NAMED ENTITY RECOGNITION IN A MEDICAL CONTEXT

by

PAVLO NAZARCHUK
URN: 6657209

A dissertation submitted in partial fulfilment of the
requirements for the award of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

May 17th

Department of Computer Science
University of Surrey
Guildford GU2 7XH

Supervised by: Suparna De

I declare that this dissertation is my own work and that the work of others is acknowledged and indicated by explicit references.

Pavlo Nazarchuk
May 17th

© Copyright Pavlo Nazarchuk, May 17th

Abstract

In today's world of increasing digitization, information is constantly being uploaded and processed through massive databases. However, the raw data is often too extensive and difficult to process manually. This task would be both time-consuming and costly, requiring multiple personnel to handle it.

With the development of Artificial Intelligence (AI), the possibility of labelling and clustering data became a possibility without the aforementioned issues. Transformers became a prominent architecture in the AI department and have shown great capabilities in Natural Language Processing (NLP).

Medical named entity recognition (NER) is a crucial task for labelling medical information, and it plays a fundamental role in analyzing electronic medical records, aggregating data, filtering text, and constructing graphs. With the extensive use of electronic medical records in every hospital, there is a need for effective clustering and filtering techniques. However, developing NER models comes with several challenges such as input noise, unreliable data, and lack of sufficient data. This report addresses these challenges and presents solutions to improve the accuracy and efficiency of NER models within a medical space. The project implementation can be found on `https://github.com/MedNer-Surrey/MedNer-FYP`.

Acknowledgements

This project was made possible because of the contribution, support and guidance of various people across this journey. I, therefore, will be taking this moment to acknowledge and express my gratitude for everyone involved.

I would to thank my supervisor, Dr. Suparna De, for this opportunity. This project idea was proposed by her. Without her expertise, guidance, extensive research and vital recommendations the advances of this project would not have been far as great as they have been. Furthermore, Dr. Suparna De directed me towards the right people for insight and further research which proved to be vital for this project.

Friends and family have also been of great help throughout the years. Completing this course and dissertation is an intense and stressful task that was eased by the support of the aforementioned.

Contents

1 Introduction **12**

1.1 Background 13

1.2 Aims & Objectives 13

2 Literature review **16**

2.1 NER 16

2.1.1 Supervised NER 17

2.1.1.1 Conditional Random Fields 18

2.1.1.2 Support Vector Machines 18

2.1.1.3 Recurrent Neural Networks 18

2.1.1.4 Transformers 19

2.1.2 Unsupervised NER 20

2.1.2.1 Clustering 20

2.1.2.2 Rule-based Methods 20

2.1.2.3 Statistical Models 21

2.2 Medical NER 21

2.2.1 DICE 22

2.2.2 Semantic information retrieval on medical texts 22

2.2.3 Historical Documents and OCR 23

3 Challenges & Dealing with these challenges **24**

3.1 Lack of annotated data . . . 24

3.1.1 Privacy concerns . . . 24

3.1.2 Paid software . . . 25

3.2 Dealing with a lack of annotated data . . . 25

3.2.1 Research of available data . . . 25

3.2.2 Curated data . . . 26

3.3 New concepts . . . 26

3.4 Noisy input . . . 26

3.4.1 Spelling mistakes . . . 27

3.5 Dealing with new concepts & Noisy input . . . 27

3.5.1 Context NER . . . 27

3.6 Lack of IT knowledge in the medical space . . . 28

3.7 Dealing with lack of IT knowledge in the medical space . . . 28

4 Implementation **29**

4.1 Model training . . . 29

4.1.1 Spacy NER . . . 29

4.1.2 MACCROBAT-EE . . . 30

4.2 Model testing . . . 32

4.2.1 Dataset differentiation . . . 32

4.2.2 Spelling mistakes effect . . . 33

4.2.3 Context variation effect . . . 34

4.3 Back end . . . 35

4.4 Front end . . . 37

4.5 Data annotator app . . . 39

4.6 Automation architecture . . . 39
4.6.1 Multi-stage training . . . 40
4.7 Containerization . . . 41

5 Conclusion & Further work **42**
5.1 Conclusion . . . 42
5.2 Further work . . . 43
5.2.1 New data . . . 43
5.2.2 Web app improvements . . . 43
5.2.3 Supervised approach . . . 44
5.2.4 Unsupervised approach . . . 44

6 Legal, Ethical, Social and Professional (LSEP) **46**
6.1 University Ethics Process . . . 46
6.1.1 What data is to be collected? . . . 46
6.1.2 How data is stored? . . . 47
6.1.3 Ethical Review . . . 47
6.2 Legal & Ethical . . . 47
6.2.1 Public Interest & Do no harm . . . 47
6.2.2 Informed Consent . . . 47
6.2.3 Confidentiality of Data . . . 47
6.2.4 Transparency . . . 48
6.3 Professional Competence and Integrity . . . 48
6.4 Social Responsibility . . . 48

List of Figures

2.1 NER model output example 16

2.2 HunFlair NER architecture [1] 17

2.3 Encoder/Decoder architecture of transformers [2] 19

4.1 Spacy architecture [3] 29

4.2 MACCROBAT loss/steps graph 31

4.3 MES-Twitter loss/steps graph 31

4.4 Medical NER loss/steps graph 31

4.5 Graph of success/context 35

4.6 Backend architecture 36

4.7 WebApp front-page 37

4.8 Tokens tab 38

4.9 Entities tab 38

4.10 Statistics tab 38

4.11 NER Annotator app 39

4.12 Cron representation 40

4.13 Docker container 41

5.1 Unsupervised architecture [4] 45

List of Tables

4.1 Dataset statistics 30

4.2 Dataset test metrics 32

4.3 MACCROBAT-EE class test metrics 33

4.4 Average success rate per word 34

5.1 Objectives table 43

Glossary

g	Gradient of loss function
f	Loss function
β, η	Parameters
m_t	First moment estimates
ASR	Average success rate function
$label_n^c$	Count of correct labels
$label_n$	Count of incorrect labels
A	Cosine vector of correct word
B	Cosine vector of misspelt word
$S_c(A, B)_n^c)$	Cosine distance of correctly labelled word
$S_c(A, B)_n)$	Cosine distance of incorrectly labelled word

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ASR	Average Success Rate
BERT	Bidirectional Encoder Representations from Transformers
CLDC	Common Law Duty of Confidentiality
CRF	Conditional Random Fields
DB	Database
DICE	Data-Efficient Clinical Event Extraction with Generative Models
ER	Emergency Room
EHR	Electronic Health Records
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HMM	Hidden Markov Model
IT	Information Technology
JSON	JavaScript Object Notation
LSEP	Legal, Ethical, Social and Professional
LSTM	Long short-term memory
NER	Named Entity Recognition
NLP	Natural Language Processing
OCR	Optical Character Recognition
RNN	Recurrent Neural Network
SVM	Support Vector Machines
UMLS	Unified Medical Language System

Chapter 1

Introduction

Named Entity Recognition (NER) is a natural language processing (NLP) task that strives to identify and categorise specific named entities, such as people, organisations, and locations, within unstructured text.

Unstructured data has been a prevalent issue in the medical realm due to system integration issues. This data is often uncategorised and unlabelled. This lack of data treatment frequently requires the creation of suitable data management and analytics solutions, which are costly. Medical personnel usually show discontentment towards this data mismanagement as central hospital IT departments undermine the opportunities to use the data towards efficiency improvements[5]. The quest for precision and efficiency in information extraction has become paramount in the rapidly evolving healthcare landscape. Amidst the vast medical data being processed daily, labelling and clustering such data has become increasingly difficult and necessary. NER is a pivotal technology architecture that provides a systematic and automated approach to extracting and labelling unstructured medical data[6].

The domain of NER has evolved in unprecedented manners in recent times. As Artificial Intelligence (AI) algorithms advance, so do NER architectures. Its efficiency and accuracy have proven useful in many contexts, so much so that NER in a medical setting has become a prevalent theme. The current state-of-the-art medical NER can recognize five entity types with high accuracy, namely cell lines, chemicals, diseases, genes and species. It merges character-level language model (LM) pretraining and joint training on multiple gold standard corpora. It is easily accessible to non-experts. However, it holds limitations such as adaptability due to its low amount of entity types and domain specificity as certain medical fields may not be well-

represented in the corpora used for training and lack of qualifier grouping as it only labels the words based on the corpora and not their qualifiers.

Beyond its role as a data extraction tool, NER can unlock profound insights from medical records. By identifying and labelling identities such as diseases, treatments, and medical conditions, NER contributes to a more comprehensive understanding of patient histories and disease trajectories. As a result, evidence-based decision-making, personalised prescriptions, and clinical research are facilitated and optimised.

Integrating NER into medical systems creates an interconnection between the health information ecosystems. This connection allows different medical fields to cluster information and produce better outcomes. This interconnectedness, in turn, creates advancements in diagnostics and treatment strategies. Such connections enable medicine to cross-match cases from past patients and arrive at conclusions more efficiently[7]. A more technical approach is the need for organised data. With the vast number of patients presenting to Emergency Rooms(ERs) daily, NER systems only allow efficient data storage, as technologies such as vector cosine distances can be implemented to match patients and cluster them into different categories.

1.1 Background

This project is inspired by the development of a NER system for Epsom Hospital[8]. It is based on their need for a model that specialises in hip/knee surgery pre-operative patient notes and ER admission notes. This development involves creating a NER system that caters to all medical fields and developing an application that can integrate simplicity and artificial intelligence. This is inspired by the need for structured data that is unavailable within the hospital's IT systems and the importance of data analysis in reaching more accurate and satisfactory medical outcomes. This scarcity of usable data can be mitigated by integrating an accurate NER system and a text search algorithm that clusters information efficiently.

