UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Automated Clinical Coding of Medical Notes into the SNOMED CT Medical Terminology Structuring System

Author:
Sergi CANTÓN SIMÓ

Supervisors:
Lauro SUMOY VAN DYCK
Laura IGUAL MUÑOZ

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

September 1, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Automated Clinical Coding of Medical Notes into the SNOMED CT Medical Terminology Structuring System

by Sergi CANTÓN SIMÓ

Automated clinical coding is the computational process of annotating healthcare free-text data by detecting relevant medical concepts and linking them to a structured medical terminology system. One of the most significant of these systems is SNOMED CT, which contains a vast array of specific medical terms, each identified by a unique ID. This work focuses on the automatic clinical coding of medical notes within the SNOMED CT system.

The study presents a comprehensive review of state-of-the-art methods in this field, followed by a detailed examination of two specific approaches, each tested and their results discussed. The first method employs a classical dictionary-based approach, while the second utilizes a deep learning BERT-based model. Additionally, the work introduces a novel contribution to one of these methods and demonstrates a practical application where automatic clinical coding facilitates the extraction of specific numerical values from medical discharge summaries.

Acknowledgements

I would like to express my deepest gratitude to all those who contributed to the successful completion of this work.

Firstly, I would like to thank Lauro Sumoy, my tutor at the Institut Germans Trias i Pujol, for his guidance and help throughout this project. His expertise and advice have greatly contributed to the quality of this work.

I am also thankful to Laura Igual, my tutor from the University of Barcelona, without whom this project would not have been possible.

I am deeply appreciative of the Institut Germans Trias i Pujol for providing the research environment and access to the supercomputer, which were crucial for for training some of the models.

A special thanks goes to Robert Benaigues, a researcher at the Institut Germans Trias i Pujol, for his assistance in running the models on the supercomputer cluster.

Lastly, I am deeply thankful to my family and friends for their constant support and encouragement throughout this journey. Their presence and belief in me have been a constant source of strength.

Contents

Chapter 1

Introduction

1.1 Medical Terminology Structuring Systems

Today, medical data is often generated by healthcare professionals and stored as free-text, making it challenging to extract valuable information. However, it is crucial to structure this data to, for example, maintain accurate electronic medical records.

To classify easily specific medical terms, healthcare organizations have created different systems to assign a code to each concept. One of them is SNOMED CT, or Systematized Nomenclature of Medicine Clinical Terms. It is a meticulously organized and computer-processable system of medical terminology that includes a vast array of codes, terms, synonyms, and definitions used in clinical documentation and reporting (U.S. National Library of Medicine, 2024). It is recognized globally as one of the most comprehensive multilingual clinical healthcare terminologies. SNOMED CT's primary role is to encode health information, supporting precise and efficient clinical data recording to enhance patient care (Gaudet-Blavignac et al., 2021). This system forms the core terminology for electronic health records, covering areas such as clinical findings, symptoms, diagnoses, procedures, body structures, organisms, causes, substances, pharmaceuticals, medical devices, and specimens.

In short, SNOMED CT identifiers can be used to classify and structure a vast amount of medical concepts. In the online browser of SNOMED International, 2024, one can search for a term, using either its code or its name, in order to get all its information. An example can be seen in Fig. 1.1.

SNOMED CT has demonstrated effectiveness in various applications, such as clinical decision support systems for preventive care (Al-Hablani, 2017). The migration of existing electronic health records (EHRs) to the SNOMED CT coding system has been proven successful, optimizing clinical use and enhancing decision support. This transition was well received by healthcare professionals and quickly adopted across all specialties at University Hospitals Birmingham (UHB) (Pankhurst et al., 2021).

In addition to SNOMED CT, several other medical terminology systems play crucial roles in structuring and coding medical data.

One such system is the International Classification of Diseases (ICD), specialized in the classification and coding of diseases and related health conditions. ICD is used globally for health statistics, billing and epidemiology. It provides a standardized language for reporting and monitoring diseases, allowing healthcare providers to communicate diagnoses consistently. The most recent version, ICD-11, reflects the latest medical knowledge and practices, offering a detailed framework for categorizing health conditions (World Health Organization, 2019).

Another key system is the Unified Medical Language System (UMLS), which integrates various health and biomedical vocabularies into a unified framework.

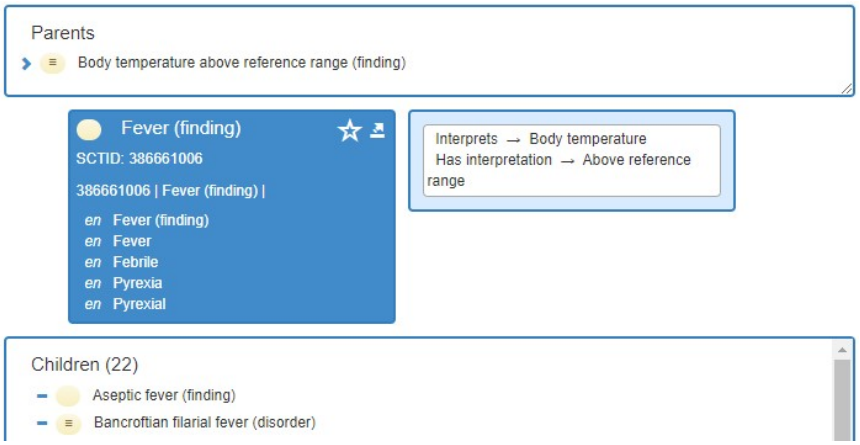

FIGURE 1.1: Example of search for the SNOMED CT identificator (SCTID) of "386661006". The browser shows that the corresponding concept is "fever", which is a "finding". Also, there are 22 children concepts, that is, terms that inherit or that are a subtype of "fever". For instance, the first one, "aseptic fever", also a "finding".

UMLS provides tools and resources to facilitate the linking of different terminology systems, ensuring interoperability and consistent data exchange across various healthcare platforms. It includes a broad range of medical concepts and terms, allowing for comprehensive data integration and enhanced semantic understanding (Bodenreider, 2004).

Current Procedural Terminology (CPT), maintained by the American Medical Association, describes medical, surgical, and diagnostic procedures and services. CPT codes are essential for billing and insurance reimbursement in the United States, ensuring that healthcare providers are compensated accurately for the services they deliver. Each CPT code corresponds to a specific medical procedure, facilitating clear communication between healthcare providers and payers (American Medical Association, 2024).

LOINC (Logical Observation Identifiers Names and Codes) is another important terminology system, primarily used for identifying laboratory and clinical observations. LOINC codes standardize the way lab tests and clinical measurements are reported, enabling the consistent exchange and interpretation of data across different healthcare systems. This system is vital for ensuring that test results are comparable, regardless of where or how they were obtained (Regenstrief Institute, 2024).

In the realm of pharmaceuticals, RxNorm provides standardized names for clinical drugs and links these names to various drug vocabularies used in pharmacy management and drug interaction software. RxNorm ensures that medications are accurately identified and that information about them is consistently communicated across different systems (Nelson et al., 2011).

For broader healthcare coverage, the Healthcare Common Procedure Coding System (HCPCS) is used, especially in the U.S., for coding products, supplies, and services not covered by CPT codes. This includes everything from durable medical equipment to ambulance services. HCPCS codes are integral to the billing process in Medicare and Medicaid, as well as in other healthcare programs (Centers for Medicare & Medicaid Services, 2024).

While these systems are essential for various aspects of healthcare, this work will focus on SNOMED CT due to its comprehensive nature and its pivotal role in clinical

documentation and electronic health records. SNOMED CT's extensive coverage of medical concepts makes it particularly well-suited for the tasks of classification and structuring of medical data, which are central to this study.

1.2 Motivation of Automatic Clinical Coding

To effectively use medical terminology systems like SNOMED CT to classify and structure concepts from free-text medical reports, it is essential to identify the relevant terms within the text and assign the appropriate code or identifier to each one.

Manually identifying relevant terms and assigning the correct codes from medical terminology systems is a complex and time-consuming task, especially given the vast amount of free-text data in medical reports. Due to the intricacies and scales involved, this process is challenging to perform by hand. Therefore, automated systems are necessary to efficiently detect and code these terms, ensuring accurate classification and structuring of medical concepts (Dong et al., 2022).

Automating the process of identifying and coding relevant terms from medical texts is far from trivial. The complexity of medical language, including nuances, abbreviations, and context-dependent meanings, makes it a challenging task for computers. However, in recent years, advancements in machine learning and deep learning have led to the development of effective methods for tackling this problem (Jiang et al., 2017). These technologies have significantly improved the ability to automatically extract and link medical concepts, offering promising solutions for managing the vast amounts of unstructured data in healthcare (Ji et al., 2022).

1.3 Automated Clinical Coding Steps

As mentioned, when employing machine or deep learning methods, usually this process consists of two main tasks: recognizing the relevant terms and linking them to their corresponding concept identifiers. In the medical field, this entire process is referred to as clinical entity linking or automated clinical coding.

More specifically, the two common steps are as follows:

- **First step:** The first task involves a NER problem. Named Entity Recognition (NER) is a process in natural language processing (NLP) that identifies and classifies key entities in text known as mentions, such as names of people, organizations, locations, dates, and other specific terms. In the context of biomedical texts, NER helps identify relevant terms such as medications, diseases, medical procedures, body structures, and more.

- **Second step:** The second task involves assigning a unique identifier to each recognized term, effectively linking each relevant mention to its corresponding concept within a predefined system, such as SNOMED CT. This process ensures that each identified term is accurately matched with a specific concept, enabling the transformation of unstructured text into structured data.

1.4 Objectives and Contents

This work focuses on both theoretical and practical research in the field of named entity recognition and linking of biomedical entities to specific terminologies or their

corresponding codes within medical terminology systems, with a particular emphasis on SNOMED CT. The specific objectives of this work can be outlined as follows:

- Gain a comprehensive understanding of the structure of the MIMIC-IV-Note dataset and how to extract information for automatic clinical coding.
- Understand the importance of SNOMED CT terminology, including its detailed functioning and potential applications.
- Provide an overview of the state-of-the-art methods in automatic clinical coding, with an emphasis on approaches that address the tasks of biomedical named entity recognition and entity linking separately.
- Study, explain in detail and test the two winning methods for entity recognition and linking from the "SNOMED CT Entity Linking Challenge" (Hardman et al., 2023a).
- Contribute to the methods by proposing, implementing, and testing modifications or improvements.
- Evaluate and discuss the results of these methods, including an analysis of their effectiveness, potential extra improvements and limitations.
- Analyze and illustrate examples of possible applications for automatic clinical coding.

