EMTE: An Enhanced Medical Terms Extractor using pattern matching rules

Monah Bou Hatoum [1], Jean Claude Charr [1,*], Alia Ghaddar [2,*], Christophe Guyeux [1,*], and David Laiymani [1,*]

[1]University of Bourgogne Franche-Comté, UBFC, CNRS, 90000 Belfort, France
[2]Department of Computer Science, International University of Beirut, Beirut P.O. Box 146404, Lebanon

Abstract—**Background**: Downstream tasks, like clinical textual data classification, perform best when given good quality datasets. Most of the existing clinical textual data preparation techniques rely on two main approaches, removing irrelevant data using cleansing techniques or extracting the valuable data using feature extraction techniques. Although both approaches showed promising results on experimental datasets, they still have limitations, mainly when applied to real-world datasets. The latter datasets are challenging because they might include incomplete, inconsistent and heterogeneous records. Therefore, understanding the nature of the data is vital in order to improve its quality before its transmission to the classification model.

Methods: This paper proposes a cleansing approach (called *EMTE*) which extracts phrases (medical terms, abbreviations, and negations) using pattern-matching rules based on the linguistic processing of the clinical textual data. Without requiring any training, *EMTE* extracts the valuable medical data from clinical textual records even if they have different writing styles. Furthermore, since *EMTE* relies on dictionaries to store abbreviations and pattern-matching rules to detect the abbreviations, negations, and medical terms, it can be easily maintained and extended for industrial use.

Experiments: To evaluate the performance of our approach, it was compared to three other techniques. All four cleansing techniques were applied to a large industrial imbalanced dataset, consisting of 2.21M samples from different specialties with 1,050 ICD-10 codes. The outputs of these cleansing techniques were fed to several Deep Neural Network (DNN) algorithms to solve the ICD-10 (International Classification of Diseases, 10^{th} edition) multi-label classification problem. In this problem, the clinical textual samples are the input data, and the ICD-10 codes are the actual labels.

Results: The experimental results on several Deep Neural Network (DNN) algorithms showed that our cleansing approach significantly improves the trained models' performance when compared to the other tested techniques and according to different metrics.

Index Terms—Deep Learning, Natural Language Processing (NLP), Computer-Aid Diagnosis, Chief Complaints, Text mining, Abbreviations, Negations, Phrases.

1 Introduction

The International Classification of Diseases, the 10^{th} edition (ICD-10), is a standard tool to classify disease diagnoses from a patient's medical signs, symptoms, and other health conditions. Governments, Health Insurance Companies, and Healthcare providers also use the ICD codes to report and communicate patients' medical cases. ICD-10 codes are hierarchical alphanumeric labels with a length between three to seven characters depending on the depth of the hierarchy and the disease's specificity level. The specificity of the ICD-10 codes is crucial and controls which treatment plan the physicians follow on the patients. For example, in Table 1, *"K40.21"* does not require a direct surgical intervention while *"K40.11"* requires an immediate surgical intervention.

On the other hand, physicians usually manually assign one or multiple ICD-10 codes to describe the patient's illness and symptoms during every patient visit. However, this manual operation is time-consuming and error-prone due to the large available number of ICD-10 codes. Consequently, hospitals and medical institutes are motivated to turn to auto-diagnosis tools. In recent years, researchers have provided several approaches that tackle different problems in healthcare, including ICD-10 prediction from clinical textual data, extracting medical terms using feature extraction techniques, and cleansing approaches to remove the irrelevant data and purify the input data.

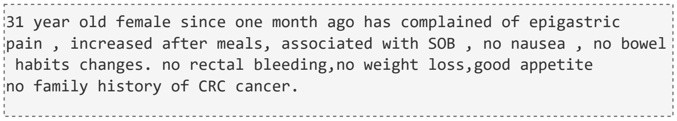
Unfortunately, most of these existing approaches have limitations and show poor results when applied to industrial datasets [1]. The significant limitations of these approaches are: studying part of the ICD-10 codes without preserving the specificity of the codes, applying data preprocessing without studying its impact on the data, and using small datasets to train models to extract features from clinical textual data without studying other essential features like the specialty [2], [3]. We believe the main reason behind this gap is the lack of knowledge about the nature of the data. Clinical textual data are diverse, incomplete, and redundant. They includes abbreviations, periods, negations, and terms of one or more words. Also, these data have different writing styles. For example, negations could appear on the left side like *"no rectal bleeding"* or the right side like *"smoker: no"*. In addition, a medical term could appear with its expanded form (*"long-term"*) *"Diabetes Mellitus"* or using an abbreviated form (*"short-term"*) *"DM"*. For all these reasons, extracting valuable information from medical datasets is complex, and it is essential to understand how physicians encode the medical phrases and the different writing styles available.

This study aims to improve the ICD-10 prediction performance by improving the data quality. In particular, we provide a preprocessing data approach for clinical textual data that

monah.bou_hatoum@univ-fcomte.fr
jean-claude.charr@univ-fcomte.fr
alia.ghaddar@liu.edu.lb
christophe.guyeux@univ-fcomte.fr
david.laiymani@univ-fcomte.fr

Code	Description
ICD-10-AM	International Classification of Diseases (10^{th} edition, Australian Modification)
Chapter XI	Diseases Of the digestive system (K00-K93)
K40-K46	Hernia
K40	Inguinal hernia
K40.0	Bilateral inguinal hernia, with obstruction, without gangrene
K40.00	Bilateral inguinal hernia, with obstruction, without gangrene, not specified as recurrent
K40.01	Bilateral inguinal hernia, with obstruction, without gangrene, recurrent
K40.1	Bilateral inguinal hernia, with gangrene
K40.10	Bilateral inguinal hernia, with gangrene, not specified as recurrent
K40.11	Bilateral inguinal hernia, with gangrene, recurrent
K40.2	Bilateral inguinal hernia, without obstruction or gangrene
K40.20	Bilateral inguinal hernia, without obstruction or gangrene, not specified as recurrent
K40.21	Bilateral inguinal hernia, without obstruction or gangrene, recurrent

Table 1. The hierarchical presentation of the *"K40"* category in the ICD-10 codes with their description.



```
31 year old female since one month ago has complained of epigastric
pain , increased after meals, associated with SOB , no nausea , no bowel
 habits changes. no rectal bleeding,no weight loss,good appetite
no family history of CRC cancer.
```

Figure 1: An example of unprocessed chief complaint that contains abbreviations like *"SOB"*, *"CRC"* and negations like *"no weight loss"*, *"no nausea"* and terms like *"epigastric pain"*.

enhances the quality of data fed to state-of-art NLP models, while preserving the maximum information possible. Our approach employs the power of Part of Speech (POS) tagging [4] and the pattern-matching rules (PMRs) to extract valuable data and eliminate irrelevant data. With the help of a medical team, we built the pattern-matching rules to extract phrases (negations with their different writing styles, abbreviations, and medical terms). In this work, the clinical textual data consists of chief complaints (CC) and History of Presence Illness (HPI) written by physicians as depicted in Figure 1. The main contributions of this study can be summarized as follows:

- Provide a cleansing approach for clinical textual data using dictionary based pattern-matching rules. Our approach extracts phrases (negations and medical terms) from clinical textual data and combines the detected words as one medical phrase.
- Replace the short-term abbreviations with their full-term forms, considering the specialties as an additional feature to reduce abbreviation ambiguity.
- Improve negation detection covering different writing styles.
- Provide a comparative study that uses a large industrial dataset with 2.21M samples and 1,050 ICD-10 codes and shows that our cleansing approach has a better impact on the performance of various NLP models for solving the multi-label ICD-10 classification problem than three different existing cleansing techniques.

