA* 309.13 (Apr. 2013), pp. 1351–1352. ISSN: 0098-7484, 1538-3598. DOI: 10.1001/jama.2013.393.

[100] Naveen S Pagad and N Pradeep. “Clinical Named Entity Recognition Methods: An Overview”. In: *International Conference on Innovative Computing and Communications*. Springer Singapore, 2022, pp. 151–165. DOI: 10.1007/978-981-16-2597-8_13.

[101] Laura Mk Pannell and Jonathan Tyrrell-Price. “Communication between primary and secondary care”. en. In: *Br. J. Hosp. Med.* 78.8 (Aug. 2017), pp. 464–466. ISSN: 1750-8460. DOI: 10.12968/hmed.2017.78.8.464.

[102] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[103] Urvish K Patel et al. “Artificial intelligence as an emerging technology in the current care of neurological disorders”. en. In: *J. Neurol.* 268.5 (May 2021), pp. 1623–1642. ISSN: 0340-5354, 1432-1459. DOI: 10.1007/s00415-019-09518-3.

[104] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[105] Matthew E Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202.

[106] Maxime Peyrard. “A Simple Theoretical Model of Importance for Summarization”. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 2019, pp. 1059–1073.

[107] Rimma Pivovarov and Noémie Elhadad. “Automated methods for the summarization of electronic health records”. en. In: *J. Am. Med. Inform. Assoc.* 22.5 (Sept. 2015), pp. 938–947. ISSN: 1067-5027, 1527-974X. DOI: 10.1093/jamia/ocv032.

[108] M F Porter. “An algorithm for suffix stripping”. In: *Programmirovanie* 14.3 (Jan. 1980), pp. 130–137. ISSN: 0132-3474, 0033-0337. DOI: 10.1108/eb046814.

[109] Barbara Postle, Nadia Koeldnik, and Tanya Miocevich. “The Coding Conundrum: A Workplace Perspective”. en. In: *Health Inf. Manag.* 38.1 (Mar. 2009), pp. 47–49. ISSN: 1322-4913. DOI: 10.1177/183335830903800106.

[110] Janet T Powell and Anders Wanhainen. “Analysis of the Differences Between the ESVS 2019 and NICE 2020 Guidelines for Abdominal Aortic Aneurysm”. en. In: *Eur. J. Vasc. Endovasc. Surg.* 60.1 (July 2020), pp. 7–15. ISSN: 1078-5884, 1532-2165. DOI: 10.1016/j.ejvs.2020.04.038.

[111] G Pratt and T C Morris. “Review of the NICE guidelines for multiple myeloma”. en. In: *Int. J. Lab. Hematol.* 39.1 (Feb. 2017), pp. 3–13. ISSN: 1751-5521, 1751-553X. DOI: 10.1111/ijlh.12581.

[112] T Q Qian, S J Zhu, and Y Hoshida. “Use of big data in drug development for precision medicine: an update”. en. In: *Expert Review of Precision Medicine and Drug Development* 4.3 (May 2019), pp. 189–200. ISSN: 2380-8993. DOI: 10.1080/23808993.2019.1617632.

[113] Fahad Quhal and Christian Seitz. “Guideline of the guidelines: urolithiasis”. en. In: *Curr. Opin. Urol.* 31.2 (Mar. 2021), pp. 125–129. ISSN: 0963-0643, 1473-6586. DOI: 10.1097/MOU.0000000000000855.

[114] A Radford et al. “Language Models are Unsupervised Multitask Learners”. In: 2019.

[115] Alec Radford and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training”. In: OpenAI, 2018.

[116] Pranav Rajpurkar, Robin Jia, and Percy Liang. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 784–789. DOI: 10.18653/v1/P18-2124.

[117] Pranav Rajpurkar et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264.

[118] Laila Rasmy et al. “Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction”. en. In: *NPJ Digit Med* 4.1 (May 2021), p. 86. ISSN: 2398-6352. DOI: 10.1038/s41746-021-00455-y.

[119] Beth A Reid et al. “Best practice in the management of clinical coding services: Insights from a project in the Republic of Ireland, Part 2”. en. In: *Health Inf. Manag.* 46.3 (Sept. 2017), pp. 105–112. ISSN: 1322-4913. DOI: 10.1177/1833358317697470.

[120] Mohammad Hosein Rezazade Mehrizi, Peter van Ooijen, and Milou Homan. “Applications of artificial intelligence (AI) in diagnostic radiology: a technography study”. en. In: *Eur. Radiol.* 31.4 (Apr. 2021), pp. 1805–1811. ISSN: 0938-7994, 1432-1084. DOI: 10.1007/s00330-020-07230-9.

[121] Paul M Robelia et al. “Information Transfer and the Hospital Discharge Summary: National Primary Care Provider Perspectives of Challenges and Opportunities”. en. In: *J. Am. Board Fam. Med.* 30.6 (Nov. 2017), pp. 758–765. ISSN: 1557-2625, 1558-7118. DOI: 10.3122/jabfm.2017.06.170194.

[122] Kirk Roberts, Amos Cahan, and Dina Demner-Fushman. “Error Propagation in EHRs via Copy/Paste: An Analysis of Relative Dates”. In: *AMIA*. 2014.

[123] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. “A primer in BERTology: What we know about how BERT works”. en. In: *Trans. Assoc. Comput. Linguist.* 8 (Dec. 2020), pp. 842–866. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00349.

[124] M K Ross, W Wei, and L Ohno-Machado. ““Big data” and the electronic health record”. en. In: *Yearb. Med. Inform.* 9 (Aug. 2014), pp. 97–104. ISSN: 0943-4747, 2364-0502. DOI: 10.15265/IY-2014-0003.

[125] Patrick Ruch et al. “Automatic medical encoding with SNOMED categories”. en. In: *BMC Med. Inform. Decis. Mak.* 8 Suppl 1 (Oct. 2008), S6. ISSN: 1472-6947. DOI: 10.1186/1472-6947-8-S1-S6.

[126] Nithya Sambasivan et al. ““Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–15. ISBN: 9781450380966. DOI: 10.1145/3411764.3445518.

[127] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: (Oct. 2019). arXiv: 1910.01108 [cs.CL].

[128] Fadil Santosa and William W Symes. “Linear Inversion of Band-Limited Reflection Seismograms”. In: *SIAM J. Sci. and Stat. Comput.* 7.4 (Oct. 1986), pp. 1307–1330. ISSN: 0196-5204. DOI: 10.1137/0907087.

