Premier University, Chittagong

Bachelor's Thesis

Sequence-to-sequence Named entity recognition and Relation extraction of biomedical data

A thesis submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science

in the

Department of Computer Science and Engineering

July 2023

Dedicated To Our Loving Parents, Our Teachers and Well-wishers ...

Approval

This thesis entitled "**Sequence-to-sequence Named entity recognition and Relation extraction of biomedical data**" prepared and submitted by *Md Mohsin Ali, Id: 1703310201393; Tonmoy Barua, Id: 1703310201373; Srabosthy Das Prama, Id: 1703310201396 of 33rd Batch* in partial fulfillment of the requirements for the degree of Bachelor of Science in the Department of Computer Science and Engineering has been examined and is recommended for approval and acceptance.

Prof. Dr. Taufique Sayeed
Dean, Faculty of Science and Engineering
&
Chairman, Department of Computer Science and Engineering, Premier University

Puja Chakraborty
Lecturer
Department of Computer Science and Engineering, Premier University
(Thesis Supervisor)

Declaration of Authorship

We, Md Mohsin Ali, Id: 1703310201393; Tonmoy Barua, Id: 1703310201373; Srabosthy Das Prama, Id: 1703310201396 of 33^{rd} Batch declare that this thesis titled, '**Sequence-to-sequence Named entity recognition and Relation extraction of biomedical data**' and the work presented in it are our own. We confirm that:

- This work was done wholly while we were a candidate for the degree of Bachelor of Science in the Department of Computer Science and Engineering at the Premier University.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourself jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed By:

- Md Mohsin Ali: _____

- Tonmoy Barua: _____

- Srabosthy Das Prama: _____

Acknowledgements

At first, We want to express gratitude to the Almighty for His endless kindness for keeping us mentally and physically fit to complete this sophisticated task.

This study would not get its own shape without the general support of the Premier University, Chittagong which provided us the chance for our Bachelor's Program. We wish to acknowledge the help provided by the technical and support staff in the CSE department of our University. The completion of this thesis is not a result of our individual effort, but is an aggregate of co-operation of many other people.

Foremost, We would like to express our deepest thanks to our supervisor, Lecturer Puja Chakraborty, who is one kind of turning point in our life. Her vast knowledge in Natural Language Processing (NLP), Machine learning, and curiosity to disseminate that knowledge, understanding capability, patience, humbleness, tolerance, dealing, attitude, behavior and wisdom are really appreciable. We desire to have such kinds of qualities. Her directions and guidelines of preparing manuscripts, reports, presentations, posters makes us more dynamic and well-organized. We do not hesitate to certify her as the best academic scholar so far We have experienced. We would also like to thank our friends and family who supported us and offered deep insight into the study.

Finally, We would like to show our gratitude to all of my teachers who helped us a lot during our Bachelor's program; without their teaching, we couldn't learn even a bit of what we've learned all these years.

Department of Computer Science and Engineering

Abstract

Sequence-to-sequence Named entity recognition and Relation extraction of biomedical data

We introduce a new *dataset*[1] for Named entity recognition and Relation extraction, which is sequence-to sequence text annotation. The dataset is based on the NHP disease and conditions archive, meant to act as ground truth texts. These texts were subsequently used to annotate by us using an annotation tool named *doccano*[2]. The dataset includes 301 clinical diseases, 18,048 files, 2,02,000 words, and total 2,69,333 tokens annotated by approximately 3 different annotators. This can serve as a basis for a variety of Named entity recognition tasks such as information extraction, machine translation, sentiment analysis and recommendation system along with Relation extraction such as knowledge graph construction, question answering systems and event extraction. Developing NER and RE systems involves data collection by crawling respective websites and collecting disease *ICD-11*[3] codes, manual pre-processing , annotation using doccano, feature extraction, model development, evaluation, fine-tuning, deployment, and monitoring with weight and bias. The choice of algorithms and techniques can vary based on the data and application domain, and the performance of the system can be improved through continuous monitoring and updates.

Keywords: Named Entity Recognition, Relation Extraction, Data Annotation, Natural Language Processing.

[1]Sample Dataset `EN-BioNER`: https://huggingface.co/datasets/mohsin-riad/EN-BioDNER
[2]Annotation Tool: https://doccano.github.io/doccano/
[3]Disease Codes: https://icd.who.int/en

Contents

List of Figures

List of Tables

Abbreviations

AI	**A**rtificial **I**ntelligence
LSTM	**L**ong **S**hort **T**erm **M**emory
ML	**M**achine **L**earning
mAP	**m**ean **A**verage **P**recision
NLP	**N**atural **L**anguage **P**rocessing
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
NER	**N**amed **E**ntity **R**ecognition
RE	**R**elation **E**xtraction
DistilBERT	**D**istilled **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
RoBERTa	**R**obustly **o**ptimized **B**idirectional **E**ncoder **R**epresentations from **T**ransformers Pretaining **a**pproach
BioBERT	**B**iomedical **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

Chapter 1

Introduction

1.1 Motivation

Data is the modish 'oil' in this era of the digital revolution. Biomedical-named entity recognition (BioNER) and relation extraction (BioRE) are essential for extracting information from biomedical literature. BioNER aims to identify and classify biomedical entities, such as genes, proteins, diseases, and chemicals in text. BioRE aims to detect and categorize the relationships between these entities, such as gene-disease associations, protein-protein interactions, and drug-drug interactions. These tasks can help researchers to summarize large-scale details on a particular biomedical or clinical problem and to integrate them into networks for further analysis.

The following factors drive our approach to work for BioNER and BioRE: The rapid growth of biomedical literature makes it challenging to extract relevant information from unstructured text manually. The existing datasets for BioNER and BioRE are often limited in scope, size, or diversity and do not cover multiple entity types and relation pairs at the document level. The current methods for BioNER and BioRE are mainly based on deep learning models that require large amounts of annotated data and computational resources and may need to generalize better across different domains or languages. Therefore, I am interested in developing novel methods for BioNER and BioRE that can overcome these limitations and achieve high performance on various biomedical tasks.

1.2 Problem Description

Definition 1.2.1. Annotation is a means of populating a corpus by examining something in the world and then recording the observed characteristics. The process of assembling metadata to a dataset is known as data annotation that usually takes tags, which can be added to any data types, including text, images, and video. Fig 1.1 represents the data annotation process.

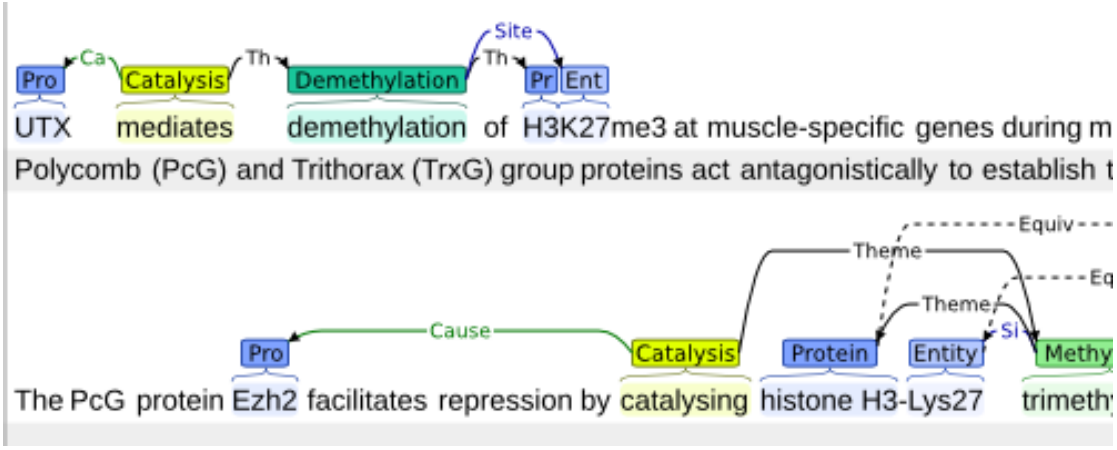

FIGURE 1.1: Data Annotation Process.

In this dissertation, we have used our EN-BioDNER dataset described in Chapter 3 and proposed an supervised way of detecting Named Entities and Relations between the Entities. Biomedical research and healthcare generate a massive amount of textual data, including scientific articles, clinical reports, and electronic health records. Extracting valuable information from this data is crucial for advancing medical knowledge and improving patient care. One important task in this domain is a biomedical disease named entity recognition (NER). The biomedical disease NER aims to identify and classify diseases mentioned within the text. This involves accurately detecting disease entities such as specific diseases, symptoms, causes, risk factors, or medical conditions and labelling them accordingly. On the other hand, relation extraction focuses on identifying and understanding the relationships between disease entities and other entities mentioned in the text, such as genes, drugs, or treatments. These relationships provide valuable insights into disease mechanisms, genetic associations, and treatment effectiveness.

1.2.1 Dataset Creation

We have created the EN-BioDNER dataset by following data annotation steps:

- Corpus Selection (Source of data): A collection of textual data.
- Crawling/Scraping: Automatic download of text files of data.
- Manual textual data collection.
- Named Entity selection.
- Selection of relation between two entities.
- Jsonl file creation.

1.2.2 Named Entity Recognition & Relation Extraction

For Named Entity Recognition, there are many approaches. Here we have used an unsupervised way that is briefly explained here.