The rest of this paper is organized as follows: Section 3 presents some of the state-of-the-art text cleansing techniques and feature extraction along with their shortcomings. Our cleansing approach is detailed in Section 4. The results of the comparative study between our approach and other cleansing methods are exposed in Section 5. Section 6 details our findings and recommendations to efficiently cleanse medical text data. This article ends with a summary of the contributions, and some future works are outlined.

2 Background

This section presents a brief background about Natural Language Processing (NLP) and the significant challenges to overcome when cleansing clinical textual data.

2.1 POS tagging

Part of Speech tagging is a Natural Language Processing (NLP) process in which every token in the text is assigned a grammatical tag based on its definition and context. A token in a text might be a word, punctuation or space. *"POS"* tags contain several labels like *"ADJ"* which stands for *"Adjective"* and *"PRON"* stands for *"Pronoun"* [5]. In addition, Dependency Parsing (*DEP*) [4] is another NLP process that builds relations between the words in the text, based on the *POS* tags [6]. Many available tools, like *CoreNLP* [7], *Spacy* [8], *Gensim* [9], and *NLTK* [10], use Machine Learning algorithms to offer many NLP functionalities like tokenization, POS tagging, sentence segmentation, dependency parsing [8], [7], and entity recognition.

As an example, Table 2 shows a part of the linguistic features output of the chief complaint show in Figure 1, generated by the tool *"Spacy"* [8]. It shows the tokenization *TEXT*, the lemmatization of the token *LEMMA*, *POS*, *TAG*, and *DEP* for every token (word, punctuation, space) in the chief complaint. This text tagging can be used to build pattern-matching rules to discover negations, multi-words phrases, etc.

2.2 Abbreviations in Healthcare

Abbreviations are frequently used in healthcare to reduce time and typos. However, they create a significant challenge for the machine learning techniques due to their ambiguity. Indeed, the same abbreviation could have different meanings depending on the context and the specialty. When analyzing clinical text, three types of abbreviations can be found:

i) General abbreviations that have a common meaning, such as *"dx"* which stands for *"diagnosis"* or *"c/o"* which stands for *"complaining of"*.

ii) Specialty-specific abbreviations, where the meaning differs from one specialty to another, such as *"CLD"* which stands for *"Chronic Liver Disease"* in the Gastroenterology department and *"Chronic Lung Disease"* in the Pulmonary unit. Similarly, *"MS"* is the abbreviation of *"Multiple Sclerosis"* in the Neurology department and *"Mitral Stenosis"* in the Cardiology and Radiology departments.

iii) An Ambiguous abbreviation that has a contextual related meaning, such as *"LFT"*, which either stands for *"Lung Function Test"* or *"Liver Function Test"*.

Many studies have tackled the abbreviation ambiguity using supervised [11], [12], [13], [14] and unsupervised [15], [16] machine learning approaches. However, most of these studies have limitations. They were only trained on clinical textual data without taking other essential parameters like the specialty into consideration. Therefore, these approaches perform poorly on multi-specialty large datasets [17] because they miss many abbreviations or suggest a wrong expanded form of the abbreviated term.

2.3 Negation detection

In clinical documents, the terms *"no", "nil", "absence of", "negative", "n't","-ve"* are often used for negation. Hence, removing negations drastically changes the semantics and the interpretation of clinical notes. For instance, removing the word *"no"* from *"patient has no cancer"* will completely change its meaning. Also, removing the punctuation "−" or "+" from *("-ve" or "+ve")* changes the meaning from *("negative" or "positive")* into *"ve"* which refers to *"vaginal examination"* in the *"Obstetrics and Gynecology"* department and *"ventricular extrasystoles"* in the *Cardiology* department. It is important to examine the negation indicators beside a word entity and merge data with semantic spaces to appropriately detect a negation. Moreover, physicians could write the negations in different forms such as: *Non smoker*, *doesn't smoke*, *smoker: no*, or *smoker: nil* which increases the complexity of detecting the negations. Most of the existing solutions rely on the Dependency Relation (DEP) like in [18], [19]. Unfortunately, most of the existing negation detectors fail to detect all the potential negations since the DEP process cannot handle all the negations writing styles in clinical textual data [20].

3 Related work

3.1 Existing Cleansing techniques

Many research studies analyzed clinical textual data using machine learning techniques. In some of these studies [21], [22], [23], researchers trained the models directly on the raw data without any preprocessing. They only relied on the power of the machine learning techniques, like deep neural networks (DNN), to discover the relationship among the data. The main limitations of such approaches are increasing the training complexity and the dimensional space, potentially leading to over-fitting problems and low testing accuracy [24].

On the other hand, other studies [25], [26], [27], [28] have applied standard cleansing (*SC*) techniques like stemming, lemmatization, stop-words removal, and punctuation removal. Unfortunately, these preprocessing steps reduce the data quality instead of improving it. For example, removing stop-words like *"no", "has", "none", "not"* change the meaning of the input data. As an example, both complaints *"a patient has a colon cancer for six months complaining from severe abdominal pain"* and *"a patient with severe abdominal pain, no colon cancer in family history"* would have the same meaning if *has* and *no* were removed. Moreover, removing punctuations from clinical textual data increases the challenge of detecting the proper abbreviations and distinguishing between dates and numbers.

3.2 Feature Extraction

Feature Extraction (FE) and Named Entity Recognition (NER) using machine learning are two tasks in Natural Language Processing (NLP) that were widely used in the last few years. The former is a process of identifying and extracting important characteristics from data, while the latter identifies and classifies named entities in text. In healthcare, extracting all essential features from data requires enormous resources and is time-consuming due to the high dimensionality of the data. FE and NER came into play to help identify the relevant data such as diseases, treatments, abbreviations, and symptoms; this helps reduce the vocabulary size and hyperspace dimension of the data. However, these approaches require massive labeled data for training using the supervised approaches [29]. Moreover, the diversity of data makes these approaches inefficient for both supervised and unsupervised approaches [30], [1]. In addition, the generated pre-trained models from these approaches require large efforts to maintain and update them to capture the new terms and to fix wrongly predicted entities, which is time-consuming. Unfortunately, with all these limitations, many of the existing FE and NER approaches are not ready for healthcare industrial use where the data are massive and complex.