1.2 Aims & Objectives

This project aims to conceptualise, research and develop a medical NER model using state-of-the-art architectures to achieve high accuracy and overcome medical NER's challenges. This research believes that models must contain a relevant amount of labelling classes to produce

detailed labelling that is deemed valid when combined with data-searching algorithms. These classes serve the purpose of introducing detail and structure into medical data to allow for better analysis of stored data. Furthermore, the models must be capable of sequential labelling to cluster detail alongside medical events inside the same label. Medical NER's challenges can be attributed to various factors, such as the lack of data processing inside medical facilities and challenges within the complexities of developing NER systems catered towards specific medical fields. Hospitals store vast amounts of data daily without having a proper infrastructure to take advantage of this data in daily tasks. Using an NER model, hospitals could store this data in a manner that is easy to search. However, developing medical NER models holds challenges, such as the lack of adaptability and the lack of publicly available data that caters to a specific medical field. This project extends its aims through the development of a medical NER application that is implemented with automation at the forefront and a careful balance between simplicity and complexity. This application strives to allow medical personnel to leverage the powers of NER models in a simple manner and overview its results without having any programming knowledge. This project also emphasizes the need for automation in the medical field by implementing an automated way of model multi-stage training to account for the lack of IT knowledge in the medical field. This training architecture removes the need for training NER models from scratch and allows the users to upload data for training and have the model train itself on the new data without removing any previous knowledge. The objectives that come to achieve such aims are as follows:

- Explore the challenges of utilising AI in a medical setting and the difference between medical and non-medical NER.
- Introduce a dataset optimised towards sequence labelling and medical annotations.
- Introduce state-of-the-art NER model architectures that overcome the challenges of medical NERs.
- Propose a novel medical NER model and conduct a detailed analysis of the model's behaviour when faced with sequence labelling, noise and context variation.
- Introduce a Web App that leverages the proposed NER model and integrates it with a simple and automated interface and integrates a data annotator feature with the app.

These objectives have as a desired outcome an improvement in the data processing within medical

facilities as well as the facilitated deployment of a robust application capable of introducing custom model training, text labelling and data annotation without the necessary skills to perform such tasks.

Chapter 2

Literature review

2.1 NER

NER is a natural language processing (NLP) technique that involves identifying and classifying named entities within a body of text. It strives to categorise name entities into classes such as organisations, locations, dates, people, and more. This labelling comes from some input text segmented, tokenised and morphosyntactically analysed. It establishes relationships between words and tries to predict what category a word belongs to based on the context[9].

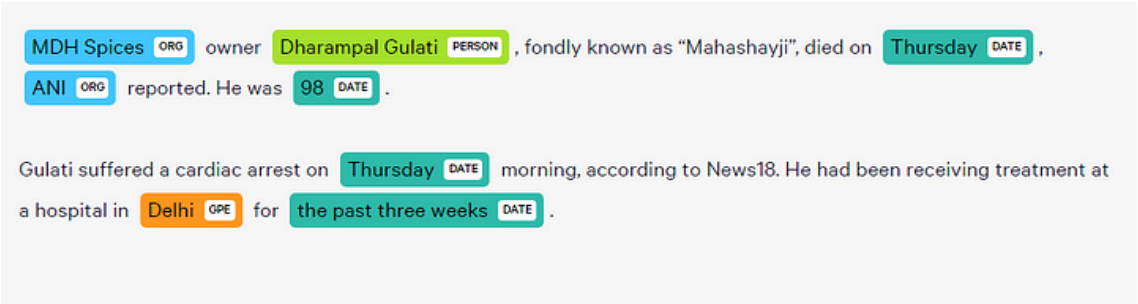

Figure 2.1: NER model output example

NER uses various algorithms and techniques. Sequence labelling models like BiLSTM-CRF [10] learn patterns and features automatically from annotated data to make predictions. Transformer-based models such as BERT and GPT have shown remarkable performance in NER tasks due to their ability to learn about contextual information effectively[11]. These models pre-train on large text corpora and fine-tune on NER-specific datasets, allowing them to achieve state-

of-the-art results. Additionally, active research focuses on leveraging multi-task learning and exploring unsupervised and semi-supervised approaches to enhance NER performance across various domains and languages.

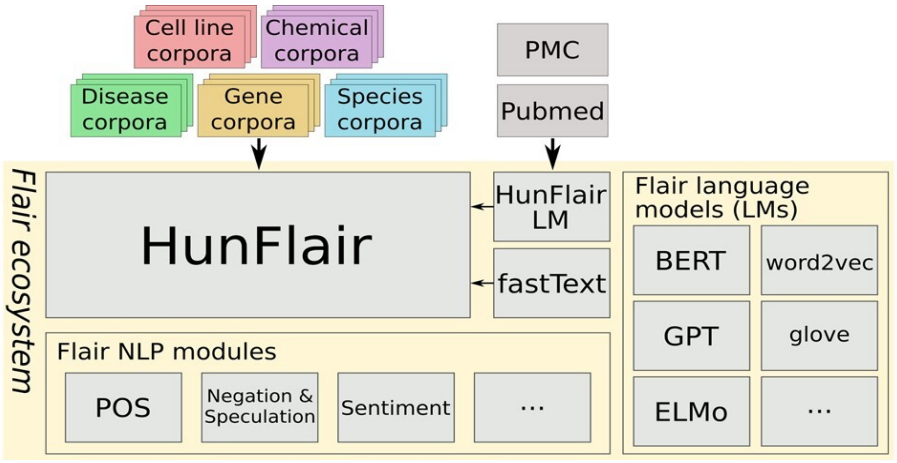

Figure 2.2: HunFlair NER architecture [1]

HunFlair uses the aforementioned transformer-based models and large corpora. It has achieved F1 scores as high as 87.71% in distinguishing gene entity types in the BIONLP CG dataset[1].

2.1.1 Supervised NER

Supervised NER relies on pre-labelled data for model training. It requires a dataset of example text along with their annotated named entities.

Algorithm 1 JSON format of NER data

$$
\begin{aligned}
&T = \{texts : t_1, \ldots, t_n\} \\
&t_1 \ldots t_n = \{keyword : k_1, \\
&\qquad\qquad label : l_1, \\
&\qquad\qquad \ldots, \\
&\qquad\qquad k_n l_n\}
\end{aligned}
$$

Supervised NER employs machine learning algorithms such as Conditional Random Fields (CRF), Support Vector Machines (SVM), Recurrent Neural Networks (RNNs) or Transformers. Recently, Transformers have been prevalent in AI as they have shown excellent capabilities in various fields [12]. The model is trained iteratively by adjusting its parameters and hyperparameters to minimise the difference between its predictions and the actual annotations in the

labelled dataset. This distance is calculated through a loss function that best represents the distance between annotations. The curation of this data is essential for an accurate model.

2.1.1.1 Conditional Random Fields

Conditional Random Fields are a type of probabilistic model and a special case of Markov Random fields. They're widely used in NLP tasks due to their flexibility and effectiveness, although they require annotated training data. This limitation of requiring annotated data is the definition of supervised learning. This type of learning requires extensive data to obtain good accuracy. CRFs have proven to obtain F-scores of up to 63.69% when applied to medical NER[13]. These scores are not high, and therefore, their applications to medical NER are not suitable.

2.1.1.2 Support Vector Machines

Support Vector Machines are a type of supervised learning model used for classification and regression tasks. They find the best separation for data points of different classes by maximising the margin. SVMs are known for handling multidimensional data and nonlinear decision boundaries using kernel functions. They are effective in various domains but require a careful selection of hyperparameters. This can be applied to NER due to its good generalisation performance and high accuracy in text categorization[14]. However, in Medical NER this would not be optimal as generalization is not the goal, instead detail attention and adaptability. When applying SVMs to medical NER, accuracy scores have been proven to reach more than 80%. However, most NER systems reach scores of around 90% [15].

2.1.1.3 Recurrent Neural Networks

Recurrent Neural Networks are specialised neural networks for processing sequential data. They cache previous inputs through hidden states. Long short-term memory (LSTM) and Gated Recurrent Units (GRUs) are popular variants that address the vanishing gradient problem and improve long-range dependency learning. RNNs are used in language modelling, speech recognition, and time series prediction. However, struggle with parallelisation and complex data structures[16]. RNNs are also limited by long-term dependencies, and this limitation can lead to difficulties in modelling complex relationships between words and their corresponding entity

labels, especially in medical texts, where context is crucial for accurately identifying entities. RNNs in medical text can reach scores of up to 81.87% [17].

2.1.1.4 Transformers

Transformers are a type of deep learning model. They have revolutionised natural language processing (NLP) and achieved state-of-the-art performance on various tasks. Transformers are based on a self-attention mechanism, which allows the model to determine the importance of different input tokens when making predictions. Transformers consist of an encoder-decoder architecture. The encoder processes the input sequence while the decoder generates the output sequence. This architecture implements multiple layers of self-attention, which are called cross-attention. Implementations that take advantage of this architecture are, for instance, BERT and GPT, which are more commonly used for their ChatBot abilities to answer questions with a humane-like style and precision[2].

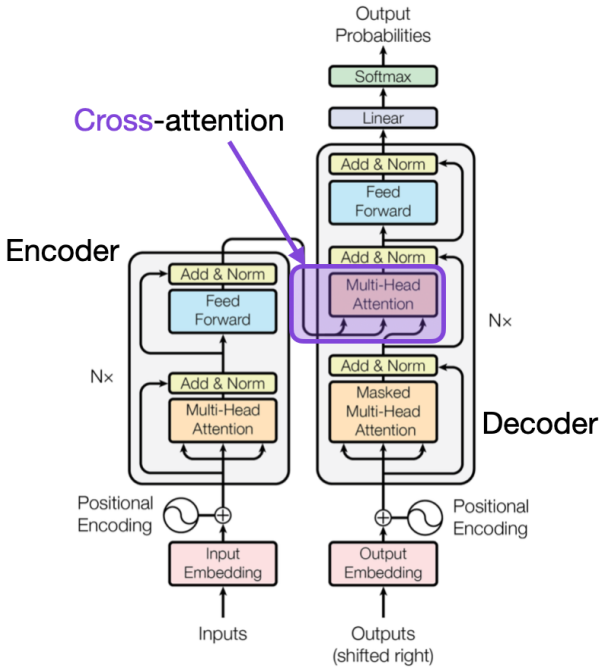

Figure 2.3: Encoder/Decoder architecture of transformers [2]

Furthermore, transformers apply "Add & Norm", allowing the neural network to go deeper without forgetting the original information. "Nx" describes the amount of times that the en-

coder/decoder block is executed. Positional encodings allow the position to flow over through all the computational processes. These techniques can be merged with medical NER to predict labels to text as these transformers learn about the features of the text in a bi-directional manner, then use this context feature knowledge and associate a sequence to a specific label. This sequential labelling is very useful in medical NER and allows for more detailed training techniques, such as boundary training. This architecture has shown the highest accuracies when applied in a medical context, making it the leading architecture, i.e. Hunflair[1].

