A DISSERTATION
SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING

Improvement of Multimodal Deep Learning Predictive Models for Electronic Health Records

by

MUGISHA Chérubin

September 2023

The thesis titled

Improvement of Multimodal Deep Learning Predictive Models for Electronic Health Records

by

MUGISHA Chérubin

is reviewed and approved by:

Chief referee

Professor Date Aug. 5, 2023

PAIK Incheon Incheon Paik

Professor Date

Yong Liu Yong Liu Aug. 8, 2023

Senior Associate Professor Date

Xin Zhu Xin Zhu Aug. 8, 2023

Senior Associate Professor Date Aug 8, 2023

Cong Thang Truong Thang

THE UNIVERSITY OF AIZU

September 2023

Contents

List of Figures

List of Tables

List of Abbreviations

BERT Bidirectional Encoder Representations from Transformers

BioBERT Biomedical language representation model for biomedical text mining

BOW Bag of Word

BPE Byte Pair Encoder

CDSS Clinical Decision Support Systems

DL Deep Learning

DNN Deep Neural Network

EHRs Electronic Health Records

FL Fuzzy Logic

LMs Language Models

LSTM Long Short Term Memory

MIMIC Medical Information Mart for Intensive Care

ML Machine Learning

NLG Natural Language Generation

NLP Natural Language Processing

NLU Natural Language Understanding

SA Sparse Attention

TF-IDF Term Frequency-inverse Document Frequency

UMAP Uniform Manifold Approximation and Projection

UMLS Unified Medical Language System

Acknowledgment

At first, I am grateful to the Almighty, who is the source of all strength and wisdom. My sincere appreciation goes to :

- The Japanese Government for giving me the opportunity of a lifetime through the prestigious MEXT(Ministry of Education, Culture, Sports, Science, and Technology) Monbukagakusho Scholarship.
- My advisor, Professor Incheon Paik, granted me the chance to come to Japan and join the University of Aizu by accepting me into his lab. Through his unwavering support, encouragement, and insightful guidance, I acquired extensive knowledge in computer science and research, enabling me to accomplish this undertaking.
- The faculty members who agreed to assess this thesis not only heightened my attention to detail but also played a crucial role in enhancing my skills in scientific communication and research presentation through their invaluable feedback.
- The entire faculty and staff at the University of Aizu, whether through their classes or in everyday interactions, provided assistance and support that greatly contributed to my journey.
- I extend my heartfelt gratitude to my parents, NIMUBONA Nicaise and SINDUHIJE Adèle, for their unwavering dedication in raising me, guiding me every step of the way, and serving as my inspiring role models.
- My siblings, cousins (with a special mention to NDIKUMANA Scotty), and friends across the globe for their unwavering support, uplifting words, and constant encouragement. Their presence in my life has been instrumental in keeping me motivated and determined to strive harder each day.
- Lastly, I would like to express my sincere appreciation to the members of the Intelligent Analytics Laboratory at the University of Aizu for making our lab a very compelling environment for a fruitful academic journey.

Abstract

The digitization of health records has resulted in electronic health records (EHRs) becoming a crucial source of information for healthcare providers, offering valuable insights into patient health and enabling better-informed decision-making. EHR data are from various sources and are stored in structured or unstructured formats. Together, these data are the perfect representation of a patient. The multimodality of a patient representation should be implemented by tools that mimic the basic reasoning of a healthcare provider when analyzing the status of a patient in a specific period.

Recently, deep learning has shown impressive results in different predictive tasks. However, those models have two main problems. On the one hand, available models for transfer learning are not well adapted to the diversity of the data from the medical field. Jargon and conventional annotations make it harder for traditional NLP to be used effectively on raw medical narratives. The complexity, variability, and inconsistency of medical structured data make it challenging to integrate and analyze the data effectively and to extract meaningful insights using traditional supervised machine learning or ensemble models. On the other hand, models are monomodal and learn specifically from a specific single type of data. While these models can be effective at analyzing the specific type of data they were trained on, they may not be able to capture complex relationships between variables or provide a comprehensive view of a patient's condition.

To address these limitations, we are proposing a study on optimizing multimodal deep learning predictive models for electronic health records to ultimately integrate data from multiple sources such as vitals, medications, diagnoses, lab results, narratives, images, and other types of EHR data. Using the MIMIC-III (Medical Information Mart for Intensive Care III) database, a publically available EHR of more than 46,000 adult patients, we conducted our research as follows:

First, in Chapter 2, knowing that 80% of EHR data are in a free text format, it was imperative to analyze different methods which have led to transformative advances in supporting clinical decision-making with NLP. The main question of this chapter is how the data preprocessing and modeling impact the predictive performance of NLP of clinical text documents for mortality prediction. The task of outcome prediction is complex and requires the ability to capture complex patterns in language data. Neural language modeling pipelines have become increasingly popular due to their ability to capture such patterns, but there are many different approaches to neural language modeling, and it is not clear which approach is best for outcome prediction. Therefore, there is a need to compare different neural language modeling pipelines for outcome prediction to determine which approach is most effective. This chapter contributes to the field of NLP by comparing different neural language modeling pipelines for outcome prediction. The research evaluates several approaches, including, Convolutional Neural Networks(CNN), long short-term memory (LSTM), and transformers, and uses several evaluation metrics. The experimental results show that mild processing and

transformer-based modeling perform better than others for outcome prediction, especially when dealing with long-range dependencies between words. Due to their ability to capture context-specific information, contextual embeddings like BERT exhibit superior performance compared to traditional word embeddings like Word2Vec and GloVe. The significance of this research lies in its contribution to providing valuable guidance on the most effective approach to neural language modeling for outcome prediction tasks. Overall, this research advances our understanding of neural language modeling pipelines for outcome prediction and provides valuable insights for the following step in this research.

In chapter 3, Inspired by the Fuzzy theory of segmenting and representing continuous values into delimited ranges to represent a modellable entity, we proposed a novel approach to transform numerical data into natural language text. We aggregated the administration data, diagnosis, vital signs, procedures, and laboratories, and generated artificial narratives from the medical tabular data to describe the patient's period of hospitalization. The aim was to unify the accuracy and power of NLP models and the completeness of medical tabular data. We evaluated our approach on a downstream NLP text classification task to predict in-hospital mortality and demonstrated the importance and competitiveness of this approach. We compared our results with a tabular medical data benchmark publication and found that our best NLP model yielded competitive accuracy with tabular medical data. We believe that this approach of generating artificial narratives can open new paths for using NLP in the medical area for predictive tasks and overcome the variability and incompleteness of structured medical data. The significance of this research lies in its ability to improve the completeness and accuracy of NLP models in predicting inpatient outcomes while we found a solution for handling missing values, inconsistent formats, errors, duplicates, and data imbalance. However, one major limitation of this approach is the biomedical language models are not adapted to understand biomedical terminologies and handle the new length of the generated narratives.

Chapter 4 addresses this issue by proposing two new Biomedical language models trained from scratch to improving the representation of biomedical texts. This chapter reviews recent studies on optimizing language models for biomedical and clinical text. Challenges and opportunities associated with this task are highlighted and various techniques are developed to improve performance. Two language models optimized for biomedical and clinical data are presented, along with their evaluation and performance on several NLP tasks, including NER, Q&A, and sentence and token classification. The chapter explores domain-specific pre-training and fine-tuning techniques, including the importance of transfer learning and different tokenizers in improving semantic representation. The significance of hyperparameter fine-tuning is also discussed to improve model robustness on clinical and biomedical text. The potential impact of optimized language models on healthcare applications, including electronic health records and clinical decision support systems, is examined. Ethical considerations and challenges are highlighted by using raw clinical data from EHR with pre-trained language models. This chapter provides a step forward in optimizing language models for biomedical and clinical text and outlines future research directions.

Chapter 1

Introduction

1.1 Overview

The field of healthcare is rapidly evolving and technological advancements have played a significant role in improving the quality of care provided to patients. With the increasing digitization of health records, Electronic Health Records (EHRs) have become a crucial source of information for healthcare providers [2, 3]. EHRs contain a wealth of patient data, including medical history, lab results, prescriptions, and demographics. Analyzing this data can provide valuable insights into patient health and help healthcare providers make better-informed decisions.

The traditional approach to analyzing EHR data has been to use statistical methods to identify patterns and correlations. However, this approach has limitations, as it relies on pre-defined rules and assumptions, and is often not able to capture complex relationships between different variables. With the advent of deep learning techniques, there is an opportunity to move beyond traditional statistical methods and develop more sophisticated predictive models [4].

Deep Learning (DL) is a subfield of Machine Learning (ML) that involves training artificial neural networks to learn from large datasets. Deep learning models are capable of automatically extracting features from data, making them well-suited to handling large and complex datasets like EHRs [5]. There are many potential applications of deep learning in healthcare, including predicting patient outcomes, diagnosing diseases, and identifying risk factors for certain conditions [6].

In particular, the use of deep learning models to predict patient outcomes based on EHR data has the potential to revolutionize healthcare by enabling healthcare providers to identify patients at risk of adverse outcomes and intervene early to prevent them [6]. However, there are several challenges associated with developing deep learning models for EHR data. First, EHR data is often incomplete and contains missing values, making it challenging to train models effectively. Second, EHR data is highly heterogeneous, meaning that different data sources may be recorded in different formats, making it challenging to integrate them into a single model [7]. Finally, deep learning models are often considered ”black boxes” because it can be challenging to interpret the results they produce [8].

Despite these challenges, there have been many successful applications of deep learning in healthcare, including predicting readmissions after hospitalization [9], identifying patients at risk of sepsis [10], and predicting outcomes in patients with cystic fibrosis [11]. These studies have demonstrated the potential of deep learning to provide

valuable insights into patient health and improve the quality of care provided.

An accurate and robust EHR data mining approach has several potential applications, such as personalized medicine, disease prevention, and population health management. Our study on multimodal EHR data can enable more accurate and comprehensive predictive models, leading to improved healthcare outcomes.

1.2 Definitions

1.2.1 Improvement

The term "Improvement" refers to the process of optimizing the performance of the predictive models. In the context of deep learning, optimization can involve a data mining process [12], a model learning technique, or adjusting the model's parameters and hyperparameters to minimize the loss function and improve the accuracy of the predictions.

Several optimization techniques have been developed for deep learning models, including gradient descent, stochastic gradient descent, and adaptive learning rate methods [13]. In healthcare applications, optimizing multimodal deep learning models is essential to ensure accurate predictions and improve clinical decision-making.

One example of optimization in healthcare is the development of deep learning models to predict patient outcomes. In a study by Rajkomar et al., a deep learning model was developed to predict patient mortality using EHR data [14]. The model was optimized using a grid search to find the best hyperparameters for the model, resulting in improved prediction performance.

1.2.2 Multimodality

The term "Multimodal" refers to the use of multiple modes or types of data in building predictive models. In electronic health records, multimodality can refer to integrating various types of data, such as medical imaging, laboratory test results, clinical notes, demographics, and medication information, among others [15]. These data types may be combined to improve the accuracy and reliability of predictive models, as different types of data may capture various aspects of a patient's health status. By leveraging multiple modalities, multimodal models can provide a more comprehensive and holistic understanding of patient health, leading to better clinical decision-making and improved patient outcomes. One example of a multimodal approach is the combination of medical imaging and clinical data for disease diagnosis and prognosis. In a study by Chassagnon et al., deep learning models were developed to perform disease quantification, staging, and outcome prediction using both computed tomography (CT) images and clinical data [16]. The multimodal approach improved the prediction performance compared to models that used only one data type.

Another example is the combination of genomic and clinical data to predict drug response. In a study by Baptista et al., a multimodal deep learning model was developed to predict drug response in patients with cancer using genomic and clinical data [17]. The model outperformed traditional machine learning models and demonstrated the potential for multimodal approaches in precision medicine.

However, developing and optimizing multimodal models for healthcare applications can be challenging due to the heterogeneity and complexity of EHR data. Data prepro-

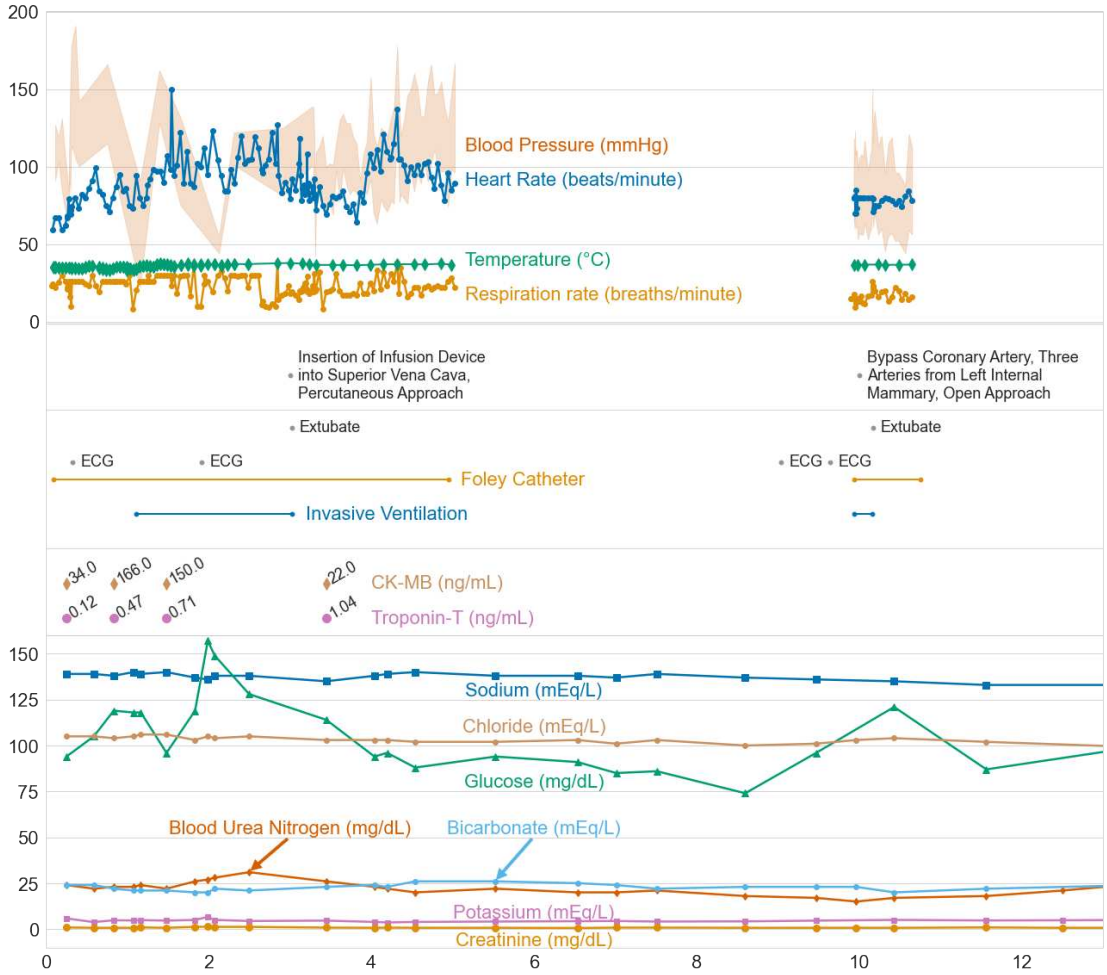


Figure 1.1: An illustration of EHR data on a patient in ICU from MIMIC3 Database

cessing, feature selection, and model optimization are critical steps in building accurate and interpretable models. Interpretability is especially important in healthcare applications, as clinicians need to understand how the model arrived at its predictions in order to make informed decisions.

1.2.3 EHR

Electronic Health Record(EHR) data refers to the digital record of a patient's health information, including their medical history, diagnoses, medications, laboratory test results, radiology images, and other clinical data [3]. EHRs are created and maintained by healthcare providers, such as hospitals, clinics, and physician practices, to support the delivery of high-quality and coordinated patient care.

As shown in Figure 1.1, EHRs data can be collected from a variety of sources, including clinical documentation systems, medical devices, and patient-generated data. The data is typically stored in a structured format that can be easily accessed and analyzed by healthcare providers, researchers, and other stakeholders.

In the context of this research, EHR data is used to train and evaluate predictive models that can make predictions about patient outcomes based on multiple sources of clinical and non-clinical data. The multimodal nature of the data, which may include imaging data, genetic data, and other types of data in addition to clinical data, presents

unique challenges and opportunities for machine learning-based approaches.

1.3 Optimization Problem for Multimodal Learning

In deep learning, multi-modal learning refers to the process of training a model to learn from multiple types of data, such as images, text, and signals. In the medical domain, The data used for training and evaluating the model often comes from different sources and may have different characteristics, which can make the optimization problem more challenging. Additionally, medical data is often sensitive and private, which means that data privacy and security must be taken into consideration when developing and deploying deep-learning models for medical applications This is typically accomplished by using a deep neural network architecture that can process multiple modalities at the same time, such as a multi-modal transformer or a multi-modal neural network.

The loss function in multi-modal learning is typically a combination of multiple modality-specific loss functions. For example, in image-text retrieval, the loss function may include a cross-entropy loss for the image modality and a cross-entropy loss for the text modality. The loss function may also include additional terms such as a regularization term or a term that encourages alignment between the different modalities.

1.4 The Loss Function

The loss function is calculated by comparing the model's predictions with the true outputs for each modality. The model's parameters are then updated in order to minimize the total loss. The optimization problem is defined as finding the set of parameters that minimize the loss function.

As in chapters 3 and 4 of this dissertation, in a multimodal learning architecture, various loss functions can be used:

- **Joint loss function:** The joint loss function combines the loss from multiple modalities into a single scalar value. It can be defined as:

$$L(\theta) = L_1(\theta) + L_2(\theta) + ... + L_n(\theta) \tag{1.1}$$

  Where $L_1, L_2, ..., L_n$ are the loss functions for each modality, and θ are the model parameters.

- **Modality-specific loss function:** Each modality is trained separately and the modality-specific loss functions are combined to calculate the total loss. For example, for an image-text retrieval task, the modality-specific loss function for the image modality can be defined as a cross-entropy loss:

$$L_{img}(\theta) = -\sum_i [y_i log(p_i) + (1 - y_i) log(1 - p_i)] \tag{1.2}$$

  Where y_i is the true label and p_i is the predicted probability for the ith image. And for the text modality, it can be defined as:

$$L_{text}(\theta) = -\sum_{j}[y_j log(p_j) + (1 - y_j) log(1 - p_j)] \tag{1.3}$$

Where y_j is the true label and p_j is the predicted probability for the j_th text.

- **Alignment loss function:** The alignment loss function calculates the loss by comparing the different modalities' output and encouraging alignment between them. For example, in an image-text retrieval task, the alignment loss function can be defined as the cosine similarity between the image and text representations:

$$L_{align}(\theta) = 1 - \frac{v_{img} \cdot v_{text}}{||v_{img}|| \cdot ||v_{text}||} \tag{1.4}$$

Where v_img and v_text are the image and text representations, respectively, and $||v_img||$ and $||v_text||$ are the L2-norms of the representations.

It's important to note that the use of these loss functions will depend on the specific task, the data, and the model architecture. Additionally, in practice, it's common to use a combination of different loss functions to improve the performance of the model.

1.5 Scope and Motivation of the Study

There are many potential applications of deep learning in healthcare, including predicting patient outcomes, diagnosing diseases, and identifying risk factors for certain conditions. In particular, the use of deep learning models to predict patient outcomes based on EHR data has the potential to revolutionize healthcare by enabling healthcare providers to identify patients at risk of adverse outcomes and intervene early to prevent them.

However, there are several challenges associated with developing deep learning models for EHR data. First, EHR data is often incomplete and contains missing values, which can make it challenging to train models effectively. Second, EHR data is highly heterogeneous, meaning that different data sources may be recorded in different formats, making it challenging to integrate them into a single model. Finally, deep learning models are often considered "black boxes" because it can be challenging to interpret the results they produce.

Figure 1.2 illustrates the dissertation's overall view. One of the biggest challenges of machine learning in the medical area is the availability of data. The scope of this dissertation covers the seven green points as we couldn't proceed with the integration of the clinical images because the Medical Information Mart for Intensive Care (MIMIC) dataset that we were using didn't have associated images for patients. However, the new version of MIMIC 4 has comprehensive x-ray imaging that can be used to complete this research in the future.

With all that being said, our research was motivated by the following reasons, which will be reflected in this dissertation.

1. **Unstructured data such as medical narratives are paramount to understanding patients and disease trajectory. Therefore, it is important to analyze the role of Natural Language Processing in clinical data and assess how they can be improved to extract comprehensive features from EHRs data.**

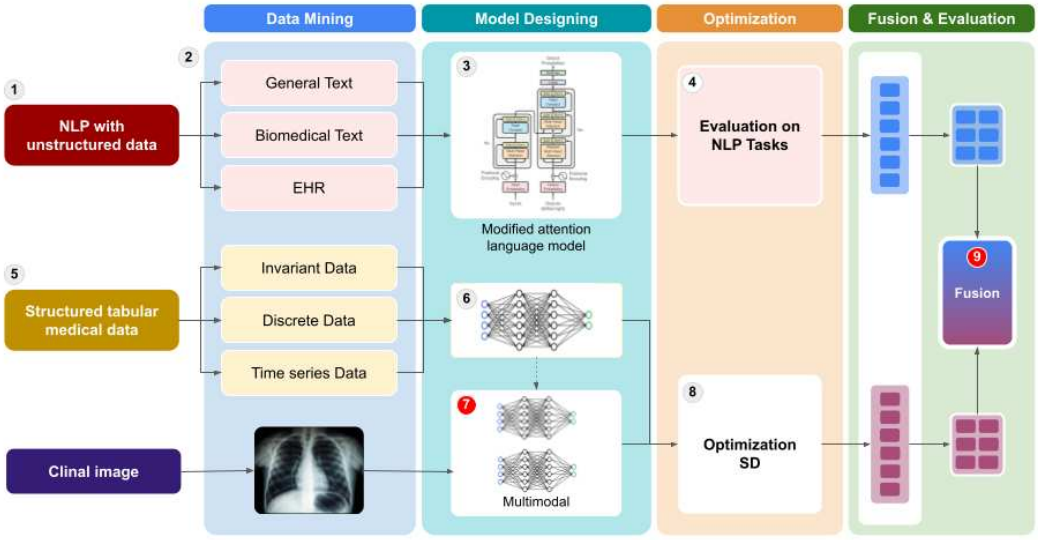


Figure 1.2: Illustration of the dissertation's overall view

2. **EHR data, like most real-world data, are characterized by in-domain challenges that need to be addressed by specific tools and methods. We analyzed and developed new language models to handle the in-domain challenges of EHR raw data.**

The ultimate goal of this dissertation is to propose optimization methods for predictive models based on data processing and transformation, customization of the existing models, and hyperparameter fine-tuning. To achieve this target, the following key components were studied and used in the proposed methods:

- **A data transformation based on fuzzy logic:** We are proposing a Fuzzy logic-based pipeline that generates medical narratives from structured EHR data and evaluates its performance in predicting patient outcomes. The pipeline includes a feature selection operation and a reasoning and inference function that generates medical narratives.
- **Biomedical Language Modeling**: Biomedical and clinical documents are characterized by various challenges that require in-domain models for effective biomedical text mining. Most of the existing biomedical LM rely on self-attention, which does not scale to clinical document length due to its space complexity and quadratic time. We introduce new biomedical language models and their tokenizers to mitigate with clinical and biomedical text data mining.

In conclusion, the motivation behind this research is to optimize the use of deep learning models in predicting patient outcomes based on EHR data. Despite the challenges associated with developing these models, there is significant potential for deep learning to revolutionize healthcare by enabling healthcare providers to identify patients at risk of adverse outcomes and intervene early to prevent them. By developing more sophisticated predictive models, we can unlock the full potential of EHR data and improve the quality of care provided to patients.

1.6 Dissertation Contributions

This dissertation presents a novel approach to analyzing Electronic Health Record (EHR) data by integrating multiple sources of data, including clinical, narratives, and potentially image data. Our work proposes using a comprehensive optimization framework for fine-tuning deep learning models toward a better analysis of multimodal data and making predictions about patient outcomes. This dissertation's contributions can be divided into three main areas:

1. **An extensive comparison of Natural Language Processing (NLP) pipelines for outcome prediction:** Our first contribution started with a comprehensive overview of the different NLP techniques and their effectiveness for outcome prediction, as well as identifying the most promising approaches for future research. This work evaluates several NLP techniques, including data processing, rule-based, and machine learning-based approaches, and compares their advantages and downsides in predicting patient outcomes from clinical notes.

2. **A novel EHR data transformation mechanism:** This research's second contribution proposes a mechanism to generate clean and comprehensive medical narratives to describe a patient through a textualization process of the medical tabular data. This textualization preserves the uncertainty and vagueness inherent in medical data while still allowing for the application of NLP methods and significantly improves the interpretability of predictions.

3. **Optimization of biomedical NLP models:**Our third contribution is a comprehensive optimization framework for fine-tuning biomedical NLP models that analyze EHR text data. Deep learning models have shown great promise in analyzing EHR data, but they require significant optimization to achieve optimal performance. Our study proposes a framework that includes techniques such as hyperparameter tuning, early stopping, and dedicated tokenizer which can significantly improve the performance of the models and the patient's representation. The optimization framework is designed to be flexible and can be applied to a wide range of deep-learning models and datasets.

Multimodal learning in the medical area has the potential to unlock new knowledge and provide more accurate and comprehensive insights into patient health and improve clinical decision-making. One of the paths we recommend for future work is to integrate more data such as images in a Contrastive Language-Image modeling where our contribution can be evaluated and enhanced with new pretraining model architectures.

1.7 Experimental Setup

This entire research was conducted locally using equipment provided by the laboratory. Most of the experiments were performed on a single computer equipped with a GPU accelerator. The hardware utilized in our experiments included a GPU Nvidia RTX3090 with 24GB of memory and an Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz processor. The software environment was based on an x86_64 Ubuntu 18.04 LTS operating system, utilizing the Anaconda environment. The CUDA version varied throughout the experiments from V10.1 to V12. CUDA was employed to leverage GPU acceleration for efficient computations. The experiments made use of the Huggingface

models, a widely recognized library in the field of NLP. The programming language employed throughout the study was Python, with the PyTorch framework serving as the primary DL framework. These details are provided to facilitate the replication and validation of the experimental results by offering comprehensive information about the hardware, software, and programming environment utilized in the research. We provide each chapter with more precise details about their related environment setups.

1.8 Dissertation Outline

The study outlined in this dissertation has several key components. Using the illustration in Figure 1.3, we describe the outline in 5 chapters.

- **Chapter 1** gives an overview of the research and introduces its background on optimization. It also defines key concepts from the dissertation's title as well as the scope of the research. We present the challenges and opportunities of multimodal approaches in the medical area and give our thought on future work.

- **Chapter 2** presents a preliminary study conducted on medical narratives in order to determine the best procedures for data processing and modeling. This work was necessary to understand from the existing solutions which provide the most accurate pipeline to handle raw medical text for a task such as an outcome prediction.

- **Chapter 3** introduces a novel approach to bridge the gap between medical tabular data and NLP by transforming tabular data into text using a fuzzy logic method. This research highlights the need for a new method to handle the downside of data regularization required by structured data. While our method was evaluated on an outcome prediction task, we demonstrated a significant improvement in the interpretability of the generated text using Shapley Values.

- **Chapter 4** develops two dedicated biomedical language models. One with a traditional architecture based on a full attention mechanism while another has sparse attention that allows the model to encode 8 times the length of regular transformer-based models. Our goal was to provide an optimized language model which can understand better raw medical notes. While most of the available biomedical language models use existing tokenizers, we trained and provided with our models a byte pair encoding-based tokenizer with the lowest fertility rate. Our models set new state-of-the-art in different biomedical NLP tasks including named entity recognition, question answering, sentence similarity, and relation extraction.

- **Chapter 5** concludes our dissertation by highlighting our contribution, and the limits of our methods with a high note on the potential future direction of the research.

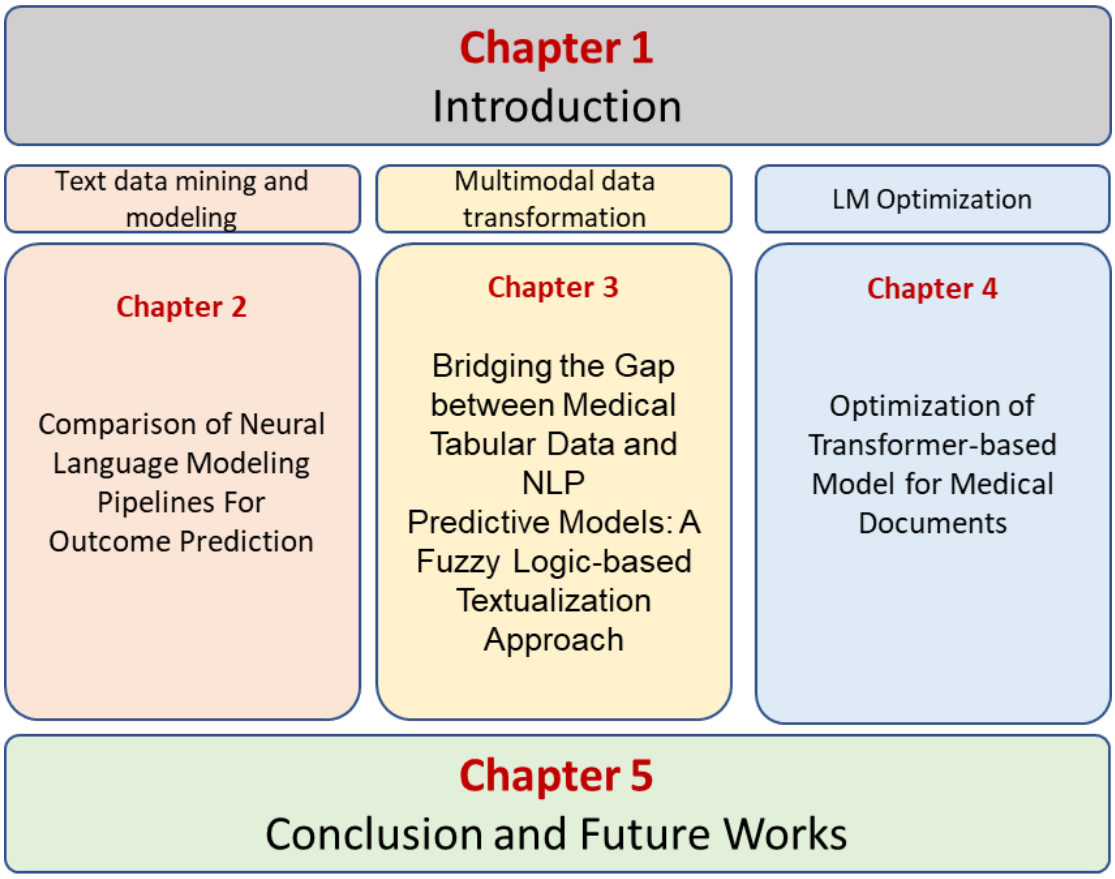


Figure 1.3: Illustration of the outline of the dissertation

1.9 Publications

The research and experimental outcomes of this dissertation have been presented or sent for consideration to several peer-reviewed journals and conferences. Chapters 2, 3, and 4 of the dissertation encompass the findings of several papers published as follows:

Major Journals

1. **Mugisha, Chérubin**, and Incheon Paik. "Comparison of Neural Language Modeling Pipelines for Outcome Prediction From Unstructured Medical Text Notes." *IEEE Access 10 (2022)*: 16489-16498.