On the other hand, other feature extraction approaches use pattern-matching rules (PMR). PMRs are a set of rules manually written to identify patterns using lexico-syntactic patterns to identify the occurrence of similar entities in NLP. PMRs are widely used for financial topics [31]. Unfortunately, rare research topics investigated these approaches in healthcare because since 2018 they mainly concentrated their work on machine learning techniques [32]. Unlike the feature extraction tools using machine learning techniques, PMRs are easy and faster to develop; they do not require labeled datasets or downstream tasks.

3.3 Word embeddings

Word Embedding is a technique used in natural language processing (NLP) that represents words in a vector form. Many techniques are available for word embeddings such as *Bag-of-Words [33],* Word2Vec "W2V" *[34],* Glove *[35],* Text Vectorization using Padding sequences "PS" *[36],* Sentence2Vec "S2V" *[37],* BERT *[38],* ELMo *[39]*. Word Embedding is an essential step for converting the textual data into numerical representation for proceeding with the downstream tasks.

This paper aims to provide an approach for improving the data quality from the data preparation phase without losing vital information and considering crucial industrial requirements such as flexibility and maintainability.

4 Materials and Methods

EMTE (Enhanced Medical Terms Extractor) is an approach that extracts phrases and eliminates irrelevant data from

TEXT	LEMMA	POS	TAG	DEP	TEXT	LEMMA	POS	TAG	DEP
31	31	NUM	CD	nummod	,	,	PUNCT	,	punct
year	year	NOUN	NN	npadvmod	no	no	DET	DT	neg
old	old	ADJ	JJ	amod	bowel	bowel	NOUN	NN	compound
female	female	NOUN	NN	nsubj			SPACE	_SP	dep
since	since	SCONJ	IN	prep	habits	habit	NOUN	NNS	compound
one	one	NUM	CD	nummod	changes	change	NOUN	NNS	conj
month	month	NOUN	NN	npadvmod	.	.	PUNCT	.	punct
ago	ago	ADV	RB	pcomp	no	no	DET	DT	neg
has	have	AUX	VBZ	aux	rectal	rectal	ADJ	JJ	amod
complained	complain	VERB	VBN	ROOT	bleeding	bleeding	NOUN	NN	ROOT
of	of	ADP	IN	prep	,	,	PUNCT	,	punct
epigastric	epigastric	ADJ	JJ	amod	no	no	DET	DT	neg
pain	pain	NOUN	NN	pobj	weight	weight	NOUN	NN	compound

Table 2. Tokenization, Lemmatization, POS tagging, and dependency parsing result of the chief complaint in Figure 1 using *Spacy*.

the clinical textual data. A phrase is a set of one or more tokens that could be abbreviations, negations, medical terms, other conditions, signs, and symptoms. A token is a word, punctuation, or number. For example, *"no rectal bleeding"* is a phrase while *"no", "rectal", and "bleeding"* are tokens.

EMTE depends on PMRs encoded using a combination of the linguistic features (POS, TAG, LEMMA, and DEP) to detect phrases from clinical textual data. In addition, it relies on JSON dictionaries to store the phrases' PMRs. Unlike the pre-trained models, JSON dictionaries are simple, flexible, and maintainable, which are desirable solutions for industrial use. *EMTE* has four main phases: (1) extraction of tokens from every chief complaint, (2) load dictionary rules, (3) extract relevant phrases, (4) generate new processed chief complaints. The dictionaries and algorithms used by EMTE are presented in the next subsections.

4.1 Dictionaries and rules

The main objective of this paper is to detect the medical terms, abbreviations and negations in chief complaints during the cleansing phase using pattern matching rules. It should improve the performance of the machine learning models applied on the cleansed data. For example, detecting the abbreviations and replacing them with their full-terms should reduce the vocabulary size and the hyperspace dimension.

The PMRs were developed as follows: First, with the help of a medical team and after analyzing many chief complaints and discharge summaries, the different structures of clinical terms (length and syntactical orders) were enumerated. Second, the discovered structures were translated into linguistic keywords (VERB, ADV, NOUN, PRONOUN, NEG, and LEMMA). Finally, the patter-matching rules to detect these structures were developed using the *Spacy* syntax for the sake of experiments. The resulting PMRs and the abbreviations were stored in two JSON dictionaries denoted, $\mathcal{R}$ and $\mathcal{A}$ respectively.

4.1.1 Abbreviation rules

As mentioned in Section 2.2, the existing solutions have limitations and do not clarify the ambiguity of abbreviations. To reduce the abbreviations ambiguity, the physician's specialty was considered while processing the clinical textual data. Furthermore, to ensure flexibility and maintainability, the abbreviation dictionary stores both general and specific abbreviations with their corresponding set of specialties.

```
    [{
        "short": ["dx","diag"],
        "full": "diagnosis",
        "specialties": []
    },{
        "short": "pmh",
        "full": "past medical history",
        "specialties": []
    }, {
        "short": ["cs","c/s","c/sec", "c.s."],
        "full" :"caesarean section",
        "specialties":["Obstetrics and Gynecology"]
    }, {
        "short": ["cs","c/s"],
        "full" :"compartment syndrome",
        "specialties":["orthopedics"]
    }, {
        "short": ["hvs"],
        "full" :"hyperventilation syndrome",
        "specialties":["emergency","pulmonary"]
    }, {
        "short": ["hvs"]],
        "full" :"high vaginal swab",
        "specialties":["Obstetrics and Gynecology"
    }]
```

Listing 1: A sample of the abbreviations dictionary in JSON.

Listing 1 shows a few entries in the JSON abbreviation dictionary. Every entry represents an abbreviation with three attributes: (*"short", "specialties", and "full"*). The key *"short"* stores the list of possible short-terms of the abbreviation, like *"dx"* and *"pmh"*. The key *"full"* corresponds to the full expansion of the abbreviated term. For example, *"hyperventilation syndrome"* is the full-term of *"hvs"*.

Finally, the key *"specialties"* stores the list of specialties where the abbreviated term can be used without ambiguity. For example, the abbreviation *"hvs"* in Listing 1 has the same meaning in the *"Emergency and Pulmonary"* departments and can be used with no ambiguity. On the other hand, it has a different meaning when used in the *"Obstetrics and Gynecology"* department. For this reason, a second entry for this abbreviation was added to the dictionary with the *"Obstetrics and Gynecology"* specialties. It must be noted that key *"specialties"* could be empty if the abbreviation is a general non-ambiguous term. For instance, it is empty for the abbreviation "dx" because it has the same meaning, *"diagnosis"*, in all specialties.

```
[{
    "type": "negation"
    "rule": {"label": "negation", "pattern":[[{’POS’
        :’NOUN’},{’IS_PUNCT’: True},{’DEP’:’neg’}],
    [{’POS’:’NOUN’},{’IS_SPACE’: True},{’DEP’:’neg’}
        ]]}
},{
    "type": "negation"
    "rule": {"label": "negation", "pattern":[[{’DEP’
        :’neg’},{’POS’: ’NOUN’},{’POS’:’NOUN’},{’
        POS’:’NOUN’}]]}
},{
    "type": "term",
    "rule": {"label": "gender", "pattern":[{’LEMMA’:
        {’IN’:["girl","boy","man","woman","lady","
        guy","female","male"]}}]}
},{
    "type": "term",
    "rule": {"label": "entity", "pattern":[{’POS’:’
        ADJ’},{’POS’:’NOUN’},{’POS’:’NOUN’}]}
},{
    "type": "term",
    "rule": {"label": "entity", "pattern":[{’POS’:’
        ADJ’},{’POS’:’ADJ’},{’POS’:’NOUN’}]}
}]
```

Listing 2: A sample of the PMRs dictionary.