- Firstly, We have converted `Jsonl` format data to `CoNLL2003` format data which is convenient for transformer architecture.
- Secondly, We tokenized the data using the Byte-Pair Encoding technique.
- Then, we created embedding per our model's id2label mapping.
- After training the pre-trained model for NER, we evaluated model's performance.
- After performing the previous analysis, we used the huggingface inference pipeline to produce Entity recognition.

For Relation Extraction, we have used an supervised way that is briefly explained here.

- Firstly, We have converted `Jsonl` object format data to `Json line` format labeled data which is convenient for transformer architecture.
- Secondly, We tokenized the data using the BertTokenizerFast which is a builtin tokenizer.

- Then, we created Marker Tokens and Span Indexes and encoded the dataset using them.
- After training the pre-trained model for RE, we evaluated the model's performance.
- Finally performing the previous analysis, we predicted the relation between the entities to produce Relation Extraction.

The method of NER and RE detailed in Chapter 4.

1.3 Purpose of the Thesis

In this dissertation, we introduce the most extensive dataset named `EN-BioDNER`. Extracting valuable information from this data is crucial for advancing medical knowledge and improving patient care. Our dataset contains 301 clinical diseases, 18,048 files, 2,02,000 words, and total of 2,69,333 tokens annotated by approximately 3 different annotators using the text annotation tool named doccano. This is currently the field's largest and most comprehensive dataset. The contents of our dataset came from a diverse Health Portal. Design and implement state-of-the-art deep learning models specifically tailored for the accurate identification and classification of disease entities within biomedical texts. Propose novel approaches for extracting relationships between disease entities and other entities, such as genes, drugs, or treatments, mentioned in the biomedical texts. The process can be extended for our future work (the end goal).

1.4 Preview Key Points and the Thesis Statement

The endure of this thesis statement as follows:
Chapter 2 initiates some related works :

- Some previous datasets with their characteristics, statistics and types.
- Several Transfer-learning based models, Named entity recognition and Relation extraction methods.

In Chapter 3, we represent data annotation, data collection process, Dataset, and contents of the dataset according to ICD disease characteristics and the Dataset statistics.

Chapter 4 presents an overall system methodology and detail of our supervised method of Text classification and Relation between them.

Chapter 5, We have considered about results and evaluation of the Named Entity Recognition and Relation Extraction process. We started with an analysis of the dataset followed by a description of the evaluation strategy. Finally, we've terminated this chapter by displaying a comparative performance within results obtained from our supervised system and the other dataset (EN-BioDNER dataset (only bio-medical data)).

Finally, In Chapter 6, we terminated this thesis with the future objectives for our unsupervised system with our dataset and our end goal.

Chapter 2

Literature Review

NER is the task of identifying and classifying real-world objects such as persons, locations, organizations, and other named entities in text. This technology began to take shape in the late 1990s, particularly during the MUC-6 (Sixth Message Understanding Conference) and MUC-7 evaluations in 1995 and 1997 respectively, where NER was a significant part of the conference's shared tasks.

Relation Extraction, on the other hand, is the task of identifying and classifying the semantic relationships between pairs of named entities in text. This task started receiving significant attention during the Automatic Content Extraction (ACE) program conducted by NIST and sponsored by the US government in the early 2000s. This program aimed to advance the development of techniques for automatically extracting information like relations and events from text.

Pretrained word embeddings (Mikolov et al., 2013), (Pennington et al., 2014) and contextualised word embeddings (Peters et al., 2018) have helped the deep learning algorithms to improve their performance in NLP tasks. ULMFiT (Howard and Ruder, 2018), introduces the transfer learning approach to Natural language processing and OpenAI GPT (Radford et al., 2018), pretrains a transformer (Vaswani et al., 2017) for learning general language representations. Similar to ULMFiT and OpenAI GPT, BERT (Devlin et al., 2018) follows this fine tuning approach and introduces a powerful bidirectional language representation model using the transformer based model architecture. BERT achieves SOTA on most NLP tasks without any heavily-engineered task specific architectures. Following the success of BERT, XLNet (Yang

et al., 2020) with generalized autoregressive pretraining and RoBERTa (Liu et al., 2019) with robust pretraining techniques experiment with different pretraining objectives. ALBERT (Lan et al., 2019) uses weight sharing and embedding factorisation to reduce memory consumption and increase the training speed. ELECTRA (Clark et al., 2020) introduces sample-efficient 'replaced token detection' pretraining technique. ELECTRAsmall, trained with very little compute outperforms GPT and performs comparably with larger models like RoBERTa and XLNet.

In Relation extraction many different methods have been proposed to solve this problem Culotta and Sorensen, 2004; Sierra et al., 2008; Sahu and Anand, 2018; Zhang et al., 2019; Su et al., 2019). However, the language model methods redefine this field with their superior performance (Dai and Le, 2015; Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019; Su and VijayShanker, 2020). Among all the language models, BERT (Devlin et al., 2019) –a language representation model based on bidirectional Transformer (Vaswani et al., 2017), attracts lots of attention in different fields. Several BERT models have been adapted for biomedical domain: BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019), BlueBERT (Peng et al., 2019) and PubMedBERT (Gu et al., 2021). BioBERT, SciBERT and BlueBERT are pre-trained based on the general-domain BERT using different pre-training data. In contrast, PubMedBERT (Gu et al., 2021) is pre-trained from scratch using PubMed abstracts.

The NCBI Disease Corpus is a resource for named entity recognition and disease normalization in biomedical text mining. It was developed by the National Center for Biotechnology Information (NCBI), a part of the U.S. National Library of Medicine. The dataset consists of 793 PubMed articles with 6,892 disease mentions annotated with concept identifiers from the Medical Subject Headings (MeSH) and Online Mendelian Inheritance in Man (OMIM). These mentions are categorized into three groups: diseases, disease symptoms, and disease syndromes. This annotated corpus is valuable for researchers in biomedical text mining and NLP who aim to develop methods for disease mention recognition and normalization.

The results obtained are promising, which encouraged us to do more research in this direction. This, however, requires a lot of training data. Our dataset is mainly targeted to achieve this in our future endeavours.

Chapter 3

Dataset

3.1 What is Data?

Definition 3.1.1. Data is an individual unit that includes raw (natural) materials which do not carry any specific meaning. It facts that can be analyzed or used to gain information or make choices. It endures and has no point beyond its being. It can be in any kind, useful or not. It appears not to have meaning in itself. In computer language, a spreadsheet usually begins by taking data. Data describes a point or statement of an event without similarity to other items.

3.2 What is Dataset?

Definition 3.2.1. A Dataset is a set or collection of data. This set is normally presented in a tabular pattern. A dataset consists of roughly two components. The two components are rows and columns. Every column describes a particular variable. It is also a representation of a data management system. Different types of anonymous quantities variable such as height, weight, size, edge, etc. of an article. Many features comparing where columns of a record present a single variable and rows is the content of the dataset.

The physical structure of each record is nearly the same, and uniform throughout a data set. This is specified in the data control block record format parameter. The fixed-length records eliminate the need for any delimiter byte value for separate records. Examples of Dataset:

- Students dataset
- Research dataset
- Intellectual Property dataset
- Human Resource dataset
- Financial dataset
- Legal dataset
- Law Enforcement dataset
- Alumni/Foundation dataset
- Health dataset
- International Students/Faculty dataset
- Email dataset
- Audit dataset
- Security dataset
- Miscellaneous dataset

3.2.1 Types of Dataset

In Statistics, there are many datasets available for various kinds of information which is used in various field. They are [1]: Numerical data sets, Bivariate data sets, Multivariate data sets, Categorical data sets, Correlation data sets.

3.2.1.1 Numerical Dataset

A set of all numerical data. It deals only with numbers. Some of the examples are:

- Weight and height of a person.
- The count of RBC in a medical report.
- Number of pages present in a book.

3.2.1.2 Bivariate Dataset

A dataset that has two variables is called a Bivariate dataset. It deals with the correlation among the two variables.

Example: To find the percentage score and age of the user in a group. Score and age are considered as two variables.

[1] https://byjus.com/maths/data-sets/

3.2.1.3 Multivariate Dataset

A data set with multiple variables.

Example: If we have to measure the length, width, height, volume of a rectangular box, we have to use multiple variables to distinguish between those entities.

3.2.1.4 Categorical Dataset

Categorical data sets represent features or characteristics of a person or an object.

Example: A person's gender (male or female), Marital status (married/unmarried)

3.2.1.5 Correlation Dataset

The set of values that demonstrate some relationship with each other indicates correlation data sets. Here the deals are found to depend on any other.

Example: A tall person is considered to be heavier than a short person. So here, the weight and height variables are dependent on each other.

3.2.2 Properties of Dataset

Before starting any statistical resolution, it is crucial to know the characteristics of the data. The properties of dataset:

- Centre of data.
- Skewness of data.
- Spread among the data members.
- Presence of outliers.
- Correlation among the data.
- Type of probability distribution that the data follows.

3.3 Data Annotation

3.3.1 What is Annotation?

Definition 3.3.1. An annotation is a piece of additional information correlated with a particular point in a document or another portion of data. Sometimes annotations are represented at the edge of text pages. We are annotating different digital media, web annotation, and text annotation[2].

3.3.1.1 Benefits of Annotating a Text

The benefits of annotation[3] include:

- Annotating supports to give attention and preserves time.
- Maintaining track of key concepts and issues.
- Assisting formulate thoughts and issues for more profound knowledge.
- Fostering separating and evaluating manuscripts.
- Encouraging the user to create deductions and draw inferences on the text.
- Allowing the user to immediately refer back to the reader without reading the text in its result.