2.1.2 Unsupervised NER

Unsupervised NER does not depend on pre-labelled data for model training. It aims to pinpoint patterns, structures, or clusters within the text data that may indicate the presence of named entities. Such is only possible by applying techniques like statistical models, clustering algorithms, or rule-based systems[18].

2.1.2.1 Clustering

Clustering is a technique used in machine learning and data mining to group similar data points based on specific criteria. There are various clustering algorithms; however, the most popular is K-Means Clustering, which divides and spreads data points into a centroid representing a particular data type and approximates them into each respective centroid iteratively. Clustering is used in customer segmentation, image segmentation, anomaly detection, and recommendation systems. However, clustering is not used directly in medical NER models as a training architecture it has been used for reducing the amount of irrelevant data in datasets such as MIMIC-III[19] by applying hierarchical clustering. The purpose of this application is due the need for manual annotation in datasets, this annotation process is time-consuming and expensive. In addition, it requires domain expertise, which makes manual annotation more difficult[20].

2.1.2.2 Rule-based Methods

Rule-based methods are a type of data mining that can be expressed as an IF-THEN rule and an inference step. Each rule consists of an antecedent (if part) and a consequent (then part). The antecedent specifies the conditions or criteria that must be satisfied for the rule to be triggered. The consequent specifies the action or conclusion when the rule is met[21]. This

method could be applied to medical NER through pattern matching of medical codes or through glossary-based matching. Medical codes are organized hierarchically, with codes grouped into chapters based on broad categories of diseases and conditions, these codes are in the form of "Axx.x" to "Zxx.x" or "0xx.x" to "9xx.x", where "x" represents a numerical digit. The first character denotes the chapter or section of the classification, while subsequent characters provide additional specificity[22]. Glossaries are specialized dictionaries that hold a list of words related to a specific medical domain and their respective definitions. This application allows text to be labelled without a labelled dataset.

2.1.2.3 Statistical Models

Statistical models are mathematical representations that analyse data, make predictions, or infer patterns. There are various types of mathematical representations, which can be descriptive or inferential. These models vary in complexity, from simple summaries like mean and median to complex algorithms like linear regression. One instance where a statistical model is applied in NER is with the application of a hidden Markov model (HMM). In HMM-based NER, each word in a sequence is associated with a hidden state representing the named entity label. The transition probabilities between states and emission probabilities of words given states are learned from labelled training data[23]. The application of HMMs in medical NER has been shown to achieve F-scores of 82% [24].

2.2 Medical NER

Medical NER's entities are specific to the healthcare domain. These labels encompass classes such as diseases, symptoms, medications, procedures and more. The vocabulary and terminology are highly technical and complex, requiring the model to understand and recognise complex medical terms accurately and interpret new nomenclature. Medical NER is also highly context-dependent, as many terms are ambiguous and have different meanings depending on the context. To create a medically accurate model a knowledge of the medical field is required which makes development challenging. On the other hand, most medical personnel do not have the necessary IT knowledge to create medical datasets for AI purposes. Medical NER's applications are particular and cannot be applied elsewhere, which, in turn, makes availability scarce[25]. Most medical data available for NER is catered towards Electronic Health Records (EHRs). As a

result, specific use cases such as Epsom's hospital hip/knee-surgery pre-operative patient notes and ER admission notes cannot be labelled correctly using the models trained with EHR data. This lack of adaptability is due to the inability to add class labels to an NER model without retraining it from scratch, and the goal labels in osteopathy are not the same as those in general medicine. Hence, the definition of the labels is of utmost importance when creating a dataset catered to a specific medical field. Furthermore, most EHR NER datasets are developed based on token labelling instead of sequential labelling and tend to label singular words without including quantifiers and qualifiers. This detail is essential in medical data and must not be overlooked. For instance, a "high fever" has a different meaning from a "low fever" and does not hold as much value as just labelling "fever" as a symptom. As aforementioned, NER is context-dependent and a common issue that arises in most medical data is the use of form-based data instead of paragraph-based. This renders most NER models unusable as their accuracy drops drastically with form data due to the lack of context preceding and following the labelling objectives.

2.2.1 DICE

DICE[26] is a generative model extraction technology that looks to extract important events from clinical texts, hospital admission notes and doctor analysis on patients that allow for the development of a dataset. This model extracts not only medical keywords but also boundaries and characterization words and medical. This type of extraction and dataset makes for a more efficient model that can recognize and compartmentalise words and give attributes to events. With the use of this model the dataset MACCROBAT-EE[27] was created. This dataset excels at boundary recognition for sequential labelling. This recognition, combined with the vast number of labelling classes, make for a detail-oriented dataset.

2.2.2 Semantic information retrieval on medical texts

Semantic information retrieval is a technique used to match words and rank them according to specific parameters such as semantic and meaning equality. This when applied to medical texts becomes a much more complex issue due to the vasts amount of data stored. In order for semantic searches to be effective, data must be pre-organized. This makes NER the perfect pre-step towards successful semantic information retrieval as label-based searching alongside semantic searches make for a promising filtering architecture [28].

2.2.3 Historical Documents and OCR

The preservation of historical medical data is essential. Most historical data is kept in a physical manner or low-resolution photo format. Due to this, the transition to digital text data can only be achieved through Optical Character Recognition (OCR). However, OCR can many times produce noisy outcomes due to the low-resolutions, font styles, document wear and low contrast between the text and background. If not properly trained, this noise can cause NER models to mislabel the documents. Therefore, the accuracy of the results will be directly linked to the noise amount and the model's ability to infer labels from context[29].

Chapter 3

Challenges & Dealing with these challenges

3.1 Lack of annotated data

3.1.1 Privacy concerns

The digitisation of medical data has revolutionised healthcare, improving efficiency and collaboration. However, it has also raised serious privacy concerns. The ease of sharing electronic health records has made them more vulnerable, increasing the risk of unauthorised access, intentional breaches, and cyberattacks.

Creating datasets for medical purposes is challenging without vast medical and technological knowledge. This process requires a team of professionals and is quite expensive. As a result, most databases are either protected by paywalls or require licenses that take months or even years to obtain. This makes it almost impossible for an average developer working on a time-sensitive project to acquire the necessary data.

Privacy laws are stringent globally, especially in the medical field. Medical data in the UK is covered by the Data Protection Act of 2018, which states that there must always be a valid lawful basis for the collection and processing of data, and the requirements of the Common Law Duty of Confidentiality (CLDC) [30] must also be met. Therefore, patients' medical data cannot be shared without prior legal consent. This process is unfeasible and costly as contacting each patient to produce data for AI research purposes would require extensive time and resources[31].

3.1.2 Paid software

Many software applications designed for annotation are proprietary, and their models are not publicly available. Due to the unique requirements of each hospital, medical facilities may be unable to purchase or license such software. Additionally, the model may not be optimised or aligned with the hospital's goals.

Annotation is a crucial task for creating datasets that are tailored to a specific objective. As there is a lack of technological expertise in the medical field, extensive research must be conducted in the most straightforward manner possible. This development serves as the foundation for a robust AI model. Instances such as TriMed[32], MetaMap[33], and cTakes [34] are paid software that requires some IT knowledge to set up and may not rectify specificity issues. All of their models are private, and the software itself is protected by Unified Medical Language System (UMLS) Certificates.

Testing these applications requires the hospitals to purchase the software. However, after testing, they may more often than not conclude that it does not perform specifically to their needs, either because it lacks a specific label that they require or because their data format is different from what the model was trained on.

3.2 Dealing with a lack of annotated data

3.2.1 Research of available data

There is barely any annotated data, as hiring medical personnel to annotate raw text is costly. Most datasets are also private and under either a paywall or a license, which is time-consuming to achieve. For instance, n2c2[35], MIMIC-III[19], i2b2/VA NLP[36], BioCreative/OHNLP[37] and ShARe/CLEF eHealth Evaluation Lab[38] are all protected behind a licensed wall.

However, some data, such as MES-Twitter[39], MACCROBAT-EE[27], Medical NER[40], and CORD-19[41], are still available to the public for developmental and research purposes. Most other public models do not target actual development but only research as they are not English-based and only target testing NER technology development to improve efficiency without the exact change in dataset data. The use of all of the datasets listed above allows for comparison and cross-test data evaluations, which results in a more generalised result that considers adaptability. The source of the data must be credible and medically accurate. If these requirements are not

met, the data will be deemed unusable.

3.2.2 Curated data

Each dataset is limited to a specific practice of the medical world. Most datasets focus on the COVID-19 era as it was the prominent boost in the digitisation of information, and other more recent datasets cater to EHR. This transition also required small-practice clinics and hospitals to transition. As a result, datasets started being developed to help with data extraction; however, this data is mainly curated towards COVID-19 symptom labelling and is not aimed towards general medicine or specific areas of the medical field such as osteopathy. To deal with this, developing a context-based NER model is necessary as it will englobe any medical field without depending on lexicology or terminology. One of the datasets that focuses on this generality is the MACCROBAT-EE[27] dataset.