This work is divided into six chapters. This first chapter has already provided an introduction to the concepts of medical terminology structuring systems and automated clinical coding, along with their importance. The second chapter offers a detailed explanation of the state-of-the-art methods in this field of research. The third chapter will then detail the necessary data required to run the methods explained in the fourth chapter. In the fourth chapter, these two methods will be explained in detail, along with the modifications and contributions made in this work. Following this, the fifth chapter will present an application of automated clinical coding, describe the experiments conducted with the selected methods and present the corresponding results. Finally, the sixth chapter will discuss these results and provide the overall conclusions of the work.

Chapter 2

State of the Art

In recent years, automatic clinical coding has gained significant attention in the fields of medicine, bioinformatics, and computational science. This surge in interest has led to the development of various techniques, ranging from classical rule-based or dictionary-based methods to more advanced approaches utilizing machine learning and deep learning. The rise of large language models like BERT in the past five years has further fueled research in entity recognition and linking, leading to the creation of powerful new models.

This chapter will first discuss notable named entity recognition methods for detecting concepts in biochemistry and biomedicine, then explore the most relevant entity linking techniques for biomedical concepts and, finally, examine methods that integrate both entity recognition and linking.

2.1 Named Entity Recognition in Biomedicine

As previously explained, named entity recognition (NER) involves identifying specific words or entities in free text that belong to a set of predefined categories. In this process, each term in the text is labeled according to its category, or it is marked as a general word if it does not fall within any of the relevant categories. In our area of study, these categories typically include medical, biological, or chemical terms, depending on the specific subject matter being addressed.

A common method used in named entity recognition is the B-I-O system, where each word in a text is classified with one of three labels: “B”, “I”, or “O”, followed by the category of the entity. The label “B” is used if the word is at the beginning of an entity, “I” if it is inside the entity, and “O” if it is not part of any entity. For example, in the phrase “The patient suffers from chronic pain”, the B-I-O system would tag “The”, “patient”, “suffers”, and “from” as O (indicating they are outside any entity). The word “chronic” would be tagged as B-FINDING, indicating the start of a medical finding, and “pain” would be tagged as I-FINDING, showing it is part of the same medical finding entity.

In the context of NER methods applied to bioscience or medical data, these approaches can be broadly categorized into machine learning and deep learning techniques. Machine learning methods rely on explicitly defined features, which are often engineered through data preprocessing. For instance, lexical features such as token text and capitalization can provide insights into the nature of the entities. Part-of-speech (POS) tags and morphological features, including prefixes and suffixes, help in understanding the grammatical and structural roles of words. Contextual features, such as surrounding words or phrases, further aid in disambiguating entities based on their context. Additionally, dictionary lookups to match terms with biological lexicons and entity co-occurrence patterns provide valuable information for accurate classification. In contrast, deep learning approaches can automatically

learn and extract relevant features from raw data, minimizing the need for manual feature engineering (Durango, Torres-Silva, and Orozco-Duque, 2023).

The review will first cover classical machine learning methods and their application in biomedical named entity recognition. Following this, deep learning methods will be examined, highlighting the solutions they provide to challenges within the field.

2.1.1 Machine Learning Approaches

Classical machine learning methods for NER, including those applied to bioscience or medical data, often utilize techniques such as maximum entropy (ME) models, support vector machines (SVMs), or conditional random fields (CRFs).

For instance, a maximum entropy-based model is used by Lin et al., 2004 to detect entities in text files from the GENIA corpus. The model classifies relevant words into various biological categories, such as "DNA", "RNA", "protein", and "cell". Additionally, dictionary-based and rule-based methods are applied during the post-processing stage to refine and enhance the accuracy of the classification.

A different process is used in the paper of Wang and Patrick, 2009, which primarily utilizes Conditional Random Fields (CRFs) for entity detection. To address potential misclassifications, the method further integrates an ensemble of a maximum entropy model and a support vector machine (SVM) for reclassification and refinement. Conditional Random Fields are also the base of the method explained in the work of Settles, 2004, this time to detect entities from "protein", "DNA", "RNA", "cell-line" and "cell-type", and used too by Liang et al., 2017 to recognize both "western drugs" and "traditional medicines" in Chinese free-text.

In the article of Zhou et al., 2004, PowerBioNE is introduced as an advanced named entity recognition system for the biomedical domain, also using data from the GENIA corpus. This system utilizes a Hidden Markov Model (HMM) combined with an HMM-based named entity recognizer. To address data sparsity issues, it incorporates a k-nearest neighbor (k-NN) algorithm. Additionally, a pattern-based post-processing technique is employed, which automatically extracts rules from the training data to further refine the entity recognition.

2.1.2 Deep Learning Approaches

In contrast to classical machine learning approaches, deep learning approaches rely on neural networks, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for processing and understanding complex text data (Li et al., 2020). In the field of natural language processing, recent advancements such as BERT models have brought about a substantial improvement in named entity recognition. BERT, or Bidirectional Encoder Representations from Transformers, significantly enhances the ability to understand context and relationships within text (Devlin, 2018). Specifically, the latest BERT-based models, such as BioBERT, which are pretrained on biomedical text, have represented a significant improvement in the field (Lee et al., 2020).

An example of a deep learning named entity recognition method utilizing classical neural networks can be found in the work of Wu et al., 2017. In this study, the detection of medical entities is explored using both a convolutional neural network and a recurrent neural network. These networks are trained with word embeddings generated from the MIMIC-II and i2b2 datasets, enabling the model to effectively recognize and classify medical terms within the free-text.

A more complex neural network structure is employed by Wu et al., 2015 for entity recognition in Chinese text. In this approach, a word embedding matrix is created from an annotated dataset of admissions to the Peking Union Medical College Hospital. Local features are extracted directly from the words, while global features are derived from the surrounding context using a convolutional neural network (CNN). These local and global features are then combined in a fully connected neural network, which predicts the probabilities of each entity class.

The method presented by Cho and Lee, 2019 combines Conditional Random Fields (CRFs) with a bi-directional long short-term memory network (BiLSTM) to identify biomedical entities. This approach, named Contextual Long Short-Term Memory Networks with CRF (CLSTM), leverages the strengths of both CRFs and BiLSTMs to capture context and sequence information more effectively. The CLSTM model is trained and tested on three prominent biomedical corpora: the Disease Corpus of the National Center for Biotechnology Information (NCBI), the BioCreative II Gene Mention Corpus (GM), and the BioCreative V Chemical Disease Relation Corpus (CDR). A similar approach, named BiLSTM-Att-CRF, is employed in the article of Li et al., 2019, where the attention mechanism is integrated into the basic BiLSTM-CRF model for entity recognition in Chinese medical text. To further enhance the model's performance, medical dictionaries and part-of-speech (POS) features are also incorporated, providing additional context and accuracy in identifying biomedical entities.

A BERT model is combined with the BilSTM-CRF model in the paper of Wang et al., 2023. First of all, a BERT model is fine-tuned with word embeddings from Chinese text. Then, the BilSTM-CRF module is utilized to extract further features from the encoded outputs of BERT in order to account for context information and improve the accuracy of semantic coding. Additionally, the pre-trained multilingual BERT model is demonstrated to be effective for Korean clinical entity recognition in the work of Kim and Lee, 2020.

2.2 Entity Linking

As previously discussed, entity linking is usually the second step in automatic clinical coding. After entities are identified and classified in the text, they need to be matched with their corresponding specific terms, each associated with a unique code. The specific terms and codes vary depending on the coding system used, such as SNOMED CT, ICD, or UMLS.

Automatic entity linking typically relies on machine learning or deep learning techniques. For instance, vector embeddings are often used to represent both the detected entities and the terms in the coding system. Similarity measures can then be applied to select the closest term for each entity. In some cases, instead of choosing a single term directly, a set of potential candidates is identified, and a subsequent re-ranking or post-processing step is used to select the most appropriate term. Additionally, tools like knowledge graphs or rule-based methods are frequently employed to improve the accuracy of entity linking.

Automatic entity linking has been relatively extensively explored within the ICD coding system for matching diagnoses and specific terms compared to other systems. The methods developed for ICD have been effective and are also applied to other coding systems such as SNOMED CT. Therefore, the review will first include a section dedicated to explaining some of the most relevant techniques for linking

entities to ICD codes. Following this, another section will cover methods for linking entities to other coding systems.

2.2.1 International Classification of Diseases

A common data structure used in ICD entity linking are knowledge graphs. Knowledge graphs are structured networks of interconnected entities and relationships that represent real-world information in a way that reflects human understanding. They organize concepts, such as diseases and symptoms, into nodes, and link them through edges that represent relationships like "is associated with" or "is a symptom of." When linking diseases to ICD codes, a knowledge graph helps by mapping disease-related terms from free text to relevant concepts within the graph. These connections guide the selection of the correct ICD code by providing contextual relationships, handling synonyms, and disambiguating terms. Knowledge graphs also facilitate the construction of embeddings, where entities and relationships are mapped into a continuous vector space. These embeddings capture the graph's structure and semantics, allowing for the calculation of distances between entities. As a result, entities that are closely related in the graph, such as a disease and its corresponding ICD code, will have similar embeddings, improving the accuracy of entity linking. Usually, ICD knowledge graphs need to be constructed before starting the linking process.

The four methods presented by Teng et al., 2020; Xie et al., 2019; Falis et al., 2019; Cao et al., 2020, leverage knowledge graphs to efficiently classify ICD entities from the MIMIC-III dataset. The first method combines a set of convolutional neural networks (Multi-CNN) with the knowledge graph, using an attention mechanism to enhance ICD concept classification. In the subsequent two papers, a specialized graph convolutional network (Graph-CNN) is employed to capture hierarchical relationships among medical codes. Additionally, the final method introduces hyperbolic representations to further refine the embeddings and improve classification performance.

A different approach is used in the article of Shi et al., 2017, where several LSTM neural networks encode diagnosis descriptions from the MIMIC-III dataset. This is followed by an attention layer and a linear projection that outputs the class probabilities.

Additionally the two papers of Ji, Hölttä, and Marttinen, 2021; Huang, Tsai, and Chen, 2022 explore the use of pretrained large language models, such as BERT, for ICD entity linking using MIMIC-III and MIMIC-II data. These models leverage their deep contextual understanding to improve the accuracy of entity classification.