3.3.1.2 Annotation Strategies

Annotating a text occurs when the user regards a text to indicate essential places or something they don't understand. Sometimes learners annotate by marking a phrase, enclosing a word, or highlighting a sentence. It also includes handwritten documents in the edge; these handwritten documents might be ideas or issues about the text. This method of annotating helps the reader keep track of ideas, issues and maintains a more profound knowledge of the text. Key strategies to highlight:

- Summarizing
- Questioning
- Predicting
- Making attachments
- Getting the main idea and crucial features Outlining document structure
- Identifying and defining new concepts [4]

[2] Annotation: https://en.wikipedia.org/wiki/Annotation
[3] Annotation Benefits: https://www.sadlier.com/school/ela-blog/
[4] Annotation Strategies: https://microcredentials.digitalpromise.org/explore

3.3.2 What is Data Annotation ?

Definition 3.3.2. Annotation is a means of populating a corpus by examining something in the world and then recording the observed characteristics. Data annotation is the method of labelling data to allow a model to make decisions and take actions. Where available in different forms like text, video or images.

It plays a significant role in ensuring our AI and machine learning projects are trained with the correct information to learn from data annotation. In the Machine learning model, data annotation and labelling provide the primary setup for supplying. Data annotation contains the text, images and videos to annotate or label the data from the images while assuring the accuracy to make sure the machines can recognize it through computer vision[5]. The dataset is essentially organized into a particular model that helps to process the needed information. In this section, we will brief about the different stages of our dataset creation and annotation process.

3.3.3 Data Annotation Types

There are several types of data annotation:

- **Text annotation:** Text annotation results from attaching a note or documents to a text, which may add highlights or marking, remarks, messages, tags, and sections (Fig. 3.1). Text annotations can consist of letters or notes written for a reader's expectations and shared annotations written for collaborative writing and analysis or social reading and sharing. In some fields, text annotation is similar to metadata and gives information about a text without modifying that original text. Text annotations reserve this term, especially for handwritten documents created in the borders of manuscripts[6].

- **Image annotation:** Image annotation appears in several forms, from bounding boxes, which are imaginary boxes drawn on images, to semantic segmentation, where each pixel within an image has attached a meaning (Fig. 3.2). This label typically maintains a machine learning model that recognizes the annotated area as a distinct object type. This type of data often serves as ground truth for image recognition models that can identify and block sensitive content, guide autonomous vehicles, or perform facial recognition tasks. Various types of famous image anno-

[5]Data Annotation: https://medium.com/anolytics/
[6]Text Annotation: https://en.wikipedia.org/wiki/Text_annotation

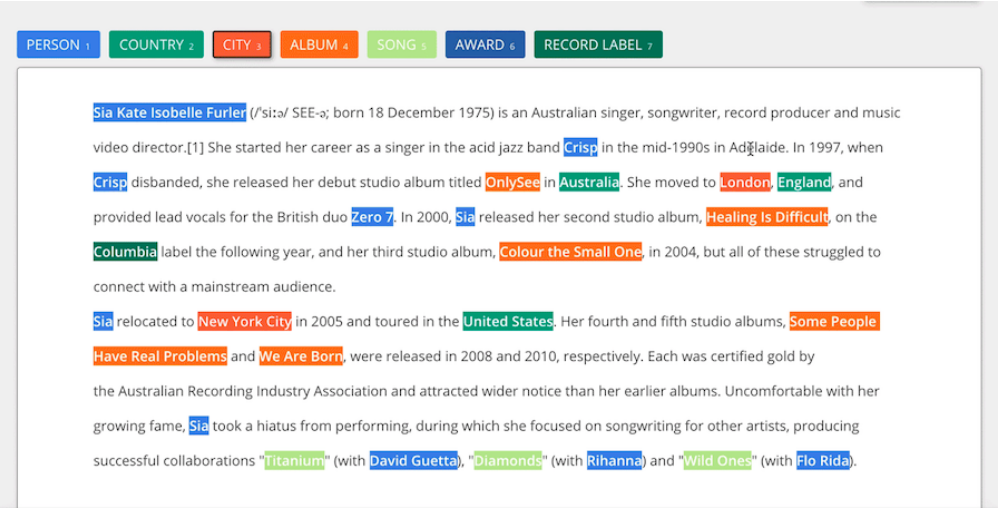

FIGURE 3.1: Text Annotation.

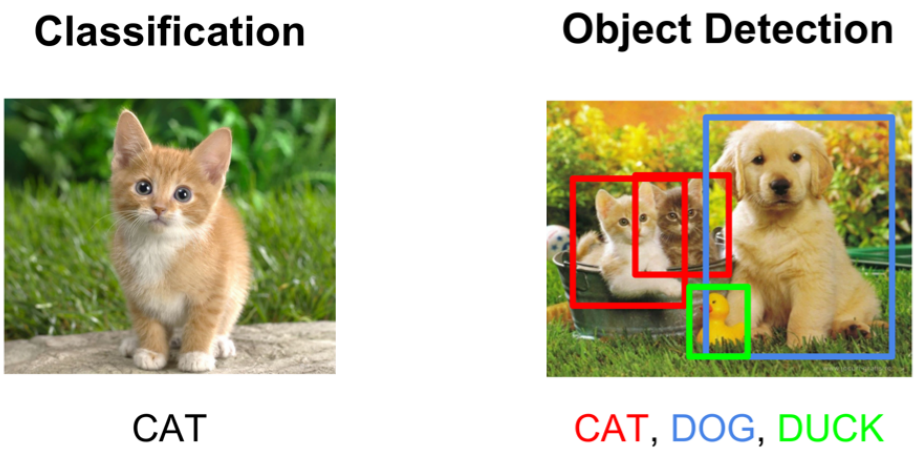

FIGURE 3.2: Image Annotation.

tation are: Bounding box annotation, polygon annotation, semantic segmentation, landmark annotation, polylines annotation and 3D point cloud annotation.

- **Linguistic annotation:** Linguistic annotation, also known as corpus annotation, is the tagging of language data in text or spoken form. Linguistic annotation seeks to identify and flag grammatical, phonetic, and semantic linguistic elements within a text or audio recording body. It is the process of formatting data in a way in which it's useful for machine learning and artificial intelligence. There are companies and resources out there that have platforms and experts that provide language data annotation services. Data containing the communication between humans

through handwritten text or languages are annotated with attached metadata and documents. It is all about the configuration of data in a way that is recognizable by machines. This technique is mainly used for machine learning and artificial intelligence. By labelling and annotating training data, machine learning models are trained to perform specific functions. Several outsourcing companies address the high demand for data annotators and provide language data annotation resources and experts that best suit the requirement.

- **Video annotation:** Like image annotation, video annotation usually requires adding bounding boxes, polygons, or critical cases to content (Fig. 3.3). That on a frame basis, those frames then joined together to help track the flow of the annotated movement or in the video itself using a video annotation tool.

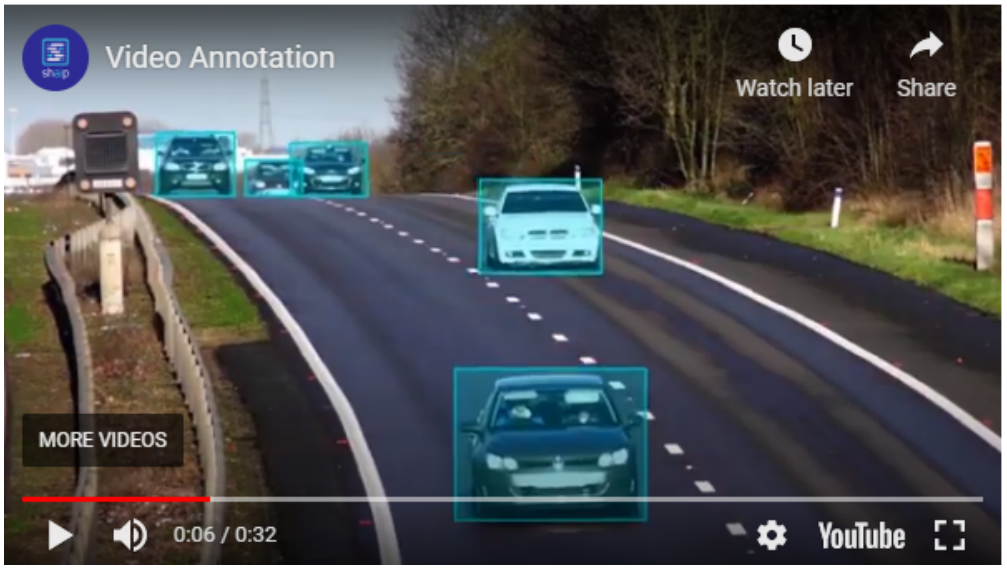

FIGURE 3.3: Video Annotation.

- **Audio annotation:** It enables the user to listen to files and, using the audio waveform image, select a subpart of a sound to annotate it. These libraries can be embedded into any web page, making it particularly easy for a developer to add the annotation feature to his web application.

3.3.3.1 Different Kind of Annotation Tools

- YOLO
- PascalVOC
- labelme
- CVAT
- doccano
- Visual Object Tagging Tool
- NeuroNER
- Universal data tool
- COCO annotator
- PolygonRNN++
- YEDDA
- Scalable
- LabeID
- 6D annotator
- Labelbox[7]

We have applied the basic doccano to perform the text annotation. doccano: In doccano annotation format, a .jsonl file with the same name is created for a disease which may contain multiple text files. Each .jsonl file contains the annotations for the corresponding text file, that is starting index, end index, token id, entity and relation between the entities.