3.3 New concepts

The medical field is dynamic and characterised by continual advancements, which regularly introduce a constant influx of new medications and nomenclature. This ever-evolving system poses a significant challenge to developing NER models, which are not catered towards context. This nomenclature introduction renders most medical NER models outdated and hinders their adaptability. For instance, public databases such as MES-Twitter[39] and Medical NER Dataset[40] cannot classify intricate and updated medical data. These datasets can be considered static and, with this unchanging nature, cannot deal with the fluidity of the medical field terminology, leading to limitations in accurately identifying and categorising the latest entities. Medical terminology may hold different meanings depending on the context of the sentence. Such a problem is noticeable in such models' precision, recall and F-score evaluation.

3.4 Noisy input

One common factor in big-data scenarios is noise. This noise can take various forms, from digitalising old documents through the scan and OCR [29] of such documents to newly digitally typed raw data with spelling mistakes. These mistakes can affect the number of false/true

positives the model produces, causing it to "hallucinate"[42].

3.4.1 Spelling mistakes

Spelling mistakes are hard to counter and are the most common type of noise in any text-based data. This type of noise is prevalent in very technical-specific language, such as medical nomenclature. Running a simple spell checker might not be enough as the cosine distance between the desired and misspelt words might be farther apart than another word depending on the spelling mistake. This would cause the data to auto-correct to the wrong word and not solve the noise factor[43]. This noise can significantly cause variations in the accuracy of a model. In the medical field, mislabeled data is a considerable concern. It may cost the hospital enormous amounts of money as a simple misdiagnosis with a tragic ending results in multiple lawsuits [44]. As the model will be directly linked to the search for associated cases with a patient, it will drastically affect the outcomes of each instance. For instance, due to a spelling mistake a label might be overlooked by the NER model and rank lower in a TF-IDF basis, which, in turn would result in the medical personnel not finding the case which might be the perfect match with their case.

3.5 Dealing with new concepts & Noisy input

3.5.1 Context NER

As aforementioned, noise is commonly present in raw text. This noise can be hard to decipher without context as it depends on vector cosine distance. This distance does not account for context and can lead to miss-corrections. Context-based NER is the most efficient model architecture for dealing with noisy input. It does not depend on the word itself to label it; instead, it predicts the labels based on multiple factors, such as context. Cosine distance is better suited for searching close terms rather than correcting words. By basing the word on the context, there is a higher chance of correct labelling and better information clustering. MACCROBAT-EE[27] also focuses on context-based data, clustering words and containing both general and medical labels[26]. Bidirectional transformers are best suited for this task as they learn text features and try to predict token features based on the text before and after that token. This architecture is present in BERT models[45].

3.6 Lack of IT knowledge in the medical space

Most medical personnel do not have the necessary knowledge to operate intricate technological systems such as terminal-based databases, Python-invoked programs, library-dependent applications, etc. This lack of expertise in these systems appeals to the development of over-simplified systems with vast computing power and automation behind every simple GUI component. This balance between complexity and simplicity creates a fine line relative to each individual. An efficient parallel computational system that can adapt to the hospital's needs is complex, and such architecture takes time to develop. The presence of easily readable statistics, access, and privacy-concerned features is a multi-knowledge task. The simplification of complexity without the loss of features is vital[46]. Most tasks must be done in the back end of the system separating the medical personnel from the technicalities of the program. The application must be reliable, fast and fully automated. For instance, medical personnel should not have to look up information about programming in order to run an app, this should be as easy to set-up as a few commands which would be done by the IT department and the medical personnel should access the tool in form of a website.

3.7 Dealing with lack of IT knowledge in the medical space

To deal with this, simplicity was introduced in the project. A simple web app capable of processing text through the NER model and storing the results visually and in a database. This rectifies the need for technological expertise, as a simple docker container will handle deployment and configuration. An annotation application alongside the main one will help doctors expand the models without having to learn how JSON objects or any data formatting works. Ideally, every desired feature must be available in a GUI format without being too complicated to use. Automation is vital to allow for ease of use.

Chapter 4

Implementation

4.1 Model training

4.1.1 Spacy NER

Spacy[47] is a Python library that trains a NER model using configuration files. It is well-optimised and facilitates the model's implementation. Spacy uses statistical models to run predictions based on current weight values obtained during training. Training in Spacy is an iterative process that uses reference annotation to calculate the loss gradient. This gradient of the loss is used to calculate the gradient of the weights through backpropagation[3]. It excels

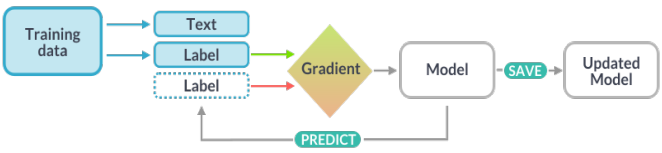

Figure 4.1: Spacy architecture [3]

at context learning as it does not memorise the words and their labels but learns text labelling based on context, no matter the token. It uses Adam [48] as the optimisation function (5.1) and

RoBERTa [49] as a base model.

$$\begin{aligned} g_t &= \nabla f(w_{t-1} - \beta m_{t-1}) \\ m_t &= \beta m_{t-1} + \eta g_t \\ w_t &= w_{t-1} - m_t \end{aligned} \tag{4.1}$$

The equation (4.1) describes the update rule for the weights w_t of the Adam gradient descent optimization algorithm with momentum. g_t is the gradient of the objective function with respect to the weights w_t at the previous iteration $w_t - 1$. It indicates the direction of the steepest ascent in the parameter space. m_t is the momentum term, which is a moving average of past gradients. It helps accelerate convergence by adding a fraction β(momentum coefficient) of the previous momentum $w_t - 1$ to the current gradient g_t. The momentum term introduces inertia into the optimization process, allowing the algorithm to continue in the same direction if the gradients maintain a consistent directionality over time. η represents the learning rate. In other words, Adam computes adaptive learning rates for each parameter by considering the gradients. It adjusts the learning rates for each parameter, allowing it to deal with sparse gradients and noise objectives. Overall, Adam is responsible for optimising the process of updating weights during training to minimise the loss and improve the model's performance [48]. RoBERTa is used as a base model, meaning the trained model will import RoBERTa's ability to identify text features and prediction abilities and merge it with the task of labelling.

4.1.2 MACCROBAT-EE

MACCROBAT-EE[27] is a dataset focused on biomedical terms that also considers the arguments of an event, not solely classifying single-word terms but the whole medical event. It was created using a Data-Efficient Clinical Event Extraction with Generative Models (DICE) [26].

	Classes	Samples
MACCROBAT-EE	41	3634
MES-Twitter	7	7 000
Medical NER	3	31

Table 4.1: Dataset statistics

A model trained on a multi-word labelling dataset identifies qualifiers as a part of an event. For

instance, "chest computed tomography" would be classified as a diagnostic procedure instead of just "computed tomography". In a single-word labelling dataset, "chest" would be a biological structure, computed would be discarded and "tomography" would be classified as a procedure. This factor enables the clustering of data in a more detailed manner. Qualitative symptom descriptions being considered when clustering allows cases to be more closely compared as often these enable medical personnel to reach conclusions more effectively.

This dataset has also been carefully reviewed to avoid overlapping and contradicting labels, which can confuse a model's weights. The dataset has also been tested to ensure an excellent accuracy-sample ratio, meaning the model will learn more without having an immense number of samples. MES-Twitter[39] contains more samples; however, the data must be treated extensively as it has repeated labels inside the same annotation. The lack of classes limits the model's use cases, as detail is vital in the medical space.

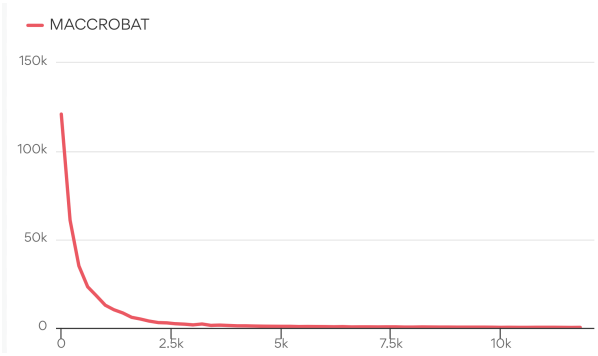

Figure 4.2: MACCROBAT loss/steps graph

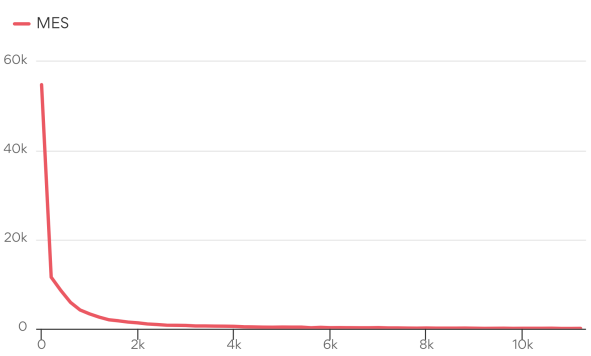

Figure 4.3: MES-Twitter loss/steps graph

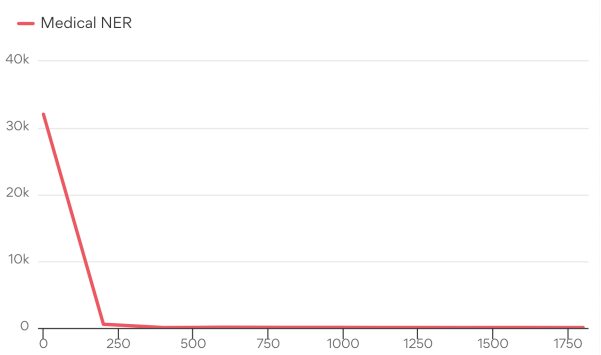

Figure 4.4: Medical NER loss/steps graph

The Medical NER dataset[40] overfits the NER model due to the lack of data (Figure 4.4), this overfitting is characterized by the sharp stagnation of learning instead of a logarithmic curve. This will cause the model not to be able to generalise its knowledge of new data as it captures noise and is overly complex. MES-Twitter's graph (Figure 4.3) shows less overfitting. However, there are still some traces of it as the curve is not fully logarithmic as the curve is not smooth and has sharp changes. MACCROBAT-EE's graph, even with less data, still manages to approximate itself to what the logarithmic curve should look like indicating it learns progressively and is adapting the base model to the new knowledge.