2.2.2 Other Systems

In the work of Hristov et al., 2023 it is proposed a novel method for classifying clinical text into SNOMED CT codes using transformer models trained on open medical ontologies. They enhance transformer BERT models with clustering, filtering, and support vector classification (SVC) for embedding-based classification. This approach addresses the challenge of limited annotated data, improving accuracy in classifying short clinical text snippets.

ClinLinker (Gallego et al., 2024) links Spanish clinical concepts to SNOMED CT codes. It starts with a SapBERT-based bi-encoder and cosine similarity to identify candidate codes for each mention, followed by a cross-encoder for re-ranking to finalize the selection of identifiers.

In the article of Achara, Sasidharan, et al., 2024, a MiniEL model is employed to generate mention encodings and identify candidate entities based on cosine similarity. These candidates are then re-ranked, resulting in efficient entity linking to UMLS codes.

Ultimately, MedRoBERTa proves to be an effective model for linking entities in Dutch to a generated Wikipedia corpus in the paper of Hartendorp et al., 2024.

2.3 Automatic Clinical Coding

After reviewing the state of the art in biomedical named entity recognition and entity linking, the next step is to examine methods that integrate these two processes to automatically detect and link entities within biomedical free-text.

First, we will review classical approaches, which do not utilize machine learning or deep learning techniques. The subsequent section will focus on deep learning methods, including the latest advancements that employ large language models such as BERT.

2.3.1 Classical Approaches

Despite the majority of medical entity linking methods using machine or deep learning techniques, some older methods were developed that do not involve either machine learning or deep learning, as well as recent approaches that are state of the art. In this section, some of these methods will be outlined.

For instance, rule-based methods use a predefined set of conditions to identify and code clinical terms. An example of this approach is described in the work of Farkas and Szarvas, 2008, where terms are found and coded in the ICD-9-CM system based on a set of hand-crafted rules. Additionally, regular expressions (regex) are employed in automatic clinical coding to recognize and extract specific patterns or terms from unstructured medical text. By defining patterns that match relevant medical concepts, regex facilitates the extraction and coding process. This approach is illustrated by Zhou et al., 2020, where regex is used to detect and code ICD-10 diseases from Chinese free-text.

A more sophisticated three-stage approach is utilized by Patrick, Wang, and Budd, 2007 to detect and link medical terms to SNOMED CT identifiers. The process begins with the use of an augmented lexicon to index the terms. Next, a term compositor establishes relationships between the identified concepts. Finally, a "negation detector" is employed to identify negative concepts, which are terms where negation is present.

In the method developed by Guy Amit and Yanover, 2024, medical terms are linked to SNOMED CT concept IDs through a dictionary-based approach. Initially, a dictionary is constructed using a combination of training data and SNOMED CT concept names, which are augmented with a set of synonyms. This dictionary is further refined with simple linguistic rules and permutations of multi-word expressions. Two versions of the dictionary are created: one that is case-insensitive and another that is case-sensitive. During the inference phase, each document is processed individually by dividing it into sections and matching terms within those sections to the dictionary entries. Successful matches are annotated with the corresponding SNOMED CT concept IDs. Overlapping matches are resolved by prioritizing longer

mentions and section-specific terms over more general ones. Finally, in the post-processing stage, annotations are further refined and expanded based on SNOMED CT relationships to enhance their specificity and accuracy.

2.3.2 Deep Learning Approaches

Deep learning approaches for automatic clinical coding generally follow a two-step process: named entity recognition and entity linking, akin to the methods discussed in previous sections. This section will now explore five of the most advanced techniques, most of which employ BERT models.

First of all, the framework named unMERL (Xu et al., 2018) is found to be effective to recognize and link Chinese free-text mentions to SNOMED CT identifiers. It is divided in two modules. The maximum entropy (ME) recognition module processes an input text to identify entity boundaries and classify entities. That is, it determines where specific terms or phrases representing distinct medical entities begin and end. The module then generates a list of these entities and their categories. For each identified medical entity, the ME linking module retrieves and ranks potential matches from a knowledge base to find the most relevant one.

In the article of Borchert and Schapranow, 2022, a method for detecting and classifying entities in Spanish text into SNOMED CT codes is detailed. Named entity recognition is performed using a RoBERTa model. The linking process comprises two main steps: candidate selection and ranking to determine the most likely identifier. Candidate selection involves calculating embeddings for both the text mentions and SNOMED CT terms with a cross-lingual SapBERT model, which utilizes a TF-IDF representation of the DisTEMIST data. This is followed by using k-nearest neighbors to identify the most similar embedding terms for each mention. The ranking phase applies a set of rules to adjust and refine the scores, ultimately retaining the best match.

Another approach to entity recognition and linking to SNOMED CT codes in Spanish using the DisTEMIST dataset is described in the work of Reyes-Aguillón et al., 2022. Named entity recognition (NER) is performed using a BERT model. For linking mentions to their corresponding SNOMED CT codes, the method calculates the cosine similarity between the average vectors of each recognized entity and the vectors of SNOMED CT terms. These vectors are obtained using the SciELO-CBOW model, which is a pre-trained Spanish language model.

Automated clinical coding for COVID-19 has been explored in the paper of Sohrab et al., 2020, where mentions are linked to UMLS system identifiers using the CORD-19 dataset, which includes free-text from numerous COVID-19 research papers. The framework, called BENNERD, consists of two stages: a Named Entity Recognition (NER) stage using a BERT model, followed by an entity linking stage. In the linking phase, an approximate nearest neighbor search first retrieves the top candidate entities. These candidates are then ranked using a fully-connected neural network that concatenates mention and entity embeddings and applies a linear layer to score them.

The SNOBERT method (Kulyabin et al., 2024) detects medical entities in the MIMIC-IV dataset and links them to SNOMED CT codes. The Named Entity Recognition (NER) step employs BiomedBERT, a BERT model tailored for biomedical terminology. Candidate generation for each mention is handled by PubMedBERT, another specialized BERT model. To refine these initial predictions, a re-ranking stage is applied using a cross-encoder model, trained on PubMed papers, to improve accuracy.

Chapter 3

Data

As this work focuses on two specific methods and the SNOMED CT terminology, this chapter will explain the most relevant data required to run these methods, as well as the SNOMED CT knowledge base. Understanding this data is crucial for grasping the functionality of both methods, which is why this section precedes their detailed explanation.

First, the elements and relationships within the SNOMED CT knowledge base will be detailed. This will be followed by an explanation of the MIMIC-IV-Note dataset and the data provided by the contest, including an exploration of the latter.

3.1 SNOMED CT Knowledge Base

As explained in the introduction, SNOMED CT is a medical terminology structuring system. The database available at https://www.nlm.nih.gov/healthit/snomedct/international.html serves as the knowledge base for SNOMED CT International Edition. This database is organized as a set of folders and files, containing all the updated information on SNOMED CT terms, including their IDs, descriptions, hierarchical relationships, and more.

The SNOMED CT files are structured as relational tables, with each line in the file representing a row in the table. The first row of each table contains the column headings, while the subsequent rows contain the data.

This database allows for the construction of a knowledge graph, where all the information about hierarchical relationships among terms, their IDs, meanings, and descriptions is consolidated into a single data structure.

3.2 MIMIC-IV-Note Dataset

MIMIC-IV is a contemporary electronic health record dataset that covers a decade of admissions between 2008 and 2019 (Johnson et al., 2023b).

The MIMIC-IV-Note dataset is a dedicated module within the MIMIC-IV framework. It consists of deidentified free-text clinical notes from the MIMIC-IV clinical database. This dataset encompasses 331,794 anonymized discharge summaries for 145,915 patients admitted to the hospital and emergency department at Beth Israel Deaconess Medical Center in Boston, MA, USA. It also includes 2,321,355 deidentified radiology reports covering 237,427 patients (Johnson et al., 2023a). Nevertheless, in this work, only the discharges will be employed.

3.3 Contest Data

The data provided by the contest has been used for training and testing purposes. This dataset consists of 204 annotated discharge summaries from the MIMIC-IV-Note dataset (Hardman et al., 2023b). Two files are provided: one containing the free-text notes and the other containing 51,574 annotated terms to SNOMED CT concepts for these texts, but only 5,334 different IDs among these terms. Each file includes the following information:

- **Notes**:
  - ID of the note.
  - Text of the note.
- **Annotations**:
  - ID corresponding to the note where the annotated term appears.
  - Starting position of the annotated term on the text note.
  - Ending position of the annotated term on the text note.
  - SNOMED CT ID corresponding to the annotated term.

Figure 3.1 shows a graphical example of an annotated text.

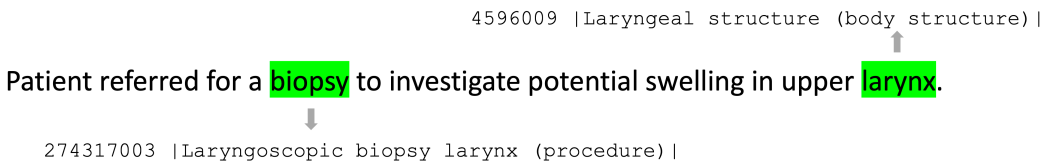

FIGURE 3.1: Example of medical text and labeled concepts. The concepts are highlighted in green, and the indicated SNOMED IDs, names, and categories are shown. Extracted from https://www.drivendata.org/competitions/258/competition-snomed-ct/page/914/

As the texts are from medical discharge summaries, all of them have a similar structure of sections. Some of those sections are, for instance, “chief complaint”, “history of present illness”, “past medical history” or “physical exam”.

A relevant observation from exploring the distribution of the annotations is that, as shown in Figure 3.2, most concepts appear infrequently. Specifically, the majority of concepts occur fewer than 100 times. This "long tail" distribution effect can make the automatic annotation of less frequent concepts more challenging.

Only a few concepts appear frequently. The three most common concepts are as follows:

- The most frequent concept, appearing 584 times, is “No abnormality detected” (ID: 281900007), a “finding”. It appears in various forms in the text, such as “NAD”, “No acute”, “intact”, “normal”, “unremarkable”, “WNL”, and “negative”.
- The second most common concept is “Red cell distribution width determination” (ID: 66842004), a “procedure” that appears a number of 489 times. Unlike the first, it primarily appears as “RDW” and “RDWSD”.