[7]Annotation tools: https://www.datasetlist.com/tools/

3.3.4 Applications of Data Annotation

There are some advantages of Data Annotation:

- Labelling data requires a lot of hand-operated work. It also helps Unsupervised learning to solve the problem by learning the data and classifying it without any labels.
- Can add the labels after the data should continue classified, which is much easier.
- It is beneficial in finding patterns in data, which cannot find using usual methods.
- It is perfect for data scientists because the unsupervised method can help understand raw data and use them with machine learning algorithms.
- Data annotation directly helps the machine learning approaches to get trained by a supervised learning method correctly for accurate prediction.
- They are sort of performance that learn from annotated data and recognize related patterns in new datasets.

3.3.5 Data Annotation Structure

The essential components of any annotation can be roughly divided into three main elements: a body, an anchor, and a marker. The body of an annotation involves reader-generated logos and writing, such as handwriting analysis or leads in the border. The anchor is what indicates the extent of the original text to the main body of the annotation refers. It may include circles around sections, brackets, highlights, underlines, etc. Annotations may be anchored to vast stretches of text (such as an entire document) or very narrow sections (such as a specific letter, word, or phrase). The marker is the visual appearance of the anchor, such as whether it is a grey underline or a yellow highlight. Annotation with a body (such as a comment in the margin) but no specific anchor has no marker.

The Data annotation building an AI or ML model that acts like a human requires large volumes of training data. For a model to make decisions and take action, it must be trained to understand specific information with high-quality, human-powered data annotation; partnerships can develop and update AI implementations.

3.4 EN-BioDNER: Data Annotation Process

In this part, we will describe our data annotation approach in details. Fig. 3.4 illustrates the process of `BN-HTRd` dataset annotation followed by the writers and annotators while doing the annotation.

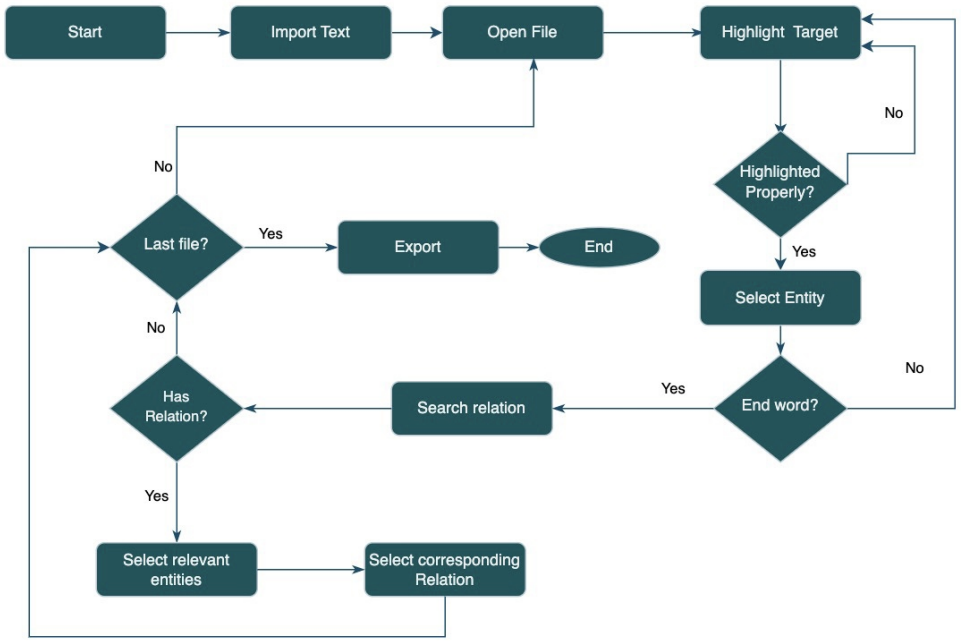

FIGURE 3.4: Flow Chart of the Annotation Process.

3.4.1 Data Source and Collection

Annotation processes are basically populating a dataset by observing something in the world and then recording the characteristics of the thing being observed. The dataset is essentially organized into a certain model that helps to process the needed information. As a first step, We have collected individual text documents from `National Health Portal`[8] as our ground truth data by automatically Crawling/Scraping the website. We mainly preferred this source for our dataset because the NHP does not require any restrictions and has an open access policy to their data for the general public. In most

[8]Corpus (National Health Portal): https://nhp.gov.in/

cases, we crawl through sites and retrieve `TEXT` files for a particular disease. The `TEXT` files were named according to their `ICD11`[9] disease code like Fig. 3.5.

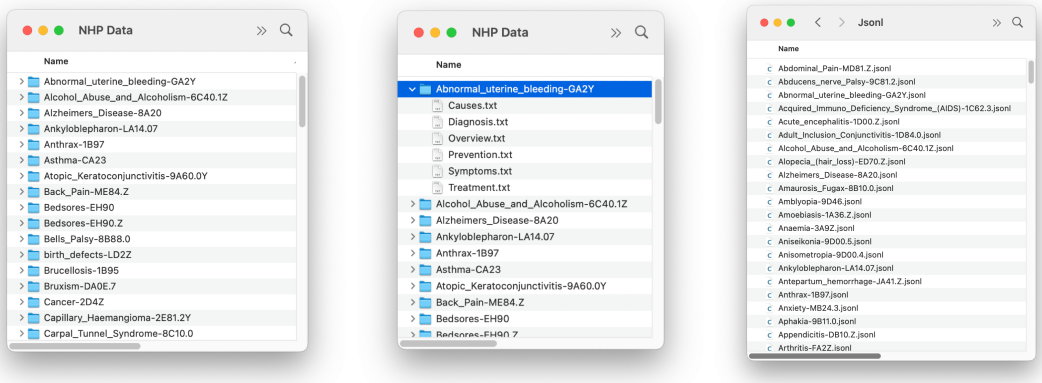

FIGURE 3.5: Pre-processing of NHP Data.

3.4.2 Annotation Guidelines

In order to Named Entity Recognition and Relation Extraction, we present comprehensive bio-medical dataset which are individual disease information. As the initial process were gathered 300 diseases information, each disease data was collected before annotating and ensured that each disease has Punctuation corrections, Integration of abbreviations, Bullet points processed and Disease Codes Integration. Thus, we have followed the following guidelines:

- Initializes the Entity labels and Relation labels.
- We have developed a schema for annotation purpose.
- We have marked a word and then choose it's label.
- If there was a dependency between two entities, then we selected the entities and then add them into a relation available from the relation label list.
- While choosing the relation we have checked it's validity from the schema we have created.

[9]ICD (International Classification of Diseases 11th edition): https://icd.who.int/ct11/icd11_mms/en/release

3.4.3 Labeling

There required two types of labeling in annotation. One is Entity labeling and another is relation labeling (entity linking).

3.4.3.1 Entity Labeling

- **Entities are labeled by : Selecting one or multiple words**

  Verification: Start annotating the text data by selecting the appropriate label for each entity or category in the text. Doccano provides an intuitive interface where we can highlight and label the relevant portions of the text. We repeated this process for each document in the dataset. After annotating a significant portion of the data, review and correct any mistakes or inconsistencies in the annotations. Doccano allows us to modify or delete annotations as needed. This step is crucial to ensure the quality and accuracy of the annotations. Annotation is an iterative process, and we may need to repeat the steps above multiple times to improve the quality and coverage of the annotations.

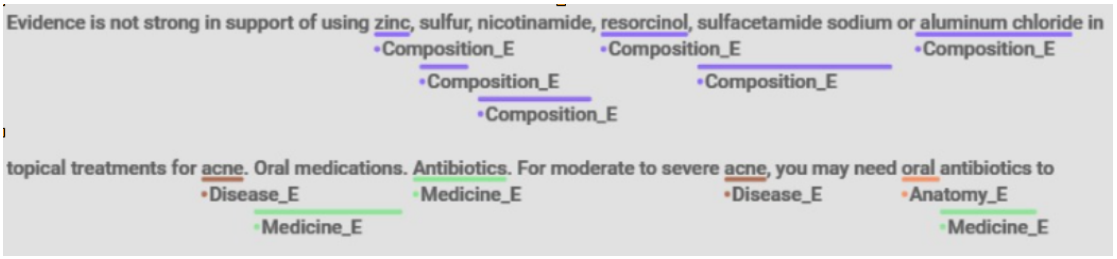

FIGURE 3.6: Entity Labeling.

3.4.3.2 Relationship Labeling

- **Relations are labeled by : Selecting two entities and their relation**

  Verification: After annotating the entities we have made relations between the entities (if any). First we select first entity and then second. Order of selecting the entities also defines the direction of the relation. After selecting two entities we can choose the relation between them.

