Word Sense Disambiguation for Clinical Abbreviations

by

Areej Mustafa Mahmoud Jaber

A dissertation submitted by in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

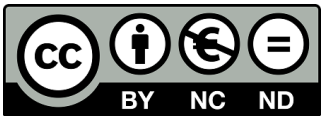
Computer Science and Technology

Universidad Carlos III de Madrid

Advisor:

Paloma Martínez Fernández

February 2022



"On this earth, what is worth living."

M. Darwish, 1941-2008

ACKNOWLEDGMENTS

The PhD experience has taught me many things -as much about how to be a researcher as about more personal matters. It has been an incomparable and life- changing process that has put me in situations where I really have had to work on myself and show resilience. This becomes a reality with the the kind support and help of many individuals. I would like to thank all of them.

First, I thank Palestine Technical University Khadoori (PTUK) to whom I am indebted for this opportunity. I am grateful to my esteemed supervisor Paloma Martínez for her useful guidance, insightful comments and constant feedback that has pushed me to improve my thinking, raising my work to a higher level. My gratitude extends to HULAT group members for their technical assistance.

Special thanks to my supportive uncle Husni who has always been by my side when I have needed him most and helped me in pursuing my dreams, after all, this journey would not be possible without him. Thanks also to his family in Spain for their hosting, love and support during these four years. I feel very fortunate to be part of this fantastic family.

My heartfelt gratitude to everyone I have met here, especially to my friend Isabel and her family who have constantly heartened me so I could achieve my goals. Thank you for reminding me that I am loved and supported.

Last but not least, many thanks to my father Mustafa, my mother Huda, who instilled the importance of learning in me and have always included me in their thoughts and prayers. Thank to my brothers, sisters and friends for their constant trust and belief in me. I love you all.

PUBLISHED CONTENT

The following publications realized by the author have been partially included as part of this thesis:

- Journals:
  - Areej Jaber and Paloma Martínez. "Disambiguating Clinical Abbreviations Using a One-Fits-All Classifier Based on Deep Learning Techniques". Methods of Information in Medicine, Feb. 2022. https://doi.org/10.1055/s-0042-1742388 [JCR,Q3].
    - This publication is partially included in Chapters 1, 2, 3 and 4 of this thesis.
- Conferences and Workshops:
  - Areej Jaber, Paloma Martínez. "Abbreviation Extraction in Spanish Clinical Text", proceeding in womENcourage, Rome, Italy, September, 2019, p. 1. https://womencourage.acm.org/2019/wp-content/uploads/2019/07/womENcourage_2019_pape_7.pdf
  - Areej Jaber, Paloma Martínez. "Abbreviation extraction and normalization in Spanish clinical text". In: Doctoral Symposium of the XXXV International Conference of the Spanish Society for Natural Language Processing. Vol 2633. Bilbao, Spain: CEUR Workshop Proceedings; 2019:13-19. http://ceur-ws.org/Vol-2633/paper3.pdf.
  - Areej Jaber, Paloma Martínez. "Disambiguating Clinical Abbreviations using Pre-trained Word Embeddings". Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2021, Volume 5: HEALTHINF, Online Streaming, February 11-13, 2021:501-508. doi:10.5220/0010256105010508
    - This publication is partially included in Chapters 1, 2, 3 and 4 of this thesis.
  - Areej Jaber, Paloma Martínez. "Participation of UC3M in SDU@AAAI-21: A Hybrid Approach to Disambiguate Scientific Acronyms". Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Intelligence, SDU@AAAI 2021,Virtual Event,Feb. 9, 2021. Vol 2831. CEUR Workshop Proceedings;2021. http://ceur-ws.org/Vol-2831/paper23.pdf.
    - This publication is partially included in Chapters 3 and 4 of this thesis.

ABSTRACT

Abbreviations are extensively used in electronic health records (EHR) of patients as well as medical documentation, reaching 30-50% of the words in clinical narrative. There are more than 197,000 unique medical abbreviations found in the clinical text and their meanings vary depending on the context in which they are used. Since data in electronic health records could be shareable across health information systems (hospitals, primary care centers, etc.) as well as others such as insurance companies information systems, it is essential determining the correct meaning of the abbreviations to avoid misunderstandings. Clinical abbreviations have specific characteristic that do not follow any standard rules for creating them. This makes it complicated to find said abbreviations and corresponding meanings. Furthermore, there is an added difficulty to working with clinical data due to privacy reasons, since it is essential to have them in order to develop and test algorithms.

Word sense disambiguation (WSD) is an essential task in natural language processing (NLP) applications such as information extraction, chatbots and summarization systems among others. WSD aims to identify the correct meaning of the ambiguous word which has more than one meaning. Disambiguating clinical abbreviations is a type of lexical sample WSD task. Previous research works adopted supervised, unsupervised and Knowledge-based (KB) approaches to disambiguate clinical abbreviations. This thesis aims to propose a classification model that apart from disambiguating well known abbreviations also disambiguates rare and unseen abbreviations using the most recent deep neural network architectures for language modeling.

In clinical abbreviation disambiguation several resources and disambiguation models were encountered. Different classification approaches used to disambiguate the clinical abbreviations were investigated in this thesis. Considering that computers do not directly understand texts, different data representations were implemented to capture the meaning of the words. Since it is also necessary to measure the performance of algorithms, the evaluation measurements used are discussed.

As the different solutions proposed to clinical WSD we have explored static word embeddings data representation on 13 English clinical abbreviations of the UMN data set (from University of Minnesota) by testing traditional supervised machine learning algorithms separately for each abbreviation. Moreover, we have utilized a transformer-base pretrained model that was fine-tuned as a multi-classification classifier for the whole data set (75 abbreviations of the UMN data set). The aim of implementing just one multi-class

classifier is to predict rare and unseen abbreviations that are most common in clinical narrative. Additionally, other experiments were conducted for a different type of abbreviations (scientific abbreviations and acronyms) by defining a hybrid approach composed of supervised and knowledge-based approaches.

Most previous works tend to build a separated classifier for each clinical abbreviation, tending to leverage different data resources to overcome the data acquisition bottleneck. However, those models were restricted to disambiguate terms that have been seen in trained data. Meanwhile, based on our results, transfer learning by fine-tuning a transformer-based model could predict rare and unseen abbreviations. A remaining challenge for future work is to improve the model to automate the disambiguation of clinical abbreviations on run-time systems by implementing self-supervised learning models.

RESUMEN

Las abreviaturas se utilizan ampliamente en las historias clínicas electrónicas de los pacientes y en mucha documentación médica, llegando a ser un 30-50% de las palabras empleadas en narrativa clínica. Existen más de 197.000 abreviaturas únicas usadas en textos clínicos siendo términos altamente ambiguos El significado de las abreviaturas varía en función del contexto en el que se utilicen. Dado que los datos de las historias clínicas electrónicas pueden compartirse entre servicios, hospitales, centros de atención primaria así como otras organizaciones como por ejemplo, las compañías de seguros es fundamental determinar el significado correcto de las abreviaturas para evitar además eventos adversos relacionados con la seguridad del paciente. Nuevas abreviaturas clínicas aparecen constantemente y tienen la característica específica de que no siguen ningún estándar para su creación. Esto hace que sea muy difícil disponer de un recurso con todas las abreviaturas y todos sus significados. A todo esto hay que añadir la dificultad para trabajar con datos clínicos por cuestiones de privacidad cuando es esencial disponer de ellos para poder desarrollar algoritmos para su tratamiento.

La desambiguación del sentido de las palabras (WSD, en inglés) es una tarea esencial en tareas de procesamiento del lenguaje natural (PLN) como extracción de información, chatbots o generadores de resúmenes, entre otros. WSD tiene como objetivo identificar el significado correcto de una palabra ambigua (que tiene más de un significado). Esta tarea se ha abordado previamente utilizando tanto enfoques supervisados, no supervisados así como basados en conocimiento. Esta tesis tiene como objetivo definir un modelo de clasificación que además de desambiguar abreviaturas conocidas desambigüe también abreviaturas menos frecuentes que no han aparecido previamente en los conjuntos de entrenaminto utilizando las arquitecturas de redes neuronales profundas más recientes relacionadas ocn los modelos del lenguaje.

En la desambiguación de abreviaturas clínicas se emplean diversos recursos y modelos de desambiguación. Se han investigado los diferentes enfoques de clasificación utilizados para desambiguar las abreviaturas clínicas. Dado que un ordenador no comprende directamente los textos, se han implementado diferentes representaciones de textos para capturar el significado de las palabras. Puesto que también es necesario medir el desempeño de cualquier algoritmo, se describen también las medidas de evaluación utilizadas.

La mayoría de los trabajos previos se han basado en la construcción de un clasificador separado para cada abreviatura clínica. De este modo, tienden a aprovechar diferentes recursos de datos para superar el cuello de botella de la adquisición de datos. Sin embargo,

estos modelos se limitaban a desambiguar con los datos para los que el sistema había sido entrenado.

Se han explorado además representaciones basadas vectores de palabras (word embeddings) estáticos para 13 abreviaturas clínicas en el corpus UMN en inglés (de la University of Minnesota) utilizando algoritmos de clasificación tradicionales de aprendizaje automático supervisados (un clasificador por cada abreviatura). Se ha llevado a cabo un segundo experimento utilizando un modelo multi-clasificador sobre todo el conjunto de las 75 abreviaturas del corpus UMN basado en un modelo Transformer pre-entrenado. El objetivo ha sido implementar un clasificador multiclase para predecir también abreviaturas raras y no vistas. Se realizó un experimento adicional para siglas científicas en documentos de dominio abierto mediante la aplicación de un enfoque híbrido compuesto por enfoques supervisados y basados en el conocimiento.

Así, basándonos en los resultados de esta tesis, el aprendizaje por transferencia (transfer learning) mediante el ajuste (fine-tuning) de un modelo de lenguaje preentrenado podría predecir abreviaturas raras y no vistas sin necesidad de entrenarlas previamente. Un reto pendiente para el trabajo futuro es mejorar el modelo para automatizar la desambiguación de las abreviaturas clínicas en tiempo de ejecución mediante la implementación de modelos de aprendizaje autosupervisados.

CONTENTS

ACRONYMS

AI Artificial Intelligent. 1

AUI Atomic Unique Identifier. 12

BARR2 The Second Biomedical Abbreviation Recognition and Resolution. xv, 15–18, 37, 41, 82

BERT Bidirectional Encoder Representations from Transformers. 7, 9, 21, 22, 32, 56, 58–60, 63, 70, 72, 74, 78, 79, 81–83

BOW Bags Of Word. 18, 27, 76

BSC Binary Spatter Code. 28, 37

CARD Clinical Abbreviation and Disambiguation. 13

CASI Clinical Abbreviations Sense Inventory. 13, 28

CBOW Continuous Bag of Words. 20, 21

CDSS Clinical Decision Support System. 1

CNN Convolutional Neural Network. 30, 35, 36, 71

CUI Concept Unique Identifier. 12, 14, 15, 27

DT Decision Trees. 26, 27, 77

EHR Electronic Health Record. 1, 75, 82

ELMo Embedding from Language Model. 21, 34, 72

FFNN Feed-Forward Neural Network. 32, 77

FN False Negative. 23

FP False Positive. 23

GloVe Global Vectors for Word Representations. 20, 21

KB Knowledge-based. v, 3, 9, 10, 25, 36, 37, 42, 51, 62, 67, 77

kNN k-Nearest Neighbor. 26, 52, 67, 73, 77

LCM Latent Meaning Cells. 36

LF Long Form. 3, 37

LR Logistic Regression. 27, 28, 71, 72

LSA Latent Semantic Analysis. 37

LSTM Long Short-Term Memory. 22, 30–32, 77

LVG Lexical Variation Generator. 12

ME Maximum Entropy. 26, 28

MEDLINE Medical Literature Analysis and retrieval System Online. xviii, 15, 28, 40, 45, 82

MeSH Medical Subject Headings. 15

MIMIC III Medical Information Mart for Intensive Care. 6, 14, 27, 34, 36, 40, 58, 72

MLM Masked Language Modeling. 22, 59, 83

NB Naïve Bayes. 26–28, 37, 46–48, 52, 54, 62, 64, 65, 67–69, 73, 77, 78

NLM National Library of Medicine. 12, 15

NLP Natural Language Processing. 1, 2, 7, 12, 14, 16–19, 21, 22, 30–33, 44, 58, 75, 77, 82

NNLM Neural Networks Language Model. 7

NSP Next Sentence Prediction. 22, 59, 78, 83

OHE One Hot Encoding. 18

OOV Out Of Vocabulary. 21

PMC PubMed Central. 27, 58, 65, 66, 73, 78

POS Part-Of-Speech. 17, 35

RNN Recurrent Neural Network. 21, 22, 30, 31, 77

SCiAD SCientific Acronyms Disambiguation. 9, 49, 53, 62, 64, 73, 78, 79

SciELO Scientific Electronic Library Online. 16

SEPLN Spanish Society for Natural Language Processing. 15, 16

SF Short Form. 3, 37

SG Skip Gram. 20, 21, 45

SSL Self Supervised Learning. 83, 84

SVM Support Vector Machine. 13, 26–28, 37, 46–48, 52, 54, 62, 64–67, 73, 77, 78

TF-IDF Term Frequency - Inverse Document Frequency. 19, 38, 76

TN True Negative. 23

TP True Positive. 23

UMLS Unified Medical Language Modeling. 4, 12–15, 36, 37, 40

UMN University of Minnesota-affiliated. 5, 6, 9, 14, 26–29, 34–36, 38, 40, 42, 43, 56, 58, 59, 61–64, 70–74, 77–79, 81

VSM Vector Space Model. 26, 28, 37, 38

VUH Vanderbilt University Hospital's. 15, 28

VUMC Vanderbilt University Medical Center. 13

WSD Word Sense Disambiguation. xv, 2, 3, 5, 7–11, 18, 21, 25, 31, 36, 37, 39, 42, 44, 46, 47, 51, 61, 75–79, 84

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Natural Language Processing (NLP) (Hirschberg and Manning, 2015) is a branch or a sub field of Artificial Intelligent (AI) concerned with understanding, manipulating and interpreting human languages by computers. Humans typically communicate each other through language, either through speech or writing. With technological advancements and the spreading use of intelligent machines in all aspects of our daily lives, such as mobile phones and computers, the lack of communication between machines and humans is actually felt. The goal of natural language processing for machines is to understand and make sense of human languages. This means that the machine, like a newborn, learns the language and then uses it to communicate.

The digitized data of the health care systems is growing at an exponential rate. The health care systems aim to save and manage the health information of the patients through the Electronic Health Record (EHR), which is defined as a digital version of a patient's medical history as kept by the health care providers. EHR data contains heterogeneous elements; it consists of radiology and laboratory test results, diagnosis, demographic data and notes. The large amount of patient data that EHR includes enables healthcare and research communities to leverage existing observational research and analysis to improve people's well-being.

Furthermore, EHR is considered the main data source for Clinical Decision Support System (CDSS), which is known as "any software designed to directly aid in clinical decision making in which characteristics of individual patients are matched to a computerized knowledge base for the purpose of generating patient-specific assessments or recommendations that are then presented to clinicians for consideration" (Hunt et al., 1998). Figure 1.1 illustrates the different tasks that could be performed by CDSS, in addition to the parties which such systems could assist.

EHR data could be classified into two categories; structured data which is represented in a formatted way that contains values from predefined dictionaries or specific numeric values. For example, vital signs, laboratory test results and administrative data. Unstructured data, which represents 80% of EHR data, is presented in a free text like clinical notes and discharge summaries, also it could include handwritten notes. Manipulating and getting information from unstructured data is challenging (P.-Y. Wu et al., 2017) be-

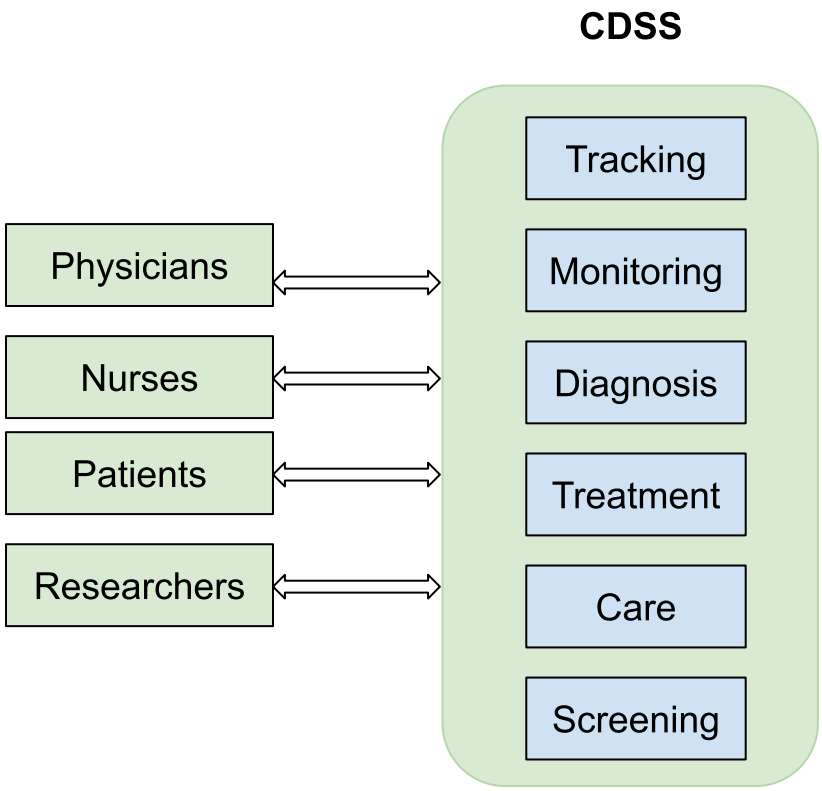


Figure 1.1: Clinical Decision Support Systems tasks based on NLP for assistance physicians, nurses, patients and researchers.

cause of many reasons: a lot of typos and misspellings, no standard formats to create new abbreviations, grammatical errors, in addition to using abbreviations extensively.

With the growing of unstructured data, the need of NLP is mandatory to manipulate and to analyze it. Several NLP tasks have been addressed in clinical domain (S. Wu et al., 2020), such as Text Classification (Colón-Ruiz et al., 2019), Named Entity Recognition (Castro et al., 2010, Akhtyamova et al., 2020), Relation Extraction (Suárez-Paniagua et al., 2019, Segura-Bedmar and Martínez, 2015), Word Sense Disambiguation (WSD) (Zotova et al., 2021)and Text Simplification (Alarcón et al., 2021).

In NLP, the WSD task determines the correct meaning or "sense" of a word among a predefined set of senses. This determination strongly depends on the context where the word appears. For example, consider the following sentences:

"The bank will not be accepting cash on Saturdays."

"The river overflowed the bank."

The word "bank" in the first sentence refers to the commercial (finance) sense of the word "bank", while in the second sentence, it refers to the river bank. The surrounding words play an essential role in determining the meaning of the word. The generic WSD task can be divided into two variants (Navigli, 2009):

- Lexical sample WSD: The goal is to disambiguate a specific set of target words, typically one per sentence. Supervised systems are commonly used in this context because they can be trained using a large number of hand-labeled examples (training set) and then the learned models are used to classify a large number of unlabeled examples (test set).

- All-words WSD: It is expected that systems will disambiguate all open-class words in a text (i.e., nouns, verbs, adjectives and adverbs). This task requires the use of systems with broad coverage. As a result, purely supervised systems may suffer from data sparsity, as it is unlikely that a training set of sufficient size covering the entire lexicon of the language of interest is available. Other approaches, such as Knowledge-based (KB) systems, on the other hand, rely on full-coverage knowledge resources and the availability of which must be ensured.

WSD is considered a form of the multi-classification task, where the set of predetermined senses are the classes, the context surrounding the target word is the evidence and each occurrence of the word is mapped to one class based on this evidence. There are two main approaches to performing disambiguation tasks. These approaches are classified based on the knowledge source that are used to differentiate the senses. The KB approach relies on external knowledge resources like dictionaries, thesauri and lexical knowledge bases. On the other side, machine learning approaches (including neural networks) which rely on corpus evidence to build the classifier. Therefore, this approach has three categories: supervised, semi-supervised and unsupervised approaches.

1.1. Motivation

Abbreviations and Acronyms are shortened formats of phrases. An abbreviation is used to represent the whole word as **"Dr."** or a set of letters of a phrasal word like **"AMA"** to refer to **"Advanced Material Age"** while Acronym is a new word which is formed from a set of phrase initial letters like **"COPE"** which is formed from the initial letters of the phrase **"Chronic Obstructive Pulmonary Emphysema"**. Abbreviations and acronyms are often interchanged. Thus, in this document, we will use "abbreviation" to mean both "abbreviation" and "acronym".

An abbreviation is also known as Short Form (SF) and its meaning could be donated as Long Form (LF), expansion, or sense. There are two types of abbreviations used in the medical domain, a local abbreviation written with their long-form in the document and used mainly in the biomedical domains. For instance, in this sentence which is extracted from the biomedical article (Franceschet et al., 2016) **"Portal biliopathy (PB) is defined as the presence of biliary abnormalities ..."** , the abbreviation **"PB"** and its meaning

"Portal biliopathy" appears in the same sentence. However, a global abbreviation used in the clinical text is written without any reference to its meaning. For example, in this sentence which is extracted from clinical text **"...have volar dislocation of the MP joints. There is no swelling..."**, the abbreviation **"MP"** is mentioned without its meaning.

Several studies have shown that abbreviations are widely used in the medical domain in both biomedical and clinical texts. Abbreviations account for 30–50% of words in the clinical text, such as doctors' notes (Grossman et al., 2018), compared to 1% in general text, such as news media (Ehrmann et al., 2013). (Billy, 2017) indicated in their work a high frequency of abbreviations used in medical records at a district hospital in a resource-limited setting. (Holper et al., 2020) reported that 8.9% were abbreviations in a study conducted on 2,336 discharge summaries. (Schwarz et al., 2021) found 750 abbreviations on 100 discharge summaries.

There are many benefits using abbreviations in the medical domain. Typically, many medical terms are long and hard to spell, so the clinicians avoid any mistakes or misspellings by using abbreviations. Furthermore, using abbreviations save time writing clinical documents. For instance, if a medical word like **"Methylenedioxymethamphetamine"** is needed to be written more than one time in a medical report, it will be a tiresome and time-consuming process, so clinicians prefer to abbreviate it. In addition, in some cases, the abbreviations are used not only to save time and space but also to hide serious or incurable illnesses to patients in medical reports.

As shown in Table 1.1, there are no standard rules for creating the abbreviations and they could contain numbers and special characters. Their meanings depend on the scope where they are used. Further, bilingual problems, for example, **"PSA"** which means in English language **"prostatic specific antigen"** could be used in Spanish clinical text to refers **"antígeno prostático específico"**, even though, the Spanish abbreviation for this term is **"APE"**. Moreover, abbreviations are ambiguous, which means that abbreviations have more than meaning with one to many relation. For example, the abbreviation **"AMA"** has three different expansions, **"AntiMitochondrial Antibody"** , **"Against Medical Advice"** and **"Advanced Maternal Age"**.

A high percentage of abbreviations in MEDLINE abstracts can have multiple meanings (64.6%) (H. Liu et al., 2001). Structured knowledge bases, such as the Unified Medical Language Modeling (UMLS)) (Lu et al., 2020), contain a significant number of ambiguous abbreviations (33.1%) (H. Liu et al., 2001). In addition, several studies conducted on clinical text, (H. Xu et al., 2007) showed that UMLS only covered 35% of expansions of abbreviations in hospital admission notes at New York-Presbyterian Hospital. This means that there are no standard abbreviations generated by consensus among domain experts. Furthermore, 80% of the abbreviations included in UMLS have am-

biguous occurrences in Medline (H. Liu et al., 2001). (Schulz et al., 2017) found 7,439 ambiguous SNOMED-CT terms and 899 ambiguous acronyms.

One of the serious challenges in patient safety in healthcare is to tackle misunderstandings caused by abbreviations in clinical narrative. According to a 2001 Sentinel Event Alert from the United States, abbreviations could account for up to 5% of prescription-related errors (Samaranayake et al., 2014). A conducted Australian survey, 1,073 of inpatient prescribing, 8.4% of orders contain at least one error-prone abbreviation, with 29.6% deemed to be at high risk of causing significant harm (Dooley et al., 2012). Moreover, misunderstanding of abbreviations led to team miscommunication; a study which was done on pediatric sign-out sheets showed that the pediatricians understood 56-94% of the abbreviations which were used in the sheets, while physicians from other fields were able to understand only 31-63% (Sheppard et al., 2008). Misinterpretation of abbreviations can result in inappropriate, delayed, or harmful patient care.

TABLE 1.1: Some examples of abbreviation creation rules.

Rule	**Abbreviation**	**Sense**
Truncating the end of long form	DIP	**DIP**ropionate
First letter initialization from each word	VBG	**V**enous **B**lood **G**as
Syllabic initialization	US	**U**ltra**S**ound
Combination if the beginning of some of the words of long-form	Ad lib	**Ad lib**itium
symbols/synonyms substitution or initialization	T3	**Tri**iodo**th**yronine

Disambiguating clinical abbreviations is considered as a form of lexical sample WSD task type, where a set of clinical sentences with ambiguous abbreviations are manually annotated by experts to their exact meanings. Typically each clinical sentence contains one abbreviation. There are a set of restricted meanings (senses) for each abbreviation that represent the classes. Various studies implemented disambiguation models. Most of them implemented one classifier for each abbreviation. Thus, they applied several methods to increase the training data automatically to avoid manual annotation. However, these methods would not able to predict unseen abbreviations and could not increase the training data for rare abbreviations. This thesis explores the state-of-the-art of the clinical abbreviation disambiguation task. In addition, it explores the latest deep learning and language modeling approaches to disambiguate unseen and rare abbreviations.

1.2. Research Hypothesis

The problem of the clinical disambiguation task is the lack of annotated examples for each abbreviation in the available clinical data set. In this thesis, we used the University

of Minnesota-affiliated (UMN) data set for 75 abbreviations; the data set contains 500 annotated examples distributed among a set of senses for each abbreviation. Furthermore, the distribution of the annotated examples is strongly imbalanced. Implementing a separated classifier for each abbreviation required more annotated examples. In addition to that, the implemented classifiers cannot predict unseen abbreviations and senses. Based on what was mentioned above, our research question is:

Research Question
Is it possible to improve a model to disambiguate unseen and rare clinical abbreviations using the architecture of language modeling?

In order to test this research question, several hypotheses could be tested in this work, we will discuss in the following:

Hypothesis 1:
If we want to generate a model that disambiguates clinical abbreviations from scratch, then we could get enough clinical annotated data to perform the experiments.

Data is the keystone of any machine learning and deep learning approaches. The more data is available, the more accurate result will be achieved. However, raw data is not enough to generate a model. Annotated data is an essential component of what allows many machine learning projects to function correctly. It provides the fundamental framework for teaching a model what it needs to understand and how to differentiate to generate correct outputs across a large and diverse range of inputs.

One of the primary challenges in working in the clinical domain is the annotation acquisition bottleneck. In addition to how costly and tedious the annotation process is for any developments task, it is not easy to get enough clinical data to work with for privacy issues. There is only one annotated corpus, which is publicly available, composed of 75 clinical abbreviations. Also, just one unannotated clinical data set with its third version is available, known as MIMIC III.

For these reasons, we believe, in this thesis, implementing a model to disambiguate clinical abbreviations using advanced deep learning technologies from scratch is inapplicable.

Hypothesis 2:

If contextualized word embeddings are used for representing the data, then semantic similarities could be represented better than using static word embedding.

In general, word embedding is a term that is used to represent words numerically by that it attempts to capture internal semantic and syntactic information by mapping words in unlabeled text data to a continuously-valued low dimensional space. The word embedding concept was first introduced in (Hinton et al., 1986) and (Mikolov et al., 2013) using the Neural Networks Language Model (NNLM). The reasoning is that words in the same context are more likely to have similar meanings (Miller and Charles, 1991).

Two main types of word embedding have been studied extensively in NLP. A Static word embedding is single vector representation for a word ignoring the context in which it appears. However, a dynamic word embedding reflects the features vectors taking into account the current context of the word. So that, we believe, in the thesis, that contextualized word embedding can represent the similarities better than the static ones.

Hypothesis 3:

If clinical Transformers based language models, such as Bidirectional Encoder Representations from Transformers (BERT), are fine tuned, then the model performance could be improved.

Since Transformer has been proposed (Vaswani et al., 2017), it is successfully identified as a prominent architecture for natural language processing. It outperformed the previous neural networks architectures, such as the recurrent neural network. The architecture evolves with the training data and model size, allowing for efficient parallel training and capturing long-range sequence features.

Transformers and fine-tuning strategies improved the performance of NLP tasks, including Word Sense Disambiguation (WSD). The fundamental rule in this approach is to reuse a model trained for one task as the starting point for training a model on target downstream task. Many pre-trained models using Bidirectional Encoder Representations from Transformers (BERT) language modeling architectures have been released for the clinical domain. We believe, in this thesis, a clinical pre-trained BERT model could improve the performance of the disambiguation better than a biomedical pre-trained BERT model.

Hypothesis 4:

If a one-fits-all classifier is implemented for all the clinical abbreviations, then the classifier could predict unseen expansions.

Traditionally, the most common approach to disambiguating clinical abbreviations is implementing a separated classifier for each abbreviation. First, this requires a separated data set for these abbreviations to train each classifier. Second, the researchers have performed different methods to expand the individual training data set because it has strongly imbalanced distributions. These approaches do not solve the problem of the rare expansions due to the natural form, which has already been rarely used. In addition to that, the classifiers are trained in a restricted set of expansions and they fail to predict unseen ones'. We believe, in this thesis, that a one-fits-all classifier could predict both unseen and rare expansions without any need for extra annotated examples.

1.3. Objectives

This Ph.D. thesis aims to present a fine-tuned model architecture that disambiguates unseen and rare clinical abbreviations by integrating a pre-trained language modeling. It can be broken down into the following sub-objectives:

- To revise the state-of-the-art of clinical abbreviation disambiguation as a WSD classification problem. This covers the four elements of the task: resources, representation of the data, evaluation measurements and the classification approach.
- To assess the effect of static word embedding as a feature in the supervised machine learning approach.
- To investigate the effect of imbalanced data set on the accuracy of the supervised models.
- To integrate a supervised model with a knowledge-based one to disambiguate acronyms that have small numbers of annotated examples.
- To fine-tune a language model to build one-fits-all classifier that improve the accuracy of the previous existing models.
- To analyze the result of the fine-tuning model on rare and unseen data.
- To identify unresolved issues in the conclusions in order to justify future studies

1.4. Document outline

The rest of this document is organized as follows:

- Chapter 2 introduces the state-of-the-art clinical abbreviation disambiguation as a WSD classification task. Revising the resources, the data presentation techniques, evaluation measurements, the different classification approaches and the current challenges in the clinical abbreviation disambiguation domain.
- Chapter 3 describes the architecture of three proposed models that used different abbreviations data sets. This includes a multi classifier supervised approach for 13 clinical abbreviations from the UMN data set. Second, a fine-tuned one-fits-all classifier using BERT for the UMN data set. Third, a hybrid approach composed of some supervised approaches and a Knowledge-based (KB) approach to disambiguate the SCiAD data set has been proposed.
- Chapter 4 illustrates the result of applying the three proposed models. We have investigated the effect of the imbalanced distribution of the data set on the accuracy among the separated supervised classifiers. We have compared the performance using accuracy on three tested pre-trained language models, in addition to the efficiency in predicting rare and unseen data.
- Chapter 5 wraps up the clinical abbreviation disambiguation task, the conclusions of the three conducted studies and addresses the research gap with a view of the future work.