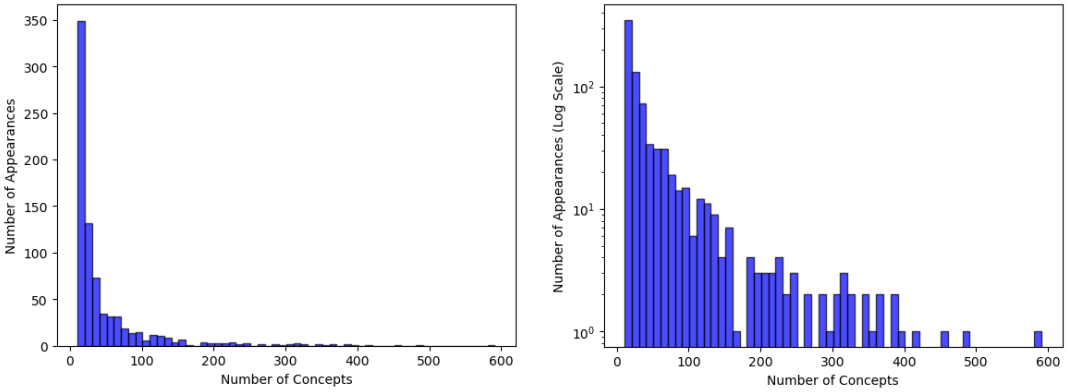

FIGURE 3.2: Distribution of the appearance frequency for each concept in both linear (left) and logarithmic (right) scales.

- The third concept is "Physical examination procedure" (ID: 5880005), also a "procedure" appearing 455 times. It is found in different forms, including "exam", "HEENT", "PHYSICAL EXAM", "Neck", and "PHYSICAL EXAMINATION".

Chapter 4

Methods

Among the various automated clinical coding methods, this work focuses on studying two in particular: the winning approaches from the "SNOMED CT Entity Linking" contest. These two methods represent state-of-the-art techniques for addressing the full task of entity recognition and linking, each employing a distinct approach. In fact, both methods were mentioned in the previous chapter.

This section provides a detailed explanation of the original methods, as well as the modifications and contributions made to them. Subsequent sections will discuss the experiments conducted, the improvements implemented, and the analysis of these advancements.

4.1 First Method

This initial method, involves constructing a dictionary that maps pairs of section headers and mentions to SNOMED CT terms. This approach does not rely on machine learning or deep learning techniques. Instead, it uses data from various sources and leverages relationships among SNOMED CT terms, word permutations, synonyms, and word replacements, among other techniques, to accurately identify and match terms within free-text. The original code for this method can be found in the GitHub repository https://github.com/drivendataorg/snomed-ct-entity-linking/tree/main/1st%20Place.

The method's workflow is divided into three stages. The first stage involves preprocessing the data. The second stage is the training phase, where dictionaries are generated from the preprocessed data. Finally, the third stage consists of annotation or inference, followed by postprocessing to generate predictions for the test texts.

Next, each of these three stages will be described in detail. However, before that, the necessary initial data will be explained.

4.1.1 Initial Data

This initial data required to run this automatic clinical coding method is sourced from or stored in five distinct locations:

- **Medical Abbreviations**: Downloaded from https://github.com/imantsm/medical_abbreviations, this dataset contains common abbreviations used by health professionals and their corresponding full meanings.
- **OMOP Synonyms**: This dataset, downloaded from https://www.ohdsi.org/data-standardization includes information on synonym relationships among medical concepts, based on the Observational Medical Outcomes Partnership (OMOP) (Observational Health Data Sciences and Informatics (OHDSI), 2024).

- **SNOMED CT Filtered Knowledge Base**: Contains updated information on all SNOMED CT terms, including the IDs of each term and the hierarchical relationships among SNOMED CT concepts, such as which concepts are inherited or are children of others.
- **Train Notes**: Free-text medical records selected for training the inference model.
- **Train Annotations**: Annotations corresponding to the Train Notes, identifying the relevant medical terms for each record.
- **Test Notes**: Free-text medical records selected for evaluating the inference model. These notes are separate from the training set and are used to assess the model's performance on unseen data.
- **Test Annotations**: Annotations corresponding to the Test Notes, identifying the relevant medical terms for each record. These annotations are used to evaluate the accuracy of the model's predictions.
- **MIMIC-IV-Note**: The complete MIMIC-IV-Note dataset, containing all free-text medical notes.

4.1.2 Preprocessing

This first task involves processing, merging, and extracting information from the initial data to create a set of tables and dictionaries that will be utilized in the subsequent stages. The general preprocessing workflow is illustrated in 4.1. Now, this process will be broken down in detail.

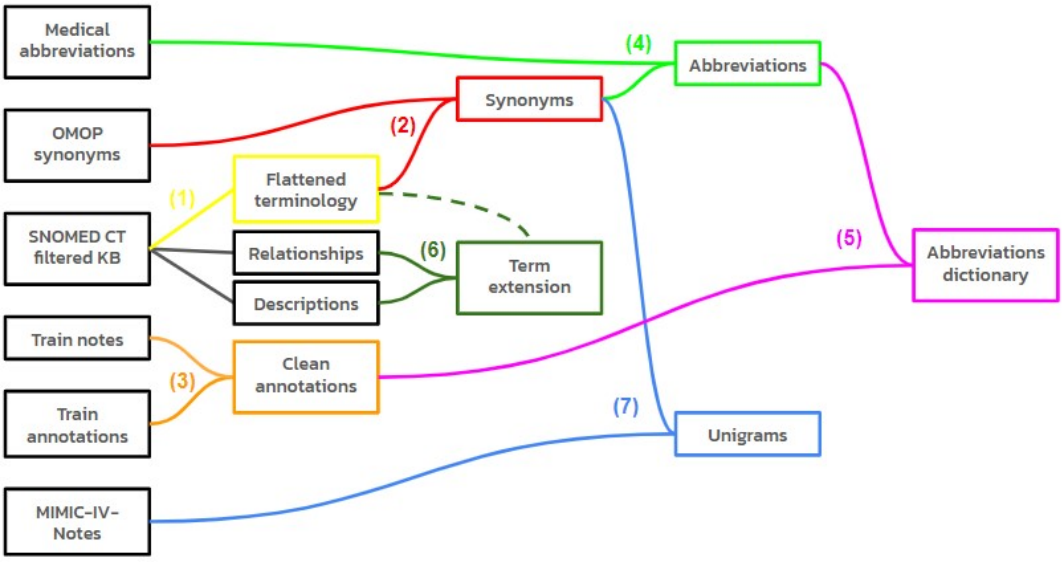

FIGURE 4.1: Main scheme of the preprocessing stage for the first method. The numbers indicate the sequence of operations, while the colors associated with each number illustrate the relationships involved and the data generated during each phase.

1. **Create Flattened Terminology**: The SNOMED CT terminology dataset, which includes all terms and associated information, is processed to produce a flattened terminology dataset. This dataset includes only concepts categorized as

"procedure", "body structure", and "finding", as well as "disorder" (a child of "finding"), "regime/therapy" (a child of "procedure"), and "morphologic abnormality" and "cell structure" (both children of "body structure"). Only one row per concept ID is retained, removing any synonyms.

2. **Generate Synonyms**: The OMOP synonyms dataset is merged with the flattened terminology to create a comprehensive table that includes all concepts (IDs and names) and their synonyms. This table also specifies the hierarchical level or category and indicates whether a term is a synonym or the primary name. This ensures that each main concept is followed by its synonyms before moving to another concept.

3. **Clean Annotations**: Training notes and training annotations are merged into a single table. This table includes annotated words, their concept IDs, the starting and ending positions of the annotations in the text, and the ID of the text note where the mention is found.

4. **Extend Abbreviations**: The dataset of synonyms is merged with the medical abbreviations file to expand the abbreviations data. This extension includes the associated SNOMED CT concept name and ID corresponding to each abbreviation and its full form.

5. **Create Abbreviations Dictionary**: An extended abbreviations dataset is used to create a dictionary that maps sections where a concept appears and its abbreviation to its corresponding ID. The dictionary is structured as:

$$\left\{\left(\left\{\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_n\right\}, \mathrm{a}_j\right) : x_i\right\}, \tag{4.1}$$

where $\{\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_n\}$ represents the set of sections where the i-th concept with SNOMED CT ID x_i, having the j-th abbreviation a_j, appears.

6. **Perform Term Extension**: The "Relationships" and "Descriptions" datasets from the SNOMED CT knowledge base are merged to produce a dataset that links general terms to specific terms. For each general term, its ID and name are repeated for every concept that derives from or inherits it. This dataset allows for the identification of specific concepts by their specific names, IDs, general terms, and the discriminative word used to classify the specific term. For instance, "Burn of ring finger" is associated with "Burn of finger" as its general term, with "ring" serving as the discriminative word. Only the terms with IDs contained in the flattened terminology are considered.

7. **Create Unigram Dictionaries**: Two unigram dictionaries are constructed in the following format:

$$\left\{\left(\text{'any'}, \mathrm{w}_i\right) : x_i\right\}, \tag{4.2}$$

where w_i represents a word and x_i is its corresponding SNOMED CT ID.

Each dictionary is built with words that appear fewer than 20,000 and 3,000 times, respectively, in the complete MIMIC-IV-Note dataset.

4.1.3 Train

After preprocessing, the training phase involves creating two types of dictionaries: one case-sensitive and one case-insensitive. These dictionaries map pairs of sections and terms to their corresponding SNOMED CT IDs. The authors of this method refer to these dictionaries as Kiri dictionaries. Thus, training in this method is the process of constructing these Kiri dictionaries.

The entire process operates by iterating through each training note, meaning each text in the training set, along with its corresponding annotations.

First, an empty dictionary is initialized. For each annotation in a text, the annotation's ID is stored in two ways: first, with a generic tuple $(\text{“any”}, \text{mention})$, and second, with a specific tuple $(\text{section}, \text{mention})$ where it was found. Certain sections that do not provide significant information are excluded from this process. Then, for each pair, only the most frequently occurring SNOMED CT ID is retained. In other words, for each pair, only the ID most commonly used to code that specific tuple is kept.

Once the primary dictionary has been created, it is divided into a case-sensitive version and a case-insensitive version. The case-insensitive dictionary is then extended using other dictionaries or by applying specific rules. Specifically, four extensions are performed:

1. **Synonyms**: The synonyms file, generated during preprocessing by merging the flattened terminology with OMOP synonyms, is used to extend the dictionary by adding tuples in the form $(\text{“any”}, \text{synonym})$, where "synonym" represents the synonyms of the words already present in the dictionary.

2. **Unigrams**: The unigrams from the preprocessing stage, including both the 3,000 and 20,000-item dictionaries, are added to the main dictionary.