[129] Max Savery et al. “Question-driven summarization of answers to consumer health questions”. en. In: *Sci Data* 7.1 (Oct. 2020), p. 322. ISSN: 2052-4463. DOI: 10.1038/s41597-020-00667-z.

[130] Guergana K Savova et al. “Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications”. In: *J. Am. Med. Inform. Assoc.* 17.5 (2010), pp. 507–513. ISSN: 1067-5027. DOI: 10.1136/jamia.2009.001560.

[131] Natalie Schluter. “The limits of automatic summarisation according to ROUGE”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, 2017. DOI: 10.18653/v1/e17-2007.

[132] M Schuster and K K Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Trans. Signal Process.* 45.11 (Nov. 1997), pp. 2673–2681. ISSN: 1053-587X, 1941-0476. DOI: 10.1109/78.650093.

[133] Mike Schuster and Kaisuke Nakajima. “Japanese and Korean voice search”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2012, pp. 5149–5152. DOI: 10.1109/ICASSP.2012.6289079.

[134] Thomas Searle, Zina Ibrahim, and Richard Dobson. “Comparing Natural Language Processing Techniques for Alzheimer’s Dementia Prediction in Spontaneous Speech”. In: *Proceedings of INTERSPEECH 2020*. June 2020.

[135] Thomas Searle et al. “Estimating redundancy in clinical text”. en. In: *J. Biomed. Inform.* 124 (Dec. 2021), p. 103938. ISSN: 1532-0464, 1532-0480. DOI: 10.1016/j.jbi.2021.103938.

[136] Thomas Searle et al. “MedCATTrainer: A Biomedical Free Text Annotation Interface with Active Learning and Research Use Case Specific Customisation”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 139–144. DOI: 10.18653/v1/D19-3024.

[137] Elena Sergeeva et al. “Negation Scope Detection in Clinical Notes and Scientific Abstracts: A Feature-enriched LSTM-based Approach”. en. In: *AMIA Jt Summits Transl Sci Proc* 2019 (May 2019), pp. 212–221. ISSN: 2153-4063.

[138] Sofia Serrano and Noah A Smith. “Is Attention Interpretable?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2931–2951. DOI: 10.18653/v1/P19-1282.

[139] Gro-Hilde Severinsen et al. “From Free-Text to Structure in Electronic Patient Records”. en. In: *Stud. Health Technol. Inform.* 265 (Aug. 2019), pp. 86–91. ISSN: 0926-9630, 1879-8365. DOI: 10.3233/SHTI190143.

[140] Ori Shapira et al. “Interactive abstractive summarization for event news tweets”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2017, pp. 109–114.

[141] Anthony Shek et al. “Machine learning-enabled multitrust audit of stroke comorbidities using natural language processing”. en. In: *Eur. J. Neurol.* 28.12 (Dec. 2021), pp. 4090–4097. ISSN: 1351-5101, 1468-1331. DOI: 10.1111/ene.15071.

[142] Stephen W Smye and Alejandro F Frangi. “Interdisciplinary research: shaping the healthcare of the future”. en. In: *Future Healthc J* 8.2 (July 2021), e218–e223. ISSN: 2514-6645. DOI: 10.7861/fhj.2021-0025.

[143] Srivastava, Hinton, Krizhevsky, et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* (2014).

[144] M Q Stearns et al. “SNOMED clinical terms: overview of the development process and project status”. In: *Proc. AMIA Symp.* (2001), pp. 662–666. ISSN: 1531-605X.

[145] Sutskever, Vinyals, and Le. “Sequence to sequence learning with neural networks”. In: *Adv. Neural Inf. Process. Syst.* (2014). ISSN: 1049-5258.

[146] The Unicode Consortium. *The unicode standard, Version 14.0.0.* Tech. rep. The Unicode Consortium, Mountain View, CA, 2021.

[147] Eric J Topol. “High-performance medicine: the convergence of human and artificial intelligence”. en. In: *Nat. Med.* 25.1 (Jan. 2019), pp. 44–56. ISSN: 1078-8956, 1546-170X. DOI: 10.1038/s41591-018-0300-7.

[148] Freek Van de Velde, Stefano De Pascale, and Dirk Speelman. “Generalizability in mixed models: Lessons from corpus linguistics”. en. In: *Behav. Brain Sci.* 45 (Feb. 2022), e34. ISSN: 0140-525X, 1469-1825. DOI: 10.1017/S0140525X21000236.

[149] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30.* Ed. by I Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008.

[150] Alex Wang, Kyunghyun Cho, and Mike Lewis. “Asking and Answering Questions to Evaluate the Factual Consistency of Summaries”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5008–5020. DOI: 10.18653/v1/2020.acl-main.450.

[151] Alex Wang et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446.

[152] Changhan Wang, Kyunghyun Cho, and Jiatao Gu. “Neural Machine Translation with Byte-Level Subwords”. en. In: *AAAI* 34.05 (Apr. 2020), pp. 9154–9160. ISSN: 2374-3468, 2374-3468. DOI: 10.1609/aaai.v34i05.6451.

[153] Fei Wang and Anita Preininger. “AI in Health: State of the Art, Challenges, and Future Directions”. en. In: *Yearb. Med. Inform.* 28.1 (Aug. 2019), pp. 16–26. ISSN: 0943-4747, 2364-0502. DOI: 10.1055/s-0039-1677908.

[154] Yequan Wang et al. “Attention-based LSTM for Aspect-level Sentiment Classification”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 606–615. DOI: 10.18653/v1/D16-1058.

[155] Yiren Wang and Fei Tian. “Recurrent Residual Learning for Sequence Classification”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 938–943. DOI: 10.18653/v1/D16-1093.

[156] Jennifer Weller, Matt Boyd, and David Cumin. “Teams, tribes and patient safety: overcoming barriers to effective teamwork in healthcare”. en. In: *Postgrad. Med. J.* 90.1061 (Mar. 2014), pp. 149–154. ISSN: 0032-5473, 1469-0756. DOI: 10.1136/postgradmedj-2012-131168.

[157] Jonathan White and Muhammad Jawad. “Clinical Coding Audit: No coding - No Income - No Hospital”. en. In: *Cureus* 12.9 (Sept. 2020), e10664. ISSN: 2168-8184. DOI: 10.7759/cureus.10664.

[158] Sarah Wiegreffe and Yuval Pinter. “Attention is not not Explanation”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20. DOI: 10.18653/v1/D19-1002.

[159] Adam B Wilcox et al. “Use and impact of a computer-generated patient summary worksheet for primary care”. en. In: *AMIA Annu. Symp. Proc.* (2005), pp. 824–828. ISSN: 1942-597X, 1559-4076.