2. **Mugisha, Chérubin**, and Incheon Paik. "Bridging the Gap between Medical Tabular Data and NLP Predictive Models: A Fuzzy-Logic-Based Textualization Approach." Electronics 12.8 (2023): 1848.

Major Conferences

1. **Mugisha, Chérubin**, and Incheon Paik. "Pneumonia outcome prediction using structured and unstructured data from EHR." In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 2640-2646*. IEEE, 2020.

2. **Mugisha, Chérubin**, and Incheon Paik. "Optimization of Biomedical Language Model with Optuna and a Sentencepiece Tokenization for NER." In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 3859-3861*. IEEE, 2022.

3. **Mugisha, Chérubin**, and Incheon Paik. "Medical Data Textualization using Fuzzy Logic for NLP Predictive Models". In 2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), IEEE, 2022.

Non-Major Conference

1. **Mugisha, Chérubin**, and Incheon Paik. "Comparison of Pre-trained Neural Language modeling pipelines for patient's outcome prediction from medical text notes." *IEICE Technical Report*; IEICE Tech. Rep. 121.51.

Chapter 2

Comparison of Neural Language Modeling Pipelines For Outcome Prediction

Motivation and Contribution

The significance of incorporating natural language processing and neural language modeling methods into clinical informatics research has been increasingly recognized in recent years and has led to transformative advances to support clinical decision-making. While statistics have shown that 80% of the information in EHRs is in a free text format [3]. Generally, clinical NLP systems are developed and evaluated on the basis of the word, sentence, or document-level annotations that model specific attributes and features, such as document content, document section types, named entities, concepts, or semantics. These methods are applied in information retrieval on both EHRs and electronic patient-authored text [18]. However, from a clinical perspective, research studies are typically modeled and evaluated at the patient or population level, such as predicting treatment response, patient monitoring, and discharge, or commonly for outcome prediction [19]. While medical notes can vehicle information between a multidisciplinary team, they can aid in patient profiling, augment hospital triage systems, and generate diagnostic models that detect early-stage chronic disease and ultimately suggest a prediction of patient outcomes [20]. The development of open-source NLP procedures, toolkits, and models has led to increased adaptability to clinical text to perform various tasks like outcome prediction.

The main question of this research is how the data preprocessing and modeling impact the predictive performance of NLP of clinical text documents for mortality prediction. Accurate prediction of future outcomes based on past data can help with decision-making and improve overall performance. However, the task of outcome prediction is complex and requires the ability to capture complex patterns in language data. Neural language modeling pipelines have become increasingly popular in recent years due to their ability to capture such patterns, but there are many different approaches to neural language modeling, and it is not clear which approach is best for outcome prediction. Therefore, there is a need to compare different neural language modeling pipelines for outcome prediction in order to determine which approach is most effective.

This chapter contributes to the field of natural language processing by comparing different neural language modeling pipelines for outcome prediction. We evaluate sev-

eral approaches, including recurrent neural networks (RNNs) and long short-term memory(LSTM), and transformers, and we use several evaluation metrics. Our experimental results show that mild processing and transformer-based modeling perform better than others for outcome prediction, especially when dealing with long-range dependencies between words. We also find that contextual embeddings, such as BERT perform better than traditional word embeddings, such as Word2Vec and GloVe, because they can capture context-specific information. The contribution of this research is significant because it provides guidance on the most effective approach to neural language modeling for outcome prediction tasks. This information can be used by practitioners in many fields, including healthcare, to improve decision-making and overall performance. Additionally, the evaluation metrics used in this research can be used as a benchmark for future research in this area. Overall, this research advances our understanding of neural language modeling pipelines for outcome prediction and provides valuable insights for practitioners and researchers alike.

2.1 Introduction

Pneumonia is an infectious disease of the lungs affecting alveoli and caused by bacteria, fungi, or viruses. Pneumonia can range in seriousness from mild to life-threatening. It remains the commonest infective reason for admission to intensive care as well as being the most common secondary infection acquired while in intensive care Unit (ICU) [21, 22].

Electronic Health Records (EHRs) are health-related information on an individual created in a health care organization. EHR systems contain structured data such as demographics, vital signs, laboratory test results, medications, and procedures. They also have unstructured medical or nonmedical data in a free format, such as imaging reports or care-provider notes [23]. In medical assessment, it is common and practical to use all types of data to understand the status of a patient or to predict his outcome. However, for caregivers, medical notes are of paramount importance.

Within a hospitalization or a clinical visit, a patient might have several note documents which can constitute a rich and long clinical history. Clinical notes provide a deep understanding of a patient's illness because they describe symptoms, clinical history, reasons for admission, and details of any intervention made by a multidisciplinary team [24]. With the medical texts representing 80% of the EHR data [3], admission notes constitute an extensive informative source used by doctors to draw a patient's profile within the first 24 hours of admission. It is then crucial to be able to use the patient's history and admission description to predict what is likely to happen during his stay.

In recent years, machine learning algorithms have been increasingly used to predict the outcome by using structured or unstructured medical data. However, using free-text notes to achieve such tasks may encounter a lot of challenges. Even though there are standards [25] when taking medical notes, most of the texts are biased by internal and conventional writing methods that make generalization harder for resulting models in a different environment. In addition, medical text notes can be too long to be handled by conventional natural language processing (NLP) models. Consequently, achieving good results requires a better algorithm that preprocesses the data and models the determinants of health condition for an overall understanding of a patient's status.

Although there are different studies on medical text classification, not many have demonstrated a clear statistical comparison of NLP pipelines to guide researchers in selecting methods to ensure the best results.

Our main insight was that admission narrative notes have the potential to predict outcomes only if, in the NLP pipeline, we can find the best combination of preprocessing methods, document representation, and learning models.

The question of this research is what combination of NLP models and preprocessing methods is appropriate to unlock the information from medical narratives. The present study aims to use medical pneumonia patient notes, written by a multidisciplinary team of care providers, to investigate among the dynamic word embeddings and static models to assess and compare their performance on the outcome prediction of an ICU hospitalization. We evaluate the resulting models using admission notes taken within 24 hours.

2.2 Outline

In section 2.1 we provide the background and motivation for the research, discussing the importance of outcome prediction and the use of neural language modeling pipelines. We also present the research question and the contribution of this research.

Section 2.3 is a literature review, which discusses previous research on neural language modeling for outcome prediction. We also provide an overview of the different approaches to neural language modeling, including RNNs, LSTMs, transformers, and contextual and non-contextual embeddings. This section also presents the evaluation metrics that we use in our experiments.

Section 2.4 describes the methodology of our research, including the dataset used, the preprocessing steps, and the experimental setup. We provide details on how we process and created different datasets, train and evaluate the different neural language modeling pipelines, and we also describe the different models that we use in our experiments.

In section 2.6, we present our experimental results. We compare the performance of the different neural language modeling pipelines using the evaluation metrics that we introduced in the literature review section. We also provide a detailed analysis of the results, discussing the strengths and weaknesses of each approach.

Section 2.7 discusses the implications of our results and provides recommendations for practitioners who want to use neural language modeling pipelines for outcome prediction. We also discuss the limitations of our research and potential directions for future research.

Finally, the chapter concludes with a summary of the research and its contributions. We highlight the main findings and discuss their implications for practitioners and researchers. We also discuss the strengths and limitations of our research and provide recommendations for future research in this area.

2.3 Background

2.3.1 Traditional Linear Models

Prediction of prognosis to inform decision-making in the ICU has a long history. Traditionally, statistical methods were widely used to evaluate the survival rate of a

patient using domain experts' features. Linear models such as Kaplan–Meier (KM) estimator were the most popular combination with Cox proportional hazards regression to handle regression problems [26]. Using advanced methods based on logistic regression, Simplified Acute Physiology Score (SAPS) and the Acute Physiologic Assessment and Chronic Health Evaluation (APACHE) demonstrated a clear improvement in assessing the disease severity of a hospitalized patient. However, those models use predetermined features, which greatly limits their applicability in a real situation.

In recent years, those traditional methods have been surpassed by more modern and accurate algorithms mainly using data-driven models based on machine learning architectures. Since the linguistic string project [27] in analyzing clinical documents, most of the work has been conducted around general medical management, treatment, test and results, patient state, and patient behavior using medical text.

However, it has been more challenging to demonstrate the real usability of nonmedical data. The nature of medical text data requires a combination of steps to unlock the information embedded in a clinical text (e.g., disease, treatment, patient status) by transforming the text into structured medical data. The automation of this process and the efficiency of models to perform tasks such as clinical text classification has been investigated [24].

2.3.2 ML Models and Documents Preprocessing

Currently, ML techniques have shown a major improvement for prediction in the general domain and particularly in the medical domain. They can perform better using either structured and unstructured data or even both through an ensemble of machine learning processes [23, 28]. An NLP task starts with a preprocessing stage, to extract useful information and structure the raw text into a format that can make use of automated computing power.

A review [29] conducted on 67 publications from 2000 to 2015 has shown that extracting information from the EHR narratives can improve case detection for classification tasks. While this process can use different techniques of information extraction, 67% of the studies incorporated rule-based, 24% used keywords, and only 9%, include machine learning in their approach. However, this trend has changed and a recent review [30] shows that machine learning-based methods are more used. The medical data transformation stage is challenging for messy medical notes because preprocessing can clean out significant information that is clinically important to predict the outcome accurately [31, 32].

To process medical text, Authors have been using the Unified Medical Language System (UMLS) in order to reduce ambiguity from abbreviations and conventional annotations. However, according to Liu et al. [33], 31% of UMLS abbreviations have multiple meanings. This can be resolved by computing the proximity of the abbreviation to its expanded form and replacing it with the most suitable term. This abbreviation disambiguation pipeline has brought a lot of controversy among the community and led to the creation of several resources for different data.

In NLP, deep learning methods such as Long Short Term Memory (LSTM) and its variants use a preprocessing pipeline with a filtering process based on predefined controlled vocabulary terms before transforming data into training vectors [34, 35].In this recent study [36], the authors propose an online medical pre-diagnosis support in which semantic and sequential features are extracted from a patient's inputs using a

CNN-RNN-based architecture model to predict a diagnosis.

Geraci et al. proposed a neural network to extract phenotype information from electronic medical record(EMR) text notes [37]. Their goal was to identify suitable candidates for medical research using doctors' narratives within a supervised learning process. They extract useful information through a Document Term Matrix (DTM) using the TF-IDF algorithm. Their results show that NLP can help to identify criteria for models to perform better for a task such as classification. Wang et al. [38] illustrated a paradigm of clinical text classification using deep representation and weak supervision. Their work demonstrated that it could be possible to use a deep neural network like CNN and outperform traditional NLP rule-based algorithms. Their approach also compared the importance of using word embeddings over count vector algorithms such as TF-IDF. However, their method has limitations because the model is trained from scratch, it requires a lot of training data. The input size is also dictated by the word embedding methods, which are usually unsuitable for long text like medical narratives.

Authors have tried to tackle this problem in recent literature by using more sophisticated NLP models. For example, models like Bidirectional Encoder Representations from Transformers (BERT) [39]have shown impressive results from an architecture of multilayer encoder models, to learn words and document representation more deeply. In the medical area, researchers have been trying to leverage the knowledge from general documents to pre-train specific-domain models for higher performance for medical-related tasks [40, 41].

Despite that evolution in NLP, we still have many publications that use either advanced machine learning models or archaic models. Although most of these studies claim to have achieved the best scores in various tasks by using very different approaches, we can only wonder if there is no room for improvement. To our knowledge, no other research trying to elucidate a fair comparison of those methods has been published.

2.3.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of artificial neural network that is specifically designed for processing sequential data. The key feature of RNNs is their ability to maintain a hidden state that can be updated and passed on to the next time step, allowing them to capture the temporal dependencies in the data. This makes RNNs particularly useful for tasks such as speech recognition, language translation, and time series prediction.

The core mathematical formula for an RNN can be expressed as:

$$h_t = f\left(W_{hh}h_{t-1} + W_{xh}x_t + b_h\right) \tag{2.1}$$

Where h_t represents the hidden state at time step t, x_t represents the input at time step t, W_{hh}, W_{xh}, and b_h represent the weight matrices and bias vector for the hidden layer, and f is the activation function. The hidden state at each time step is updated based on the previous hidden state, the current input, and the learned weights.

RNNs have been widely used in biomedical applications, including medical image analysis, electronic health record analysis, and disease diagnosis. For example, RNNs have been used to classify electroencephalography (EEG) signals for detecting epileptic seizures [42], predicting the progression of Alzheimer's disease using MRI scans [43], and predicting hospital readmissions based on patient data from electronic

health records [44]. RNNs have also been used for drug discovery and development, such as predicting drug efficacy and toxicity based on molecular structure and identifying potential drug candidates from large databases [45].

Despite their usefulness, RNNs have limitations, including the issue of vanishing gradients and the lack of memory capacity, as discussed earlier. However, recent advancements such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) have been developed to address these issues and improve the performance of RNNs in sequence modeling tasks.

2.3.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that was designed to address the vanishing gradients problem and the lack of memory capacity in traditional RNNs [46]. LSTMs are particularly effective for processing sequential data with long-term dependencies, making them useful for natural language processing, speech recognition, and time series prediction.

The key feature of LSTMs is the use of memory cells that can store information over long periods of time. The memory cells are controlled by gates that regulate the flow of information in and out of the cells. The gates are composed of the sigmoid and element-wise multiplication operations, allowing the LSTM to forget or remember information from previous time steps selectively.

One way to represent the central mathematical formula of an LSTM cell is by:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right)\ i_t \qquad = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right) = \tag{2.2}$$

$$\tilde{C}t = \tanh\left(W_C[ht-1, x_t] + b_C\right)\ C_t \qquad = f_t * C_{t-1} + i_t * \tilde{C}t = \tag{2.3}$$

$$o_t = \sigma\left(W_o[ht-1, x_t] + b_o\right)\ h_t \qquad = o_t * \tanh(C_t) \tag{2.4}$$

where h_t is the output at time step t, x_t is the input at time step t, C_t represents the memory cell at time step t, and f_t, i_t, and o_t represent the forget, input, and output gates, respectively. The weight matrices and bias vectors are denoted by W_f, W_i, W_C, W_o, b_f, b_i, b_C, and b_o. The sigmoid function σ and the hyperbolic tangent function $\tanh$ are the activation functions used in the LSTM.

LSTMs have been used in various biomedical applications, such as electrocardiogram (ECG) signal analysis, predicting patient outcomes in intensive care units, and predicting drug-drug interactions. For example, LSTMs have been used to predict the onset of atrial fibrillation using ECG signals [47], predict patient mortality and length of stay in intensive care units based on electronic health record data, and predict drug-drug interactions based on molecular structures.

Compared to traditional RNNs, LSTMs have demonstrated superior performance in modeling long-term dependencies and handling vanishing gradients, making them more effective for text modeling tasks. Additionally, LSTMs are more interpretable, allowing researchers to better understand the reasoning behind the model's predictions.

2.3.5 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of neural network that have been widely used in image and video processing tasks. However, they have also shown great potential in Natural Language Processing (NLP), including text classification and sentiment analysis tasks.

The key idea behind CNNs is to use filters (also called kernels) to convolve over a fixed-length window of input data, producing a feature map that captures local patterns in the data. These feature maps are then pooled to reduce the dimensionality and extract the most relevant features for downstream tasks.

Mathematically, a 1D CNN can be expressed as follows:

$$h_i = f\left(W x_{i:i+k-1} + b\right) \tag{2.5}$$

Here, $x_{i:i+k-1}$ represents the input data of length k starting from the i-th position, W is the weight matrix, b is the bias term, and f is the activation function. The resulting output h_i is the feature map obtained by convolving the filter over the input window.

In the context of biomedical text modeling, CNNs have been applied to various tasks such as disease classification, drug identification, and medical image captioning. For instance, in the task of identifying diseases from clinical text, CNNs have been shown to outperform traditional machine learning methods, achieving state-of-the-art performance [48].

Overall, CNNs have demonstrated their effectiveness in NLP tasks by capturing local patterns in the data and extracting relevant features, making them a powerful tool for biomedical text modeling.

2.3.6 Embeddings

In NLP, word embeddings are a way to represent words as vectors in a high-dimensional space, where each dimension represents a semantic feature of the word. Word embeddings are used to capture the semantic and syntactic relationships between words, and are commonly used in NLP tasks such as text classification, sentiment analysis, and language translation [49].

Traditional embeddings for language modeling are a type of representation of words in a low-dimensional space that aim to capture the semantic and syntactic similarities between them. There are several types of traditional embeddings, such as one-hot encoding, count-based methods (e.g., Term Frequency-inverse Document Frequency (TF-IDF), and neural network-based embeddings (e.g., Word2Vec, GloVe).

One-hot encoding represents each word as a sparse vector with a single ”1” in the position corresponding to its index in a vocabulary and ”0”s elsewhere. Count-based methods represent each word by its frequency of occurrence in a corpus, taking into account the frequency of other words in the same document or corpus. Neural network-based embeddings use a neural network to learn a dense representation of words in a low-dimensional space, where semantically similar words are mapped to nearby points. Word2Vec [49] and GloVe [50] are two popular neural network-based embedding methods that use different techniques for training the embeddings.

The most commonly used approach for generating word embeddings is the Word2Vec algorithm, which uses a neural network to learn the embedding vectors from a large corpus of text. The neural network is trained to predict the context words surrounding

a target word, or to predict the target word from its context words, and the resulting weight matrix of the neural network represents the word embeddings.

Mathematically, given a vocabulary of N words, each word is represented by a d-dimensional vector, where d is the dimensionality of the embedding space. Let w_i denote the ith word in the vocabulary, and let v_i be its corresponding d-dimensional embedding vector. The goal of the Word2Vec algorithm is to learn the embedding vectors such that the dot product of two vectors captures the similarity between the corresponding words. This is achieved by minimizing the negative log-likelihood of observing the context words given the target word, or vice versa:

The Skip-gram model of word2vec can be written mathematically as:

$$\sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j} \mid w_t) \tag{2.6}$$

where T is the total number of words in the corpus, m is the window size (i.e., the maximum distance between the predicted word and the context words), w_t is the target word, and w_{t+j} is the context word. $P(w_{t+j} \mid w_t)$ is the conditional probability of observing the context word w_{t+j} given the target word w_t, which is modeled using the softmax function and the dot product of the word vectors:

$$P(w_{t+j} \mid w_t) = \frac{\exp(v_{w_{t+j}}^T u_{w_t})}{\sum_{w=1}^{W} \exp(v_w^T u_{w_t})} \tag{2.7}$$

where $\boldsymbol{v}_w$ is the vector representation (i.e., embedding) of word w, and V is the vocabulary of the corpus. The objective of the Skip-gram model is to maximize the average log probability of predicting the context words given the target words:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j} \mid w_t) \tag{2.8}$$

This objective can be optimized using stochastic gradient descent (SGD) and back-propagation through time (BPTT) to update the word vectors during training.

2.4 Methods and Materials

The task of prediction has been extensively researched in academia. Establishing a relationship between the copious amounts of clinical data and the outcome has the potential to enhance our understanding of life and the factors involved in its end. In the ICU, time could be a crucial factor, and medical notes offer a viable option to identify critical patient issues when other sources are unavailable, providing valuable and easily interpretable information. The concise descriptions of patients contained in text narratives can inform caregivers about their status upon admission to the ICU. However, these narratives may include unnecessary information, such as duplicates of structured data or repetition from different contributors, making it difficult to model them for prognosis or prediction. Machine learning and NLP have proven to be incredibly adept at learning from even the messiest of data, with the potential to produce model outputs. Nonetheless, constructing a consistent and valuable model requires a profound understanding of the data, including what preprocessing steps are necessary and what model

architecture is best suited for the data.

2.4.1 Study Workflow Diagram

This research was conducted in several steps consisting of three main processes illustrated in Fig. 2.1. A summarized workflow for the study is as follows:

1. **Step1**: Data sampling and selection from a large dataset of MIMIC notes.
2. **Step2**: Data preprocessing through a light(A), thorough(B), and extreme(C) cleaning performed by extracting medical entities, producing three separate datasets.
3. **Step3**: Use of various embeddings to train different NLP classifier models. α, β, and γ illustrate different embeddings for each model. α used Global Vectors for the three datasets, β used CountVector and TF-IDF from the NER dataset, while γ used BERT embeddings for all three datasets

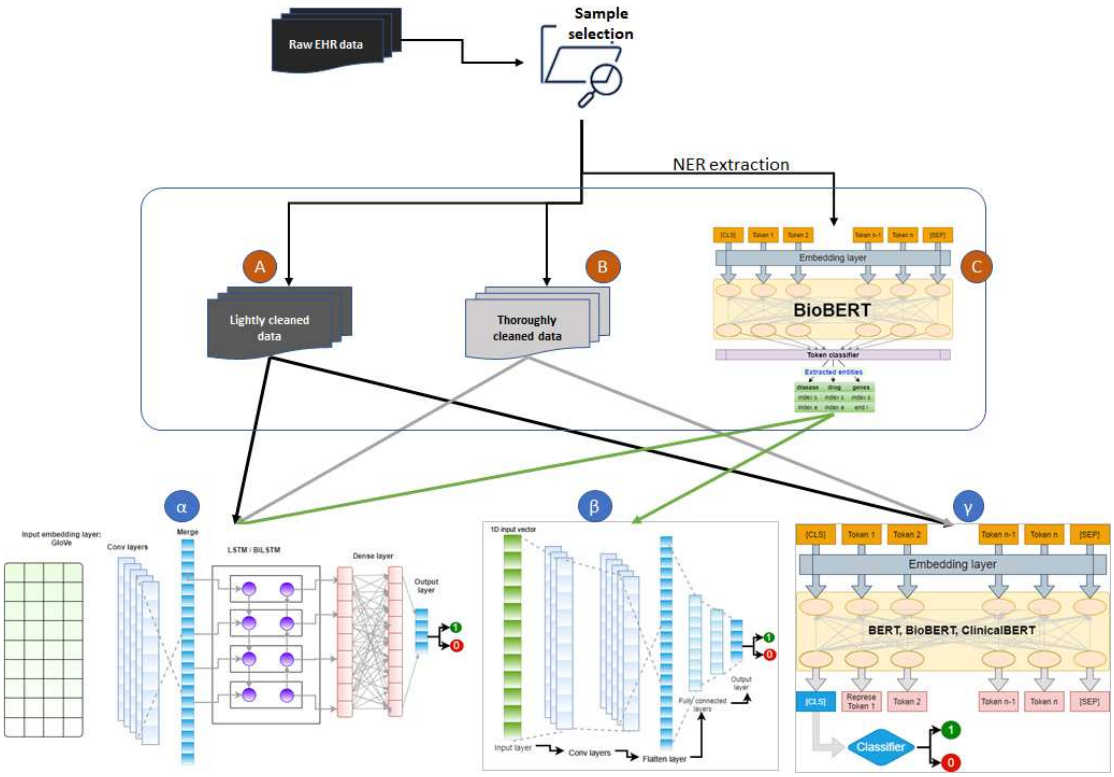


Figure 2.1: Study Framework.

Detailed descriptions of each of these steps are provided in the following sections.

2.4.2 Data Cleaning

Data cleaning refers to steps that we took to standardize our data and to remove text and characters that are not relevant to be left with a clean text dataset that is ready to be analyzed.

Authors have suggested many methods of text data cleaning. Most of the NLP cleaning tasks are based on basic rules such as converting text to lowercase, regular expression and word replacement, punctuation, and nonalphanumeric character removal.

Advanced preprocessing will include more tasks such as stop-words and tokenization, stemming and lemmatization, word tagging, or Named Entity Recognition (NER). All of these will depend on the data, whether it has a dictionary or not, or simply the NLP task we want to perform. For data privacy, an illustration sample of the cleaning results is provided as a reference in Fig. 2.2. This table shows an example of the transformation of the raw data (first column), through a cleaning process, producing three different datasets A,B, and C. To create our three datasets and analyze in detail the impact of each cleaning approach, especially for medical notes, we proceed as follows:

Minor Cleaning

This cleaning task follows the basic rules of the NLP cleaning task utilizing the natural language toolkit (NLTK). For case sensitivity, we converted all text into lowercase and used regular expressions to remove punctuation extra white spaces, line breaks, and nonregular expressions. In addition, We utilized the Stop-words dictionary to filter out irrelevant entities.

Thorough Cleaning

To take our cleaning process even further, and harmonize clinical abbreviations and acronyms, we manually built up a matching dictionary of 80 terms. Using the UMLS [51, 52] with its metathesaurus inventory, we selected the top used acronyms in medical notes presented in these studies [53, 54] and added predominant risk factors for pneumonia such as acute respiratory distress syndrome (ARDS) and acute respiratory failure (ARF) [55, 56]. We also removed de-identification characters and harmonized typos and conventional spellings (e.g., pt, dr, W/O). These two cleansing processes are suitable for the emergent bidirectional models because their tokenization uses a word-piece technique and does not need a deep cleaning.

Named Entity Recognition

Narratives can be very long and full of information for which it could be necessary to filter out domain-related data. NER has shown the ability to process data semantically by identifying and categorizing key information (entities) in text [57–59]. Traditionally, dictionary NER-based models have been used for text data mining, and recently, deep learning-based models have shown outstanding progress leveraging pretrained language models. For our case, we addressed this step as sentence-level biomedical information extraction tasks. The biomedical language representation model for biomedical text mining (BioBERT) [40] is a domain-specific language model that has been trained on medical text data. BioBERT NER (BERN) [60] is one of its modules for recognizing biomedical entities and discovering new entities. We used BERN to extract entities related to disease, drugs/chemicals, genes/proteins, and species. However, the resulting entities are independent of each other, so they can only be used by nonsequential models and their association would be considered as correlated features.

Raw Data sample	Minor cleaning (Dataset A)	Thorough Cleaning (Dataset B)	NER (Dataset C)
85yo woman w recent dx PE, refractory HTN, DM, CKD (bl Cr1.4-1.6), anemia who presents from NH with increasing SOB. PerNH records, she became febrile to 103 w HR 79, BP 180, O2 sat89% RA. Per patient, she noted increasing DOE over the last 24h.She has had increasing fatigue and has been less mobile for [**12-13**]	85yo woman recent dx pe refractory htn dm ckd bl cr141 anemia who presents from nh with increasing sob pernh records she became febrile to 103 hr 79 bp 180 o2 sat89 ra per patient she noted increasing doe over the last 24h she has had increasing fatigue and has been less mobile for 1213	85yo woman with recent diagnosis pulmonary embolism refractory htn dm ckd bl cr141 anemia who presents from nh with increasing shortness of breath pernh records she became febrile to 103 hr 79 bp 180 o2 sat89 ra per patient she noted increasing dyspnea on exertion over the last 24h she has had increasing fatigue and has been less mobile for 1213	Pulmonary Embolism, , HTN, DM, CKD,anemia, O2 ,Dyspnea, fatigue

Figure 2.2: Illustration of the three types of data cleaning.

2.5 NLP Models Used in the Study

In this study, we utilized various NLP models, ranging from traditional to modern ones, to ensure a comprehensive comparison. In the medical field, count-vector-based models, word-embedding-based models, and transformer-based models have been commonly used in NLP tasks. To conduct a fair comparison, we employed all of these models and prepared the data accordingly.

2.5.1 Text Representation Techniques

To transform the raw text into a machine-understandable format, we used different text encoding techniques, including:

Bag of Words (BOW)

BOW is a statistical method that represents text by the frequency of occurrence of each word or sentence in a document [61]. BOW has been widely used in text classification [62,63] due to its effectiveness in keyword-based classification. However, it only represents words in a one-dimensional vector and lacks semantic and syntactic information. We hypothesized that discriminative terms could be identified through term frequency counting. The CountVectorizer, a low-level one-hot encoder, converts text into a vector based on the frequency of each term in the entire document. This approach can be effective if each class has distinctive identifying terms. Another approach, the Term Frequency-Inverse Document Frequency (TF-IDF) measures the relevance of a given term by multiplying its frequency by the logarithm of the inverse document frequency of that term across the entire corpus.

$$x_i j = TF_i j * log(|J|/TF_i j) \tag{2.9}$$

where TF_ij is the term frequency and J is the number of documents in the corpus. TF-IDF is a more efficient technique compared to other methods since it scales up the importance of rare terms and reduces the significance of frequently occurring words like "patient" in our context. For our dataset, which involved ignoring or breaking the order of words (such as NER), we utilized both Count Vectorizer and TFIDF Encoder. Furthermore, we experimented with varying the vocabulary size between 1000 and 5000 words.