3. **Word Replacements**: A set of predefined simple replacements are applied to the current dictionary to further expand its entries. For each item in the dictionary, if its mention contains a specific string from a defined set, that string is replaced with another predefined string. The modified mention, along with the original section and ID, is then added to the dictionary. For example, if an item $(\text{section}, \text{mention})$ undergoes a replacement in the mention, the new item added to the dictionary will be $(\text{section}, \text{modified mention})$, where the mention has been altered based on the replacement.

   The specific replacements applied are:

   - A comma is replaced by an empty space.
   - The phrase “and” is replaced by “with”.
   - The phrase “with" is replaced by “and”.
   - The word “valve” is replaced by an empty space.
   - The phrase “of” is replaced by “of the”.

4. **Permutations**: A set of permutations is applied to the mentions in the dictionary, adding the newly permuted mentions while retaining the original sections and IDs. If a mention contains the word “of” and consists of either three or four words, “of” is removed, and the remaining words are rearranged. For example, the term “pain of appendicitis” would be permuted to “appendicitis pain”, and a new item with the same section and ID would be added to the dictionary under this new, permuted name. If “of” is not in the mention but

the mention still has three or four words, then all possible word permutations are added to the dictionary. Nevertheless, if the mention contains a different number of words than three or four, no permutation is performed.

4.1.4 Annotation and Postprocessing

This final stage utilizes the Kiri dictionaries to annotate the test notes. Initially, predictions are made, followed by a postprocessing phase to refine and enhance the results.

First Prediction

The initial prediction of relevant terms in the text is carried out twice: once with the case-insensitive dictionary and once with the case-sensitive dictionary, which is extended with the abbreviations dictionary created during preprocessing.

For each dictionary, the prediction process is performed text by text. Each text is first divided into sections. Then, mentions from the dictionary are identified in the text using a set of regular expressions. If the section specified in the dictionary matches the section where the term is found in the text, or if the section is"any", the starting and ending positions of the matched term are recorded. Specific sections are given preference over the generic "any".

After predictions are made with each dictionary, they are combined. In cases of overlap, conflicts are resolved by prioritizing longer and more specific mentions and section-specific terms over more general ones.

Postprocessing

After the initial prediction, a subsequent phase of postprocessing is necessary to improve and fine-tune the results.

For each predicted term, the SNOMED CT hierarchical structure is used to specify more general terms. Each detected term is expanded by adjusting its starting and ending positions. Matches are sought among concepts that inherit from the predicted more general concept. If a more specific term is found among the related children concepts, this more specific term is retained and prioritized over the general one. This process utilizes the previously generated data of term extensions.

4.2 Second Method

The second method employs a completely different approach, divided into two main steps: named entity recognition and entity linking.

After preprocessing the data, a BERT model is used to perform the NER task. To simplify the NER process, only three broad categories are considered: "Procedure", "Finding", and "Body Structure". Other relevant categories that inherit from these are grouped under the general categories during the NER phase. Following this, a candidate matching step assigns the top k candidate SNOMED CT terms to each detected mention using a cosine similarity metric. Finally, a re-ranker refines the scores and selects the best candidate from the identified terms. The main workflow is illustrated in 4.2. The original code for this method can be found in the GitHub repository `https://github.com/drivendataorg/snomed-ct-entity-linking/tree/main/2nd%20Place`.

Each step will be detailed after an explanation of the necessary data required to use this method.

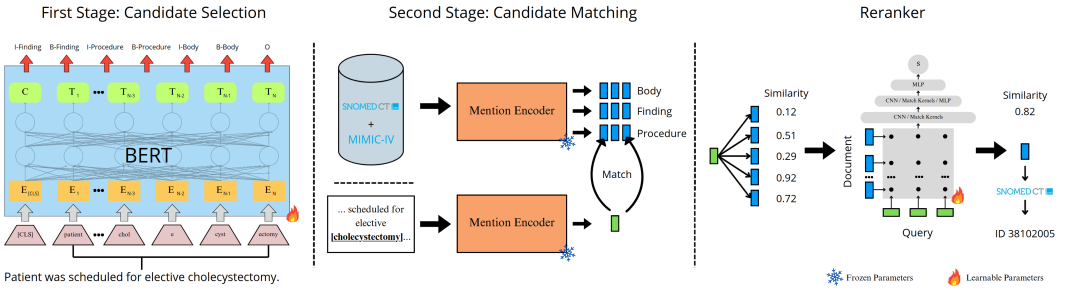

FIGURE 4.2: Main scheme of the second method.

4.2.1 Initial Data

Unlike the first method, this approach requires less initial data. It only needs the following:

- **SNOMED CT Knowledge Base**
- **Train Notes**
- **Train Annotations**
- **Test Notes**
- **Test Annotations**

All of this data has already been explained in the previous method.

4.2.2 Preprocessing

In the preprocessing stage, the initial notes are processed and the necessary data for the subsequent steps is generated. The specific operations performed during this stage are as follows:

1. **Process the Training Notes**: The training notes contain several inaccuracies that need correction. Issues such as shifts caused by tags are resolved, and sections with a high proportion of missing values are removed to ensure they do not affect the training process. Additionally, all HTML markup elements, such as line breaks and new lines, are stripped from the text.
2. **Create the Knowledge Graph**: The SNOMED CT knowledge base is transformed into a knowledge graph serialized object. This allows for efficient handling of dependencies and hierarchies among SNOMED CT concepts in future steps.
3. **Extract the Most Common Concepts**: A dictionary is generated that maps each term in the train annotations to the SNOMED CT ID that has most frequently associated.
4. **Generate Word Embeddings**: Word embeddings are created using a **SapBERTEmbedder**. The embeddings are generated separately for each of the three main categories: “Procedure”, “Finding” and “Body Structure”.

4.2.3 Candidate Selection (Named Entity Recognition)

As previously explained, the NER task involves detecting entities categorized as "Procedure", "Finding" and "Body Structure". To address this task, the traditional B-I-O (Begin-Inside-Outside) system is employed. In this system, "B" denotes the beginning of an entity, "I" indicates words inside an entity, and" O" signifies words outside any relevant entity. Therefore, the system considers the following seven classes: "I-Finding", "B-Finding", "I-Procedure", "B-Procedure", "I-Body", "B-Body", and "O".

This NER task is addressed by fine-tuning a pre-trained **BiomedBERT** model using the embeddings created during preprocessing.

4.2.4 Candidate Matching

The aim of this section is to match the detected entities to the most similar SNOMED CT terms that share the same class type.

For each class type, the embeddings of the recognized entities are compared with SNOMED CT term embeddings of the same class, retaining the top k most similar terms. Similarity is calculated using **cosine similarity**, which can be computed as:

$$\cos(\theta) = \frac{\sum_i^n v_i u_i}{\sqrt{\sum_i^n v_i^2}\sqrt{\sum_i^n u_i^2}}, \tag{4.3}$$

where $v \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ are the vectors being compared.

4.2.5 Re-ranking and Postprocessing

After selecting the top k SNOMED CT candidates for each recognized mention in the notes, a re-ranker refines these predictions by applying more specific criteria.

The re-ranking is performed using the MedCPT model (Jin et al., 2023), which was trained on 18 million semantic query-article pairs from PubMed. MedCPT employs a cross-encoder architecture, initialized with a pretrained PubMedBERT model. A cross-encoder is a deep learning NLP model designed to compare and classify pairs of sentences or text sequences by computing the similarity between them. Although MedCPT is originally designed to retrieve the most relevant PubMed articles based on their titles in response to specific queries, in this context, it is used to compare and re-rank the recognized mentions against SNOMED CT terms, ultimately predicting the most accurate SNOMED CT concept corresponding to each mention.

Moreover, the terms from the dictionary of the most common concepts, created during preprocessing and found in the text, are also added to the prediction and linked to their respective IDs.

4.3 Modifications and Contribution

Before testing the methods, several modifications to both the methods and the data were made to address specific issues and ensure the code runs effectively. In addition, a contribution was made to the second method by incorporating an idea from the first method. All the new code with all the modifications and contributions can be found in the GitHub repository `https://github.com/SergiCSim/SNOMED-CT-AutomatedClinicalCoding`.

This chapter will first discuss the modifications made, followed by an explanation of the contribution to the second method.

4.3.1 Adaptations and Modifications

First, the code for the first method was originally designed to convert the entire MIMIC-IV-Note dataset text into a single string, which caused a memory error. To resolve this, the code was modified to read and process the text in chunks, thereby avoiding memory issues.

Besides that, new code was written to automate the training and testing processes for the first method. The original code was intended to train on the entire contest dataset and generate submissions for new notes. A new Bash script now automatically handles the entire training and testing process, including necessary file deletions and movements. It also divides the notes into training and testing sets, along with their corresponding annotations.

Another modification involved addressing issues related to the data. Since the SNOMED CT knowledge base is frequently updated, the version of SNOMED CT data originally intended for use in the method did not match the newly updated data. This mismatch led to inconsistencies in the second method, particularly due to changes in the activation or deactivation of relationships between concepts. A total of 46 problematic IDs were identified out of 5,336, which represents approximately 0.86% of the SNOMED CT concepts in the annotations. These problematic IDs appeared in 218 out of 51,574 rows in the annotations, accounting for roughly 0.42% of the rows.

Given the small proportion of problematic rows and IDs, the chosen solution was to delete the rows containing these problematic IDs. Consequently, all experiments in both the first and second methods were conducted using the updated annotations, with the problematic IDs removed.

4.3.2 Contribution

In the first method, text notes are divided into sections, which play a crucial role. As previously explained, the text is broken down into sections, and pairs of (section, mention) are linked to their corresponding IDs. However, the structure of the text in sections is not utilized in the second method, where all the text is treated uniformly without considering sections. The contribution made to the second method aims to slightly improve its predictions by incorporating the section structure into the process.

As explained in Section 4.2.5, the re-ranking process selects the best SNOMED CT concept from the top k candidates. The similarity score, initially calculated as cosine similarity in the previous phase, is recalculated using more specific criteria related to the content of the mentions and the SNOMED CT candidate terms. The candidate term with the highest score after re-ranking is then selected as the best match.

Instead of considering only the score of the re-ranker, s_1, a linear combination of this score and another one related to the section, s_2, can be considered. As in the named entity recognition the entities are categorized either “procedure”, “body structure” or “finding” and each mention is linked to a SNOMED CT concept within the same category, the second score can be calculated only taking into account the category of the detected mention.