1.5. Funding

This thesis has been partially supported by:

- The Research Program of the Ministry of Science and Innovation - Government of Spain (ACCESS2MEET project-PID2020-116527RB-I00) and Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17) and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).
- Palestinian Authorities, Ministry of high education with Technical University - kadoori, Tulkarm, Palestine.

Chapter 2

RELATED WORK

In this chapter, we will review the clinical abbreviation disambiguation task as a type of the lexical sampleWSD task. This task comprises four crucial parts, which will be discussed in separated sections. In the first section, we will define the task as a WSD multi-class classification task. In the second section, we will comprehensively discuss two types of resources commonly required in disambiguation tasks, senses inventory and annotated and unannotated corpora, in addition to the biomedical resources that were leveraged in this task.

The third section summarizes the data representation techniques applied to this task, starting from traditional feature extractions to different word embedding models. The fourth section presents the different classification approaches that were followed to disambiguate the clinical abbreviations. These approaches varied from machine learning, deep neural networks and KB approaches. In the last section, we will discuss the challenges of this domain.

2.1. Clinical Abbreviation Disambiguation Definition

Clinical Abbreviation Disambiguation is defined as the capacity to computationally recognize the right expansion among a set of predefined list of expansions based on the given context. The task is considered a form of Lexical sample (or targeted WSD) where a system must disambiguate a limited number of target words that typically occur one per sentence. In this situation, supervised systems are commonly used since they can be trained using a set of hand-labeled examples (training set) and then used to categorize a set of unlabeled examples (test set). So, if we have a sentence **S** that contains a sequence of words $(w_1, w_2, w_t, ..., w_n)$, where w_t is the targeted abbreviation, WSD task is described as mapping the appropriate **expansion(e)** to a w_t based on its surrounding words (context) where **expansions(e)** is a set of expansions that was determined previously (Navigli, 2009).

Figure 2.1 depicts an example of clinical disambiguation as a classification task where ***"AB"*** is the ambiguous abbreviation. Based on the annotated corpus ***"AB"*** could have four meanings (4 classes). Considering the context of the sentence, the expansion ***"Abor-***

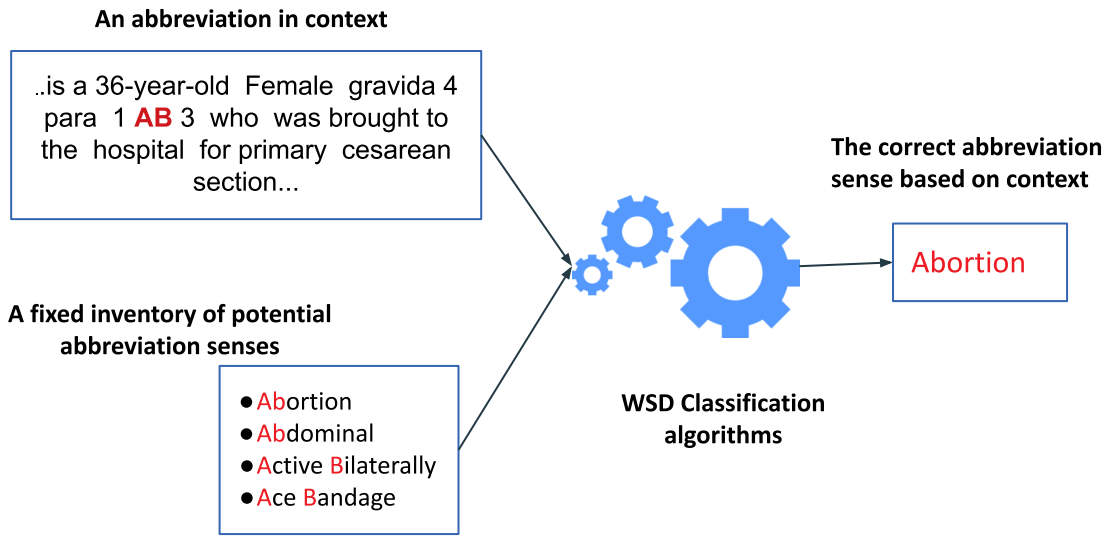


Figure 2.1: Clinical abbreviation Disambiguation as a WSD task: a clinical note with the targeted abbreviation "AB" and a list of its senses. Based on the context, the proper sense is "Abortion."

tion" is the right class in this situation.

2.2. Resources

One of the fundamental components of WSD tasks, including disambiguate clinical abbreviations, is Knowledge. Knowledge resources provide essential information to determine which expansion is meant on the targeted abbreviation. Knowledge resources in the clinical domain could be varied from senses inventories to corpora of clinical notes, either annotated or unannotated, machine-readable dictionaries and ontologies. However, these resources are considered one of the bottlenecks problem in this domain. Due to privacy issues, it is not easy to get clinical data. A description of different resources that are used to disambiguate clinical abbreviation will be described below:

2.2.1. Senses Inventories

The disambiguation task aims to determine the exact meaning among a predefined set of expansions related to the ambiguous word. Hence, an inventory that recognizes this set of meanings is required to define them. A sense inventory is essential for effective abbreviation management because it provides target expansions for disambiguation that correspond to the clinical abbreviation. In the following, we will describe a set of available senses inventories used as a reference to abbreviation expansions from the biomedical and clinical domain.

- Unified Medical Language Modeling (UMLS) (Lu et al., 2020): The United States National Library of Medicine (NLM) provides a collection of biomedical and clinical information services. UMLS is one of these services, including a set of files and software that enable interoperability between computer systems. In addition to providing these resources, it provides ontological representations of medical concepts and relations between these medical terms. The Metathesaurus, Semantic Networks and SPECIALIST Lexicon are the main parts of UMLS. The Metathesaurus is a multilingual lexical database that semi-automatically incorporates information from biomedical and clinical sources regarding biomedical and health-related terms into a unified representation. The Metathesaurus derives concepts from numerous sources and assigns a Concept Unique Identifier (CUI) to each concept. A CUI can be used to refer to numerous terms from different terminologies. Atomic Unique Identifier (AUI)s are used as a label to these concepts. The term could be assigned for more than one CUI, which refers to be ambiguous. The Semantic Networks used Metathesaurus concepts to create semantic and relations between them. A semantic type is a collection of concepts that are connected in some way.

  The SPECIALIST Lexicon is a collection of biomedical and common English terminology used in the biomedical and health-related fields. A part of this collection, LRABR file is included to represent a set of acronyms and abbreviations that could be found in the biomedical texts. 41,512 acronyms and abbreviations are found in this file, 11,164 have more than two expansions (around 30%). NLP tools like the SPECIALIST minimal commitment parser and Lexical Variation Generator (LVG) complement the SPECIALIST Lexicon.

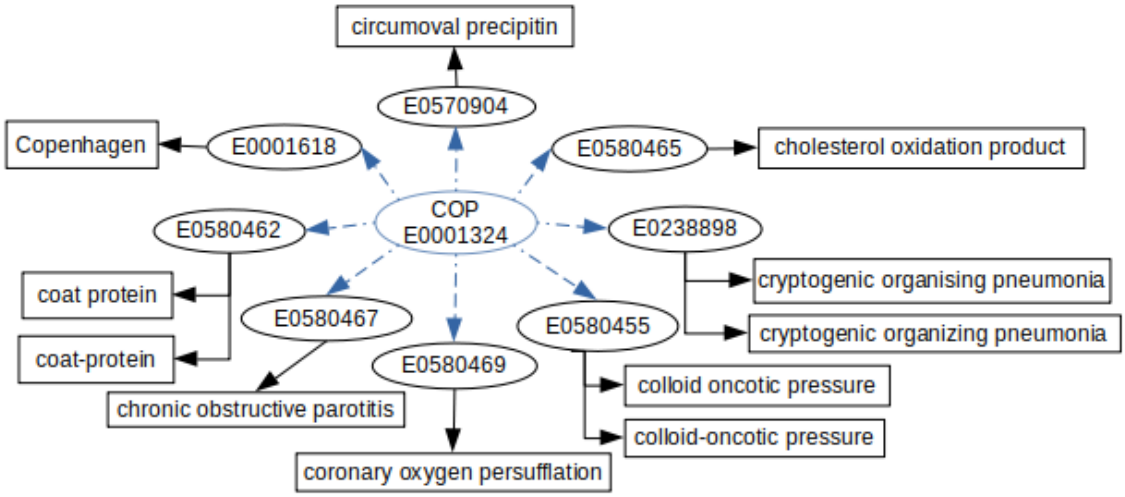


Figure 2.2: Example of "COP" abbreviation ambiguity in UMLS

- Columbia Corpus-based sense inventory (H. Xu et al., 2009): An inventory of 12 abbreviations and 40 senses semi-automatically generated using 16,949 inpatient

admission notes from the Hospitalist Service between 2004 to 2006. The generation of this inventory was done in two steps. First, a decision tree machine language method was applied to detect the abbreviations from the clinical notes. 19,965 abbreviations were extracted and 977 abbreviations had occurrences more than 100 times that were connected to UMLS. 171 abbreviations among the 977 were ambiguous; due to the high cost related to the annotated process, the author built the inventory of just 12 abbreviations. Second, Clustering algorithms were applied for each abbreviation instance (around 1,000). For each cluster centroid, an instance was chosen to be annotated manually by an expert.

- Clinical Abbreviation and Disambiguation (CARD) (Y. Wu et al., 2017): This senses inventory is built from clinical documents taken from Vanderbilt University Medical Center (VUMC)'s synthetic derivatives database. First, the Support Vector Machine (SVM) algorithm was implemented to predict 27,317 and 107,303 distinct abbreviations from discharge and clinical notes, respectively. Then, a clustering algorithm was applied in the context of these abbreviations to determine their senses. Eventually, two senses inventories were constructed for the 1,000 most frequent abbreviations in these two corpora. Discharge summaries corpora contain 915 abbreviations and 1,299 expansions, clinical notes corpora with 954 abbreviations and 1,499 expansions.

- Clinical Abbreviations Sense Inventory (CASI) (Moon et al., 2014): A sense inventory was generated from 352,267 clinical notes by University of Minnesota research groups. The clinical notes were extracted from different hospitals belonging to the University of Minnesota affiliated Fairview Health Services between 2004 to 2007. These clinical notes include admission notes, consultation notes and discharge summaries. A set of heuristic rules was used to extract the abbreviations from the clinical notes. Two clinical specialists manually annotated the correct sense for each abbreviation. 440 abbreviations were annotated with 949 expansions from 220,000 instances.

- Medical Abbreviation and Acronyms Meta-Inventory (L. G. Liu et al., 2021): A comprehensive harmonization of eight source inventories from various healthcare specializations and contexts, such as online repositories, UMLS-LRABR and peer-reviewed scientific literature, yielded 104,057 abbreviations and 170,426 senses. The authors inspired with UMLS Metathesaurus ontology for the harmonization procedures, each row in the inventory contains the abbreviation, its long-form, source of the inventory, unique identifier and another two fields contain normalized short form and long-form.

- Other databases of Abbreviations from biomedical research: In attempting to get

around the clinical data bottleneck, many researchers create inventories from biomedical resources using rule-based and statistical methods and then used them to disambiguate clinical abbreviations (like SaRAD (Adar, 2004), ARCH (Wren and Garner, 2002) and ALICE (Ao and Takagi, 2005)).

One of the most representative of abbreviations sense inventory is ADAM (Zhou et al., 2006). ADAM provides 59,403 pairs of short and long forms and displays the word frequency of various terms and other statistical data to show how each abbreviation or acronym is used in the biomedical literature. The inventory was created using the title and abstracts from Medline 2006. However, ADAM contains a significant level of redundancy between different long-form expressions due to the lack of syntactic or semantic normalization between different expressions.

Regarding the Spanish language, MedLexSp (Campillos-Llanos, 2019) is a primary step to build a unified medical lexicon for the Spanish language, which gathers terms extracted from many terminological resources such as UMLS and SNOMED CT. In addition, this resource includes 1,225 acronyms and abbreviations which were extracted from three different resources: (a) a set of Spanish abbreviations that are used in Spanish hospitals, (b) the list of Spanish and acronyms that are published in Wikipedia and (c) a collection of acronyms and abbreviations that are used on the second IberEval Challenge 2018 on Biomedical Abbreviation Recognition and Resolution (Intxaurrondo et al., 2018). These terms have also been matched to UMLS terms, adding the corresponding CUIs.

2.2.2. Corpora

Corpora represent a collection of text data that is considered the backbone of NLP fields. Two types of corpora exist, raw (unannotated), which forms a collection of machine-readable text without any modification and sense-annotated, typically created to sample a specific problem in the NLP domain and most of them manually created by experts. Following, we will describe both types of corpora that are used in disambiguate clinical abbreviations.

- Medical Information Mart for Intensive Care (MIMIC III) (Johnson et al., 2016): A comprehensive clinical data of patients admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts. MIMIC III is the only public free relational database available for researchers and it provides two million unannotated different types of notes.

- University of Minnesota-affiliated (UMN) corpus (Moon et al., 2012): It was gathered via admission notes, inpatient consult notes, operative notes and discharge

summaries, using sentences with window size 12 from left and right sides. The whole data set contains 75 abbreviations of the most frequent acronyms and abbreviations with 351 senses in total, with an average of 4.7 senses per abbreviation.

- Vanderbilt University Hospital's (VUH) corpus (Y. Wu et al., 2013): The VUH data set contains 25 abbreviations. For each abbreviation, up to 200 sentences containing the abbreviation were randomly selected and manually annotated by domain experts.

Since there are few clinical abbreviations corpora, researchers leveraged the biomedical domain corpora to increase the required data set to implement any form of machine languages architectures. Below, we describe biomedical corpora which are used in the clinical domain:

- Medical Literature Analysis and retrieval System Online (MEDLINE) ("National Library of Medicine - National Institutes of Health", n.d.): It is a bibliographic database maintained by NLM that contains over 18 million citations to journal publications in the biomedical area. Citations date back to 1947 and come from about 5,400 journals in 39 different languages, the majority of the publications are academic journals. Although the NLM controlled a set collection of term descriptors used to create Medical Subject Headings (MeSH) Headings, MEDLINE is indexed based on these headings. One of the sources included in the UMLS metathesaurus is MeSH. Because MeSH is established mainly to give indexing terms for MEDLINE, the headings are not ambiguous, but when MeSH is combined with additional sources in the metathesaurus, ambiguity arises.

- MSH WSD data set (Jimeno-Yepes et al., 2011): The MSH WSD corpus is not generated with manual annotation, instead of that, the corpus leveraged the MeSH hierarchical structure, which is used to index the MEDLINE database and included in UMLS terminologies. MSH WSD data set consists of 203 ambiguous terms, a part of them 106 abbreviations.

- ShARe/CLEF eHealth challenge 2013 Task 2 (Mowery et al., 2016): The organizers of this task annotated 300 clinical reports from the MIMIC II data set. The corpus contains 3,805 and 3,775 abbreviations in the training and test data set, respectively. The senses for these abbreviations were labeled with CUIs in the UMLS. If the sense does not have a CUI in the UMLS, it is annotated with "CUI-less."

- The Second Biomedical Abbreviation Recognition and Resolution (BARR2) (Intxaurrondo et al., 2018): As an example of the Spanish language clinical domain, an annual series of workshops held by the Spanish Society for Natural Language

Processing (SEPLN) aims to encourage all activities related to NLP in the Spanish language. BARR2 task provided a corpus of 3,343 records collected from clinical cases were extracted automatically from Scientific Electronic Library Online (SciELO) (Packer et al., 1998). This task aimed to disambiguate Spanish clinical abbreviations. The collected records were distributed on four data set training, background, development and testing data sets.

Training data set was composed of 730 abbreviations with 87 abbreviations have more than one expansion. Appendix A illustrates the abbreviations of the BARR2 corpus that have more than one expansion. The data set contains 12 abbreviations with one character, this length of abbreviations usually are excluded. In addition, different abbreviations have the same expansion, for example, both abbreviation **"F"**, **"FR"** has the same expansion **"french"**. Also **"i.v."** **"iv"** has the same expansion **"intravenosa"**. In addition to **"no"**, **"NO"** have the same expansion.

Furthermore, the corpus has annotation errors in many examples: one of "H_2O" expansion is **"centímetro de agua,"** and one of **"kg"** expansions is **"centímetro"**. Figure 2.3 shows the number of expansions frequencies for the 87 abbreviations, 62 of the abbreviations has just two expansions (71%). Also Figure 2.4 shows the size of the training examples for each abbreviations among them. 78 abbreviations have less than 50 training examples , one abbreviation has 400 examples which is **"mg"**, but it has 399 examples for **"miligramo"** and one for **"magnesio"**.

2.3. Data Representation

Like other unstructured data, text data is not manipulated directly through the computerized systems, though a mechanism is needed to represent these data in a numerical form to be prepared for manipulating through the computerized systems. Various steps could be applied to the text before feeding it to any machine model, depending on using this data. A brief description of the several pre-processing steps that are applied to clinical data will be described below:

1. **Cleaning**: A process to rearrange the unstructured data to be machine-readable text that includes removing noise data such as special characters, URLs and stop words. Also, the process could include converting all the text into lower case forms.

2. **Tokenization** consists of partitioning the whole sentence into separated individual words called tokens. Tokenization is a fundamental step in both traditional and advanced NLP tasks. Three different levels of tokenization could be applied to the raw text: character level, word level and sub-word level.

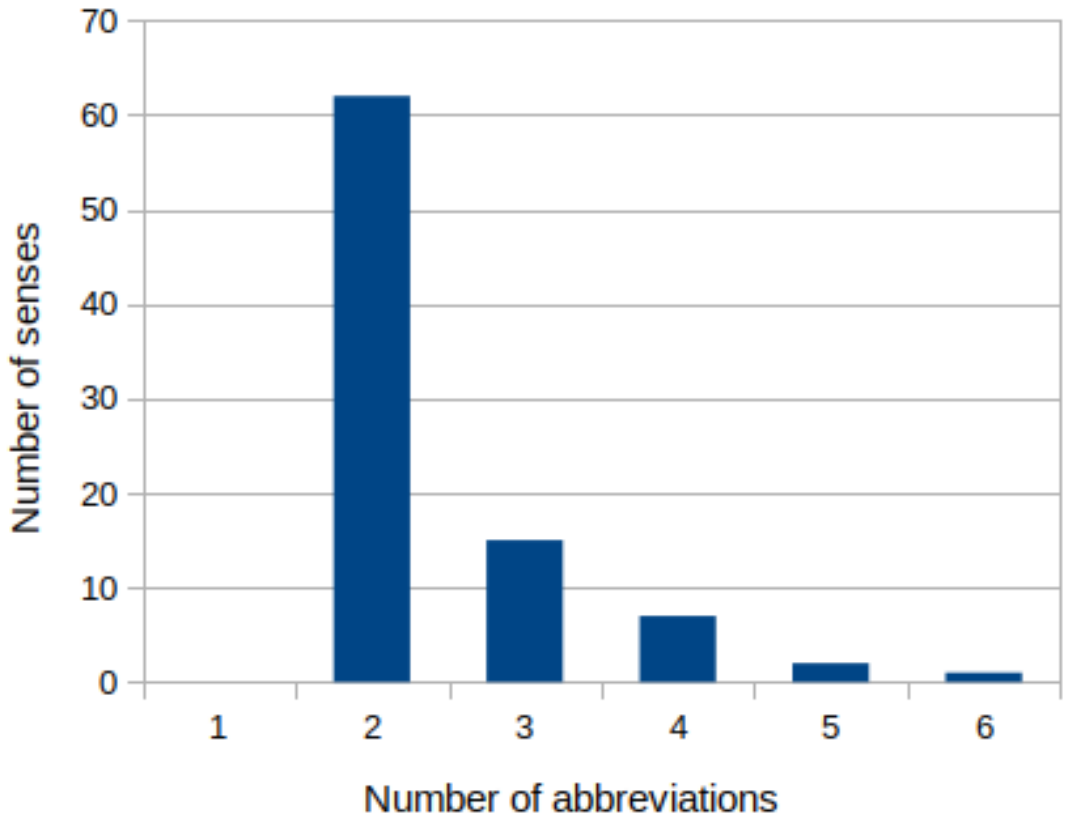


Figure 2.3: Distribution of expansions over 87 abbreviations in BARR2 corpus.

3. **Stemming and Lemmatization**: In human language, the words could have multiple forms; when computers manipulate them, words need to be normalized to their stem words. However, lemmatization tries to find the dictionary words instead of truncating them as stemming does; it is more accurate than stemming.

4. **Part-Of-Speech (POS) tagging** obtains a specific label attached to each word in an input sequence in a corpus to identify the part of speech. Other grammatical categories such as tense number (plural/singular) are added in so many other situations.

Figure 2.5 illustrates an example of these pre-processing steps which could be performed on the text. The sentence **"which was accomplished in _%#MM#%_ of 2001 for a missed AB at approximately 7 weeks gestation"** was cleaned from special characters such as **#** and **%**. Then, it is tokenized into its separated words. Each token in the sentence was identified to its root, for example, the root of ***was*** is ***be***. Another pre-processing step is assigning POS tags to each token in the sentence, **"which"** is a **determiner**, **"week"** is a **noun**... etc.

NLP applications expect an input belongs to a vector space and this is called feature extraction or vectorization process. So a need to transform text data into representative

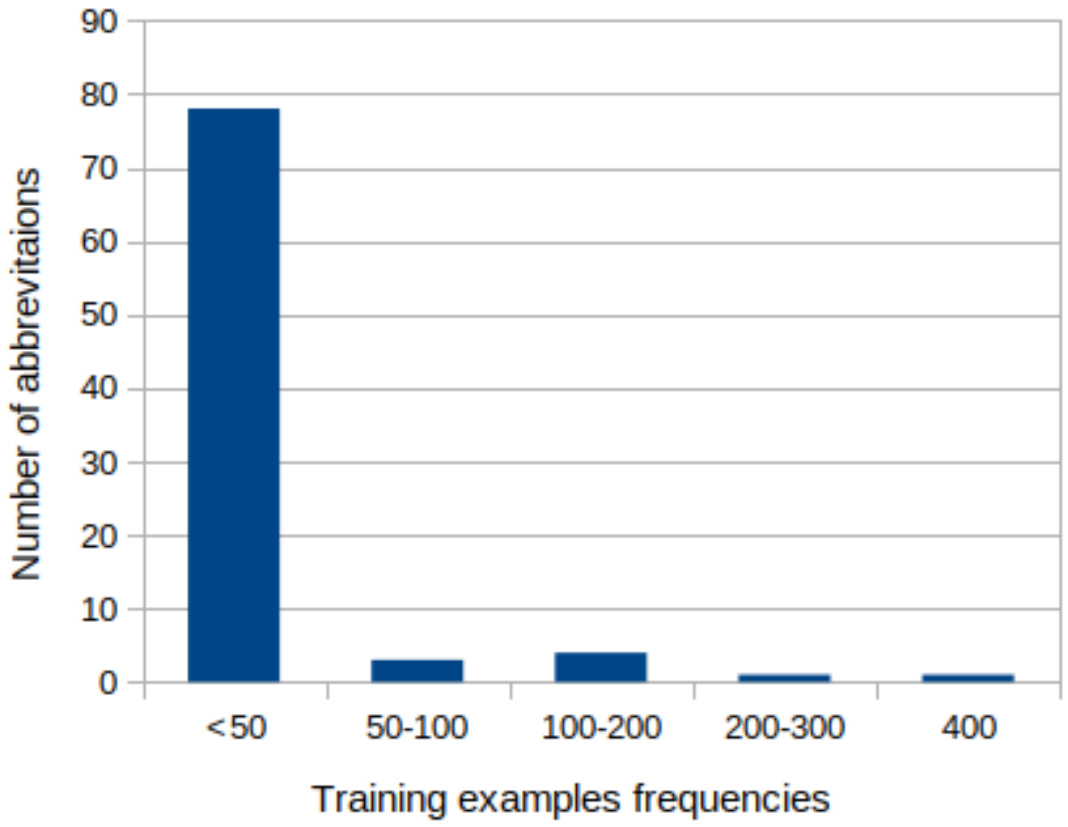


Figure 2.4: Number of training examples distribution in each 87 abbreviation in BARR2 corpus.

numeric data is a mandatory step toward achieving the goal of NLP applications. Table 2.1 illustrates a set of features that are used in the previous works related to the classification task.

For decades, researchers used many statistical approaches to represent a text like One Hot Encoding (OHE) which is considered the simplest form of the word representation. The idea behind OHE representation is that the vocabulary of the whole corpus is associated with an index. The length of the index represents the total number of words in the corpus. Then, a vector representation for each word of length n-dimension array is set to zero except for its corresponding index. Count Vectorizer is another approach that depends on words frequencies. Generally, the model creates a matrix based on specific metrics depending on the goal of the representations, for more information on both methods, see (Pilehvar and Camacho-Collados, 2020).

For WSD, a word-context matrix is used to measure words similarity. The matrix rows correspond to words and the columns to the context (words surrounding the center word). In Bags Of Word (BOW) (McCray et al., 1994) approach each document in the corpus is represented in a vector with the whole unique words numbers length. Each element in

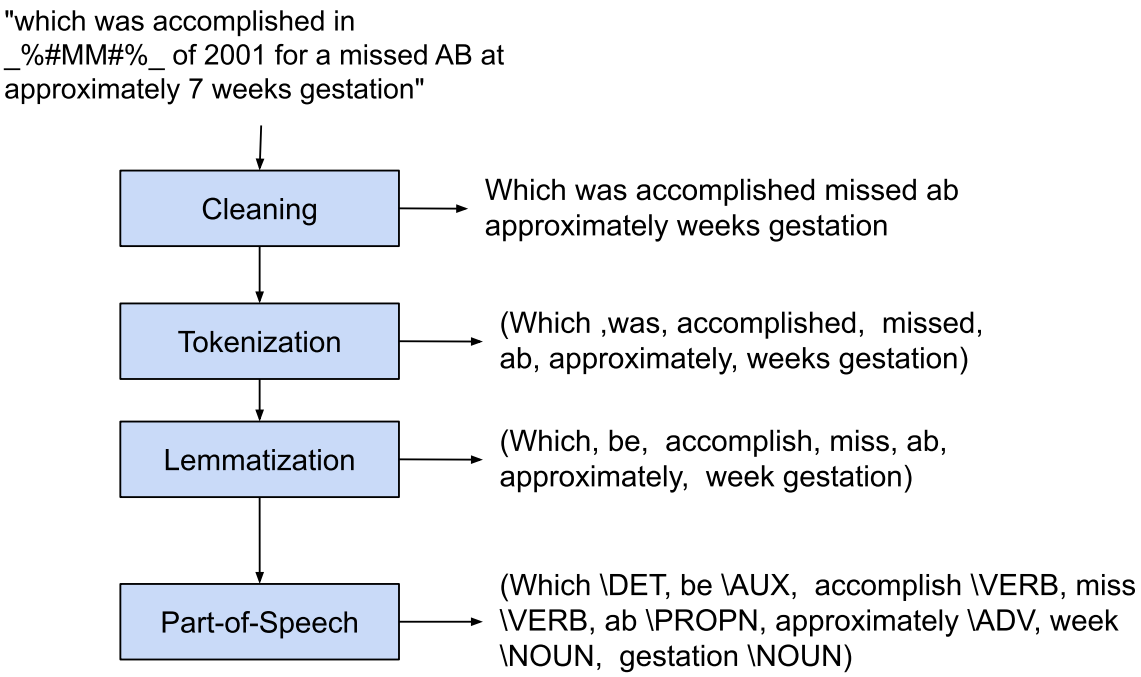


Figure 2.5: An example of some pre-processing steps which are performed on text.

this vector represents the frequency of each word in the specific document.

N-grams (Jurafsky and Martin, 2009) is another approach that is used to represent the text given the previous word. The most simple language modeling aims to estimate the probability of the last word of a group of words. Then it assigns the probabilities to the whole sequence. A group of words could be composed of two words (bi-gram), three words (tri-gram), or could be any number of words (n-grams).

Furthermore, Term Frequency - Inverse Document Frequency (TF-IDF) (Jones, 2004) is a statistical approach to generate feature vector for each word in corpora by creating two matrices. The first one is about how many times the words are used in a document (term frequency). The second, dividing the total number of documents by the number of documents that contain the word to represent the percentage of the existence of the word in a document (inverse document frequency). A word score is calculated by multiplying these two values. The greater the score, the more important the word in that document is.

Among the most recent and remarkable additions to word vectors is distributed vector representation or embedding. Embedding has received a lot of attention and has become a technique in the toolbox of NLP researchers. Semantic similarity between two words can be described in terms of their contexts (i.e., words with similar contexts have similar meanings), according to distributed hypothesis.

Word embeddings, which are relied on distributed hypotheses, display words as dense, low-dimensional, fixed-length vectors in a continuous vector space, ensuring that words with similar meanings are closed together. Hundreds of dimensions make up a word em-

TABLE 2.1: A summary of features which was used for disambiguation clinical abbreviation.

Feature Categories	Feature Types	Description	Features
Linguistic	Lexicon-based	A dictionary or the vocabulary of a language	Bag of words (BoW)
	Orthographic	A set of conventions for writing a language	Spelling: capitalization: special character
	Syntactic	Syntactic patterns presented in the text	Part-of-Speech
Statistical	Statistical corpus features	Features generated through basic statistical methods	Word length, TF-IDF, co-occurrence
	Vector-based representation	One-hot encoding, Word embedding, Sentence encoding, Paragraph encoding	Word2vec,BERT, one-hot character-level encoding.
General document	Pattern and rule-based feature	A label for a note if certain rules are satisfied	Logic (if-then) rules and expert system
	Document structural	Structural and organizational patterns presented in the text and document	Section information

bedding and each dimension represents a feature. As a result, the meaning of a word is distributed across dimensions in a word embedding. Word embedding is particularly well suited to deep learning models, which use matrix operations to find high-level representations of text data across multiple layers.

The limitation of dimensionality and lack of syntactic and semantic information in representations is overcome by embedding, which translates variable-length text to dense vector representations. Furthermore, embeddings are unsupervised learned, capturing knowledge in a huge unlabeled corpus and transferring it to downstream tasks using small labeled data sets. As a result, embedding has become an unavoidable choice for text representation in the recent deep learning era.

Static word embedding such as Word2Vec (Mikolov et al., 2013), Fast Text (Bojanowski et al., 2017) and Global Vectors for Word Representations (GloVe) (Pennington et al., 2014) are considered independent context models, which means assigning a single vector representation for a word ignoring the context in which it appears. Word2vec proposed two models based on feed-forward neural architecture to generate embeddings. Continuous Bag of Words (CBOW) and Skip Gram (SG). The fundamental distinction be-

tween them is how they make predictions. CBOW aims to predict the targeted word using the surrounding context. On the other hand, the SG model aims to predict the surrounding context given the targeted word.

The GloVe is another example of learning word embedding. It does not use a neural network to just predict but also it uses a co-occurrence matrix. For a corpus of vocabulary size **V**, the co-occurrence matrix is **V x V**. The frequency of the words that occur together inside a set window size is represented in the matrix. So that, the vector embedding is learned by minimizing the error between co-occurrence statistics predicted by the model and global co-occurrence statistics observed in the training corpus.

Availability of a massive amount of data does not mean covering all the vocabulary in a specific domain. Out Of Vocabulary (OOV) term refers to the missing word representations in the pre-trained models, which is the specific problem that FastText solves it by extending the word2vec SG model with internal sub-word information to learn the word presentation at character level instead of word level. Besides learning embeddings to words that appear in the training data, the model also learns their character-level embeddings. Thus, the embedding of an unseen word could be calculated by averaging the embedding vectors for its' characters. There are many open access pre-trained models which are built from clinical and biomedical resources, which are used later to disambiguate clinical abbreviations (Pyysalo et al., 2013, Beam et al., 2020 Y. Wang et al., 2018).