4.1.2 Negations and medical terms rules

Physicians use medical terms and negations in different ways. For example, they may use the full-terms (e.g., *past medical history*), or the short-terms (e.g., *pmh*). They may also put the negations on the left side (e.g., *no pmh*), or the right side (e.g. *pmh: no*). Fortunately, these different writing styles follow some patterns, which can be captured using PMRs. For example, the medical term, *"epigastric pain"*, was detected using a rule that catches the pattern: *"Adj"* followed by a *"Noun"*. If for the same phrase more than one rule can be applied, the one with the most tokens is applied. For example, the rule that detects the negation, *"no epigastric pain"* consisting of three tokens, is applied instead of the one that just detects the medical term, *"epigastric pain"* consisting of just two tokens. If two rules concern the same number of tokens, the priority is given to the negation rule, otherwise the first rule is selected. Figure 2 shows the detected phrases from the raw chief complaint, presented in Figure 1, using *EMTE*.

Listing 2 shows a part of the rules dictionary. These rules were built with the help of a medical team after analyzing the different writing styles of physicians working in a multinational Saudi private hospital. Every entry in the JSON represents a rule definition that contains two attributes (*"type" and "rule"*). The first key represents the type of the rule, which is either a *"negation"* or a *"medical term"*. The second attribute stores the pattern-matching rule. For example, the second entry in Listing 2 detects a negation phrase formed of a negation determiner followed by three nouns, such as *"no bowel habits changes"*. *EMTE* can also detect the words that start with the "non" prefix, such as "nonsmoker", "non-stick" and "nonfat". These words are replaced by the following form: "non smoker" " non stick" and "non fat". Splitting these words helps in unifying the terms and reducing the vocabulary size.

4.2 Preliminary definitions and notations

This subsection defines the notations used in the presentation of the EMTE approach.

Let $\Sigma = \{\sigma_j\}_{j=1}^{z}$ be the set of the z available specialties. Let $\mathcal{C} = \{S_i\}_{i=1}^{n}$ be a raw data corpus consisting of n samples, S_i. Every $S_i \in \mathcal{C}$ is a tuple with three attributes, $S_i = (\tau_i, \sigma_i, \lambda_i)$. S_i contains the chief complaint text τ_i, the specialty $\sigma_i \in \Sigma$, and the set of true labels λ_i (i.e. *"ICD-10 codes"*). Also, let Γ be the annotation function that splits the chief complaint τ_i into tokens (word, punctuation, space) and applies POS tagging on the resulting tokens. Finally, let Ψ be the parser function that applies a set of rules on a given set of tokens.

$Token$: A chief complaint τ_i contains a set of tokens $\{t_{ij}\}_{j=1}^{k}$ where k is the number of tokens. Every token t_{ij} contains five attributes ($text_{ij}$, $lemma_{ij}$, pos_{ij}, tag_{ij}, dep_{ij}) where $text_{ij}$ is the splitted token, $lemma_{ij}$ is the lemmetization of t_{ij} and pos_{ij}, tag_{ij} and dep_{ij}) represent the linguistic features of t_{ij}.

$Abbreviation$: Every abbreviation d_j is a tuple containing three attributes, $d_j = (\alpha_j, \delta_j, \epsilon_j)$, where α_j contains the short-terms of the abbreviation, δ_j corresponds to the full-term (i.e the expanded form) of the abbreviation, and ϵ_j stores the list of specialties where the abbreviation α_j can be used without ambiguity. ϵ_j is empty if α_j is a general abbreviation. Thus, $\epsilon_j \subset \{\phi\} \cup \{\sigma_p \mid \sigma_p \in \Sigma\}_{p=1}^{l}$ where l is the number of allowed specialties for the given abbreviation α_j.

$Rule$: A rule can be applied to detect negations or medical terms. Thus, every rule r has two attributes, i.e. $r = (e, p)$, where e is the type of the rule and p is the pattern-matching rule.

$Dictionaries$: The PMRs and abbreviations are stored in two JSON dictionaries denoted, $\mathcal{A}$ and $\mathcal{R}$ respectively. $\mathcal{A} = \{d_j\}_{j=1}^{s}$ is a dictionary of s abbreviations d_j. While, $\mathcal{R} = \{r_j\}_{j=1}^{u}$ is the list of all negations and medical terms rules, r_j. It is worth mentioning that the PMRs are independent from the specialties.

$Phrase$: Every phrase m_j has two attributes, the "$label$" from the available set of labels (*"gender", "negation", "term", "period", and "abbreviation"*) and the "$phrase$", which is a set of detected tokens $\{t_{jp}\}_{p=1}^{c}$ of size $c \leq k$. The labels are used for reporting and tracing purposes.

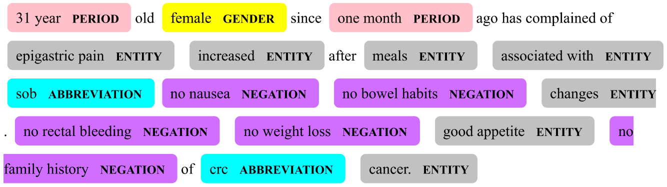
Let $\Psi : (\gamma, \omega) \mapsto \{m_j\}_{j=1}^{b}$ be the parser function that parses linguistically annotated set of tokens γ based on the set of rules ω. The Ψ function generates b phrases.

For instance, $\psi_i = \Psi(\gamma_i, \omega_i) = \{m_{ij}\}_{j=1}^{b}$ are the set of detected phrases from sample S_i.

4.3 Steps of *EMTE* approach

Our aim is to apply the PMRs on the corpus in order to extract the relevant phrases from every chief complaint τ_i. Algorithm 1 shows the pseudo-code of *EMTE* which takes the abbreviations dictionary $\mathcal{A}$, the negations and terms dictionary $\mathcal{R}$, and the corpus $\mathcal{C}$ as inputs. The algorithm returns the processed corpus $\mathcal{C}'$. *EMTE* loops on all $S_i \in \mathcal{C}$ and performs several steps as follows:

- Extract all the tokens from every chief complaint: First, *EMTE* splits every τ_i into tokens. It uses the linguistic annotator method $\Gamma : \tau \mapsto \gamma$, which is available in many NLP tools [8], [7], [10]. The method returns a set of tokens and generates their linguistic features. Thus, for every chief



31 year PERIOD old female GENDER since one month PERIOD ago has complained of
epigastric pain ENTITY increased ENTITY after meals ENTITY associated with ENTITY
sob ABBREVIATION no nausea NEGATION no bowel habits NEGATION changes ENTITY
. no rectal bleeding NEGATION no weight loss NEGATION good appetite ENTITY no
family history NEGATION of crc ABBREVIATION cancer. ENTITY

Figure 2: The detected phrases of chief complaint sample in Figure 1 using our approach *EMTE*.

complaint τ_i, $\gamma_i = \Gamma(\tau_i) = \{t_{ij}\}_{j=1}^{w_i}$ is the set of w_i tokens in τ_i.