[160] Ronald J Williams and David Zipser. “A learning algorithm for continually running fully recurrent neural networks”. en. In: *Neural Comput.* 1.2 (June 1989), pp. 270–280. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco.1989.1.2.270.

[161] Yonghui Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: (Sept. 2016). arXiv: 1609.08144 [cs.CL].

[162] Pengtao Xie and Eric Xing. “A Neural Architecture for Automated ICD Coding”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 1066–1076.

[163] Zichao Yang et al. “Hierarchical Attention Networks for Document Classification”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1480–1489. DOI: 10.18653/v1/N16-1174.

[164] Lixia Yao et al. “Electronic health records: Implications for drug discovery”. en. In: *Drug Discov. Today* 16.13-14 (July 2011), pp. 594–599. ISSN: 1359-6446, 1878-5832. DOI: 10.1016/j.drudis.2011.05.009.

[165] Rosita Zakeri et al. “Biological responses to COVID-19: Insights from physiological and blood biomarker profiles”. en. In: *Curr Res Transl Med* 69.2 (May 2021), p. 103276. ISSN: 2452-3186. DOI: 10.1016/j.retram.2021.103276.

[166] Tianyi Zhang et al. “BERTScore: Evaluating Text Generation with BERT”. In: *Eighth International Conference on Learning Representations*. Apr. 2020.

[167] Yuhao Zhang et al. “Learning to Summarize Radiology Findings”. In: *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 204–213. DOI: 10.18653/v1/W18-5623.

[168] Yuhao Zhang et al. “Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5108–5120. DOI: 10.18653/v1/2020.acl-main.458.

[169] Peng Zhou et al. “Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 207–212. DOI: 10.18653/v1/P16-2034.

Appendix A

Appendix: Language and Free-Text Analysis Pipelines

This appendix firstly introduces natural language and written free-text as a flexible and efficient storage medium for data. We briefly summarise the relevant areas of linguistics that have informed the specific sub-fields of free-text analysis and text mining, providing an abstract view of a typical text analysis pipeline.

A.1 Language and Free-Text

A natural language (or language here on in) is a language that has organically grown without explicit definitions and planing by one or more persons [30]. This is in contrast to a formal language such as computer programming language or logic language that has been explicitly codified according to some language creation process. Natural languages, due to their organic development, are closely linked with a culture of those that speak, write or read it. Languages afford users a rich, accessible, efficient method to communicate our interpretation of the world around us [5]. A written language accompanies a language system such as a spoken language, or sign language that can be acquired without explicit learning, i.e. a child learning to speak their native language from engaging with parents.

Written languages enable easier storage and transfer of data compared with spoken / sign languages and since the invention of distributable media, i.e. the printing press / digital media / the internet, written language has facilitated mass, instant, global communication.

Linguistics provides a framework for the study of language, providing objective, scientific analysis of language, their structures and phenomena. All written languages include common linguistic constructs such as: morphemes - the smallest unit of meaning that a symbol or collection of symbols can possess in a language, and morphology - the rules that govern the assembly of units of meaning through combining morphemes. Together these support the building of a given language's vocabulary or lexicon. Syntax defines how words are ordered to form sentences and semantics provides rules for identifying the meaning conveyed by words and sentences. Syntax and semantics are central to a languages grammar - i.e. the rules of language that define how to effectively build sentences. Typical native speakers of modern languages such as English or French have vocabulary sizes of 20-35k words [63]. Grammars can be vast and complex with many exceptions to the specified grammar rules. Moreover, languages are constantly changing and evolve as their users refine, develop or discard parts of a language [15].

Large-scale analysis of texts was previously only possible through manual means, relying on human efforts to meticulously read, catalogue and store texts for further analysis at a later date. With microprocessor technologies, text mining and analysing processes that would have otherwise taken months or years if done manually can now be performed in seconds. Firstly, we will review how text is stored and computed within modern computing architectures then we will describe from a high level the steps involved within the field of text analysis.

A.2 Encoding Free-Text

(Modern) Computing systems operate on data at binary i.e. 0 and 1-bit level, representations. All data such as numerical, text, image, audio, video or sensor output signals can be encoded into a binary representation for processing and storage. Most mediums have standardized protocols and encodings allowing for the full breadth of possible data to be represented, stored and computed over by a variety of computing architectures. In the case of text, a universal encoding for a language is one that can encode all possible symbols or graphemes for that language, i.e. the languages alphabet plus additional characters for punctuation and whitespace, brackets and so forth. Initially, the standard encoding scheme was ASCII (American Standard Code for Information Exchange) [44], which was only able to represent English text, with 7 bits for characters and 1 bit for parity, representing 128 possible characters. ASCII-256 or the extended-ASCII code allowed for up to 8 bits for characters, or 256 possible characters including possible characters for Latin and Germanic based languages. The Unicode standard [146] and the associated encodings (utf-8, utf-16 and utf-32) are supersets to ASCII that provide: (near) universal means to represent and handle most modern languages simultaneously, flexibility to grow as new symbols are added, and only some extra storage requirements. With unicode any texts in any supported language can be stored, retrieved and computed over allowing text analysis developments and research that is initially done in one language to be relevant and useful with corpora in one or more different languages. Universal standards and encodings are relevant to this research, as with our increasingly globalised world, it is important that we reflect on the applicability of our work, and how others that do not use English exclusively, can use and benefit from these developments.

This thesis exclusively focuses on the analysis of English or American English text as these are the data sources available, however we reference relevant background work both from the clinical informatics and NLP domains and highlight where our own work is relevant for non-English text.