Therefore, this second score can be defined as the probability of finding the SNOMED CT concept with ID i among all the concepts in the same category k in the section where the mention has been found. It can be calculated empirically using the information from the training notes and annotations.

If a specific mention is found in section ("sec") j, the probability that this mention corresponds to the SNOMED CT term "T" with ID i and category ("cat") k can be mathematically estimated as:

$$\Pr(\text{ID} = i \mid \text{sec} = j) = \frac{\Pr(\text{ID} = i \cap \text{sec} = j)}{\Pr(\text{sec} = j \cap \text{cat} = k)} = \frac{\frac{\#\{\text{T} : \text{ID}=i \cap \text{sec}=j\}}{\#\{\text{T} : \text{cat}=k\}}}{\frac{\#\{\text{T} : \text{sec}=j \cap \text{cat}=k\}}{\#\{\text{T} : \text{cat}=k\}}} \tag{4.4}$$

Hence,

$$s_2 = \frac{\#\{\text{T} : \text{ID} = i \cap \text{ sec} = j\}}{\#\{\text{T} : \text{sec } = j \cap \text{cat} = k\}}, \tag{4.5}$$

where $\#A$ is the cardinality of the set A.

It should be noted in 4.3.2 that, when calculating $\Pr(\text{sec } = j)$, it is implicitly understood that the category must be k, as this is the same category corresponding to the ID i.

Chapter 5

Experiments and Results

This chapter will first explain the experiments performed using the methods, incorporating the modifications and adaptations described in Section 4.3.1, as well as the second method modified according to the contribution outlined in Section 4.3.2.

Next, a brief application of automated clinical coding will be discussed, along with the results of the related experiment.

5.1 Methods

5.1.1 Metrics

To assess the performance of the methods, three metrics were considered: the macro-averaged character intersection-over-union (mIoU), the global F1 score (F1) and the macro-averaged F1 score (mF1).

Macro-averaged Character Intersection-over-union

The intersection-over-union (IoU) is a widely used metric for evaluating performance by calculating the overlap between two sets. In this context, the overlap is determined by comparing the characters of the predicted mentions with those of the ground truth, assessing both the intersection and the union based on characters at the same positions in the text.

However, the metric employed here is the macro-averaged intersection-over-union. This variant averages the IoU across classes, with each class representing a SNOMED CT concept.

This metric was the one evaluated in the "SNOMED CT entity linking" contest, where the objective was to achieve the highest macro-averaged intersection-over-union (mIoU) on a set of notes with annotations that were unknown to the participants and known only to the contest organizers.

The IoU for one class can be calculated as:

$$\text{IoU}_{\text{class}} = \frac{P^{\text{char}}_{\text{class}} \cap G^{\text{char}}_{\text{class}}}{P^{\text{char}}_{\text{class}} \cup G^{\text{char}}_{\text{class}}}. \tag{5.1}$$

And, the macro-averaged one, as:

$$\text{mIoU} = \frac{\sum_{\text{classes} \in P \cap G} \text{IoU}_{\text{class}}}{N_{\text{classes} \in P \cap G}}, \tag{5.2}$$

where $P^{\text{char}}_{\text{class}}$ is the set of characters in all predicted spans for a given class, and $G^{\text{char}}_{\text{class}}$ is the set of characters in all ground truth spans for the given class, and classes $\in P \cap G$ is the set of classes present in either the ground truth or the predicted spans.

F1 Score

The F1 score is a metric that combines precision and recall into a single value. It is particularly useful for imbalanced datasets. The F1 score is the harmonic mean of precision and recall.

Precision is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.3}$$

Recall is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.4}$$

The F1 score is calculated as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.5}$$

Macro-Averaged F1 Score

The macro-averaged F1 score is used in multi-class classification problems. It computes the F1 score for each class independently and then averages these scores. This method treats all classes equally.

Let F1_{class} denote the F1 score for a specific class. The macro-averaged F1 score is calculated as:

$$\text{mF1} = \frac{1}{N_{\text{classes}}} \sum_{\text{class}} \text{F1}_{\text{class}} \tag{5.6}$$

where N_{classes} is the number of classes, and F1_{class} is the F1 score for each individual class.

5.1.2 Experiments

The experiments involve training and testing both methods, as well as the modified second method. Metrics are calculated for each prediction to evaluate the accuracy of the predictions in relation to the test annotations, which serve as the ground truth.

The data used in the experiments consists of the dataset with problematic concepts filtered, as described in Section 4.3.1. The 204 medical notes were divided such that 80% of the notes, a number of 164, were used for training and the remaining 20%, a quantity of 40, for testing. Accordingly, the training annotations were selected from the annotated terms of the training notes, while the test annotations were drawn from the annotated terms of the test notes.

In the modified second method, the new mixed score, s_{mixed}, is calculated by considering the conditioned probabilities of the sections (Section 4.3.2). The mixed score is computed as

$$s_{\text{mixed}} = w_1 s_1 + w_2 s_2, \tag{5.7}$$

where s_1 and s_2 are combined linearly according to the weights $w_1 \in \mathbb{R}$ and $w_2 \in \mathbb{R}$.

In the experiments with the modified second method, various combinations of weights were tested to evaluate the results with different linear combinations of the two scores.

TABLE 5.1: mIoU, F1 and mF1 metrics of the first method, second method and modified second method using different configurations of weights. The first method does not use weights.

Method	w_1	w_2	**mIoU**	**F1**	**mF1**
1st	—	—	0.43241	0.72390	0.45358
2nd (modified)	0	1	0.34509	0.69644	0.37050
2nd (original)	1	0	0.42740	0.71670	0.44752
2nd (modified)	1	1	0.42741	0.71651	0.44749
2nd (modified)	1	2	0.42762	0.71344	0.44739
2nd (modified)	1	3	0.42718	0.71329	0.44688
2nd (modified)	1	4	0.42703	0.71287	0.44683
2nd (modified)	1	5	0.42554	0.71227	0.44571
2nd (modified)	1	6	0.42556	0.71161	0.44553
2nd (modified)	1	7	0.42478	0.71107	0.44482
2nd (modified)	1	8	0.42523	0.71071	0.44527

Since the magnitude of the scores does not impact the final result, as only the highest score is retained, and the values of the second score are typically much smaller than those of the first, w_1 has been set to $w_1 = 1$. Nine integer values of w_2, ranging from 0 to 8, have been tested with $w_1 = 1$, that is, $w_2 = 0, 1, \ldots, 8$. The run with $w_1 = 1$ and $w_2 = 0$ corresponds to the original second method. Additionally, an experiment with $w_1 = 0$ and $w_2 = 1$ has also been conducted.

The first method was tested on a system equipped with an Intel® Core™ i5-6200U processor running at 2.30GHz, 8GB of RAM, and a 2.40GHz CPU.

For the second model, a BiomedNLP-BiomedBERT-large-uncased-abstract was utilized during the natural language processing phase. This model was trained on four NVIDIA A100-SXM4-40GB GPUs, which together provide 27,648 CUDA cores and 1 TB of RAM. The same pre-trained BERT-based model was used across all configurations for the modified second method. During the entity linking phase, each weight configuration for the modified method was processed once using a single NVIDIA GeForce RTX 4060 GPU with 3,072 CUDA cores, 8GB of RAM, and an Intel® Core™ i7-14700F processor running at 2.10GHz, supported by 32GB of RAM.

5.1.3 Results

The results are presented in Table 5.1. The dictionary-based first method outperforms all others across the three metrics. However, within the experiments using the modified second method, the configuration with $w_2 = 2$ yields the highest mIoU. Conversely, the original second method achieves the best F1 and mF1 scores. The poorest performance is observed when only the second score is considered, specifically when $w_1 = 0$. Nevertheless, the differences between the metrics across various weight configurations in the modified second method appear to be minimal.

5.2 Application

Automated clinical coding has multiple applications, primarily including the structuring of electronic medical records into organized data to efficiently manage relevant medical information.

This section will demonstrate a specific case where automated clinical coding proves useful.

5.2.1 Description and Experiments

In the MIMIC-IV-Note medical summaries, there are multiple terms that have a number associated. For instance, quantites of medicaments, concentration of determinated substances, etc.

One interesting numerical term is "vital signs", which has the SNOMED CT ID of 118227000. In SNOMED CT, the relevant term is not the values of the vital signs themselves but rather the set of words that identify them. For example, some terms found in the medical texts that are categorized as vital signs include "VS", "vital signs", "Vitals", "VITALS", and "VSS". There are 313 annotations categorized with the vital signs ID in the annotation data.

Extracting the vital signs from free-text is not always an easy task, due to the varied ways in which the five vital signs are reported. The vital signs can appear in different forms, orders, and there can be missing values that are not present in the text. Specifically, the five vital signs are:

- **Temperature**
- **Blood Pressure** (Systolic and Diastolic)
- **Heart Rate**
- **Respiration Rate**
- **Oxygen Saturation**

Thus, the experiment performed consists of determining whether knowledge of where the vital signs are found in the text, thanks to automated clinical coding, could facilitate the extraction of numerical values for these vital signs from the text more effectively compared to when this information is not known.

Specifically, given the identifier term, each text has been filtered to only include the information that is 100 characters after the end of the identifier. This approach aims to extract the vital signs from this filtered text, rather than attempting to obtain them from the entire text.

Due to the limited number of samples (only 313 annotations with vital signs), utilizing a deep learning NLP model to extract vital signs numbers has proven ineffective due to insufficient training data. Instead, a solution was implemented using a set of conditions based on regular expressions to extract the vital signs values. The details of this solution can be found in the GitHub repository of this project.

5.2.2 Results

The evaluation of the method for vital signs detection is challenging due to the absence of ground truth data. Nevertheless, it has been observed to be effective in most cases and this can be checked in the available code. To illustrate this, five examples of vital signs extraction with the selected partial texts will be presented, along with a comparison of the results when applying the same extraction code to the entire text.

For the first text:

" remained stable. Please follow up at next PCP ___. [ ] Patient had a significant headache during ad",

TABLE 5.2: First example comparing predicted vital signs from partial versus full text with actual values.

Vital Sign	Prediction (Full Text)	Prediction (Partial Text)	Real Value
Temperature	98.1	None	None
Blood Pressure	130/88	None	None
Heart Rate	50	None	None
Respiration Rate	5	None	None
Oxygen Saturation	saturation	None	None

TABLE 5.3: Second example comparing predicted vital signs from partial versus full text with actual values.