- Generate and load the dictionary rules: Since some abbreviations might depend of the specialties, the abbreviation PMRs are built for every sample S_i according to the specialty σ_i. Therefore, *EMTE* first finds all the abbreviations' short-terms μ_i that satisfy the specialty σ_i in sample S_i. Then, it auto-generates the PMRs ρ_i that are specific to this specialty:

$$\rho_i = \{t_{ij} \mid lemma_{ij} \in \mu_i\}_{j=1}^{k}$$

where,

$$\mu_i = \bigcup\{\alpha_j \mid \epsilon_j = \phi \vee \sigma_i \in \epsilon_j\}_{j=1}^{v}$$

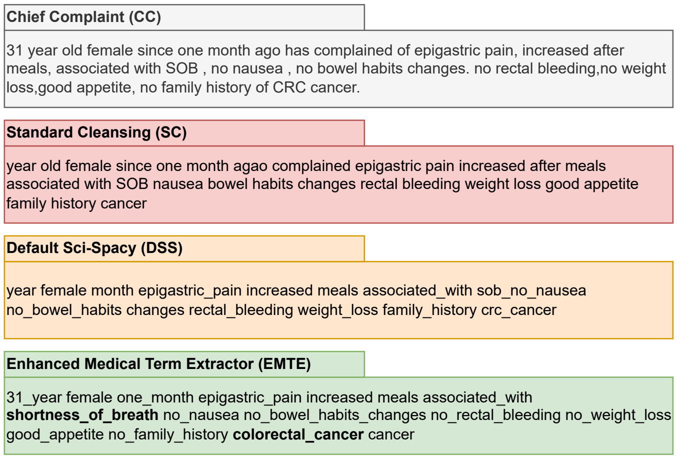
- Extract relevant phrases: The previously generated rules are used to extract phrases from each clinical text. Let $\omega_i = \mathcal{R} \cup \{\rho_i\}$ be the set of PMRs to be applied on the tokens γ_i for each sample S_i. The parser $\Psi(\gamma_i, \omega_i)$ is called to generate ψ_i, the list of all detected phrases.

Figure 3 shows the result of all pattern matching rules in action. For example, *shortness_of_breath* is the result of the abbreviation pattern matching rule that detected the abbreviation *SOB*, and replaced it with its full term while replacing the spaces with underscores. Also, *no_nausea* is a result of a negation rule that detects the left side negations.

Algorithm 1 *EMTE* algorithm

```
Input: $\mathcal{A}$, $\mathcal{R}$, $\mathcal{C}$
Output: $\mathcal{C}'$ (the processed version of corpus $\mathcal{C}$)
Initialize $\mathcal{C}' \leftarrow \phi$
for each sample $S_i \in \mathcal{C}$ do
    Annotate $S_i$ to build the POS tagging: $\gamma_i \leftarrow \Gamma_i(S_i)$
    From $\mathcal{A}$, load into $\rho_i$ all the abbreviation rules having an empty specialties attribute or containing $\sigma_i$
    Apply the abbreviation and medical terms detection rules $\omega_i = \mathcal{R} \cup \{\rho_i\}$ on the annotated document $\gamma_i$
    Replace the detected abbreviations with their full-term
    Convert the detected phrases to words by merging their tokens with underscores
    $\mathcal{C}' \leftarrow \mathcal{C}' \cup S_i$
end for
return $\mathcal{C}'$
```

- Generate the new corpus: *EMTE* converts every detected phrase m_{ij} that has a label *"abbreviation"* from its short-term into its full-term representation and obtains the updated sample S_i'. Then, it merges the tokens of every detected phrase using underscores to form one word as shown in Figure 3. Finally, it reconstructs the sample S_i' using the detected phrases and adds the processed sample S_i' to $\mathcal{C}'$, the new processed corpus.



Chief Complaint (CC)

31 year old female since one month ago has complained of epigastric pain, increased after meals, associated with SOB , no nausea , no bowel habits changes. no rectal bleeding,no weight loss,good appetite, no family history of CRC cancer.

Standard Cleansing (SC)

year old female since one month agao complained epigastric pain increased after meals associated with SOB nausea bowel habits changes rectal bleeding weight loss good appetite family history cancer

Default Sci-Spacy (DSS)

year female month epigastric_pain increased meals associated_with sob_no_nausea no_bowel_habits changes rectal_bleeding weight_loss family_history crc_cancer

Enhanced Medical Term Extractor (EMTE)

31_year female one_month epigastric_pain increased meals associated_with **shortness_of_breath** no_nausea no_bowel_habits_changes no_rectal_bleeding no_weight_loss good_appetite no_family_history **colorectal_cancer** cancer

Figure 3: A chief complaint sample before and after using the cleansing methods *SC*, *DSS*, and *EMTE*

5 Experiments and Results

To evaluate the performance of our approach, a large clinical textual dataset (Chief Complaints and History of Presence Illness) was cleansed using four cleansing methods including our approach. The resulting datasets were fed to different machine learning models to solve the ICD multi-label classification problem.

Besides *"EMTE"*, the following cleansing methods were considered:

- *RAW*: no cleansing techniques were applied to the original data.
- *SC*: the standard cleansing steps, such as lemmatization, stemming, stop-words removal and punctuation removal were applied on the *RAW* data.
- *"DSS"*: it is based on the *"SciSpacy"* NER pre-trained model that extracts medical terms from the *RAW* data.

Each one of the four cleansed datasets was fed to the following word embedding techniques: Padding Sequence (*"PS"* [36]), Sentence2Vec (*"S2V"* [37]), and BERT-based word embeddings (*"Clinical_BERT"* [40], *"BERT_base"* [38]) as shown in Figure 4.

5.1 Industrial Medical data

The experiments were applied to medical data retrieved from the outpatient departments of a private Saudi hospital. The data covers three years and consists of anonymous records. Each record corresponds to a patient's visit and contains the chief complaints CCs (textual data), the list of diagnoses (represented by ICD-10 codes), and the physician's specialty. The imbalanced dataset included samples from 24 specialties such as Pediatrics, Internal Medicine, Gastroenterology, Cardiology, and Oncology. The data consisted of over 2.21M records with 1,050 different ICD-10 codes.