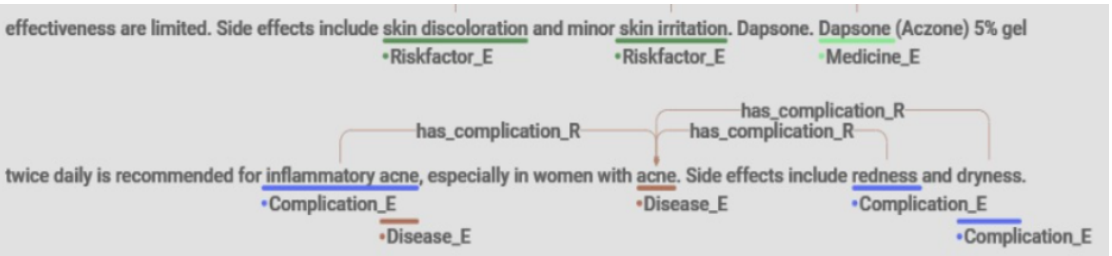

FIGURE 3.7: Relation Labeling

3.5 Dataset

3.5.1 Statistics of the Dataset

The dataset statistics originate from actual observations gathered by sampling statistical annotated data. The EN-BioDNER dataset contains information on 301 diseases, with each disease comprising multiple files. These files include essential details such as an overview of the disease, symptoms, causes, risk factors, complications, diagnosis criteria, precautions, and treatment options. In total, the dataset consists of 1804 files, covering a wide range of diseases and their associated information.

As per our schema, we have 11 named entities such as Anatomy, Cause, Complication, Composition, Diagnosis, Disease, Medicine, Precaution, Riskfactor, Surgery, and Symptom. Also, our schema supports the relationship between two entities such as affects, caused by, diagnosis on, has complication, has diagnosis, has precaution, has risk factor, has side effect, has symptom, influence, made with, needs, prescribed for, surgery for, surgery on.

By calculating their entity frequencies, we can list them like Table 3.8.

Named Entity	Count
Disease	6.4k
Complication	3k
Symptom	2.8k
Medicine	1.9k
Cause	2.4k
Diagnosis	1.6k
Anatomy	1.8k
Precaution	1k
Riskfactor	900
Surgery	500
Composition	150

FIGURE 3.8: Count of Named entities.

By calculating their relation frequencies, we can list them like Table 3.9.

Relationship	Count
has_symptom	2.6k
caused_by	2.1k
prescribed_for	1.7k
has_complication	1.6k
has_diagnosis	1.4k
has_precaution	950
has_risk_factor	700
surgery_for	450
affects	400
diagnosis_on	200
influence	200
made_with	150
surgery_on	80
has_side_effect	50
needs	20

FIGURE 3.9: Count of Relations.

3.6 Challenges of Data Annotation

We have faced many challenges while executing such tasks, from classifying or categorizing text to annotating them 'stage by stage' for making them identifiable for machine learning or computer-vision-based models. We will explain the configuration decisions that we have met. We discuss the challenges focusing on primary data annotation points are bellowed:

- **Time-consumption:** The data annotation is quite time-consuming work. Preparing English Biomedical Disease text documents with annotation takes too much time and effort because manually annotating data is a lengthy process.

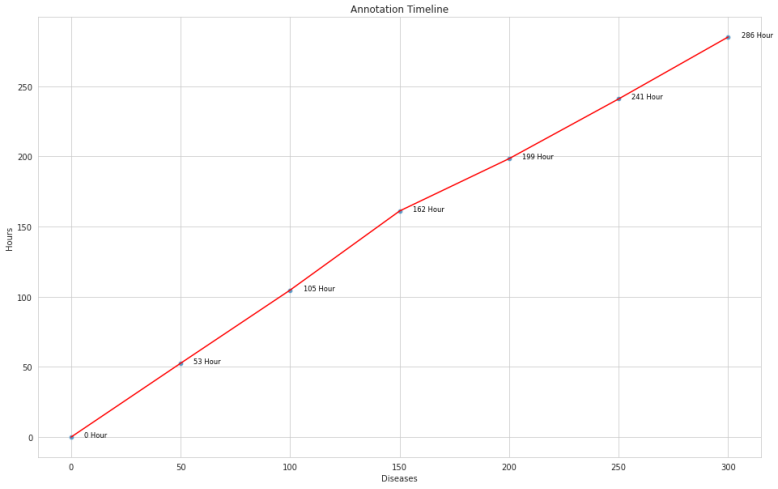

FIGURE 3.10: Hour vs Disease graph.

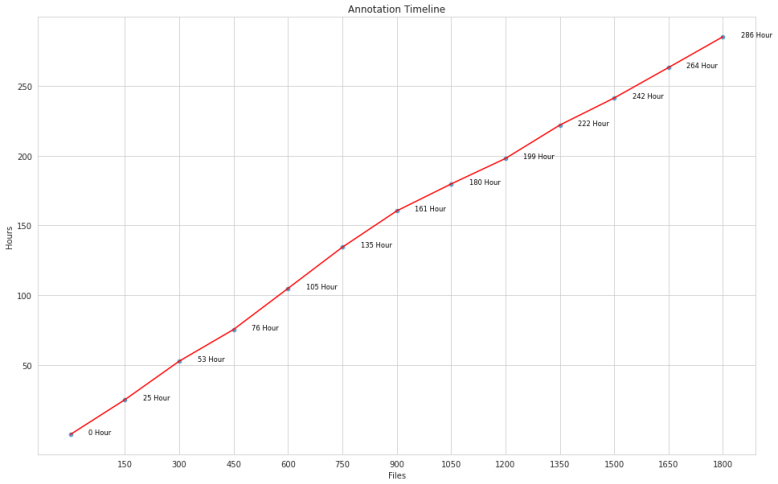

FIGURE 3.11: Hour vs Files graph.

- **Selecting the Right Tools and Techniques:** We were searching for a tool to annotate data manually and other tools for Named Entities and Relations to enable the users to generate automatically annotated data. To create high-quality training datasets sequence, choosing the right tools is very important for data annotation groups.

- **Consistent Quality and Data Tagging:** We have checked the text file quality to the point, corrected them, and validated the dataset to get better result in our system.

- **Data Security Guidelines:** Text Annotation is based on guidelines and training. Having detailed annotation guidelines is crucial for training annotators. Each annotator annotated the data individually from each others. The annotation requires annotators to develop detailed guidelines, which would be valuable when we will annotate new data.

Those challenges involve changing the annotation to specific needs of our analysis, training the participating annotators to have an approving opinion of the data, characteristics of the cases for the annotation. Worked as a starting position, we further study various information that has been slightly discussed in the related works and describe exciting future directions.

Chapter 4

Methodology and Design

In this section, we will describe our Named Entity Recognition and Relation Extraction approach. Fig. 4.2 reflects `EN-BioDNER` being processed and ready to be used in NER and RE system.

4.1 Data Preprocessing

Let's start with Data Collection & Prepossessing. This process is being obtained by data scrapping from `NHP` and `Mayoclinic`. After collecting significant amount of data, we stored those in proper directories to maintain a collaborative structure. We've created a schema diagram 4.1 to annotate stored data. To annotate those labels, we have used `doccano` for sequence to sequence labeling.

We have different labels for NER & RE system. Following entities are specified to label our stored text data:

- Disease
- Anatomy
- Cause
- Code
- Diagnosis
- Precaution
- Riskfactor
- Symptom
- Medicine
- Composition
- Complication
- Surgery

Following relations are specified to label our relations between the entities:

- affects
- caused by
- has code
- has diagnosis
- has precaution
- has risk factor
- has symptom
- made with
- prescribed for
- has complication
- influence
- has side effect
- surgery on
- diagnosis on
- needs
- surgery for

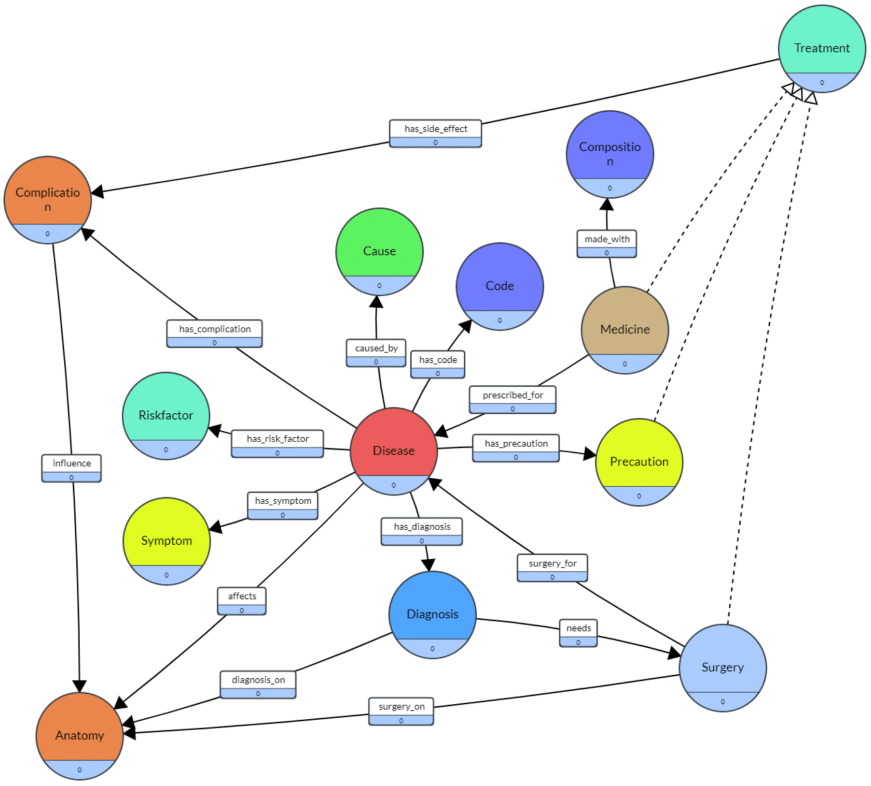

FIGURE 4.1: Schema Diagram

Annotated texts are basically stored in separated `Jsonl` files. Then we have merged those `Jsonl` files into one single `Json` object.