Even though the static word embeddings have carried a new shift of word encoding for many NLP tasks, the representation of the word in numeric form is still the main challenge of NLP, especially WSD tasks. The primary limitation of these technologies is that they previously provided a set of words with their embeddings without any concern for the context. For example, static embedding models will assign the same vector to both words **"bank"** in the sentence **"The employee left bank and played on the bank of river,"** even though those two words have completely different meanings (despite having the exact spelling) and thus, should not be represented using the same vector.

To tackle this problem, a new type of word embedding appears to solve this. Embedding from Language Model (ELMo) (Peters et al., 2018) is a character level embedding approach that takes into account the whole sentence to assign embedding to its' words. ELMo is based on a bi-directional Recurrent Neural Network (RNN) which means that embeddings will be varied for the same word based on its' surrounding words. ELMo presents a new important task which is the ability to predict the next word in a sequence of words without needing for annotated data. It is called Language Modeling.

Following improvements are found in different Language Modeling approaches to represent the semantic meanings of the words. Another approach is Bidirectional En-

coder Representations from Transformers (BERT) (Devlin et al., 2019) which uses Transformers (Vaswani et al., 2017). The transformer is based on Encoder-Decoder architecture which performs better than previous RNN-Long Short-Term Memory (LSTM) approaches with handling long terms dependencies and proves its efficiency in machine Translation task (Q. Wang et al., 2019).

BERT presents two novel pre-trained language model training strategies based on the encoder side of the transformers to look forward and backward of a sequence at the same time, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The MLM strategy that randomly masks 15% of the input sequence by replacing them with [mask] tokens and then, the model is trained to predict them. However, some NLP tasks need to predict something about the whole sentence, not to a specific word such as sentiment analysis (Feldman, 2013) and question answering (Lukovnikov et al., 2019). Therefore, NSP represented a way to handle a relation between two sentences.

XLNet (Yang et al., 2019) is a language modeling that does not differ from BERT, but it redefines the MLM training strategy as a permutation language modeling, where all tokens are predicted in random order, on the contrary of BERT, which predicts a fixed number of tokens. Figure 2.6 summarizes the list of word embeddings models which were used to disambiguate clinical abbreviations.

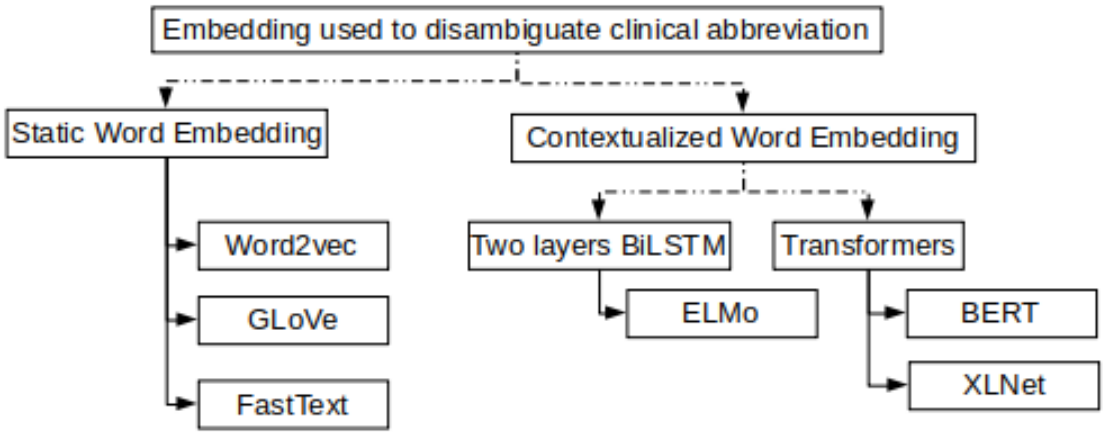


Figure 2.6: List of word embedding approaches that was used in clinical abbreviation disambiguation.

2.4. Evaluation Measurements

When evaluating and comparing different classification models or machine learning techniques, performance indicators are required (Ma et al., 2014). Many metrics can be used to evaluate the performance of a classification task (Müller and Guido, 2016). These metrics prove helpful at various stages of the development process, such as comparing the

performance of two different models or analyzing the behavior of the same model by tuning different parameters. There are various ways of evaluating classification performance. The most commonly used evaluation metrics are accuracy, confusion matrix, precision and recall which we will explain below.

Confusion Matrix

It is a table that keeps count of the number of instances in a data set that fall into a specific category. In a binary training set, the class label can have two possible values: positive class and negative class. As shown in Table2.2, the number of positive and negative instances correctly predicted by a classifier are referred to as True Positive (TP) and True Negative (TN). False Positive (FP) and False Negative (FN) are the instances that are incorrectly classified.

TABLE 2.2: Confusion matrix for binary classification.

		Predicted class	
		Positive	**Negative**
Actual class	**Positive**	True Positive	False Negative
	Negative	False Positive	True Negative

Precision

The Precision is calculated by dividing True Positive elements by the total number of positively predicted units (column sum of the predicted positives). True Positive elements have been labeled as positive by the model. They are actually positive, whereas False Positive elements have been labeled as positive by the model but are actually negative.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2.1}$$

Recall

The recall is calculated by dividing True Positive elements by the total number of positively classified units (row sum of the actual positives). False Negative elements are those

that have been labeled as negative by the model but are actually positive.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{2.2}$$

The Recall reflects the model's predictive accuracy for the positive class: intuitively, it investigates the model's ability to find all positive units in the data set.

F-score

F1-Score evaluates classification model performance by aggregating Precision and Recall measures using the harmonic mean.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.3}$$

F-score is a evaluation measurement for multi-classification tasks. The idea behind computing the f-score for the multi-classification task is to consider one class as a positive class and compute the f-score for it. The other classes are considered negative. The overall f-score could be averaged using one of the following strategies:

- **macro**: averaging will be calculated by computing the metrics independently taking the average: hence, all the classes will be treated equally no matter the size of each one.
- **weighted**: averaging will be calculated by computing the mean f-score for all classes, considering the proportion of each class in the data set.
- **micro**: averaging will be calculated first by computing false negatives, false positives and true positives. Then precision, recall and f-score will be calculated based on this counting.

Accuracy

Accuracy is a popular metric in classification tasks. The Accuracy takes into account the sum of True Positive and True Negative elements in the numerator and the sum of all model results in the denominator. True Positives and True Negatives are correctly classified by the model and are located on the confusion matrix's main diagonal. At the same time, the denominator includes all elements located outside of the main diagonal

that the model has incorrectly classified. In other words, consider picking a random unit and predicting its class. Accuracy is the likelihood that the model prediction is correct.

$$Accuracy = \frac{\#\ correct\ answers\ provided}{\#\ all\ provided} \tag{2.4}$$

Accuracy returns an overall measure of how well the model predicts the entire set of data. The single individuals in the data set are the fundamental metric element: each unit has contributed equally and in the same influence to the accuracy estimation. When we consider classes rather than individuals, there will be classes with many units and others with a small number.

In this case, classes with a high population density will be given more weight than classes with a low population density. As a result, Accuracy is best suited when we only care about a single individual rather than multiple classes. Consequently, due to the imbalanced abbreviations data set, most of related work chose accuracy as an evaluation measurement rather than other measurements.

2.5. Classification Approaches for WSD

WSD approaches are frequently categorized based on the main source of knowledge used in sense differentiation (Edmonds and Agirre, 2008). Dictionary-based or KB methods are those which directly depend on dictionaries, thesauri and lexical knowledge bases instead of on corpus evidence. Unsupervised methods are those that work without (nearly) any external information and work directly from raw, unannotated corpora (adopting terminology from machine learning). Finally, annotated corpora are used in supervised and semi-supervised WSD as seed data to train a model. Whatever approach is used, all WSD systems extract contextual features from the target word (in text) and compare them to a list of its senses information to determine the correct one. In the following sub sections, we will review the WSD approaches that were applied on disambiguating clinical abbreviations.

2.5.1. Machine Learning Approaches

One of the most successful approaches and the most applied in the clinical domain is supervised approach. Supervised approaches learn to map input Space $\boldsymbol{X}$ to a discrete set $\boldsymbol{Y=\{1,2,..N\}}$, from a training set. The training set contains a number of training instances $\boldsymbol{m}$, hence, the learned function is $\boldsymbol{S = ((x_1, y_1), ..., (x_m, y_m))}$. Each pair represents a feature vector $\boldsymbol{x}$, which represents a numerical vector describing the relevant properties about the

textual example and the value which is associated with each training instances ***y*** are called classes.

For example, suppose that we want to disambiguate the abbreviation **"BAL"** which has two expansions (senses), as mentioned in the UMN corpus, **"bronchoalveolar lavage,"** and **"blood alcohol level"**. Each sentence in the corpus containing **BAL** abbreviation is considered an instance that will be transformed into training example **x**, annotated with its correct expansion (either ***bronchoalveolar lavage*** or ***blood alcohol level***). Both its expansions are considered the classes that belong to space **Y**.

Supervised Machine learning approaches are categorized into a traditional machine learning algorithms and Neural Network approaches. Supervised Machine learning approaches could be divided, based on the induction principle they employ to build their classification models, into three categories.

1. **Probabilistic Methods**: It uses statistical methods typically estimate a set of probabilistic parameters that express the conditional probability of each given category in a given context (described as features). These parameters can then be combined to assign the set of categories with the highest probability on new examples such like Naïve Bayes (NB) (Hart et al., 2000), Maximum Entropy (ME) (Berger et al., 1996).

2. **Methods Based on the Similarity of the Examples**: These methods disambiguate a new example by comparing it with a set of pre-generated vectors (a set of senses) by applying one of the similarity metrics. Some examples of these methods are Vector Space Model (VSM) (Schütze, 1992) and k-Nearest Neighbor (kNN) (Ng and Lee, 1996).

3. **Methods Based on Discriminating Decision Rules**: These methods learn specific rules for each word senses. Given a polysemous word, the system chooses the sense that confirms some of the rules that determine one of the senses, like Decision Trees (DT) (Black, 1988).

4. **Linear Classifiers and Kernel-Based Approaches**: Linear methods classify a data set into a discrete number of classes based on a linear aggregation of its informative variables. However, kernel approaches predict nonlinear problems by applying linear classifier methods through transforming the nonlinear data into a high-dimensional space. One of the well-known algorithms is Support Vector Machine (SVM) (Boser et al., 1992) which will be described in detail in the following chapter.

Most of these algorithms were tested for disambiguating clinical abbreviations. In the

following we will discuss the previous works that applied supervised machine learning approaches. Current traditional supervised machine leaning approaches, which were implemented to disambiguate clinical abbreviations, varied by types of features used. There are techniques to increase the training examples by applying different data augmentations techniques.

(Joshi et al., 2006) and (Moon et al., 2012) tested different sets of features by NB, DT and SVM. (Joshi et al., 2006) identified four sets of features (POS tags, Uni-grams, Bi-grams and the combination of the previous three) on 16 abbreviations from the UMN data set. A separated classifier for each abbreviation was implemented. The four features sets were tested on a flexible window size range between 1 to 10. (Joshi et al., 2006) models achieved accuracy exceeded 90% for the three machine learning algorithms regardless of the distributions of senses (classes) on each abbreviation data set.

Furthermore, uni-gram features and all-features combinations achieved a very limited performance. In addition to (Joshi et al., 2006)'s features set, (Moon et al., 2012) also used BOW, section information as a local contextual feature, the position of targeted abbreviations based on its local and global locations and CUI as features. These features were tested using 50 abbreviations from the UMN data set with window sizes ranging from 3 to 60 (Moon et al., 2012). SVM achieved slightly high accuracy with window size 40 via BOW and CUI features. Also, they concluded that 125 annotated samples for each sense are required as a minimum to train the supervised machine learning algorithms. In addition to, a larger left-sided window than a right-sided one was better to get the best accuracy on SVM.

CLASSE-GATOR (Kashyap et al., 2020) created a sense inventory of 1,257 abbreviations and 8,287 expansions from 31,764 prenatal-exposure papers. This sense inventory was used to create the training examples from 2,227,674 PubMed Central (PMC) Open Access Subset. For each pair of (abbreviation, expansion) 150 examples were extracted to build the prediction model. A separated Logistic Regression (LR) classifier was built for the set of abbreviations. 1,000 test examples for each abbreviation were extracted from the MIMIC III corpus. The model was evaluated with two data sets, 9 abbreviations from 245 clinical notes that were annotated manually. 52 abbreviations from the UMN and were found in PubMed. CLASSE-GATOR achieved an average accuracy of 87.9 % across 1,256 acronyms.

(Y. Wu, Xu, et al., 2015) applied advanced steps towards feature extraction to combine a set of standard features with generated embeddings. (Y. Wu, Xu, et al., 2015) got the vector embedding for the context of the target abbreviations by training a neural network architecture (Collobert et al., 2011) on unannotated MIMIC II corpus. Then, they tested three derived features vectors, the first was generated by averaging the set of

embedding vectors for the context (SBE(w)). The second was generated the embedding vectors for the context with taking into accounts the directions of the surrounding context (LR-SBE(w)). The third feature vector was generated by getting the MAX score of each embedding dimension over all the surrounding words (MAX-SBE(w)). In addition, these feature vectors were combined with a set of conventional ones like position, the distance of the word context. A separated SVM classifier was implemented for each abbreviation in two data sets (VUH, UMN). MAX-SBE(w) with the conventional features achieved average accuracy of 93.01% and 95.79% on VUH, UMN, respectively.

(S. V. Pakhomov, 2002) proposed a methodology for gathering training data for supervised machine learning approaches. The methodology was based on the idea that an abbreviation's expansion (or sense) and the abbreviation itself occur in similar contexts. The expansions that indicated the senses of an ambiguous abbreviations were found in clinical records. After the expansion was found in the corpus, the context in which it was found was recorded and utilized to train statistical predictive models for disambiguation abbreviations. The methodology was applied on 6 abbreviations which its training data set was gathered from 10,000 clinical notes from Mayo clinic medical system. A separated ME classifier was implemented for each abbreviation. The average accuracy achieved was 89.14%. Furthermore, a unified ME classifier was performed on the whole data set, the average accuracy achieved was 89.17%.

(S. Pakhomov et al., 2005) improved his previous work (S. V. Pakhomov, 2002) by leveraging the World Wide Web, MEDLINE Abstracts and 1.7 million clinical notes from Mayo clinic medical systems to increase the training examples and thus, the number of disambiguated abbreviations. A separated VSM was implemented for 8 abbreviations with 64 senses (4,314 instances). The best average accuracy achieved was 67.8% when the training data composed of the clinical notes and MEDLINE abstracts.

(Finley et al., 2016) integrated CASI sense inventory to generate a training data set. Elasticsearch (Gormley and Tong, 2015) algorithm was used to auto-generate labels for 207 senses that were collected from the inventory on 827,647 clinical notes from the Fairview Health Services system. Co-occurrence counts within a fixed window size surrounded by the targeted abbreviations were used to generate the feature vectors with a BoW representation. Several approaches were used, supervised machine learning algorithms (NB, LR, SVM), cosine similarity and hyper-dimensional indexing (RI, BSC). Two data sets were tested on these models. the LR model achieved the best average accuracy on the UMN, the auto-generated corpus, of 96.6% and 99.0%, respectively.

Unsupervised approaches of word sense discrimination that are knowledge-light do not rely on external information sources like machine-readable dictionaries, concept hierarchies, or sense-tagged text. They do not assign sense tags to words, instead, they use

information from unannotated corpora to distinguish between word meanings.

This approach was applied by (M. Peng and Quan, 2020) which implemented K-mean cluster algorithms on the UMN data set to disambiguate clinical abbreviations. BioELMo (Jin et al., 2019a) was used to get the contextualized representations for the feature vectors, which was trained on the PubMed corpus with 10 million abstracts (Jin et al., 2019b). Three different clusters were used to extract senses (10, 15 and 20(. The majority vote was used to determine which sense belongs of these clusters. Using the second hidden state layer of BioELMo to generate the feature vectors improved the performance of the model on 20 clusters. The accuracy achieved was 94.6% and 95.4% on the training and testing data (400,100 samples for each abbreviation), respectively.

2.5.2. Deep Neural Network Approaches

Deep Neural Network or Deep learning is just a form of machine learning. Perceptron or neuron is the primary essential unit in the neural network and was inspired by human neuron brain cells. Each neuron is composed of five main components: input, weight, bias, activation function and output, as shown in Figure 2.7. A collection of neurons that do specific mathematical operations together is named a layer. Thus, a neural network contains three types of layers: input layer, hidden layer and output layer. The word **"deep"** represents the fact that the model is composed of a stack of layers chained together (Chollet, 2021).

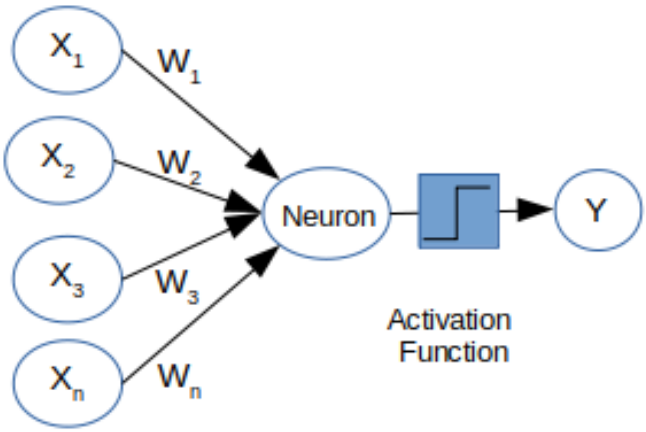


Figure 2.7: Single neuron components.

The input layer receives numeric values converted from the raw data. These data are multiplied by weights that are passed to the most critical part of any deep neural network, hidden layers, which perform mathematics operations, trying to learn from the data by minimizing an error/cost function. The input data passes from a hidden layer to another and then the output layer produces the predicted output.

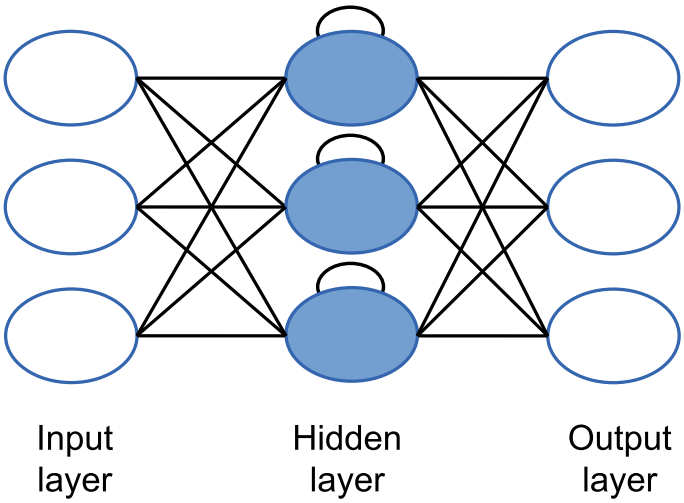


Figure 2.8: An illustration for Recurrent Neural Network architecture.

Two principal processes are executed during the training, feed propagation and back-propagation. In the feed propagation, as shown in equation 2.5, each weight $\boldsymbol{w_i}$ is multiplied by its input data $\boldsymbol{x_i}$, then the bias $\boldsymbol{b}$ will be added. The results for all inputs are summed and passed to the neuron, where an activation function (like (softmax (Goodfellow et al., 2016), RelU (Agarap, 2018)) is executed to decide which neuron should be activated to extract features. All neurons follow the exact process in all the hidden layers. In the end, the result is sent to the output layer. The predicted value is compared with the actual value. Then, the cost function will be applied to minimize the error between the predicted and actual output. And thus, the back-propagation process will be done to adjust the weights and biases again. These processes are executed many times until the result fit all the training model.

$$\sum_{i=0}^{n} w_i.x_i + b \tag{2.5}$$

Different deep neural network models have been applied in different NLP tasks. Neural networks can be used for classification by selecting appropriate activation functions for the output layers (e.g., using a soft-max layer). Four deep neural network architectures will be described below: Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Convolutional Neural Network (CNN) (Schmidhuber, 2015), Attention (Bahdanau et al., 2015) and Transformers (Vaswani et al., 2017).

Understanding languages could not happen from scratch every time we read a text. To understand the whole sentence, we need to understand each word. Traditional neural network architectures do not figure out this problem. The Recurrent Neural Network (RNN) (Medsker and Jain, 1999) is the first deep neural network model that addresses this problem. RNN architecture enables a loop that passes information from one step to

the next in the network, which successfully manipulates the sequence data (see Figure 2.8). However, sometimes we need information not just from the recent previous in the actual text. Instead, we could need far information (long dependencies) to understand the sentence. Hence, RNN in its current architecture fails to solve this gap.

LSTM (Hochreiter and Schmidhuber, 1997) is a special form of RNN that is capable of learning with long dependencies. Figure 2.9 illustrates that LSTM differs from RNN in that it is composed of a cell state that enables to save or regret information from the previous context based on a set of mathematical operations. LSTM had been the-state-of-the-art in many NLP tasks, especially in WSD classification task (M. Le et al., 2018).

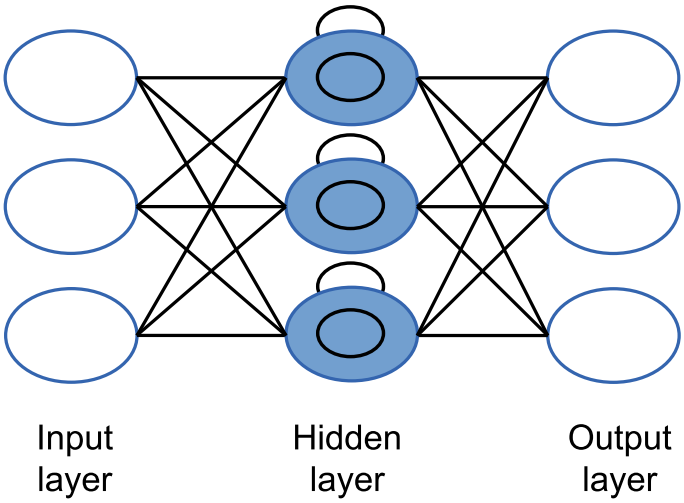


Figure 2.9: The LSTM architecture that displays the difference from the traditional RNN architecture.

Machine translation and speech recognition are NLP tasks whose sequence output length could differ from the input sequence length. Thus, new challenges raised for deep neural networks architecture to deal with. The LSTM could perform this, but for equal length in both input and output text, this is inapplicable in natural languages. So that, new advancements in neural networks architectures were proposed to deal with this problem which is known as Encoder-Decoder or Sequence to Sequence (Seq2Seq) (Sutskever et al., 2014).

As shown in Figure 2.10, the Encoder-Decoder model consists of two blocks. Both the encoder and the decoder are composed of a stack of LSTM cells. The input sequence is fed to the encoder to generate one fixed-length vector (commonly known as the last hidden state or context vector) that represents the whole input sequence. Then, the context vector is fed to the decoder to predict the output sequence.

One weakness of seq2seq architecture is that the whole input is encapsulated in one hidden state, which fails to capture the complete information of the input sequence, especially for those longer than the training input sequences. Allowing the decoder to access

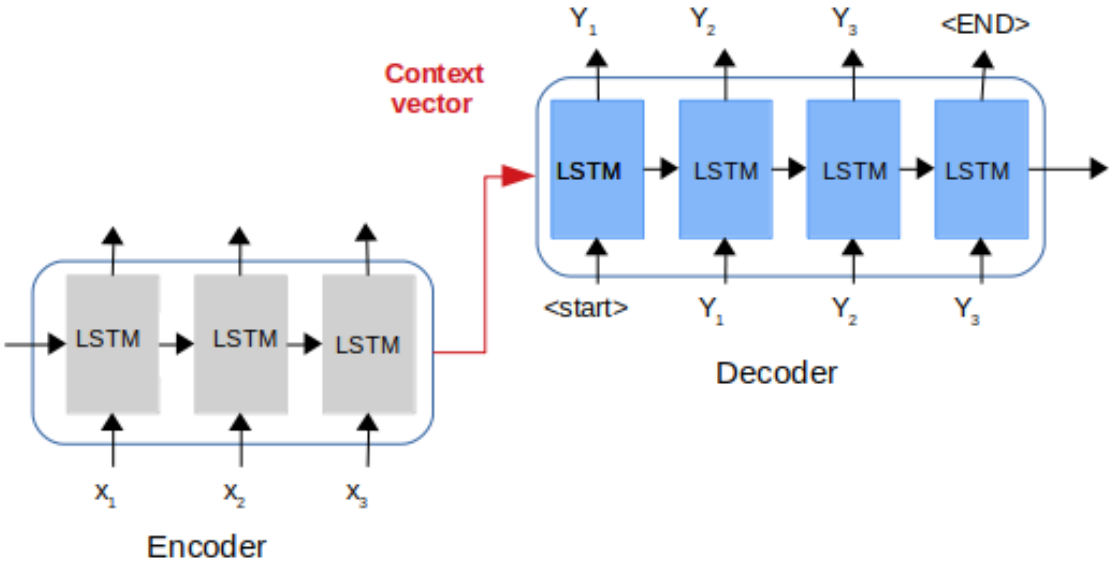


Figure 2.10: Sequence to Sequence model architecture.

all generated hidden states in the encoder is one solution to work this weakness. This mechanism is called Attention (Bahdanau et al., 2015).

The attention mechanism consists of two alterations to the previous model, as shown in Figure 2.11. First, the decoder will access all the hidden states generated from the encoder instead of accessing one hidden state. Second, the decoder will assign weights based on the previous state to "pay attention" to the most related input for the current state. Those weights could be learned during the training process.

Later, Transformers (Vaswani et al., 2017) alter this architecture by replacing the LSTM inside the Encoder-Decoder with a combination of self-attention layers and Feed-Forward Neural Network (FFNN) (Figure 2.12). The self-attention mechanism shifted the processing of sequences from a sequence form to fully parallel, allowing training with a larger corpus efficiently. Furthermore, a separated representation for each token in the sequence that depends on the surrounding context is now possible through the self-attention mechanism.

Transfer learning is a type of machine learning method that trains deep neural network architecture on huge unannotated data for the general task. Then, the generated model could be used as a starting point on another task. Transfer learning has been a common approach in computer vision for several years (ResNet (He et al., 2016), Imagenet (Deng et al., 2009)). The approach was hard to adopt in the NLP domain because the natural language is inherently more difficult than the image. Furthermore, NLP tasks require many annotated data to achieve high performance. With transformers architecture, the transfer learning approach is viable on NLP. Different sub-models from transformer architecture has been generated, encoder-only models such as BERT (Devlin et al., 2019), ELECTRA

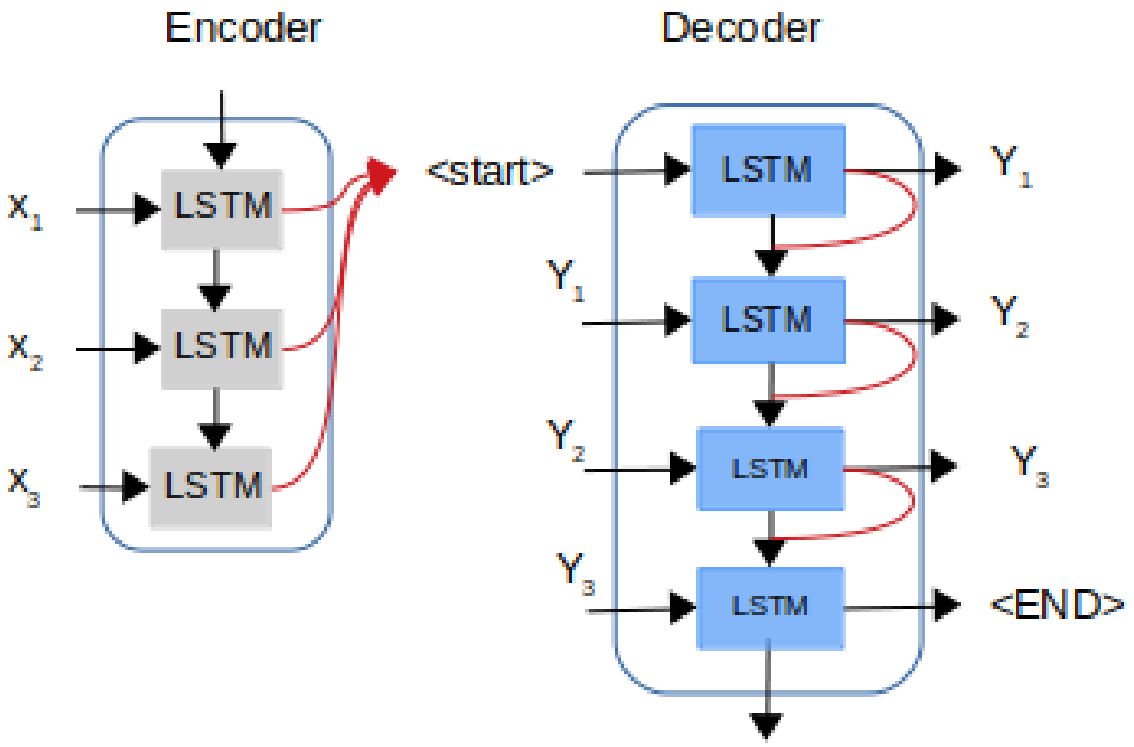


Figure 2.11: Sequence to Sequence model architecture with attention mechanism.

(Clark et al., 2020), decoder-only models such as GPT-3 (Floridi and Chiriatti, 2020) and encoder-decoder models such as BART (Lewis et al., 2020).

Transfer learning aims to transfer knowledge across different domains to solve the lack of data in some specific domains (Zhuang et al., 2021). The traditional machine learning approaches depend heavily on annotated data, which is considered a time-consuming and expensive process to obtain. The semi-supervised approach is partially deal with this problem by requiring small labeled data. However, it still needs a large amount of unannotated data. Nevertheless, there are many domains where it is not easy to collect unlabeled data. Thus, transfer learning has been a promising approach to solve this issue.

Language modeling architectures could be used as a transfer leaning by fine tuning mechanisms. As shown in Figure 2.13 (a), instead of training the model from scratch, the model weights are adjusted from the previous training step and a classifier layer is put at the bottom of the model to re-train the model with a specific data set. The fine-tuning process could involve training all the model layers or part of its depending on the goal and the model's performance, in addition to the data set size and type. On the other hand, in Figure 2.13 (b), the model is used as contextualized word embeddings feature extraction, the extracted vector is used as input feature vectors in any classification model.

Fine-tuned language modeling for text has been executed for many NLP tasks. Thus,

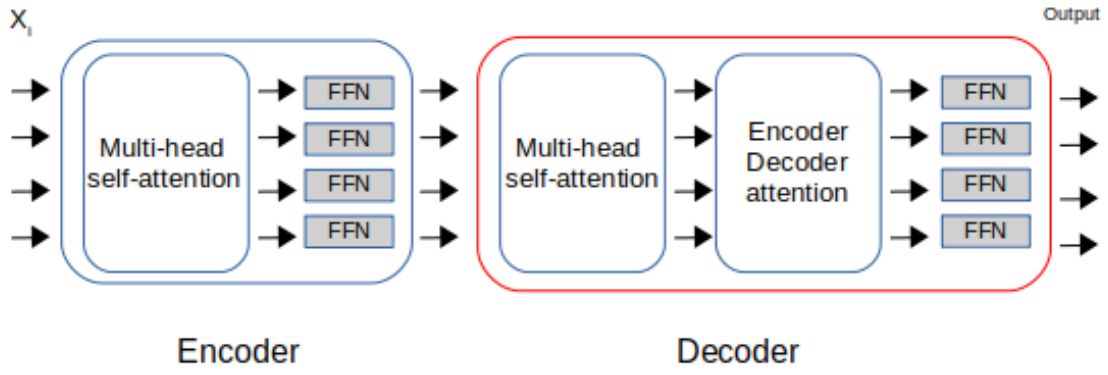


Figure 2.12: Novel transformer architecture.