4.2 Model testing

There are many ways of testing an NER model. Each NER application has its benchmarks. A more generalised test of the model's performance is calculating precision, recall and f-scores. However, other factors need to be considered when applying NER, such as its ability to label words that are not in its vocabulary. Furthermore, a need for a model to perform equally as well when common spelling mistakes are made is vital.

4.2.1 Dataset differentiation

Datasets	Precision	Recall	F-Score
MACCROBAT-EE	99.30	99.30	99.30
MES-Twitter	38.72	40.05	38.59
Medical NER	33.80	49.42	40.15

Table 4.2: Dataset test metrics

MACCROBAT-EE[27] performed better overall, although it contained less data than the MES-Twitter[39] dataset. This could be attributed to data usefulness. MES's data is based on data scrapping done to Twitter public tweets with specific mentions of COVID and other medical words. These can be very inconsistent, as many may contain medically irrelevant information, vulgar language, and noise. Medical NER's data[40] is well formatted but in way lesser quantities than MACCROBAT-EE[27] and MES-Twitter[39]. MACCROBAT-EE[27] performs perfectly in labelling most classes. The classes that seem to be the weakest links are "DE-

TAILED DESCRIPTION", "DISEASE DISORDER", "CLINICAL EVENT", "LAB VALUE", "SIGN SYMPTOM", "DISTANCE", "OTHER ENTITY",

Classes	Precision	Recall	F-Score
DETAILED DESCRIPTION	99.91	100.00	99.95
DISEASE DISORDER	99.90	99.90	99.90
CLINICAL EVENT	99.80	100.00	99.90
LAB VALUE	99.64	99.59	99.61
SIGN SYMPTOM	100.00	99.96	99.98
DISTANCE	100.00	99.00	99.50
OTHER ENTITY	83.33	83.33	83.33

Table 4.3: MACCROBAT-EE class test metrics

"OTHER ENTITY" was the worst-performing class due to its ambiguity. The NER model also seems to have issues identifying the labels of numerical values, as these require precise context and cannot be distinguished independently.

4.2.2 Spelling mistakes effect

Spelling mistakes can have a significant effect on understanding a word. While most mistakes can be easily discerned, others may lead readers to assume a completely different word was intended to be written. In the medical field, it is crucial to accurately determine what the correct word is supposed to be, as it may affect analysis outcomes. To calculate the effects of spelling mistakes on the model and the average success rate, the following formula was used:

$$ASR = \frac{\sum_{i=1}^{n}(label_n^c) \times \sum_{i=1}^{n}(S_c(A,B)_n)}{\sum_{i=1}^{n}(label_n) \times \sum_{i=1}^{n}(S_c(A,B)_n^c)} \tag{4.2}$$

ASR is the Average Success Rate function that takes into account the cosine similarity [50] of the correct (A) and misspelt word (B) to calculate how the model performs based on how "wrong" the word is. It takes the sum of the correct labels ($label_n^c$) and multiplies it by the sum of the cosine distances of the words mislabeled ($S_c(A,B)_n$). It averages out through the aforementioned division by multiplying the opposing values ($S_c(A,B)_n^c$, $label_n$). This equation predicts the average success rate in n misspelling variations. The cosine similarity function is as follows:

$$S_c(A,B) := \frac{A \bullet B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \bullet \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.3}$$

Testing was performed on a variety of base words and lengths. For each word, ten common misspellings were tested on a template sentence: The patient has a biological structure, and it is his *BIOLOGICAL STRUCTURE*.

Word	ASR
hip	100%
head	94.3%
heart	100%
thorax	100%
stomach	100%
Score	98.86%

Table 4.4: Average success rate per word

The word misspellings for each word were the following:

```
hip -  hipe, hup, hep, hiip, hhip, hippe, hipp, hipo, hopp, hipps
head -  hedd, hed, haeed, heaad, haed, hedda, headd, heed, hede, hedd
heart - heartt, heert, hearht, hart, hert, heatr, haert, hearrt, hheart, hear
thorax - thoraks,thorx,thoraks,thorayx,thorx,throrax,thoraz,toraxh,thoraks,thoracs
stomach |- stomacch,stomahc,stomac,stomahch,stomch,stmacoh,stommach,
stomaach,stomatch,stomachh
```

4.2.3 Context variation effect

To understand why the *ASR* scores are so high, another theory needed to be tested to discern whether the model was intrinsically able to spot spelling mistakes or if there was another reason for it being able to label words with such accuracy. The concept of context variation is the concept in which the words are not labelled for their meaning but rather for the context leading to the word. To test this theory, changes were made to the template sentence.

```
Test sentences:
1- The patient's WORD.
2- It was identified in a patient a broken WORD.
3- The doctor diagnosed a rare condition affecting the patient's respiratory system in
his WORD.
```

`[...]`

`8- The imaging results indicated an anomaly situated in the central region of the patient's WORD.`

Each sentence was tested with different biological structures, and it was found that the sentence with the least context has a success rate of *75.9%*. The more context there was, the higher the success rate.

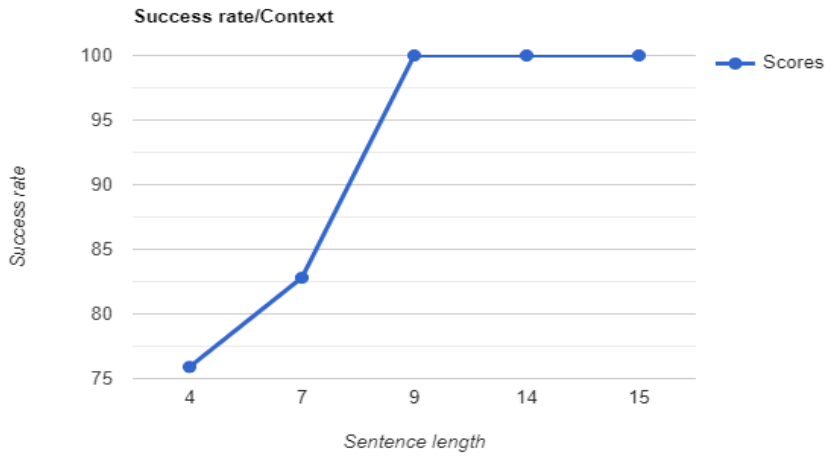

Figure 4.5: Graph of success/context

The graph(Figure 4.5) indicates that the model does not consider the word itself. It considers the context and predicts the label based on that context association. Furthermore, it shows the direct correlation of the success rate with the sentence length. This correlation means that the more context there is before the target sequence, the bigger the chance that the label will be correct.

4.3 Back end

The back-end was built in Flask [51]. Flask is a Python library that acts as an Application Programming Interface (API) for web apps. It fetches the models from HuggingFace[52] and loads them; the client can then parse POST requests with text, which the API will receive, process and return the model results to the client. The API also holds a thread that checks

whether the models have been updated through MongoDB[53]. This Mongo database holds all the training data for each model and a check table that informs the back-end whether new data was introduced to the models.

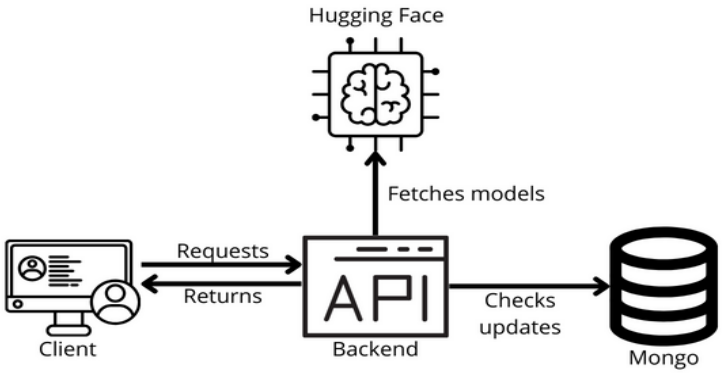

Figure 4.6: Backend architecture

If there is an update, the back-end will fetch the updated models without the client facing any downtime, as it is done through threading. MongoDB is different from other databases in that it does not have tables but holds objects. This makes data manipulation easier, as the data comes in dictionary arrays.

Algorithm 2 Data from check collection (MongoDB)

```
{
    last_check : "03/05/2024  16 : 36 : 38"
    updated : false
}
```

The API knows there was an update through the "updated" value. After this value is checked and the update is performed, the value is updated to *false*. This prevents the API from constantly fetching the model from Hugging Face for performance reasons. Another feature that the API holds is that it saves all the previously processed data on MongoDB. If someone needs to check a previous query, they can do so by accessing the database's *processed* collection.