5.2 Tools and Technical Challenges

To implement these experiments, the *"SciSpacy"* NER tool [41] which is an extension from *"Spacy"* [8], was used. It contains a *NER* pre-trained model *"en_core_sci_lg"*, consisting of around 785k vocabulary and trained on biomedical data. This model generated the *"DSS"* corpus. Moreover, the *"EntityRuler"* and *"PatternMatcher"* components of *"Spacy"* were used in the *EMTE* approach to implement our PMRs and execute them. In addition, the Deep Learning training tasks were based on the *Keras* [42] and *Tensorflow* [43] libraries. Table 3 shows the hyper parameters used in these experiments.

Hyper parameters	
Optimizer	Adam
Loss function	Binary Cross Entropy
Batch Size	64
Learning Rate	$3e^{-4}$ or $5e^{-5}$
Threshold	0.5
Monitor	val_micro_F1
Epsilon	$1e^{-8}$
Neurons	2/3 Input + Output
Patience	10
Minimum Delta	$1e-3$
Maximum Epochs	200
Dropout	0.3

Table 3. The hyper parameters settings used with the *Keras* deep learning training tasks.

Since the *"BERT Tokenizer"* splits words into chunks and subwords if they do not belong to the size limited *BERT* vocabulary, it was modified to let the *BERT* embeddings work with the medical phrases detected by *EMTE*.

5.3 Results

In this section, we show how *EMTE* outperforms both cleansing methods, *DSS* and *SC*, in terms of features extraction. In addition, we compare the impact of our cleansing approach to the other considered cleansing methods in improving the ICD-10 multi-label classification using four machine learning models.

Method	Abbreviations		Negations		Medical Terms	
	Total	Samples	Total	Samples	Total	Samples
Gold Truth	2,540	575	1,329	359	4,891	1,000
DSS	1,493 **(58.79%)**	411 **(71.48%)**	915 **(68.85%)**	312 **(86.91%)**	3,561 **(72.81%)**	904 **(90.40%)**
EMTE	2,523 **(99.33%)**	566 **(98.43%)**	1,314 **(98.87%)**	359 **(100%)**	4,843 **(99.02%)**	1,000 **(100%)**

Table 4. A qualitative comparison between the outputs of *EMTE* and *DSS* when applied on a small dataset of 1,000 samples. *EMTE* detected 99.33% of the abbreviations and 98.87% of the negations.

5.3.1 Qualitative Analysis

Figure 3 shows the outputs of *EMTE*, *DSS* and *SC* when applied to a given chief complaint. *SC* approach removed the determiner *"no"*, which is a negation, thus, it changed the meaning of the input data. Moreover, this method fails to detect the medical terms (phrases). *DSS* extracted many medical terms, but some were inaccurate. For example, DSS detected *"no bowel habits"* instead of *"no bowel habits changes"*. It also combined the medical abbreviation *"SOB"* that stands for *"shortness of breath"*, with the *"no nausea"* term. Moreover, it inaccurately identified *"weight loss"* instead of detecting its negation.

On the other hand, with the help of the pattern-matching rules, our approach could identify most medical terms, negations, and abbreviations. For example, the medical abbreviations *"SOB"* and *"CRC"* were correctly replaced with their correct full-terms.

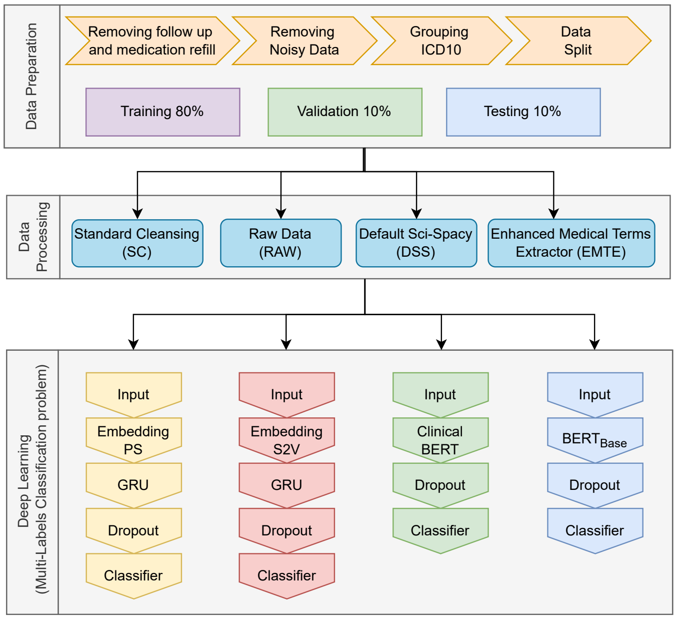


Figure 4: The experiments structure were the dataset was processed by three different cleansing methods *"SC"*, *"DSS"*, *"EMTE"* and four different training tasks using different word embeddings and classifiers were applied.

Furthermore, to qualitatively evaluate the performance of *"EMTE"* and compare it to *"DSS"*, a dataset of $1,000$ random samples was constructed. First, the medical team manually counted the total number of abbreviations, negations, and medical terms found in the $1,000$ samples. They found $2,540$ abbreviations in 575 samples, $1,329$ negations in 359 samples, and $4,891$ medical terms out of the $1,000$ randomly selected samples. This *"Gold Truth"* is compared to the results of *"EMTE"* and *"DSS"* in Table 4. Our approach improved the abbreviations detection by 68.99%, the negations detection by 43.61%, and medical terms detection by 36% when compared to *DSS*. Moreover, our approach detected abbreviations, negations, and medical terms in more samples than *DSS* by 37.71%, 15.06%, and 10.62%, respectively. The major limitations in *DSS* is that it does not detect abbreviations with punctuations such as *"u/a"*,*"-ve"*, *"+ve"*. Moreover, *DSS* failed to detect negations with *"nil"* value and the negations that were located on the right side like *"pmh: no"*. In addition, *DSS* failed to detect medical terms like *"vaginal discharge"* and *"right sided breast pain"* as one phrase.

It is worth mentioning that even our approach failed to detect some negations that contained typos. For example, it did not detect the term *"noone"* as negation since it contained typos. In addition, our approach did not catch ambiguous abbreviations like *"CLD"* (Chronic Lung Disease or Chronic Liver Disease) since these abbreviations are ambiguous and used in same specialty (Category iii 2.2).