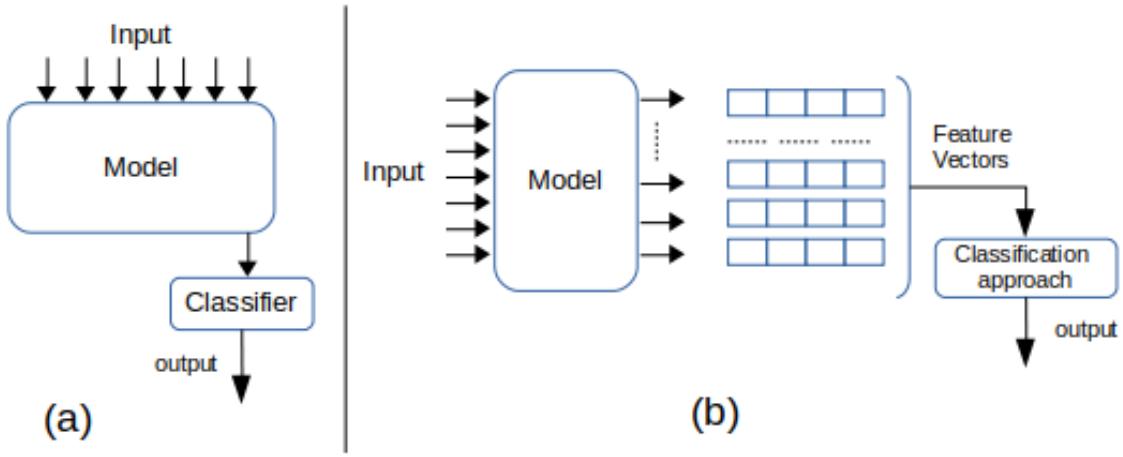


Figure 2.13: Two different ways could be used by language modeling, (a) fine tuning model or (b) features extraction model.

several works leverage these models to disambiguate clinical abbreviations. (Li et al., 2019) proposed a model to disambiguate 30 abbreviations from the UMN data set by fine-tuning ELMo. MIMIC III was used to train the ELMo model in an unsupervised way. The clinical topic representations were computed from the pre-trained Doc2vec model (Q. V. Le and Mikolov, 2014). A combination of sentence representation from the trained ELMo and topic representations for each sample in the training data set were fed as input to the model. A fine-tuning model for each abbreviation was implemented. The average accuracy was 74.76%.

(Kim et al., 2020) used relative position encoding to hide the length of the candidate expansion during the training phase to overcome the problem that a set of candidate expansions could be in different lengths. The classifier was fed with the embeddings of candidate expansion contextualized by the context as input. The idea was applied using XLNet language modeling and tested on three different data sets. The accuracy achieved on the UMN data set was 98.34%.

Convolutional layers are another form of deep neural network architecture that uses a weight-sharing scheme to allow the network to learn local features. It is beneficial in image recognition (Zeiler and Fergus, 2014). Convolutional layers exploit the local structure of input data, allowing convolutional networks to learn features of interest from training data rather than requiring hand-crafted features. Despite the potential for automatically learning features of interest, only a few studies have used this approach to gain this kind of recognition.

A Convolutional Neural Network (CNN) (Schmidhuber, 2015) has multiple hidden layers of convolutional layers. As shown in Figure 2.14, CNN is composed of different neural networks layers starting from the embedding layer that represents the input. Then, a convolutional layer which is responsible for feature extraction is added. Pooling layers are implemented between convolutional layers to reduce the spatial size of the input size. Flatten layer is used to avoid over-fitting the model. After that, fully connected layers are implemented to capture the final representation details. Finally, the output layer predicts the classes.

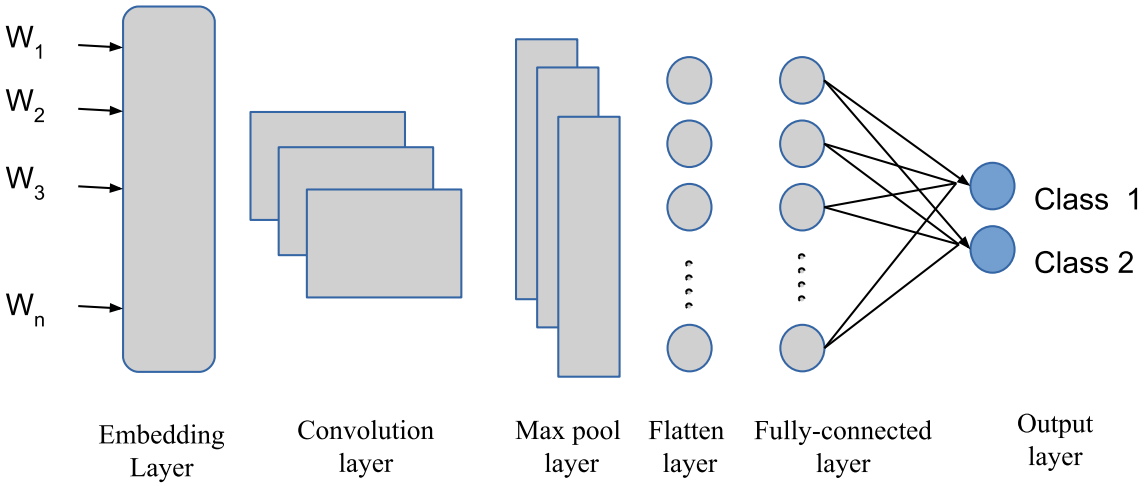


Figure 2.14: Convolutional Neural Network Architecture.

(Joopudi et al., 2018) used 117,526 clinical notes from 1,001 de-identified longitudinal patient records from Cleveland Clinic, Ohio (USA) to generate two training data sets. Data sets are composed of annotated data for 383 abbreviations by applying reverse substitution method and 206 abbreviations that manually annotated (169 used in the study), in addition to 50 abbreviations from UMN. A separated CNN was implemented to train the model for each abbreviation using a pre-trained word embedding generated from PubMed (Pyysalo et al., 2013), in addition to POS and clinical notes features. (Joopudi et al., 2018) reported three experiments results; the first experiment was conducted on the data set that were generated by the reverse substitution methods (383 abbreviations)wih an average accuracy of 97.92%. In the second experiment, the models were trained on the auto-generated training data set and tested on the manually annotated (169 abbrevia-

tions), achieving an micro average accuracy of 77.83%. The last experiment was trained and tested on 50 abbreviations from the UMN data set obtaining a average accuracy of 95.14%.

(Skreta et al., 2019) increased the training samples for each expansion with texts containing closely related medical concepts. The relation was identified by embedding distance for a given abbreviation. Both local and global features were used to generate the feature vectors. A separated CNN classifier was trained for each abbreviation. (Skreta et al., 2019) reported that generated feature vectors from combining local and global features improve the model's performance. But the proposed augmentation method decreases the performance of the model when it was tested on MIMIC III. The best accuracy achieved was 76% on the augmented UMN data set with global features. On the other hand, the best accuracy achieved was 94.4% tested with MIMIC III without the augmentation methods.

(Skreta et al., 2021) implemented the same their previous work (Skreta et al., 2019) with added extra features related to the structural terminologies of medical concepts in UMLS. They used a hierarchical medical ontology to connect similar concepts. The model were evaluated in the same data set, in addition to i2b2 (Sun et al., 2013) hand labeled one. The best accuracy achieved was 84.1% with augmented data methods using the global feature and hierarchical medical ontology on 67 abbreviations from the UMN data set. The work was tested on two forms of i2b2 data sets: manually labeled i2b2 and reverse substitution i2b2. The accuracy achieved for both data set was 85.9% and 88.9%, respectively.

As traditional machine learning approaches, deep neural networks could be used in an unsupervised way, where the labeled data is not required. The most advanced approach, which was applied on clinical abbreviation domain is Autoencoder architecture (Baldi, 2012). (Adams et al., 2020) generated a deep contextualized representation from the MIMIC III repository, taking into account the header section as a feature that could enhance the disambiguation process and give a good clue about the proper sense. Latent Meaning Cells (LCM) is a deep probabilistic neural network model used in this work to generate the feature vectors. This work aimed to learn the distributional properties space by feeding abbreviation, context and metadata, then inference if the candidate expansion belongs to this space or not. The system was tested in three different data sets on 51 abbreviations from UMN data set. The accuracy achieved was 71%.

2.5.3. Knowledge-based Approaches

Knowledge-based (KB) is another type of WSD classification task approach that relies primarily on dictionaries, thesauri and linguistic knowledge bases without using any cor-

pus evidence. Rule-based is one of these KB approaches that were applied to abbreviation disambiguation by applying regular expressions that represent lexical patterns containing terms and context represented by concepts (such as diseases, symptoms, etc.).

The winner in BARR2 shared task (León, 2018) implemented a set of 30 templates extracted from 500 Spanish clinical cases and 130 rules that model SF occurring in different parts of a clinical case. The system obtained 82.89% of F1 in BARR2 WSD sub-task 2 by combining templates with a n-gram frequency based approach that compares the content word list for each LF for a given SF to the frequency profile for the clinical case text (the best scored LF is selected). Semantic-based systems use lexical resources to map the ambiguous abbreviation to the most feasible definition. Consequently, KB approaches are useful to process languages with available resources, such as the case of English UMLS that incorporates abbreviations lists.

2.5.4. Other Approaches for WSD

Binary Spatter Code

Binary Spatter Code (BSC) (Kanerva, 1996) approach belongs to the distributional models such as Latent Semantic Analysis (LSA) (Deerwester et al., 1990), which is a high dimensional vector represents the meaning of the ambiguous word and its surrounding context. Furthermore, BSC could represent the relation between concepts. (Moon et al., 2013) evaluated a variant of Binary Spatter Code Word Sense Disambiguation (BSC-WSD) algorithm on 50 ambiguous abbreviations from clinical text. BSC-WSD uses reversible vector transformations to encode ambiguous terms and their context-specific senses into vectors representing surrounding terms. The variant took into account the direction and distance of the context words with respect to the ambiguous term. This approach achieved an average accuracy of 94.55% and outperformed SVM and NB classifiers.

Additionally, unlike other supervised learning models, with BSC-WSD researchers did not have to train separate models for each acronym. A single BSC-WSD model resulted in an average accuracy of 93.91%, which is not commonly seen with other machine learning models requiring individual models for each acronym. The Precision of this WSD model were improved from 0.792 to 0.875.

Vector Space Model

Vector Space Model (VSM) (Lowe, 2001) represents texts based on frequency counting, then generating the raw frequency vectors taking into consideration reducing the dimen-

sionality. In the end, the similarity between the generated vectors will be computed.

(H. Xu et al., 2012) used two types of data set. A collection of dictated discharge summaries from NYPH during the years of 2003 and 2004, which included 38,273 notes in total. The second was a corpus consisted of physician-typed hospital admission notes from NYPH during 2004-2006, amounting to 16,949 notes. (H. Xu et al., 2012) applied the reverse substitution method to increase the training data. Then, transformation procedures were applied to the created corpora to normalize it. The feature vectors was generated by using three types of features: stemmed words, positional information within a window size of 5 of the target abbreviation, in addition to section header of the admission note where the abbreviation occurs. These features were vectorized using TF-IDF and combined with sense frequency as a feature which was generated from a previous study (Inadomi, 2011). Separated VSM were implemented for 13 abbreviations, the average accuracy achieved was 79.2%.

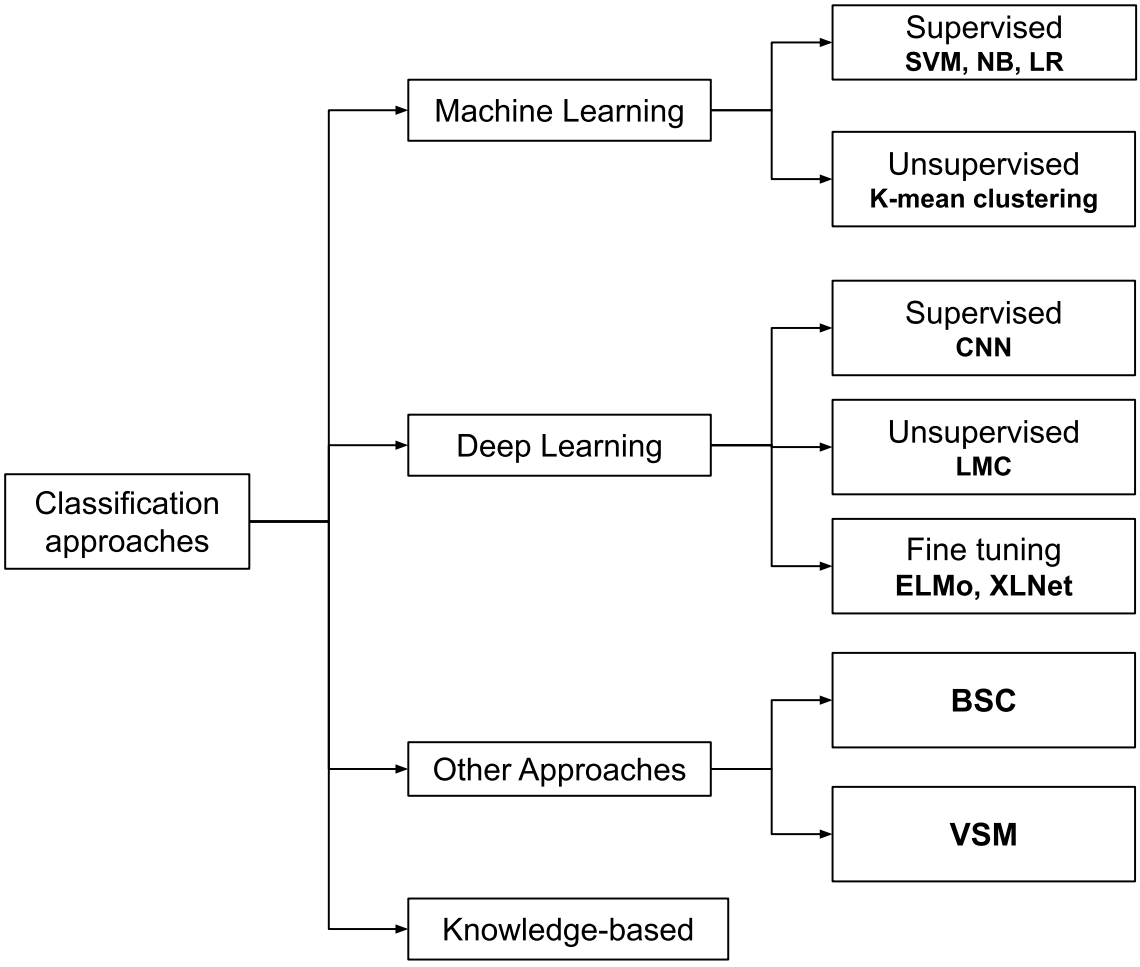


Figure 2.15: A summary of classification approaches that were applied on clinical abbreviation disambiguation.

Table 2.3 summarizes the different approaches which were tested on the UMN data set. Supervised approaches, traditional machine learning algorithms and deep neural networks

are the most applied. Also, it is noticeable that the different number of disambiguated abbreviations used to test each model number ranges between 8 to 75 abbreviations.

Figure 2.15 illustrates the approaches that were applied in clinical abbreviation domain. It is noticeable that supervised approaches are the most applied one, even there is a restricted number on annotated abbreviations, but researchers have attempted to overcome this bottleneck by following one of the simplest methods of data augmentation which is reverse substitution.

TABLE 2.3: Summary of approaches that were tested on the UMN data set.

Supervised Machine Learning				
Work	**no. of abb.**	**Features**	**Algorithms**	**Acc**
(S. Pakhomov et al., 2005)	8	BoW	Max Entropy DT	95.8% 93.9%
(Joshi et al., 2006)	16	POS, N-gram	NB, DT, SVM	>90%
(Moon et al., 2012)	50	BoW, POS, CUI Relative position	NB, DT, SVM	>90%
(Y. Wu, Xu, et al., 2015)	74	GloVe	SVM	95.79%
(Finley et al., 2016)	75	BoW	SVM, LR,NB	>90%
(Kashyap et al., 2020)	52	Ngrams	LR	*
Supervised Deep Learning				
(Joopudi et al., 2018)	50	WE , POS, section information	CNN	95.14%
(Skreta et al., 2019)	65	FastText	CNN	76%
Fine Tuning				
(Li et al., 2019)	30	topic header	ElMo	74.76%
(Kim et al., 2020)	75	hide the length of the expansion	XLNet	98.34%
Unsupervised Machine Learning				
(M. Peng and Quan, 2020)	74	BioELMo	K clustering	95.4%
Unsupervised Deep Learning				
(Adams et al., 2020)	51	Bayesian Skip gram	LMC	71.0%
Other approaches				
(H. Xu et al., 2012)	13	TF-IDF	VSM	*
(Moon et al., 2013)	50	BoW	BSC-WSD	94.55%

2.6. Discussion and Challenges

In this section we will highlight the main contribution in clinical abbreviations disambiguation as a WSD task and outline the challenges of this task.

It is evident from the previous sections that the disambiguation of clinical abbreviations task has a lack of resources. Just a publicly annotated clinical corpus with just 75 abbreviations, the UMN corpus contains 500 samples for each abbreviation. They are strongly unbalanced. 68 abbreviations have a sense accounting for over 50% of instances and 35 abbreviations have a dominant sense for more than 80% of instances. Furthermore, some samples are annotated with **"unsure,"** which could be considered without labeling, implying the wrong annotation, for example, under the term **"LE"**, we found samples where the accurate term label should be **"LV"**.

On the other hand, even though unannotated clinical data is available to the researchers, MIMIC III contains a considerable number of clinical texts. Manual annotation is a time consuming task that requires an expert in the domain to annotate. Most of the previous related work tried to avoid this step and increase their annotated data by applying one of the simplest data augmentation methods, reverse substitution. Researchers have used data augmentation to achieve many goals, increasing the existing annotated data training samples using several external resources, such as WWW, MEDLINE abstracts, private clinical notes MIMIC III (S. Pakhomov et al., 2005, (Finley et al., 2016). Another goal was to gain more pairs of abbreviations, expansions with more training samples, by leverage public senses inventories like UMLS and ALLAcronyms (Joopudi et al., 2018).

A problematic issue is that the reverse substitution method failed to figure out is rare and unseen abbreviations. Since the abbreviation is rarely used, increasing the training data set for them does not work with the reverse substitution. A solution is researchers utilize UMLS ontology to extract training samples not only for a specific sense but also for the most similar to this sense based on Euclidean distance (Skreta et al., 2019). For example, the expansion of **“intravenous fluid”** and **“in vitro fertilization”** are alternative expansions for the abbreviation **“IVF.”** Moreover, different versions for the same abbreviations or expansions used in clinical notes, such as **"gestation age / gestation"** and **"gestational / gestational ages / gestational age"**, could all be the same expansion (Kashyap et al., 2020).

The researchers also tried to overcome the limited data bottleneck by leveraging public biomedical data to increase the training data set. The previous related work proved that the clinical abbreviations have a characteristic that differs from other domains. UMLS has previously been proven to have inadequate coverage of abbreviations and acronyms. A study conducted that UMLS only covered about 35% of the abbreviations and acronyms that the authors investigated in the clinical domain (H. Xu et al., 2007). Also, (H. Liu et al., 2001) showed that the UMLS covered 66% of the studied abbreviations and acronyms with less than six characters. Further research has investigated 1269 biomedical abbreviations in clinical notes and they found that 727 (57.29%) are used on the clinical notes (Skreta et al., 2019).

Numerous features were used in addition to focusing on extending the training samples, for instance, traditional linguistic features, such as stemming, POS tags, relative positions (Joshi et al., 2006, Moon et al., 2012). The meaning of a word could be expressed with the rise of word embedding. Static word embedding improved the performance of disambiguating models when they have used features for supervised approaches (Y. Wu, Xu, et al., 2015). However, contextualized embeddings were used for generated feature vectors for targeted abbreviations (M. Peng and Quan, 2020) or fine-tuning models, which means adjusting the weights of pre-trained models to execute extra training steps before applying the classification step (Kim et al., 2020, Li et al., 2019).

Implementing a separated classifier for each abbreviation makes it challenging to detect any new or unseen expansions. Increasing the annotated training data for each expansion is sometimes hard to reach because of rare expansions. One-fits-all classifier could be a good solution to detect the unseen expansions.

Concerning Spanish language, unfortunately, the lack of resources represents an obstacle starting clinical WSD. Even though the BARR2 corpus contains 87 abbreviations with more than two expansions, the number of training examples for each abbreviation is too small to implement any classification approach. Furthermore, pre-trained word embedding in both biomedical and clinical data still does not exist.

There is still room for improvement, implementing a separated classifier for each abbreviation is an unpractical solution considering that abbreviations are highly used clinical narrative. It is not always possible to increase the training data, mainly if the abbreviation is rarely used. Furthermore, due to the productive nature of human languages, new abbreviations are continuously arising. Hence, a model is needed to predict these unseen abbreviations. The following chapters will illustrate how our proposed model improves these issues.

Chapter 3

METHODS

In this chapter, we describe in detail the three WSD approaches applied in this work. In the first experiment (Jaber and Martínez, 2021a), we implemented two separated supervised machine learning classifiers to disambiguate 13 clinical abbreviations from the UMN data set exploring two static pre-trained word embeddings to test different forms of features vectors. In addition we study the effect of each data set distribution on the accuracy of the systems.

In our next step, we improved the first experiment methodology to disambiguate scientific acronyms (Jaber and Martínez, 2021b) by implementing three separated supervised machine learning classifier, in addition to KB approaches to disambiguate acronyms with few annotated examples.

One final experiment (Jaber and Martínez, 2022) was conducted to disambiguate 75 clinical abbreviation from the UMN data set by fine tuning transformer-based architectures with a unified classifier for all abbreviations in the UMN data set in order to improve prediction of the rare and unseen abbreviations.

In the following sections, a complete description of each experiment including the data set, description of feature vectors generations and the proposed architecture.

3.1. Separated classifiers on a set of clinical abbreviations

Supervised machine learning classification approaches learn a classifier from a set of annotated examples. First, we have to collect the data, thus, we have obtained 13 abbreviations from publicly annotated clinical notes. Second, data preparation and preprocessing are applied to clean the data from any noise. Third, feature engineering step which is divided into selection and transformation, determines the important features to include in the learning phase then transforming them into numeric values to deal with. In this subsections, we will explain this process related to our experiment.

TABLE 3.1: List of 13 abbreviations from the UMN data set that were used in this study with their senses.

Abb	Sent.	Tokens	Senses	No.	(%)
AMA	2,881	37,887	against medical advice	444	88.8
			advanced maternal age	31	6.2
			antimitochondrial antibody	25	5.0
ASA	6,117	37,047	acetylsalicylic acid	404	80.41
			American Society of Anesthesiologists	93	18.98
			aminosalicylic acid	3	0.61
BAL	3,267	38,483	bronchoalveolar lavage	457	91.4
			blood alcohol level	43	8.6
BK	3,721	37,687	BK (virus)	343	68.35
			below knee	157	31.65
C3	3,270	39,901	cervical (level) 3	249	49.8
			(complement) component 3	243	48.6
			propionylcarnitine	6	1.2
			(stage) C3	2	0.4
CVA	5,212	36,616	cerebrovascular accident	278	55.6
			costovertebral angle	222	44.4
CVP	3,919	37,573	central venous pressure	436	87.2
			cyclophosphamide, vincristine, prednisone	62	12.4
			cardiovascular pulmonary	2	0.4
CVS	2,224	36,722	chorionic villus sampling	457	91.4
			cardiovascular system	41	8.2
			customer, value, service	2	0.4
ER	3,199	37,013	emergency room	448	89.52
			extended release	34	6.85
			estrogen receptor	18	3.63
FISH	3,129	39,248	fluorescent in situ hybridization	449	89.8
			GENERAL ENGLISH TERM	51	10.2
NAD	6,417	41,364	no acute distress	377	75.30
			nothing abnormal detected	123	24.70
OTC	6,173	37,356	over the counter	469	93.8
			ornithine transcarbamoylase	31	6.2
SBP	3,867	38,000	spontaneous bacterial peritonitis	417	83.4
			systolic blood pressure	83	16.6

3.1.1. Data set collection

Related to clinical abbreviations, there is just one existing corpus for 75 clinical abbreviations. A subset of a publicly annotated clinical notes data set from the University of Minnesota-affiliated (UMN) Fairview Health Services in the Twin Cities (Moon et al., 2012) was used in the this approach. The whole data was gathered from admission notes, inpatient consult notes, operation notes and discharge summaries.

A partial data set of 13 abbreviations was chosen for this experiment, totaling 6,588 annotated examples, (summarized in Table 3.1); 88 cases were removed due to annotation problems. The first column displays the abbreviations. The second and the third columns provide the number of sentences and tokens per abbreviation in the data set. The abbreviations senses in this data set are listed in the fourth column. The number of occurrences per sense and the frequency percentage for each sense was displayed in the fifth and sixth columns, respectively. There are 33 different senses in the data set, with an average of 2.5 senses per abbreviation.

3.1.2. Features Vector

As mentioned previously in chapter 2, extracting features represents an essential step in any machine learning algorithm. Two types of feature extraction and text representation were used in our approach; as a baseline, we extracted traditional linguistic features which are used for several NLP tasks, especially WSD tasks (Y. Wu, Denny, Rosenbloom, et al., 2015). Then, we explored several strategies of aggregating static pre-trained word embedding that are publicly available to create feature vectors.

Linguistic Features

Diverse features were used to disambiguate clinical abbreviations respecting both left and right words of the target abbreviation. In this approach, we extracted a set of linguistic features that have been successfully used in WSD (Y. Wu, Denny, Rosenbloom, et al., 2015). These features will be illustrated using the following sentence as an example : " ...**Last time she was discharged AMA and since she** ...".

1. **Word Features**- stemmed words within a window size 5 for each side of the target abbreviation.

   Example: {last, time, she, wa, discharg, and, sinc, she }.

2. **Word features with direction**- The relative direction (left or right side) of stemmed words.

   Example: {l_last, l_time, l_she, l_wa, l_discharg, r_and, r_sinc, r_she }.

3. **Position features**- The distance between the feature word and the target abbreviation.

   Example: {l5_last, l4_time, l3_she, l2_wa, l1_discharg, r1_and , r2_sinc, r3_she }.

TABLE 3.2: Pre-trained models specifications.

Details	Model 1	Model 2
Language	English	English
Resource	PMC	PMC, PubMed
Documents	672,589	22,792,858
Sentences	105,194,341	229,810,015
Tokens	2,591,137,744	5,487,486,225
Vector size	200	200
Algorithms	Skip gram	Skip gram

4. **Word formation features** from the abbreviation itself including special characters, capital letters and numbers.

Word Embedding Features

Two pre-trained models which were trained with Word2vec using SG with a window size 5 to create 200-dimensional vectors (Pyysalo et al., 2013) were used in this study. Both models were trained on unlabelled biomedical data resources. However, one of them (model 2) was trained by using extra biomedical resources, which is PubMed (nearly 23 million abstracts from MEDLINE). In addition to four million English Wikipedia articles (see Table 3.2). Four different strategies of combining embeddings were tested to generate the feature vectors for each training sample on the data set.

As indicated in the equation 3.1, the sum of the embedding row vector of surrounding words for the abbreviation within window size 5 was calculated for each annotated example.

$$SUM\_WE(w) = \sum_{i=j-5}^{j+5} Emb(S(i)) \tag{3.1}$$

Where $\boldsymbol{w}$ is the target abbreviation to disambiguate, $\boldsymbol{j}$ is the index of $\boldsymbol{w}$, $\boldsymbol{S}$ is the sentence containing $\boldsymbol{w}$ and $\boldsymbol{S(i)}$ is the word indexed by position $\boldsymbol{i}$ in sentence $\boldsymbol{S}$.

Second and third strategies were computed by taking the maximum and the minimum value for each embedding dimension for the surrounding words, as shown in the following equation 3.2 and equation 3.3 respectively.

$$MAX\_WE(w)_j = MAX\{Emb_j(S(i))\} \tag{3.2}$$

$$MIN\_WE(w)_j = MIN\{Emb_j(S(i))\} \tag{3.3}$$

The last strategy was generated by computing the average for the word embedding vectors surrounding the abbreviation, as shown in equation 3.4

$$AVG\_WE(w) = \sum_{i=j-5}^{j+5} \frac{Emb(S(i))}{2W} \tag{3.4}$$

3.1.3. Supervised Machine Learning Algorithms

In this approach, two supervised machine learning algorithms were implemented: Support Vector Machine (SVM) and Naïve Bayes (NB) algorithms. We will briefly describe both of them in the following subsections.

Support Vector Machine Algorithm

This method (developed by (Boser et al., 1992)) aims to learn a linear hyper-plane from the training set that distinguishes positive and negative examples. The hyper-plane is in hyperspace where the distance between the closest positive and negative examples is maximized (called support vectors). In other words, Support Vector Machine (SVM) tries to decrease the empirical classification error while also increasing the geometric margin between positive and negative cases.

The geometric intuition is illustrated in Figure 3.1. The bold line denotes the plane that separates the two classes of cases, while the two dotted lines denote the plane tangential to the closest positive and negative examples. A weight vector $\boldsymbol{w}$ perpendicular to the hyper-plane (which accounts for the training set and whose components represent features) and $\boldsymbol{a}$ bias $\boldsymbol{b}$ that defines the hyper-offset plane's from the origin make up the linear classifier. If $\boldsymbol{f(x) = w.x + b \geqslant 0}$ for an unlabeled example x, it is considered as positive (negative otherwise). It is possible that the hyper-plane will not be able to split space linearly.

SVM is a binary classifier so that, it must be converted to multi-class classification before being used for WSD (where the senses of a target word represent the number of the classes). For example, reducing the multi-class classification problem to a series of binary classifications of the kind sense $\boldsymbol{S_i}$ vs all other senses is an easy option. As a logical consequence, the sense with the highest amount of confidence is adopted.

It can be proven that the SVM classification formula may be simplified to a function of the support vectors, which determines the dot product of pairs of vectors in its linear

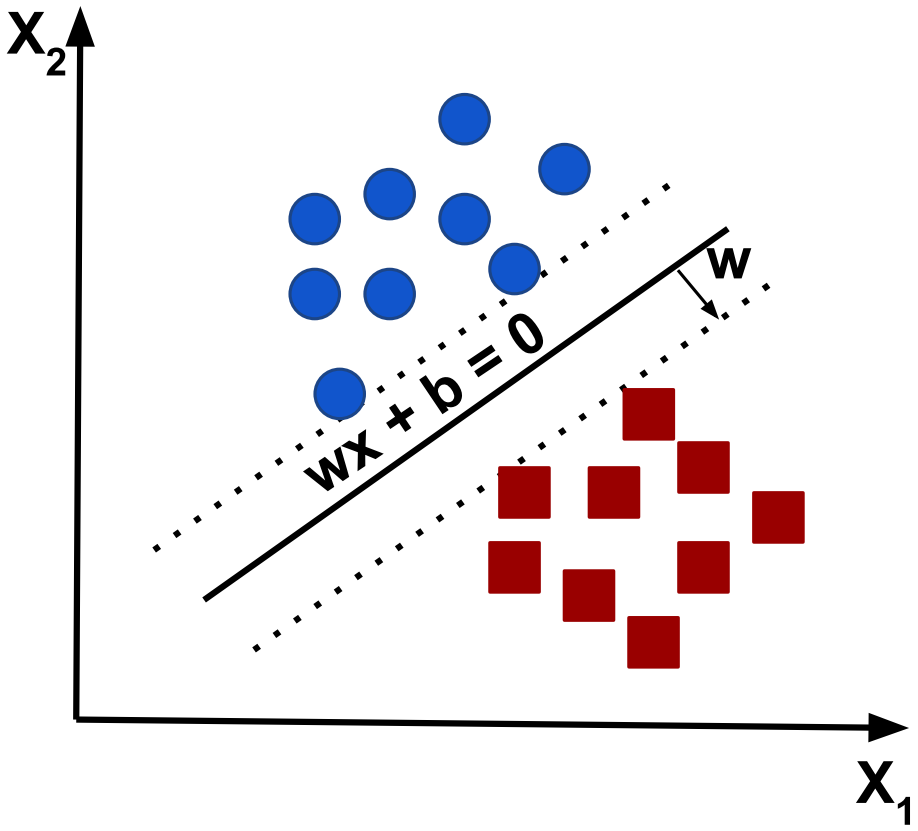


Figure 3.1: Support Vector Machine Representation.

form. Generally, a kernel function is used to calculate the similarity between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, which maps the original space into a feature space like $\boldsymbol{k(x,y) = \Phi(x).\Phi(y)}$, where $\boldsymbol{\Phi}$ could be in its simplest form $\boldsymbol{k(x,y) = (x.y)}$. One of SVM important success elements is its ability to map vector spaces to higher dimensions using kernel approaches, as well as its high degree of adaptability based on parameter tuning.