4.4 Front end

The front end is simple and efficient. It is connected to the back end through NGINX, a proxy service that routes the back end to the same port as the front end. It is a simple page with a text box, a model selector, an *apply* button and a *Data annotator* button. It was developed using React for dynamic tasks and performance. Once a user presses the *apply nlp* button, the

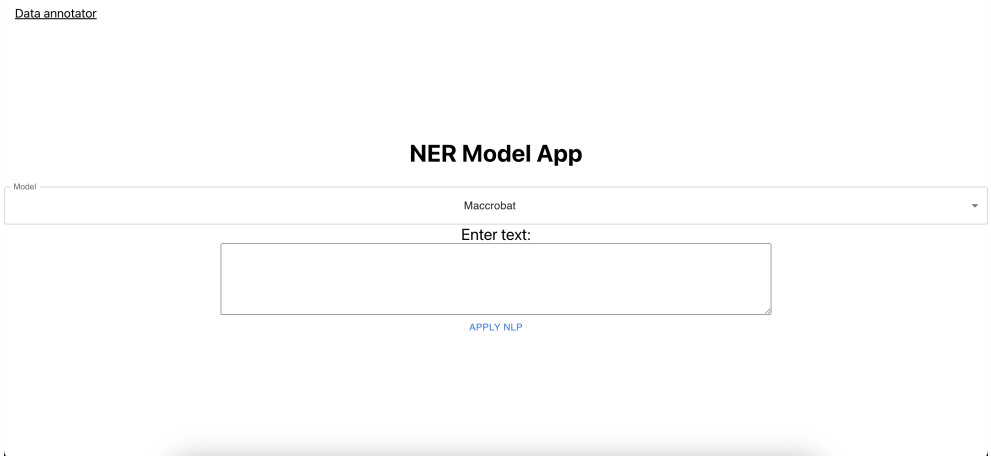

Figure 4.7: WebApp front-page

client sends a *POST* request to the back-end containing the model name and the text in the text box. Upon the arrival of the response, the client will offer the user a tab containing *Text, tokens, entities and statistics*, which allows the user to visualise the data returned. The text tab shows the text being processed. The tokens tab demonstrates a table with the text split into tokens and each token's respective start and end indexes. The Entities tab shows a table with the tokens that were labelled. Each labelled entity has a specific colour, depending on its label. The statistics tab shows a pie chart with a count of tokens per label. The front-end's simplicity does not contradict the accessibility to features, as they are all accessible through the client.

TEXT TOKENS ENTITIES STATISTICS

ID	Token Text	Start Index	End index
0	John	0	4
1	Doe	5	8
2	a	8	9
3	45yearold	10	11
4	male	12	14
5	office	14	15
6	clerk	15	19
7	presents	19	20
8	with	20	23

Figure 4.8: Tokens tab

TEXT TOKENS ENTITIES STATISTICS

ID	Text	Label	Start index	End index
0	45-year-old	AGE	12	23
1	male	SEX	24	28
2	office clerk	OCCUPATION	29	41
3	presents	CLINICAL_EVENT	43	51
4	persistent	DETAILED_DESCRIPTION	57	67
5	frontal	BIOLOGICAL_STRUCTURE	68	75
6	headaches	SIGN_SYMPTOM	76	85
7	7/10	QUANTITATIVE_CONCEPT	92	96
8	pain scale	DIAGNOSTIC_PROCEDURE	104	114

Figure 4.9: Entities tab

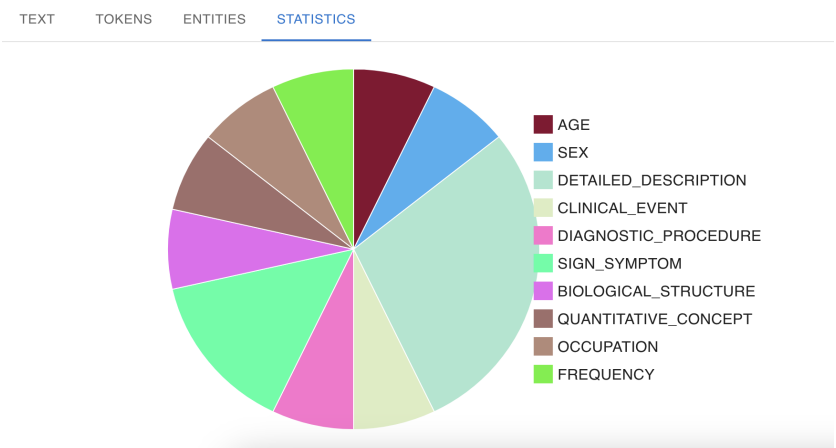

Figure 4.10: Statistics tab

4.5 Data annotator app

The data annotator app has many features that allow the client to annotate raw data inside a text file. These features include importing text files, tags, and already annotated data and exporting tags and the annotated data in the app. This app allows medical personnel to annotate data more efficiently without programming knowledge. This app enables the project to be dynamic and updates the model's training data depending on the hospital's new requirements. Furthermore, it would allow medical personnel to update the models to account for new data formats according to their needs without contacting the original developer to create a new model just for that task. The user would import the text file containing the text that needs to be labelled, create the desired labels, label the tokens respectively and then export the results using the *Annotations export* button. This would then download a formatted JSON file with all the labelled entities. The user would then upload this data into the database, and the back end would pick up on this data and train the model with it [54].

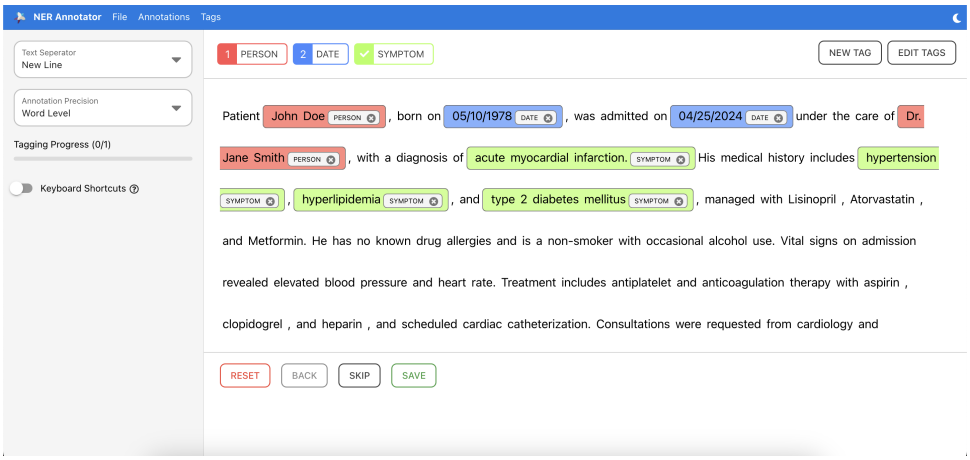

Figure 4.11: NER Annotator app

4.6 Automation architecture

A cron job was developed to run parallel to the back-end and front-end to simplify the web app further. This job is responsible for detecting if new data was introduced into the data collections of each model and training them on the new data. It also uploads the new models into Hugging Face and updates the MongoDB "updated" value by setting it to *true*. It uses the *last_ check*

value and compares it to each object inside a model's collection to see if any new data was introduced since the last check. Once all the operations are completed, the job will update the *last_ check* value to its last start time. This update ensures no data goes missing and prevents racing conditions from happening. The job also ensures no racing conditions can be introduced as the data is stored before processing the data inside the database; otherwise, depending on the training time, if data were to be uploaded into the database, it would be ignored in the next check.

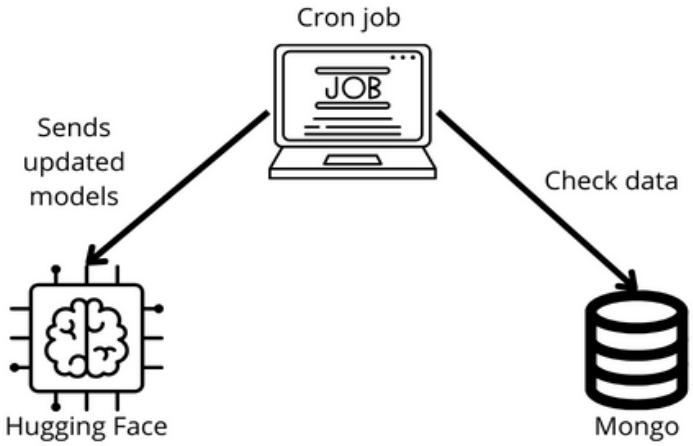

Figure 4.12: Cron representation

4.6.1 Multi-stage training

The corn job performs multi-stage training[55], constantly training the model with new data. This enhances the performance of the training task, as it does not require training from the base. This saves time as the project manager does not have to worry about creating a dataset, merging data, learning about model training and uploading models into Hugging Face. Multi-stage training is practical as the model does not lose the base weights, rather it adapts them to account for bias. In case a hospital notices a lack of labelling or changes its data format, it can produce labels accordingly and feed them into the database, and the multi-stage training will make the model perform accordingly.

4.7 Containerization

As a final touch of simplicity, everything was containerised using Docker[56] and routed through NGINX[57] to make the different apps act as one. This further tackles the lack of IT knowledge in the medical space, as no dependencies need to be installed to run the client and server. This Docker container is responsible for installing all the dependencies and running and routing the different applications.

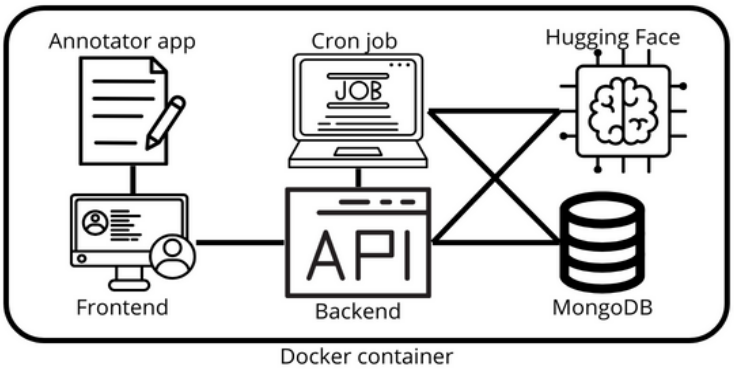

Figure 4.13: Docker container

Instead of running each app separately and finding dependencies that match all of the apps, Docker will run a container for each app with its required dependencies. Then, a process called compose merges all the containers into a bigger container. Therefore, to run all the applications, no IT knowledge is required besides copying and pasting commands specified in the GitHub README file.