Global Vectors for Word Representation

To fully capture the various dimensions of text data, a model that can vectorize text based on precise syntactic and semantic word relationships must be employed. Global Vectors for Word Representation (GloVe) [50] represents words as real-valued vectors in a vector space with relatively low dimensions compared to their vocabulary size. As a result, words that are semantically and syntactically similar will be closer in the vector space. This approach differs from previous embeddings like Word2Vec [64], where the frequency of co-occurrences within context windows is crucial for semantic information to be retained. For our study, we assumed that frequent co-occurring words are critical in determining the outcome, and the global corpus statistics are already embedded in the GloVe vectors. We obtained pre-trained word vectors and used them as our word embeddings for training a sequence model.

BERT Embeddings

BERT is a word vector representation that is contextualized, meaning it creates different vectors for a word depending on the context it is used in. Its encoding is done by the transformer encoder, which captures the relationships between distant words more efficiently than traditional bidirectional encoders, thus resulting in higher-dimensional space. It can encode any words or subwords, using its position in the input sequence, and utilizes a vocabulary size of more than 30000 tokens. In addition, BERT provides a [CLS] token to each sequence, which can be used for a classification task. In the biomedical text, subword utilization has an advantage as it enables the encoding of uncommon medical terminologies more accurately.

2.5.2 Learning Models

Recurrent Networks

In the realm of deep learning, recurrent neural networks (RNNs) have traditionally been applied to time series data, including sound and monitoring data in medical contexts [65]. However, standard RNNs are often slow and can suffer from the vanishing or exploding gradient problem when dealing with long sequences [66]. To address these issues, researchers have developed variants of RNNs, such as LSTMs, that can handle longer sequential inputs and bidirectional dependencies through attention mechanisms like BiLSTMs [67]. These models employ an encoder component for classification tasks, where multiple recurrent cells process each input vector and propagate it forward.

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t) \tag{2.10}$$

Hidden states h_t are computed by applying some weights $w^{(hh)}$ on the previous input vector x_i where i is the order of the input words. For the output, a decoder part will calculate a vector where each value will represent a probability score for each class.

$$y_t = SoftMax(W^s h_t) \tag{2.11}$$

Here, h_t represents the output of the encoder for an input i, and W^s is the respective weight applied by the decoder before feeding to a SoftMax function [68].

Transformer

RNNs and LSTMs suffer from slowness because they require sequential processing of data. However, transformer-based models can take advantage of the progress in computing technology and perform parallel processing, allowing them to learn more quickly. In the field of medical text analysis, researchers have proposed several language models based on the knowledge gained from models like BERT. For our study, we specifically utilized ClinicalBERT and BioBERT, which are domain-specific models for biomedical text. Despite their domain specificity, these models were pre-trained on different data. BioBERT was initialized using BERT weights and trained on PubMed abstracts and Central full-text articles, while ClinicalBERT utilized BioBERT weights and was pre-trained on MIMIC medical notes. To use these models for our classification task, we fine-tuned their weights and tried different approaches suggested by the authors of [39] for classification using the [CLS] token. For our study, we obtained a sentence embedding vector by vector-wise summation of the last four layers and used it as an input to train a logistic regression classifier, which computed a binary probability.

$$P(outcome = 1|h_n) = ArgMax(Wh_n), \tag{2.12}$$

where h_n is the averaged output of n hidden layers and W is the parameter matrix of the classifier.

2.5.3 Experiment Design

Our experimental design involved using free-text narratives from the MIMIC-III database to develop a model for predicting a binary outcome through the NLP process, starting from the cleaning stage and proceeding to prediction. As illustrated in Fig. 2.1, our experiment was comprised of three main stages.

The first stage involved data gathering and selection, whereby only patients with pneumonia as the primary disease were chosen for our experiment.

The second stage was related to data preprocessing, which aimed to enhance data quality and prepare it for models. To this end, we proposed three cleaning methods that resulted in three distinct datasets, which we refer to as set A, B, and C. These sets were generated using minor cleaning, thorough cleaning, and medical entity extraction (NER).

The third stage was focused on optimizing the NLP machine learning models. In this study, we evaluated the performance of different machine learning algorithms using both static and contextualized word embeddings.

Static Word Embeddings

The first method we used in this study is static word embeddings, which assign a single vector to each word. These vectors are dense and have lower dimensionality compared to the vocabulary size. We utilized two vectorization techniques, namely count vectorization and TF-IDF, to generate simple document vectors. However, these models do not consider the context and meaning of a word in a document. Therefore, they are suitable for bag-of-words representations, such as those obtained from NER methods, which do not have any sequential relationship. To improve the discrimination of our classification, we used n-gram (n = 1) vectors for medical terms with low frequency. We passed the resulting vectors to a logistic regression model and set the maximum number of iterations to 5000, the solver to liblinear, and other parameters to their default values.

For a more advanced static word embedding method, we utilized Global vectors for word representation (GloVe) [50]. These pretrained word embedding vectors include the sequential dimension and were trained on a Wikipedia dataset. Each of the approximately 6000 words is represented by a vector of size 300. These embeddings are suitable for sequential models such as LSTM and BiLSTM.

Dynamic Word Embeddings

To analyze the importance of contextualized word embeddings for medical free text, we relied on 12 layers of language representation models (BERT, BioBERT, and ClinicalBERT). To handle long narratives, we shrunk them into small sentences of 380 words, to leave some room for the tokenization process that will output 512 tokens.

2.5.4 Experimental Setup

Our hypothesis was to test the effectiveness of using text notes for predicting outcomes by making predictions as soon as a patient is admitted to the ICU. To test this, we created a test set using only admission notes. However, we found that the database did not have any tags to identify admission narratives, so we utilized SQL queries to filter admission notes based on certain criteria, such as uniqueness per admission ID, and taken by a nurse within the first 24 hours. We sampled this set as the test set and split the rest into a 90% training set and a 10% validation set to have a separate validation set. The training and validation sets consisted of progress, nursing, and procedure notes.

One limitation of contextualized encoding is that it has a restriction on sentence length. Medical narratives are typically lengthy and do not indicate which part contains the most relevant information. Thus, we had to truncate each long note to a maximum size of 380 words to use in all embeddings. As a result, our dataset changed from 85,085 long notes to 1,101,524 notes, with each note producing nearly 12 small consecutive notes that we labeled as their original narratives. Although we trained these notes separately, we averaged the predicted classes to calculate the loss. We will describe the distribution of our datasets in the results section (Table 2.2).

2.6 Results

For a fair comparison, given the low mortality rate in the data (29%), we calculated the accuracy in terms of sensitivity and specificity, respectively by recall and precision metrics. As an overall evaluation metric, we reported the F1-scores and balanced accuracy

$$(\frac{1}{2} * \frac{TP}{P} * \frac{TN}{N}) \tag{2.13}$$

We reported results with Matthews correlation coefficient (MCC) metrics. MCC takes into account true positives, true negatives, false positives, and false negatives, providing a more balanced measure of classification performance than metrics such as accuracy, precision, and recall, especially when the classes are imbalanced like in our case.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.14}$$

In contrast, we evaluated each model with different datasets resulting from our cleaning methods. This evaluation was conducted on the aforementioned test set made from admission notes. The scores demonstrate how well each model with a particular preprocessing can predict the outcome using the information described by admission notes.

Table 2.1: Evaluation and comparison of BOW-based models. A logistic regression classifier used a unigram representation from count-vectorizer and TF-IDF of 1000 and 5000 vocabulary size, respectively.

Vectorization	Accuracy	Precision	Recall	Specificity	F1-Score
B dataset					
CountVec-1000	0.51	0.743	0.532	0.831	0.621
CountVec-5000	0.59	**0.798**	0.63	**0.867**	0.71
TF-IDF-1000	0.657	0.761	0.804	0.743	0.789
TF-IDF-5000	**0.778**	0.784	**0.928**	0.725	**0.852**
C dataset					
CountVec-1000	0.521	0.777	0.567	0.863	0.656
CountVec-5000	0.62	**0.837**	0.656	**0.911**	0.736
TF-IDF-1000	0.697	0.802	0.827	0.792	0.815
TF-IDF-5000	**0.801**	0.805	**0.993**	0.728	**0.889**

From the very basic BOW, Table 2.1 shows that vectorization from NER leads to better results than a thorough cleaning process by all metrics. Between Count-vectorizer and TF-IDF, the latter performs better with accuracy, recall, and F1-score of 0.801, 0.993, and 0.889, respectively.

Table 2.2 shows a deep comparison of the static and contextualized embeddings as well as the accuracy of models to predict the outcome. For LSTM and BiLSTM, we performed a k = 10-fold cross-validation, using the training set and we trained both models for 10 epochs. Contextualized embeddings demonstrated a higher ability to understand the medical narratives than the static embeddings with a difference of 6%

between their respective best scores. For static embedding, BiLSTM shows a better F1-score of 92.01% using the B dataset, however, using independent entities from the C dataset, BOW outperforms LSTM with an F1-score of 88.9% from TF-IDF against 54.6% from LSTM.

2.6.1 Performance on the best-performing models

Fig. 2.3 reports values obtained from the training of contextualized word representation using different word-piece tokenizations and embeddings. Each model was trained for 50 epochs using a validation set of 20% to control the over-fitting, this shows a comparison of the training, validation, and accuracy of BERT, BioBERT, and ClinicalBERT Extensive training of 50 epochs on the B dataset shows that BioBERT and ClinicalBERT have a more stable logarithmic training loss curve while BERT needs more training epochs. This is also illustrated by the validation (Fig. 2.3b) and the test (Fig. 2.3c), where BioBERT and ClinicalBERT performed similarly but BERT needed more than 20 epochs to gain stability.

However, despite BERT's pre-training on a general corpus, all models exhibited notable discriminatory capability, as illustrated in Table 3.2 where F1-scores of 98.2%, 97.4%, and 98.2% were achieved.

Although these scores appear to be similar, the precision scores of 98.1%, 96.7%, and 97.4% clearly indicate that BioBERT has a comparatively limited ability to handle imbalanced data. Despite the lower performance of BioBERT on Dataset A, it is noteworthy that it exhibits the best specificity score among the models. This suggests that BioBERT has the lowest false positive ratio and is reliable in correctly identifying true negatives. Notably, the most effective model relied on the B dataset, highlighting the significance of a thorough data-cleansing process prior to training a contextualized model. Substantial improvements of 9.79%, 7.7%, and 4.3% in MCC scores were observed for BERT, BioBERT, and ClinicalBERT, respectively.

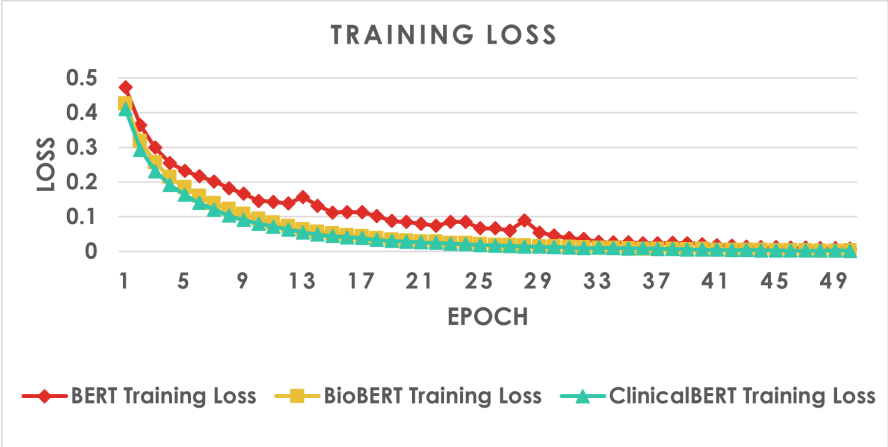
2.6.2 Additional analysis

Word and document embeddings serve as foundational elements for representing the underlying knowledge within the input text. In order to gain insights into how various models and embeddings interpret medical notes differently, we sought to elucidate these distinctions through vector similarity analysis. Once our narratives were vectorized, we employed statistical similarity methods for further exploration. However, it is important to note that multidimensional vectorization techniques like BERT cannot be easily reshaped into two dimensions without sacrificing crucial information. Conversely, BOW-based models lack semantic and contextual information, which poses a limitation in initiating any clustering behavior right from the outset of the NLP pipeline.

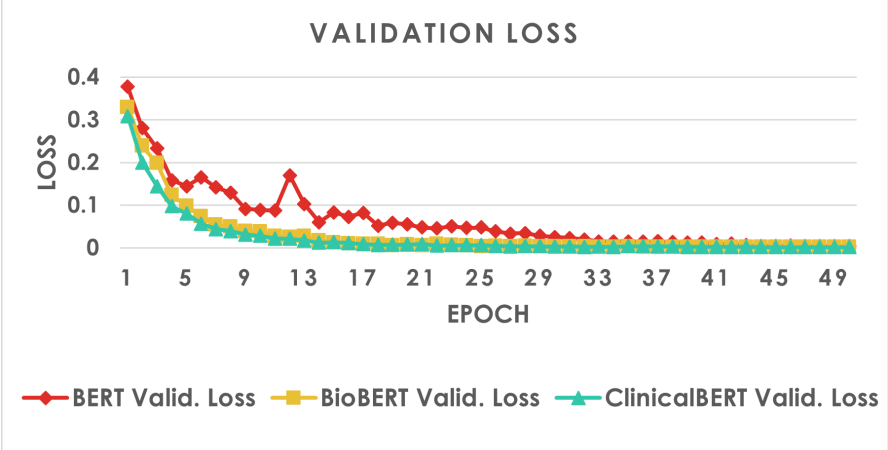
As depicted in Figure 2.4, the application of cosine similarity distance analysis revealed no discernible evidence of clustering between the feature representations of the two classes. This analysis was conducted on a sample of 12 inputs, consisting of 6 notes from discharged patients and 6 notes from deceased patients. By randomly selecting samples and organizing them into six positive and six negative notes, we examined the potential of cosine similarity to identify any similarities between the two classes. However, none of the noncontextualized models exhibited such discriminatory capability.

Table 2.2: Results from Global vectors and BERT-based vectorization

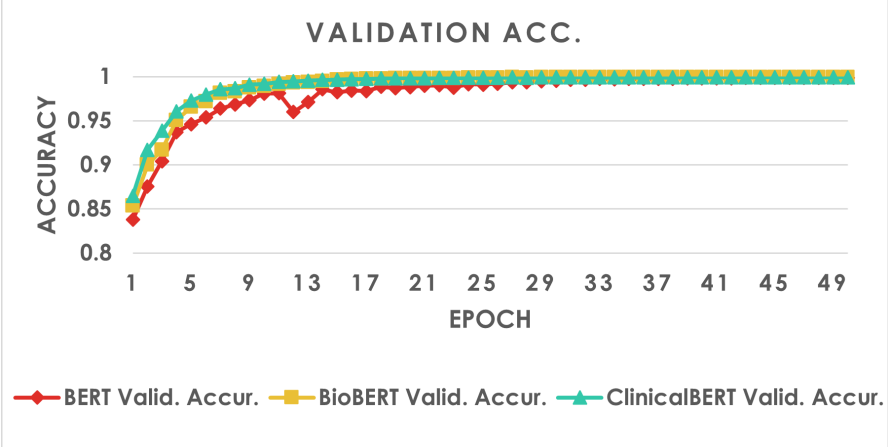
Models	Data	Training size	Test size	Tokenization	Embeddings	MCC	Acc.	Precision	Recall	Spec.	F1
LSTM	A	1101524	2239	GloVe 300	GloVe 300	-	89.04	80.7	58.2	85.86	70.1
	B	1101524	2239	GloVe 300	GloVe 300	-	92.5	89.3	67.7	94.26	77.0
	C	1101524	2239	GloVe 300	GloVe 300	-	84.8	52.7	40.5	55.50	54.6
BiLSTM	A	1101524	2239	GloVe 300	GloVe 300	-	93.2	81.4	80.6	81.58	81.3
	B	1101524	2239	GloVe 300	GloVe 300	-	97.05	93.14	90.91	93.65	92.01
BERT	A	1101524	2239	BERT Wordpiece	BERT	88.01	96.07	90.6	90.6	90.60	90.6
	B	1101524	2239	BERT Wordpiece	BERT	**97.8**	**99.3**	**98.1**	98.3	98.05	98.2
BioBERT	A	1101524	2239	BioBERT Wordpiece	BioBERT	89.4	99	98	86.2	**98.71**	91.7
	B	1101524	2239	BioBERT Wordpiece	BioBERT	97.1	99.1	96.7	98.6	96.26	97.4
ClinicalBERT	A	1101524	2239	BioBERT Wordpiece	ClinicalBERT	93.5	96.9	96.2	92.7	97	94.4
	B	1101524	2239	BioBERT Wordpiece	ClinicalBERT	**97.8**	**99.3**	97.4	**99**	97.03	**98.2**



(a) Training

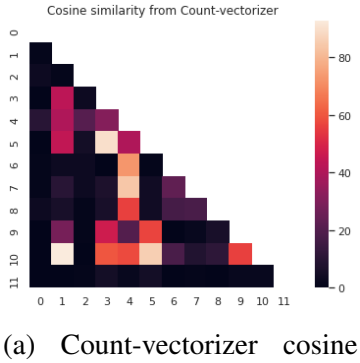


(b) Validation

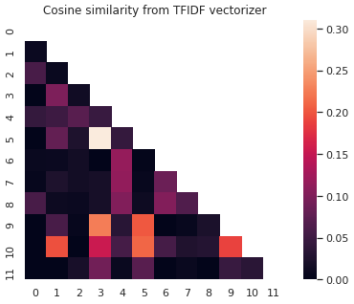


(c) Val. accuracy

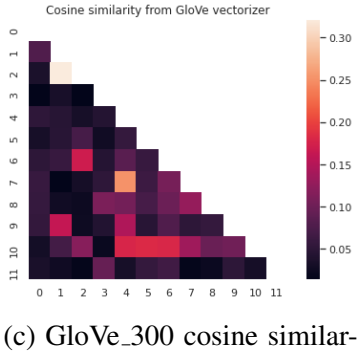
Figure 2.3: BERT-based models training.



(a) Count-vectorizer cosine similarity

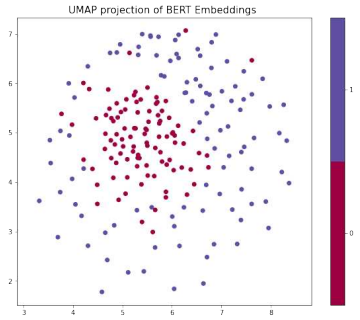


(b) TF-IDF cosine similarity

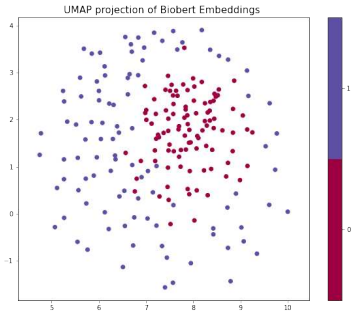


(c) GloVe_300 cosine similarity

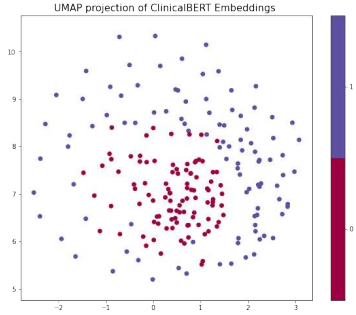
Figure 2.4: Cosine similarity for non-contextualized models.



(a) BERT embeddings clustering



(b) BioBERT embeddings clustering



(c) ClinicalBERT embeddings clustering

Figure 2.5: Contextualized embeddings clustering.

Alternatively, employing the dimensionality reduction technique known as Uniform Manifold Approximation and Projection (UMAP), we reduced the dimension of the contextualized embeddings obtained from 100 notes per class. Figure 2.5 showcases distinct clusters corresponding to the two classes prior to the learning and classification processes. These figures substantiate that the utilization of UMAP for dimension reduction effectively facilitates the identification of similarities between the embedding vectors derived from the two classes.

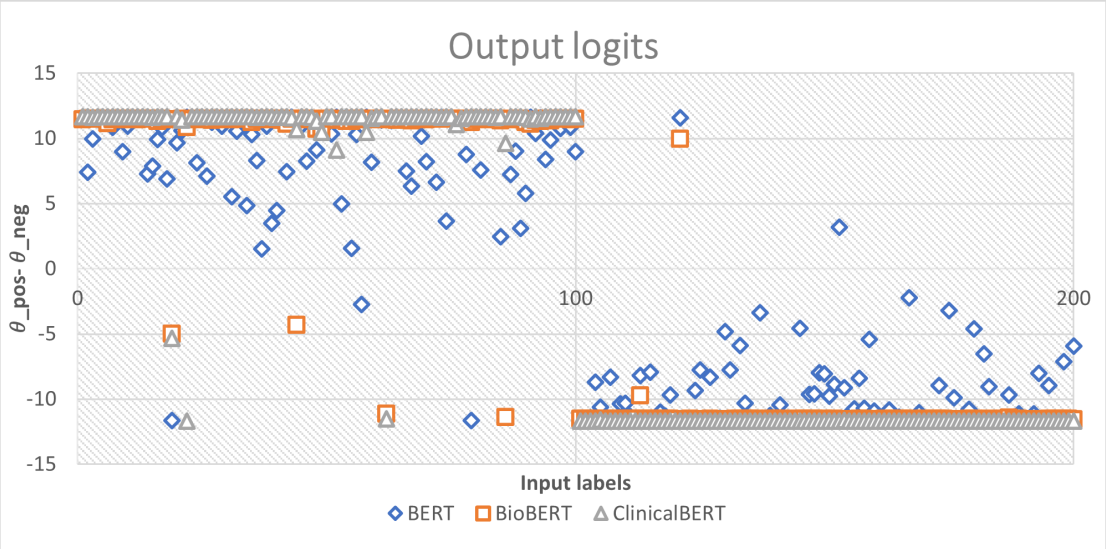


Figure 2.6: Logits from 200 input samples represented by $y = \theta_{pos} - \theta_{neg}$.

To assess the reliability of advanced models in terms of prediction certainty, we conducted an analysis using a random sample of 200 instances. The logits, obtained from the final layer of each model prior to the activation function, were extracted for further examination. Figure 2.6 illustrates the distribution of unnormalized scores (θ) across the three models, with each class represented by 100 consecutive samples. Specifically, sentences from deceased patients are depicted on the left, while those from discharged patients are presented on the right. The y-axis represents the evaluation scale, reflecting the polarity of each model's output. The results demonstrate that both BioBERT and ClinicalBERT consistently exhibit superior distance scores between the two classes compared to BERT. This indicates that when a score approaches 0, the probability for the corresponding sequence to belong to a specific class is approximately 0.5. Consequently, such scores can be interpreted as low confidence in the prediction.

2.7 Discussion

With this research, pneumonia patients were selected among ICU EHR data. Through the NLP pipeline, we aim to demonstrate the ability to predict outcomes using the narratives taken during a patient's stay by comparing the existing NLP approaches. Cleaning thoroughly improves the contextual understanding of the inputs, as demonstrated by ClinicalBERT with an MCC score of 4.3%.

BERT-based domain-specific models can perform slightly better than the general BERT, but the difference resides mainly in the convergence time between the training and the validation process. In case we want to use independent entities, such as

NER, BOW models are more suitable to handle prediction tasks because only term frequency over inverse document frequency is more relevant. Nonetheless, the preprocessing methods may change with a different EHR that utilizes a different notation. Besides the performance of a prediction model, it should be interpretable. For our case, the accuracy of the prediction should be quantified by diagnosis, drugs, bioinformation, or even demographics. However, the high dimensionality of modern NLP models contains abstract features for which we have no words or mental concepts. To visualize which feature or medical terminology activated the model along with the input text does not necessarily have a meaning for us. Therefore, we judged that interpreting the outcomes from the narratives is beyond the scope of this research.
There are also some limitations of this study. First, we limited this analysis to pneumonia patients who stayed in the ICU, and our prediction test used admission notes. NLP models require a lot of data to be generalizable and avoid over-fitting the models. Therefore, our guarantee for reproducibility using different domain data is limited.
A second limitation is common to EHR-driven prediction models for supervised learning. It is rare to have a sufficient balance between classes; however, for our case, the minority class represented by 29% of the data did not give an alarming false-negative for high-dimensional models, and its precision score was as high as the majority class.

2.8 Summary

Through this chapter, we present a deep comparison of Neural Language modeling pipelines for outcome prediction from medical text notes using pneumonia patients. We compare the performance of medical notes preprocessing, their representation as well as a supervised learning mechanism to predict outcomes of ICU admissions. We demonstrated that text preprocessing is of paramount importance as the first step of the pipeline. Light preprocessing will not achieve results as good as deeper processing. Replacing medical jargon and abbreviation terms to harmonize the data will have a high positive impact. For example, changing ”dx PE” to ”diagnosis Pulmonary Embolism” will allow models to add more weight to each of those tokens as related to pneumonia and appears multiple times. However, extreme processing such as NER will limit the applicable models besides cutting off some useful information from the text. The choice of embeddings depends mostly on the input type, size, and domain. Current NLP models, utilizing transformers as the fundamental structure, understand the medical text better at the expense of prediction interpretability. A meticulous data cleaning and subword level representation from a medical domain embedding and a fine-tuned transformer-based model yielded better optimal results.

Chapter 3

Bridging the Gap between Medical Tabular Data and NLP Predictive Models: A Fuzzy Logic-based Textualization Approach

Recent works have demonstrated an impressive ability for transformers-based models of learning and predicting from various contexts. Our approach is inspired by the Fuzzy theory of segmenting and representing continuous values into delimited ranges to represent a modelable entity. We used that idea to transform the numerical clinical data into descriptive narratives. We aggregated the administration data, the diagnosis, the vital signs, the procedures, and the laboratories. The prediction results from this natural language-generated text show competitive results that can average the numerical-based outcome prediction performance.

With machine learning and artificial intelligence, researchers have presented models to predict inpatient outcomes using structured or unstructured medical information collected through EHRs systems. We propose an approach to unify the accuracy of NLP models and the completeness of the medical tabular data using a Fuzzy Logic (FL) theory by generating artificial narratives from the medical tabular data to describe the patient's period of hospitalization. To evaluate our approach, we performed an extensive application on a downstream NLP text classification task to predict in-hospital mortality. Additionally, we demonstrated the importance and competitiveness of our approach by comparing our results with a tabular medical data benchmark publication, which shows an F1-score of 93,7% using tabular medical data, while the evaluation of our best NLP model yielded a similar F1-score, it has better sensitivity(recall) score of 93.11% on an intensive care unit mortality prediction task. This approach of generating artificial narratives is important to open new paths of using NLP in the medical area for predictive or entity recognition models.

3.1 Introduction

EHRs generally contain various information concerning a patient to promote continuity of care among caregivers. This data may be from any source or format grouped into two subclasses [19]. On the one hand, we have well-documented structured data

describing the patient with demographic information, diagnosis, monitoring data, and others. However, the characteristics of this data may vary from one patient to another depending on their disease or period of stay. On the other hand, unstructured data such as narrative reports are noisy and particularly challenging for NLP applications. The quality of this EHR data is arguably different from general care to intensive care. Due to the superior medical care provided in ICU, their data are not always easily accessible. Structured medical data impose a feature selection as a data regularization mechanism for standard models such as neural networks, tree-based models, or other mainstream modeling methods [69].

With the adoption of machine learning algorithms, artificial data have been one of the predominant solutions to tackle some challenges such as imbalanced data for classification, data augmentation, and generation for image processing and language translation [70]. In NLP, Gated Recurrent Unit (GRU) [71] has achieved impressive performance in text data generation for machine translation and medical synthetic data generation [72, 73]. Clinical report notes contain tremendous information about the patients, events, diagnoses, opinions, and different interventions made by a multidisciplinary team. The purpose of the notes is to share relevant information for informed decision making users ignore the existence of a set of guidance dictating the format and the grammar of clinical notes. However, using NLP for such data requires certain standards since document representation relies on dictionaries and vocabularies from common natural languages [74]. On the one hand, using structured data with standard machine learning models or clinical reports with NLP models have different challenges and drawbacks. On the other hand, studies have shown promising possibilities of building knowledge-based models by employing FL rule-based algorithms for medical diagnosis systems [75]. Few researchers propose a focus on the data to adapt it to the existing pipeline [76] for the sake of managing imprecise and vague knowledge [77]. Generating synthetic narratives of a patient is a way of creating domain-oriented data for models to guide attention to the most essential features and sensitive information.

Through this research, we propose a rule-based pipeline to describe a patient using structured data by generating a text document and evaluating the usefulness of the synthetic summary by transformers-based models to predict a patient's outcome.

The first step consists of feature selection inspired by a baseline study from the literature review [28], then from the selected features. We divide the data extraction task into small clinical services-related data clusters. We then textualize the features using preconceived prompts according to the availability of the feature values. We finally experimented the importance of the generated text on a downstream text classification task using several transformers-based NLP models such as an optimized RoBERTa based model [78],BERT [79] and a pre-trained biomedical language representation model (BioBERT) [80] models.

The contribution of this research is summarized as follows:

- We are proposing a novel approach that consists of generating clean and comprehensive medical narratives to describe a patient through a textualization process of the medical tabular data.

- A superficial application of the Fuzzy theory through a defuzzification to create a syntax dictionary substituting the numerical values of medical parameters. This

textualization preserves the uncertainty and vagueness inherent in medical data while still allowing for the application of NLP methods.