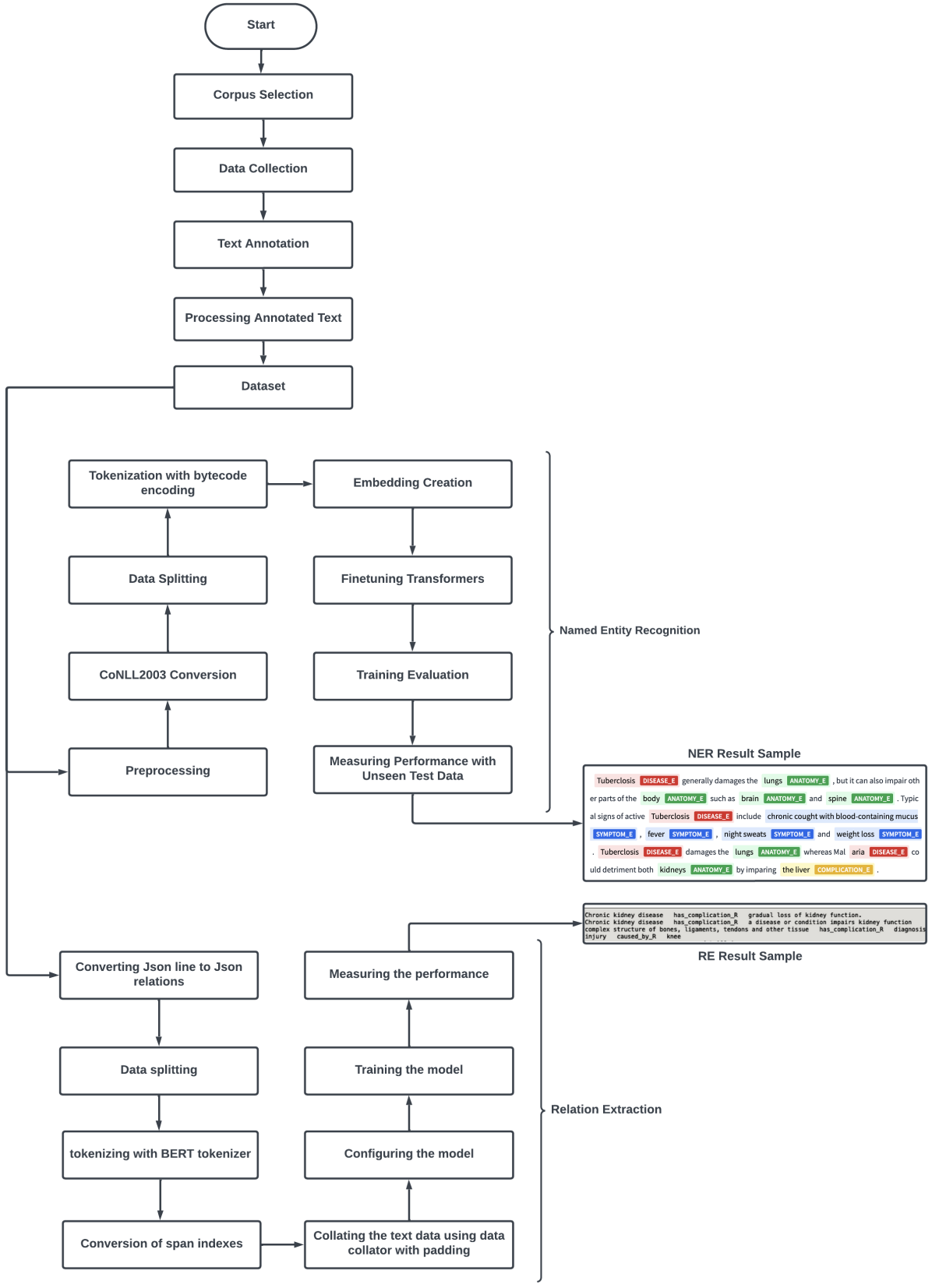

FIGURE 4.2: Overall System Architecture.

The system architecture for the NER and RE combined system consist of the following components:

4.2 Named Entity Recognition

1. **Data Preprocessing:**

   The data preprocessing stage involves cleaning and preparing the input data for the named entity recognition (NER) system. This includes removing any irrelevant information, handling noise, and normalizing the data. Preprocessing ensures that the data is in a suitable format for further processing and analysis.

2. **Convert Data to CoNLL 2003:**

   In order to train the NER model, the data needs to be transformed into the `CoNLL 2003` format. This format organizes the data into tokenized sentences with corresponding labels for named entity annotations. Each token is associated with its respective label, indicating whether it belongs to a named entity and, if so, the entity type.

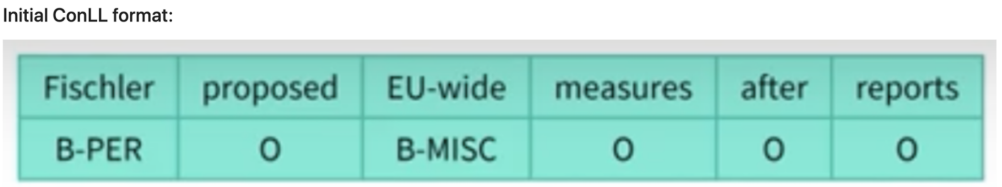

FIGURE 4.3: CoNLL 2003 format (byte pair encoding)

3. **Data Splitting into Train, Validation, and Test:**

   The annotated data is divided into three sets: train, validation, and test. The train set is used for model training, the validation set is used for hyperparameter tuning and model selection, and the test set is used to evaluate the final performance of the NER system. Proper data splitting ensures unbiased evaluation and helps prevent overfitting.

4. **Tokenize with HF Autotokenizer (Byte pair Encoding):**

   The tokenization process involves splitting the text into individual tokens or subwords, which are the input units for the NER model. The Huggingface autotokenizer is used, which leverages byte pair encoding (BPE) or similar techniques

to tokenize the text efficiently. Byte pair encoding converts text into a numeric representation suitable for input into the transformer-based models.

`BERT` is capable of handle sequences up to 512 tokens long, and while the vast majority of `CoNLL-03`'s paragraphs are below that value some sequences are longer than 512 tokens. In addition the padding strategy used by [8] is the WordPiece Tokenizer. It splits the following string "Jim Henson was a puppeteer" into ['Jim', 'Hen', '##son', 'was', 'a', 'puppet', '##eer'].

After tokenization:

[CLS]	Fi	##sch	##ler	proposed	EU	-	wide	measures	after	reports	[SEP]
None	0	0	0	1	2	2	2	3	4	5	None
None	B-PER	I-PER	I-PER	O	B-MISC	I-MISC	I-MISC	O	O	O	None
-100	1	2	2	0	7	8	8	0	0	0	-100

FIGURE 4.4: Tokenization

This creates flexibility, as the tokenizer can always create tokens for a given sequence, regardless if the word has been seen previously by the model. This is especially useful for NER as some names may be very unusual and not occur in the training dataset. Fixed input models like BERT have some issues with varying length sequences, and the `WordPiece Tokenizer` may extend the sequence length above 512 tokens. Randomly splitting sentences does not work well for NER, and [8] does not specify how they handle longer sequences. If a data point, after tokenization, is longer than 510 tokens (to fit `[CLS]` and `[SEP]` tokens required by `BERT` described by [8]) the pre-processor will try to split data points on sentences. Each individual slice may be 510 tokens or less, but should always constitute grammatically correct and complete sentences. If no suitable sentence could be found the number of possible splitting tokens are extended to ['.', '-', ')', '/', ',', '1', '2', '3', '4', '5', '6', '7', '8', '9'] and the pre-processor will again try to find a suitable data point length.

5. **Embedding/Padding the Data:**

   To process the tokenized text, it needs to be converted into fixed-length vectors. This is achieved through embedding, where each token is represented by a dense

vector that captures its semantic meaning. Additionally, padding is applied to ensure all sequences have the same length, enabling batch processing during training and inference.

6. **Finetuning Specific Transformer Model:**

   The pre-trained transformer model, such as BERT or other variants, is fine-tuned on the NER task using the annotated data. Fine-tuning involves training the model with the tokenized and embedded data, optimizing the model's parameters to improve its performance on the NER task specifically.