Vital Sign	Prediction (Full Text)	Prediction (Partial Text)	Real Value
Temperature	98.3	98.3	98.3
Blood Pressure	137/69	137/69	137/69
Heart Rate	83	83	83
Respiration Rate	16	16	16
Oxygen Saturation	98%	98%	98%

where no vital sign numbers are present but a vital signs identifier was mentioned earlier, the algorithm's predictions are shown in Table 5.2. It is apparent that the predictions based on the full text do not align with the actual values. Instead, the algorithm generates incorrect values along with a single word from the text.

In contrast, for the second text:

": 98.3, 83, 137/69, 16, 98% RA GEN: NAD, AAO x 3 CV: RRR PULM: CTAB ABD: Laparoscopic incisions open",

where all vital signs are explicitly included and separated with commas at the starting of the text, the predictions in Table 5.3 are accurate and align well with the provided information.

For the third text:

"- T 97.9 BP138/67 HR78 RR20 SaO2 98RA GENERAL: Alert, oriented, no acute distress HEENT: Sclera a"

where all vital signs are present but preceded by identifiers such as "T" for temperature, "BP" for blood pressure, etc., the predictions in Table 5.4 demonstrate accurate identification of all vital signs in the partial text. However, the predictions are incorrect when applied to the full text.

In the fourth text:

": Temp 98.8; BP 140/47; HR 102; RR 20; SpO2 93% on 3L. . Currently, her dyspnea is much better. ",

where the vital signs are expressed similarly to the previous text but with different identifiers ("Temp" for temperature and "SpO2" for oxygen saturation) and separated by semicolons, the predictions in Table 5.5 show mixed results. For two cases, the predictions remain consistent between the full and partial texts. However,

TABLE 5.4: Third example comparing predicted vital signs from partial versus full text with actual values.

Vital Sign	Prediction (Full Text)	Prediction (Partial Text)	Real Value
Temperature	97.5	97.9	97.9
Blood Pressure	126-148/64-87	138/67	138/67
Heart Rate	None	78	78
Respiration Rate	19	20	20
Oxygen Saturation	100%	98%	98%

TABLE 5.5: Fourth example comparing predicted vital signs from partial versus full text with actual values.

Vital Sign	Prediction (Full Text)	Prediction (Partial Text)	Real Value
Temperature	98.8;	98.8;	98.8
Blood Pressure	70/30	140/47;	140/47
Heart Rate	95	102	102
Respiration Rate	28;	20;	20
Oxygen Saturation	80%	93%	93%

for the remaining cases, only the predictions based on the partial text are accurate, though some predictions include semicolons.

For the fifth text:

" prior to transfer were: 56 105/55 18 94% RA Upon arrival to the floor, pt CP free. No SOB, dizzi",

where it appears that there is no temperature value provided, the predictions shown in Table 5.6 indicate that only the respiration rate is accurately predicted when using the full text. The algorithm incorrectly predicts a temperature value with the full text, even though no temperature should be present. Nonetheless, all predictions based on the partial text are correct.

TABLE 5.6: Fifth example comparing predicted vital signs from partial versus full text with actual values.

Vital Sign	Prediction (Full Text)	Prediction (Partial Text)	Real Value
Temperature	98.1	None	None
Blood Pressure	140/70	105/55	105/55
Heart Rate	64	56	56
Respiration Rate	18	18	18
Oxygen Saturation	96%	94%	94%

Chapter 6

Discussion and Conclusions

In this final chapter, the results obtained from the experiments conducted in the previous chapter will be discussed. Subsequently, the overall conclusions of the work will be presented.

6.1 Discussion of the Method's Results

6.1.1 Original Methods

Analyzing the results from the experiments conducted using different methods, it appears that the first method yields the best outcomes. However, this does not necessarily indicate that the first method is the most effective overall.

It is crucial to consider how well the method generalizes to other systems and datasets. The first method, which relies on resources like the SNOMED CT knowledge base and the complete MIMIC-IV-Note dataset, may encounter issues when applied to less commonly used systems or datasets other than MIMIC-IV-Note. Additionally, extending this method to other languages, such as Spanish or Chinese, would be challenging. For instance, medical abbreviations or synonyms in these languages might need to be sourced from different databases, making the method less adaptable. On the other hand, the second method is more versatile and could be applied to other systems and datasets with minimal modifications. Moreover, if the necessary pretrained BERT models are available, this method could be easily adapted to annotate free text in languages other than English.

Regarding the running time, it is true that the first method requires significantly less time and computational resources to create the Kiri dictionaries compared to the time needed to train the BERT-based NER model in the second method. However, once both models are trained, the inference time is slightly shorter for the second method without considering the modifications introduced.

In addition, it is important to consider the broader context of deep learning advancements. As the field evolves, frequent improvements and increased computational power contribute to the development of more sophisticated methods. Currently, BERT models represent some of the state-of-the-art techniques in natural language processing. Yet, in a few years, emerging frameworks could surpass these existing models.

This means that deep learning approaches like the second method, which rely on large language models, could benefit significantly from future advancements. As more powerful large language models become available, they could enhance the performance of these methods, potentially achieving better results than current approaches, including the first method.

When developing predictive models, such as the two employed in this work, a key consideration is the risk of overfitting. Overfitting occurs when models become

overly tailored to the training data, leading to poor performance on testing data because they fail to generalize to new, unseen data. Nevertheless, in this particular case, overfitting is not a relevant concern for the following reasons:

In the first method, a dictionary is created that contains all the terms from the training data, along with the most common IDs associated with each term. This dictionary is further expanded with new terms, such as synonyms, permutations, and abbreviations. Similarly, the second method involves constructing a dictionary based on the most common IDs for each term in the annotations.

As a result, in both methods, the predictions for the training data are generated using these dictionaries. Since the prediction process for the training data differs from that of the testing data, the issue of overfitting is not directly applicable in this context.

6.1.2 Contribution

With respect to the contribution, the computation of the second score involves an additional step that slows down the overall inference process. Therefore, it is necessary to evaluate whether this extra step is worthwhile.

Using the contribution approach, the results show an increase in the macro-averaged intersection-over-union when $w_2 = 2$. However, no improvement is observed in either the F1-score or the macro-averaged F1-score, and the interpretation of these results is not immediately clear.

One aspect suggesting that the improvement might be relevant is that the metrics obtained when $w_1 = 0$, where only the second score from the contribution is considered, produce meaningful results rather than those corresponding to random predictions. Nonetheless, it is also necessary to consider that the candidates to which these scores are applied had already been filtered by selecting the top 10 candidates through re-ranking.

The IoU metric provides a more nuanced interpretation because it recognizes similar predictions if there is some overlap between them, while F1-related scores treat two similar predictions as just an error. However, it is still unclear whether the difference between the best IoU score achieved with the contribution and the IoU score of the original approach is significant enough.

There is something else that needs to be considered. The approach used in the contribution is sensitive to the number of samples from each ID. A concept that appears a few times will have a very small value of s_2, as $\#\{\text{T} : \text{ID} = i \cap \text{sec} = j\}$ will be very small, while $\#\{\text{T} : \text{sec} = j \cap \text{cat} = k\}$ will be very large. The contribution of s_2 could be useful, but it might be overshadowed by s_1. Increasing w_1 would likely not be beneficial, as it would overly emphasize the second score for IDs that appear more frequently.

Considering $\Pr(\text{sec} = j \mid \text{ID} = i)$ instead of $\Pr(\text{ID} = i \mid \text{sec} = j)$ might mitigate this issue, but it would still be problematic. For instance, an ID with only one sample appearing once in the entire dataset, and that sample occurring in a specific section, would benefit disproportionately from the second score compared to an ID appearing in 500 samples, 400 of which are in that section.

In reality, additional experiments with different datasets would be necessary to determine if there is a genuine improvement from combining the two scores. Conducting experiments on a more balanced dataset might provide clearer results and mitigate the problem of IDs that appear too infrequently.

6.2 Discussion of the Application's Results

In this work, a small-scale application of automated clinical coding has been demonstrated, and the results can be further discussed.

The results of the predictions using the regular expressions-based method for detecting vital signs from partial texts have been presented. However, it would have been necessary to evaluate this model on a set of texts using annotated test data to compute relevant metrics, such as accuracy measures. Unfortunately, due to the lack of data with test labels for the actual vital signs, this evaluation was not possible.

Despite the lack of test labels to accurately evaluate the model, the work highlights the importance of knowing where vital signs are likely to appear in order to extract them efficiently. When applied to full text, the model struggles to accurately extract vital sign values because they are mixed with other information. A more complex model would be required to identify vital signs using the full texts. Such a model, based on regular expressions or on deep learning, would likely need to first detect vital sign term identifiers, such as "VS" or "vital signs", before extracting the values, which is already what has been done in automatic clinical coding.

The results shown are promising. Nonetheless, the regular expressions-based model still makes some errors, such as including incorrect characters or confusing some numbers. An NLP deep learning model would likely have been a better choice and could have produced better results but, given the limited number of 313 samples, it was not feasible to apply a traditional machine learning or deep learning train-test approach.

Despite having used a relatively simple model that is not perfect, it could still be effectively applied to detect vital signs in texts from the same physicians, who typically have consistent writing styles. Nevertheless, to generalize to texts written by other specialists, a deep learning model trained on a larger dataset would be necessary.

Another consideration is that detecting vital signs is especially challenging, as there are five different values to identify, which can be expressed in numerous ways, and sometimes values may not be written in the text at all. A similar regular expressions approach could be developed to detect other types of numerical values associated with a SNOMED CT ID, such as doses of medicines or concentrations of substances, which might be much simpler and also useful.

6.3 Conclusions

In conclusion, the objectives of this work have been successfully accomplished.

Firstly, the significance of the SNOMED CT medical terminology structuring system, as well as the importance of structuring medical data, has been thoroughly understood and demonstrated. The structure and utility of the SNOMED CT knowledge base and the MIMIC-IV-Note dataset have also been comprehensively explained and utilized.

An extensive study of the state of the art in named entity recognition and entity linking has been conducted, with two methods examined in detail serving as effective examples of automated clinical coding techniques. These methods, each employing distinct approaches, one a classical dictionary-based method and the other a BERT deep learning-based model, have both produced strong results.

Moreover, an improvement inspired by the section-breaking technique of the first method was implemented in the second method. While this contribution shows

potential, further work and experiments are necessary to determine its significance and practical value.

This work has not only presented abstract models and results but also demonstrated a practical application of automated clinical coding in detecting vital sign values. Although there is room for improvement, this small application serves as an example of the vast importance and potential applications of this research field.