- An extensive study of using the generated data to solve a patient outcome prediction problem based on an NLP classifier optimization method.

As related to this research, to the best of our knowledge, no prior research has been conducted to demonstrate the ability to transform tabular data into text to apply NLP to perform a task such as a prediction.

3.1.1 Outline

This chapter presents a novel approach to combining structured medical data with natural language processing (NLP) techniques to improve predictive modeling and explainability.

The chapter is divided into five main sections.

1. *The first section 3.1* introduces the problem of using traditional methods on structured medical data and relates the limitations of NLPs in the medical field. This section also introduces the proposed solution, which is to use fuzzy logic-based textualization to transform unstructured medical narratives into structured data that can be used in predictive models.

2. *The section 3.2* provides a literature review of related work. It highlights related publications and preliminary concepts. We also discuss the limitations of existing approaches to FL and textualization, which typically rely on manually defined rules or templates and do not scale well to large datasets.

3. *The section 3.3* presents the proposed approach in detail. It describes how FL can be used to generate linguistic variables from medical values and how these linguistic variables can be combined with natural language processing techniques to generate medical narratives. We describe the algorithm of our methods and give examples of how this approach can be used to describe a patient's hospital stay.

4. *Section 3.4* presents results from experiments using real-world medical data. We demonstrated the usefulness of our proposed approach for predicting patient outcomes, particularly when it comes to dealing with noisy and unstructured data. We also demonstrate the ability of our generated narratives to provide highly interpretable outcomes.

5. Finally, *section 3.5* provides a conclusion and discusses limitations and future work.

3.2 Literature Review

Biomedical data mining aims to extract knowledge from large amounts of biomedical data. The goal of this process is to identify and understand patterns and relationships within the data that can be exploited later to improve healthcare and understand the outcome. With machine learning, biomedical data mining requires a data transformation

that involves converting raw data into a format that can be easily manipulated with the available tools for better performance [81]. Various normalization techniques include:

- *Standardization*, which scales data to a common range.
- *Normalization*, which scales data to a common distribution.
- *Discretization*, which converts continuous data into discrete data.

Data discretization can be performed by binning which groups data into a specified number of bins, or by clustering data based on similarity. Discretization strives to improve the interpretability of biomedical data. For EHR data, these methods can be computationally expensive but can also lead to a massive loss of information.

In recent years, many studies have proposed various techniques to process and analyze medical data. For instance, deep learning models have been used to predict clinical outcomes, such as patient mortality, length of stay, and readmission rates, using electronic health records (EHR) data [82, 83]. A study by Choi et al. [84] proposed a recurrent neural network (RNN) model that uses clinical notes, to predict hospital readmission. Their approach proposes an interpretable predictive model for healthcare that uses a reverse time attention mechanism to capture relevant information from the patient's historical medical records. Similary, a study by Purushotham et al. (2018) proposed a deep learning model that incorporates both structured and unstructured data from EHRs to predict patient mortality [83]. In data transformation, several works have been presented. For instance, Arnaud et al. [85] proposed a distillation method to extract structured data from unstructured text.

However, few studies have suggested transforming structured data into unstructured free text. The structured data are naturally accurate for machine learning models and interoperability, while NLPs are still a black box. Subsequently, processing unstructured data, such as clinical notes, can be challenging due to the variability and complexity of clinical language [86].

In their book, Jang et al. [87] proposed a comprehensive theory on applying FL and machine learning to address the uncertainty in a data transformation while emphasizing the interpretability of the result. This work inspires us to solve the vagueness inherent in medical data.

3.2.1 Fuzzy Theory

Traditionally, FL is a science that makes machines think and understand the way humans do [88] by proposing fuzzy sets to manage imprecise and vague knowledge [89]. As a computational Intelligence technique, for effective decision-making, fuzzy methods are used to bridge the gap between human and machine intelligence by resolving the ambiguity of terms. The paradigm of computing with words was a rational consequence of the fuzzy theory reasoning for computers [90]. However, FL, in its concept of a linguistic variable and application to approximate reasoning, is a method of computing with words [91]. While Today's technologies can only simulate that computation, we still cannot compute with words as long as the encoding process transforms words back into numbers. Therefore, this approach can be assimilated into a rule-based algorithm to define numerical variables with words. While numbers are used in statistical

and machine learning models, humans understand better in natural language. Therefore, using NLP should require a data transformation of the numerical values into terms that are more meaningful for such models.

3.2.2 Hybrid Fuzzy-based Models for Text Generation

Hybrid modeling integrating Deep Neural Network (DNN) and fuzzy systems has been defined in various ways for diverse reasons [92, 93]. One of the main motivations for that symbiosis is DNN optimization [94, 95]. As an illustration of how FL is used in NLP is in Natural Language Understanding (NLU). Fuzzy logic can be used to interpret the meaning of a natural language input by taking into account the context and the degree of uncertainty of the input [96]. For example, a statement like *"The patient has a high blood pressure ..."* could be interpreted differently. Fuzzy logic can be used to determine the degree of membership of the input in different categories, such as "$normal$", "$elevated$", "$High$" or "$Hypertensive$" to make a more accurate interpretation of the input based on that information.

Another example of the use of fuzzy logic in NLP is in Natural Language Generation (NLG). Fuzzy logic can be used to generate natural language output that is more human-like and less rigid than traditional rule-based systems by taking into account context and degree of certainty [90, 97]. Let's assume a set of linguistic variables that represent different features or attributes of the text:

$$X = x_1, x_2, ..., x_n \tag{3.1}$$

and a set of fuzzy sets that represent the values of the linguistic variables.

$$A = A_1, A_2, ..., A_n \tag{3.2}$$

The membership functions of the fuzzy sets are used to represent the degree of membership of each value in a linguistic variable.

Fuzzy logic can be utilized to generate text by using a fuzzy inference system, which consists of a set of rules that define the relationships between the linguistic variables. The rules can be defined as *IF* x_1 *is* A_1 *AND* x_2 *is* A_2 *THEN* x_3 *is* A_3. The rules are used to emanate a set of fuzzy output variables that are fused and a reverse engineering mechanism(defuzzification) is applied to generate the final text. This fuzzy text generation can be expressed as:

$$y = \sum_{i=1}^{n} (w_i * \mu A(x_i)) \tag{3.3}$$

Where y is the output text, w is the weight of each rule, and $\mu A(x)$ is the membership function of the fuzzy set for each linguistic variable.

3.2.3 Defuzzification

The fuzzy membership degrees are used to define a crisp output or a single, definite-meaning representation of the input [98, 99]. This reverse engineering mechanism has three main methods:

- *Centroid Method:* It calculates the center of mass of the fuzzy set, which describes the average value of the set.

$$x_{centroid} = \frac{\sum_{i=1}^{n} x_i * \mu A(x_i)}{\sum_{i=1}^{n} \mu A(x_i)} \tag{3.4}$$

  Where $x_{centroid}$ is the crisp value resulting from defuzzification, x_i is a sample value, and $\mu A(x_i)$ is the membership degree of x_i in fuzzy set A.

- *Maximum Membership Degree Method:* This method specifies the value with the highest membership degree as the crisp output.

$$x_{max} = \arg\max_{x} \mu A(x) \tag{3.5}$$

  Where x_{max} is the crisp value resulting from defuzzification and $\arg\max_x$ is the argument that maximizes the membership function.

- *Mean of Maximum (MOM) Method:* MOM method calculates the average value of the values that have maximum membership degrees.

$$x_{MOM} = \frac{\sum_{i=1}^{n} x_i * [\mu A(x_i) = \mu_{max}]}{\sum_{i=1}^{n} [\mu A(x_i) = \mu_{max}]} \tag{3.6}$$

  Where x_{MOM} is the crisp value resulting from defuzzification, x_i is a sample value, $\mu A(x_i)$ is the membership degree of x_i in fuzzy set A, μ_{max} is the maximum membership degree in fuzzy set A, and $[\mu A(x_i) = \mu_{max}]$ is a binary variable equal to 1 if $\mu A(x_i) = \mu_{max}$ and equal to 0 otherwise.

With this research, we are converging this traditional use of fuzzy logic theory in NLU and NLG. We are proposing a way of using balanced linguistic theory and clinical features occurring in a tabular format to build comprehensive patient descriptive documents using defuzzification methods. In the following section, we describe our approach to the construction of the fuzzy set, and how our application yielded the best performance on a patient outcome prediction task.

3.3 Approach and Methods

3.3.1 Introduction

The ultimate goal of textualizing tabular data is to propose a predictive model based on a general understanding of a patient's status. The best part of this is the use of as much information as available from the EHR, without compromising on using certain parameters in the process of handling missing data and outlier values. Numerical models require the regularity of the input features. However, medical data are full of such irregularities that an extensive data processing step, including sample selection, data balancing, and normalization, is needed. Ideally, a patient's complete medical description should:

- Include patient's name, a unique identifier, and location of hospitalization

- Reflect the continuum of patient care in a chronological order contain data recorded on admission, handover, and discharged
- Be dated and signed by its author.

In most of the publicly available medical datasets, this important information is missing due to the de-identification process. This won't make any exceptions for the synthetic data when describing a patient.

In EHR, an equivalent description can mostly be found in clinical reports, which usually use natural language to describe a patient. However, free narratives are irregular and hard to process due to conventional writing, which can vary from one health center to another. Our objective is to describe a patient in a natural way using medical data, mimicking real-world datasets. Moreover, we want to bring the benefits of NLP and transformers to more use cases in medical predictive models while avoiding the pre-processing step required by the EHR narratives and comparing our results with the existing tabular-based models.

3.3.2 Data Acquisition and Mining

Medical Information Mart for Intensive care-III(MIMIC-III) is a publicly available dataset with real medical data from over 38,597 distinct patients admitted to the intensive care Unit (ICU) [100]. The data are distributed as CSV files that can be imported and mapped to a relational database such as MySQL. Using SQL queries, datasets were extracted and processed in a python notebook. In order to benchmark later the effectiveness of our proposed method, we utilize the same data inclusion criteria as our baseline model from the literature [28]. We selected patients admitted or transferred to the Cardiac Surgery Recovery Unit (CSRU), Medical ICU(MICU), Surgical ICU (SICU), and "emergency or Urgent" as ADMISSION_TYPE. Please, refer to the mentioned paper for details on the inclusion criteria and statistics. To keep the relations between entities for the next step, we query the database in five distinct dataset clusters containing Administrative information, Diagnoses related information, laboratory tests, vital signs, and procedures Fig. 3.1 shows a summary of the process from the data extraction down to the next generation.

A common problem for any medical outcome prediction studies is class imbalance. Within our dataset, few patients are those who died during their hospitalization in the ICU, representing a minority of 5058 among 37111 unique admissions. This shows a fatality rate of 13.62% in our population. Different techniques for handling imbalanced data exist; for our case, in order to keep the integrity of the data, downsampling the majority class by a random selection was utilized. However, this technique has the consequence of cutting out some potential knowledge from the majority class. To limit this information loss, we sampled the new dataset to 40% for the fatality class and 60% for the discharged class.

3.3.3 Proposed Model: Data Textualization

Literature has shown the most relevant features to predict a patient's outcome in an ICU [28, 82]. However, most of the authors are also limited by constant missing data across the population sets and the targeted case of study. In our case, our limit could be determined by the NLP model itself for not performing well with numerical data. As

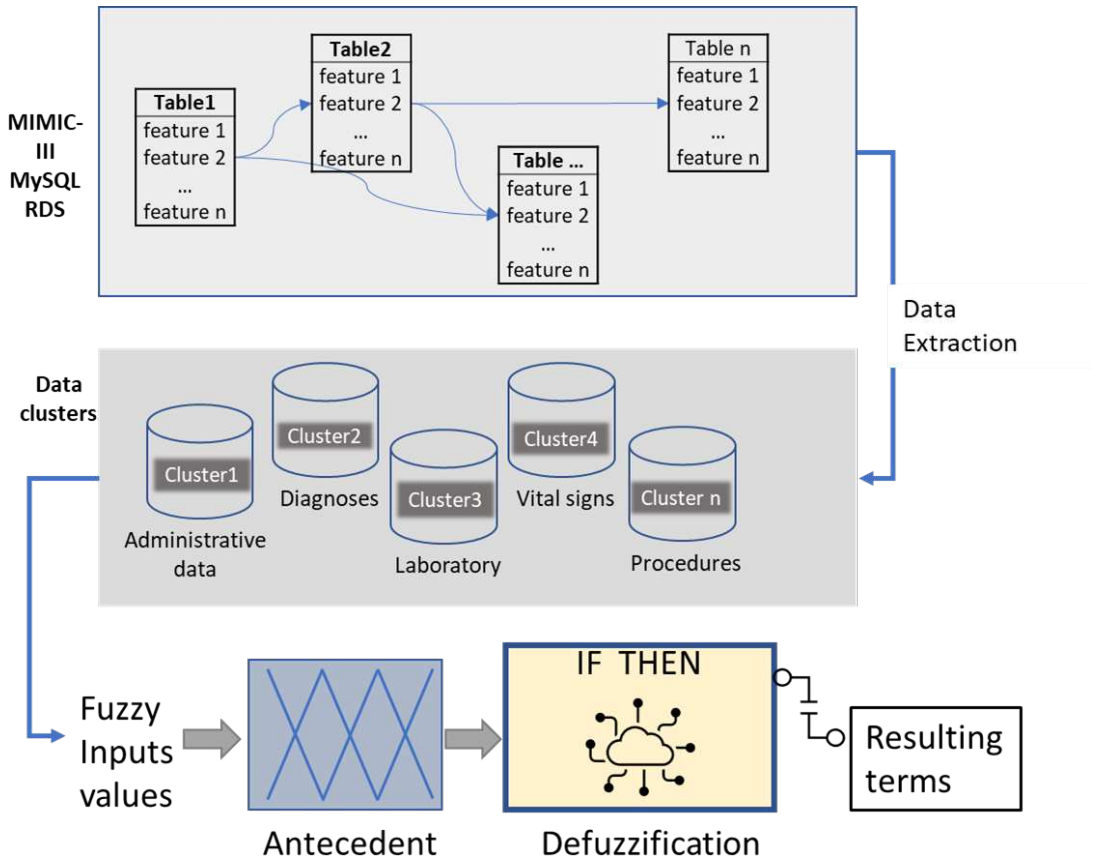


Figure 3.1: Summary of the data extraction and synthetic narratives generation pipeline

described in the following section, a rule-based algorithm based on fuzzy logic theory can be used to map numerical values with classes that can be understood by a modern LM. Generating synthetic text data with fewer numbers and more key terms allows us to build a more comprehensive NLP model to accomplish a task such as classification.

To describe a patient inherently, we conducted the generation of the narratives with the help of key phrases. These key phrases connect medical parameters extracted from different EHR tables to ensure semantic and syntactic integrity and relevance of the generated text.

3.3.4 Feature Engineering

One of the most typically used methods in fuzzy logic is the membership function, which assigns a degree of membership to each element of the input set, based on its resemblance to the set or category in question.

MIMIC-III dataset uses the International Classification of Diseases in its ninth version (ICD9) to encode and classify diagnoses. These codes are the primary source of information related to the patient's main complaint, comorbidities, and phenotypes. For our textualization process, we extracted all the codes related to each patient of our population found in the ADMISSIONS and DIAGNOSES_ICD tables. To map the codes with their label, we used a python library (icd9cms) [101] that takes the ICD9 codes as an input and we specifically output the most granular label from the hierarchy of the ICD9 nomenclature tree. This step provides us with the clinical name of the ICD9 code which we append to other prompts and texts related to patient identification (administrative information), procedures, vitals, and laboratory test and results.

In the following section, we describe our methods of transforming all those numerical data into text using a defuzzification system.

3.3.5 Defuzzification System Architecture

Our ultimate goal is to evaluate our approach to a language model. However, language models understand better textual context than numerical context. Therefore, a patient with a blood pressure annotated like *"140/101 mmHg"* has not much meaning for a language model. However, its interpretation in medical terms **(hypertension, specifically stage 2 hypertension)** has more potential to be well understood by an LM.

The fuzzy theory defines the linguistic variable by:

$$(X, T(x), U, G, M) \tag{3.7}$$

X is the variable and $T(x)$ is the set of terms, U is the universe of discourse, G represents the syntax rules, and M defines the semantic rules. For our case, X represents the medical variables, $T(x)$ are clinical interpretations, and U groups the values of each medical parameter [91].

A defuzzification dictionary for blood pressure readings could be mapped to categories such as "Low", "$Normal$", and "$High$". In our algorithm, instead of computing the centroid of the fuzzy output, we simply compute the maximum degree of membership among all categories using a binary rule approach. This approach can be more efficient and easier to implement, but it may not be as accurate as the centroid approach in certain cases.

Algorithm 1 Defuzzification for Blood Pressure Categories (Binary Rule)

Fuzzy blood pressure reading x Blood pressure category Compute the degree of membership of x in each category using fuzzy sets or rules, such as:

- Low: $\mu_{Low}(x) = \begin{cases} 1, & \text{if } x \leq 90 \text{ mmHg} \\ 0, & \text{otherwise} \end{cases}$
- Normal: $\mu_{Normal}(x) = \begin{cases} 1, & \text{if } 90 \text{ mmHg} < x \leq 139 \text{ mmHg} \\ 0, & \text{otherwise} \end{cases}$
- High: $\mu_{High}(x) = \begin{cases} 1, & \text{if } x > 139 \text{ mmHg} \\ 0, & \text{otherwise} \end{cases}$

Compute the maximum degree of membership among all categories, such that:

$$\mu_{max} = \max \mu_{Low}(x), \mu_{Normal}(x), \mu_{High}(x)$$

If $\mu_{max} = \mu_{Low}(x)$, return "Low" as the blood pressure category.
If $\mu_{max} = \mu_{Normal}(x)$, return "Normal" as the blood pressure category.
If $\mu_{max} = \mu_{High}(x)$, return "High" as the blood pressure category.

The membership function $\mu_{Normal}(x)$ returns a value of 1 if the blood pressure reading x falls within the range of *90 mmHg* to *139 mmHg*, indicating that the reading is *"Normal"*. The value of $\mu_{Normal}(x)$ is 0 for readings outside of this range. Similarly, membership functions can be defined for each of the other categories which we intend to substitute with words in the textualization process. However, the encoding part of the logic can be handled by a multidimensional LM to vectorize these entities of words. As in fuzzy theory, where each linguistic variable is described by a "set of terms", to textualize our medical features, each feature's value is represented by a term instead of a number. Our approach utilizes binary discrimination [102] to allocate a category to each value. However, this approach has one limitation. On the one hand, the accuracy of the resulting model depends directly on the size of the universe of discourse grouping the classes, and on the other hand, for some features, there is no deterministic way of establishing boundaries between those classes. Table 3.1 reports the set of terms with the range and source of reference for each parameter.

3.3.6 Machine Learning Models

In NLP, transformer-based models [1] have become a reference as the state-of-the-art on several natural language understanding tasks. In this research, we decided to use this representation over the fuzzification since it captures relations between neighboring and distant words while the Fuzzy encoder only considers one single word as an independent entity.

BERT

BERT model is a high bidirectional, unsupervised language representation pre-trained on unlabeled plain text corpus from books and English Wikipedia. The original

Table 3.1: Medical parameters, set of category terms and their ranges

Parameter	Range	Category	Reference
Age	15-40	Young adult	[103]
	41-60	Middle-aged adult	
	61-89	Old-aged adult	
	90+	Very old-aged adult	
Arterial Blood Pressure	Sys $<$ 90mmHg	Low	[104]
	Sys:90-139mmHg	Normal	
	Sys $>$ 139mmHg	High	
Heart rate(HR)	$<$ 60 BPM	Low	[104]
	60-100 BPM	Normal	
	$>$ 100 BPM	High	
SpO2	$<$ 92% BPM	Low	[105]
	$>$ 92% BPM	Normal	
Heart Rate(HR)	$<$ 60 BPM	Low	[104]
	60-100 BPM	Normal	
	$>$ 100 BPM	High	
Respiratory Rate	$<$ 12 BPM	Low	[104]
	12-25 BPM	Normal	
	$>$ 25 BPM	High	

model was presented with two versions, the $BERT_{BASE}$ with 12 encoders and 12 self-attention heads and $BERT_{LARGE}$ with 24 encoders and 16 bidirectional self-attention heads. We omit more details on the architecture as it is well described in [79]. BERT utilizes the transformer encoder architecture based on a self-attention mechanism to represent a sequence of words or tokens in a higher dimensional space. We utilized the $BERT_{BASE}$ version since our inputs had an average of 353 tokens.

BioBERT

The Biomedical language representation model for biomedical text mining (BioBERT) is a domain-specific language model [80]. This baseline model initialized its weights from BERT and uses PubMed abstracts and PMC full-text articles to fine-tune its understanding of the medical domain. Please, refer to the original paper for more details on the training process and the performance of the resulting model.

During the tokenization process, two additional tokens are used: a [CLS] token as an input starter and [SEP] to mark the end of the input sequence. Thus, a sequence S for these models is represented by $[cls, t1, ..., tn, sep]$ where t is a word or a subword of S. The maximum length of the input sequence is 512 tokens. The goal of using tokens is to represent any words and avoid OOV words. However, BERT and BioBERT are token based-models, thus, some words will be broken down into a character if that entity is not present in the 30.000 token vocabulary file of those models.

BioBERTa

BioBERTa is a pre-trained RoBERTa-based language model designed specifically for the biomedical domain [106]. Like other domain-specific LM, BioBERTa has been

trained on a diverse range of biomedical texts mostly electronic health records, and raw medical notes to learn the language patterns, terminologies, jargon, and knowledge relevant to the biomedical domain. BioBERTa was optimized in the pretraining process by adopting the modifications of the source model such as dynamic masked language modeling, no next-sentence prediction task, and most importantly, a WordPiece tokenizer that suppresses the out-of-vocabulary (OOV) occurrences. This model demonstrated high performance on several named entity recognition tasks and showed the best fertility rate for biomedical texts.

To fine-tune these three models for a classification task, we append a classification layer on top of the last hidden layer with a given loss function, and this can be performed on the output of the [CLS] token alone. For our case, we utilized the [CLS] token and a logistic regression classifier. We perform a hyperparameter search to find the best set of training epochs, learning rate, and batch size that optimizes the result [107].

3.4 Results

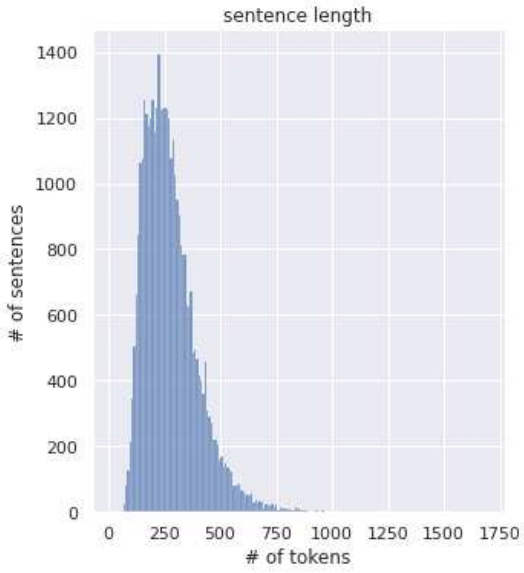
The process of generating data begins with the extraction of features from the main MIMIC-III dataset. The extracted features were then individually merged in a fusion process to form a more comprehensive representation of the patient. In this operation, the features were contextualized using key phrases to semantically link them, thereby creating a coherent representation of a narrative. To ensure the quality of the generated data, a grammatical assessment was carried out to eliminate any unnecessary duplication of features or syntax errors that may have been introduced during the fusion process. This grammatical assessment helped to improve the coherence and consistency of the generated data and provided a better sequence of the features extracted from the main MIMIC-III dataset.

3.4.1 Generated Data

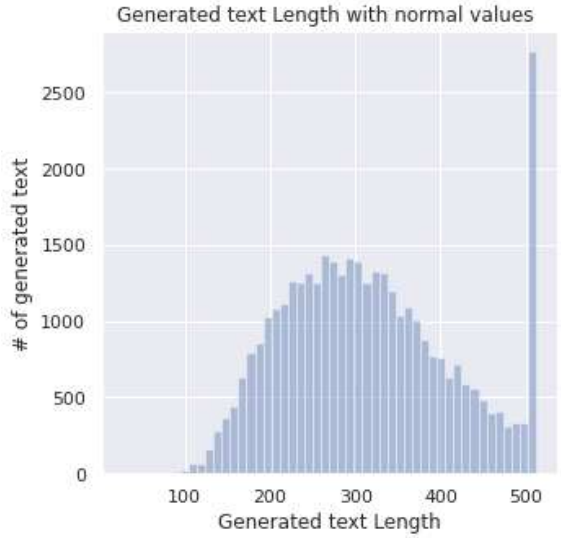
In order to provide a comprehensive understanding of each patient's hospitalization, we generated narratives for each of the 37110 admission IDs in the dataset. The length of these generated texts was determined by the number of parameters each patient had, resulting in a dataset that includes both admission IDs and labels indicating the outcome of the patient's hospitalization.

When preparing the generated dataset for use with classification models, it was essential to ensure that it would fit within the limitations of the models. Using the BERT tokenizer, we counted the tokens of each input sentence, and the results were 1686 and 67, respectively, for the longest and shortest sentence with a median of 258 tokens. With this in mind, we made the decision to exclude normal values of the parameters from the training data. The rationale for this was that the primary purpose of medical procedures is to identify or treat abnormalities. Figure 3.2 shows the variation of the narrative's lengths before and after this step.

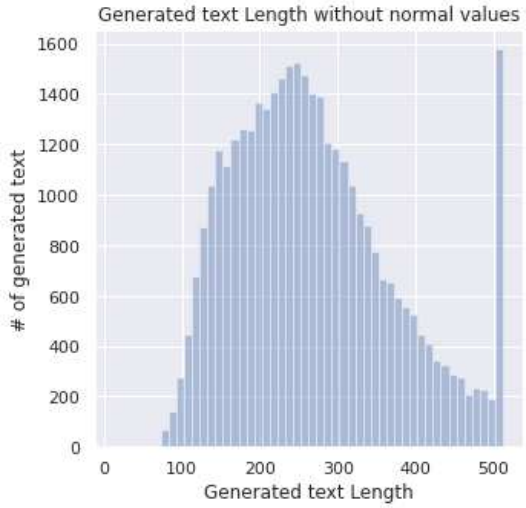
By keeping the normal values of the parameters, more than 2700 narratives have over the 512-token limit of our classification, while only less than 1600 will hit than limit without normal values.



(a) overview of the generated text



(b) Sentences of $\leq$ 512 tokens with normal values



(c) Sentence of $\leq$ 512 tokens without normal values

Figure 3.2: Generated narratives lengths: These three graphs provide an overview of the lengths of our synthetic texts.

3.4.2 Classification Results

To train and evaluate our two models, we used 10,116 input sentences and tested their performance on 2,529 narratives. To ensure compatibility, we utilized the bert-base-uncased tokenizer for BERT and BioBERT's tokenizer and the vocabulary that came with the pre-trained BioBERT files. For BioBERTa, it has a custom byte-pair encoding(BPE) tokenizer of 50265 tokens.

Input Length Variation Study

To understand the behavior and determine the optimal input size for the model, we conducted experiments using different input lengths of 512, 350, and 255 tokens. This allowed us to determine the most effective input size to achieve the best results. Preliminary results revealed that the best performance was achieved using a maximum input length of our models(512 tokens). For the rest of the experiment, we used tokenized inputs of a maximum length of 512 tokens for the three models.

Hyperparameters Optimization

Hyperparameter optimization in NLP consists of selecting the optimal values for the model's hyperparameters to achieve the best performance on a downstream task by effectively capturing the patterns in the data and avoiding overfitting or underfitting. These hyperparameters define the configuration of the model, such as the learning rate, the batch size, and the number of hidden layers. For our case, we focused our attention on the training batch size, the learning rate, and the training epochs.

To develop an adaptive (sequential) hyper-parameter search, we utilized a random search algorithm to erratically select different combinations in the provided ranges [108]. Figure 3.3a shows that an accuracy of 93.47 can be achieved using a batch size of 32 and 4 training epochs and a learning rate of 3.86e-05.

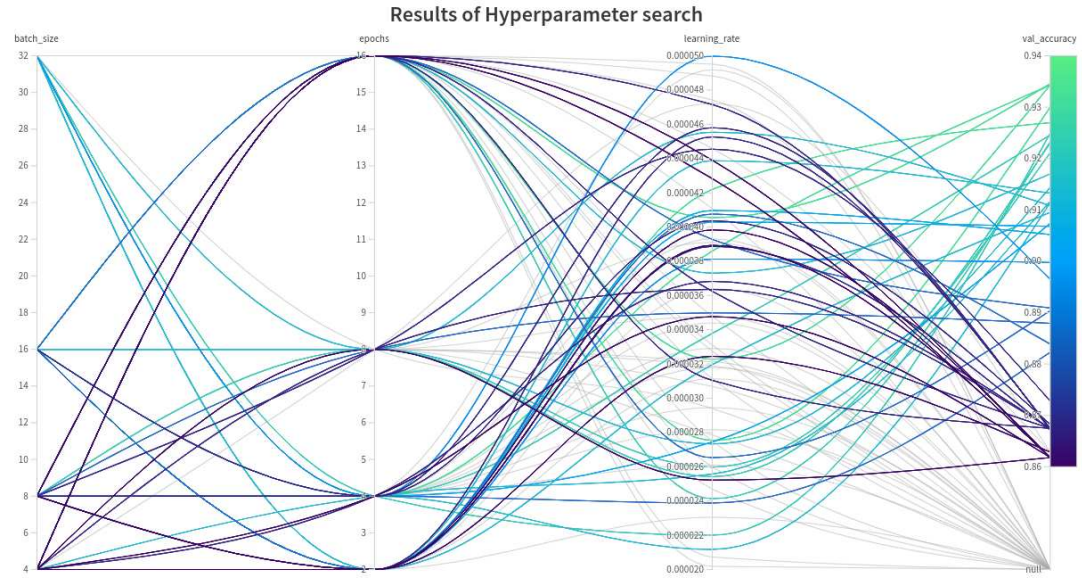
Outcome Prediction Results

Our evaluation was to use the generated data and evaluate its importance in solving the problem of hospitalization outcomes. Our data were labeled as "0" if the patient was discharged and "1" if died during his hospitalization.