5.3.2 ICD-10 multi-label classification results

In this section, the impact of EMTE on the ICD-10 multi-label classification results are presented and compared to the use of other cleansing techniques (*SC* and *DSS*) and the *RAW* dataset. Table 5 presents the results of the experiments with four different word embedding and DNN techniques. The columns present the results of the evaluation metrics

DNN	Data Set	Training (%)					Evaluation (%)				
		Accuracy	Recall	F1-Score			Accuracy	Recall	F1-Score		
				Macro	Micro	Weighted			Macro	Micro	Weighted
PS	SC	70.07	54.93	67.03	68.31	67.20	59.55	46.85	57.11	58.58	57.27
	DSS	70.35	55.00	67.06	68.38	67.27	61.41	48.38	58.41	60.17	58.83
	RAW	71.89	55.13	67.88	69.23	68.51	62.37	48.98	59.42	60.78	59.07
	EMTE	**74.14**	**57.93**	**69.86**	**71.25**	**70.15**	**66.19**	**51.03**	**61.31**	**62.90**	**61.64**
S2V	SC	71.13	54.81	70.76	70.66	69.99	59.78	49.98	60.13	61.20	59.71
	DSS	72.12	55.47	70.29	70.77	69.93	61.35	50.19	62.10	62.85	61.75
	RAW	72.51	55.84	70.76	71.05	70.20	61.48	50.80	62.81	63.27	62.13
	EMTE	**75.60**	**59.08**	**72.90**	**73.97**	**72.03**	**65.35**	**52.36**	**64.93**	**65.54**	**64.31**
BERT Base	SC	78.10	64.42	76.70	77.17	76.49	64.63	53.60	63.83	64.60	63.77
	DSS	78.27	64.63	76.94	77.36	76.65	64.88	53.55	63.37	64.47	63.49
	RAW	78.64	65.02	77.28	78.01	76.99	65.40	54.09	64.13	65.08	64.04
	EMTE	**79.76**	**65.72**	**77.93**	**78.46**	**77.74**	**67.46**	**55.93**	**66.05**	**67.98**	**66.19**
Clinical BERT	SC	78.18	60.19	73.99	76.29	74.17	66.83	55.63	63.20	64.69	64.79
	DSS	78.42	60.57	74.51	77.56	75.51	66.69	56.12	64.62	64.76	65.84
	RAW	79.83	61.89	74.10	78.00	75.68	66.99	56.59	65.18	65.36	66.18
	EMTE	**81.52**	**63.47**	**77.49**	**79.51**	**78.44**	**69.33**	**58.08**	**67.62**	**69.68**	**68.59**

Table 5. The results of ICD-10 multi-label classification experiments using different cleansing techniques and training models.

(Accuracy, Recall, Macro-F1 (Macro), Micro-F1 (Micro), and Weighted-F1 (Weighted).

The experiments that applied *EMTE* outperformed all the others for all the evaluated word embeddings and for all the considered metrics. The percentage of gain from applying EMTE instead of any other method and according to any metric, was computed as follows:

$$[\%gain] = 100 \times \frac{[EMTE\%] - [otherMethod\%]}{[otherMethod\%]} \tag{1}$$

For instance, the *Micro-F1*, obtained with the testing data and the *BERT_base* model, was improved by 5.44% when using *EMTE* instead of *DSS*, 5.23% instead of *SC*, and 4.46% instead of *RAW*. Moreover, the same metric *Micro-F1*, obtained with the testing data and the *Clinical_BERT* model, showed a 7.61% gain over *DSS*, 7.71% over *SC*, and 6.61% over *RAW*. The gain in performance, when cleansing the dataset with EMTE, was reflected on all the considered metrics in both training and evaluation datasets and with the four considered word embeddings.

Moreover, the gain in performance was not limited to the evaluation metrics, the use of EMTE reduced the training time of the four models and the required the number of epochs to converge. Table 6 shows for each cleansed dataset, the average execution time for an epoch and the number of epochs required for each model to coverage.

The experiments that used *EMTE*, to cleanse the dataset, converged faster than the others. For instance, the *BERT_Base* model required 19, 32, 24, and 26 epochs with the datasets cleansed by *EMTE*, *RAW*, *DSS*, and *SC* respectively. Furthermore, the experiments that used *EMTE* required 43.69% to 53.39% less execution time per epoch than *RAW*, 8.63% to 12.09% less than *DSS* and 16.48% to 26.48% less than *SC* for the different considered models.

Finally, the vocabulary size generated by *EMTE* is smaller than the ones generated by the other methods. For example, *RAW* vocabulary size was 286,891 words while *SC* contained 229,102 words (-20.14%), *DSS* contained 183,403 words (-36.07%), and *EMTE* contained 178,917 words (-37.64%).

6 Discussion

EMTE outperformed the other approaches for the following reasons: First, the experiments on *RAW* data had the biggest vocabulary since the same medical term might be represented by many data points. For example, *"Blood Pressure"* had two different data points, *"BP"* and *"Blood Pressure"*. This large vocabulary required additional resources for training and a longer execution time. Second, *SC* is also inefficient since it leads to data loss and degrades the data quality. For example, using *SC*, important information such as the negations and important punctuations like *"-"* in the abbreviation *"-ve"* were lost. Third, *DSS's* performance strictly depends on the used feature extraction tool's performance. The *"SciSpacy"* NER pre-trained model has some limitations. For example, abbreviations could be wrongly identified by *DSS* when studying datasets including different specialties. Moreover, as shown in Figure 3, *DSS* wrongly combined the abbreviations *"SOB"* and *"CRC"* with the medical terms *"no nausea"* and *"cancer"* respectively. Furthermore, *DSS* failed to detect *"no rectal bleeding"* as a negation and *"good_appetite"* as a medical term.

DNN	Dataset	Time (sec)	# Epochs
PS	SC	1167	23
	DSS	976	22
	RAW	1841	28
	EMTE	**858**	**20**
Clinical BERT	SC	1369	31
	DSS	1164	26
	RAW	2005	32
	EMTE	**1034**	**26**
BERT Base	SC	1347	26
	DSS	1256	24
	RAW	1998	32
	EMTE	**1125**	**19**
S2V	SC	1297	51
	DSS	1112	44
	RAW	1912	58
	EMTE	**1016**	**33**

Table 6. The execution time per epoch and the number of epochs per cleansing method and word embedding technique.

In this paper, we studied the medical service specialty feature along with the textual data of a chief complaint. More information can be added in the future, like the body site which refers to the location of the disease in the body like upper abdomen and lower abdomen, gender and age, to reduce the abbreviation ambiguity especially for those used in the same specialty.

One of the main advantages of the *EMTE* approach is its flexibility and maintainability. The dictionaries can be updated at any time without any need to retrain the models on new medical terms. In addition, *EMTE* can be used as a document quality enhancer as it can unify the negations writing styles and replace the abbreviations with their full-terms.

7 Conclusion and Future work

This paper presented a cleansing approach that improves the quality of medical terms extraction from unstructured clinical data using pattern matching rules based on dictionaries. The solution was conceived with flexibility and maintainability in mind for industrial use. The experiments showed that our approach helps solving the the ICD-10 prediction problem by improving the quality of the data fed to the DNNs. As a result, the performance of the trained models was improved according to various metrics. The proposed approach also reduced the required resources to train the models and decreased the training time by accelerating the convergence of the models.

In future works and in order to improve furthermore the quality of the medical data, we aim to extend this work to improve data quality by tackling several challenges like: medical term synonyms, improve abbreviation detection by adding more features (e.g. body site, gender, and age), and medical investigation results (laboratory and radiology) in chief complaints.

Acknowledgment

All computations have been performed on the Mésocentre de Calcul de Franche-Comté, France.