Naive Bayes Algorithm

The Naïve Bayes (NB) classifier is a probabilistic method for estimating probabilistic parameters that has a significant role to play in WSD. For each sense of an abbreviation, the conditional probability is computed using the Bayes theory for which a set of features is defined $\boldsymbol{(x_1, x_2, .., x_m)}$. Let $\boldsymbol{P(sense)}$ and $\boldsymbol{P(x_i|sense)}$ are the probabilistic parameters of the model and they can be estimated from the training set using relative frequency counts

(equation 3.5).

$$
\begin{aligned}
argmax P(sense|x_1, ..., x_m) &= argmax \frac{P(x_1, ..., x_m|sense)P(sense)}{P(x_1, ..., x_m)} \\
&= argmax P(sense) \prod_{i=1}^{m} P(x_i|sense)
\end{aligned}
\tag{3.5}
$$

Where ***m*** represents the number of features, thus, equation 3.5 is constructed by the naive assumption which refers that features are conditionally independent given the sense (the denominator is also discarded as it does not influence the calculations).

3.1.4. Proposed supervised model architecture

Figure 3.2 depicts a high-level overview of the supervised machine learning approach. There are three phases: data set preparation, which includes various pre-processing procedures to prepare and clean data. Second, training a machine learning model for classification utilizing various features and testing the model on the test data set. The following subsections will describe the several techniques involved in the disambiguation of the clinical abbreviations used for integrating pre-trained models over the two supervised machine learning algorithms SVM and NB classifiers.

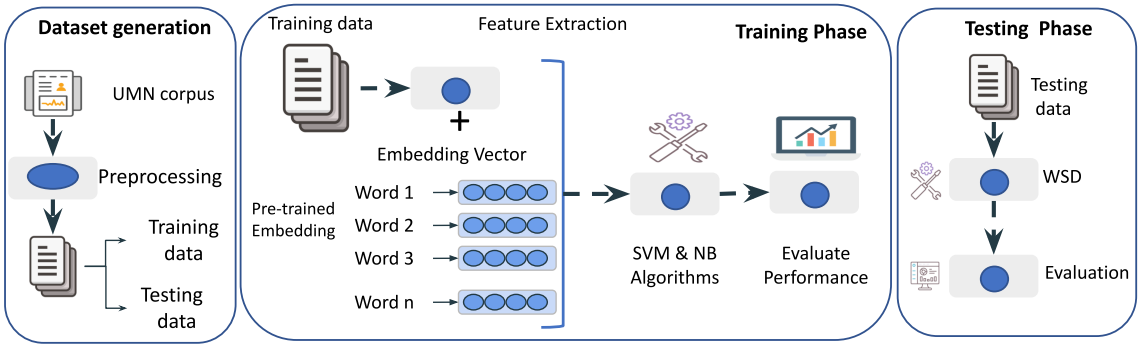


Figure 3.2: Overview of supervised approach to disambiguate clinical abbreviations. Training and testing phases are repeated for each abbreviation in the data set.

3.1.5. Experiment Specification

The experiment was implemented in Python programming language via Google Colab environment. For machine learning algorithms implementations scikit-learn library (Pedregosa et al., 2011) was used. GaussianNB algorithm was used for implementing NB. SVM was implemented with hyper-parameter c=1 and linear kernel. The code is available

in GitHub repository:
https://github.com/AreejJaber18/Supervised_WSD

3.2. Hybrid approach for disambiguation rare abbreviations

Moving forward toward our goals, coping with the problem of reaching the minimum required number to apply the supervised machine learning algorithms, a hybrid approach was proposed to disambiguate scientific acronyms using a data set with a variety of acronyms with different number of annotation examples.

The idea behind this experiment was that the classification approach for each acronym would be decided based on the number of its available annotated examples in the data set. Thus, if the acronym has more that 20 annotated examples, a separated supervised machine learning classifier was implemented to disambiguate it. However, if the acronym has less than 20 annotated examples, a knowledge based approach (cosine similarity) was implemented to disambiguate this acronym.

The SDU@AAAI-21 lunched a shared task for disambiguate scientific acronyms as part of Scientific Document Understanding workshop in 2021. The purpose of this task was to predict the correct meaning of an ambiguous acronym in a given sentence. The system was fed a sentence containing an ambiguous acronym and a dictionary containing possible expansions (i.e., long-forms) of this acronym. In the following, we will describe the data set, existing approaches and our proposed model which was applied to this data set.

3.2.1. Data Set collection

This task was created by SDU@AAAI-21 shared task organizers, providing the SCientific Acronyms Disambiguation (SCiAD) (Veyseh et al., 2020) corpus. SCiAD was built using 6,786 English papers from arXiv, totaling 2,031,592 sentences. Table 3.3 displays the total number of annotated samples on three data sets: training, development and test data set. The data set was provided in two phases: the developing phase to design and develop the model by training/development data set. Then the generated models are tested with test data set.

The provided training and development data sets contain 731 and 611 acronyms, respectively. The number of annotated examples differ between acronyms. 299 acronyms from the training data set have less than 20 examples. Figure 3.3 illustrates these frequencies among all the acronyms on the data set.

TABLE 3.3: Description of training, development and test data sets.

Number of.	Training	Development	Test
Sentences	50,034	6,189	6,218
Tokens	1,548,278	190,654	190,111
Acronyms	731	611	618
Expansions	2,150	1,233	-

The organizers also provided an acronyms dictionary as a sense inventory. There are 732 acronyms and 2,308 senses in the dictionary, with an average of 3.15 senses per acronym. Figure 3.4 depicts the distribution of senses for acronyms in the dictionary. Moreover, appendix C illustrates the distribution of the training examples among the senses for each acronyms for the top 10 of these frequencies.

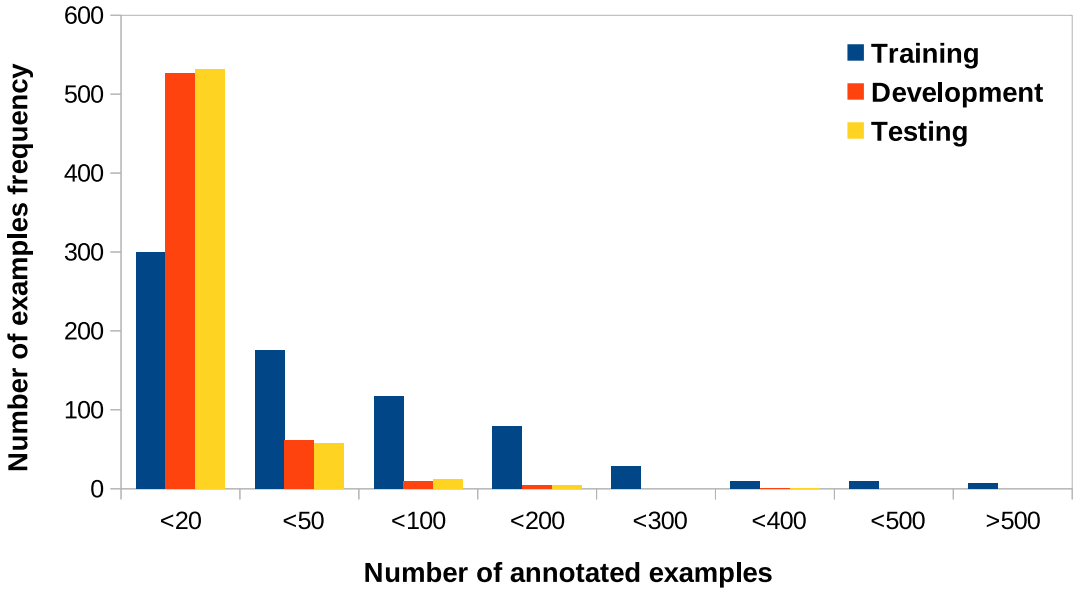


Figure 3.3: Frequency of each number of examples per acronym across train, development and test data sets.

3.2.2. Feature Vectors

Two types of features were used. The same Linguistic Features, which used in the work introduced in section 3.1.2, were extracted from the context. There features are: stemmed word, POS tags, position and orthographic features. The second feature type was static pre-trained word embeddings, which were generated using FastText word embedding model (Joulin et al., 2017). The embeddings were created from several English resources such as Wikipedia and data from the typical crawl project (Mikolov et al., 2018) with vector dimension 300. Before extracting the features, the data set was exposed to sev-

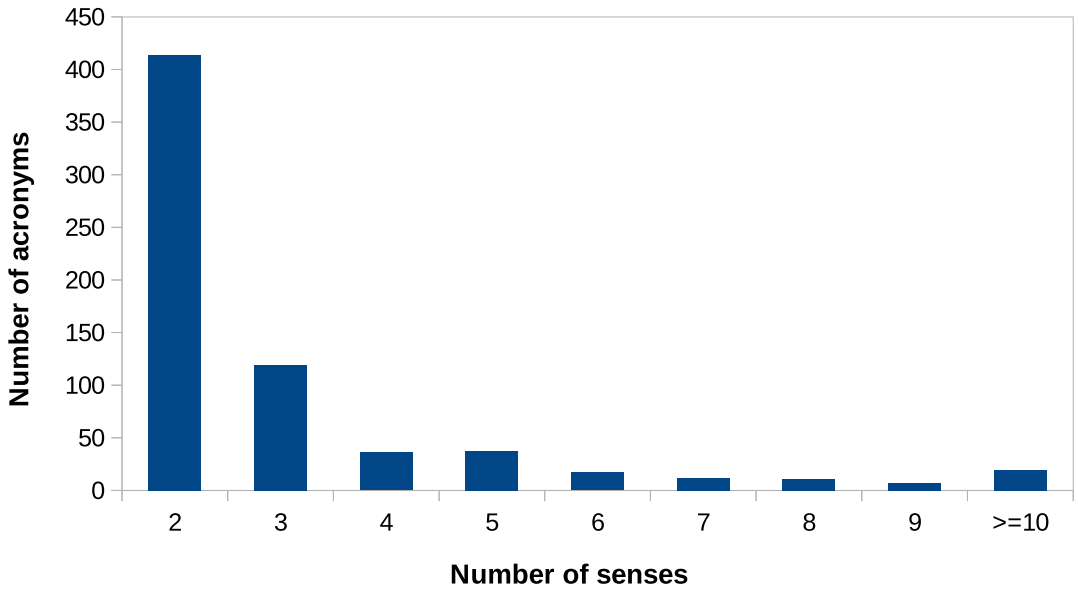


Figure 3.4: Number of senses per acronym in the dictionary. E.g. we see that there are 437 acronyms with two expansions.

eral pre-processing steps, including the removal of stop words, special characters and word stemming. For supervised machine learning approaches, features were formed by combining WSD lexical features and the summation strategy from the pre-trained word embeddings which were generated based on the following equation:

$$S = \sum_{i=0}^{|W|} v(W(i)), i \neq k \tag{3.6}$$

Where $\boldsymbol{W}$ is a list of words which surrounding the targeted acronym. $\boldsymbol{|W|}$ is the length of the list and $\boldsymbol{v}$ is a FastText pre-trained word embedding and $\boldsymbol{k}$ is the position of the target acronym.

For the KB approach, however, only the summation strategy of pre-trained word embedding vectors was generated for each annotated example and for the candidate expansions extracted from the acronyms dictionary.

3.2.3. Models Description

The organizers provided a rule-based baseline in the code directory to familiarize the participants with the task. This baseline computed the frequency of expansions in the training data set and then it chose the expansion with the highest frequency as the final prediction for each acronym in the development data set. The expansion that appeared first in the dictionary among all tied expansions was chosen as the final prediction if there was a cor-

relation. Our proposed model was a hybrid approach: a supervised approach which was applied on acronyms that have annotated examples of more than 20. Cosine similarity was applied to acronyms that have less than 20 annotated examples on training/development data set. In the following subsections, we will illustrate the supervised machine learning algorithms and the cosine similarity approach.

Supervised Machine Learning Algorithms

Three Supervised machine learning algorithms (SVM, NB, kNN) were implemented to disambiguate acronyms with more than 20 examples. Thus, a separated classifier of each acronym was implemented. SVM and NB were discussed in detail in the previous section (see section 3.2.2). How the kNN algorithm could be used as a classifier is explained below.

k-Nearest Neighbor Algorithm

kNN is a supervised machine learning algorithm which is considered one of the exemplar based classifier. The kNN algorithm presumes that similar things exist nearby and, in other words, similar things are close together.

For classification a new example $\boldsymbol{x} = (\boldsymbol{x_1}, ..., \boldsymbol{x_m})$. First, it should be represented as its feature vector. Then, the distance between this feature vector and **k** neighbor feature vectors which are stored previously is calculated based on different equations such as Euclidean distance (equation 3.7). The smallest distance will determine for which sense the context belongs.

Figure 3.5 depicts the classification procedure based on the kNN classifier. The triangle represents a new example that we want to decide for which class it belongs among the three existing ones. The distance between the new data vector will be calculated to the **k** nearest neighbor from the three classes and then, the data vector will be considered part of the class with smallest distance with it.

$$d(v_1, v_2) = \sqrt{\sum_{i=1}^{m} (v_1 - v_2)^2} \tag{3.7}$$

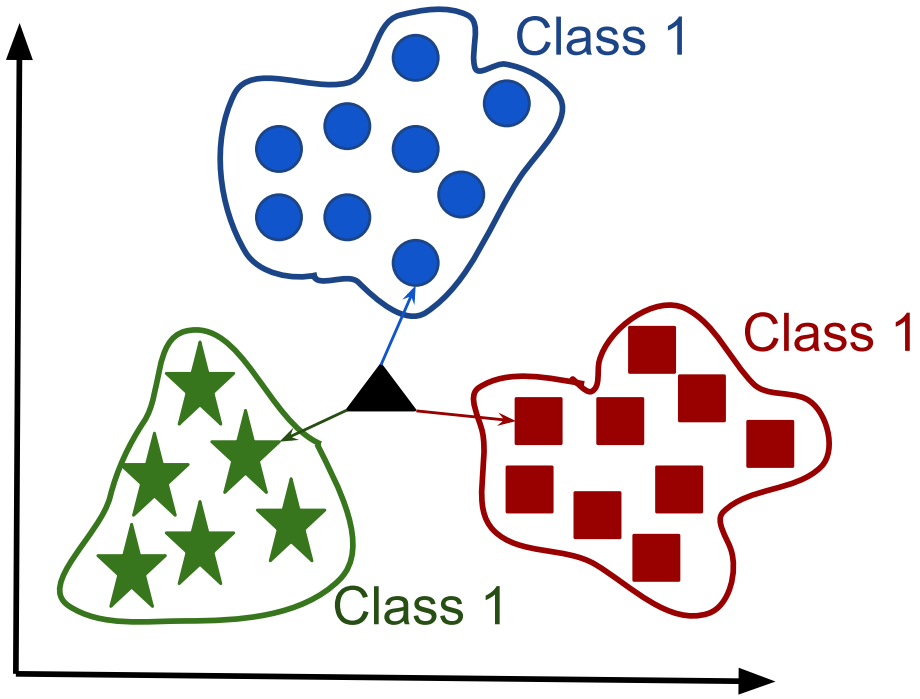


Figure 3.5: K Nearest Neighbor.

Cosine Similarity

The cosine similarity (Singhal, 2001) of two vectors was calculated by dividing their dot product by the product of their norms, as shown in equation 3.8. The result will be in range between 0 and 1. When the result was close to 1, two vectors are said to be similar and when it is close to 0, they are said to be dissimilar.

$$cosim(v_1, v_2) = \frac{v_1.v_2}{||v_1||||v_2||} \tag{3.8}$$

We hypothesized that the correct acronym expansion will have a high cosine similarity to its context (Figure 3.6). Before proceeding, all of the acronyms expansions' signatures and their context must be mapped to sentence vectors using the various configurations described in the preceding section.

3.2.4. The proposed hybrid approach

A hybrid approach was applied to the SCiAD data set. Figure 3.7 summarizes the overall process for the proposed system. First, preprocessing steps were performed on the data set. Then, the data set was separated based on the number of annotated examples for each acronym. Second, for each acronym that have less than 20 annotated examples, cosine similarity approach was applied to disambiguate it, otherwise, supervised machine

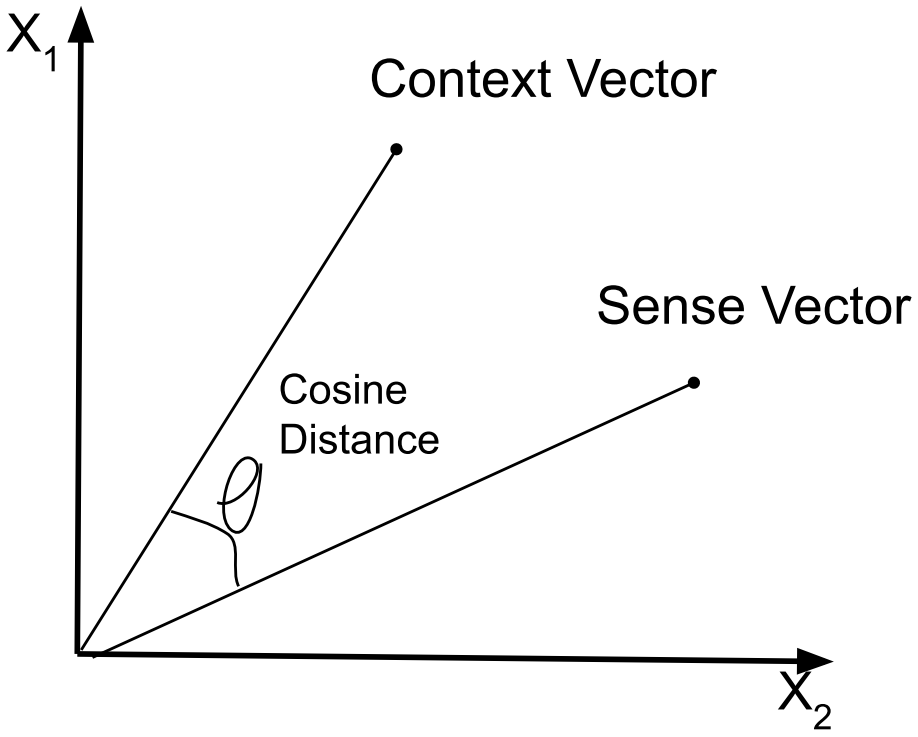


Figure 3.6: Cosine similarity.

learning algorithms were implemented to disambiguate the acronyms.

3.2.5. Experiment Specification

The experiment was implemented in Python programming language via Jupyter notebook. For machine learning algorithms implementations scikit-learn library was used, GaussianNB algorithm was used for implementing NB. SVM was implemented with hyperparameter c=1 and linear kernel. KNeighborsClassifier with n_neighbors = 3. The experiment was executed on a device with 16GB RAM and processor Intel Core i7-10510U CPU @ 1.80GHz × 8. The code is available in the GitHub repository:
https://github.com/AreejJaber18/AcronymsWSD

3.3. One-fits-all classifier for disambiguate unseen abbreviations

The aim of the transfer learning is to share knowledge from a related source task with the target task. This type of method compensates for the lack of adequate training data in the target task. With deep learning models, fine-tuning is currently the most frequently used approach for transfer learning. It begins with a pre-trained model on the source task and then trains it further on the target task. When we decide to fine-tune a model, we should consider the size of the new data and its similarity with the original data.

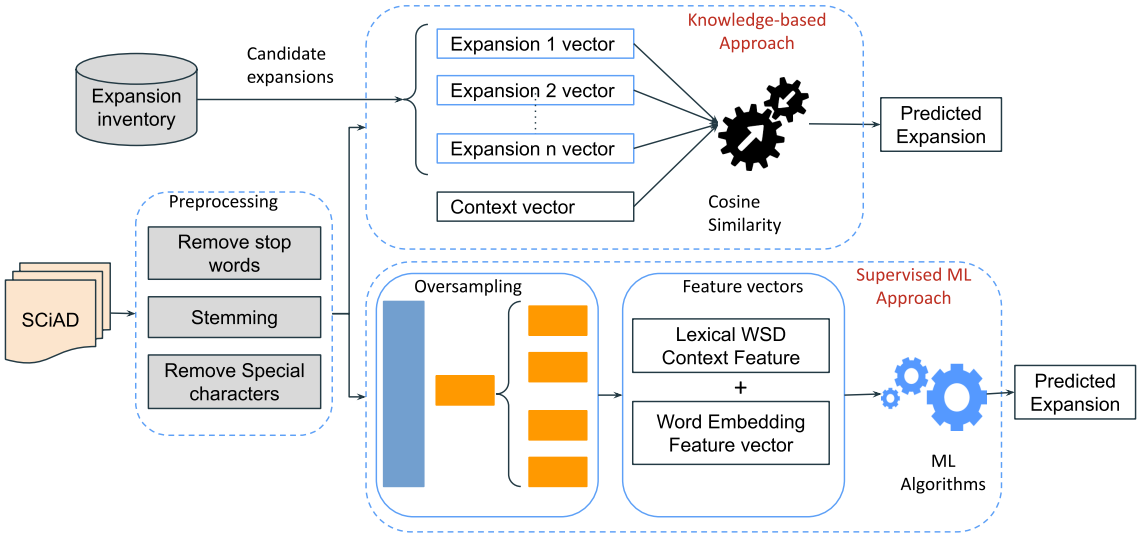


Figure 3.7: Overview of the proposed approach to disambiguate acronyms.

Fine-tuning could be performed in three ways as shown in Figure 3.8. If we have a large data set that differs completely from the original data set, **strategy (1)** is preferable to apply which retraining all the entire model. If we have a small data set and it is distinct from the pre-trained model data set, it is better to apply **strategy (2)** which is freezing part of the layers and training the rest part. However, if the data set is very similar to the pre-trained model data set, **strategy (3)** is the best option to fine-tuning the added layers.

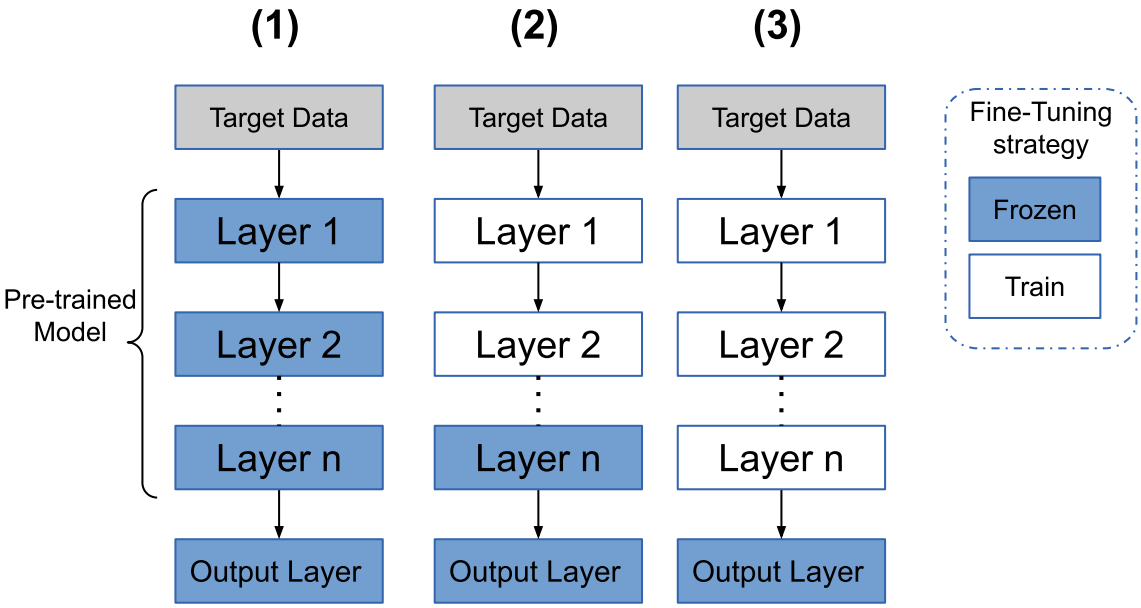


Figure 3.8: Fine tuning strategies on a pre-trained model.

Transformer-based models facilitate transfer learning on text data since processing input text is executed independently. Thus, in our third experiment, to compensate for the lack of annotated data in the clinical abbreviations task and leverage transferring knowledge to disambiguate rare and unseen abbreviations or expansions. We fine-tuned

a pre-trained transformer-based language modeling with one classifier for all the clinical abbreviations in the UMN data set.

In the following subsections, we will review the UMN data set that was used in this experiment, the transformer-based architectures and the proposed architecture.

3.3.1. Data Set collection

This experiment was applied to the whole UMN data set. The data set contains 75 abbreviations of the most frequent acronyms and abbreviations from clinical repository (Moon et al., 2012). Each abbreviation has 500 sentences that were annotated with different senses. The data set contains abbreviations about devices, places (US, DC) and names (LE, RT). Also, it has 219 examples belonging to different abbreviations that were annotated as **"UNSURED SENSE"** because the annotator could not identify the exact expansion for the target abbreviation. However, 319 examples were annotated as a **"GENERAL ENGLISH"**, which means that the abbreviation does not represent a clinical abbreviation. In general, there are 351 senses with an average of 4.7 senses per abbreviation (highly ambiguous abbreviations)(see appendix B).

3.3.2. Transformer-based architecture

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is an embedding layer representation or language model built utilizing transformer architecture (Vaswani et al., 2017) with an attention mechanism that learns contextual semantic relationships (or sub-words) in an unannotated text. The transformer consists of an encoder to read the input and a decoder to produce the prediction. BERT, as language modeling goals, depends on the encoder with two model sizes: BERT_{base} and BERT_{large}. The difference between them is the number of encoders that are used. BERT_{base} consists of stack of 12 encoders, on the other hand, BERT_{large} consists of stack of 24 encoders. Moreover, they differ in the number of the hidden units (768, 1,024) and the attention heads (12,16). Unlike directional models that read the input sequence from left to right or right to left, BERT reads the entire sequence of words in both directions simultaneously. The most valuable feature for BERT is that it allows the model to represent words based on their surrounding contexts.

Figure 3.9 depicts the architecture of the encoder which is used in BERT implementation. As shown, the input of the encoder is a vector of embedding vector with length 768, which is concatenated with another vector which is called positional encoding. Because feed-forward network architectures could not remember how the sequence is fed into the

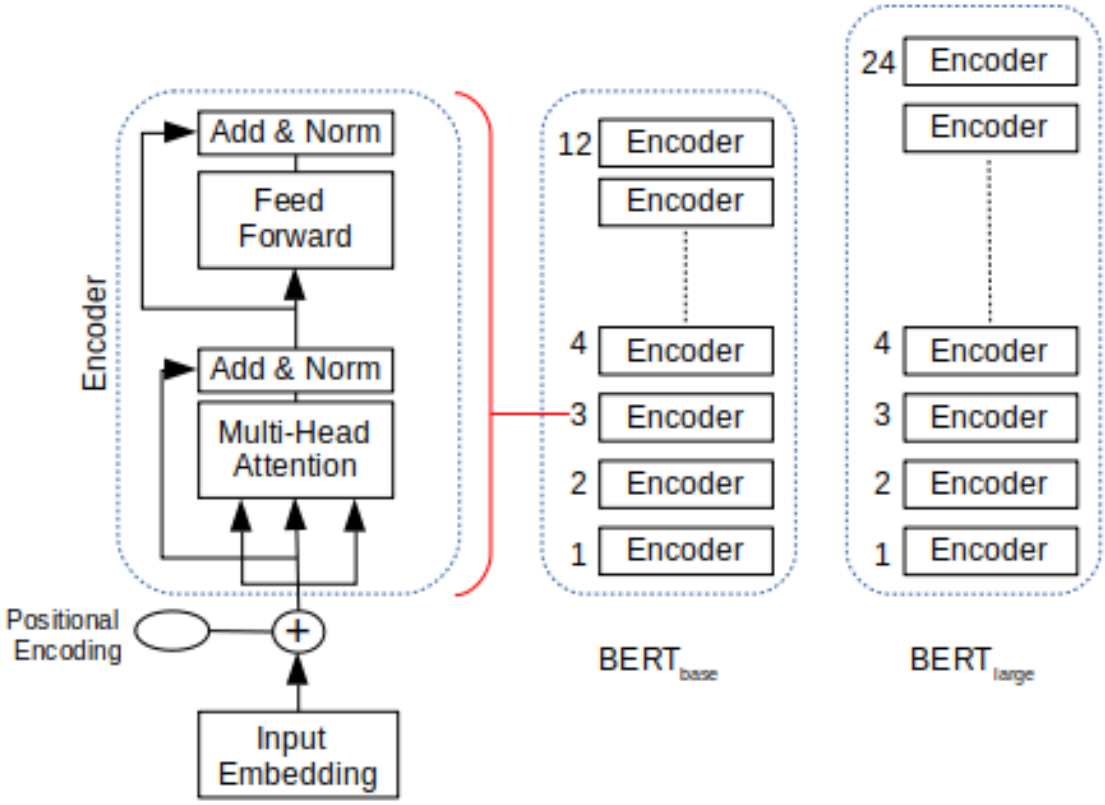


Figure 3.9: BERT based on Encoder Architecture.

model, they need to attach a relative position for each word since the order of the word sequence is essential in the representation. The position of each word is calculated based on it position order. if the word position is even, equation 3.9 will be executed. If the position is odd, equation 3.10 will be executed where $\boldsymbol{d_{model} = 512}$. At the end, the values will be stacked in a one-hot vector.

$$PE(pos, 2i) = sin\left(\frac{pos}{1000^{2i/d_{model}}}\right) \tag{3.9}$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{1000^{2i/d_{model}}}\right) \tag{3.10}$$

After that, for each token in the sequence, three vectors will be generated to calculate the attention vector based on the equation 3.11, where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ are queries vectors matrix that represent a token in a sequence. A multi-head attention layer runs through attention calculation several times to take into account all the possibilities of presentation of the sequence such as long-term, short-term dependencies. The different independent outputs of the attentions are concatenated, then linearly transformed as in equation 3.12, where $\boldsymbol{head_i}$ is calculated from the previous equation 3.11 and $\boldsymbol{W}$ is a learnable parameter

TABLE 3.4: The pre-trained models architecture is used in this study.

Characteristic	No.
layers	12
hidden units	768
self-attention heads	12
Total trainable parameters	110M

matrix.

$$Attention\,(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.11}$$

$$Multi-head\,(Q,K,V) = [head_1,...,head_h]W_0 \tag{3.12}$$

For this work, we fine-tuned three pre-trained BERT models with a one-fits-all classifier applied to the UMN data set. These three models were trained using BERT_{base} architecture that means they have the same model specifications as mentioned in Table 3.4. Moreover, the variance between them is the type and size of the resources used to generate the models. We will go over them in detail in the following:

- Clinical BioBERT (Alsentzer et al., 2019): This model was generated by fine-tuning BioBERT (Lee et al., 2020) model with 2 million clinical notes from the MIMIC III database. BioBERT was generated by fine-tuning BERT_{base} on a corpus of biomedical resources such as PMC full articles and PubMed abstracts. The model improved the performance over a variety of NLP biomedical tasks.