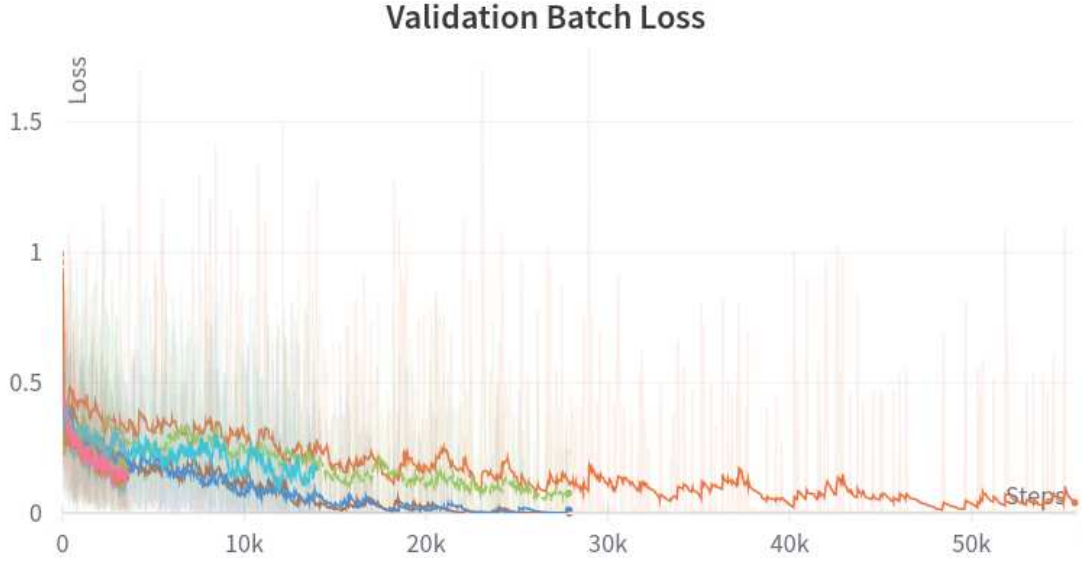
We measured the efficiency of the in-hospital predictive models by the evaluation metrics, F1-score, precision, and recall, between the fatality and survivor classes. Each reported score in Table 3.2 is an average of 5 experiments on both BERT, BioBERT, and BioBERTa models.

The benchmark paper [28] explores two different methods for approaching the task at hand. The first method involves the utilization of a list of unimodal baseline classifiers, including K-Nearest Neighbor (KNN), Multilayer Perceptron (MLP), Linear Discriminate Analysis (LDA), Logistic Regression (LR), and Decision Tree (DT), applied to various experimental feature sets. The second method involves ensemble models, such as random forest, voting, bagging, and boosting, to improve the performance of the best single models. These evaluations were conducted both with and without a feature-selection step.

Building upon these two approaches, the paper introduces a stacking classifier algorithm based on the generalization stacking ensemble model, using LR as the meta-



(a) Hyperparameter search on a predetermined range of values



(b) Loss on the different models' configuration

Figure 3.3: Hyperparameter Search and Validation Loss

Table 3.2: Outcome prediction results from three different NLP models and a tabular data-based stacking model as a baseline

Model	Input length	F1	P	Sensitivity	Specificity
Stacking Model [28]	-	**0.937**	**0.964**	0.911	-
BERT	L=256	0.849	0.815	0.886	0.767
	L=360	0.848	0.825	0.873	0.793
	L=512	0.858	0.847	0.870	0.832
	L=512(optimized)	0.897	0.887	0.895	0.882
BioBERT	L=256	0.851	0.817	0.887	0.770
	L=360	0.860	0.865	0.855	0.872
	L=512	0.881	0.894	0.908	0.869
	L=512(optimized)	0.925	0.931	0.926	0.934
BioBERTa	L=256	0.854	0.797	0.921	0.714
	L=360	0.860	0.845	0.875	0.825
	L=512	0.879	0.849	0.891	0.821
	L=512(optimized)	**0.937**	0.94	**0.931**	**0.946**

classifier. This stacking technique demonstrated impressive accuracy, with F1-score, precision, recall, and AUC scores of 0.937, 0.964, 0.911, and 0.933, respectively.

The results displayed in Table 3.2 demonstrate the highly competitive performance of both models, with BioBERTa exhibiting better performance than other language models. It is evident that fine-tuning the hyperparameters plays a crucial role in the model's performance, as the results show a difference of up to 6.5% in the f1-score. This highlights the need for proper tuning to achieve optimal results and underscores the significance of this aspect in the development of language models.

Figure 3.4 reports different results obtained by evaluating each model with varying configurations on the test set. We noticed a high variability in performance based on the model's hyperparameters.

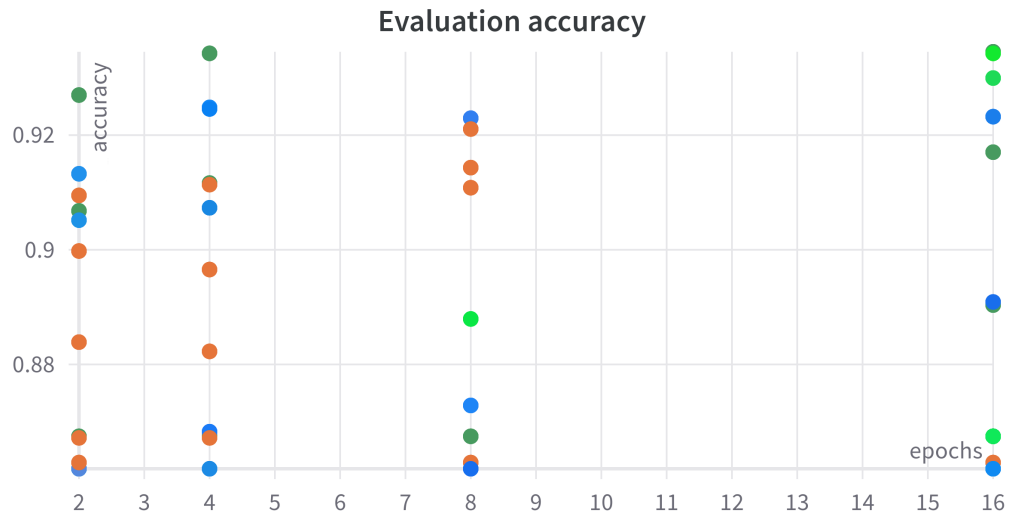
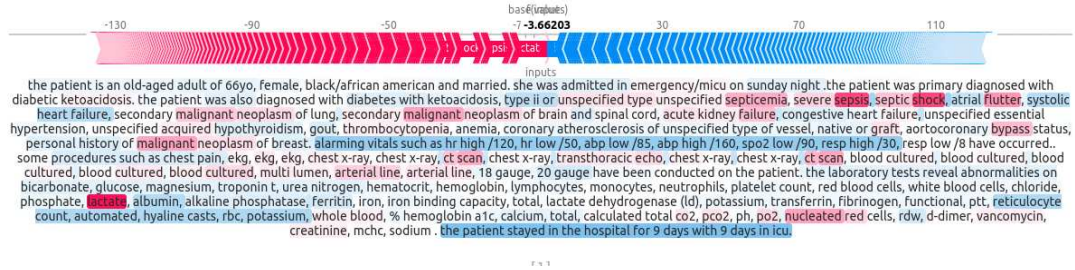


Figure 3.4: Our Different Models Prediction Accuracy

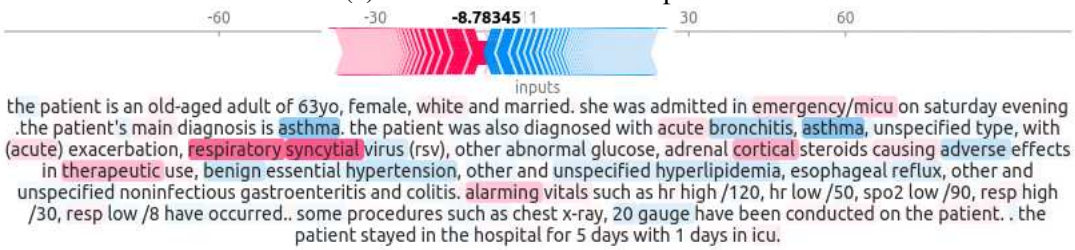
Overall, our approach has shown to perform comparably with the benchmark baseline models while exhibiting slightly improved results in terms of recall. We believe that this performance is the result of the specific data sampling technique that we implemented during the training phase, which aimed to balance the data distribution. By leveraging this approach, we were able to address the class imbalance and improve model performance effectively.

3.4.3 Interpretability of the Generated Text

The interpretability of models, as illustrated in Figure 3.5, plays a crucial role in understanding a model's decision-making process and predictions, especially in medical applications [109]. Using fuzzy theory in defuzzification processes helps to deal with uncertain and ambiguous information. Still, this uncertainty can also impact the interpretability of the models trained on such data.



(a) Narrative of a deceased patient



(b) Narrative of a discharged patient

Figure 3.5: Interpretability visualization using SHapley Additive exPlanations on the narratives from two different classes

The explainability of models on the text generated from a defuzzification process depends on various factors, such as the choice of the defuzzification method, the structure of the model, and the complexity of the generated text. The rule-based text provides more nuanced data by structuring the narratives into a more comprehensive and interpretable construction.

Figure 3.5 shows a visualization of BioBERTa of two generated texts using Shapley values [110], revealing the importance of each token. Red regions correspond to parts of the text that increase the model's output when they are included pushing the model to predict the patient as "deceased". In contrast, blue regions decrease the output of the model to predict a "discharged" patient.

It can be seen in Figure 3.5a that even if vitals and length of stay helped the model to increase the output values, the shade of red seen on "$sepsis$" and "$lactate$" was too high to predict the fatal outcome. This is more understandable as sepsis is life-threatening

on its own; a high serum lactate level as a consequence of sepsis may predict death within 24 hours [111]. Figure 3.5b shows a narrative where the primary diagnosis indicates a significant contribution to the survival of the patient despite an elevated value given to the patient's coinfection of "$respiratory\ syncytial\ virus$", a less lethal infection [112].

3.5 Conclusion

This research aimed to analyze the impact of the generated data on the prediction of in-hospital outcomes. The defuzzification process of generating narratives involved extracting features from the MIMIC-III dataset and fusing them to represent the patient exhaustively. The generated data were then subjected to a grammatical assessment to eliminate errors and improve the quality of the generated narratives. The data was generated for 37110 admission IDs in the dataset, and the length of the narratives varied based on the number of parameters each patient had.

To train and evaluate the models, 10,116 input sentences were used, and the performance was tested on 2,529 narratives. The BERT, BioBERT, and BioBERTa models were trained using the bert-base-uncased tokenizer and the BioBERT tokenizer, respectively. The study also involved hyperparameter optimization, where a random search algorithm was used to select the optimal values of hyperparameters, such as the batch size, learning rate, and training epochs. The best performance was achieved with a batch size of 32, 4 training epochs, and a learning rate of 3.86e-05.

The evaluation of the models was based on the prediction of the outcome of the patient's hospitalization, where the data was labeled as 0 for patients who were discharged and 1 for those who died. The results were measured using the F1-score, precision, and recall between the fatality and survivor classes. The results demonstrated the highly competitive performance of both the BERT and BioBERT models, with BioBERTa exhibiting better performance compared to the other language models. The results showed that the best performance was achieved using a maximum input length of 512 tokens, with hyperparameters optimized.

In conclusion, the study demonstrates that FL and rule-based approaches can play a significant role in generating comprehensive and interpretable medical narratives to extensively describe a patient. The results of the study demonstrate the potential of fine-tuned language models such as BioBERTa to improve the accuracy of predictions and provide a better understanding of the hospitalization outcomes of patients. The interpretability of models trained on the text generated from a defuzzification process is crucial for ensuring the transparency and reliability of the model's predictions.

However, our approach has two significant limitations. Firstly, the accuracy of the resulting model depends directly on the size of the universe of discourse grouping the classes, and for some features, there is no deterministic way of establishing boundaries between classes. Subsequently, this approach requires domain expertise to determine the appropriate linguistic rules and the potential for bias in the textualization process. In addition to that, the experimental results show that the performance of LM relies heavily on hyperparameter fine-tuning.

In future work, we plan to explore the use of neuro-fuzzy theory, in combination with current state-of-the-art LMs, and investigate methods for reducing expert dependency by incorporating external data sources such as ontology. Overall, this study

provides a step toward improving healthcare outcomes through data-driven decision-making processes.

Chapter 4

Optimization of Transformer-based Model for Medical Documents

4.1 Motivation and Contribution

With the ever-increasing availability of biomedical and clinical data, the use of natural language processing (NLP) techniques to analyze and extract information from unstructured medical narratives has become increasingly important. Language models, in particular, have shown significant promise in improving the accuracy of clinical decision-making and medical research. However, language models trained on general-purpose text datasets may not perform optimally on clinical and biomedical text, as such text often contains specific terminologies, abbreviations, and jargon.

In this chapter, we review recent studies on the optimization of language models for biomedical and clinical text. We highlight the challenges and opportunities associated with this task, as well as the various techniques that have been developed to improve the performance of language models on such text. As a result, we provide two language models trained and optimized for biomedical and clinical data as well as their evaluation and performances on several NLP tasks.

We first discuss the need for creating specialized language models for biomedical and clinical text. We then explore the use of domain-specific pre-training and fine-tuning techniques, including transfer learning, to improve the performance of language models. We also review the importance of different tokenizers to improve the in-domain semantic representation of clinical text.

Additionally, we examine the importance of hyperparameters fine-tuning in order to improve the robustness of language models on clinical and biomedical text.

Finally, we discuss the potential impact of optimized language models on various healthcare applications, including electronic health records (EHRs) and Clinical Decision Support Systems (CDSS). We also highlight some of the ethical considerations and challenges associated with the use of raw clinical data from EHR with pre-trained language models.

Overall, this chapter provides a step forward in the optimization of language models for biomedical and clinical text and outlines future directions for research in this field.

4.2 Introduction

As demonstrated in chapters 2 and 3, recent research has demonstrated the potential of language models for processing and understanding human expression for a wide variety of tasks in the general domain [113–115]. These techniques have greatly improved the general understanding of the biomedical text and information extraction by means of named-entity recognition (NER), relation extraction (RE), and classification [116, 117]. The adoption of electronic health records (EHR) by more than 86% of healthcare facilities in developed countries has increased the volume of biomedical data [118]. EHRs contain a tremendous amount of structured and unstructured data, which can be used to fine-tune predictive algorithms and drug compatibilities and help to understand the course of diseases and patients. Various researchers have proposed adapted NLP models to address better biomedical documents [116, 117, 119, 120]. With medical texts representing 80% of the EHR data [113], it's imperative to develop more robust and efficient language models which can be used to understand and extract relevant information contained in those texts.

Unstructured medical texts, which can be clinical notes, surgical records, discharge records, radiology reports, or pathology reports, are written primarily for communication purposes between healthcare actors. These texts are usually too long for conventional biomedical models that are generally built for a maximum of 512 position embedding. This limitation is mainly due to the quadratic computational and memory growth of the self-attention mechanism in the traditional transformer models [121]. Recent technics have emerged to propose sparse attention that grows linearly with the input length [122, 123].

In previous research, BioBERT [116], which was trained using BERT architecture [114] on English biomedical data from books, PubMed abstracts, and full-text articles, showed a significant contribution by enriching its dictionary with terms and expressions that were not included in the general domain corpus used by BERT. This critical step reduces the over-segmentation [124, 125] significantly, preserving more meaningful biomedical terms. However, pretraining a model exclusively on clinical data such as MIMIC [126] will prevent the model from expanding its application to general biomedical tasks.

While traditional transformer-based models have demonstrated the effectiveness of having a full attention-based model, their architecture has a high computational and memory cost limitation. In the clinical domain, this drawback yields models with poor performance in real-world applications. This study aims to demonstrate that adapting a model such as BigBird [122] to a biomedical domain with a focus on unstructured EHR data has the potential to contribute to the biomedical NLP community for any downstream tasks. Within this chapter, our contribution could be summarized as follow:

1. We introduce **BioBERTa**, a RoBERTa-based biomedical language model trained on biomedical and electronic health record corpora.

2. Utilizing a combination of random attention, window attention, and global attention, inspired by BigBird architecture, we provide a sparse attention-based model, referred to as **Medical BigBERTa**, which can handle eight times more tokens than the traditional models including **BioBERTa**.

3. Equipped with a biomedical dedicated tokenizer, we trained from scratch a sentencePiece tokenizer to enhance the embedding capabilities of in-domain terms,

grammatical errors, and conventional annotations.

4. We perform a modeling optimization, using a Bayesian-based algorithm to fine-tune hyperparameters on each dataset.

5. Finally, we aim to publicly share our models and their tokenizers with the research community, which we believe will help in other biomedical data mining studies.

The prevailing transfer learning methods for LMs take a general-domain LM and its vocabulary and fine-tune it with specific-domain data. However, some authors have suggested that domain-specific vocabulary can outperform that mixed-domain approach [127]. We assumed that this vocabulary inheritance has two consequences. It helps the general model to transfer its weights to tokens adequately, yet those tokens might not be similarly contextualized in the specialized domain because, as a source, the general domain text is substantively different from the target text. We thus conducted a tokenizer training that generated a new vocabulary file, combining biomedical and clinical corpora used to train and fine-tune the final model. The main difference between our approach and most BERT variants is that those LM are technically based on a continuous training approach where the source model is fine-tuned on a specific domain corpus [125], while ours adopts a similar approach as [120] by including in the process a dedicated tokenizer.

Table 4.1: Data description of the four datasets used for our experiment

Data name	Data size	#Seq	av. words/seq	Tokenizer training	Model training
Medical text for text classification	36MB	28.8K	183	36MB	36MB
Medical transcriptions	17MB	3.9K	409	17MB	17MB
PubMed title abstract baseline 2019	21.63GB	5.1M	218	3GB	21GB
MIMIC III	2GB	1.1M	1258	2GB	2GB
Total				5.05GB	23.05GB

4.3 Outline

- *In section 4.2*, we provide the background and motivation for the research, discussing the importance of a tailored LM and hyperparameter fine-tuning. We also present the research question and the contribution of this research.

- *Section 4.4* is a literature review that discusses previous research on biomedical neural language modeling and hyperparameter fine-tuning. We also provide an overview of the different approaches for in-domain optimization, modeling for long sequences, and various tokenization methods.

- *Section 4.5* describes our research methodology, including the dataset used, our tokenization process, evaluation tasks, and hyperparameter optimization in the

following section 4.6. We also provide an overview of our inspired architecture of the LM and the training process of our proposed two new models.

- *In section 4.7*, we present our experimental results. We compare the performance of our new LM on several downstream tasks. We demonstrate the importance of our new tokenizers as well as the hyperparameter search and the sparse attention mechanism.

- *Section 4.9* discusses the implications of our results and the effect of a dedicated tokenizer.

- Finally, *section 4.10* concludes the chapter by summarizing our findings, the limitations of our research, and potential directions for future research.

4.4 Literature Survey

4.4.1 Language Models

Language Models (LMs) have revolutionized the field of natural language processing (NLP) in recent years, achieving state-of-the-art results in a wide range of tasks such as machine translation, sentiment analysis, text classification, and question-answering. A language model is a type of NLP model that learns the patterns and relationships between words in a text corpus and uses this knowledge to predict the likelihood of a given sequence of words.

One of the earliest and most widely used language models is the n-gram model [128], which estimates the probability of a word given its previous n-1 words. However, n-gram models suffer from the curse of dimensionality and struggle to capture long-term dependencies between words [129]. This led to the development of recurrent neural network (RNN) [71] based language models such as the long short-term memory (LSTM) [46] and gated recurrent unit (GRU) models, which can capture long-term dependencies through their recurrent connections. Nonetheless, LSTMs can be computationally expensive and difficult to train on large datasets due to the complex nature of their architecture and the need to propagate gradients over many time steps [130]. Additionally, LSTMs can suffer from the problem of vanishing or exploding gradients, which can lead to issues with training stability and convergence [130].

More recently, attention-based transformer models such as BERT [114], GPT-2 [131], and T5 [132] have emerged as the state-of-the-art in many NLP tasks. These models use self-attention mechanisms [1] to capture the relationships between all words in a text sequence and have achieved remarkable results in tasks such as language generation, text classification, and question-answering. The key to the success of transformers is their ability to process entire sequences of words at once without being limited by the sequential processing of RNNs.

4.4.2 Tranformers

The transformer model consists of an encoder and a decoder, each consisting of a stack of multi-head self-attention and fully connected feed-forward layers. The self-attention mechanism allows the model to weigh the importance of different words in

a sequence when making predictions, while the feed-forward layers apply non-linear transformations to the input.

Given an input sequence of length T, the encoder maps it to a sequence of hidden states $H = h_1, h_2, ..., h_T$ as follows:

$$h_i = f(x_i) \text{ for } i \in [1, T], \tag{4.1}$$

where x_i is the i-th input token, and $f(\cdot)$ is a function that applies multi-head self-attention and feed-forward layers to the input.

The decoder then uses the encoder's output to generate a sequence of target tokens $Y = y_1, y_2, ..., y_U$, where U is the length of the output sequence. At each time step u, the decoder predicts the next token y_u based on the previous tokens and the encoder's output using the following equation:

$$p(y_u|y_1, y_2, ..., y_{u-1}, H) = g(y_{u-1}, z_u), \tag{4.2}$$

where z_u is a context vector computed by attending to the encoder's output, and $g(\cdot)$ is a function that applies multi-head self-attention and feed-forward layers to the decoder's inputs and the context vector. The global attention mechanism is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{4.3}$$

where Q, K, and V are the query, key, and value matrices, respectively, with d_k denoting their dimensionality. The dot-product of the query and key matrices is divided by $\sqrt{d_k}$ to avoid the dot product from becoming too large or too small, which could result in slow learning or numerical instability. The softmax function is applied to the scaled dot-product to compute the weights for each value vector. Finally, the output is computed as the weighted sum of the value vectors.

The transformer's effectiveness stems from its ability to capture long-range dependencies in sequences using self-attention, which allows it to process entire sequences in parallel and make more accurate predictions. Additionally, the use of residual connections and layer normalization helps mitigate the vanishing gradient problem and improve training stability.

Moreover, they can be fine-tuned on specific tasks with relatively few additional parameters, making them highly versatile and efficient. Figure 4.1 relates the transformer architecture.

4.4.3 Modeling Long Sequences

Most NLP models, such as BERT and RoBERTa are provided with longer versions that can encode text up to 1024 tokens. Few models, such as BioMegatron$_{800}$ [119], BioMegatron$_{1.2}$, and BioM-ALBERT$_{xxLarge}$ [133] were pre-trained in a multi GPU environment to provide models with up to 4096 hidden size, trading off the embedding position size to mitigate with the memory cost. The larger the hidden size, the more complex patterns and relationships the model can learn and the more accurate its predictions may be. However, a larger hidden size also requires more computational resources and may increase the risk of overfitting the training data, which can lead to poorer performance on new or unseen data. We provide in table 4.2 details of biomedical larger

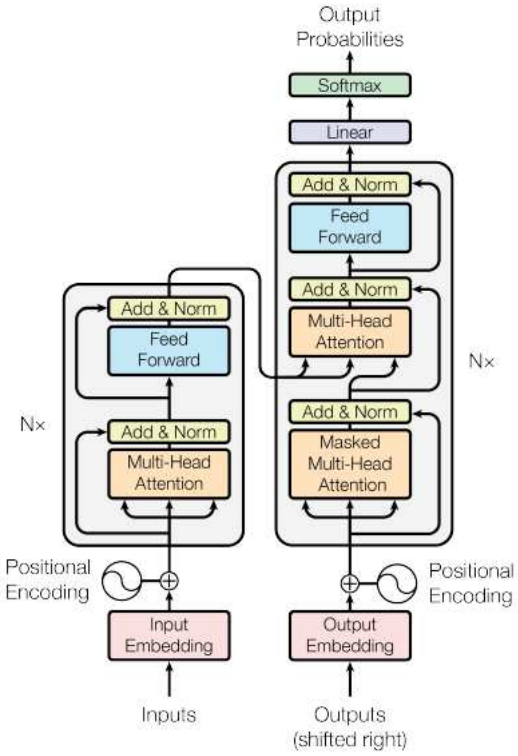


Figure 4.1: The Transformer model architecture [1]

LM from the literature to understand the necessity of a different approach to handling longer input sequences. Various approaches have been proposed to handle long input sequences for transformers. Text truncation is the most used method to encode texts that exceed the model input size. It consists of splitting input sequences into segments of fixed size and overlapping consecutive segments to reconstitute the context [134, 135].

However, those methods are interpreted as segment representations other than sequence representations. For biomedical, segment representation could harm the model performance where, for a given task, important information such as a disease, a drug, or a procedure could be located in one of the truncated segments. This technique can not ensure the capture of the long-range dependencies between segments.

Moreover, authors have reported this to be a direct consequence of the attention mechanism [121] despite its efficiency. The drawback of models with a full-attention has brought authors to think about alternatives that can reduce that exponential growth to a more reasonable dimension to allow encoders to process longer sequences that can fit in the current computational resources, as demonstrated by authors of BigBird [122] and Longformer [123]. For this work, We leveraged the architecture and the initial weights of BigBird, a transformer model for longer sequences [122]. For the sake of brevity, we refer readers to the original paper for more details. In the biomedical area, Li Y. et al. [136] provided two models based on BigBird [122] and LongFormer [123] architectures. However, this research did not consider the paramount importance of a tailored tokenizer for a specific domain using the default vocabulary from the source model.

Table 4.2: Backbone of large biomedical language models in comparison with our model

Model	Par.	Corpus	Tokenizer	Voc.	Hid. size	Position embed.
BioBERT$_{Large}$ [116]	364M	Wiki + Books + PubMed	WordPiece	59k	1024	512
BioLinkBERT$_{Large}$ [125]	334M	PubMed	WordPiece	29k	1024	512
BioMegatron$_{345}$ [119]	345M	PubMed + PMC	SentencePiece	29k&30k	1024	512
BioMegatron$_{800}$ [119]	800M	PubMed + PMC	SentencePiece	30k&59k	1280	512
BioMegatron$_{1.2}$ [119]	1.2B	PubMed + PMC	SentencePiece	30k&59k	2048	512
BioM-ELECTRA$_{Large}$ [133]	340M	PubMed	WordPiece	29k	1024	512
BioM-ALBERT$_{xxLarge}$ [133]	223M	PubMed	SentencePiece	30k	4096	128
PubMedBERT$_{Large}$ [120]	340M	PubMed	WordPiece	29k	1024	512
Our model	127M	BigBird+data in tab4.1	SentencePiece	50K	768	4096

4.4.4 Sparse Attention

Sparse Attention (SA) in transformers is a modification to the standard self-attention mechanism that reduces the computational complexity of the attention calculation by only considering a subset of the tokens in the input sequence.

Let Q, K, and V be the query, key, and value matrices, respectively, with dimensions $d_q \times n_q$, $d_k \times n_k$, and $d_v \times n_v$, where d_q, d_k, and d_v are the dimensions of the query, key, and value vectors, and n_q, n_k, and n_v are the number of tokens in the input sequence.

In the standard self-attention mechanism, the attention weights A are computed as follows:

$$A = \text{softmax}\left(\frac{Q^T K}{\sqrt{d_k}}\right) \in \mathbb{R}^{n_q \times n_k} \tag{4.4}$$

However, in sparse attention, the attention weights are computed based on a subset of the key vectors, which are selected using a predefined pattern or a learned attention mask. Let $M \in 0, 1^{n_q \times n_k}$ be the attention mask, where $M_{ij} = 1$ if the jth key vector can attend to the ith query vector, and $M_{ij} = 0$ otherwise. Then, the attention weights A are computed as follows:

$$A = \text{softmax}\left(\frac{Q^T K M}{\sqrt{d_k}}\right) \in \mathbb{R}^{n_q \times n_k} \tag{4.5}$$

By restricting the attention operation to a subset of the input tokens, sparse attention reduces the computational cost of the self-attention mechanism while still allowing the model to capture important dependencies between the tokens.

Sparse attention has been shown to be effective in various NLP tasks, including language modeling, machine translation, and text classification, particularly when the input sequence is long or the model has limited computational resources.

4.4.5 In-domain Optimization of LMs

There are several ways to optimize in-domain language models:

1. **Increase the amount of in-domain data**: One of the most effective ways to optimize in-domain language models is to train them on more data specific to the domain of interest. This can be achieved through data augmentation techniques such as paraphrasing, back-translation, and domain-specific dictionaries [137].

2. **Fine-tune pre-trained language models**: Fine-tuning pre-trained language models such as BERT and GPT-2 on in-domain data can significantly improve their performance on domain-specific tasks. Fine-tuning involves retraining the model on a small amount of in-domain data, which allows it to adapt its parameters to the specific characteristics of the domain [116].

3. **Use domain-specific embeddings**: Another approach is to use domain-specific embeddings instead of generic word embeddings. Domain-specific embeddings are trained on in-domain data and capture domain-specific semantics and concepts that are not present in generic embeddings [137].

4. **Incorporate domain-specific knowledge**: Incorporating domain-specific knowledge into the model architecture can also improve its performance on domain-specific tasks. For example, incorporating domain-specific ontologies or taxonomies can help the model better understand the relationships between domain-specific concepts.