Chapter 5

Conclusion & Further work

5.1 Conclusion

To conclude, I have conducted a novel NER model capable of labelling medical data and general data. The model can group tokens into a single label, enhancing the labelled data's search capabilities. The Spacy architecture combined with the MACCROBAT-EE[27] dataset has excelled in all the tests. The model successfully tackles the noisy input challenge by relying on context-based label predictions. It can also accurately label tokens with an acceptable context length threshold. The other datasets are not accurate enough, nor are they catered towards general and medical labelling, rendering them useless. The testing performed in the different models considers the challenges that affect medical NER. The model was developed based on the assumptions of the needs of the Epsom Hospital in order to make it as compatible as possible with the available data. The project was supposed to be originally developed on data provided by the Epsom Hospital and tested on a subset of that same data. Unfortunately, legal and ethical clearances were not done in time. Therefore, the model was developed as best as possible considering the Epsom Hospital's objectives and other hospital's needs. The web app uses the novel model's power and allows users to test their data on the worse-performing models. This power is combined with careful GUI simplicity. The web app can apply labels to unstructured text and displays the outcomes of this labelling in easily readable statistics. The app's back end is powerful and automates complicated processes such as multi-stage model training and dynamically updating models. The web app is enveloped in a complex docker container file that allows users who want to deploy the app locally not to worry about learning the specifics of

the technologies employed in the project. This automation, therefore, tackles all IT knowledge challenges issues in a medical setting.

Objectives mentioned in (1.2)	Completed
Explore the challenges of utilising AI in a medical setting and the difference between medical and non-medical NER.	✓
Introduce a dataset optimised towards sequence labelling and medical annotation.	✓
Introduce state-of-the-art NER model architectures that overcome the challenges of medical NERs.	✓
Propose a novel medical NER model and conduct a detailed analysis of the model's behaviour when faced with sequence labelling, noise and context variation.	✓
Introduce a Web App that leverages the proposed NER model and integrates it with a simple and automated interface and integrates a data annotator feature with the app.	✓

Table 5.1: Objectives table

5.2 Further work

5.2.1 New data

In case a hospital provides its input of data and objectives, it can adapt the MACCROBAT-EE[27] model to its needs by inputting its data. However, if it does not satisfy their needs, the option of integrating the web app with a completely new model is available. They are not limited to the models developed in this project. With the data from the Epsom Hospital, a new model catered towards hip/knee surgery pre-operative patient notes and ER admission notes could be created. This model would have to undergo the same analysis to determine its behaviour with labelling.

5.2.2 Web app improvements

The web app could be developed further. Visual improvements could be made, and further automation to the data annotator app could be integrated to automatically upload the data to the Mongo database in the specific model collection. A more comprehensive statistics tab could

be created to better visualise the results of the NER model data. A feature for uploading PDF files could be developed, as well as ZIP files that contain a list of PDF files containing medical data. The transition to more easily import new models needs to be developed either through a YAML file that is easy to understand and edit or through an automated system that would detect the collection names and match the different models uploaded to a specific Hugging Face repository created for the project. Finally, a feature to train models by uploading a JSON file through the client side that would allow to select a model name and train it automatically in the back end would further help with the lack of IT knowledge in the medical field.

5.2.3 Supervised approach

Some hospitals hold their information in a form rather than descriptive text. This rules out the usability of context-based labelling. Creating an extensive dataset that accounts for both form-based labelling and context-labelling would be the best approach towards having a better-performing NER model. Integrating a text generator such as GPT to produce medical forms where the values of each form key are from a medically approved glossary would allow the production of extensive datasets with randomness and accurate, medically accurate labelling. Merging this form data with context data from MACCROBAT-EE[27] would make for a dataset that theoretically labels both formats. The data should be equally extensive to avoid biases towards a specific format. MedSpacy[58] is another architecture in Spacy that is catered towards medical NER, and it could be used to improve the model's outcomes. Its features include sentence splitting, section detection, asserting negation, family history, and uncertainty.

5.2.4 Unsupervised approach

Unsupervised learning is gaining traction within the Artificial Intelligence community as a standard approach for developing AI models. This method has demonstrated exemplary accuracy scores by learning directly from data features rather than relying on pre-labelled data. Medical NER's main challenge is the scarcity of available data due to privacy constraints. However, unsupervised methods offer a compelling solution, as they do not require pre-labelled data for training, thereby circumventing privacy concerns and enhancing the scalability and effectiveness of NER systems in the medical domain.

Instead of using already-labelled data, defining entity classes holding semantic groups, types,

and concepts merged with UMLS standards to create seed terms would act as class signatures for labelling. These signatures would create similarity scores within chunks of medical corpora data. The highest score would be the label defined if passed above a certain threshold[4].

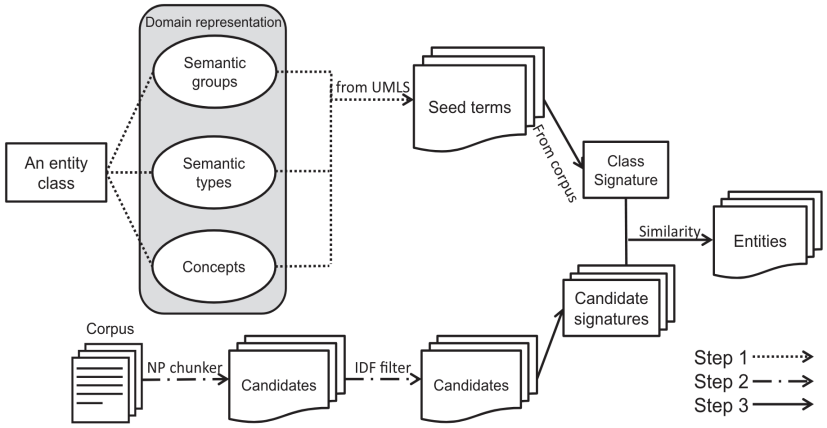

Figure 5.1: Unsupervised architecture [4]

Chapter 6

Legal, Ethical, Social and Professional (LSEP)

Throughout all the stages of this dissertation, ethics and its implications are carefully taken into account, including legal, social, ethical and professional considerations. The works conducted in this study comply with ethical and legal principles set out by the law, government and relevant third parties. Additionally, this aligns with the University of Surrey's ethical practices and code of conduct, including protecting the University's reputation and respecting the welfare and interests of the wider community. Additional guidance that this research significantly relies on is the British Computer Society's code of conduct. This provides vital professional principles that set out ethical and appropriate behaviours in the community, including public interest, professional competence and integrity, duty to the relevant authority and duty to the profession. Therefore, it can be ensured that this dissertation is carried out according to legal, social, ethical and professional principles, contributing to society and human well-being.

6.1 University Ethics Process

6.1.1 What data is to be collected?

The data used during this research is obtained from online sources available to the public for both non-commercial and commercial use. The datasets used are called MACCROBAT-EE[27], Medical NER (Kaggle)[40] and MES-Twitter[39], which contain generated medical data and

public Twitter posts

6.1.2 How data is stored?

The data is stored in a database protected by a username and password. Furthermore, no sensitive data is being used in the project. All the datasets are public.

6.1.3 Ethical Review

This research has appropriately taken into account ethics. All ethical data was carefully analysed and determined to be generated without revealing any personal information and is purely fictional.

6.2 Legal & Ethical

6.2.1 Public Interest & Do no harm

This dissertation was developed with usefulness in mind. It is ensured that it serves the public interest without any discrimination. The proposed model was designed to be helpful in various sectors of the medical field. This technology is easily accessible as the source code is publicly available through GitHub. Also, all the works in this study are conducted according to the laws and relevant legislation. This includes the Computer Misuse Act 1990, which protects personal data from unauthorised access and covers various computer misuse offences. No illegal activity was committed at any stage of the research. No sensitive information is present in the datasets as they are functionally generated and do not contain any personally identifiable information.

6.2.2 Informed Consent

There is no active human participation in this project. Hence, it is not applicable.

6.2.3 Confidentiality of Data

No confidential data is used or stored during the development of this project. The data is generated without any access to private and confidential information. Furthermore, the data is

stored in a password-protected database. Hence, it is not applicable.

6.2.4 Transparency

All sources are appropriately identified during the development and description of this project. For instance, all the dataset sources are appropriately referenced. The ownership of all third-party data and software is acknowledged. This is to respect legal considerations of copyrights and intellectual property as a third party.

6.3 Professional Competence and Integrity

The British Computer Society sets out a code of conduct, which includes professional competence and integrity principles. The applicable principles are the following:

1. **NOT claim any level of competence that you do not possess.** All the works performed in this dissertation have been done to my fullest potential and to the best of my knowledge. I do not claim to have any competence beyond what I have.
2. **Develop your professional knowledge, skills and competence on a continuing basis, maintaining awareness of technological developments, procedures, and standards that are relevant to your field.** This point is corroborated through the extensive literature review and the employment of up-to-date technologies across the project.
3. **Respect and value alternative viewpoints and seek, accept and offer honest criticisms of work.** During the development, guidance from multiple people was received, mainly from my supervisor, Dr. Suparna De. This guidance was valued and respected and is reflective of the work.
4. **Reject and will not make any offer of bribery or unethical inducement.** No bribery or unethical inducement was offered or accepted.

6.4 Social Responsibility

The proposed work can be further applied to create a medical data management application capable of improving the data in a medical facility. It can label data that can later be used for

research and analysis purposes. However, some aspects of AI dependency need to be considered. The data passed through this system must be carefully reviewed to avoid misleading doctors who use the processed data in real-life medical cases. This misinformation can have devastating effects and lead to multiple issues, legally and ethically. The project should also be reviewed to store the data in an encrypted manner accordingly in case multiple systems share it.

Bibliography

[1] L. Weber, M. Sänger, J. Münchmeyer, M. Habibi, U. Leser, and A. Akbik, "Hunflair: An easy-to-use tool for state-of-the-art biomedical named entity recognition a preprint," 08 2020. [Online]. Available: https://arxiv.org/pdf/2008.07347

[2] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 06 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762

[3] "Training pipelines & models · spacy usage documentation," Training Pipelines & Models. [Online]. Available: https://spacy.io/usage/training#basics

[4] S. Zhang and N. Elhadad, "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts," *Journal of Biomedical Informatics*, vol. 46, pp. 1088–1098, 12 2013.