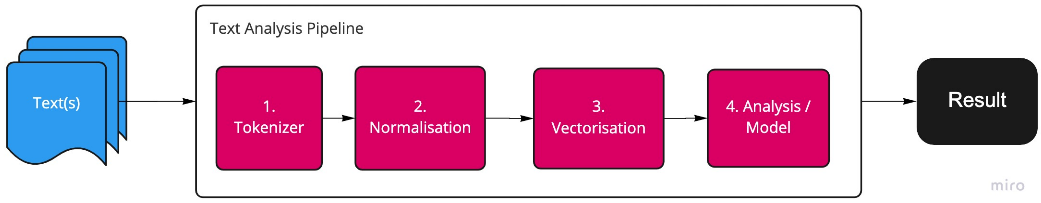


Fig. A.1 A typical text analysis pipeline and the stages involved to produce a result, insight or finding

A.3 Text Mining and Analysis

The flexibility afforded by a natural language presents difficulties in aggregate analysis of text. Text mining often refers to the transformation of text into normalised structures allowing for easier analysis. Common techniques include tokenization, normalisation i.e. stemming, lemmatizing and stop word removal. Text analysis often refers to the analytical processes or methods used to derive useful information from normalised texts. However, both these terms can be used interchangeably and are even often included within definitions of NLP. Fig. A.1 shows the typical steps carried out during the process of text analysis. The pipeline may include all or only some of these stages to produce the final intended result. Results are the aims of running the text-analysis pipeline and can be any arbitrary aggregate 'finding' that is somehow within the texts. Text analysis pipeline *results* with clinical notes could include: counting the number of patients that have been prescribed a drug, understanding the sentiment (i.e. positive or negative) of patients following a recent care experience, building a patient cohort for a clinical trial from recently recorded clinical findings / symptoms / disorders, or even a predictive model for in-patient mortality or hospital readmission. The final stage in the pipeline i.e. the *Analysis / Model*, can also be the final *result* although arguably trained AI models are only useful if they are performant according to target metrics, a further result.

Before running text through a pipeline researchers may pre-process it to discard sentences, paragraphs and documents that do not fit within a provided criteria. This process is known as *cleaning* and can be complex and time-consuming depending on the source text(s) under analysis. We will not go into great detail within this appendix but we describe the cleaning process in the thesis chapters where relevant. The second stage of the described pipeline, normalisation, could also be considered *cleaning* as it manipulates and potentially discards select tokens, the output of the tokenizer stage. *Data wrangling* is also another common term and is the process in which data is transformed into a usable format for processing.

We will now briefly describe each stage in the abstract text-analysis pipeline and review the relevant background research, providing a basis for the following chapter's methodological contributions and their applications.

A.4 Tokenization

Arguably the first step in any text-analysis pipeline is a tokenizer. This breaks up a contiguous block of characters into *tokens*. For some text T that is a sentence, paragraph, document, corpus or corpora, a tokenizer is a function F that produces sequence of 1 to n tokens X, that is $x_{1...n}$. Each token x_i is then further processed by each text-analysis pipeline step.

$$F(T) = X = (x_1 \dots x_n) \tag{A.1}$$

A simple, common tokenizer F breaks up T via white space characters such as spaces (\s). However, F can be any function that operates on sequences of characters. For example, a tokenizer could treat each sentence as a token, using end-of-sentence markers such as full stops (periods), new lines(\n) or carriage returns(\r) to identify token boundaries. If T is HTML, a markup language that allows content to be displayed within modern web browser technology, tokens could be split by common tags such as '<p>...</p>' or '<span>...</span>' or '<br>'. In practise, F can be of two categories. The first are fixed rule-based tokenizers that define token boundaries via one or more regular expressions [68].

Regular expressions (Regex) are a formal language that allows patterns of characters to be recognised and split according to arbitrary criteria as described earlier, e.g. whitespace, newlines, HTML tags etc. Regex rules are easy to implement, fast to run, and are built-in across multiple programming languages and environments i.e. computer servers, mobile devices and web browsers. However, a single regex can quickly grow in complexity and be difficult to debug for errors; they can also be brittle and perform in unintended ways if text data appears that was not originally coded for.

The second category of tokenizers do not define any explicit rules but are parameterised by running a parameterized algorithm over T. These tokenizers have recently become popular due to their ability to optimise the tokenization of T according to a target vocabulary size. This is practically important as down-stream stages of text analysis pipeline, i.e. stages 3 (vectorisation) and 4 (analysis/model), require each token to be represented and operated over by a fixed length numerical vector. This will be discussed in further detail in Sections A.5 and A.6. For now it is sufficient to know setting an intended maximum vocabulary size and allowing the algorithm to find an optimal tokenization F is useful downstream. A vocabulary V is the set of tokens of the sequence of tokens X outputted from the tokenizer F over texts T:

$$V(X) = x_{1 \ldots n}; x_i \in X \tag{A.2}$$

A.4.1 Sub-Word Tokenizer Algorithms

Zipf's law [95] is an empirical law in linguistics that states that word occurrences in any language and therefore our texts T, are inversely proportional to their rank. In other words, it describes the word occurrence relationship between where a word ranks and its volume of frequency of occurrence. For example, the first most common word in T will occur proportionally $2x$ the second most occurring word and the second most occurring word will proportionally occur $2x$ the third most occurring word and so on. This results in the frequent words in T dominating the occurrence distribution with a long tail of infrequent words. Given Zipf's law, a simple tokenizer that simply identifies tokens on white space and adds all unique entries to a vocabulary is likely to later encounter words that have never been seen before. These unseen tokens are known as out-of-vocabulary (OOV) tokens, and are often simply ignored or removed from the source text in downstream analysis.

An example algorithm for optimal V is Byte Pair Encoding [133] (BPE), initially used for lossless compression of strings. BPE takes all unique characters in T, counts and ranks all two character pairs adding the highest occurring pair to the vocabulary V. This joining and ranking process continues iteratively with the addition of each newly added item to V until the target vocabulary size is reached. The original algorithm allows for frequent sequences to be retained as full tokens whereas suffixes or prefixes of words are naturally split and treated as separate tokens. This allows for contextualised vectorisation (embedding) approaches, discussed first in 2.1 and then in Section 2.5, to represent previously OOV tokens.

BPE supported lossless compression as a byte to sequence mapping table allowed for long common sequences to be replaced with a single byte in the original source text T. Recent work has improved the performance of BPE using UTF-8 encoded representations of characters rather than characters themselves [152]. This was shown to be especially effective for for character rich languages such as Chinese and Japanese, and training multi-lingual tokenziers.

WordPiece [161] and SentencePiece [75] have also recently become popular due to recent state-of-the-art downstream models using these algorithms for tokenization. Broadly, these tokenizers continually train a model to optimise for the likelihood of each possible token given the dataset, balancing segmenting rare tokens into their common constituent parts and keeping common words whole.

A.4.2 Normalisation

Normalisation aims to remove or transform tokens into a *standardized* form ready for further downstream use. This stage can be considered an optional stage in the text analysis pipeline or even part of a pre-processing stage. Normalisation can provide performance gains depending on the following stages but is often only used in combination with rules based tokenizers, i.e. the first category discussed in the prior section, as these normalisation methods operate on whole words only. These category of tokenizers produce variable length tokens i.e. character, sub-word, word and multi-word tokens. It can be argued that they have an inherent normalisation step. We will now briefly review the common normalisation methods.