- Blue-BERT (Y. Peng et al., 2019): The model was a result of a set of experiments to fine-tuned both BERT architectures on PubMed abstracts and MIMIC III clinical notes. The best result was obtained by fine-tuning over a combination of 4K million words from PubMed abstracts and 500 million words from MIMIC III, which was adopted in our experiment.

- MS-BERT ("MS-BERT", n.d.): Blue-BERT was used as a starting point to generate the MS-BERT. MS-BERT was fine-tuned on a corpus of 35.7 million words from clinical notes neurological examination for Multiple Sclerosis (MS) patients at St. Michael's Hospital in Toronto, Canada.

3.3.3. Proposed BERT Fine-tuned Architecture

Compared with other language modeling models, BERT proposes two novel strategies for the training process as shown in Figure 3.10. Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The MLM training strategy aims to predict 15% of masked words in the sequence text input and these words are replaced with [MASK] tokens before starting the training process. On the other hand, the NSP training strategy aims to predict if a pair of sentences are related to each other or not. Thus, BERT can generate vectors for each word in a sequence text as well as their semantic meanings using these two strategies.

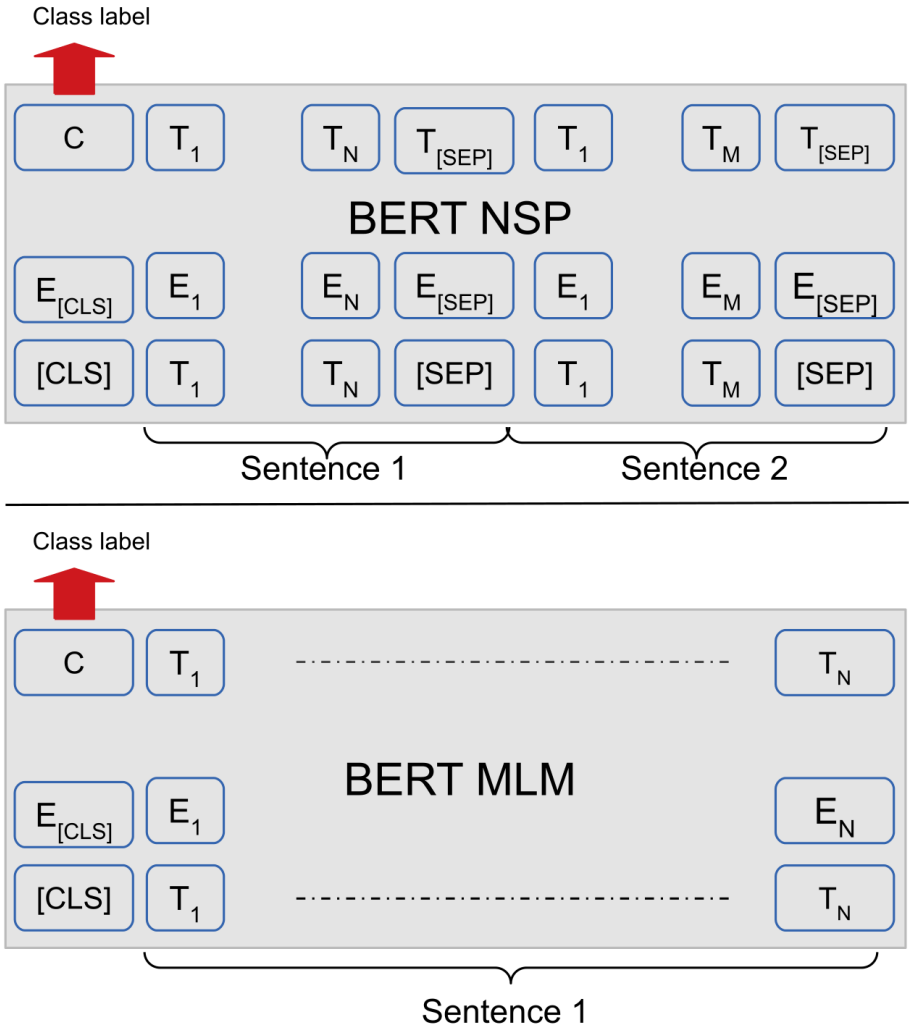


Figure 3.10: BERT training strategies.

For this experiment, we adopted a NSP training strategy that requires a appropriate input sequence format in order to obtain the corresponding embedding to each sentence, as illustrated in Figure 3.10. To achieve that, a set of steps were performed on each annotated example in the UMN data set. The pre-processing steps will be described below:

- Cleaning: Converting all the characters to lowercase in addition to eliminating the special characters and punctuation was applied on all input sequence using NLTK python library (Bird and Loper, 2004).

- Tokenization: BERT employs a tokenizer known as a Word-Piece tokenizer (Y. Wu et al., 2016). It works by dividing words into their full forms (e.g., one word becomes one token) or into word pieces (e.g., one word can be broken down into multiple tokens). For example, **"play"** word will be converted to **["play"]** token, but **"playing"** word will be tokenized to **["play","##ing"]**.

- The [CLS] and [SEP] tokens: The classification task requires a single vector representing for the whole input sequence. In BERT, the process is done by the hidden state of the first token in the sequence. So, [CLS] tokens should be added manually or by existing packages at the beginning of the sequence to achieve that goal. To inform the model that the input sequence is composed of two sentences, [SEP] tokens are used to determine the first and second sentences. As a result, two [SEP] tokens were inserted into the sequence, one at the end of the first sentence and the other at the end of the second.

- The [PAD] tokens and the attention mask: length of all input sequences should be the same so that we adjusted the length on 512 tokens; Hence, each sentence that has more than 512 tokens will be truncating, on the other hand, if the sentence has less than 512 tokens, [PAD] tokens will be added at the end of the sentence. Attention masks were used to make the model distinguish between the actual token and the [PAD] ones.

- Token IDs: after applying all the previous steps, each token should be mapped into its ID (also called segment IDs) that provided by the pre-trained model based on its set of vocabulary. If the token does not exist in the model, it will be replaced with [UNK] to be marked as an unknown word.

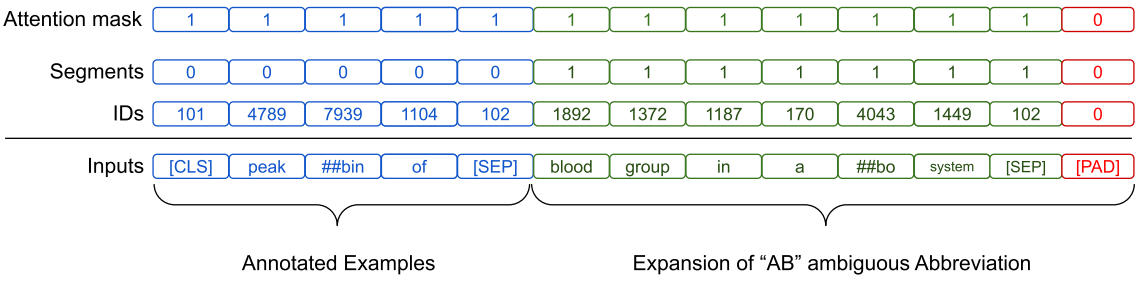


Figure 3.11: An example of input representation for one sequence including [CLS], [SEP] and [PAD] tokens, in addition to added segments, attention mask.

Figure 3.11 shows an example of the input sequence for the abbreviation **AB**. Here, the context is **"... received photo-therapy for a peak bilirubin level of 11.5 mg%.**

Her blood type was AB positive…" and the correct expansion is **blood group in ABO system**. [CLS] token was added at the beginning of the context, two [SEP] tokens were added to indicate two sequence classifications and the [PAD] tokens were added to unify the length of all sequences as mentioned above. Then the tokens were mapped to their IDs. The first segment was denoted by adding 0 and 1 for the second segment. Lastly, the attention mask layer marked the actual tokens with 1 and [PAD] tokens with 0.

Final proposed architecture

WSD systems that are fine-tuned directly adjust the pre-trained weights on annotated corpora rather than learning new weights from scratch. We fine tuned three separated pre-trained models: clinical BioBERT, Blue-BERT and MS_BERT. At the top of these pre-trained model we added a classifier which was composed from three layers, the feed-forward layer (Schmidhuber, 2015) with 512 nodes, activation ReLU (Agarap, 2018) and another feed forward layer with 348 nodes as the following equation:

$$P = L_2(ReLU(L_1(f))) \tag{3.13}$$

Where $\boldsymbol{L_i = W_i x + b_i}$ are fully-connected linear layers, $\boldsymbol{W_1 \in R^{HxH}}$,$\boldsymbol{W_2 \in R^{|S_{wp}|xH}}$.

One-fits-all-classifier was proposed instead of a separated classifier for each abbreviation in the data set. Parameters were updated during the training process by minimizing the cross-entropy loss between the true label y and the sense distribution **p**:

$$L = -\frac{1}{M}\sum_{M=1}^{M}\sum_{s=1}^{|S_m|}[y_m]_s \, log\,[p_m]_s \tag{3.14}$$

Where M is the number of examples in the data set and $\boldsymbol{y_m}$ is a one-hot vector which represents the true label of $\boldsymbol{w_m}$.

Figure 3.12 illustrates the sequence of the fine-tuning approach. Our main contribution was circled with a dotted blue triangle. The first step, as mentioned in the previous section, was to prepare the data to be fed to the models. Then the model weights were adjusted. Lastly, the classifier was added on the top of the model and trained on the UMN data set. These steps were applied to the three pre-trained models.

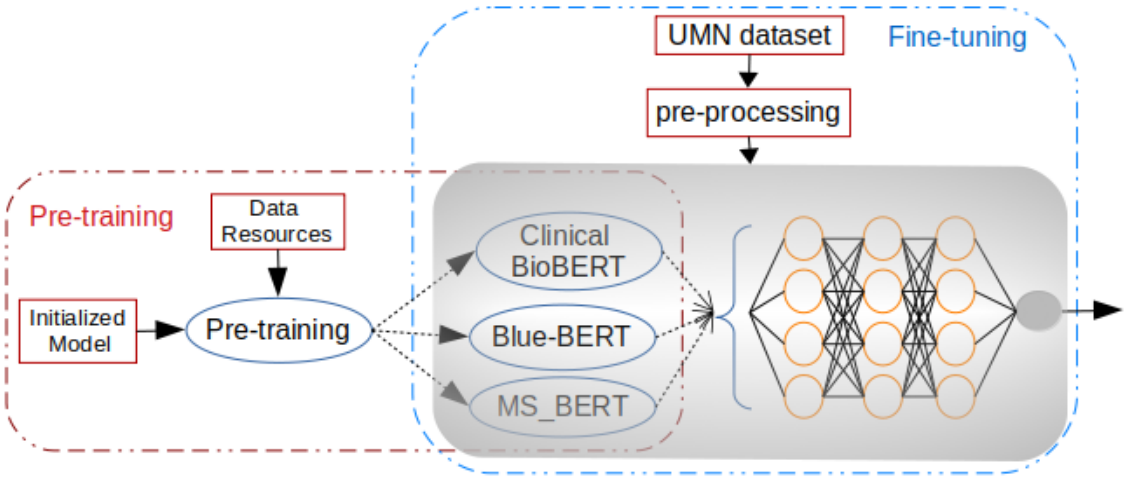


Figure 3.12: Proposed Fine-tuning model that was applied on three pre-trained models to disambiguate clinical abbreviations.

3.3.4. Experiment Specification

The experiment was implemented in Python programming language via Jupyter notebook. The hyper-parameters for all experiments were identically adjusted in the three pre-trained models. The batch size was 8 and 5 epochs for the models. The learning rate was $\mathbf{1x10^{-5}}$. Adam optimizer (Kingma and Ba, 2015) was used. Transformers and torch python library were used for accessing the the pre-trained models. The experiments were executed with Nvidia GeForce RTX 2080 Ti graphics card, an Intel(R) Core(TM) i5 CPU and 16GB of RAM at 1600 MHz. with 6 hours for each pre-trained model. The code is available in the GitHub repository: https://github.com/AreejJaber18/One-Fits-All-Classfier

3.4. Conclusion

To sum up, through the timeline of this study, three models were proposed to disambiguate abbreviations, with a specific goal for each of them. The first proposed model implemented two supervised machine learning algorithms, SVM and NB. 13 separated classifiers for 13 clinical abbreviations were implemented from the UMN data set. Two static pre-trained word embeddings were generated based on the word2vec model to create a feature vectors for the ambiguous abbreviations with context window size 5.

To improve the first model, rare abbreviations should be figured out to reflect the reality of clinical abbreviations nature. Since this annotation data is not available with clinical data, the second proposed model was applied to scientific acronyms SCiAD data set. The model combined two classification approaches, supervised machine learning and KB approaches. For each abbreviation with more than 20 annotated examples, super-

vised machine learning algorithms were implemented to build the disambiguation model. The idea was to divide the data set based on the size of each abbreviation in the data set. However, if the abbreviation data set size was less than 20 (represent clinical rare abbreviations), the feature vector for each pair of context and expansion were generated from the summation of the pre-trained word embedding vectors from the context with five windows size. The similarity was calculated using the cosine similarity.

The third model was proposed to improve the disambiguation of unseen abbreviations, which are considered the most challenging problem in the clinical abbreviations disambiguation task. The model was focused on implementing a one-fits-all classifier by fine-tuning three pre-trained BERT models that differ in the size of the clinical data that were used to generate the model. The proposed model was applied to the whole UMN data set which is conducted from 75 clinical abbreviations.

Chapter 4

EXPERIMENTATION

This chapter discusses the results of the three proposed models, which we amply described with their architectures in the previous chapter. In the first section, we will illustrate the result that the models achieved on applying the supervised machine algorithms (SVM, NB) on 13 clinical abbreviations from the UMN corpus. This model tested four aggregations methods for generating the feature vectors fed to these separated classifiers of supervised machine learning algorithms. Two static word embedding pre-trained models were used to test these methods. Also, we studied the effect of the distribution of the annotated data among a set of the expansion on the performance of the proposed models (Jaber and Martínez, 2021a).

In the second experiment, we go deeper to focus on different annotated numbers of annotated examples for abbreviations instead of the equal size of the data set as in the previous one (500 sentences for each abbreviation). The result of implementing the hybrid approach that the SCiAD data set. The disambiguation model was determined based on the available annotated data for each abbreviation. This work was part of our participation in the SDU@AAAI-21 shared task that was held in 2021 (Jaber and Martínez, 2021b).

In the third section, the result of implementing one-fits-all classifier, which was applied on the UMN data set, will be analyzed. First, the result will be compared among the three pre-trained models and then it will be compared with the most related work. Moreover, we will evaluate the models' performance in predicting unseen expansions (Jaber and Martínez, 2022).

4.1. Separated classifiers on a set of clinical abbreviations

This experiment aimed to implement a separated classifier for each abbreviation in the data set so that the data set was separated into 13 data sets based on the abbreviations name. Each data set contains 500 annotated examples randomly split for 80% 20% as training and testing data, respectively. Three experiments were executed on these data sets. So that, 13 *3 experiments were implemented in addition to implementing a 5-fold cross-validation strategy.

Baseline experiments were implemented with traditional linguistic features as feature vectors that were fed to SVM and NB algorithms. The average accuracy achieved was 94.3% and 91.82% by SVM and NB, respectively. To improve the performance of the classifiers, two additional experiments were conducted for each abbreviation using the pre-trained word embeddings, which have been explained in the previous chapter (section 3.1.2).

The second round of experiments were implemented using a pre-trained word embedding generated via PMC biomedical resources; a 200-dimensional feature vector was extracted for window size five around the targeted abbreviation. Then, four aggregation methods were applied to generate the feature vectors. These methods were named SUM_WE, MAX_WE, MIN_WE and AVG_WE. The same processes were followed in the third phase of the experiments. The second pre-trained model was used to extract the feature vectors. This model was trained on PubMed abstracts and Wikipedia's resources in addition to PMC.

Table 4.1 summarizes the results of these experiments. As shown, two aggregation strategies MIN_WE and MAX_WE improved all the models across SVM and NB. On the contrary, SUM_WE and AVG_WE failed to improve the models with NB classifiers, but the improvement was achieved on SVM classifiers. It is noticeable also that SVM with word embedding features improved the performance %2.005 and %2.485 from the baseline experiments.

About the four aggregation strategies, a slight difference between their performance was achieved, MIN_WE achieved the best average accuracy (96.61%) when SVM was applied with PMC word embedding feature vectors. On the other hand, MAX_WE achieved the best result among all the experiments with SVM with PMC, PubMed and Wikipedia feature vectors with average accuracy 97.08%.

Our work differs from (Y. Wu, Xu, et al., 2015) in that they trained their own MIMIC II word embedding model, whereas we used a generated ones from a combination of biomedical and general resources. Furthermore, they achieved the best results by combining traditional and word embedding features. Our best results were obtained by utilizing the pre-trained word embedding features alone.

In order to tackle the effect of the distribution of senses examples among abbreviations data set, we analyzed the result of the model for these abbreviations. Our data set has four abbreviations with a majority sense distribution less than 80% (C3, CVA, BK, NAD). The rest of the abbreviations (AMA, CVP, CVS, BAL, ASA, OTC, FISH, ER, SBP) have a majority sense greater than 80%.

In Figure 4.1 and Figure 4.2, we present accuracy results for different aggregation

TABLE 4.1: Average accuracy of the WSD systems using pre-trained word embedding on 13 abbreviations selected from the UMN data set.

Feature resources	Experiment	Average accuracy (%)	
		SVM	NB
Linguistic	Baseline	94.30	91.82
Pre-trained WE by PMC	MIN_WE	96.61	92.91
	MAX_WE	96.15	93.00
	SUM_WE	96.47	90.59
	AVG_WE	95.99	84.59
Pre-trained WE by Wikipedia PubMed PMC	MIN_WE	97.07	92.91
	MAX_WE	**97.08**	93.34
	SUM_WE	96.69	90.82
	AVG_WE	96.30	86.60

strategies, which were obtained by implementing SVM on word embedding models that were trained on PMC, PubMed and Wikipedia resources since this model were achieved the best results. The abbreviations of these figures were selected based on the distribution of the annotated examples among the difference senses for them. Figure 4.1 includes the abbreviations that have a majority sense over 80%. Figure 4.2 includes those that have a majority of senses less than 80%.

From these two figures, it is clear that all the models achieved a high accuracy regardless of the distribution of the majority sense. The accuracy of each abbreviation model achieved above 90% over the fourth aggregation strategies (except NAD). This result implies that the skewed distribution of senses does not affect the performance of the models.

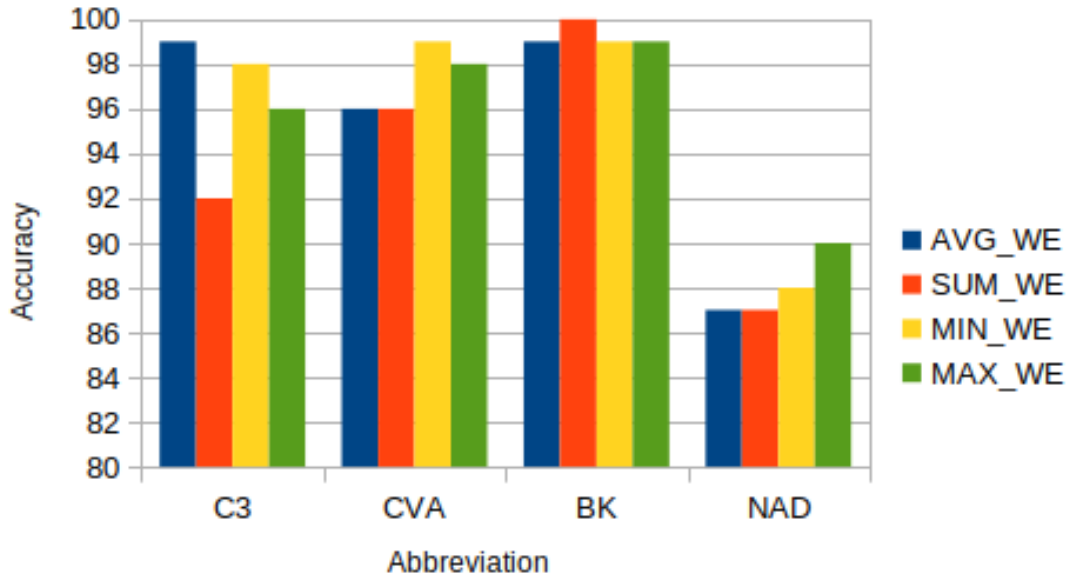


Figure 4.1: Disambiguation accuracy of abbreviations with majority sense > 80%.

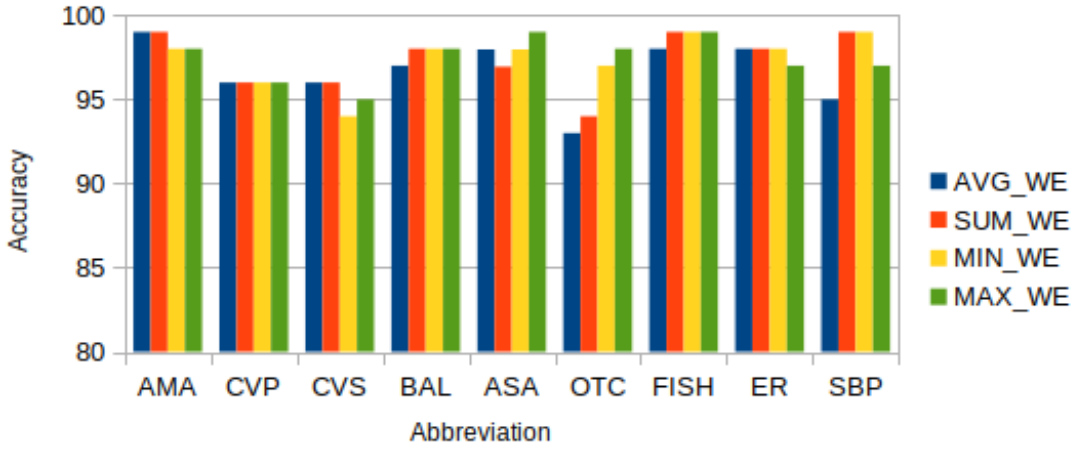


Figure 4.2: Disambiguation accuracy of abbreviations with majority sense < 80%.

4.2. Hybrid approach for disambiguation rare abbreviations

This section discusses the result of our proposed hybrid approach, which was followed in the Acronym Disambiguation shared task. The acronyms distribution among the two approaches will be illustrated. In addition to the performance of the models in both training and tested phases will be reported.

Training Phase

During this phase, training and development data sets were merged to expand the data set size for each acronym. Our goal was to create a separated classifier for every acronym that has more than 20 examples. So that, the training data was separated into 450 data sets (for 450 acronyms). The distribution of the entire data set for machine learning and KB approaches is shown in Table 4.2. 450 acronyms with 53,702 annotated examples are disambiguated by three machine learning models (SVM, NB, kNN). At the same time, the cosine similarity method disambiguated 282 acronyms with 2,521 annotated examples.

TABLE 4.2: Distribution of data sets, acronyms over two proposed models in the training phase.

Model	Data set	Acronyms	Expansions
Machine Learning	53,702	450	1,601
Knowledge based	2,521	282	594
Total	56,223	732	2,195

The training data set contains 634 expansions with fewer than ten annotated examples from various acronyms. These expansions were replicated through oversampling

techniques using the scikit-learn library to balance the data set. Then, for all acronyms in machine learning models, 5 fold cross-validation was used. Furthermore, the training data set contains ten non-ambiguous acronyms, each of which has one expansion in their data set.

Table 4.3 shows our result on training phase, NB with cosine similarity achieved the highest performance with precision 90.31% , recall 87.16% and F1-macro 84.37%.

TABLE 4.3: The average performance of the three proposed hybrid approaches implemented in the training phase.

Model	Precision	Recall	F1-macro
NB-KB	**90.31%**	**87.16%**	84.37%
SVM-KB	90.20%	86.78%	**88.16%**
KNN-KB	83.85%	79.59%	79.53%

Testing Phase

When the organizers released 6,218 annotated examples for 618 acronyms for the testing data set, for each acronym we checked if the acronym was provided in the training phase and if there was a generated predicted model built by supervised way or not. We found that 444 acronyms had predicted models by supervised approach so that those predicted models were used to disambiguate them. On the other hand, 174 acronyms had less than 20 annotated examples, so cosine similarities were used to disambiguate them. Figure 4.3 depicts the flow chart of the data in both phases.

TABLE 4.4: Distribution of data sets, acronyms over two proposed models in the testing phase.

Model	Data set size	# of acronyms
Machine Learning	5,876	444
Knowledge based	342	174
Total	6,218	618

TABLE 4.5: The average performance of the three proposed hybrid approaches in testing data set.

Model	Precision	Recall	F1-macro
NB-KB	**92.15%**	**77.97%**	84.47%
SVM-KB	91.66%	73.33%	**81.48%**
KNN-KB	90.26%	67.51%	77.25%

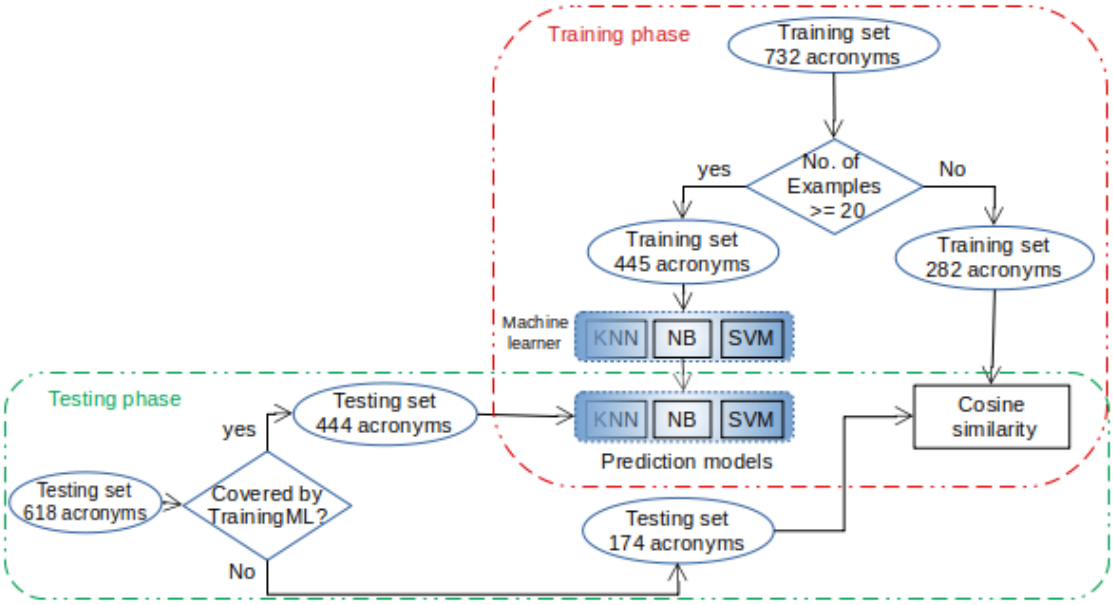


Figure 4.3: Data flowchart of Acronyms over Hybrid approach in training and testing phases.

Table 4.5 shows the final scores for our systems were reported by the organizers. The best performance achieved precision 92.15%, recall 77.97% and F1-macro 84.47%, for a hybrid approach with NB and cosine similarity.

TABLE 4.6: Performance of the participating systems in Acronym Disambiguation task.

Team Name	Precision	Recall	F1-macro
UC3M	**92.15**	**77.97**	**84.37**
AccAcE	93.57	83.77	88.40
GCDH	94.88	87.03	90.79
Spark	94.87	87.23	90.89
Dumb AI	95.95	89.59	92.66
SciDr	96.52	90.09	93.19
hdBERT	96.94	90.73	93.73
DeepBlueAI	**96.95**	**91.32**	**94.05**
Baseline (Freq.)	89.00	46.36	60.97
Human Performance	97.82	94.45	96.10

Comparing our results with the participants of the task, from Table 4.6 we can see that our model failed to outperform any of the participants' models that are based on neural networks approaches (Veyseh et al., 2021).. The winner of this task (Pan et al., 2021) employed pre-trained BERT model NSP strategy and formulate the problem as a binary classification task.

4.3. One-fits-all classifier for disambiguate unseen abbreviations

In this work, three pre-trained BERT models were fine-tuned to disambiguate 75 clinical abbreviations from the UMN data set. One-fits-all classifier was implemented on the top of the three models. The data set was split for 60% training, 20% validation and 20% testing data sets corresponding to 25,865, 3,695 and 3,695 examples, respectively. For multi-classifier demands, the senses of all abbreviations were labeled from 0 to 347, representing the model classes after dropping all examples annotated with **"UNSURED SENSE"** and **"GENERAL ENGLISH"**.

Figure 4.4 and Figure 4.5 illustrate the changes in the accuracy during the 5 epochs, as shown, the accuracy in the first epoch of the MS_BERT was the lowest and then in the next epoch, the model accuracy increased to achieve the best performance among the other two models.

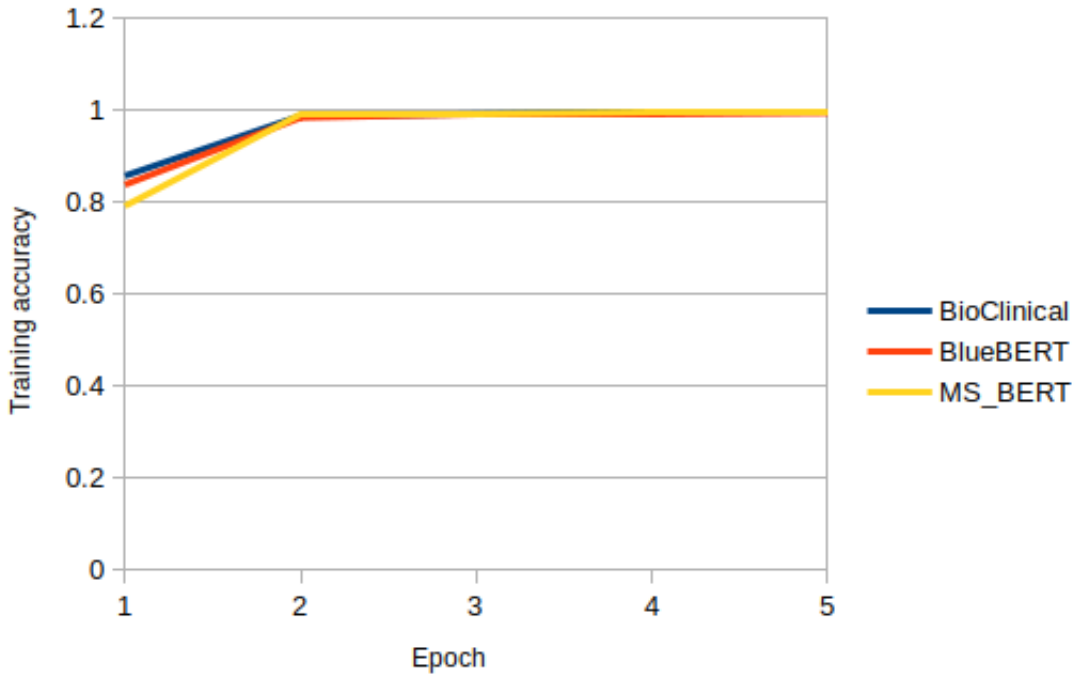


Figure 4.4: Accuracy of the three models during the training phase.

Table 4.7 summarizes the performance of the three models. MS_BERT achieved the best performance with an accuracy of 99.13%. Then, the following model was the Bio_Clinical model with an accuracy of 98.99%. Thus, BlueBERT achieved the lowest accuracy with 98.75%, all these results achieved on the testing data set.