[5] R. Jackson, I. Kartoglu, C. Stringer, G. Gorrell, A. Roberts, X. Song, H. Wu, A. Agrawal, K. Lui, T. Groza, D. Lewsley, D. Northwood, A. Folarin, R. Stewart, and R. Dobson, "Cogstack - experiences of deploying integrated information retrieval and extraction services in a large national health service foundation trust hospital," *BMC Medical Informatics and Decision Making*, vol. 18, 06 2018.

[6] P. Malik, M. Pathania, and V. Rathaur, "Overview of artificial intelligence in medicine," *Journal of Family Medicine and Primary Care*, vol. 8, p. 2328, 07 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6691444/

[7] M. C. Durango, E. A. Torres-Silva, and A. Orozco-Duque, "Named entity recognition in electronic health records: A methodological review," *Healthcare Informatics Research*, vol. 29, pp. 286–300, 10 2023. [Online]. Available: https://e-hir.org/upload/pdf/hir-2023-29-4-286.pdf

[8] "Home," South West London Elective Orthopaedic Centre | SWLEOC. [Online]. Available: https://www.eoc.nhs.uk/

[9] N. Barney, "What is named entity recognition (ner)? | definition from techtarget," WhatIs.com, 03 2023. [Online]. Available: https://www.techtarget.com/whatis/definition/named-entity-recognition-NER

[10] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv:1508.01991 [cs]*, 08 2015. [Online]. Available: https://arxiv.org/abs/1508.01991

[11] "Distilling large language models into tiny models for named entity recognition corresponding author : Yining huang email: huangyining1987@gmail.com," arxiv.org. [Online]. Available: https://arxiv.org/html/2402.09282v3

[12] Parser, "Unleashing the power of generative ai: Understanding transformers," Medium, 09 2023. [Online]. Available: https://medium.com/@parserdigital/unleashing-the-power-of-generative-ai-understanding-transformers-de00e8689bb5

[13] R. Herwando, M. A. Jiwanggi, and M. Adriani, "Medical entity recognition using conditional random field (crf)," in *2017 International Workshop on Big Data and Information Security (IWBIS)*, 2017, pp. 57–62.

[14] A. Ekbal and S. Bandyopadhyay, "Named entity recognition using support vector machine: A language independent approach," vol. 39, 01 2010.

[15] Z. Ju, J. Wang, and F. Zhu, "Named entity recognition from biomedical text using svm," *5th International Conference on Bioinformatics and Biomedical Engineering, iCBBE 2011*, pp. 1–4, 05 2011.

[16] A. Amidi and S. Amidi, "Cs 230 - recurrent neural networks cheatsheet," Stanford.edu, 2019. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[17] L. Li, L. Jin, Z. Jiang, D. Song, and D. Huang, "Biomedical named entity recognition based on extended recurrent neural networks," in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015, pp. 649–652.

[18] J. Delua, "Supervised vs. unsupervised learning: What's the difference?" IBM Blog, 03 2021. [Online]. Available: https://www.ibm.com/blog/supervised-vs-unsupervised-learning/

[19] A. Johnson, T. Pollard, and R. Mark, "Mimic-iii clinical database," Physionet.org, 09 2016. [Online]. Available: https://physionet.org/content/mimiciii/1.4/

[20] X. Han, C. K. Kwoh, and J.-j. Kim, "Clustering based active learning for biomedical named entity recognition," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1253–1260.

[21] J. Fürnkranz, "Rule-based methods," *Encyclopedia of Systems Biology*, pp. 1883–1888, 2013.

[22] W. H. Organization, "Icd-10 version:2019," icd.who.int, 2019. [Online]. Available: https://icd.who.int/browse10/2019/en

[23] P. Padmanabhan, "Named entity recognition using statistical model approach," *International Journal of Computer Applications*, vol. 73, pp. 31–33, 07 2013.

[24] J. Frei and F. Kramer, "German medical named entity recognition model and data set creation using machine translation and word alignment: Algorithm development and validation," *JMIR formative research*, vol. 7, pp. e39 077–e39 077, 02 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10015355/

[25] H.-J. Song, B.-C. Jo, C.-Y. Park, J.-D. Kim, and Y.-S. Kim, "Comparison of named entity recognition methodologies in biomedical documents," *BioMedical Engineering OnLine*, vol. 17, 11 2018.

[26] M. D. Ma, A. K. Taylor, W. Wang, and N. Peng, "Dice: Data-efficient clinical event extraction with generative models," arXiv.org, 05 2023. [Online]. Available: https://arxiv.org/abs/2208.07989

[27] M. D. Ma, "derekmma/dice," GitHub, 04 2024. [Online]. Available: https://github.com/derekmma/DICE

[28] L. Tamine and L. Goeuriot, "Semantic information retrieval on medical texts: Research challenges, survey, and open issues," *ACM Comput. Surv.*, vol. 54, no. 7, sep 2021. [Online]. Available: https://doi.org/10.1145/3462476

[29] M. Ehrmann, A. Hamdi, E. L. Pontes, M. Romanello, and A. Doucet, "Named entity recognition and classification in historical documents: A survey," *ACM Computing Surveys*, 06 2023.

[30] D. of Health, "The common law duty of confidentiality | department of health," Department of Health, 05 2016. [Online]. Available: https://www.health-ni.gov.uk/articles/common-law-duty-confidentiality

[31] N. Yadav, S. Pandey, A. Gupta, P. Dudani, S. Gupta, and K. Rangarajan, "Data privacy in healthcare: In the era of artificial intelligence," *Indian Dermatology Online Journal*, vol. 14, p. 788, 11 2023. [Online]. Available: https://journals.lww.com/idoj/fulltext/2023/14060/data_privacy_in_healthcare__in_the_era_of.5.aspx

[32] "Trimed," shiny.dei.unipd.it. [Online]. Available: http://shiny.dei.unipd.it/TriMED/

[33] "Metamap versions," lhncbc.nlm.nih.gov. [Online]. Available: https://lhncbc.nlm.nih.gov/ii/tools/MetaMap/use-MetaMap.html

[34] "Apache ctakes™ - clinical text analysis knowledge extraction system," ctakes.apache.org. [Online]. Available: https://ctakes.apache.org/whycTAKES.html

[35] "National nlp clinical challenges (n2c2)," n2c2.dbmi.hms.harvard.edu. [Online]. Available: https://n2c2.dbmi.hms.harvard.edu/

[36] "i2b2: Informatics for integrating biology & the bedside," I2b2.org, 2022. [Online]. Available: https://www.i2b2.org/NLP/DataSets/

[37] "Papers with code - bc5cdr dataset," paperswithcode.com. [Online]. Available: https://paperswithcode.com/dataset/bc5cdr

[38] "Clef ehealth task 2013 – dataset | clef ehealth lab series." [Online]. Available: https://clefehealth.imag.fr/?page_id=441

[39] A. RAJAN, "Mes-twitter dataset." [Online]. Available: https://github.com/MedNer-Surrey/MedNer-FYP/tree/mes-cov/data

[40] "Medical ner," www.kaggle.com. [Online]. Available: https://www.kaggle.com/datasets/finalepoch/medical-ner

[41] "Covid-19 open research dataset challenge (cord-19)," www.kaggle.com. [Online]. Available: https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge

[42] IBM, "What are ai hallucinations? | ibm," www.ibm.com. [Online]. Available: https://www.ibm.com/topics/ai-hallucinations

[43] K. H. Lai, M. Topaz, F. R. Goss, and L. Zhou, "Automated misspelling detection and correction in clinical free-text records," *Journal of Biomedical Informatics*, vol. 55, pp. 188–195, 06 2015. [Online]. Available: https://core.ac.uk/download/pdf/82646657.pdf

[44] PinnacleCare, "The human cost and financial impact of misdiagnosis," Medium, 01 2018. [Online]. Available: https://pinnaclecare.medium.com/the-human-cost-and-financial-impact-of-misdiagnosis-b50ead6f53f4

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv.org, 10 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[46] M. O. Kim, E. Coiera, and F. Magrabi, "Problems with health information technology and their effects on care delivery and patient outcomes: A systematic review," *Journal of the American Medical Informatics Association*, vol. 24, p. ocw154, 02 2017. [Online]. Available: https://academic.oup.com/jamia/article/24/2/246/2723147

[47] spaCy, "spacy · industrial-strength natural language processing in python," spaCy, 2015. [Online]. Available: https://spacy.io/

[48] V. Bushaev, "Adam - latest trends in deep learning optimization." Towards Data Science, 10 2018. [Online]. Available: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[49] "Roberta," huggingface.co. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta

[50] W. Contributors, "Cosine similarity," Wikipedia, 03 2019. [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity

[51] Flask, "Flask documentation," flask.palletsprojects.com, 2010. [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/

[52] H. Face, "Hugging face - on a mission to solve nlp, one commit at a time." huggingface.co. [Online]. Available: https://huggingface.co/

[53] MongoDB, "The most popular database for modern apps," MongoDB, 2019. [Online]. Available: https://www.mongodb.com/

[54] Arunmozhi, "tecoholic/ner-annotator," GitHub, 04 2024. [Online]. Available: https://github.com/tecoholic/ner-annotator

[55] S. hedfi, "Update ner model with new data," Medium, 06 2023. [Online]. Available: https://medium.com/@sirinehedfi95/update-ner-model-with-new-data-604fa364486d

[56] Docker, "Enterprise application container platform | docker," Docker. [Online]. Available: https://www.docker.com/

[57] NGINX, "Nginx | high performance load balancer, web server, & reverse proxy," NGINX, 2018. [Online]. Available: https://www.nginx.com/

[58] "medspacy · spacy universe," medspaCy. [Online]. Available: https://spacy.io/universe/project/medspacy