Stemming is the removal of suffix characters to reduce a token to its stem form irrespective of the context of the word. The intuition being that words with equivalent stems have equivalent meanings and therefore enables the reduction in the number of semantically distinct words, i.e. the vocabulary V, to be analysed. An example implementation, the Porter Stemmer [108], uses a suffix list and bespoke rules to transform tokens to their stem, although others could use word and stem dictionaries.

Lemmatization is similar to stemming but seeks to normalise a provided token into its lemma form, or the form of the token that would be found within a dictionary. Therefore, alongside the lemmatization algorithm, external dictionary resources are often used to reference lemma forms of words, although it is also possible to ‘train’ a lemmatizer to learn these base forms [98]. Lemmatization is an improvement upon stemming as it takes the tokens context and morphological features such as Part-Of-Speech (POS) tags into account. POS tags are the set of grammatical groups that each token can be assigned, for example a verb, adverb, adjective, noun etc. A common set of tags used is the Penn TreeBank tag set [86]. POS tagging is now considered as mostly a solved problem wrt. English, as algorithms have reached fast accurate performance with now >97% accuracy [4]. POS tags are useful during lemmatization as a tokens lemma can depend directly on the POS tag. Take for example the token ‘leaves’, the plural of leaf as well as the word to indicate leaving. A sentence such as ‘He always leaves early’ vs ‘the leaves are falling’, the former sentence would lemmatize the last token to the singular leave whereas the second sentence will use leaf for the singular of a tree leaf.

Our final common normalisation method is stop word removal. This is a simple technique to remove irrelevant tokens that uses a dictionary to filter out terms that should not be included in any further downstream analysis. This step is often performed after lemmatisation or stemming, is language specific, and for English often includes common terms such as ‘a’, ’the’, ‘that’, ‘has’, ‘it’, ‘who’ etc. Stop word removal is a simple but often effective approach at removing ‘uninformative’ parts of text. However, if this ‘uninformativeness’ measure is difficult to define this method may inadvertently remove important text.

A.5 Vectorisation

Stage 3 of our text-analysis pipeline is vectorisation, a process where each entry in Vocabulary V is transformed into a fixed size vector, or sequence of floating point numbers to encode the ‘important’ features of these words, i.e. the semantics and syntactical features, allowing computer architectures such as some analytical procedure to encode, modify and compute over these words.

Each item within V, regardless of how each entry was found in the previous two stages is a sequence of characters encoded via some encoding as discussed in Section. A.2. So, why cannot an encoding schema such as Unicode be used to encode words, to provide a consistent, transmissible bit representation? Firstly, an item in V can vary between a sub-word, word, multi-word token so the range of possible inputs is far greater than than the limited character sets used in most languages. Secondly, the building blocks of meaning, the morphemes of a language, are within words so such an encoding scheme would need to be able to universally encode this as well as any possible context that the words were found in, i.e. the syntax. The semantic and syntax representations are not a consideration in the encoding of letters or the graphemes of a language. As a research community, to achieve a universal encoding scheme for even a single language's morphology, the semantics, and the syntax would bring us one step closer to true NLU [13].

Despite the lack of a universal encoding scheme, we can still build meaningful and useful representations of text allowing further processing in the final stage of the pipeline. To build these useful representations, the text T now is assumed to be suitably large and is comprised of many individual texts $T = (t_1 \dots t_m)$, with each t_i outputting token sequences via our Tokenizer $F(t_i) = (x_1^{t_i} \dots x_n^{t_i}), x_n \in V$.

A.5.1 Sparse Token Representations

One method to choose a dimension for each Vocabulary entry vector is to simply use m, the number of documents in T. Stacking all vocabulary vectors then outputs a matrix with dimensions $|T| \times |V|$. This matrix can be populated by the counts of occurrences for each document and each vocabulary item column. For example, for the two sentences, $t_1 =$ 'patient diagnosis: heart disease' and $t_2 =$ 'discharge diagnosis: heart failure, heart attack', we have the vocabulary ('patient', 'diagnosis', 'heart', 'disease', 'discharge', 'failure', 'attack'), and therefore matrix M is:

$$\underset{2\times 8}{M} = \begin{array}{c} \\ t_1 \\ t_2 \end{array} \begin{array}{ccccccc} patient & diagnosis & heart & disease & discharge & failure & attack \\ \end{array}$$

$$\underset{2\times 8}{M} = \begin{array}{c|ccccccc} & patient & diagnosis & heart & disease & discharge & failure & attack \\ \hline t_1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ t_2 & 0 & 1 & 2 & 0 & 1 & 1 & 1 \end{array}$$

An alternative is 'one-hot-encoding' that represents the binary presence or absence of a word in a text, instead of of counting the number of occurrences. A more advanced but still commonly used method to vectorize T is term-frequency, inverse document frequency (tf-idf) [85]. This uses the previously found count matrix, i.e. the term-frequency, and normalises each count by the inverse document frequency that is the log scaled proportion of occurrences of the given vocabulary token as it appears across all documents. In simpler terms, the idf scales the document term count representation for each document aiming to highlight the words that occur infrequently across the entire corpus, the intuition being that the rare words are likely important and should be weighted in favour of those words. Applying this transformation to our 2 sentence corpus provides the following $\hat{M}$:

$$\underset{2\times 8}{\hat{M}} = \begin{array}{c} \\ t_1 \\ t_2 \end{array} \begin{array}{ccccccc} \textit{patient} & \textit{diagnosis} & \textit{heart} & \textit{disease} & \textit{discharge} & \textit{failure} & \textit{attack} \\ \end{array}$$

$$\begin{array}{c|ccccccc} & patient & diagnosis & heart & disease & discharge & failure & attack \\ \hline t_1 & 0.58 & 0.41 & 0.41 & 0.58 & 0 & 0 & 0 \\ t_2 & 0 & 0.3 & 0.61 & 0 & 0.43 & 0.43 & 0.43 \end{array}$$

Note how in t_2 the previous 'diagnosis' column vector is now penalized for appearing in both documents, with 'discharge', 'failure' and 'attack' being weighted favourably despite their equivalent term count. With a large set of real documents, and a meaningfully large vocabulary, the majority of the entries of this matrix will be 0, as any given document row is unlikely to contain all possible vocabulary entries entries available in V. This sparsity can present some difficulties, firstly a very large vocabulary and very large corpus can be become computationally unwieldy to manipulate for further downstream processing, and storage can be challenging depending on how the matrix representation is stored.

Each column of M and $\hat{M}$ can be referred to as a *feature* and can be used as input into the next stage of the text analysis pipeline. Tf-idf can be useful to provide interpretable features of results, as impactful features can directly be mapped back to specific words.