As mentioned in the previous chapter, there were many works for disambiguating the clinical abbreviations on the UMN data set. Each work had a different goal behind this disambiguation. Some focus on the representation of the data and others focus on increas-

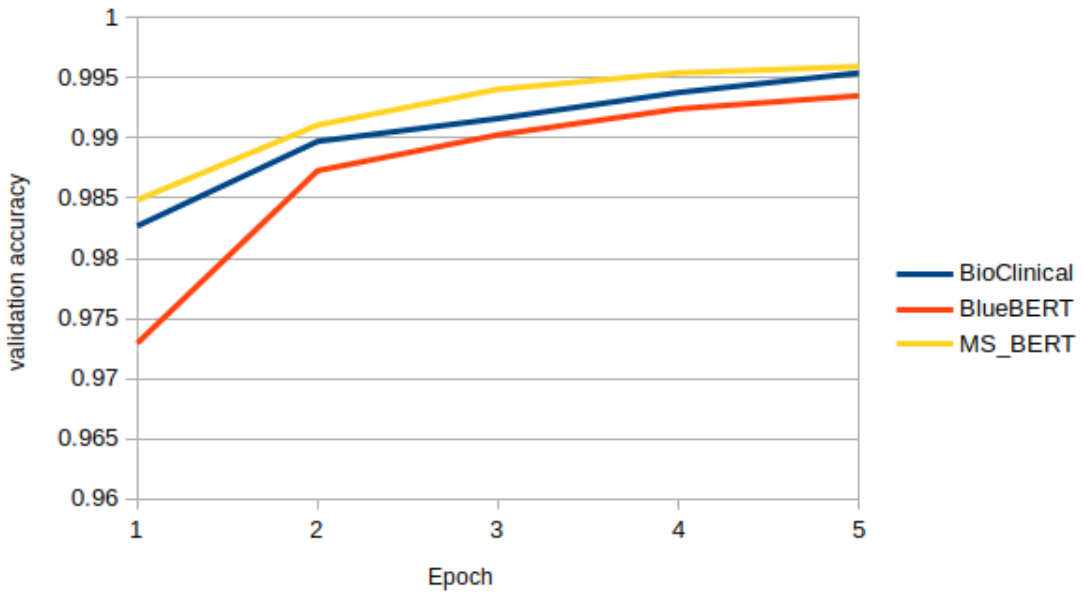


Figure 4.5: Accuracy of the three models during the validation phase.

TABLE 4.7: Accuracy results for the UMN data set. Slightly differences between the three pre-trained models.

Model	**Accuracy (%)**		
	Training	Validation	Testing
MS_BERT	**98.98**	**99.11**	**99.13**
Bio_Clinical	98.85	98.97	98.99
BlueBERT	98.46	98.73	98.75

ing the number of abbreviations to be disambiguated collaborated with the biomedical domain. We will compare our results with the most recent and similar approaches to our goal.

Table 4.8 shows five previous related work results, which are compared with our work. (Joopudi et al., 2018) and CLASSE-GATOR (Kashyap et al., 2020) implemented a separated classifier for each abbreviation on the data set so that data augmentation techniques, mainly the reverse substitution method was applied to increase the number of training examples. (Joopudi et al., 2018) trained separated CNN on the generated data then tested on the manually annotated data set (169 abbreviations). The micro accuracy achieved was 77.83%. The last experiment was trained and tested on 50 abbreviations of the UMN data set achieved an average accuracy of 95.14%.

On the other hand, in the CLASSE-GATOR approach, the researchers extracted a set of abbreviations and their senses from PubMed, trained separated LR classifiers for each abbreviation. They evaluated the model on two data sets, the first one contained nine abbreviations of 245 clinical notes annotated manually. The second one was 52

TABLE 4.8: Accuracy results across several previous works. Our model achieves the state-of-the-art with MS_BERT pre-trained model.

Multi - Classifier			
Approach	Data set	Abb. No.	Acc(%)
CNN	UMN	50	95.14%
	manually annotated	169	77.83%
CLASSE-GATOR	manually annotated	9	63.04%
	UMN	52	76.62%
One-fits-all Classifier			
Approach	Data set	Abb. No.	Acc(%)
ELMo+Topic	UMN	30	74.76%.
Candidate Classification	MSH	203	96.83%
	ShARe/CLEF 2013 Task 2	996	77.97%
	UMN	75	98.39%
LMC	UMN	41	71%
	MIMIC RS	*	74%
	LMIRS	*	69%
MS_BERT (Our approach)	UMN	75	**99.13%**

abbreviations from the UMN. CLASSE-GATOR achieved an average accuracy of 87.9 % across 1,256 acronyms. CLASSE-GATOR achieved an average accuracy of 63.46% of the UMN abbreviations when using LR with equal class distribution, 75.00% when using BERT and 76.92% when using BioBert.

Three recent works implemented a one-fits-all classifier. ELMo+Topic (Li et al., 2019), Candidate classification (Kim et al., 2020) and LMC (Adams et al., 2020). The first two approaches were fine-tuned ELMo and XLNet language models. ELMo+Topic disambiguated 30 abbreviations from the UMN data set. The training examples were represented by combining a contextualized word embedding generated from trained ELMo on MIMIC III data set concatenating with their clinical topic. A fine-tuning model for each abbreviation was implemented with an average accuracy of 74.76%.

Candidate Classification approach (Kim et al., 2020) used relative position encoding to hide the length of the candidate expansion during the training phase to overcome the problem that a set of candidate expansions could be in different lengths. The classifier was fed with the embeddings of candidate expansion contextualized by the context as input. The idea was applied using XLNet language modeling and tested on three different data sets. The accuracy achieved on the UMN data set was 98.34%.

The latter approach applied a deep probabilistic neural network model used in this work to generate the feature vectors. This work (Adams et al., 2020) aimed to learn the distributional properties space by feeding abbreviation, context and metadata. Then, the model inferred if the candidate expansion belongs to this space or not. The system was

tested in three different data sets, apart of them, 41 abbreviations from UMN data set, the accuracy achieved was 71%. Our fine-tuning model by MS_BERT outperformed the state-of-the-art of recent work with accuracy achieved 99.13%.

TABLE 4.9: Three samples of mispredicted abbreviations by MS_BERT model.

Abb.	True Expansion	No. of annotated examples	Predicted Expansion	No. of annotated examples
AVR	Aortic Valve Resistance	4	Aortic Valve Replacement	381
LE	Left Ventricle: LV	5	Right Atrium	394
GT	guttae: GGT	1	gutta	16

Since MS_BERT achieved the highest performance among the tested models, we analyzed its performance and determined the erroneous model predictions for helping us to improve the model in the future. We chose three annotated examples that were mispredicted. As shown in Table 4.9, for **AVR** and **LE** abbreviations, the model failed to distinguish between two expansions if they belong to the same topic, in our case, both **AVR** and **LE** actual and predicted expansions related to heart as a public domain. Both times, the systems were biased to the expansions that have more annotated data. Another observation from the result, the model failed to differentiate between singular and plural forms as **GT** abbreviation, since **gutta** is a plural form of **guttae**. The model biased to the more annotated data.

4.4. Conclusion

Pre-trained word embedding models, which were trained on PubMed, Wikipedia and PMC, that were used as a feature vector for 13 abbreviations from the UMN data set, improved the average accuracy of these models using MAX_WE equation to generate the vector features for each example in the data set. Meanwhile, SVM achieved better average accuracy than NB algorithms; the average accuracy achieved was 97.07% by SVM. In this experiment, we also studied the effect of the distribution of the training data set examples among the senses for each abbreviation. The result showed that the imbalanced distribution of the data set does not affect the performance of the model.

Based on our proposed model in the second experiment, which was applied to SCiAD as part of participation on SDU@AAA-21 task 2, 445 and 444 acronyms were disambiguated by supervised machine algorithms in the training and testing phase. Meanwhile, 282 and 174 acronyms (considered rare abbreviations) were disambiguated using the cosine similarity approach. Among the three applied supervised algorithms which were SVM, NB and kNN. NB with cosine similarity achieved the best performance in both

training and testing phases. Unfortunately, the model failed to achieve good performance compared with the other participants of this shared task who implemented their models based on neural network architectures.

One-fits-all classification by fine-tuning three BERT-based systems outperformed the state-of-the-art of the previous work on 75 clinical abbreviations from the UMN data set with a slight difference between them (Bio_ClinicalBERT, BlueBERT, MS_BERT). MS_BERT achieved 98.75% of accuracy. This model predicted 26 unseen abbreviations. Meanwhile, it failed to predict the proper expansions if they belong to a related topic such as heart parts and the model predicted the expansion with more annotated data.

Chapter 5

CONCLUSIONS AND FUTURE WORK

The summary of this research work is firstly presented in this chapter, then the contributions will be mentioned in addition to the hypothesis validations. Finally, the limitations of this study and possible future work are addressed.

5.1. Summary

Digital worlds are overwhelmed with unstructured data; it has been proven that these accumulated data provide a wealth of knowledge for decision-makers. Its lack of structured data makes it hard to process and analyze directly by the computerized systems. From here, the importance of NLP to analyze texts as part of unstructured data. The capability of generating structured data from unstructured data allows use these data in decision-making systems. In particular, this is specially necessary in the healthcare domain

Nowadays, health information systems document the medical history of the patients and the different treatments received for them in the form of EHR. EHR contains valuable patient data such as laboratory results, diagnosis, demographic information, radiological images, procedures and treatments but data semantics is barely considered; this prevents to combine and reuse data for different purposes. This vast amount of data could be used in several ways to infer new knowledge and to monitor patients in a more effective way, for instance, by generating summaries of patient episodes to help in clinical practice or helping in treatments prescriptions.

Focusing on clinical data, transforming clinical narrative in structured controlled vocabulary is a challenge for several reasons; apart of the complexity of extracting relevant facts from free text it is susceptible to spelling errors, ungrammatical sentences and containing a large number of medical abbreviations because it is speedy written.

Abbreviations represent a portion of clinical text ranging from 3% to 10%. Up to 33% of them are ambiguous. As an important consequence, misunderstanding abbreviations could cause serious problems related to patient safety.

WSD is an essential NLP task because ambiguity is an inherent characteristic of human language. WSD aims to determine the correct meaning of an ambiguous word in a

sentence. Several approaches have been implemented to address WSD, starting from simple ruled-based approaches, traditional machine learning to advanced neural networks.

Clinical abbreviation disambiguation task as a WSD task encompasses four crucial parts. First, resources represent the vital part of this task, being a real bottleneck. It is hard to get available clinical data due to privacy issues. The UMN (Moon et al., 2012) public corpus is labeled with 75 abbreviations, which are tiny numbers of abbreviations related to the current number of existing abbreviations; there are more than 197,000 unique medical abbreviations found in the clinical text according to this study (Y. Liu et al., 2015). However, many inventories, ontologies and corpora related to the biomedical domain have been used to overcome the data bottleneck, but the studies proved that clinical abbreviations differ from other types of abbreviations such as the biomedical and scientific ones.

Many researchers have tried to face this issue by automatically generating their training examples, avoiding manual annotation to mitigate the cost and time-consuming process. This process is called "Reverse Substitution". Increasing the number of training examples could be possible but this does not solve the imbalanced distribution among abbreviation senses. Furthermore, every day new abbreviations appear and consequently unseen abbreviations remain a problem because models can only predict those abbreviations for which they have been trained.

As known, computers cannot process text data directly, so that text data should be represented in numerical ways that capture semantic relations among the words, which is considered a requirement for any WSD task. Different traditional linguistic features had been used to represent words : POS tags, relative positions, the form of the words. And then, statistical approaches have been used to form numerical feature vectors; BOW (McCray et al., 1994) and TF-IDF (Jones, 2004) are examples of these approaches.

With the arrival of word embeddings, representation of text into numerical feature vectors holding semantic information achieved better performance than the traditional ones. Static word embeddings obtained by training deep neural network architectures in large unannotated data could be used to feed into a classifier. The main drawback of this approach is that the feature vector for any words is always the same, whatever the context of these words. From here, the language modeling-based approaches, known as contextualized word embeddings, come to take into account the word's context while generating the feature vectors, which has made these vectors more representative of semantics.

Evaluation measurements are required to indicate the benefits of the proposed models and compare the performance with other models. Several measurements have been used to evaluate the classification task of WSD. Each one is chosen based on the goal of the model. If the study of the model aims to track the distribution of the training data among classes, Precision, Recall and F-macro are the best choice for this goal. Meanwhile, if the

study does not focus on this issue, accuracy could be a good choice. Since the clinical abbreviation corpus data set is strongly imbalanced. Most of the expansions have one training example. The majority of the previous works have used accuracy to evaluate the performance of their models.

In this thesis WSD task is addressed as a classification problem that can be managed using traditional machine learning algorithms and deep neural networks. Furthermore, these algorithms are classified into supervised, semi-supervised and unsupervised as well as KB-based approaches.

Supervised learning has been proved its efficiency in clinical abbreviation disambiguation task. These models have been varied from probabilistic such as NB (Hart et al., 2000), methods based on similarity of examples like kNN (Ng and Lee, 1996), based on discriminating rules like DT (Black, 1988) or could be based on kernel functions like SVM. Almost all of these methods were applied in previous works, adopting different feature vectors generation techniques.

On the other hand, supervised deep neural networks have advanced the performance of the disambiguation models by capturing similarity and adapting the context in the vector features as an input to their models. LSTM, a type of RNN architectures, was used to represent long-term dependencies for the sequence of text. But unfortunately, not for too long sentences. Seq2seq models (Sutskever et al., 2014) proposed a new architecture based on the encoder and decoder connected through a context vector representing the whole input sequence. The Seq2seq architecture, which is based on a stack of LSTM, keeps processing the sequence word by word requiring long time of training.

The Transformers architecture gets state-of-the-art results for many NLP tasks replacing the LSTM(Hochreiter and Schmidhuber, 1997) with FFNN architectures from the seq2seq models adding a self-attention layer. Transformers open up processing the sequence of words in parallel, which means learning will not need a long time compared with the previous architectures. Secondly, the architecture could be easily parallelized with GPUs. Thirdly and the most important one, is the self-attention mechanism, which allows each token to generate its attention vector that aims to "pay attention" into specific other words in the input sequence.

The contribution of this thesis is reflected in three different models on various abbreviations data sets to improve prediction of rare and unseen abbreviations. First, a traditional supervised machine learning was implemented to disambiguate 13 clinical abbreviations from the UMN data set. An existing static pre-trained word embedding generated from a combination of biomedical and general language resources was used to represent each token surrounding the target abbreviation with a window size of 5. These extracted vector features were aggregated by summation, average, minimum and maximum operations.

And then, they were fed into SVM and NB models. Two models (SVM, NB) X 13 Abbreviations) X 5 (baseline plus 4 aggregations operations for feature vectors) classifiers were implemented for 13 abbreviations. The best average accuracy achieved using the maximum aggregation operation for word embedding generated from a combination of PMC, PubMed and Wikipedia, was 97.08%.

As part of rare abbreviations disambiguation, a system to participate in SDU@AAAI-21 shared task was defined. A hybrid approach was developed to disambiguate abbreviations based on the availability of the annotated data. The released shared task aimed to disambiguate scientific acronyms and hence, the SCiAD (Veyseh et al., 2020) data set where launched. Our result outperformed the baseline, but it failed to outperform the participant's models that implemented neural network approaches. Implementing a separated classifier for each abbreviation is an unpractical approach because the generated model will not be able to predict unseen abbreviations
expansions since they learned to predict a specific set of expansions for each abbreviation. To tackle this problem, we proposed a new approach based on a one-fits-all classifier.

A one-fits-all classifier was implemented to disambiguate 75 abbreviations in the UMN data set. Three fine-tuned Transformer-based (BERT) architectures (Devlin et al., 2019) were tested to achieve the goal. The difference between these models is the data source used to generate the models. NSP training approach was used in the fine tuning process, which means each input data is represented from two sentences: the sentence which has a target abbreviation and the sentence with the correct expansion of this abbreviation. MS_BERT, which is fine-tuned with extra 35.7 million words of clinical notes, achieved the best accuracy of 99.13% and achieved state-of-the-art results. Furthermore, the model could predict 26 unseen abbreviations.

5.2. Contributions

This research proposed a model to improve clinical abbreviation disambiguation task. The following are the main contributions of this thesis:

- A complete description of the state-of-the-art of clinical abbreviation disambiguation task. Available resources and data representation are detailed as well as the evaluation measurements used in WSD and the different approaches that have been previously applied to solve the problem of disambiguation.
- The first approach proposed in this research defines is a separated classifier to disambiguate 13 abbreviations from the UMN data set, static pre-trained word embeddings are used as feature vectors to fed supervised machine learning algorithms.

- Separated classifiers for disambiguate acronyms using in the SCiAD corpus provided by SDU@SDU@AAAI-21challenge that have more than 20 annotated examples per abbreviation. Static pre-trained word embeddings are used as feature vectors to fed several supervised machine learning algorithms. For each acronym having less than 20 annotated examples (rare acronyms) a knowledge-based approach is used to disambiguate them.
- Three fine tuned pre-trained BERT models tested with the UMN data set comparing the performance of the three models.
- Evaluation of the three fine tuned pre-trained BERT models for the ability of predict unseen and rare data.

5.3. Hypothesis validation

In this section, the hypothesis posed in the first chapter are related to the results obtained in this thesis.

Hypothesis 1:
If we want to generate a model that disambiguates clinical abbreviations from scratch, then we could get enough clinical annotated data to perform the experiments.

The essential part of any machine learning model is data availability, as long as we have a large size of data for our specific task, as the model could learn enough and achieve high performance. In the WSD task, three types of data resources could be used. Sense inventory could be represented as a dictionary that provides a set of different meanings for the ambiguous word. Annotated Corpora aim to learn the models from a set of examples and thus "supervise" the model to solve new ones. On the other hand, unlabeled corpora are demanded to either learn deep neural networks models or to be used to generate more annotated data.

In Chapter two, we review all the available resources that have been used to disambiguate clinical abbreviations. First, it is hard to get unlabeled data (raw clinical text) due to privacy issues, in addition to the pre-processing steps for this data to be available as a de-identification step. Hence, just one hospital makes its data available for the public to advance the research, which is known as MIMIC, with 2 million records from different types of medical data. Regarding annotated corpus, also UMN corpus from Minnesota university is also publicly available.

The UMN corpus comprises 75 abbreviations with 500 annotated examples per abbreviation. First, the number of abbreviations and available clinical senses inventories do not reflect current clinical abbreviations used in the clinical narrative because abbreviations are constantly emerging. Second, the corpus itself is strongly imbalanced between expansions annotated sentences. Third, since 500 annotated sentences for each abbreviation are not enough to implement any neural network models, several studies have tended to avoid manual annotations due to its difficulty by applying the reverse substitution method to increase the training data set. Reverse substitution method fails to deal with unseen and rare abbreviations. Furthermore, much work is required to have a complete inventory of senses that could be automatically updated, for instance, UMLS covers just 35% of clinical expansions.

In summary, abbreviations are created by the doctors and the clinicians with no standard rules and this prevents to easily automate recognizing new abbreviations in medical texts. For these reasons, annotated clinical data are scarce resources and limits the evolution of any machine learning model.

Hypothesis 2:

If contextualized word embeddings are used for representing the data, then semantic similarities could be represented better than using static word embeddings.

Machine learning algorithms could not manipulate text data directly. With the evolution of computer science, one of the foremost NLP challenges is how text data could be represented to make the models understand the language as humans do. This type of representation could not reflect the meaning of the words, as it is known that words may have many meanings based on the context where it is used. This is a relevant issue working with abbreviations that are highly ambiguous terms. Several approaches have been tested, beginning with statistical approaches, which count the frequency and the present/absence of the word in a document.

In neural networks, text is represented as low-dense dimension vectors allowing capturing relations among words in a sequence data. Static word embeddings leverage this architecture to learn the relations between words, although one drawback for this model is that it assigns one vector to the token then uses it to represent the token without taking into account the context. However, contextualized word embedding architectures improve the representation based on the context in which the token is used.

In this research work static word embeddings have been tested on disambiguating scientific acronyms showing that these embeddings failed to improve the model's performance compared with the contextualized word embedding approaches that included deep

learning methods.

Hypothesis 3:

If clinical Transformers based language models, such as Bidirectional Encoder Representations from Transformers (BERT), are fine tuned, then the model performance could be improved.

A Transformer is a novel architecture to deal with long dependencies in the sequence to sequence models. The architecture depends on the self-attention mechanism to compute the representation of the inputs and outputs without depending on RNN architectures. This novel approach outperformed the previous approaches for most NLP tasks.

The third proposal, we fine-tuned three BERT-based models improving the performance of the disambiguation task and outperforming all the previous works.

Hypothesis 4:

If a one-fits-all classifier is implemented for all the clinical abbreviations, then the classifier could predict unseen expansions.

Previous research works defined a separated classifier for each clinical abbreviation. Under this assumption, there is a need of increasing the training data for each abbreviation to have enough data for each abbreviation expansion in the data set. First, related to clinical abbreviations nature, many abbreviations are rarely used due to several reasons such as the scope of the term; abbreviation could be local to hospitals or care centers or could have a wider scope (national or international). Consequently, there are not enough data to increase the training data sets. Secondly, the classifiers will fail to predict new expansions because they learned on a restricted set of senses. However, the one-fits-all classifier will generalize to capture new expansions and thus, it will be no need to increase the training data since it could predict any new example. Related to the proposed model in this thesis, it could predict 26 unseen expansions, which validates this hypothesis 4.

5.4. Challenges and limitations

The state-of-the art in clinical WSD has shown that lack of resources is the most significant issue. For example, the public UMN data set contains just an annotation for just 75 abbreviations, which is considered very little in relation to the most widespread used abbreviations. In addition, there is no any exhaustive clinical sense inventory that covers the vast number of existing abbreviations.

English language dominates most NLP research. There are some models for other less resources languages, but it is hard to fine-tune or pre-train them because data scarcity. For example, the BARR2 (Intxaurrondo et al., 2018) corpus contains 3,343 records for 730 abbreviations in the Spanish language; 87 abbreviations have more than one sense. Furthermore, among these 87 abbreviations, there are many annotation mistakes, for example, both **"F" "FR"** is annotated for the same expansion **"French"**.

Since we have not gotten access to any Spanish EHR, we tried to apply the reverse substitution method to increase the BARR2 data set by leveraging sentences from MEDLINE Spanish publications. We selected the most frequent abbreviations in BARR2. Table 5.1 shows the total number of sentences from MEDLINE in addition to the number of sentences that BARR2 has for each abbreviation.

TABLE 5.1: Extracted sentences number from MEDLINE in Spanish language.

Abb	Expansion	BARR	MEDLINE	Total
SNC	sistema nervioso central	1	119	**129**
	sindrome nefrotico congenito	1	8	
PL	percepcion de luz	1	0	**78**
	periodo de lavado	1	0	
	Puncion lumbar	9	67	
MM	mieloma multiple	3	117	**165**
	movimiento de mano	5	0	
	milimetro	1	39	
AV	agudeza visual	15	84	**137**
	Auriculoventricular	18	1	
	acceso vascular	6	1	
FA	Fosfatasa alcalina	12	8	**77**
	Fibrilación auricular	56	1	

Apart from training examples in Spanish clinical texts, there is a lack of pre-trained models that could be helpful to improve any suggested model to disambiguate clinical abbreviations. For instance, there are two versions of Spanish Clinical Embedding that have been trained using 315 million clinical words extracted from EHR and Ph.D. medical theses (Gutiérrez-Fandiño, Armengol-Estapé, Carrino, et al., 2021). Unfortunately, there is no further information about these generated vectors, such as the architecture that is used to generate these vectors.

For Transformers-based pre- models, BETO (Canete et al., n.d.) is a BERT based model trained on a collection of Spanish texts from Wikipedia and OPUS project (Tiedemann, 2012). The model uses around 31K sub-words. Spanish RoBERTa(Gutiérrez-Fandiño, Armengol-Estapé, Pàmies, et al., 2021) another pre-trained model using 570GB text from the National Library of Spain.

5.5. Future work

As the objectives of this thesis had been achieved, there is still much room for improvement concerning clinical abbreviations processing. Some of remaining issues are given below:

New transformers-based language model fine-tuning strategies

BERT model comes with two training approaches: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM is as fill in a blank task, where tokens are masked in sentences and are used to fed the model to predict masked tokens based on surrounding words (context). A different fine tuning strategy applied to the clinical abbreviation disambiguation task could be masking senses of ambiguous abbreviations. The result could be compared to the results of the NSP fine tuning strategy testing in this thesis.

Extending abbreviation inventories

With the most recent work related to medical terms inventory (L. G. Liu et al., 2021), the wok could be extended to disambiguate 104,057 abbreviations and 170,426 expansions which are collected and unified from different clinical and biomedical inventories and ontologies.

Self supervised learning

Deep learning algorithms are widely acknowledged to be data-hungry. Traditional deep learning models are known as "end-to-end," which refers to that model takes an input **X** and the model's last layer produces the predicted value **Y**. Normally, these systems fail to predict data out of the training data distribution. However, despite the abundance of unlabeled web data in the big data era, high-quality data with manual human annotation may be prohibitively expensive.

Self Supervised Learning (SSL) (X. Liu et al., 2021)is one of the latest learning approaches that leverage the availability of a massive number of unannotated data to learn as a supervised model without the necessity of any labeled data. The intuition idea behind the SSL that learning deeper patterns from the training data instead of just learning similarities. Many existing models based on transformers architecture could implement the SSL such as GPT-3 (Floridi and Chiriatti, 2020).

Applying SSL for the clinical abbreviation disambiguation task could improve the system performance in disambiguating unseen abbreviations and automate the disambiguation process for on-time abbreviation creation.

Replication

This work could be replicated in any lexical sample WSD task, to different languages other than English such as Arabic and Spanish.

BIBLIOGRAPHY

Adams, G., Ketenci, M., Perotte, A. J., & Elhadad, N. (2020). Zero-shot clinical acronym expansion with a hierarchical metadata-based latent variable model. *CoRR*, *abs/2010.02010*. https://arxiv.org/abs/2010.02010

Adar, E. (2004). SaRAD: A simple and robust abbreviation dictionary. *Bioinform.*, *20*(4), 527–533. https://doi.org/10.1093/bioinformatics/btg439

Agarap, A. F. (2018). Deep learning using rectified linear units (ReLU). *CoRR*, *abs/1803.08375*. http://arxiv.org/abs/1803.08375

Akhtyamova, L., Martínez, P., Verspoor, K., & Cardiff, J. (2020). Testing contextualized word embeddings to improve NER in spanish clinical case narratives. *IEEE Access*, *8*, 164717–164726. https://doi.org/10.1109/ACCESS.2020.3018688

Alarcón, R., Moreno, L., & Martínez, P. (2021). Lexical simplification system to improve web accessibility. *IEEE Access*, *9*, 58755–58767. https://doi.org/10.1109/ACCESS.2021.3072697

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., & McDermott, M. B. A. (2019). Publicly available clinical BERT embeddings. *CoRR*, *abs/1904.03323*. http://arxiv.org/abs/1904.03323

Ao, H., & Takagi, T. (2005). ALICE: An algorithm to extract abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, *12*(5), 576–586. https://doi.org/10.1197/jamia.M1757

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*. http://arxiv.org/abs/1409.0473

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In I. Guyon, G. Dror, V. Lemaire, G. W. Taylor, & D. L. Silver (Eds.), *Unsupervised and transfer learning - workshop held at ICML 2011, bellevue, washington, usa, july 2, 2011* (pp. 37–50). JMLR.org. http://proceedings.mlr.press/v27/baldi12a.html

Beam, A. L., Kompa, B., Schmaltz, A., Fried, I., Weber, G. M., Palmer, N. P., Shi, X., Cai, T., & Kohane, I. S. (2020). Clinical concept embeddings learned from massive sources of multimodal medical data. *Pacific Symposium on Biocomputing 2020, Fairmont Orchid, Hawaii, USA, January 3-7, 2020*, 295–306. https://psb.stanford.edu/psb-online/proceedings/psb20/Beam.pdf

Berger, A. L., Pietra, S. D., & Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Comput. Linguistics*, *22*(1), 39–71.

Billy, T. (2017). Use of Abbreviations and Acronyms among Healthcare Workers in a Resource Limited Setting. *Journal of Healthcare Communications*, *02*. https://doi.org/10.4172/2472-1654.100063

Bird, S., & Loper, E. (2004). NLTK: the natural language toolkit. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26, 2004 - Poster and Demonstration*. https://aclanthology.org/P04-3031/

Black, E. (1988). An experiment in computational discrimination of english word senses. *IBM J. Res. Dev.*, *32*(2), 185–194. https://doi.org/10.1147/rd.322.0185

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, 144–152.

Campillos-Llanos, L. (2019). First steps towards building a medical lexicon for spanish with linguistic and semantic information. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 152–164.

Canete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Pérez, J. (n.d.). Spanish pre-trained bert model and evaluation data. *PML4DC at ICLR 2020.*

Castro, E., Iglesias, A., Martínez, P., & Castaño, L. (2010). Automatic identification of biomedical concepts in spanish-language unstructured clinical texts. In T. C. Veinot, Ü. V. Çatalyürek, G. Luo, H. Andrade, & N. R. Smalheiser (Eds.), *ACM international health informatics symposium, IHI 2010, arlington, va, usa, november 11 - 12, 2010, proceedings* (pp. 751–757). ACM. https://doi.org/10.1145/1882992.1883106

Chollet, F. (2021). *Deep learning with python*. Manning Publications.

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: pre-training text encoders as discriminators rather than generators. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. https://openreview.net/forum?id=r1xMH1BtvB

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, *12*, 2493–2537. http://dl.acm.org/citation.cfm?id=2078186

Colón-Ruiz, C., Segura-Bedmar, I., & Martínez, P. (2019). Análisis de sentimiento en el dominio salud: Analizando comentarios sobre fármacos. *Proces. del Leng. Natural*, *63*, 15–22. http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6090

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(1950), 4171–4186.

Dooley, M. J., Wiseman, M., & Gu, G. (2012). Prevalence of error-prone abbreviations used in medication prescribing for hospitalised patients: Multi-hospital evaluation. *Internal medicine journal*, *42*(3), e19–e22.

Edmonds, P., & Agirre, E. (2008). *Word sense disambiguation* (Vol. 3). https://doi.org/10.4249/scholarpedia.4358

Ehrmann, M., Rocca, L. D., Steinberger, R., & Tanev, H. (2013). Acronym recognition and processing in 22 languages. *Recent Advances in Natural Language Processing, RANLP 2013, 9-11 September, 2013, Hissar, Bulgaria*, 237–244. https://aclanthology.org/R13-1031/

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Commun. ACM*, *56*(4), 82–89. https://doi.org/10.1145/2436256.2436274

Finley, G. P., Pakhomov, S. V. S., McEwan, R., & Melton, G. B. (2016). Towards Comprehensive Clinical Abbreviation Disambiguation Using Machine-Labeled Training Data. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, *2016*, 560–569. http://www.ncbi.nlm.nih.gov/pubmed/28269852%7B%5C%%7D0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5333249

Floridi, L., & Chiriatti, M. (2020). GPT-3: its nature, scope, limits, and consequences. *Minds Mach.*, *30*(4), 681–694. https://doi.org/10.1007/s11023-020-09548-1

Franceschet, I., Zanetto, A., Ferrarese, A., Burra, P., & Senzolo, M. (2016). Therapeutic approaches for portal biliopathy: A systematic review. *World journal of gastroenterology*, *22*(45), 9909.

Goodfellow, I. J., Bengio, Y., & Courville, A. C. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org/

Gormley, C., & Tong, Z. (2015). *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. " O'Reilly Media, Inc."