Our approach used both strategies through a combination of data selection, model fine-tuning, and in-domain tokenizer and incorporated domain-specific knowledge using raw EHR data.

Optimizing language models for these domains can have significant practical implications, such as improving accuracy and efficiency in clinical decision-making, information extraction, and knowledge discovery from medical texts.

Therefore, this study aims to explore different approaches for optimizing language models for biomedical and clinical text, with a focus on both training domain-specific models and adapting pre-trained models. By reviewing the current state of the art in this area, this chapter will provide insights into the challenges, opportunities, and future directions for developing more effective language models for biomedical and clinical text.

4.4.6 Transfert Learning: Biomedical Language Models

Transformer-based models like BERT are the most used for various domains, and the biomedical is no exception. After its breakthrough in 2018 on different tasks with its simple yet efficient architecture has made its reputation. This pre-trained model has inspired researchers, considering that training from scratch can be expensive. Moreover, continuous training can be enriched with more knowledge, such as clinical, through a transfer learning mechanism. Models such as BioBERT [116] and clinicalBERT [138] used the knowledge acquired by the original BERT and fine-tuned it to clinical and biomedical documents to improve their performance on biomedical-related tasks. Despite that these models set the state-of-the-art in this domain, they inherited the strengths and weaknesses of their original model. As demonstrated in the RoBERTa paper [115], the BERT model was undertrained, and its hyperparameters were not optimized. Most importantly, we consider that transfer learning should be accompanied by setting optimum parameters for the data and the downstream tasks. Recent models that set state-of-the-art on the Biomedical Language Understanding Evaluation(BLUE) benchmark [139] used the same vocabulary inherited from their original models. However, the SciBERT paper [137] demonstrated the importance of building a dedicated tokenizer with an in-domain vocabulary for scientific documents.

4.4.7 In-domain Tokenization

General language representation models similar to BERT have been trained on a large variety of English documents such as Wikipedia and Book corpus. This pre-training process gives them a high ability to contextualize individual words and tokens because of the attention mechanism in their architecture. However, the same models and tokenizers don't perform well on domain-specific texts such as biomedical or scientific documents [137]. Researchers understood that subsequent efforts should be focused on using additional in-domain text to provide a more accurate representation of the related contexts. BioBERT [116] uses 4.5B and 13.5B words from respectively

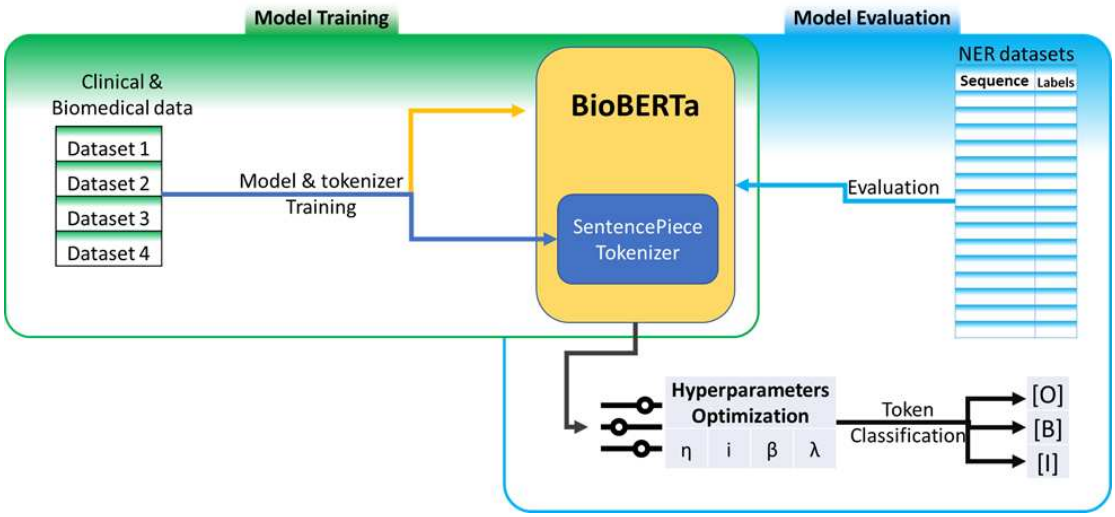


Figure 4.2: BioBERTa Training overview

PubMed and PMC to pre-train their model. In addition, ClinicalBERT [138] utilized real-world biomedical data from MIMIC to capture unusual grammatical structures that most models are not paying enough attention to. Although some authors [140] claim that a continual pre-training of the general domain to a domain-specific LM doesn't necessarily help the resulting model to perform better, we believe that some important steps of an LM development, such as a proper tokenizer can challenge that assumption.

4.5 Experiment, Methods, and Materials

The essence of our experiment relies on introducing an adapted tokenizer and a fine-tuned model for biomedical and clinical data. We extended our approach and provided a sparse attention-based model to represent long biomedical sequences better. In this section, we describe our tokenization process, the model training, and the experimental setup that leads to **BioBERTa** and later to **Medical BigBERTa**, as illustrated in Fig 4.2 and Fig 4.3.

4.5.1 Tokenization Process

Tokenization is the process of splitting a sequence of sentences into words or subwords in order to identify entities by looking them up in a vocabulary table. This transformation is crucial for computers to understand words in a numeric environment by replacing tokens with their respective IDs from the table. Comparable to a new language, clinical texts such as EHR notes and reports require dedicated dictionaries to handle inner-domain jargon and conventional annotations to grasp their contextual representations.

With regard to mitigating rare words with subwords, Byte Pair Encoder (BPE) was introduced in 2015 [141, 142] and has since been used in LMs such as GPT2 [131] and RoBERTa [115]. BPE relies on a 2-step tokenization process. A pre-tokenization simply splits the training data into words and associates them with their occurrence frequency. The second step consists of segmentation of words to a character level and

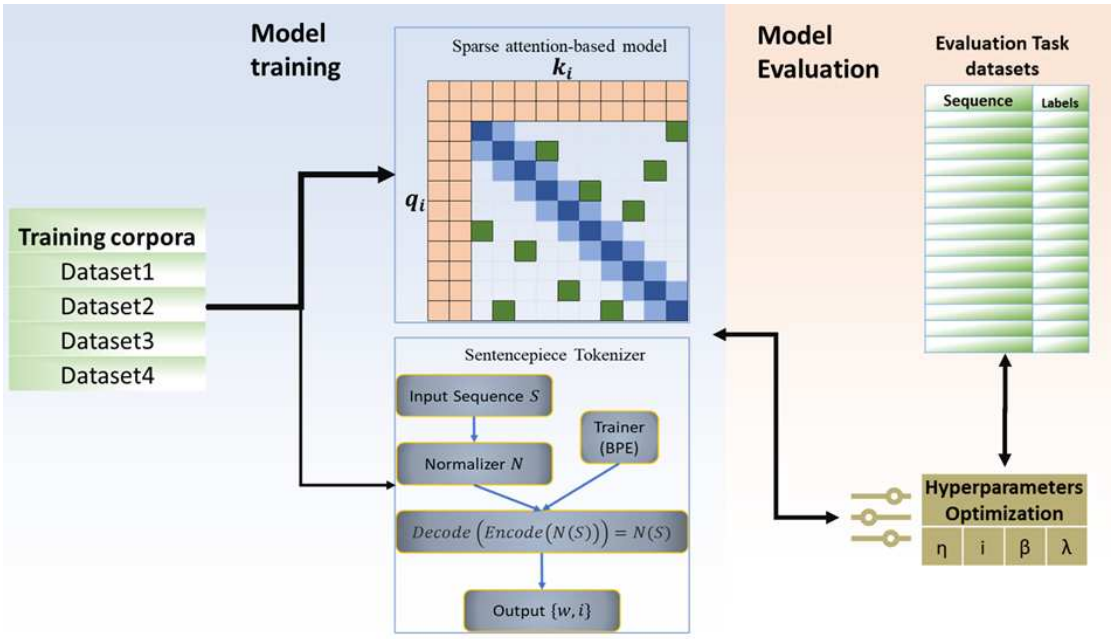


Figure 4.3: Biomedical BigBERTa

learning merging rules to constitute the most frequent [1]character associations limited by the vocabulary size defined as a hyperparameter. Training a BPE for domain-specific LM has two major advantages. First, a new tokenizer for any specific domain is necessary to build the domain-related vocabulary. In addition, BPE can balance the vocabulary size to the frequency of words,sub-words, and characters on a gradual scale since common entities will be merged earlier, resulting in a dictionary that provides a more accurate representation of the domain-related context. In pursuance of reducing the vocabulary size, we utilized bytes as the base vocabulary, a technique called Byte-level BPE proposed in GPT2 [131]. Byte-level BPE tokenization better suits the need to represent various vocabulary of the raw medical texts without using the $<UNK>$ token, and where abbreviations need to be interpreted within their contexts in each note.

To train our tokenizer, we combined approximately 43.6M tokens of raw EHR notes from MIMIC III [126] and about 215M tokens from the general biomedical text, totalizing over 5GB of text data as shown in Table 4.1. By treating sequences as a series of Unicode characters, this tokenization supports multiple subword algorithms, such as BPE [142], unigram language models [143] and others. BPE stands between character and word-level language modeling by taking advantage of both word-level inputs for frequent sequences of symbols and character-level inputs for rare symbol sequences. This attribute gives the resulting representation a tremendous benefit over any character-based tokenization for a domain-specific document by excluding out-of-vocabulary tokens, which restrict the space of a contextual input representation. Our BPE tokenizer can encode any text within the UTF-8 characters, which requires only 256 uni-characters in its base vocabulary.

[1]In contrast to Wordpiece algorithm, which is based on likelihood instead of frequency

4.5.2 Experimental Datasets

Inspired by the training approach of BlueBERT [139], we trained our model by employing myriad biomedical and clinical datasets. We used both real-world medical text notes from the publicly available dataset, MIMIC III [126] and general English biomedical text corpus from three different datasets obtained from Kaggle [2]. Language models are sensitive to the data distribution, we integrated these corpora in order to balance and generalize the tokenization with the off-domain word frequency. A brief description is given below, and we refer readers to Table 4.1 for more details.

- **MIMIC III** (Medical Information Mart for Intensive Care) is a widely used dataset for medical purposes. it aggregates deidentified medical data from more than 40.000 patients who were admitted to the critical care units of the Beth Israel Deaconess Medical Center from 2001 to 2012. We extracted over 1 million sequences with an average of 1258 words per sequence, containing procedure notes, reports from different services, and discharge summaries. To standardize the data, we applied the preprocessing methods described as a thorough cleaning in our previous work [144].

- **PubMed title abstract baseline 2019** is a public dataset, aggregating titles and abstracts from the PUBMED database for articles published in 2019. However, we consider this dataset as general English texts with biomedical terminologies.

- **The medical transcription and the Medical Text for Text Classification** contains sample medical transcriptions and reports from various medical specialties describing a patient. although they are purely medical-related and not relatively large, both datasets provide text with grammatically and syntactically rich sequences.

4.5.3 Biomedical Language Model 1: BioBERTa

As for most language models, we utilize a combination of pre-training and supervised fine-tuning. As suggested in the BERT paper [114], we used our cased tokenizer to evaluate our model on a NER task. We created a conda environment on a single GPU RTX3090 with 24GB memory. As for the training, we followed a similar configuration as RoBERTa-base [115] for the optimization and hyperparameter arguments. Although some authors recommend freezing the embedding while performing continuous training, we needed to train all the layers since we used our own tokenizer. In order to optimize our computing power, we concatenated and then chunked all the training sequences in samples of the model input length. Our training took over 446 hours with a maximum input length of 512 and a batch size of 16. With a perplexity score of 3.35, we didn't perform any hyperparameter search at this stage since we believed it to be either a task-dependant or data-dependant optimization.

4.5.4 Biomedical Language Model 2: Medical BigBERTa

Medical BigBERTa is a transformer trained for autoregressive language modeling, trained to predict only a hidden token *[mask]* given a context on its left. This model

[2]The three datasets are available in 1, 2, 3

shares the same architecture with BigBird [122], which was trained over RoBERTa [115] weights. We implemented our model's architecture to fit with BigBird in order to evaluate our contribution in contrast with the existing work and simplify its sharing with the community.

BigBird

In disparity to predominant transformer-based models that use a fully quadratic self-attention mechanism, BigBird and our model are designed with a sparse attention scheme based on a few inner products of selected tokens. This innovative attention mechanism is an efficient approximator of the traditional full attention to allow design models that can operate longer sequences. Consequently, it does not seek to be better than the latter. It consists of three types of blocks of tokens:

- Global tokens g are composed of a window of tokens that attend to all other sequence tokens.
- Local tokens l where each token attends to its neighbors and itself.
- Random tokens r where all tokens attend to a set of random tokens in the sequence.

Fig. 4.4 illustrates the sparse attention mechanism used in the BigBird model [122]

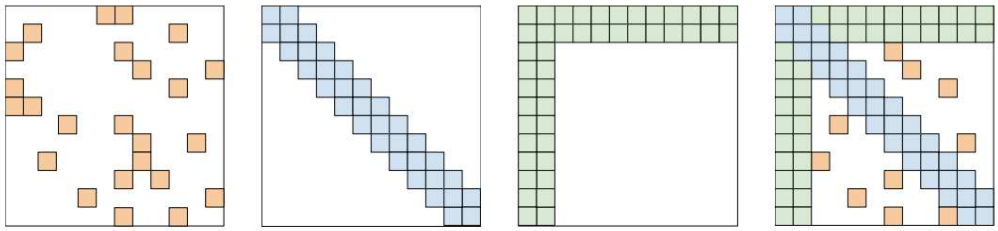


Figure 4.4: An illustration of the sparse attention mechanism

Our Model Configuration

Given the computational limitation of our environment (Single GPU, Nvidia RTX 3090, 24GB), we couldn't follow the original setup of BigBird. To optimize the training process, our model uses the internal transformer construction (ITC) configuration with a block size b of 64 and g = 2 x b, l = 3 x b and r = 3 x b. We pretrained it with only the prediction objective of a masked token for 7.8M steps and a learning rate of 5e-5 with a batch size of 4.

The sparse attention architecture is kept intact only if the input length is more than 1024 tokens. Differently, the model automatically switches to the quadratic full attention. One of the techniques of training such a model to its full potential is to either pad the inputs to the maximum length or bucketing by subgrouping samples according to their lengths [145]. However, these approaches will only use resources without adding any values. Thus, to accelerate our training, we concatenated all the training sequences and chunked them into samples of length $max\_size = 4096$. As in RoBERTa,

our model was trained with dynamic masking that changes each epoch of the training. The masking procedures were kept the same as BigBird. This choice has the advantage of preventing the model from overfitting within the new domain while we ensure a constant gradient descent. We trained two separate models with two different input lengths of 2048 and 4096 separately. However, we observed that the shorter model was not long enough to highlight a significant contribution compared to the existing long version models such as BioMegatron [119] or BioBERT large.

4.5.5 SentencePiece Tokenizer

To train our tokenizer, we combined approximately 43.6M tokens of raw EHR notes from MIMIC III [126] and about 215M tokens from the general biomedical text, totalizing over 5GB of text data as shown in Table 4.1. By treating sequences as a series of Unicode characters, this tokenization supports multiple subword algorithms, such as byte-pair-encoding (BPE) [142], unigram language models [143] and others. BPE stands between character and word-level language modeling by taking advantage of both word-level inputs for frequent sequences of symbols and character-level inputs for rare symbol sequences.This attribute gives the resulting representation a tremendous benefit over any character-based tokenization for a domain-specific document by excluding out-of-vocabulary tokens, which restrict the space of a contextual input representation. Below is the synthesized algorithm of how we created our tokenizer using the BPE method:

Algorithm 2 BPE-based Tokenizer

procedure BPETOKENIZER(S, V, k)
    Initialize the vocabulary V with all unique characters in the input text S
    for $i = 1$ to k **do**
        Compute the pair frequency of all pairs of characters in V
        Select the most frequent pair (a, b)
        Add the new token ab to V
        Replace all occurrences of (a, b) in S with the new token ab
    end for
    return BPE tokenizer with the vocabulary V
end procedure
procedure TOKENIZE($tokenizer, S$)
    Initialize an empty list T
    for each substring s in S **do**
        Append the tokens obtained by applying the tokenizer to s to T
    end for
    return List of tokens T
end procedure

In this algorithm, the first procedure, 'BPETokenizer', creates a BPE-based tokenizer using the input parameters: the input text S, the initial vocabulary V, and the number of BPE iterations k. The algorithm first initializes the vocabulary with all unique characters in the input text. It then enters a loop, where it repeatedly computes the pair frequency of all pairs of characters in the vocabulary, selects the most frequent pair, adds the new token to the vocabulary, and replaces all occurrences of the

pair in the input text with the new token. The loop runs for the specified number of BPE iterations. At the end of the loop, the algorithm returns a BPE tokenizer with the final vocabulary V. The second procedure, 'Tokenize', takes the BPE tokenizer created by 'BPETokenizer' and the input text S as parameters. It initializes an empty list T and iterates over each substring s in the input text. For each substring, it applies the BPE tokenizer to s and appends the resulting tokens to T. At the end of the loop, the algorithm returns the list of tokens T, which represents the BPE tokenization of the input text.

In the end, our BPE tokenizer can encode any text within the *UTF-8* characters, which requires only 256 uni-characters in its base vocabulary.

4.5.6 Evaluation Tasks

Biomedical language models have been improved on various ranges of downstream tasks. Like in general-domain, comprehensive benchmarks such as GLUE [146] have been used to evaluate the evolution of language models on different tasks, giving a clear orientation to researchers on where they should focus on expanding the NLP boundaries. Our initial model BioBERTa was evaluated only on the Named Entity Recognition(NER) to help us to determine the contribution of an extended optimization. Our ultimate goal was to evaluate the performance of our long model on various tasks in comparison to other state-of-the-art biomedical domain-oriented LM. Biomedical Language Understanding Evaluation (BLUE) [147] was the first publicly available benchmarking for biomedical LM. However, we decided to use a more recent benchmark with more coverage on datasets used in recent work for this evaluation. The Biomedical Language Understanding and Reasoning Benchmark(BLURB) [140] is a comprehensive collection of thirteen corpora covering six different tasks, while BLUE covered five tasks and used ten datasets. Furthermore, we would have liked to include the real-world clinical datasets evaluation provided with the BLUE benchmark. However, the MedNLI corpus [147], for instance, was not available at the indicated location. In the following sections, we discuss the tasks we evaluated our model on as well as the datasets utilized for each task. Table 4.3 provides a brief description of each dataset as well as its evaluation metrics. For the sake of brevity, we refer readers to the original BLURB benchmark paper [140] for more details about the data preparation and description.

Named Entity Recognition(NER)

This task consists of predicting mentioned spans of the input document. These entities range from chemicals, diseases, drugs, proteins, and others. The performance is measured by comparing the set of predicted tags and spans with a set of ground truth labels to the entity level. This task is regarded as a token classification problem involving the measurement of the model and its tokenization. We evaluate the model by calculating the average f1-score for each class using the precision and recall on the entity level.

Relation Extraction(RE)

The relation extraction task aims to predict the relation between a pair of entities mentioned as artifacts in the input document. Biomedical mentions can be classified as drugs, chemicals, proteins, diseases, or genes. This task highlights the ability of a

Table 4.3: Summary of our considered datasets for model evaluation

Dataset	Domain	task	Evaluation metrics
BC5CDR-Chem [148]	Biomedical	NER	F1 entity-level
BC5CDR-disease [148]			F1 entity-level
NCBI-disease [149]			F1 entity-level
BC2GM [150]			F1 entity-level
JNLPBA [151]			F1 entity-level
BC5CDR [148]			F1 entity-level
Linnaeus [152]			F1 entity-level
BC4CHEMD			F1 entity-level
Species-800 [153]			F1 entity-level
ChemProt [154]	Biomedical	RE	Micro F1
DDI [155]			Micro F1
BIOSSES [156]	Biomedical	SS	Pearson corr.
BioASQ 4b [157]	Biomedical	QA	Accuracy
BioASQ 5b [157]			
PubMedQA [158]			
SQUAD V1 [159]	General	QA	Accuracy
SQUAD V2			Accuracy

model to extract structured information from unstructured inputs. The model requires a classification layer as the task is regarded as a sequence classification problem into classes of relations. The evaluation measures the micro-averaged F1 score of the predicted classes.

Sentence Similarity (SS)

This task measures and predicts how similar two sequences are. Capturing the semantic information and calculating how close the sequences are, this task gauges the ability of an LM to retrieve and cluster biomedical information. We evaluate this similarity by calculating the Pearson correlation coefficients between pairs of inputs.

Evidence-Based Medicine(EBM)PICO

The EBM PICO(Patient, Intervention, Comparison, and Outcome) dataset contains clinical questions formulated in the PICO format and corresponding biomedical articles or evidence-based documents that provide answers or information relevant to the questions. It consists of extracting information from clinical text, answering clinical questions, or supporting evidence-based decision-making in healthcare settings. We evaluated using a micro F1 score on words in the model's output.

Question Answering(QA)

The question Answering task consists of extracting an answer from a given document. The model takes a context text and a question and returns an answer. The answers can be a reference text from the context input (extractive QA), a factoid, or a

label(yes, no, maybe). Although this task is evaluated on F1-score or Exact Match, we used accuracy to compare our results with the baselines on the BLURB benchmark.

4.6 Hyperparameter Optimization

Hyperparameters are parameters that cannot be learned directly from the training data but rather must be set before training [160]. For performance optimization, the practice of using recommended hyperparameters does not guarantee optimum results with the subsequent model. Moreover, we understand that while predefined learning parameters could lead to good performance, optimizing hyperparameters should yield better results [160, 161].

4.6.1 Optimization Problem

An optimization problem is a mathematical problem that involves finding the best solution from a set of possible solutions that satisfies a set of constraints [162,163]. The best solution is typically the one that maximizes or minimizes an objective function. For our case, it involves finding the optimal set of hyperparameters that result in the best performance of a given model on a specific task. While this optimization problem can be solved using a variety of methods, Some common methods include:

- **Bayesian Optimization**: Bayesian optimization aims to find the maximum of an unknown function $f(x)$ with a minimum number of function evaluations [161]. It models the unknown function as a Gaussian process, which allows it to trade off exploration (sampling in unexplored regions) and exploitation (sampling in regions where the function is expected to be high). The acquisition function is defined as a trade-off between the mean $\mu(x)$ and variance $\sigma^2(x)$ of the Gaussian process at a candidate point x:

$$x_{t+1} = \text{argmax}_{x \in \mathcal{X}}; \alpha(x; D_t) \tag{4.6}$$

  where $D_t = (x_i, y_i)_{i=1}^t$ is the history of function evaluations up to time t, and $\alpha(x; D_t)$ is an acquisition function that balances exploration and exploitation. The most commonly used acquisition functions are the upper confidence bound (UCB) and the expected improvement (EI):

$$\text{UCB}(x; D_t) = \mu(x) + \sqrt{\beta_t}\sigma(x) \tag{4.7}$$

$$\text{EI}(x; D_t) = \begin{cases} (f(x) - f(x_{\text{best}}))\Phi(Z) + \sigma(x)\phi(Z), & \text{if } \sigma(x) > 0, \\ 0, & \text{if } \sigma(x) = 0 \end{cases} \tag{4.8}$$

  where β_t controls the trade-off between exploration and exploitation, Z is a standard normal random variable, Φ and ϕ are the cumulative distribution function and probability density function of the standard normal distribution, and x_{best} is the current best point found.

If $\sigma(x) > 0$, the expected improvement consists of two terms: the first term $(f(x) - f(x_{\text{best}}))\Phi(Z)$ represents the exploitation term, encouraging the search to focus on areas where the surrogate model predicts improvements over the current best observation. The second term $\sigma(x)\phi(Z)$ represents the exploration term, encouraging the search to explore regions with high uncertainty.

If $\sigma(x) = 0$, it means that the surrogate model's predicted standard deviation (uncertainty) at point x is zero. In other words, the surrogate model is certain about the objective function value at that particular point because it has already been observed in the historical data D_t.

In this case, if $\sigma(x) = 0$, the expected improvement function $\text{EI}(x; D_t)$ is defined as:

$$\text{EI}(x; D_t) = 0 \tag{4.9}$$

When the uncertainty is zero, there is no need for exploration because the objective function value at point x is already known. Therefore, the expected improvement is set to zero in order to avoid unnecessary exploration of already observed points. This ensures that the algorithm does not waste computational resources by evaluating the objective function at points that have already been sampled. This Gaussian process makes this algorithm more efficient and powerful for hyperparameter search.

- **Random Search**: Random search [160, 164] is a simple and popular method for hyperparameter optimization . It involves randomly sampling hyperparameters from a specified range and evaluating them to find the best set of hyperparameters:

$$\theta_{\text{best}} = \arg\max_{\theta \in \Theta} f(\theta) \tag{4.10}$$

where Θ is the set of all possible hyperparameters and $f(\theta)$ is the objective function evaluated at hyperparameters θ.

- **Grid Search**: Grid search [164] involves searching over a pre-defined grid of hyperparameters and evaluating them to find the best set of hyperparameters:

$$\theta_{\text{best}} = \arg\max_{\theta \in \Theta} f(\theta) \tag{4.11}$$

where Θ is the grid of hyperparameters and $f(\theta)$ is the objective function evaluated at hyperparameters θ.

- **Tree-structured Parzen Estimator (TPE)**: TPE is another Bayesian optimization method based on dividing the hyperparameter space into two regions: good and bad. It uses kernel density estimation to model the probability density function of each region. It finds the hyperparameters that maximize the ratio of the probabilities of being in the good region over the bad region.

  The objective function is modeled as a conditional distribution:

$$p(y|x) = \begin{cases} l(x), & y = f(x) \\ g(x), & y \neq f(x) \end{cases} \tag{4.12}$$

Where $f(x)$ is the black-box function that is being optimized, and $l(x)$ and $g(x)$ are probability density functions of the good and bad regions, respectively.

Although all these methods have been used in machine learning optimization, we could not evaluate all of them. Our choice of an optimization tool was guided by the following considerations:

- **Scalability**: It should be scalable and can handle optimization tasks with a large number of hyperparameters.
- **Flexibility**: It should support various hyperparameters, including continuous, discrete, and categorical parameters.
- **Interoperability**: It should integrates easily with our machine learning frameworks(PyTorch).
- **Efficient**: It employs state-of-the-art algorithms for optimization, such as TPE, which have been shown to be highly efficient in finding optimal hyperparameters.

4.6.2 Optuna

The hyperparameter exploration was conducted using Optuna framework [165], an open-source library written in Python that uses state-of-the-art optimization algorithms to automatically search for the best hyperparameters of a machine learning model. Assume that we have a hyperparameter search space H and an objective function $f(h)$ that maps a hyperparameter configuration $h \in H$ to a scalar value representing the performance of the corresponding model. The goal is to find the hyperparameter configuration h^* that maximizes or minimizes the objective function:

$$h^* = \arg\max / \min f(h) \tag{4.13}$$

To achieve this, Optuna builds a probabilistic model of the objective function using a Gaussian Process (GP) [166] model, which represents the objective function as a probability distribution over the search space of hyperparameters. Specifically, the GP model estimates the mean and variance of the objective function at each point in the search space based on the evaluations of the objective function at previous points.

Given the probabilistic model of the objective function, Optuna uses an acquisition function to suggest the next hyperparameter configuration to evaluate. The acquisition function balances the exploration of new regions of the search space (where the uncertainty is high) with the exploitation of promising regions (where the objective function is expected to be high). The expected improvement (EI) [161, 165] criterion is commonly used as the acquisition function in Optuna, which is defined as:

$$EI(h) = E[\max(f(h) - f(h^*), 0)] \tag{4.14}$$

where h^* is the current best hyperparameter configuration found so far, and $E[\cdot]$ denotes the expected value. The hyperparameters are then sampled from the probabilistic model

using a sampling algorithm, such as the Tree-structured Parzen Estimator (TPE) or the CMA-ES algorithm, and evaluated using the objective function. The new evaluation is then added to the set of previous evaluations, and the probabilistic model is updated using the new information. This algorithm can be summarized by the following steps:

Algorithm 3 Optimizing Multiple Hyperparameters with TPE

Require: Hyperparameter search space Θ, objective function $f(\theta)$
Ensure: Optimal set of hyperparameters θ^*
1: Define the search space Θ
2: Define the objective function $f(\theta)$
3: Choose the TPE algorithm
4: Initialize the search process by creating a study object and specifying the search algorithm, search space, and objective function
5: **while** search not complete **do**
6: Optimize the hyperparameters by calling the ***optimize*** method of the study object
7: Evaluate the objective function for each suggested set of hyperparameters
8: Update the search space based on the results of previous trials
9: $i \leftarrow i + 1$
10: Compute the probability of improvement using a TPE
11: Sample new hyperparameters from the search space based on the probability of improvement
12: **end while**
13: **return** the set of hyperparameters that produced the best objective value, θ^*

Optuna uses a while loop to repeat the optimization process until a stopping criterion is met. During each iteration of the loop, the algorithm suggests new sets of hyperparameters using TPE, evaluates the objective function for each set of hyperparameters, updates the search space, and samples new hyperparameters based on the probability of improvement. Finally, the algorithm returns the set of hyperparameters that produced the best objective value

4.7 Results

A new Tokenizer and an LM fine-tuned with biomedical and clinical data have a transient objective of evaluating our approach using a single task. However, our ultimate goal is to demonstrate the utility of optimizing biomedical LMs by combining an implementation of sparse attention, a dedicated tokenizer, and hyperparameter finetuning. Moreover, we aim to evaluate our proposed model on a variety of NLP tasks to compare its performance with existing models. We discuss in the following subsections the evaluation of our model and its tokenizer. We demonstrate the model performance on different tasks by comparing its results with the BLURB [140] leaderboard as of October 2022, as well as the SoTA from the literature.