Importantly, vocabulary V is a set and is therefore unordered. This gives count vectorisers and tf-idf the name 'bag-of-words' approaches, as the sequence order of each document is lost during vectorisation. So, given a row vector t_i of M it is not possible to retrieve the original text. However, the sequence order of tokens can be very important in the text's overall meaning. Let us consider the text: 'no sign of diabetes, positive for hypertension', a tf-idf representation would not be able to discern whether the negation refers to diabetes or hypertension.

A field of linguistics, distributional semantics, is built around the idea that terms that appear in similar contexts i.e. appear with similar words around them, share similar semantic properties, which emerge when analysing large corpora [48, 39]. An initial approach to capture this could be to build a large co-occurrence count matrix with dimensions $|V| \times |V|$. Each entry is the count of the token at row i appearing in the *context* of the column j token. The *context* is some predetermined window that could be one, three or even 10 token to the left and right of the token under consideration. This matrix is large, quadratic with respect to V, and sparse as only a small fraction of tokens will appear in the context of another token given a sufficiently large V.

Further vectorisation methods crucial to the thesis are covered extensively in Section 2.1.

A.6 Analysis / Model

Given the previous 3 stages we now have the methods to take a text corpus T, tokenize, normalise and vectorise into a matrix M, that capture both the semantic and syntactic properties of the text. This next stage can vary from simply ranking, or visualising directly a tf-idf matrix to understand important words or phrases in a corpus or conducting simple search queries. For example, in an information retrieval (IR) scenario, given a vectorised query string and a tf-idf matrix, we could rank all documents by the cosine-similarity, as shown in Equation. 2.6, returning results that are not exact matches, alongside those that are semantically similar.

This stage of the text-analysis pipeline is the most open-ended and varied. Many use cases utilise machine learning methods, a subset of Artificial Intelligence (AI), to transform this vectorised representation of T into some final *useful*, humanly interpretable result.

The application of machine learning techniques can be seen across use cases (classification, regression, clustering, generation) and data modalities (text, images, video, speech / signal, processes etc.). Prediction models that rely on *classical* statistical models such as linear and logistic regression analysis can be considered within the realm of machine learning.

Appendix B

Artificial Neural Networks

Artificial Neural Networks (ANNs) are able to model non-linear relationships between X and y, unlike logistic or softmax regression. ANNs are based on the *connectionist* theory of computation [41] where many small, simple and interconnected units or *nodes* of computation are arranged to model higher order complexity. Connections between nodes are reinforced or diminished according to the input signal that flows through each node, its corresponding *activation* from carrying out some computation, and finally most importantly the feedback signal supplying the means for the node to modify its internal state so future *activations* are an improvement over the prior.

Neural models can also be called representation learners, as each node gradually adjusts its parameters to produce a latent *representation* of the original input via the produced activation. Through this layered composition of functions, representations of data are produced that allow for *features* that would otherwise be hidden in the data to be found and further used for the downstream purposes, i.e. classification or regression.

ANNs are often organised into *layers* of these simple computational units, and are configured with specific connection *architectures* and computational configurations. I will now review the broad range of neural model architectures relevant to our work. This includes initially fully-connected / feed-forward networks and the theoretical framework of neural models, then I will review recurrent neural networks that are ideal for sequence modelling tasks as they specifically allow for a 'memory' within the model. I will then briefly review the Transformer, and its effectiveness across a broad range of tasks. I will close this section with a review of language modelling, transfer learning and their relevance to clinical NLP and summarisation.

B.1 Feed Forward Neural Networks

A neural network model is the combination of the network topology and the parameters represented as matrices or weights connecting the nodes. Figure B.1 shows a simple feed-forward, fully connected neural network [55] for binary classification as signified by the single output neuron on the right hand side, alongside the weight matrices that represent the parameters of the network W^1 and W^2. Given the input samples X and output labels y. A forward pass through the network computes the *error* or *loss* between the output $\hat{y}$ and the intended labels y. In the example X has two features, x_1, x_2, represented by two network nodes on the left hand side. The example has a single hidden layer with 3 nodes, the middle layer. Each hidden layer node receives both inputs x_1, x_2. The *activation* at each node is computed by the dot product of the weight vector $\vec{w}$ and the input $\vec{x}$. A non-linear activation function f outputs:

$$z(x) = f(w \cdot x) \tag{B.1}$$

Example activation functions are:

Sigmoid:

$$f(x) = \frac{1}{1 + \exp(-x)}$$

ReLU:

$$f(x) = max(0, x)$$

Leaky ReLU:

$$f(x) = \begin{cases} \alpha x, x < 0 \\ x, otherwise \end{cases}$$

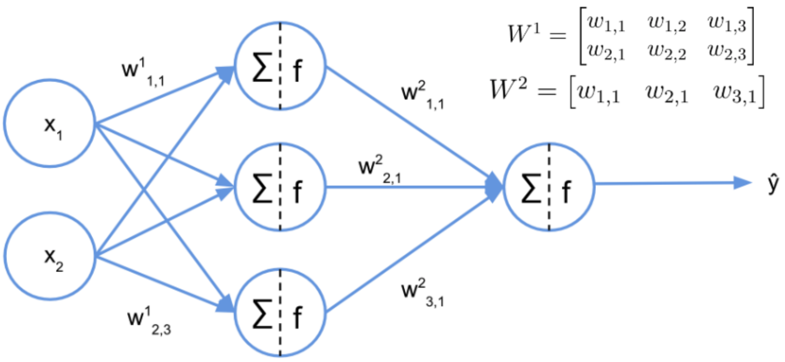

$W^1 = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}$
$W^2 = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \end{bmatrix}$
$w^1_{1,1}$
x_1
x_2
$w^1_{2,3}$
Σ f
Σ f
Σ f
$w^2_{1,1}$
$w^2_{2,1}$
$w^2_{3,1}$
Σ f
ŷ

Fig. B.1 Example Fully-Forward Neural Network Architecture

Activation functions are often simple fixed functions that provide a non-linearity between inputs and outputs. As inputs flow through the network and are transformed by composing dot product and activation functions, the *forward pass* completes at the *output* layer. In the example architecture f at the last node is the sigmoid function providing a probability value for binary classification problem. For 3 or more classes the soft-max as described in 2.3.2, is applied as the final activation function with an output layer node per class.