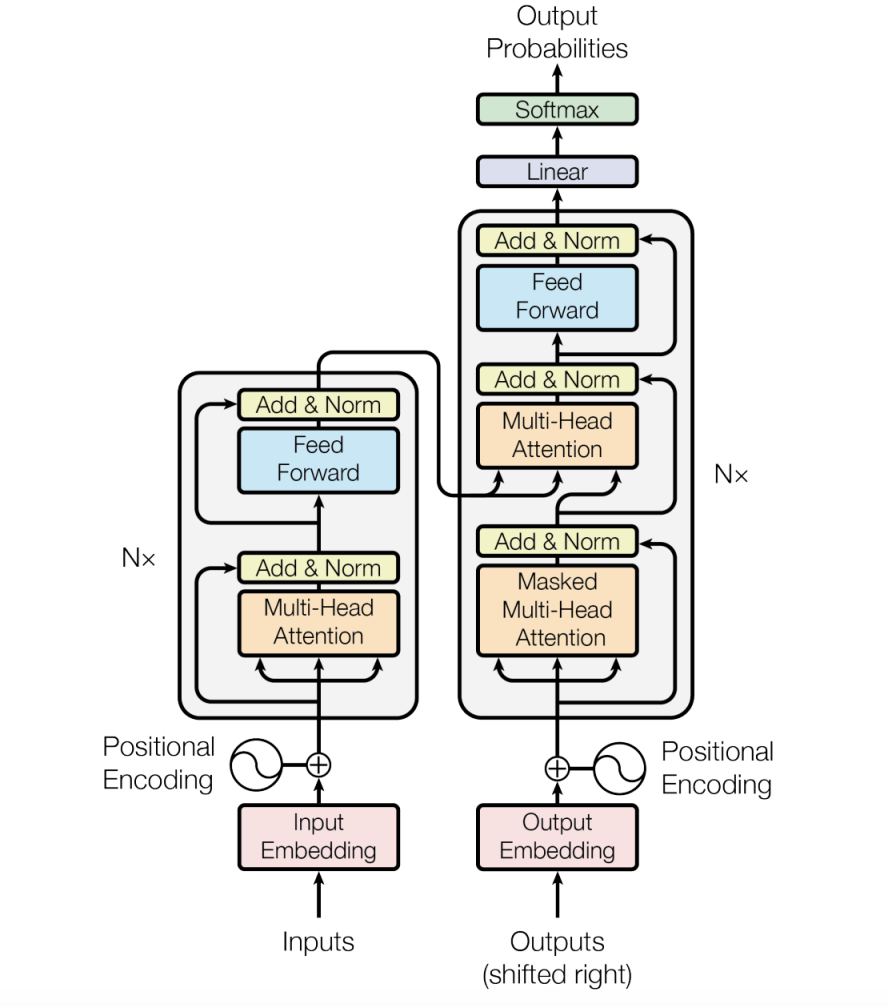

FIGURE 4.5: BERT Architecture

7. **Training the Model:**

   The performance of the NER system is evaluated using the test data. Metrics such as precision, recall, and F1-score are commonly used to assess the model's

ability to correctly identify named entities. These metrics provide insights into the system's accuracy and its ability to generalize to unseen data.

These components collectively form a named entity recognition system, starting from data preprocessing and conversion to `CoNLL 2003` format, through model training and evaluation, and finally to the deployment of the system for entity recognition using the Hugging Face token classification pipeline.

4.3 Relation Extraction

1. **Converting Json line to Json relations:**

   This step involves converting the input data in JSON format, which may be in a single line, into JSON relations. Which means parsing the JSON data, extracting relevant information, and structuring it in a way that represents relationships between different entities.

FIGURE 4.6: Json line to json relation

2. **Data Splitting:**

   This task involves splitting the dataset into different subsets, such as training, validation, and testing sets. This division is essential to evaluate the performance of our model on unseen data and prevent overfitting.

3. **Tokenizing with BERT tokenizer:**

   BERT (Bidirectional Encoder Representations from Transformers) is a popular pre-trained language model that requires input data to be tokenized into smaller units. In this task, we have used BERT tokenizer to break down the text into

tokens, which could be individual words, subwords, or characters, depending on the tokenizer's settings.

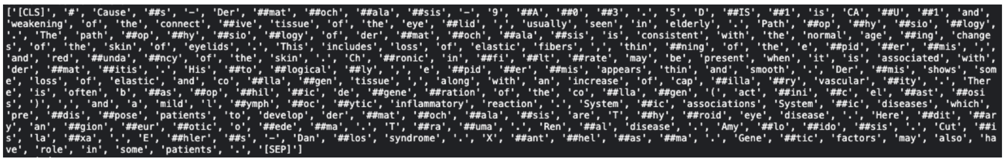

FIGURE 4.7: Tokenized data

4. **Conversion of Span indexes:**

   Relation extraction often involves identifying the positions or spans of entities or phrases within the tokenized text. In this step, we have converted these spans into indexes that correspond to the tokens in the tokenized sequence. This conversion allows us to locate and extract the relevant information easily during model training and evaluation.

FIGURE 4.8: Json line to json relation

5. **Collating the text data using data collator with padding:**

   Collating the data involves organizing and combining multiple samples into batches, which can be processed efficiently during training. In this case, we have used a data collator that not only groups the samples but also applies padding to ensure that all samples within a batch have the same length. Padding is necessary because inputs to the BERT model must have consistent dimensions.

6. **Configuring the BERT model:**

   Configuring the BERT model involves setting up the architecture and parameters of the BERT model for relation extraction. It includes specifying the number of layers, hidden units, attention heads, and other hyperparameters. This step prepares the model for training and subsequent tasks.

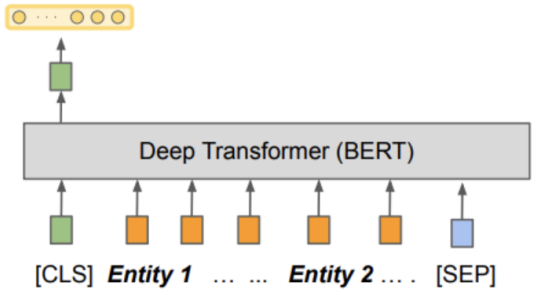

FIGURE 4.9: Standard - [CLS]

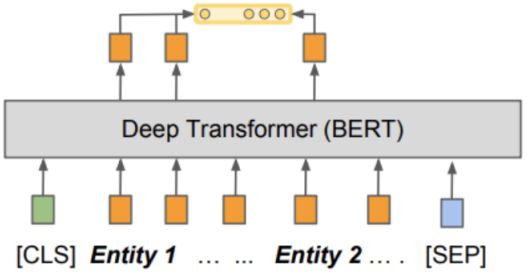

FIGURE 4.10: Standard - Mention Pooling

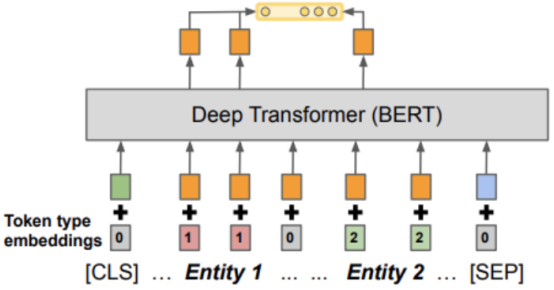

FIGURE 4.11: Positional EMB - Mention Pool

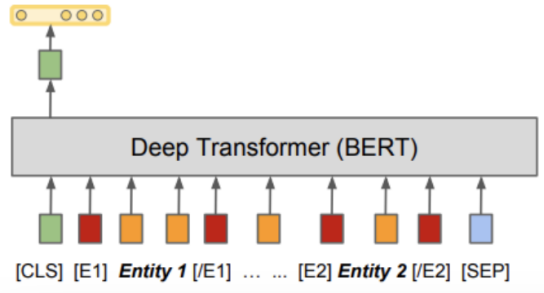

FIGURE 4.12: Entity Markers - [CLS]

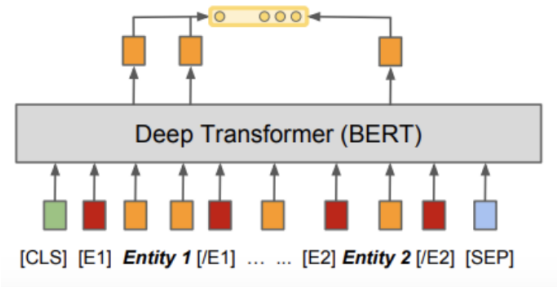

FIGURE 4.13: Entity Markers - Mention Pool

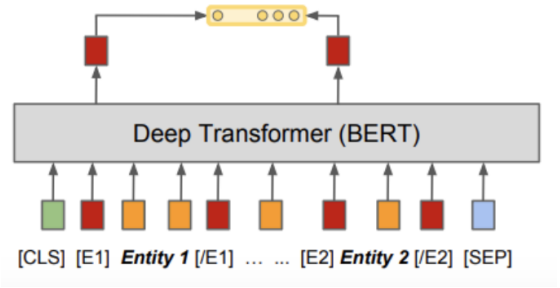

FIGURE 4.14: Entity Markers - Entity Start

7. **Training the model:**

   This task involves training the BERT model on your prepared dataset. During training, the model learns to predict the relationships between entities based on the input data and the provided labels or annotations. The model is typically trained using techniques like backpropagation and gradient descent to optimize its parameters and minimize the loss function.

8. **Running the model:**

   After training the model, it is crucial to evaluate its performance on unseen data to assess its effectiveness. This step involves using evaluation metrics such as precision, recall, F1 score, or accuracy to measure how well the model predicts the relationships between entities. The performance metrics help you understand the model's strengths and weaknesses and make any necessary improvements.

Chapter 5

Results and Evaluation

In this chapter we highlight the experiments conducted throughout this research, each experiment contains the hypothesis, observations, and results. These experiments use the Hugging Face BERT model which can be finetune using pytorch. We run these experiments on Amazon Sagemaker compute cloud and Google cloud engine. The machine is equipped with high frequency Intel Xeon E5-2686 v4 (Broadwell) processors 2ith 16 vCPUs and a Tesla V100 GPU with 5,120 CUDA Cores and 640 Tensor Cores.