For future work, it would be beneficial to study, test, and compare different models, especially deep learning ones, to those explored in this study. It would also be worthwhile to test the improvement in a more balanced dataset or to propose and evaluate new enhancements to the tested models, such as expanding the dictionary-based method with additional terms from other sources or refining the deep learning-based method by experimenting with other types of large language models. Furthermore, it would be valuable to extend these and other automated clinical coding methods to systems beyond SNOMED CT and to a range of languages, including widely spoken ones like Spanish, as well as less commonly spoken but still important languages used by healthcare workers, such as Catalan.

Bibliography

Achara, Akshit, Sanand Sasidharan, et al. (2024). "Efficient Biomedical Entity Linking: Clinical Text Standardization with Low-Resource Techniques". In: *arXiv preprint arXiv:2405.15134*.

Al-Hablani, Bader (2017). "The use of automated SNOMED CT clinical coding in clinical decision support systems for preventive care". In: *Perspectives in health information management* 14.Winter.

American Medical Association (2024). *Current Procedural Terminology (CPT)*. Accessed: 2024-08-31. URL: https://www.ama-assn.org/amaone/cpt-current-procedural-terminology.

Bodenreider, Olivier (2004). "The unified medical language system (UMLS): integrating biomedical terminology". In: *Nucleic acids research* 32.suppl_1, pp. D267–D270.

Borchert, Florian and Matthieu-P Schapranow (2022). "HPI-DHC@ BioASQ DisTEMIST: Spanish Biomedical Entity Linking with Pre-trained Transformers and Cross-lingual Candidate Retrieval." In: *CLEF (Working Notes)*, pp. 244–258.

Cao, Pengfei et al. (2020). "HyperCore: Hyperbolic and co-graph representation for automatic ICD coding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3105–3114.

Centers for Medicare & Medicaid Services (2024). *Healthcare Common Procedure Coding System (HCPCS)*. Accessed: 2024-08-31. URL: https://www.cms.gov/Medicare/Coding/MedHCPCSGenInfo.

Cho, Hyejin and Hyunju Lee (2019). "Biomedical named entity recognition using deep neural networks with contextual information". In: *BMC bioinformatics* 20, pp. 1–11.

Devlin, Jacob (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Dong, Hang et al. (2022). "Automated clinical coding: what, why, and where we are?" In: *NPJ digital medicine* 5.1, p. 159.

Durango, María C, Ever A Torres-Silva, and Andrés Orozco-Duque (2023). "Named entity recognition in electronic health records: A methodological review". In: *Healthcare Informatics Research* 29.4, p. 286.

Falis, Matúš et al. (2019). "Ontological attention ensembles for capturing semantic concepts in ICD code prediction from clinical text". In: *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pp. 168–177.

Farkas, Richárd and György Szarvas (2008). "Automatic construction of rule-based ICD-9-CM coding systems". In: *BMC bioinformatics*. Vol. 9. Springer, pp. 1–9.

Gallego, Fernando et al. (2024). "Clinlinker: Medical entity linking of clinical concept mentions in spanish". In: *International Conference on Computational Science*. Springer, pp. 266–280.

Gaudet-Blavignac, Christophe et al. (2021). "Use of the systematized nomenclature of medicine clinical terms (SNOMED CT) for processing free text in health care: systematic scoping review". In: *Journal of medical Internet research* 23.1, e24594.

Guy Amit Yonatan Bilu, Irena Girshovitz and Chen Yanover (2024). *SNOMED CT Entity Linking Challenge 1st Place Solution*. URL: https://github.com/drivendataorg/snomed-ct-entity-linking.

Hardman, Will et al. (2023a). *SNOMED CT Entity Linking Challenge*. PhysioNet. Version 1.0.0. Published: Dec. 19, 2023. DOI: 10.13026/s48e-sp45. URL: https://doi.org/10.13026/s48e-sp45.

— (2023b). *SNOMED CT Entity Linking Challenge (version 1.0.0)*. https://doi.org/10.13026/s48e-sp45.

Hartendorp, Fons et al. (2024). "Biomedical Entity Linking for Dutch: Fine-tuning a Self-alignment BERT Model on an Automatically Generated Wikipedia Corpus". In: *arXiv preprint arXiv:2405.11941*.

Hristov, Anton et al. (2023). "Clinical Text Classification to SNOMED CT Codes Using Transformers Trained on Linked Open Medical Ontologies". In: *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pp. 519–526.

Huang, Chao-Wei, Shang-Chi Tsai, and Yun-Nung Chen (2022). "PLM-ICD: Automatic ICD coding with pretrained language models". In: *arXiv preprint arXiv:2207.05289*.

Ji, Shaoxiong, Matti Hölttä, and Pekka Marttinen (2021). "Does the magic of BERT apply to medical code assignment? A quantitative study". In: *Computers in biology and medicine* 139, p. 104998.

Ji, Shaoxiong et al. (2022). "A unified review of deep learning for automated medical coding". In: *ACM Computing Surveys*.

Jiang, Fei et al. (2017). "Artificial intelligence in healthcare: past, present and future". In: *Stroke and vascular neurology* 2.4.

Jin, Qiao et al. (2023). "MedCPT: Contrastive Pre-trained Transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval". In: *Bioinformatics* 39.11, btad651.

Johnson, Alistair et al. (2023a). *MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2)*. https://doi.org/10.13026/1n74-ne17.

Johnson, Alistair EW et al. (2023b). "MIMIC-IV, a freely accessible electronic health record dataset". In: *Scientific data* 10.1, p. 1.

Kim, Young-Min and Tae-Hoon Lee (2020). "Korean clinical entity recognition from diagnosis text using BERT". In: *BMC Medical Informatics and Decision Making* 20, pp. 1–9.

Kulyabin, Mikhail et al. (2024). "SNOBERT: A Benchmark for clinical notes entity linking in the SNOMED CT clinical terminology". In: *arXiv preprint arXiv:2405.16115*.

Lee, Jinhyuk et al. (2020). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4, pp. 1234–1240.

Li, Jing et al. (2020). "A survey on deep learning for named entity recognition". In: *IEEE transactions on knowledge and data engineering* 34.1, pp. 50–70.

Li, Luqi et al. (2019). "An attention-based deep learning model for clinical named entity recognition of Chinese electronic medical records". In: *BMC Medical Informatics and Decision Making* 19, pp. 1–11.

Liang, Jun et al. (2017). "A novel approach towards medical entity recognition in Chinese clinical text". In: *Journal of Healthcare Engineering* 2017.1, p. 4898963.

Lin, Yi-Feng et al. (2004). "A maximum entropy approach to biomedical named entity recognition". In: *Proceedings of the 4th International Conference on Data Mining in Bioinformatics*. Citeseer, pp. 56–61.

Nelson, Stuart J et al. (2011). "Normalized names for clinical drugs: RxNorm at 6 years". In: *Journal of the American Medical Informatics Association* 18.4, pp. 441–448.

Observational Health Data Sciences and Informatics (OHDSI) (2024). *Standardized Data: The OMOP Common Data Model*. URL: https://www.ohdsi.org/standardized-data/.

Pankhurst, Tanya et al. (2021). "Introduction of Systematized Nomenclature of Medicine–Clinical Terms Coding Into an Electronic Health Record and Evaluation of its Impact: Qualitative and Quantitative Study". In: *JMIR medical informatics* 9.11, e29532.

Patrick, Jon, Yefeng Wang, and Peter Budd (2007). "An automated system for conversion of clinical notes into SNOMED clinical terminology". In: *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*, pp. 219–226.

Regenstrief Institute (2024). *LOINC (Logical Observation Identifiers Names and Codes)*. Accessed: 2024-08-31. URL: https://loinc.org/.

Reyes-Aguillón, Javier et al. (2022). "Clinical Named Entity Recognition and Linking using BERT in Combination with Spanish Medical Embeddings." In: *CLEF (Working Notes)*, pp. 341–349.

Settles, Burr (2004). "Biomedical named entity recognition using conditional random fields and rich feature sets". In: *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP)*, pp. 107–110.

Shi, Haoran et al. (2017). "Towards automated ICD coding using deep learning". In: *arXiv preprint arXiv:1711.04075*.

SNOMED International (2024). *SNOMED CT Browser - Full Perspective*. https://browser.ihtsdotools.org/?perspective=full&conceptId1=404684003&edition=MAIN/2024-08-01&release=&languages=en. Accessed: 2024-06.

Sohrab, Mohammad Golam et al. (2020). "BENNERD: A neural named entity linking system for COVID-19". In: *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 182–188.

Teng, Fei et al. (2020). "Explainable prediction of medical codes with knowledge graphs". In: *Frontiers in bioengineering and biotechnology* 8, p. 867.

U.S. National Library of Medicine (2024). *SNOMED CT: Systematized Nomenclature of Medicine – Clinical Terms*. Accessed: 2024-06. U.S. National Library of Medicine. URL: https://www.nlm.nih.gov/healthit/snomedct/index.html.

Wang, Tingzhong et al. (2023). "A hybrid model based on deep convolutional network for medical named entity recognition". In: *Journal of Electrical and Computer Engineering* 2023.1, p. 8969144.

Wang, Yefeng and Jon Patrick (2009). "Cascading classifiers for named entity recognition in clinical notes". In: *Proceedings of the workshop on biomedical information extraction*, pp. 42–49.

World Health Organization (2019). *International Classification of Diseases for Mortality and Morbidity Statistics (ICD-11)*. 11th Revision. Geneva, Switzerland: World Health Organization. URL: https://icd.who.int/.

Wu, Yonghui et al. (2015). "Named entity recognition in Chinese clinical text using deep neural network". In: *MEDINFO 2015: eHealth-enabled Health*. IOS Press, pp. 624–628.

Wu, Yonghui et al. (2017). "Clinical named entity recognition using deep learning models". In: *AMIA annual symposium proceedings*. Vol. 2017. American Medical Informatics Association, p. 1812.

Xie, Xiancheng et al. (2019). "EHR coding with multi-scale feature attention and structured knowledge graph propagation". In: *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 649–658.

Xu, Jing et al. (2018). "Unsupervised medical entity recognition and linking in Chinese online medical text". In: *Journal of healthcare engineering* 2018.1, p. 2548537.

Zhou, Guodong et al. (2004). "Recognizing names in biomedical texts: a machine learning approach". In: *Bioinformatics* 20.7, pp. 1178–1190.

Zhou, Lingling et al. (2020). "Construction of a semi-automatic ICD-10 coding system". In: *BMC medical informatics and decision making* 20, pp. 1–12.