4.7.1 An Adapted Tokenizer

The main goal of a tokenizer is to split a sequence of texts into units with a semantic meaning. With this objective, keeping words unsplit could be ideal. However, a significant drawback of a word-level tokenizer is the huge vocabulary size created, especially from documents with such a large amount of lexical variation or lexical diversity due to typos or the medical domain itself. Our goal for a new tokenizer was to produce a vocabulary that would split as few words as possible to preserve a word's original context and semantics in a sequence. We built a 50358 vocabulary-size tokenizer. We borrow the fertility rate (FR) concept defined in statistical machine translation [167] as the ratio of the lengths of sequences generated from a translation. In our case, FR measures the average number of tokens produced from a word. It reveals how your tokenizer vocabulary is adapted to the documents. Thus, the ideal FR of 1 indicates that each word of the input text is included in the tokenizer vocabulary. Moreover, a tokenizer with the lowest FR has the advantage of generating the shortest input sequence, whereas it preserves a consistent word representation in any context.

Fig. 4.5a shows the fertility of each model on various datasets. Fig. 4.5b ranks the models while Fig. 4.5c reports the mean average of all the models showing that ours has the lowest FR.

Figure 4.5 provides the results of FR and some baseline models. Figure4.5a shows the results of the models over different datasets, while figure4.5b shows the rank of the models on each dataset. Our model demonstrates the lowest FR on three datasets out of 6, with the lowest average FR of 1.3893 in figure 4.5c.

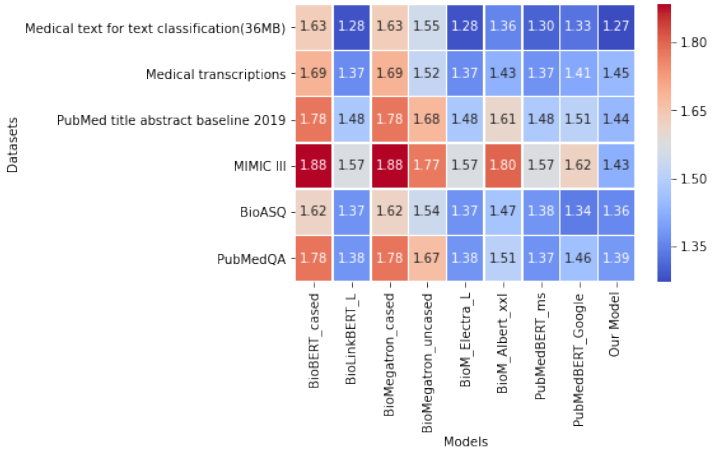
This ability to recognize entities demonstrates that our tokenizer is less susceptible to breaking the semantic meaning of medical terminologies where an over-segmentation is observed on the counterpart tokenizers. As demonstrated and stated by Rust P. et al. [124], "both the data size and the tokenizer are among the main driving forces of downstream task performance."

4.7.2 Model 1: BioBERTa

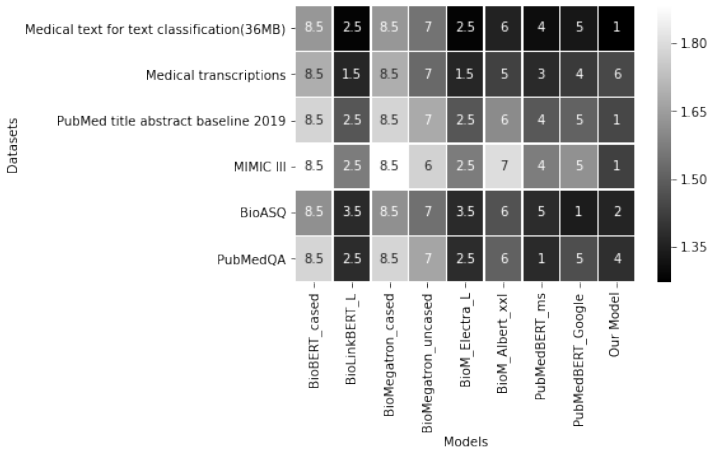
As suggested in the BERT paper [114], we used our cased tokenizer to evaluate our model on a NER task. We created a conda environment on a single GPU RTX3090 with 24GB memory. As for the training, we followed a similar configuration as RoBERTa-base [115] for the optimization and hyperparameter arguments. Although some authors recommend freezing the embedding while performing continuous training, we needed to train all the layers since we used our own tokenizer. In order to optimize our computing power, we concatenated and then chunked all the training sequences in samples of the model input length. Our training took over 446 hours with a maximum input length of 512 and a batch size of 16. With a perplexity score of 3.35, we didn't perform any hyperparameter search at this stage since we believed it to be either a task-dependant or data-dependant optimization.

4.7.3 Model 2: Biomedical BigBERTa

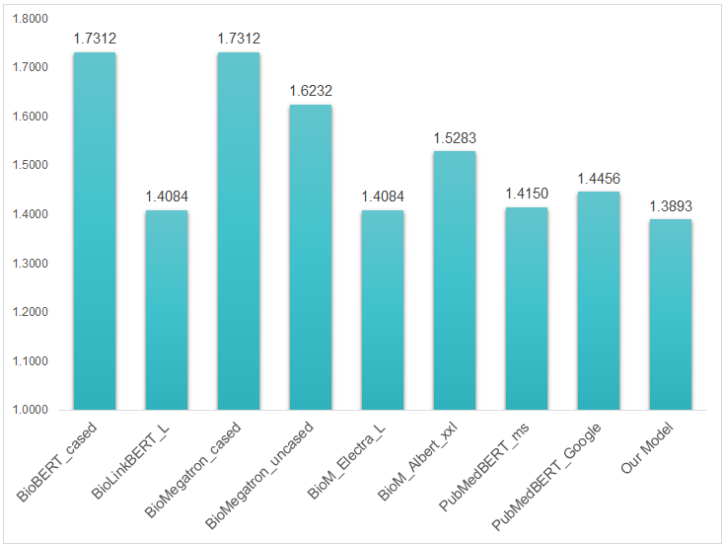
We introduced parse attention for biomedical documents as a solution to mitigate long sequence encoding with a relative tolerance on the model performance. To evaluate that effectiveness, we compare our model with a large transformer-based model.



(a) The fertility rates

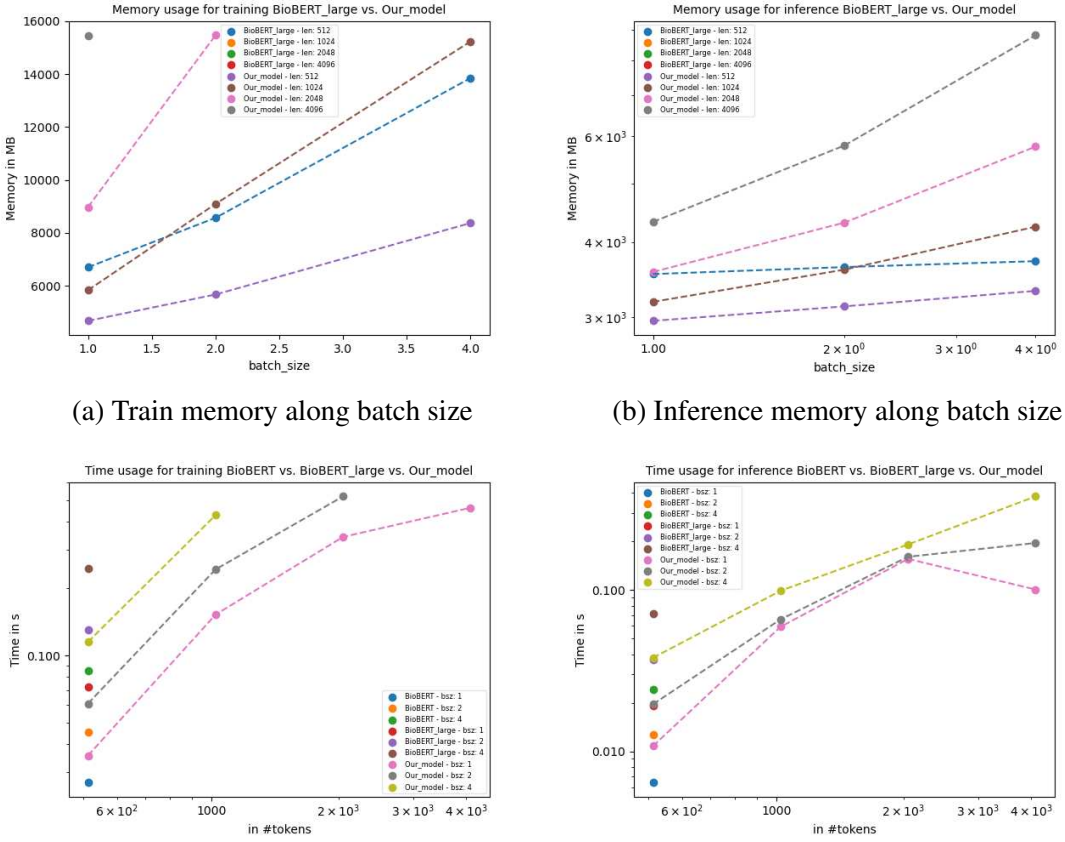


(b) The fertility ranks



(c) Mean of the fertility rates of each model

Figure 4.5: Comparison of the fertility rates.



(a) Train memory along batch size

(b) Inference memory along batch size

(c) Train time along input size

(d) Inference time along input size

Figure 4.6: Train and inference benchmarks.

Fig. 4.6 demonstrate the effectiveness of the sparse attention in reducing the time and memory cost while we still can encode four times the input size of a base model such as BioBERT and BioBERT large. Our proposed approach proves that sparse attention is an efficient pruning technique to improve the handling of longer clinical documents.

Fig. 4.6a and 4.6b show the memory cost over the increase of the batch size as long as the change of input lengths. Fig. 4.6c and 4.6d show a comparison of our model on the training and inference time along the input size while we change the batch size(bsz).

Fig. 4.6a suggests that our model requires relatively less memory in comparison to BioBERT_large on the same input size. That affirmation is supported by Fig. 4.6b showing the average memory required for inference. Fig. 4.6c, and 4.6d both show that Medical BigBERTa trains and infers faster and on larger input sequences.

4.7.4 NER

Table 4.4 reports the named entity recognition results. The first column shows each dataset's baseline result from the BLURB leaderboard. The literature has reported several results obtained on similar data without being listed on the leaderboard; we reported the best results found within the literature. In our experiment for NER, we utilized two versions of the model. "BigBERTa$_{2048}$" and "BigBERTa$_{4096}$" can respectively encode 2048 and 4096 tokens. We find that the shorter version performed slightly better than the longer one. A tradeoff between the embedding length and accuracy was established

in favor of the embedding size for other tasks. That mild difference can be explained by the fact that both models use full attention for all the NER datasets because the input sequences are relatively short of triggering the use of sparse attention. Moreover, our model performs better on most of the datasets. This is partially due to the tokenizer that better bridges the sub-word representation and the token-level prediction by providing a less segmented vocabulary rich in biomedical terms.

Table 4.4: Evaluation results of our models on the named entity recognition task.

Dataset	BLURB	Lit.	BigBERTa$_{2048}$	BigBERTa$_{4096}$
BC5-chem	94.04	**94.88** [168]	92.37	92.3
BC5-disease	86.39	88.5 [119]	**89.08**	87.83
NCBI-disease	88.76	90.48 [168]	**92.48**	92.4
BC2GM	85.18	**88.75** [168]	88.44	88.37
JNLPBA	80.06	82.0 [169]	**85.72**	85.68
Linneaus	-	**89.81** [116]	84.90	82.72
Species-800	-	**82.59** [168]	77.68	76.25
BC4CHEMD	-	94.39 [168]	**94.47**	93.34

4.7.5 Relation Extraction, PICO, and Sentence Similarity

Table 4.5 reports the results of three tasks. On our token classification task with the Evidence-Based Medicine (EBM) corpus, spans of the inputs are annotated with P, I, and O(Participant, Intervention, and Outcome). Our model sets a new SOTA over 8% of the Pearson correlation score. This dataset contains a lot of medical terminologies, and our model takes advantage of its rich medical vocabulary learned from MIMIC to limit the over-segmentation of terms since the accuracy is evaluated on the word level. The proposed model outperformed existing models on 2 out of 3 relation extraction datasets. However, it didn't show higher performance on the sentence similarity task. We suspect that it is due to the dataset size, which is relatively too small for a larger model without using any extra training data as illustrated by counterpart models [125].

Table 4.5: Evaluation results from evidence-based medical information extraction (PICO), relation extraction(RE), and sentence similarity(SS) tasks

Dataset	Task	BLURB	Lit.	BigBERTa$_{4096}$
EBM PICO	PICO	74.19	74.19 [125]	**82.99**
ChemProt	RE	80.0	88.95 [170]	**89.87**
DDI	RE	83.35	83.35 [125]	**94.04**
GAD	RE	**84.90**	84.90 [125]	81.83
BIOSSES	SS	**94.49**	93.63 [125]	90.12

4.7.6 Q&A

On the question-answering task, we had two datasets from the BLURB benchmark. The BioASQ [140] evaluates LMs only on the yes/no questions(task7b), as well as Pub-

MedQA [140], which focuses only on the yes/maybe/no questions. Table 4.6 provides our results from the two corpora compared to the leaderboard and the literature SOTA. Our model performed poorly on the BioASQ 7b dataset due to insufficient training data(670 questions). As recommended by BLURB, other authors have used data from previous tasks (2747 training questions) to pre-train their models before stepping up to the 7b dataset. However, our model outperformed the SOTA on PubMedQA by a 5% F1 score. This performance was achieved by taking advantage of our large model. We incorporated the provided long answers into the context as a single input sequence.

Nonetheless, one of the best ways to evaluate a model on a language understanding task is its ability to generate answers on a factoid-type QA task. This reveals whether an LM understands or simply memorizes the input sequence [171]. To assess that ability, we additionally evaluate our model on BioASQ task4b and 5b used by Lee et al. [116]. The results are shown in table 4.7, where our model outperformed the results obtained with BioBERT. We reproduced the BioBERT results using the same batch size=8 as for our model.

Table 4.6: Comparison of our model and SOTA on Biomedical Q&A tasks

	BLURB	Lit.	Our Model
BioASQ(7b) [140]	**94.82**	94.8 [125]	89.28
PubMedQA [140]	72.18	72.2 [125]	**77.87**

Table 4.7: Results on biomedical and general question answering tasks

Models	Metrics	SQUAD V2	BioASQ 4b	BioASQ 5b
BERT	EM	**78.81**	80.52	83.91
	F1	86.70	80.85	83.27
BioBERT[3]	EM	-	**82.11** *	86.21
v1.1	F1	-	**83.35** *	88.04
BigBird	EM	76.20	79.89	84.50
ITC	F1	82.63	77.90	86.38
Our	EM	78.30	81.98	**87.11**
model	F1	**86.94**	82.13	**88.36**

4.8 Further Analysis

4.8.1 Ablation Studies

To assess the impact of our novel approach incorporating the new tokenizer and the sparse attention-based model, we conducted an ablation study following these steps. Initially, we pretrained the RoBERTa model using the identical dataset and subsequently fine-tuned it on multiple tasks utilizing our custom tokenizer. Additionally, we gauged

[3]* BioBERT results were obtained from our environment, using the same batch size as our model

the influence of the tokenizer on our model's performance by substituting it with the standard tokenizer from RoBERTa. Throughout the experimentation, we maintained consistent hyperparameter settings as presented in Table 4.9.

Table 4.8: Ablation studies: Comparison of our proposed approach with a combination of (1) BigBERTa and RoBERTa tokenizer, and (2) RoBERTa model and BigBERTa tokenizer.

Dataset	BigBERTa	RoBERTa ($_{base}$)	BigBERTa+ Tok.$_{RoBERTa}$	RoBERTa+ Tok.$_{BigBERTa}$
BC5-Chem	92.37	**93.81**	90.05	90.81
BC5-disease	**89.08**	88.04	78.13	69.01
NCBI-disease	**92.48**	88.44	81.13	76.72
BC2GM	**88.44**	86.47	82.47	78.06
JNLPBA	**85.72**	85.15	82.15	79.63
BioASQ	**89.28**	65.29	65.29	65.29
PubMedQA	**77.87**	73.8	57.4	58.2
EBM PICO	**82.99**	77.11	69.43	66.90
ChemProt	**89.87**	88.58	83.89	83.97
DDI	94.04	**95.99**	90.59	90.83
GAD	**81.83**	80.71	72.59	75.59

From the results in Table 4.8, we can observe that BigBERTa ($_{base}$) consistently outperforms RoBERTa across most datasets, demonstrating its effectiveness for biomedical natural language processing tasks. However, RoBERTa demonstrated high competitiveness due to its full attention and the input size of different datasets. Interestingly, BigBERTa outperforms the RoBERTa-based model in most datasets when combining different tokenizers with the models. On one hand, the combination of BigBERTa with the RoBERTa tokenizer shows a significantly lower score compared to the individual models. This indicates that the RoBERTa tokenizer might not be well-suited for tokenizing the biomedical domain text present in the NER task. On the other hand, the combination of RoBERTa model with the BigBERTa tokenizer performs relatively well on the EBM PICO, ChemProt, DDI, and GAD datasets. This suggests that the tokenization strategy of BigBERTa better suits the relation extraction task and complements the strengths of RoBERTa, leading to improved performance. In conclusion, the ablation study highlights the importance of considering both the model architecture and the tokenizer when tackling in-domain NLP tasks such as biomedical.

4.8.2 Hyperparameters Fine-tuning

For performance optimization, the practice of using recommended hyperparameters does not guarantee optimum results with the subsequent model. Moreover, we understand that while predefined learning parameters could lead to good performance, optimizing hyperparameters should yield better results. This hyperparameter search task is regarded as task-dependant or, more granularly, as data-dependant optimization. The hyperparameter exploration was conducted using Optuna framework [165] by searching for the optimum combination of parameters for a given input data. This algorithm utilizes a Tree-structured Parzen Estimator(TPE) to perform a dynamic search

loop through a parameter space for a determined number of trials. Since this probabilistic approach constructs models to approximate the performance of hyperparameters based on historical measurements, its efficiency largely depends on the number of trials. To limit the search cost, we defined a set of parameters and a value range that should be investigated and kept the rest of the parameters in their base configuration as predefined in the training arguments.

- learning_rate $\eta = [1e-4 : 1e-2]$
- training_epoch $i = [5 : 100]$
- per_device_train_batch_size $\beta = [4, 8, 12, 16, 32, 64]$

Table 4.9 provides all the resulting hyperparameters used for each dataset.

Table 4.9: Hyperparameters from Optuna obtained and used to fine-tune our model for each dataset

Dataset name	learning_rate	train_epochs	batch_size	Replications
BC5-Chem	2.9968e-05	29	8	5
BC5-disease	3.8067e-05	13	4	5
NCBI-disease	1.3479e-05	58	4	5
BC2GM	1.3297e-05	44	32	4
JNLPBA	1.8522e-05	56	64	4
Linnaeus	2.1036e-05	30	64	4
Species-800	1.7597e-05	30	8	4
BC4CHEMD	1.5543e-05	31	4	4
BioASQ	1.1637e-05	10	8	7
PubMedQA	2.0840e-05	5	16	4
EBM PICO	4.5284e-05	40	4	4
ChemProt	3.5699e-05	5	8	4
DDI	1.5971e-05	24	8	4
GAD	1.4770e-05	14	16	6
BIOSSES	7.1114e-05	53	64	5

Using these customized hyperparameters helped to boost our performance up to +3.11% F1 score on average on all downstream tasks by running 200 trials for each search.

4.8.3 Tokenizer Analysis

As our tokenizer was oriented toward improving real-world clinical document embedding, we wanted to understand its limits and performances. The fertility rate measures the average number of tokens produced from an input sequence of words. The ideal rate of 1 means that each input word has a corresponding token in the tokenizer's vocabulary. A tokenizer with a lower word segmentation has the advantage of generating the shortest input sequence size, whereas it preserves a consistent word representation in any context. The fertility rate being one of the best tools for measuring

how good a tokenizer [124], we systematically compare our tokenizer with general and biomedical-oriented tokenizers. Table 4.10 shows that our tokenizer has the lowest fertility rate on 7 of 9 datasets. A simple direct interpretation of these results highlights that our tokenizer can respectively encode on average 15.1% and 27.4% longer sequences than RoBERTa and BioBERT. However, we observed a poor performance on general English NER datasets such as CoNLL-2003 [172]. This generalization problem is mainly due to our tokenizer type, which is uncased. It is also due to the essence of the training objective, where we only used clinical and biomedical documents.

Table 4.10: Fertility rate of BioBERTa on NER datasets

Datasets	BERT	BioBERT	SciBERT	RoBERTa	Ours
BC5-Chem	1.258	1.343	**1.122**	1.224	1.127
BC5-disease	1.302	1.390	1.128	1.254	**1.056**
NCBI-disease	1.290	1.354	1.126	1.237	**1.096**
BC2GM	1.277	1.362	1.119	1.245	**1.088**
JNLPBA	1.456	1.544	1.244	1.409	**1.207**
BC5CDR	1.350	1.441	1.167	1.287	**1.074**
Linnaeus	1.215	1.285	**1.104**	1.189	1.107
BC4CHEMD	1.291	1.374	1.126	1.246	**1.094**
Species-800	1.268	1.344	1.147	1.241	**1.126**
CONLL2003	**1.237**	1.338	1.356	1.278	1.458

While it has been established that specialized tokenizers improve the downstream performance of the dedicated LM in almost every task and language [124], we noticed that some models utilized pre-existing tokenizers without considering the importance of a tailored vocabulary. For instance, we found that BioLinkBERT$_{Large}$ [125] shares the same tokenizer vocabulary with BioM-ELECTRA$_{Large}$ [133], BioMegatron$_{cased}$ [119] uses the BioBERT$_{cased}$ [116] tokenizer, while BioMegatron$_{uncased}$ [119] utilized the BERT$_{uncased}$ [114] vocabulary. Further experiments should demonstrate the consequence of default tokenizers for a domain-specific LM.

4.9 Discussion

4.9.1 Effect of a Dedicated Tokenizer

Employing an open dictionary of medical terms of about 98119 terminologies, we assessed the embedding capability of our models in a pure medical domain. We found out that our tokenizer vocabulary has the highest (13.01%) occurrences of whole medical terminologies compared to BioBERT$_{Large}$, PubMedBERT$_{microsoft}$ (9.05%), and BioLinkBERT$_{Large}$ (3.97%) while BigBird$_{base}$ initially has only 2.35%. We also observed an averaged embedding length gain of +7.71% across datasets compared to the source model.

Moreover, the design choices of a domain-specific language model have a considerable impact on the performance of the consequent model. We demonstrated that training a language model among diverse text genres is important. The combination of a byte-based tokenizer and a real-world in-domain document provides a tokenizer with

minimum subword fertility to prevent a potential over-segmentation of medical terminologies. Ultimately, our tokenizer extensively increased the input sequence length and intuitively exhibited that the tokenizer is well suited to the specific domain.
However, we noticed that having a cased and uncased vocabulary file can subsequently affect the tokenization process for general terms. For example, as reported in the table 4.7, using our uncased vocabulary file, our tokenizer was unable to recognize a basic word as *'Chicken-'* in *'Chickenpox'* simply because this token started with a capital *"C"* which is not present in our vocabulary.

4.9.2 Effect of Sparse Attention

We observed that our method heavily relies on supervised training in order to perform some tasks, such as QA, especially when the input is not long enough for the model to use sparse attention. These large input embedding models might have the potential to speed up even further LMs for reading comprehension applied to QA where the answer is searched in a potentially very large corpus of documents. This assumption is well demonstrated by the results on PubMedQA, where we encoded the context as one single input improved the results of +5% F1 score. As assessed with NER tasks, a shorter sparse attention model(2048) has the potential to perform better on relatively short inputs(+0.776 F1) since the blocks will skip fewer tokens. As for the input size, we noticed that the more data we have, the more memory we need; therefore, controlling the batch size is the ultimate key.

4.10 Conclusion

The design choices of a domain-specific language model have a considerable impact on the performance of the consequent model. We demonstrated that training a language model among diverse text genres is important. The combination of a byte-based tokenizer and a real-world in-domain document provides a tokenizer with minimum subword *fertility* to prevent a potential over-segmentation of medical terminologies. In the end, our tokenizer extensively increased the input sequence length and intuitively exhibited that the tokenizer was well suited to the specific domain. However, we noticed that having a cased and uncased vocabulary file can subsequently affect the tokenization process for general terms. Even if we couldn't reach state-of-the-art results on all datasets, we observed a constant improvement over the default values from the hyperparameters fine-tuning.

Sparse attention was initially proposed to approximate full attention in the most efficient way [122], with no striving to outperform the latter. However, our results suggest that a combination of diversity in training data, the ability to embed long-term dependencies, and an appropriate set of hyperparameters yield better performance.

Medical BigBERTa enhanced the performance of Biomedical language models without compromising on training and inference in terms of time and memory costs. We provide this model with a significantly adapted tokenizer that prevents over-segmentation and breaks through clinical and biomedical domains. We improved our performance through a tailored data-dependant hyperparameters optimization.

In summary, while the use of language models for biomedical and clinical text holds great promise, there are several challenges that need to be addressed by further research.

Type	Input examples	BioBERT	RoBERTa	BioBERTa
Drugs and chemicals	methylprednisolone, Omeprazole, ibuprofen, naproxen'	['met', '##hyl', '##p', '##red', '##nis', '##olo', '##ne', ',', 'O', '##me', '##pra', '##zo', '##le', ',', 'i', '##bu', '##p', '##ro', '##fen', ',', 'nap', '##ro', '##xen']	['methyl', 'pred', 'nis', 'ol', 'one', ',', 'ĠOm', 'ep', 'raz', 'ole', ',', 'Ġib', 'up', 'ro', 'fen', ',', 'Ġ', 'Ġnap', 'rox', 'en']	['methyl', 'prednisolone', ',', 'ĠOm', 'eprazole', ',', 'Ġibuprofen', ',', 'Ġ', 'Ġnaproxen']
prefixes and suffixes	Anticholinergic, Dyspnea, Hypothyroidism, Endometriosis	['Anti', '##cho', '##liner', '##gic', ',', 'D', '##ys', '##p', '##nea', ',', 'H', '##y', '##pot', '##hy', '##roid', '##ism', ',', 'End', '##ome', '##tri', '##osis']	['Ant', 'ich', 'olin', 'ergic', ',', 'ĠDys', 'p', 'nea', ',', 'ĠHyp', 'othy', 'roid', 'ism', ',', 'ĠEnd', 'omet', 'ri', 'osis']	['Antic', 'hol', 'inergic', ',', 'ĠDys', 'pnea', ',', 'ĠHyp', 'oth', 'yroidism', ',', 'ĠEnd', 'ometriosis']
General medical notes	pt remains on PSV with no vent changes today, his MDI puffs inc. to 6-8pfs Q4H	['p', '##t', 'remains', 'on', 'PS', '##V', 'with', 'no', 'vent', 'changes', 'today', ',', 'his', 'MD', '##I', 'pu', '##ffs', 'in', '##c', '.', 'to', '6', '-', '8', '##pf', '##s', 'Q', '##4', '##H']	['pt', 'Ġremains', 'Ġon', 'ĠPS', 'V', 'Ġwith', 'Ġno', 'Ġvent', 'Ġchanges', 'Ġtoday', ',', 'Ġhis', 'ĠMD', 'I', 'Ġp', 'uffs', 'Ġinc', '.', 'Ġto', 'Ġ6', '-', '8', 'p', 'fs', 'ĠQ', '4', 'H']	['pt', 'Ġremains', 'Ġon', 'ĠPSV', 'Ġwith', 'Ġno', 'Ġvent', 'Ġchanges', 'Ġtoday', ',', 'Ġhis', 'ĠMDI', 'Ġpuffs', 'Ġinc', '.', 'Ġto', 'Ġ6', '-', '8', 'p', 'fs', 'ĠQ', '4', 'H']
Diseases and diagnostics	Gastroesophageal, Parkinson's Disease, Chickenpox, Pneumonia	['Gas', '##tro', '##es', '##op', '##hage', '##al', ',', 'Parkinson', "'", 's', 'Disease', ',', 'Chicken', '##pox', ',', 'P', '##ne', '##um', '##onia']	['G', 'ast', 'ro', 'es', 'oph', 'age', 'al', ',', 'ĠParkinson', "'s", 'ĠDisease', ',', 'ĠChicken', 'pox', ',', 'ĠP', 'neum', 'onia']	['G', 'astro', 'esophageal', ',', 'ĠParkinson', "'s", 'ĠDisease', ',', 'ĠCh', 'icken', 'pox', ',', 'ĠP', 'neumonia']
Procedures	Electrocardiogram, Thermotherapy, Mastectomy, Vertebroplasty	['El', '##ec', '##tro', '##card', '##io', '##gram', ',', 'The', '##rm', '##otherapy', ',', 'Ma', '##ste', '##ct', '##omy', ',', 'V', '##ert', '##eb', '##rop', '##last', '##y']	['Elect', 'ro', 'card', 'i', 'ogram', ',', 'ĠTher', 'mother', 'apy', ',', 'ĠMast', 'ectomy', ',', 'ĠVer', 'te', 'bro', 'pl', 'asty']	['Elect', 'rocardiogram', ',', 'ĠTherm', 'otherapy', ',', 'ĠMast', 'ectomy', ',', 'ĠVer', 'teb', 'roplasty']
Acronyms and abbreviations	cap, pt, ANED, ARF, Rx, Dx, DM, ABG, ADHD	['cap', ',', 'p', '##t', ',', 'AN', '##ED', ',', 'AR', '##F', ',', 'R', '##x', ',', 'D', '##x', ',', 'D', '##M', ',', 'AB', '##G', ',', 'AD', '##HD']	['cap', ',', 'Ġpt', ',', 'ĠAN', 'ED', ',', 'ĠAR', 'F', ',', 'ĠRx', ',', 'ĠD', 'x', ',', 'ĠDM', ',', 'ĠAB', 'G', ',', 'ĠADHD']	['cap', ',', 'Ġpt', ',', 'ĠAN', 'ED', ',', 'ĠARF', ',', 'ĠR', 'x', ',', 'ĠD', 'x', ',', 'ĠDM', ',', 'ĠAB', 'G', ',', 'ĠADHD']
Cells and body	Neutrophil, leukocytes, cardiovascular, mammillitis	['N', '##eu', '##tro', '##phi', '##l', ',', 'le', '##uk', '##ocytes', ',', 'card', '##iovascular', ',', 'ma', '##mm', '##ill', '##itis']	['Ne', 'ut', 'roph', 'il', ',', 'Ġle', 'uk', 'ocytes', ',', 'Ġcardiovascular', ',', 'Ġmamm', 'ill', 'itis']	['Neut', 'rophil', ',', 'Ġleukocytes', ',', 'Ġcardiovascular', ',', 'Ġmamm', 'ill', 'itis']

Figure 4.7: Example of tokenization of random biomedical and clinical terms

These challenges include the lack of large annotated datasets, domain-specificity of clinical text, lack of transparency and interpretability, and ethical concerns. Addressing these challenges will facilitate the development of optimized language models toward improving healthcare outcomes and accelerating medical research.