5.1 Evaluation Metrics

The **Recall** is the measure of our model correctly identifying True Positives.

$$Recall = \frac{TruePositive(TP)}{TruePositive(TP) + FalseNegative(FN)} \tag{5.1}$$

Precision estimates how accurate your predictions are. i.e. the percentage of your predictions is accurate.

$$Precision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \tag{5.2}$$

F1 score estimates how many times a model made a correct prediction across the entire dataset.

$$F1 = \frac{TruePositive(TP)}{TruePositive(TP) + \frac{1}{2}(FalsePositive(FP) + FalseNegative(FN))} \tag{5.3}$$

True Positive (TP) is an outcome (result) where the model accurately predicts the positive class. Similarly, **True Negative (TN)** is an outcome where the model accurately predicts the negative class. **False Positive (FP)** is an outcome where the model incorrectly predicts the positive class. And **False Negative (FN)** is an outcome where the model incorrectly predicts the negative class.

5.2 Named Entity Recognition Results

After preparing the training data from the `EN-BioDNER` dataset we use `BERT-base-cased` and `RoBERTa` model to train according to their optimal parameters. We've found `RoBERTa` finetuned on our `EN-BioDNER` performs significantly better.

The following table shows the results of this experiment:

Table 5.1: NER Classification Report (Validation Data).

Entities	**Precision**	**Recall**	**f1-score**
Anatomy_E	0.375276	0.555556	0.447958
Cause_E	0.331384	0.46832	0.388128
Complication_E	0.312189	0.473585	0.376312
Composition_E	0.013432	0.02112	0.08263
Diagnosis_E	0.557692	0.649254	0.60001
Disease_E	0.75857	0.895009	0.821161
Medicine_E	0.529833	0.636103	0.578125
Precaution_E	0.361111	0.435754	0.394937
Riskfactor_E	0.251232	0.305389	0.275676
Surgery_E	0.642105	0.73494	0.685393
Symptom_E	0.645299	0.803191	0.71564

Table 5.2: NER Classification Report (Test Data).

Entities	**Precision**	**Recall**	**f1-score**
Anatomy_E	0.408027	0.373089	0.389776
Cause_E	0.162791	0.446809	0.238636
Complication_E	0.376	0.303226	0.335714
Composition_E	0.428571	0.384615	0.405405
Diagnosis_E	0.586207	0.664063	0.622711
Disease_E	0.695444	0.814607	0.750323
Medicine_E	0.459716	0.419913	0.438914
Precaution_E	0.342342	0.59375	0.434286
Riskfactor_E	0.484848	0.387097	0.430493
Surgery_E	0.556962	0.637681	0.594595
Symptom_E	0.51462	0.656716	0.577049

5.3 Relation Extraction Results

Similarly in Relation Extraction we use `BERT-base-cased` and `BioBERT` model to train according to their optimal parameters. We've found `BioBERT` finetuned on our `EN-BioDNER` performs significantly better. The following table shows the results of this experiment:

Table 5.3: RE Classification Report (Validation Data).

Relations	**Recall**	**Precision**	**f1-score**
affects_R	0.71	0.63	0.67
caused_by_R	0.96	0.95	0.95
diagnosis_on_R	0.40	0.33	0.36
has_complication_R	0.88	0.88	0.88
has_diagnosis_R	0.93	0.92	0.92
has_precaution_R	0.95	1.00	0.98
has_risk_factor_R	0.94	0.92	0.93
has_side_effect_R	0.00	0.00	0.00
has_symptom_R	0.97	0.98	0.97
influence_R	0.36	0.62	0.46
made_with_R	1.00	0.14	0.25
prescribed_for_R	0.84	0.95	0.89
surgery_for_R	0.71	0.44	0.54
surgery_on_R	0.00	0.00	0.00

Table 5.4: RE Classification Report (Test Data).

Relations	**Recall**	**Precision**	**f1-score**
affects_R	0.57	0.55	0.56
caused_by_R	0.94	0.91	0.92
diagnosis_on_R	0.57	0.81	0.67
has_complication_R	0.82	0.98	0.89
has_diagnosis_R	0.92	0.87	0.90
has_precaution_R	0.97	0.98	0.98
has_risk_factor_R	0.97	0.85	0.90
has_side_effect_R	0.00	0.00	0.00
has_symptom_R	0.98	0.98	0.98
influence_R	0.40	0.31	0.35
made_with_R	0.33	0.08	0.13
prescribed_for_R	0.90	0.95	0.93
surgery_for_R	0.62	0.40	0.49
surgery_on_R	0.00	0.00	0.00

After using several BERT models, the performance of the model **BioBERT** :

Table 5.5: RE Overall Performance (Validation Data).

No. of Disease	**Precision**	**Recall**	**f1-score**
50	0.81	0.85	0.82
100	0.77	0.80	0.77
150	0.75	0.78	0.76
200	0.80	0.82	0.82
250	0.83	0.85	0.83
300	0.83	0.85	0.84

Table 5.6: RE Overall Performance (Test Data).

No. of Disease	**Precision**	**Recall**	**f1-score**
50	0.54	0.58	0.55
100	0.77	0.67	0.67
150	0.67	0.62	0.64
200	0.68	0.63	0.62
250	0.64	0.63	0.63
300	0.64	0.61	0.62

5.4 Limitations

1. Our system cannot extract all kind of biomedical diseases, due to fine-tuning on pretty small dataset which doesn't contains all diseases as per ICD.
2. Our system is unable to detect entities and relation between the entities at the same time.
3. This system cannot extract overlapping entities.
4. This system cannot extract relations between more than two entities at the same time.

Chapter 6

Future Directions and Conclusion

Throughout the development of the investigation and taking into account the limitation of the same, some lines can be stablished for the improvement in further developments. The following points are consider in order to improve the proposed system:

- Extend the NER analysis beyond the intra sentence context by combining the proposed solutions with rule-based analysis in order to interpret semantics using POS tagging. This will make possible to extract relations between entities in different sentences of a corpus by analyzing the semantics of the same.

- Improve the existing dataset by annotating more entities and relations, allowing increase the training information. This will result in more training instances and therefore re-training the models of the system will increase the performing scores.

- Augment the Relation Extraction task by adding more biomedical related entities to the analysis. This improve will allow the system to increase the range of relations extracted. The proposed solution performs Relation Extraction between biomedical entity types, a new combination of related entities could be added.

- Add the capacity to the developed system of extracting the entities in different formats. Currently, the proposed solution allows to view the detected relations on a text; in the future download formats could be added like Pdf, XML, allowing the creation of graph-based data.

Our dataset is ready to deal with these objectives. We look forward to the research community around the globe who will use this dataset to achieve:

"`Sequence-to-sequence Named entity recognition and Relation extraction of biomedical data`"

Bibliography

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

[2] Aitor Gonzalez-Agirre, Montserrat Marimon, Ander Intxaurrondo, Obdulia Rabal, Marta Villegas, and Martin Krallinger. (2019). *PharmaCoNER: Pharmacological substances, compounds and proteins named entity recognition track.* In Proceedings of the BioNLP Open Shared Tasks (BioNLP-OST), Hong Kong, China. Association for Computational Linguistics.

[3] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. *Adversarial training for multi-context joint entity and relation extraction.* In Empirical Methods in Natural Language Processing (EMNLP), pages 2830–2836. (2018)

[4] Alsentzer, E., et al.: *Publicly available clinical BERT embeddings.* In: Proceedings of the 2nd Clinical Natural Language Processing Workshop, pp. 72–78 (2019)

[5] Oliver Bender, Franz Josef Och, and Hermann Ney. (2003). *Maximum Entropy Models for Named Entity Recognition.* In Proceedings of CoNLL-2003.

[6] Y. H. Liu, M. Ott, N. Goyal et al. *RoBERTa: a robustly optimized BERT pretraining approach.* 2019.

[7] P. D. Soomro, S. Kumar, A. A. Banbhrani, A. A. Shaikh, and H. Raj. *Bio-NER: biomedical named entity recognition using rule-based and statistical learners.* International Journal of Advanced Computer Science and Applications, vol. 8, no. 12, pp. 163–170, (2017).

[8] Jana Strakova, Milan Straka, and Jan Hajic. (2019). *Neural architectures for nested NER through linearization.* In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5326–5331, Florence, Italy, July. Association for Computational Linguistics.

[9] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. *BioBERT: a pre-trained biomedical language representation model for biomedical text mining.* Bioinformatics 2020 Feb 15;36(4):1234-1240

[10] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer. *Neural Architectures for Named Entity Recognition.* in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics, 2016, pp. 260–270.

[11] Y. Peng, S. Yan, Z. Lu. *Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets.* in: Proceedings of the 18th BioNLP Workshop and Shared Task, 2019, pp. 58–65.

[12] WangY. et al. *Supervised methods for symptom name recognition in free-text clinical records of traditional Chinese medicine: An empirical study.* J. Biomed. Inform. (2014)

[13] ZhuF. et al. *Biomedical text mining and its applications in cancer research.* J. Biomed. Inform. (2013)

[14] Kalpit Dixit and Yaser Al-Onaizan. *Spanlevel model for relation extraction.* In Association for Computational Linguistics (ACL) (2019), pages 5308– 5314.

[15] Agichtein, E., Gravano, L. *Snowball: Extracting relations from large plain-text collections.* Proceedings of the Fifth ACM International Conference on Digital Libraries. (2000).

[16] Zelenko D., Aone C., Richardella A. *Kernel methods for relation extraction.* Journal of Machine Learning Research. (2003)

[17] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. *Distant supervision for relation extraction* via piecewise convolutional neural networks. In EMNLP. pages 1753–1762. (2015)

[18] Duyu Tang, Bing Qin, and Ting Liu. *Document modeling with gated recurrent neural network for sentiment classification.* In EMNLP. pages 1422– 1432. (2015)