References

[1] M. Dugas, P. Neuhaus, A. Meidt, J. Doods, M. Storck, P. Bruland, and J. Varghese, “Portal of medical data models: information infrastructure for medical research and healthcare,” p. bav121, 2016.

[2] A. Chraibi, D. Delerue, J. Taillard, I. Chaib Draa, R. Beuscart, and A. Hansske, “A deep learning framework for automated icd-10 coding,” May 2021.

[3] S. S. Azam, M. Raju, V. Pagidimarri, and V. C. Kasivajjala, “Cascadenet: An lstm based deep learning model for automated icd-10 coding,” p. 55–74, Feb 2019.

[4] M.-C. de Marneffe, C. D. Manning, J. Nivre, and D. Zeman, “Universal Dependencies,” *Computational Linguistics*, vol. 47, no. 2, pp. 255–308, 07 2021.

[5] “Universal pos tags,” Jan 2022, accessed: 2022-01-20. [Online]. Available: https://universaldependencies.org/u/pos/

[6] D. Q. Nguyen and K. Verspoor, “From pos tagging to dependency parsing for biomedical event extraction,” Feb 2019.

[7] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60.

[8] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, “spaCy: Industrial-strength Natural Language Processing in Python,” 2020.

[9] R. Rehurek and P. Sojka, “Gensim–python framework for vector space modelling,” *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[10] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* "O’Reilly Media, Inc.", 2009.

[11] M. Skreta, A. Arbabi, J. Wang, and M. Brudno, “Training without training data: Improving the generalizability of automated medical abbreviation disambiguation,” 2019. [Online]. Available: https://arxiv.org/abs/1912.06174

[12] I. Li, M. Yasunaga, M. Y. Nuzumlalı, C. Caraballo, S. Mahajan, H. Krumholz, and D. Radev, “A neural topic-attention model for medical term abbreviation disambiguation,” *arXiv preprint arXiv:1910.14076*, 2019.

[13] Z. Wen, X. H. Lu, and S. Reddy, “Medal: Medical abbreviation disambiguation dataset for natural language understanding pre-training,” 2020.

[14] A. Koptient and N. Grabar, “Disambiguation of medical abbreviations in french with supervised methods,” May 2021.

[15] M. Skreta, A. Arbabi, J. Wang, and M. Brudno, “Training without training data: Improving the generalizability of automated medical abbreviation disambiguation,” in *Proceedings of the Machine Learning for Health NeurIPS Workshop*, ser. Proceedings of Machine Learning Research, A. V. Dalca, M. B. McDermott, E. Alsentzer, S. G. Finlayson, M. Oberst, F. Falck, and B. Beaulieu-Jones, Eds., vol. 116. PMLR, 13 Dec 2020, pp. 233–245.

[16] M. Ciosici, T. Sommer, and I. Assent, “Unsupervised abbreviation disambiguation contextual disambiguation using word embeddings,” 2019.

[17] L. Grossman Liu, R. H. Grossman, E. G. Mitchell, C. Weng, K. Natarajan, G. Hripcsak, and D. K. Vawdrey, “A deep database of medical abbreviations and acronyms for natural language processing,” Jun 2021.

[18] Y. Huang and H. J. Lowe, “A novel hybrid approach to automated negation detection in clinical radiology reports,” p. 304–311, May 2007.

[19] S. Mehrabi, A. Krishnan, S. Sohn, A. M. Roch, H. Schmidt, J. Kesterson, C. Beesley, P. Dexter, C. Max Schmidt, H. Liu, and M. Palakal, “Deepen: A negation detection system for clinical text incorporating dependency relation into negex,” p. 213–219, Apr 2015.

[20] S. Wu, T. Miller, J. Masanz, M. Coarr, S. Halgrim, D. Carrell, and C. Clark, “Negation’s not solved: Generalizability versus optimizability in clinical natural language processing,” p. e112774, Nov 2014.

[21] A. Atutxa, A. D. de Ilarraza, K. Gojenola, M. Oronoz, and O. Perez-de Viñaspre, “Interpretable deep learning to map diagnostic texts to icd-10 codes,” p. 49–59, Sep 2019.

[22] Y. Shaalan, A. Dokumentov, P. Khumrin, K. Khwanngern, A. Wisetborisu, T. Hatsadeang, N. Karaket, W. Achariyaviriya, S. Auephanwiriyakul, N. Theera-Umpon, and T. Siganakis, “Ensemble model for pre-discharge icd10 coding prediction,” *arXiv*, 2020.

[23] I. Makohon and Y. Li, “Multi-label classification of icd-10 coding & clinical notes using mimic & codiesp,” Jul 2021.

[24] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” p. 137–142, 1998.

[25] Q. Chen, J. Du, S. Kim, W. Wilbur, and Z. lu, “Deep learning with sentence embeddings pre-trained on biomedical corpora improves the performance of finding similar sentences in electronic medical records,” *BMC Medical Informatics and Decision Making*, vol. 20, 04 2020.

[26] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, and Z. Lu, “Ml-net: multi-label classification of biomedical texts with deep neural networks,” *Journal of the American Medical Informatics Association*, vol. 26, no. 11, p. 1279–1285, 06 2019.

[27] F. R. Lucini, F. S. Fogliatto, G. J. da Silveira, J. L. Neyeloff, M. J. Anzanello, R. S. Kuchenbecker, and B. D. Schaan, “Text mining approach to predict hospital admissions using early medical records from the emergency department,” *International Journal of Medical Informatics*, vol. 100, pp. 1 – 8, 2017.

[28] T. Bai and S. Vucetic, “Improving medical code prediction from clinical text via incorporating online knowledge sources,” in *The World Wide Web Conference*, ser. WWW ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 72–82.

[29] K. Adnan and R. Akbar, “Limitations of information extraction methods and techniques for heterogeneous unstructured big data,” p. 184797901989077, Jan 2019.

[30] P. Li, H. Wang, H. Li, and X. Wu, “Employing semantic context for sparse information extraction assessment,” p. 1–36, Jul 2018.

[31] Y. Zheng, Y.-W. Si, and R. Wong, “Feature extraction for chart pattern classification in financial time series,” p. 1807–1848, May 2021.

[32] P. Bose, S. Srinivasan, I. Sleeman, William C., J. Palta, R. Kapoor, and P. Ghosh, “A survey on recent named entity recognition and relationship extraction techniques on clinical texts,” p. 8319, Sep 2021.

[33] M. A. Rahman and M. H. Seddiqui, “Comparison of classical machine learning approaches on bangla textual emotion analysis,” 2019.

[34] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” 2013.

[35] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation.” in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[37] M. Pagliardini, P. Gupta, and M. Jaggi, “Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features,” in *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[39] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” 2018.

[40] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, “Publicly available clinical BERT embeddings,” in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 72–78.

[41] M. Neumann, D. King, I. Beltagy, and W. Ammar, “Scispacy: Fast and robust models for biomedical natural language processing,” *arXiv*, 2019.

[42] F. Chollet *et al.*, “Keras,” 2015. [Online]. Available: https://github.com/fchollet/keras

[43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org.