Linear functions, such as the weighted sum of a networks connection, even when composed i.e. stacked in layers, would still only be able to model linear functions. The non-linearity of the activation function provides the means for ANNs to be universal approximators for a given function space as suggested in prior work [55], given a sufficient number of hidden nodes (breadth) and layers (depth) to a network.

B.1.1 Backpropagation

In the binary classification scenario, a forward pass through the network produces an output probability vector $\vec{\hat{y}}$ for each sample i that can be compared with our labels $\vec{y}$ producing the loss L:

$$L = \sum_i \hat{y}_i - y_i \tag{B.2}$$

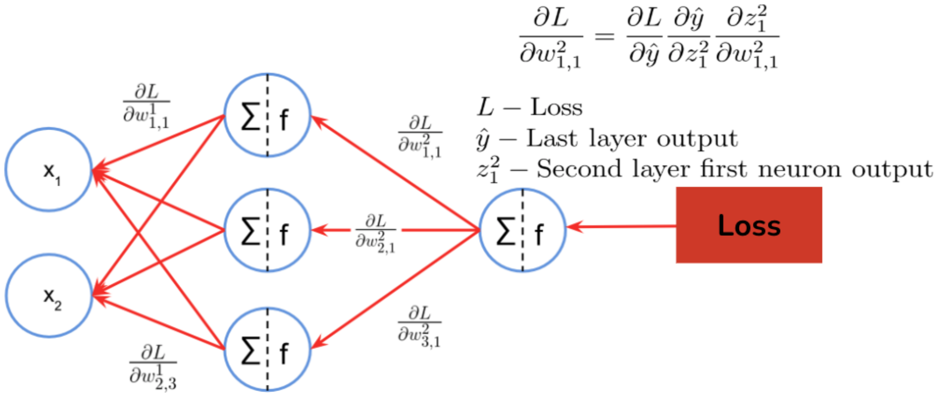

$\frac{\partial L}{\partial w^2_{1,1}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^2_1} \frac{\partial z^2_1}{\partial w^2_{1,1}}$
L – Loss
$\hat{y}$ – Last layer output
z^2_1 – Second layer first neuron output
$\frac{\partial L}{\partial w^1_{1,1}}$
$\frac{\partial L}{\partial w^2_{1,1}}$
$\frac{\partial L}{\partial w^2_{2,1}}$
$\frac{\partial L}{\partial w^2_{3,1}}$
$\frac{\partial L}{\partial w^1_{2,3}}$
x_1
x_2
Σ f
Σ f
Σ f
Σ f
Loss

Fig. B.2 An example backward pass through our network, and the partial derivatives of the Loss (L) wrt. to each weight vector at each vector connection (parameter).

Backpropagation is the backward pass that computes the amount of loss or error of our current network configuration given the inputs X. This uses the same gradient descent calculation described in Section 2.3.1 for logistic regression, the θ parameters are the weight matrices of the network W^1 and W^2. Importantly, backpropagation requires all forward pass functions to be differentiable. This allows for the calculation of the derivative of the loss L wrt. to the network weights W. Similar to logistic regression, this is the sequence of partial derivatives of the loss wrt. each parameter $w^i_{j,k}$.

The forward pass in a general feed-forward ANN composes weighted sums, non-linear functions at each layer. To take the partial derivative of the loss wrt. each parameter the chain-rule is repeatedly applied as shown on the right of Figure B.2 for a single parameter from a hidden node to the output node.

Each partial derivative can be thought of as the amount of influence that the incoming signal should be either reinforced or diminished by to minimise the loss. Given all the partial derivatives $(\frac{\partial L}{\partial w^1_{1,1}} \cdots \frac{\partial L}{\partial w^i_{j,k}})$ Equation 2.4 can be followed to update the parameters. The forward / backward process then starts over. The loss via the forward pass is computed again with these revised parameters, the partial derivatives of the loss wrt. each parameter is calculated and each parameter is adjusted according to their derivatives. This process repeats until convergence, that is, until the loss no longer continues to reduce.

For NLP problems, a feed-forward NN could use tf-idf matrix as the input layer with $|V|$ input neurons, one for each vocab entry, or in our word vector example, we could average all embeddings in a sentence or paragraph to produce each $\vec{x}_i$ in X. This example would have an input neuron for each word vector component.

To avoid overfitting, which is the result of generalising poorly to unseen test data whilst simultaneously performing overly well on the training data, a common regularisation method for neural networks is Dropout [143]. During model training this randomly drops outputs of nodes within a layer with a set probability ρ, forcing the parameters or weights of receiving nodes to not overly rely on single or a small number of nodes from sample to sample. During validation and testing the $\rho = 0$ means all nodes are active during these stages.

B.2 Sequence Modelling in Neural Networks

As first discussed in Section A.5.1, using bag-of-words such as tf-idf and word vector averages removes the sequence order of inputs. Sequence order in language is important but feed-forward neural networks, regardless of network breadth or depth are unable to explicitly model the order of input sequences.

Recurrent neural networks (RNNs) are a class of network topology that contain a loop or recurrence. The simplest network would be a single node with a single connection to itself. Figure B.3 shows this network and the *unrolled* version i.e. a version of the network topology without loops. This shows that each successive input item in the sequence $x_1 \cdots x_n$ e.g. each word in a sentence is processed in order by the node. At each sequence input item the node produces an output and hidden node state. Successive nodes then receive a hidden state, the *memory*, of the past sequence alongside the next input item.

General RNNs are known to be susceptible to the problem of vanishing or exploding gradients [70]. Unrolling a recurrent node for the total length of a input sequence involves creating a network as *deep* as the sequence. Optimising such a network is now via

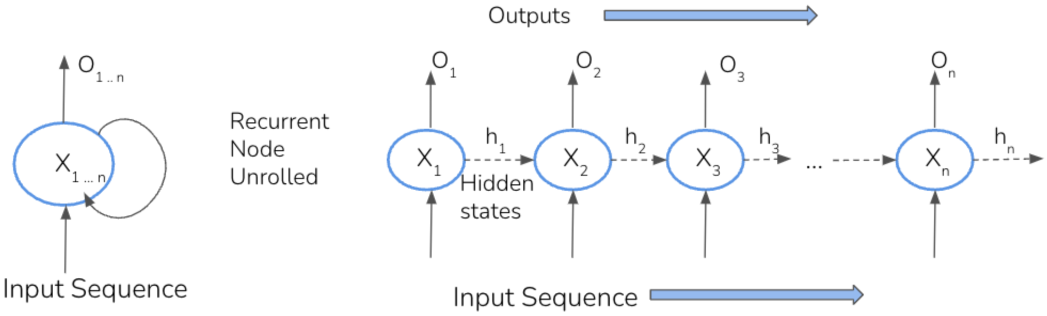


Fig. B.3 A single node recurrent neural network, and the unrolled version

backpropagation *through time*, which similar to regular backprop involves the product of multiple gradients through the successive applications of the chain-rule. During the training process if gradients become small the calculation quickly reduces to ≈ 0, or if gradients are large value, the calculation can quickly overflow. Specific RNN architectures tackle this through *gating* logic, such as the long short-term memory (LSTM) [52] and the more recent gated recurrent unit (GRU) [29]. These architectures contain parameters that *gate* how information flows through the node, protecting the internal state or *cell* from both the previous hidden state input and the current input item. Prior work showed that these gates minimised the convergence issues of vanishing or exploding gradients, but also allowed dependencies between input sequences to be more accurately modelled. For clinical NLP these RNNs have been used for the effective modelling of negations [137], temporal expressions [46] and named entity recognition [100] amongst others.