Chapter 5

Conclusion

In this thesis, we have focused on the development and optimization of multimodal deep-learning predictive models for electronic health records. Through our research, we have covered various aspects.

We investigated various neural language modeling pipelines for outcome prediction in medical text data and identified the most effective approaches for analyzing clinical text documents for mortality prediction. Our experiments revealed that contextualized models, specifically those based on BERT and BioBERT, outperformed traditional word embedding models such as GloVe and Word2Vec. Furthermore, we found that combining different models and embeddings, as well as implementing transfer learning techniques, can further improve prediction performance. Our findings have important implications for the development of NLP-based clinical decision support systems that can assist healthcare professionals in making more accurate and timely patient management decisions. Overall, chapter 2 highlights the importance of selecting appropriate neural language modeling pipelines and utilizing advanced techniques to achieve optimal prediction results.

In the optic of handling the multimodality of the medical structured data, chapter 3 presented a novel approach to bridge the gap between medical tabular data and NLP predictive models using fuzzy logic. The proposed approach addresses the challenges of extracting useful information from structured medical data by transforming them into narrative texts that can be used for predictive modeling. The results of the experiments show that the approach has high accuracy and significantly improves interpretability, a crucial point for the healthcare domain. We believe that this research paves the way for a more unified and comprehensive analysis of structured and unstructured EHR data

With the power of language models demonstrated, it was natural to push the boundaries of existing models in order to build more adapted to raw and long medical documents such as narratives. Chapter 4 has discussed several optimization methods for transformer-based models that have been proposed in recent literature, including pretraining techniques, sparse attention, and hyperparameter fine-tuning approaches. The combination of these practices has shown to improve the performance of transformer-based models for various medical prediction tasks, from classification and similarity to Q&A. Our optimization approach substantially improves the processing and understanding of biomedical texts for different tasks in healthcare applications.

However, there is still room for improvement, and further research is needed to fully exploit the potential of these models. With the development of new optimization techniques and the availability of more extensive medical datasets, the future looks

bright for transformer-based models in medical document processing.

5.1 Limitations

Despite our achievements, our work has several limitations.

- **Data availability**: Our research is based on MIMIC-III dataset, focussing primarily on text-based EHR data, both structured and unstructured. However, the unavailability on time of a subset of the dataset, particularly images, significantly limited the scope and generalizability of a multimodal research project. Having been released recently to the public, we couldn't embrace the depth and richness of a fully integrated multimodal learning of EHR data. While there are often practical limitations and constraints around data access, the lack of timely and comprehensive data can be a significant challenge for researchers in multimodal analysis.
- **Quality of generated artificial narratives**: Although our approach to transforming numerical data into natural language text demonstrated promising results, the generated artificial narratives may not be as rich and informative as actual clinical narratives written by healthcare providers. The artificial narratives may not capture certain nuances, contextual information, or the expertise of healthcare providers in describing the patient's condition. In addition, the proposed approach utilizes a limited number of medical features due to the manual feature engineering on the related universe of discourse.
- **Reliance on a single EHR dataset**: Our research is based on the MIMIC-III dataset, which, although comprehensive, may not fully represent the diversity of EHR data from other sources or countries. The generalizability of our findings could be limited if the models and approaches developed in this study do not perform as well on other datasets with different characteristics.

These limitations highlight the areas where our research could be improved or expanded upon, offering valuable insights for future studies to build upon the foundation laid by our work. The following section suggests ways to address these limitations and develop more compelling, comprehensive, and robust multimodal deep-learning predictive models for electronic health records.

5.2 Future Research

Building upon the findings of our research, there are several directions for future work:

- Further exploration and refinement of the proposed approach to generate artificial narratives from medical tabular data, with a focus on improving the quality and informativeness of the generated text. This could also explore more on the explainability using different
- Expansion of the research to include other types of multimodal data, such as medical imaging, and audio recordings, to develop more comprehensive and integrated predictive models for healthcare applications.

- The development of more advanced and specialized biomedical large language models(LLM) and large multimodal models (LMM) that can handle a broader range of biomedical data and adapt to the specific characteristics of different EHR datasets.

- Evaluation and comparison of the proposed approaches on other EHR datasets, including those from different sources or countries, to assess their generalizability and applicability in various healthcare contexts.

In conclusion, our research has demonstrated the potential of multimodal deep-learning predictive models for electronic health records, providing valuable insights and innovative approaches to improve clinical decision-making. As the field of healthcare informatics and artificial intelligence continue to evolve, we believe that our findings and the future research directions outlined above will contribute significantly to advancing this critical area of study.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[2] O. W. of The Office of the National Coordinator for Health Information Technology (ONC). (2022) What are electronic health records (ehrs). [Online]. Available: https://www.healthit.gov/topic/health-it-and-health-information-exchange-basics/what-are-electronic-health-records-ehrs

[3] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[4] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

[5] Y. LeCun, Y. Bengio, G. Hinton *et al.*, "Deep learning. nature, 521 (7553), 436-444," *Google Scholar Google Scholar Cross Ref Cross Ref*, p. 25, 2015.

[6] I. Yoo, P. Alafaireet, M. Marinov, K. Pena-Hernandez, R. Gopidi, J.-F. Chang, and L. Hua, "Data mining in healthcare and biomedicine: a survey of the literature," *Journal of medical systems*, vol. 36, pp. 2431–2448, 2012.

[7] J. He, S. L. Baxter, J. Xu, J. Xu, X. Zhou, and K. Zhang, "The practical implementation of artificial intelligence technologies in medicine," *Nature medicine*, vol. 25, no. 1, pp. 30–36, 2019.

[8] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao *et al.*, "Interpretability of deep learning models: A survey of results," in *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*. IEEE, 2017, pp. 1–6.

[9] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 361–370, 2017.

[10] K. E. Henry, D. N. Hager, P. J. Pronovost, and S. Saria, "A targeted real-time early warning score (trewscore) for septic shock," *Science translational medicine*, vol. 7, no. 299, pp. 299ra122–299ra122, 2015.

[11] P. Rajpurkar, J. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. P. Langlotz *et al.*, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists," *PLoS medicine*, vol. 15, no. 11, p. e1002686, 2018.

[12] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[14] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ digital medicine*, vol. 1, no. 1, p. 18, 2018.

[15] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[16] G. Chassagnon, M. Vakalopoulou, E. Battistella, S. Christodoulidis, T.-N. Hoang-Thi, S. Dangeard, E. Deutsch, F. Andre, E. Guillo, N. Halm *et al.*, "Ai-driven quantification, staging and outcome prediction of covid-19 pneumonia," *Medical image analysis*, vol. 67, p. 101860, 2021.

[17] D. Baptista, P. G. Ferreira, and M. Rocha, "Deep learning for drug response prediction in cancer," *Briefings in bioinformatics*, vol. 22, no. 1, pp. 360–379, 2021.

[18] C. Dreisbach, T. A. Koleck, P. E. Bourne, and S. Bakken, "A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data," *International journal of medical informatics*, vol. 125, pp. 37–46, 2019.

[19] S. Velupillai, H. Suominen, M. Liakata, A. Roberts, A. D. Shah, K. Morley, D. Osborn, J. Hayes, R. Stewart, J. Downs *et al.*, "Using clinical natural language processing for health outcomes research: overview and actionable suggestions for future advances," *Journal of biomedical informatics*, vol. 88, pp. 11–19, 2018.

[20] S. Locke, A. Bashall, S. Al-Adely, J. Moore, A. Wilson, and G. B. Kitchen, "Natural language processing in medicine: a review," *Trends in Anaesthesia and Critical Care*, vol. 38, pp. 4–9, 2021.

[21] A. C. Morris, "Management of pneumonia in intensive care," *Journal of Emergency and Critical Care Medicine*, vol. 2, no. 0, 2018. [Online]. Available: https://jeccm.amegroups.com/article/view/4830

[22] I. Rudan, K. O'Brien, H. Nair, L. Liu, E. Theodoratou, S. Qazi, I. Luksic, C. Walker, R. Black, H. Campbell, and G. Reference, "Epidemiology and etiology of childhood pneumonia in 2010: Estimates of incidence, severe morbidity,

mortality, underlying risk factors and causative pathogens for 192 countries," *J Glob Health*, vol. 3, p. 10401, 05 2013.

[23] C. Mugisha and I. Paik, "Pneumonia outcome prediction using structured and unstructured data from ehr," 12 2020, pp. 2640–2646.

[24] I. Li, J. Pan, J. Goldwasser, N. Verma, W. P. Wong, M. Y. Nuzumlalı, B. Rosand, Y. Li, M. Zhang, D. Chang *et al.*, "Neural natural language processing for unstructured data in electronic health records: a review," *arXiv preprint arXiv:2107.02975*, 2021.

[25] R. Mann and J. Williams, "Standards in medical record keeping," *Clinical medicine (London, England)*, vol. 3, pp. 329–32, 07 2003.

[26] N. R. Council, *Biomedical Models and Resources: Current Needs and Future Opportunities*. Washington, DC: The National Academies Press, 1998. [Online]. Available: https://www.nap.edu/catalog/6066/biomedical-models-and-resources-current-needs-and-future-opportunities

[27] J. L. Schlossberg, "Book review: Medical language processing: Computer management of narrative data by naomi sager, carol friedman, and margaret s. lyman (addison-wesley 1987)," *SIGCHI Bull.*, vol. 20, no. 1, p. 70–71, Jul. 1988. [Online]. Available: https://doi.org/10.1145/49103.1046397

[28] N. El-Rashidy, S. El-Sappagh, T. Abuhmed, S. Abdelrazek, and H. El-Bakry, "Intensive care unit mortality prediction: An improved patient-specific stacking ensemble model," *IEEE Access*, vol. PP, pp. 1–1, 07 2020.

[29] E. Ford, J. Carroll, H. Smith, D. Scott, and J. Cassell, "Extracting information from the text of electronic medical records to improve case detection: A systematic review," *Journal of the American Medical Informatics Association*, vol. 23, p. ocv180, 02 2016.

[30] S. Sheikhalishahi, R. Miotto, J. T. Dudley, A. Lavelli, F. Rinaldi, and V. Osmani, "Natural language processing of clinical notes on chronic diseases: systematic review," *JMIR medical informatics*, vol. 7, no. 2, p. e12239, 2019.

[31] B. Goldstein, A. Navar, and M. Pencina, "Opportunities and challenges in developing risk prediction models with electronic health records data: A systematic review," *Journal of the American Medical Informatics Association*, vol. 24, p. ocw042, 05 2016.

[32] V. Osmani, l. li, M. Danieletto, B. Glicksberg, J. Dudley, and O. Mayora, "Automatic processing of electronic medical records using deep learning," 05 2018, pp. 251–257.

[33] H. Liu, Y. Lussier, and C. Friedman, "A study of abbreviations in the umls," *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pp. 393–7, 02 2001.

[34] G. Maragatham and S. Devi, "Lstm model for prediction of heart failure in big data," *Journal of Medical Systems*, vol. 43, 03 2019.

[35] R. AlSaad, Q. Malluhi, I. Janahi, and S. Boughorbel, "Interpreting patient-specific risk prediction using contextual decomposition of bilstms: application to children with asthma," *BMC Medical Informatics and Decision Making*, vol. 19, 11 2019.

[36] X. Zhou, Y. Li, and W. Liang, "Cnn-rnn based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[37] J. Geraci, P. Wilansky, V. de Luca, A. Roy, J. Kennedy, and J. Straus, "Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression," *Evidence-based mental health*, vol. 20, 07 2017.

[38] Y. Wang, S. Sohn, S. Liu, F. Shen, L. Wang, E. Atkinson, S. Amin, and H. Liu, "A clinical text classification paradigm using weak supervision and deep representation," *BMC Medical Informatics and Decision Making*, vol. 19, 01 2019.

[39] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[40] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, and C. So, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics (Oxford, England)*, vol. 36, 09 2019.

[41] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *CoRR*, vol. abs/1904.05342, 2019. [Online]. Available: http://arxiv.org/abs/1904.05342

[42] S. Alhagry, A. A. Fahmy, and R. A. El-Khoribi, "Emotion recognition based on eeg using lstm recurrent neural network," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, 2017.

[43] R. Cui, M. Liu, A. D. N. Initiative *et al.*, "Rnn-based longitudinal analysis for diagnosis of alzheimer's disease," *Computerized Medical Imaging and Graphics*, vol. 73, pp. 1–10, 2019.

[44] B. K. Reddy and D. Delen, "Predicting hospital readmission for lupus patients: An rnn-lstm-based deep-learning methodology," *Computers in biology and medicine*, vol. 101, pp. 199–209, 2018.

[45] Z. Yi, S. Li, J. Yu, Y. Tan, Q. Wu, H. Yuan, and T. Wang, "Drug-drug interaction extraction via recurrent neural network with multiple attention layers," in *Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, Singapore, November 5–6, 2017, Proceedings 13*. Springer, 2017, pp. 554–566.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] G. Petmezas, K. Haris, L. Stefanopoulos, V. Kilintzis, A. Tzavelis, J. A. Rogers, A. K. Katsaggelos, and N. Maglaveras, "Automated atrial fibrillation detection using a hybrid cnn-lstm network on imbalanced ecg datasets," *Biomedical Signal Processing and Control*, vol. 63, p. 102194, 2021.

[48] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, "Medical text classification using convolutional neural networks," in *Informatics for Health: Connected Citizen-Led Wellness and Population Health*. IOS Press, 2017, pp. 246–250.

[49] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[50] J. Pennington, R. Socher, and C. 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.

[51] O. Bodenreider, "The unified medical language system (umls): Integrating biomedical terminology," *Nucleic acids research*, vol. 32, pp. D267–70, 02 2004.

[52] L. Grossman, E. Mitchell, G. Hripcsak, C. Weng, and D. Vawdrey, "A method for harmonization of clinical abbreviation and acronym sense inventories," *Journal of Biomedical Informatics*, vol. 88, 11 2018.

[53] G. Finley, S. Pakhomov, R. McEwan, and G. Melton, "Towards comprehensive clinical abbreviation disambiguation using machine-labeled training data," *AMIA Annual Symposium Proceedings*, vol. 2016, pp. 560–569, 02 2017.

[54] H. Xu and P. Stetson, "A study of abbreviations in clinical notes," *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, vol. 2007, pp. 821–5, 02 2007.

[55] J. Rello, A. Rodriguez, A. Torres, J. Roig, J. Violán, J. Garnacho-Montero, M. Torre, J. Sirvent, and M. Bodí, "Implications of copd in patients admitted to the intensive care unit by community-acquired pneumonia," *The European respiratory journal : official journal of the European Society for Clinical Respiratory Physiology*, vol. 27, pp. 1210–6, 07 2006.

[56] H. Çelikhisar, G. Ilkhan, and C. Arabaci, "Prognostic factors in elderly patients admitted to the intensive care unit with community-acquired pneumonia," *The Aging Male*, pp. 1–7, 06 2020.

[57] R. Puttini, A. Toffanello, R. Chaim, G. Alves, J. Rotzsch, E. Carvalho, E. Ishikawa, A. Araújo, and E. C. Oliveira, "Semantic framework for electronic health records," 01 2017, pp. 334–337.

[58] H. Sun, K. Depraetere, J. De Roo, G. Mels, B. De Vloed, M. Twagirumukiza, and D. Colaert, "Semantic processing of ehr data for clinical research," *Journal of biomedical informatics*, vol. 58, 10 2015.

[59] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Data processing and text mining technologies on electronic medical records: A review," *Journal of Healthcare Engineering*, vol. 2018, pp. 1–9, 04 2018.

[60] D. Kim, J. Lee, C. So, H. Jeon, M. Jeong, Y. Choi, W. Yoon, and M. Sung, "A neural named entity recognition and multi-type normalization tool for biomedical text mining," *IEEE Access*, vol. PP, pp. 1–1, 06 2019.

[61] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, 12 2010.

[62] ——, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 43–52, 12 2010.

[63] L. Viii, K. Intelligenz, and T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," 01 1999.

[64] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 10 2013.

[65] E. Choi, T. Bahadori, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," *CoRR*, vol. 56, 11 2015.

[66] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[67] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "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. [Online]. Available: https://www.aclweb.org/anthology/P16-2034

[68] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, 09 2014.

[69] G. Lemmon, S. Wesolowski, A. Henrie, M. Tristani-Firouzi, and M. Yandell, "A poisson binomial-based statistical testing framework for comorbidity discovery across electronic health record datasets," *Nature computational science*, vol. 1, no. 10, pp. 694–702, 2021.

[70] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style data augmentation for end-to-end asr," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 426–433.

[71] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[72] J. Guan, R. Li, S. Yu, and X. Zhang, "A method for generating synthetic electronic medical record text," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 1, pp. 173–182, 2021.

[73] J. Ive, N. Viani, J. Kam, L. Yin, S. Verma, S. Puntis, R. N. Cardinal, A. Roberts, R. Stewart, and S. Velupillai, "Generation and evaluation of artificial mental health records for natural language processing," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–9, 2020.

[74] M. Müller, M. Salathé, and P. E. Kummervold, "Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter," *arXiv preprint arXiv:2005.07503*, 2020.

[75] J. B. Awotunde, O. E. Matiluko, and O. W. Fatai, "Medical diagnosis system using fuzzy logic," *African Journal of Computing & ICT*, vol. 7, no. 2, pp. 99–106, 2014.

[76] I. Spasic and G. Nenadic, "Clinical text data in machine learning: systematic review. jmir med inform. 2020 mar 31; 8 (3): e17984. doi: 10.2196/17984."

[77] F. Bobillo and U. Straccia, "fuzzydl: An expressive fuzzy description logic reasoner," in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 923–930.

[78] C. Mugisha and I. Paik, "Optimization of biomedical language model with optuna and a sentencepiece tokenization for ner," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2022, pp. 3859–3861.

[79] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[80] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[81] P. J. M. Ali, R. H. Faraj, E. Koya, P. J. M. Ali, and R. H. Faraj, "Data normalization and standardization: a technical report," *Mach Learn Tech Rep*, vol. 1, no. 1, pp. 1–6, 2014.

[82] C. Mugisha and I. Paik, "Pneumonia outcome prediction using structured and unstructured data from ehr," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 2640–2646.

[83] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *Journal of biomedical informatics*, vol. 83, pp. 112–134, 2018.

[84] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," *Advances in neural information processing systems*, vol. 29, 2016.

[85] É. Arnaud, M. Elbattah, M. Gignon, and G. Dequen, "Deep learning to predict hospitalization at triage: Integration of structured data and unstructured text," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 4836–4841.

[86] R. Blumberg and S. Atre, "The problem with unstructured data," *Dm Review*, vol. 13, no. 42-49, p. 62, 2003.

[87] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]," *IEEE Transactions on automatic control*, vol. 42, no. 10, pp. 1482–1484, 1997.

[88] C. Gupta, A. Jain, and N. Joshi, "Fuzzy logic in natural language processing–a closer view," *Procedia computer science*, vol. 132, pp. 1375–1384, 2018.

[89] J. A. Goguen, "La zadeh. fuzzy sets. information and control, vol. 8 (1965), pp. 338–353.-la zadeh. similarity relations and fuzzy orderings. information sciences, vol. 3 (1971), pp. 177–200." *The Journal of Symbolic Logic*, vol. 38, no. 4, pp. 656–657, 1973.

[90] J. Kacprzyk and S. Zadrozny, "Computing with words is an implementable paradigm: Fuzzy queries, linguistic data summaries, and natural-language generation," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 461–472, 2010.

[91] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—i," *Information sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[92] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Applied soft computing*, vol. 92, p. 106275, 2020.

[93] E. Vlamou and B. Papadopoulos, "Fuzzy logic systems and medical applications," *AIMS neuroscience*, vol. 6, no. 4, p. 266, 2019.

[94] J. Zhang, C. Tao, and P. Wang, "A review of soft computing based on deep learning," in *2016 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*. IEEE, 2016, pp. 136–144.

[95] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (anfis) training approaches: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2263–2293, 2019.

[96] Y. Jiang, C. Yang, and H. Ma, "A review of fuzzy logic and neural network based intelligent control design for discrete-time systems," *Discrete Dynamics in Nature and Society*, vol. 2016, 2016.

[97] D. Feng, G. Burns, and E. Hovy, "Extracting data records from unstructured biomedical full text," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 837–846.

[98] J. L. Castro and M. Delgado, "Fuzzy systems with defuzzification are universal approximators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 149–152, 1996.

[99] A. Jain and A. Sharma, "Membership function formulation methods for fuzzy logic systems: A comprehensive review," *Journal of Critical Reviews*, vol. 7, no. 19, pp. 8717–8733, 2020.

[100] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[101] T. Searle, "icd9cms 0.2.1," https://pypi.org/project/icd9cms/, 2018, [ICD9CMS 0.2.1, a Python Package Index accessed 21-July-2021].

[102] J. Aitchison and C. G. Aitken, "Multivariate binary discrimination by the kernel method," *Biometrika*, vol. 63, no. 3, pp. 413–420, 1976.

[103] N. Geifman, R. Cohen, and E. Rubin, "Redefining meaningful age groups in the context of disease," *Age*, vol. 35, no. 6, pp. 2357–2366, 2013.

[104] D. Van Kuiken and M. M. Huth, "What is' normal?'evaluating vital signs," *Pediatric nursing*, vol. 39, no. 5, p. 216, 2013.

[105] R. Beasley, J. Chien, J. Douglas, L. Eastlake, C. Farah, G. King, R. Moore, J. Pilcher, M. Richards, S. Smith *et al.*, "Target oxygen saturation range: 92–96% versus 94–98%," *Respirology*, vol. 22, no. 1, pp. 200–202, 2017.

[106] C. Mugisha and I. Paik, "Optimization of biomedical language model with optuna and a sentencepiece tokenization for ner," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2022, pp. 3859–3861.

[107] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[108] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[109] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[110] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[111] A. Blum, S. Kuria, N. Blum *et al.*, "High serum lactate level may predict death within 24 hours," *Open medicine*, vol. 10, no. 1, p. 4, 2015.

[112] C. L. Hansen, S. S. Chaves, C. Demont, and C. Viboud, "Mortality associated with influenza and respiratory syncytial virus in the us, 1999-2018," *JAMA Network Open*, vol. 5, no. 2, pp. e220 527–e220 527, 2022.

[113] A survey of surveys (nlp ml). [Online]. Available: https://github.com/NiuTrans/ABigSurvey

[114] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[115] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[116] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[117] I. Spasic, G. Nenadic *et al.*, "Clinical text data in machine learning: systematic review," *JMIR medical informatics*, vol. 8, no. 3, p. e17984, 2020.

[118] Office of the national coordinator for health information technology. office-based physician electronic health record adoption, health it quick-stat 50. [Online]. Available: https://www.healthit.gov/data/quickstats/office-based-physician-electronic-health-record-adoption

[119] H.-C. Shin, Y. Zhang, E. Bakhturina, R. Puri, M. Patwary, M. Shoeybi, and R. Mani, "Biomegatron: Larger biomedical domain language model," *arXiv preprint arXiv:2010.06060*, 2020.

[120] R. Tinn, H. Cheng, Y. Gu, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Fine-tuning large neural language models for biomedical natural language processing," *arXiv preprint arXiv:2112.07869*, 2021.

[121] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[122] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[123] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv:2004.05150*, 2020.

[124] P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych, "How good is your tokenizer? on the monolingual performance of multilingual language models," *arXiv preprint arXiv:2012.15613*, 2020.

[125] M. Yasunaga, J. Leskovec, and P. Liang, "Linkbert: Pretraining language models with document links," *arXiv preprint arXiv:2203.15827*, 2022.

[126] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[127] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021.

[128] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[129] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[130] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[131] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[132] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[133] S. Alrowili and K. Vijay-Shanker, "Biom-transformers: building large biomedical language models with bert, albert and electra," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, 2021, pp. 221–227.

[134] H. Gong, Y. Shen, D. Yu, J. Chen, and D. Yu, "Recurrent chunking mechanisms for long-text machine reading comprehension," *arXiv preprint arXiv:2005.08056*, 2020.

[135] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, "Hierarchical transformers for long document classification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 838–844.

[136] Y. Li, R. M. Wehbe, F. S. Ahmad, H. Wang, and Y. Luo, "Clinical-longformer and clinical-bigbird: Transformers for long clinical sequences," *arXiv preprint arXiv:2201.11838*, 2022.

[137] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[138] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[139] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.

[140] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021.

[141] P. Gage, "A new algorithm for data compression," *C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[142] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[143] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, "Character-level language modeling with deeper self-attention," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3159–3166.

[144] C. Mugisha and I. Paik, "Comparison of neural language modeling pipelines for outcome prediction from unstructured medical text notes," *IEEE Access*, vol. 10, pp. 16 489–16 498, 2022.

[145] W. Zhang, W. Wei, W. Wang, L. Jin, and Z. Cao, "Reducing bert computation by padding removal and curriculum learning," in *2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2021, pp. 90–92.

[146] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[147] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," in *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, 2019.

[148] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, "Biocreative v cdr task corpus: a resource for chemical disease relation extraction," *Database*, vol. 2016, 2016.

[149] R. I. Doğan, R. Leaman, and Z. Lu, "Ncbi disease corpus: a resource for disease name recognition and concept normalization," *Journal of biomedical informatics*, vol. 47, pp. 1–10, 2014.

[150] L. Smith, L. K. Tanabe, C.-J. Kuo, I. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, M. Torii *et al.*, "Overview of biocreative ii gene mention recognition," *Genome biology*, vol. 9, no. 2, pp. 1–19, 2008.

[151] J.-D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier, "Introduction to the bio-entity recognition task at jnlpba," in *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*. Citeseer, 2004, pp. 70–75.

[152] M. Gerner, G. Nenadic, and C. M. Bergman, "Linnaeus: a species name identification system for biomedical literature," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–17, 2010.

[153] E. Pafilis, S. P. Frankild, L. Fanini, S. Faulwetter, C. Pavloudi, A. Vasileiadou, C. Arvanitidis, and L. J. Jensen, "The species and organisms resources for fast and accurate identification of taxonomic names in text," *PloS one*, vol. 8, no. 6, p. e65390, 2013.

[154] O. Taboureau, S. K. Nielsen, K. Audouze, N. Weinhold, D. Edsgärd, F. S. Roque, I. Kouskoumvekaki, A. Bora, R. Curpan, T. S. Jensen *et al.*, "Chemprot: a disease chemical biology database," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D367–D372, 2010.

[155] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, and T. Declerck, "The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions," *Journal of biomedical informatics*, vol. 46, no. 5, pp. 914–920, 2013.

[156] G. Soğancıoğlu, H. Öztürk, and A. Özgür, "Biosses: a semantic sentence similarity estimation system for the biomedical domain," *Bioinformatics*, vol. 33, no. 14, pp. i49–i58, 2017.

[157] S. Baker, I. Silins, Y. Guo, I. Ali, J. Högberg, U. Stenius, and A. Korhonen, "Automatic semantic classification of scientific literature according to the hallmarks of cancer," *Bioinformatics*, vol. 32, no. 3, pp. 432–440, 2016.

[158] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, "Pubmedqa: A dataset for biomedical research question answering," *arXiv preprint arXiv:1909.06146*, 2019.

[159] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[160] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[161] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[162] S. Wright, J. Nocedal *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[163] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[164] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: a big comparison for nas," *arXiv preprint arXiv:1912.06059*, 2019.

[165] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[166] M. Seeger, "Gaussian processes for machine learning," *International journal of neural systems*, vol. 14, no. 02, pp. 69–106, 2004.

[167] S. A. Della Pietra, M. Epstein, S. Roukos, and T. Ward, "Fertility models for statistical natural language understanding," in *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, 1997, pp. 168–173.

[168] V. Kocaman and D. Talby, "Biomedical named entity recognition at scale," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 635–646.

[169] Z. Yuan, Y. Liu, C. Tan, S. Huang, and F. Huang, "Improving biomedical pretrained language models with knowledge," *arXiv preprint arXiv:2104.10344*, 2021.

[170] L. N. Phan, J. T. Anibal, H. Tran, S. Chanana, E. Bahadroglu, A. Peltekian, and G. Altan-Bonnet, "Scifive: a text-to-text transformer model for biomedical literature," *arXiv preprint arXiv:2106.03598*, 2021.

[171] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, "A neural network for factoid question answering over paragraphs," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 633–644.

[172] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.