B.2.1 Sequence to Sequence Modelling

Our review of methods so far has mostly discussed problems of binary or multi-class classification. This presents problems as a matrix $X = (x_1, \cdots x_n)$ where each row vector $\vec{x_i}$ outputs a single label $\vec{y} = (y_1, \cdots y_n)$ indicating the intended class. Multi-label problems allow for y_i to be a sequence of applicable class labels. Another common problem is the modelling of sequence-to-sequence (seq2seq) problems that have the same input X but now each y_i is another sequence over the shared vocabulary V, e.g. $\vec{y_i} = (v_1 \cdots v_k)$. For clinical

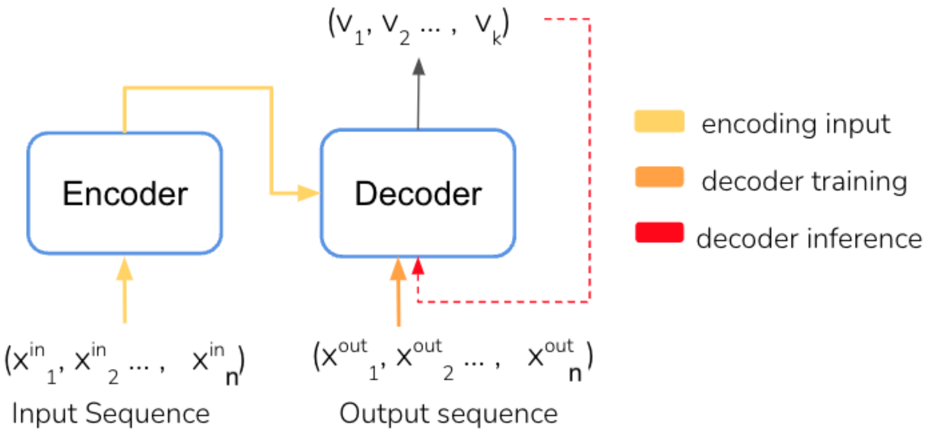


Fig. B.4 An abstract Encoder / Decoder model architecture showing the variation in decoder inputs according to training or inference modes

NLP this includes problems such summarisation [167], medical question answering [129] and medical translation or simplification [32].

Prior work has shown seq2seq models perform well in an *encoder-decoder* architecture [145]. Figure B.4 shows an example encoder-decoder architecture showing distinct encoder and decoder networks and the input / output data flow during training and inference time.

The *encoder* model is tasked with learning representations of the input resulting in a fixed dimensional vector for the variable length sequence. This is similar to the models we have discussed for classification of sequences. The decoder portion uses two inputs. Firstly, the encoded input sequence from the encoder, and secondly the output sequence. During training time the output sequence, or the intended sequence to be generated, is fed into the decoder alongside the encoded input. This is known as *teacher-forcing* and improves training convergence [160]. During inference time the output sequence is the previously output sequence from the decoder. All seq2seq models vocabularies include special characters to indicate the start and end of sequences e.g. ‘<start>’, ‘<end>’. During inference time the decoder is provided the encoder input and the ‘<start>’ sequence token and continues to produce new tokens conditioned on the previously decoded tokens until the ‘<end>’ token.

Early work focused on encoder-decoder seq2seq models used stacks of LSTMs [145], however these models often were not able to model long range dependencies between items in a given sequence. The *memory* of a cell in the LSTM case or simply a node in a general RNN are stored in the latest hidden state i.e. h_{t-1}. This means with each new input item, there is an opportunity for previous states to become corrupted or inaccessible. Natural language often has many potential long range dependencies occurring in even the simplest of text. For example, a clinical note might introduce a person with their name and then subsequently use a pronoun. Usage of the pronoun could be after many input tokens, so the model must be able to reference back to the initial introduction of the person via their name.

B.3 Long Range Dependencies in Sequences

Attention is a method to provide models a view of prior input states alongside the directly previous hidden state h_{t-1}. Originally a method to allow an encoder-decoder architecture model to allow the decoder portion to selectively *attend* to prior hidden states previously output by the encoder [8]. The authors propose a *soft-alignment* model via a feed-forward neural network that computes a context vector c_i for each timestep used in the calculation of the next hidden state h_i. The context vector c_i is the weighted sum of all previous encoder hidden states, with the weights α_{ij} calculated as the softmax of the *alignment* of each previous hidden and current hidden state. The equations are as follows:

$$h_i = f(h_{i-1}, y_{i-1}, c_i) \tag{B.3a}$$

$$c_i = \Sigma \alpha_{ij} h_j \tag{B.3b}$$

$$\alpha_{ij} = \text{softmax}(e_{ij}) \tag{B.3c}$$

$$e_{ij} = a(h_j, h_i) \tag{B.3d}$$

For the alignment calculation $a(h_{i-1}, h_i)$, the original paper uses the concatenated hidden state of bidirectional LSTM hidden state for the current h_i. Bidirectional LSTMs [132], enable the unrolled LSTM node to view input sequences from both left-to-right and right-to-left. This provides improved representation learning as often sequence items to the right may affect the interpretation of a current item, which would otherwise not be interpretable to a unidirectional LSTM. a is modelled using a simple feed-forward neural network with one hidden layer.

The concept of attention in RNNs have been heavily studied [42, 138, 158] and applied beyond the encoder-decoder and translation use cases [163, 169, 154]. However, RNNs are by design only able to process one sequence input item at a time. Current hidden and output states depend on either the direct or n many prior hidden states via an attention mechanism. In comparison to feed-forward networks, entire layers, i.e. each neuron activation, can be calculated in parallel, making them highly efficient for long input sequences.