Grossman, L. V., Mitchell, E. G., Hripcsak, G., Weng, C., & Vawdrey, D. K. (2018). A method for harmonization of clinical abbreviation and acronym sense inventories. *Journal of biomedical informatics*, *88*, 62–69.

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Carrino, C. P., de Gibert, O., Gonzalez-Agirre, A., & Villegas, M. (2021). Spanish biomedical and clinical language embeddings. *CoRR*, *abs/2102.12843*. https://arxiv.org/abs/2102.12843

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Gonzalez-Agirre, A., Armentano-Oller, C., Penagos, C. R., & Villegas, M. (2021). Spanish language models. *CoRR*, *abs/2107.07253*. https://arxiv.org/abs/2107.07253

Hart, P. E., Stork, D. G., & Duda, R. O. (2000). *Pattern classification*. Wiley Hoboken.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G. E. et al. (1986). Learning distributed representations of concepts. *Proceedings of the eighth annual conference of the cognitive science society*, *1*, 12.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holper, S., Barmanray, R., Colman, B., Yates, C. J., Liew, D., & Smallwood, D. (2020). Ambiguous medical abbreviation study: challenges and opportunities. *Internal Medicine Journal*, *50*(9), 1073–1078. https://doi.org/https://doi.org/10.1111/imj.14442

Hunt, D. L., Haynes, R. B., Hanna, S. E., & Smith, K. (1998). Effects of computer-based clinical decision support systems on physician performance and patient outcomes: A systematic review. *Jama*, *280*(15), 1339–1346.

Inadomi, J. M. (2011). A new clustering method for detecting rare senses of abbreviations in clinical notes. *J Am Coll Surg*, *212*(6), 1049–1060. https://doi.org/10.1016/j.jamcollsurg.2011.02.017.Cost-Effective

Intxaurrondo, A., Marimon, M., Gonzalez-Agirre, A., López-Martıén, J. A., Rodriguez, H., Santamarıéa, J., Villegas, M., & Krallinger, M. (2018). Finding mentions of abbreviations and their definitions in spanish clinical cases: The BARR2 shared task evaluation results. *2150*, 280–289. http://ceur-ws.org/Vol-2150/overview-BARR2.pdf

Jaber, A., & Martínez, P. (2021a). Disambiguating clinical abbreviations using pre-trained word embeddings. *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies*, *5*, 501–508.

Jaber, A., & Martínez, P. (2021b). Participation of UC3M in sdu@aaai-21: A hybrid approach to disambiguate scientific acronyms. In A. P. B. Veyseh, F. Dernoncourt, T. H. Nguyen, W. Chang, & L. A. Celi (Eds.), *Proceedings of the workshop on scientific document understanding co-located with 35th AAAI conference on artificial inteligence, sdu@aaai 2021, virtual event, february 9, 2021*. CEUR-WS.org. http://ceur-ws.org/Vol-2831/paper23.pdf

Jaber, A., & Martínez, P. (2022). Disambiguating clinical abbreviations using a one-fits-all classifier based on deep learning techniques. *Methods of Information in Medicine*. https://doi.org/10.1055/s-0042-1742388

Jimeno-Yepes, A. J., McInnes, B. T., & Aronson, A. R. (2011). Exploiting mesh indexing in medline to generate a data set for word sense disambiguation. *BMC bioinformatics*, *12*(1), 1–14. https://doi.org/10.1186/1471-2105-12-223

Jin, Q., Dhingra, B., Cohen, W. W., & Lu, X. (2019a). Probing biomedical embeddings from language models. *CoRR*, *abs/1904.02181*. http://arxiv.org/abs/1904.02181

Jin, Q., Dhingra, B., Cohen, W. W., & Lu, X. (2019b). Probing biomedical embeddings from language models. *CoRR*, *abs/1904.02181*. http://arxiv.org/abs/1904.02181

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*, 1–9. https://doi.org/10.1038/sdata.2016.35

Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, *60*(5), 493–502. https://doi.org/10.1108/00220410410560573

Joopudi, V., Dandala, B., & Devarakonda, M. (2018). A convolutional route to abbreviation disambiguation in clinical text. *Journal of Biomedical Informatics*, *86*(June), 71–78. https://doi.org/10.1016/j.jbi.2018.07.025

Joshi, M., Pakhomov, S., Pedersen, T., & Chute, C. G. (2006). A comparative study of supervised learning as applied to acronym expansion in clinical reports. *AMIA annual symposium proceedings*, *2006*, 399.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, 427–431. https://doi.org/10.18653/v1/e17-2068

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2nd edition*. Prentice Hall, Pearson Education International. https://www.worldcat.org/oclc/315913020

Kanerva, P. (1996). Binary spatter-coding of ordered k-tuples. *Artificial Neural Networks - ICANN 96, 1996 International Conference, Bochum, Germany, July 16-19, 1996, Proceedings*, *1112*, 869–873. https://doi.org/10.1007/3-540-61510-5%5C_146

Kashyap, A., Burris, H., Callison-Burch, C., & Boland, M. R. (2020). The CLASSE GATOR (Clinical Acronym SenSE disambiGuATOR): A Method for predicting acronym sense from neonatal clinical notes. *International journal of medical informatics*, *137*, 104101.

Kim, J., Gong, L., Khim, J., Weiss, J. C., & Ravikumar, P. (2020). Improved clinical abbreviation expansion via non-sense-based approaches. *Machine Learning for Health*, 161–178.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980

Le, M., Postma, M., Urbani, J., & Vossen, P. (2018). A deep dive into word sense disambiguation with LSTM. *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 354–365. https://aclanthology.org/C18-1030/

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, *32*, 1188–1196. http://proceedings.mlr.press/v32/le14.html

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, *36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

León, F. S. (2018). ARBOREx: Abbreviation resolution based on regular expressions for BARR2. *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018*, *2150*, 302–315. http://ceur-ws.org/Vol-2150/BARR2%5C_paper3.pdf

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Li, I., Yasunaga, M., Nuzumlali, M. Y., Caraballo, C., Mahajan, S., Krumholz, H. M., & Radev, D. R. (2019). A neural topic-attention model for medical term abbreviation disambiguation. *CoRR*, *abs/1910.14076*. http://arxiv.org/abs/1910.14076

Liu, H., Lussier, Y. A., & Friedman, C. (2001). A study of abbreviations in the UMLS. *AMIA 2001, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 3-7, 2001*. https://knowledge.amia.org/amia-55142-

a2001a-1.597057/t-001-1.599654/f-001-1.599655/a-079-1.599902/a-080-1.599899

Liu, L. G., Grossman, R. H., Mitchell, E. G., Weng, C., Natarajan, K., Hripcsak, G., & Vawdrey, D. K. (2021). A deep database of medical abbreviations and acronyms for natural language processing. *Scientific Data*, *8*(1), 1–9.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Y., Ge, T., Mathews, K., Ji, H., & McGuinness, D. L. (2015). Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. In K. B. Cohen, D. Demner-Fushman, S. Ananiadou, & J. Tsujii (Eds.), *Proceedings of the workshop on biomedical natural language processing, bionlp@ijcnlp 2015, beijing, china, july 30, 2015* (pp. 92–97). Association for Computational Linguistics. https://doi.org/10.18653/v1/W15-3810

Lowe, W. (2001). Towards a theory of semantic space. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *23*(23).

Lu, C. J., Payne, A., & Mork, J. G. (2020). The unified medical language system SPECIALIST lexicon and lexical tools: Development and applications. *J. Am. Medical Informatics Assoc.*, *27*(10), 1600–1605. https://doi.org/10.1093/jamia/ocaa056

Lukovnikov, D., Fischer, A., & Lehmann, J. (2019). Pretrained transformers for simple question answering over knowledge graphs. *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, *11778*, 470–486. https://doi.org/10.1007/978-3-030-30793-6%5C_27

Ma, C., Zhang, H. H., & Wang, X. (2014). Machine learning for big data analytics in plants. *Trends in plant science*, *19*(12), 798–808.

McCray, A. T., Srinivasan, S., & Browne, A. C. (1994). Lexical methods for managing variation in biomedical terminologies. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 235.

Medsker, L., & Jain, L. C. (1999). *Recurrent neural networks: Design and applications*. CRC press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Y. Bengio & Y. LeCun (Eds.), *1st international conference on learning representations, ICLR 2013, scottsdale, arizona, usa, may 2-4, 2013, workshop track proceedings*. http://arxiv.org/abs/1301.3781

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki,*

Japan, May 7-12, 2018. http://www.lrec-conf.org/proceedings/lrec2018/summaries/721.html

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, *6*(1), 1–28.

Moon, S., Berster, B.-T., Xu, H., & Cohen, T. (2013). Word Sense Disambiguation of clinical abbreviations with hyperdimensional computing. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, *2013*, 1007–16. http://www.ncbi.nlm.nih.gov/pubmed/24551390%7B%5C%%7D0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3900125

Moon, S., Pakhomov, S., & Melton, G. B. (2012). Automated disambiguation of acronyms and abbreviations in clinical texts: window and training size considerations. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, *2012*, 1310–1319.

Moon, S., Pakhomov, Serguei, Melton, & Genevieve. (2014). A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of the American Medical Informatics Association*, *21*(2), 299–307. https://doi.org/10.1136/amiajnl-2012-001506

Mowery, D. L., South, B. R., Christensen, L., Leng, J., Peltonen, L.-M., Salanterä, S., Suominen, H., Martinez, D., Velupillai, S., Elhadad, N., et al. (2016). Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: Share/clef ehealth challenge 2013, task 2. *Journal of biomedical semantics*, *7*(1), 1–13.

Ms-bert. (n.d.). Retrieved January 3, 2022, from https://huggingface.co/NLP4H/ms_bert

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with python: A guide for data scientists*. " O'Reilly Media, Inc."

National library of medicine - national institutes of health: Medline. (n.d.). Retrieved January 3, 2022, from https://www.nlm.nih.gov/index.html

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, *41*(2). https://doi.org/10.1145/1459352.1459355

Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In A. K. Joshi & M. Palmer (Eds.), *34th annual meeting of the association for computational linguistics, 24-27 june 1996, university of california, santa cruz, california, usa, proceedings* (pp. 40–47). Morgan Kaufmann Publishers / ACL. https://aclanthology.org/P96-1006/

Packer, A. L., Biojone, M. R., Antonio, I., Takenaka, R. M., Garcıéa, A. P., Silva, A. C. d., Murasaki, R. T., Mylek, C., Reis, O. C., & Delbucio, H. C. R. F. (1998). Scielo: Uma metodologia para publicação eletrônica. *Ciência da informação*, *27*, nd–nd.

Pakhomov, S. V. (2002). Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. *Proceedings of the 40th An-*

nual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA, 160–167. https://aclanthology.org/P02-1021/

Pakhomov, S., Pedersen, T., & Chute, C. G. (2005). Abbreviation and acronym disambiguation in clinical discourse. *AMIA 2005, American Medical Informatics Association Annual Symposium, Washington, DC, USA, October 22-26, 2005*. https://knowledge.amia.org/amia-55142-a2005a-1.613296/t-001-1.616182/f-001-1.616183/a-118-1.616335/a-119-1.616332

Pan, C., Song, B., Wang, S., & Luo, Z. (2021). Bert-based acronym disambiguation with multiple training strategies. *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*, *2831*. http://ceur-ws.org/Vol-2831/paper25.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, M., & Quan, H. (2020). Clinical abbreviation disambiguation using deep contextualized representation. *Studies in Health Technology and Informatics*, *270*, 88–92. https://doi.org/10.3233/SHTI200128

Peng, Y., Yan, S., & Lu, Z. (2019). Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, 58–65. https://doi.org/10.18653/v1/w19-5006

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, october 25-29, 2014, doha, qatar, A meeting of sigdat, a special interest group of the ACL* (pp. 1532–1543). ACL. https://doi.org/10.3115/v1/d14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10.18653/v1/n18-1202

Pilehvar, M. T., & Camacho-Collados, J. (2020). Embeddings in natural language processing: Theory and advances in vector representations of meaning. *Synthesis Lectures on Human Language Technologies*, *13*(4), 1–175.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., & Ananiadou, S. (2013). Distributional Semantics Resources for Biomedical Text Processing. *Aistats*, *5*, 39–44. https://github.com/spyysalo/nxml2txt

Samaranayake, N. R., Cheung, D. S., Lam, M. P., Cheung, T. T., Chui, W., Wong, I. C., & Cheung, B. M. (2014). The effectiveness of a 'do not use'list and perceptions of healthcare professionals on error-prone abbreviations. *International journal of clinical pharmacy*, *36*(5), 1000–1006.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Schulz, S., Martıénez-Costa, C., & Miñarro-Giménez, J. A. (2017). Lexical ambiguity in SNOMED CT. *2050*. http://ceur-ws.org/Vol-2050/ODLS%5C_paper%5C_9.pdf

Schütze, H. (1992). Dimensions of meaning. In R. Werner (Ed.), *Proceedings supercomputing '92, minneapolis, mn, usa, november 16-20, 1992* (pp. 787–796). IEEE Computer Society. https://doi.org/10.1109/SUPERC.1992.236684

Schwarz, C. M., Hoffmann, M., Smolle, C., Eiber, M., Stoiser, B., Pregartner, G., Kamolz, L.-P., & Sendlhofer, G. (2021). Structure, content, unsafe abbreviations, and completeness of discharge summaries: A retrospective analysis in a University Hospital in Austria. *Journal of Evaluation in Clinical Practice*, 1–9. https://doi.org/https://doi.org/10.1111/jep.13533

Segura-Bedmar, I., & Martínez, P. (2015). Pharmacovigilance through the development of text mining and natural language processing techniques. *J. Biomed. Informatics*, *58*, 288–291. https://doi.org/10.1016/j.jbi.2015.11.001

Sheppard, J. E., Weidner, L. C., Zakai, S., Fountain-Polley, S., & Williams, J. (2008). Ambiguous abbreviations: An audit of abbreviations in paediatric note keeping. *Archives of disease in childhood*, *93*(3), 204–206.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, *24*(4), 35–43. http://sites.computer.org/debull/A01DEC-CD.pdf

Skreta, M., Arbabi, A., Wang, J., & Brudno, M. (2019). Training without training data: Improving the generalizability of automated medical abbreviation disambiguation. *Machine Learning for Health Workshop, ML4H@NeurIPS 2019, Vancouver, BC, Canada, 13 December 2019*, *116*, 233–245. http://proceedings.mlr.press/v116/skreta20a.html

Skreta, M., Arbabi, A., Wang, J., Drysdale, E., Kelly, J., Singh, D., & Brudno, M. (2021). Automatically disambiguating medical acronyms with ontology-aware deep learning. *Nature Communications*, *12*(1), 1–10.

Suárez-Paniagua, V., Zavala, R. M. R., Segura-Bedmar, I., & Martínez, P. (2019). A two-stage deep learning approach for extracting entities and relationships from medical texts. *J. Biomed. Informatics*, *99*. https://doi.org/10.1016/j.jbi.2019.103285

Sun, W., Rumshisky, A., & Uzuner, Ö. (2013). Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *J. Am. Medical Informatics Assoc.*, *20*(5), 806–813. https://doi.org/10.1136/amiajnl-2013-001628

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, 2214–2218. http://www.lrec-conf.org/proceedings/lrec2012/summaries/463.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Veyseh, A. P. B., Dernoncourt, F., Nguyen, T. H., Chang, W., & Celi, L. A. (2021). Acronym identification and disambiguation shared tasks for scientific document understanding. *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*, *2831*. http://ceur-ws.org/Vol-2831/paper33.pdf

Veyseh, A. P. B., Dernoncourt, F., Tran, Q. H., & Nguyen, T. H. (2020). What does this acronym mean? introducing a new dataset for acronym identification and disambiguation (D. Scott, N. Bel, & C. Zong, Eds.), 3285–3301. https://doi.org/10.18653/v1/2020.coling-main.292

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., & Chao, L. S. (2019). Learning deep transformer models for machine translation. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 1810–1822. https://doi.org/10.18653/v1/p19-1176

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., & Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, *87*, 12–20.

Wren, J. D., & Garner, H. R. (2002). Heuristics for identification of acronym-definition patterns within text: Towards an automated construction of comprehensive acronym-definition dictionaries. *Methods of information in medicine*, *41*(05), 426–434.

Wu, P.-Y., Cheng, C.-W., Kaddi, C. D., Venugopalan, J., Hoffman, R., & Wang, M. D. (2017). Omic and electronic health record big data analytics for precision medicine. *IEEE Transactions on Biomedical Engineering*, *64*(2), 263–273. https://doi.org/10.1109/TBME.2016.2573285

Wu, S., Roberts, K., Datta, S., Du, J., Ji, Z., Si, Y., Soni, S., Wang, Q., Wei, Q., Xiang, Y., Zhao, B., & Xu, H. (2020). Deep learning in clinical natural language processing: A methodical review. *J. Am. Medical Informatics Assoc.*, *27*(3), 457–470. https://doi.org/10.1093/jamia/ocz200

Wu, Y., Denny, J. C., Rosenbloom, S. T., Miller, R. A., Giuse, D. A., Song, M., & Xu, H. (2015). A Preliminary Study of Clinical Abbreviation Disambiguation in Real Time. *Applied Clinical Informatics*, *6*(2), 364–374. https://doi.org/10.4338/ACI-2014-10-RA-0088

Wu, Y., Xu, J., Zhang, Y., & Xu, H. (2015). Clinical abbreviation disambiguation using neural word embeddings. In K. B. Cohen, D. Demner-Fushman, S. Ananiadou, & J. Tsujii (Eds.), *Proceedings of the workshop on biomedical natural language processing, bionlp@ijcnlp 2015, beijing, china, july 30, 2015* (pp. 171–176). Association for Computational Linguistics. https://doi.org/10.18653/v1/W15-3822

Wu, Y., Denny, J., Rosenbloom, S. T., Miller, R. A., Giuse, D. A., Song, M., & Xu, H. (2013). A prototype application for real-time recognition and disambiguation of clinical abbreviations. *Proceedings of the 7th international workshop on Data and text mining in biomedical informatics*, 7–8.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wu, Y., Denny, J. C., Trent Rosenbloom, S., Miller, R. A., Giuse, D. A., Wang, L., Blanquicett, C., Soysal, E., Xu, J., & Xu, H. (2017). A long journey to short abbreviations: Developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD). *Journal of the American Medical Informatics Association*, *24*(e1), e79–e86. https://doi.org/10.1093/jamia/ocw109

Xu, H., Stetson, P. D., & Friedman, C. (2007). A study of abbreviations in clinical notes. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 821–5. http://www.ncbi.nlm.nih.gov/pubmed/18693951%7B%5C%%7D0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2655910

Xu, H., Stetson, P. D., & Friedman, C. (2009). Methods for Building Sense Inventories of Abbreviations in Clinical Notes. *Journal of the American Medical Informatics Association*, *16*(1), 103–108. https://doi.org/10.1197/jamia.M2927

Xu, H., Stetson, P. D., & Friedman, C. (2012). Combining corpus-derived sense profiles with estimated frequency information to disambiguate clinical abbreviations.

AMIA 2012, American Medical Informatics Association Annual Symposium, Chicago, Illinois, USA, November 3-7, 2012. https://knowledge.amia.org/amia-55142-a2012a-1.636547/t-003-1.640625/f-001-1.640626/a-114-1.640887/a-115-1.640884

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 5754–5764. https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, *8689*, 818–833. https://doi.org/10.1007/978-3-319-10590-1%5C_53

Zhou, W., Torvik, V. I., & Smalheiser, N. R. (2006). ADAM: Another database of abbreviations in MEDLINE. *Bioinformatics*, *22*(22), 2813–2818. https://doi.org/10.1093/bioinformatics/btl480

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Zotova, E., Pablos, A. G., & Cuadros, M. (2021). Vicomtech at MEDDOPROF: automatic information extraction and disambiguation in clinical text. *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021*, *2943*, 776–787. http://ceur-ws.org/Vol-2943/meddoprof%5C_paper5.pdf

APPENDIX A

BARR2 clinical abbreviations that have more than one expansion.

Abb.	Expansion	Freq.	Total
A	adenina	1	3
	adenine	1	
	antígeno leucocitario humano a	1	
AAS	ácido acetil salicílico	1	2
	al ácido acetil salicílico	1	
AFP	alfa-fetoproteína	2	3
	alfafetoproteína	1	
ALT	alanina aminotransferasa	2	8
	alanine transaminase	2	
	alanine transferase	4	
AMA	american medical association	1	8
	antimitochondrial antibodies	1	
ANA	anticuerpo antinuclear	1	6
	anticuerpos antinucleares	1	
	antinuclear antibodies	1	
	antinuclear antibody	3	
AP	amiloide p	1	5
	anatomía patológica	1	
	atención primaria	3	
AST	aspartate aminotransferase	2	8
	aspartate and alanine aminotransferase	3	
	aspartato aminotransferasa	1	
	aspartato transaminasa	1	
	aspartato-aminotransferasa	1	
B	basófilo	1	2
	beta	1	
BCG	bacillus calmette-guerin	4	8
	bacilo de calmette y guérin	3	
	bacilo de calmette-guérin	1	
C	centígrado	37	44
	centímetro	1	

BARR2 clinical abbreviations that have more than one expansion (continued).

Abb.	Expansion	Freq.	Total
C	coding dna	1	
	coding dna sequence	1	
	cysteine	1	
	peak c	3	
C3	componente 3	1	4
	tercer componente del complemento	1	
	tercera vértebra cervical	2	
C4	componente 4	1	5
	cuarta vértebra cervical	3	
	cuarto componente del complemento	1	
Ca	calcio	7	12
	cámara anterior	2	
	cancer antigen	3	
CD10	cluster of differentiation 10	2	3
	cúmulo de diferenciación 10	1	
CD1a	cluster of differentiation 1a	2	3
	cúmulo de diferenciación 1a	1	
CD3	cluster of differentiation 3	1	4
	cúmulo de diferenciación 3	3	
CD30	cluster of differentiation 30	1	2
	cúmulo de diferenciación 30	1	
CD4	cúmulo de diferenciación 4	1	3
	linfocitos cooperadores inductivos	2	
CD8	cúmulo de diferenciación 8	2	4
	linfocitos t citotóxicos	2	
CEA	carcinoembrionary antigen	5	8
	carcinoembryonic antigen	3	
CK	citoqueratina	1	4
	creatin kinase	1	
	creatine kinase	1	
	cytokeratin	1	
CK20	citoqueratina 20	1	2
	cytokeratin 20	1	
cm	centímetro	216	217
	centimetro de agua	1	
CO	canal óptico	4	5

BARR2 clinical abbreviations that have more than one expansion (continued).

Abb.	Expansion	Freq.	Total
	cobalto	1	
CT	computerized tomography	1	2
	tomografía computerizada	1	
CV	calcificación vascular	1	3
	calcificaciones vasculares	2	
d	aspartic acid	1	4
	día	3	
ELISA	análisis de inmunoabsorción ligado a enzimas	2	3
	enzyme-linked immunosorbent assay	1	
EMG	electromiografía	2	4
	electromiograma	2	
EMLA	eutectic mixture local anaesthetics	4	5
	eutectic mixture of local anesthetics	1	
EPR	epitelio pigmentario de la retina	1	5
	epitelio pigmentario retiniano	4	
EVA	escala visual analógica	6	6
	evidencia la mejoría clínica	1	
F	fosfatasa	1	2
	french	1	
FA	autofluorescencia de fondo	1	16
	fibrilación auricular	1	
	fluorescein angiography	9	
	fosfatasa alcalina	5	
FC	fragment crystallizable	3	6
	frecuencia cardiaca	3	
FL	femtolitro	3	4
	fístula linfática	1	
FR	factor reumatoide	2	3
	french	1	
g	gramo	107	109
	guanina	1	
	guanine	1	
GGT	gama glutamil transferasa	1	15
	gamma glutamil transpeptidasa	1	
	gamma-glutamil transpeptidasa	11	

BARR2 clinical abbreviations that have more than one expansion (continued).

Abb.	Expansion	Freq.	Total
	gammaglutamiltranspeptidasa	2	
GOT	glutamic oxalic transaminase	10	11
	glutámic oxaloacetic transaminase	1	
GPT	glutamate pyruvate transaminase	11	14
	glutamato piruvato transaminasa	1	
	glutamic pyruvic transaminase	2	
h	histidine	1	93
	hora	90	
	horas	1	
	microgramo	1	
H2O	agua	9	10
	centímetro de agua	1	
Hg	mercurio	2	4
	milímetro de mercurio	2	
hiper-FA	hiperautofluorescencia de fondo	3	4
	hiperfluorescein angiography	1	
HSD	hematoma subdural	1	5
	hemorragia subaracnoidea	4	
i.v.	intravenosa	1	16
	intravenoso	15	
iv	intravenosa	1	19
	intravenoso	18	
kg	centímetro	1	82
	kilogramo	80	
	kilogramos	1	
l	leucocito	2	172
	linfocito	1	
	litro	169	
LDH	lactato-deshidrogenasa	17	23
	lactatodeshidrogenasa	6	
LH	linfoma de hodgkin	4	5
	luteinizing hormone	1	
lpm	latido por minuto	7	13
	latidos por minuto	6	
m	metro	4	10
	minuto	1	

BARR2 clinical abbreviations that have more than one expansion (continued).

Abb.	Expansion	Freq.	Total
m	monocito	2	
	malformaciones arterio-venosas	1	
	malformaciones arteriovenosas	2	
MCT	masa celular total	1	2
	medium-chain triglycerides	1	
mg	magnesio	1	400
	miligramo	399	
min	minuto	22	24
	minutos	2	
mm	milimetro	113	114
	milímetro de mercurio	1	
MTT	metatarso	1	4
	metatarsos	3	
NO	nervio óptico	7	13
	número	6	
no	nervio óptico	4	6
	número	2	
P	fósforo	5	10
	phosphorus	3	
	protein	2	
PCR	parada cardiorrespiratoria	1	30
	polymerase chain reaction	4	
	proteina c reactiva	24	
	reacción en cadena de la polimerasa	1	
PCT	porphyria cutanea tarda	1	4
	procalcitonina	3	
ppm	oxido nitrico inhalado	2	4
	pulsaciones por minuto	2	
PTH	parathyroid hormone	2	4
	paratohormona	2	
QA	quiste aracnoideo	2	3
	quistes aracnoideos	1	
QT	q time	2	6
	quimioterapia	4	
ROT	reflejo osteo-tendinoso	1	3
	reflejos osteotendinosos	2	

BARR2 clinical abbreviations that have more than one expansion (continued).

Abb.	Expansion	Freq.	Total
RVS	resistencia vascular sistémica	1	2
	respuesta viral sostenida	1	
s	seattle	1	13
	segundo	2	
	soluble	10	
SDRC	síndrome de dolor regional complejo	2	3
	síndrome doloroso regional complejo	1	
SNC	síndrome nefrótico congénito	1	2
	sistema nervioso central	1	
T	tesla	1	2
	tubular	1	
TA	temperatura	1	6
	tensión arterial	5	
TAC	tomografía axial computadorizada	1	106
	tomografia axial computarizada	96	
	tomografía axial computerizada	9	
TC	tomografia computarizada	47	48
	tomografía computerizada	1	
TG	triglicérido	1	5
	triglicéridos	4	
TGO	transaminasa glutámico oxalacética	1	2
	transaminasa glutámico-oxalacética	1	
TGP	transaminasa glutámico pirúvica	1	2
	transaminasa glutámico-pirúvica	1	
U	unidad	51	52
	unidades	1	
ul	microlitro	1	4
	unidad internacional	1	
	unidad litro	2	
VEB	virus de epstein barr	4	8
	virus de epstein-barr	4	
VHS	virus del herpes simple	1	2
	virus herpes simple	1	
VSG	velocidad de eritrosedimentación	1	19
	velocidad de sedimentacion globular	18	
VVZ	virus varicela zóster	3	4

BARR2 clinical abbreviations that have more than one expansion (continued).

Abb.	Expansion	Freq.	Total
	virus varicela-zoster	1	
μl	microgramo	1	19
	microlitro	18	

APPENDIX B

A description of clinical abbreviations and their senses distribution from the UMN data set.

Abb.	Expansion	No.	Abb.	Expansion	No.
Abb.	Expansion	No.	Abb.	Expansion	No.
AB	abortion	345	ITP	idiopathic thrombocytopenic purpura	500
	blood group in ABO system	137	IVF	in vitro fertilization	308
	type A, type B	8		intravenous fluid	188
	atrioventricular:AV	2		UNSURED SENSE	3
	X-ray finding	1		inferior vena cava:IVC	1
	UNSURED SENSE	1	LA	long-acting	426
	NAME	1		Los Angeles	40
	MISTAKE:abduction	1		left atrial	30
	arteriovenous:AV	1		Louisiana	2
	arterial blood	1		UNSURED SENSE	1
	antipyrine benzocaine	1		antinuclear antibody:ANA	1
	ankle-brachial	1	LE	leukocyte esterase	345
AC	(drug) AC	161		lower extremity	134
	acromioclavicular	158		UNSURED SENSE	5
	adriamycin cyclophosphamide	118		left ventricle:LV	5
	before meals	42		lupus erythematosus	3
	assist control	9		lymphedema	3
	acetate	4		NAME	2
	angiotensin-converting enzyme:ACE	3		long-acting:LA	2
	abdominal circumference	2		sinemet-levodopa	1
	anticoagulation	1	MOM	multiples of median	439
	antecubital	1		milk of magnesia	57
	alternating current	1		GENERAL ENGLISH	3
	ad lib on demand	407		Mall of America:MOA	1
	adrenoleukodystrophy	88		metacarpophalangeal	179

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
ALD	alanine aminotransferase:ALT	3	MP	mercaptopurine	107
	acetyl lysergic acid diethylamide	1		metatarsophalangeal	105
	left anterior descending:LAD	1		metatarsophalangeal	55
AMA	against medical advice	444		metabolic panel	12
	advanced maternal age	31		UNSURED SENSE	11
	antimitochondrial antibody	25		nurse practitioner:NP	11
ASA	acetylsalicylic acid	404		metarsophalangeal	6
	American Society of Anesthesiologists	93		(device) MP	5
	aminosalicylic acid	3		military police	4
AV	atrioventricular	374		(drug) MP	2
	arteriovenous	116		menstrual period	1
	aortic valve	8		mesangial proliferative	1
	UNSURED SENSE	2		milligram:mg	1
AVR	aortic valve replacement	381	MR	magnetic resonance	314
	augmented voltage right arm	103		mitral regurgitation	176
	aortic valve regurgitation	5		GENERAL ENGLISH	5
	aortic valve resistance	4		mental retardation	3
	rapid ventricular response:RVR	4		medical record	1
	UNSURED SENSE	2		myocardial infarction:MI	1
	auditory brainstem response:ABR	1	MSSA	modified selective severity assessment	418
BAL	bronchoalveolar lavage	457		methicillin-susceptible Staphylococcus aureus	82
	blood alcohol level	43	NAD	no acute distress	377
BK	BK (virus)	343		nothing abnormal detected	123
	below knee	157	NP	nurse practitioner	438
BM	bowel movement	459		nasopharyngeal	53
	breast milk	25		UNSURED SENSE	5

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
BM	bone marrow	14	NP	nasopharynx	2
	UNSURED SENSE	2		(drug) NP	1
BMP	basic metabolic profile	456		natriuretic peptide	1
	beta-natriuretic peptide:BNP	36	OP	oropharynx	308
	bone morphogenetic protein	7		oblique presentation/occiput posterior	121
	bone marrow transplant:BMT	1		operative	55
C&S	conjunctivae and sclerae	434		UNSURED SENSE	6
	culture and sensitivity	47		ophthalmic	5
	protein C and protein S	16		occiput posterior	3
	central nervous system:CNS	2		outpatient	1
	carcinosarcoma:CaS	1		ova and parasites	1
C3	cervical (level) 3	249	OR	operating room	466
	(complement) component 3	243		GENERAL ENGLISH	32
	propionylcarnitine	6		(drug) OR	1
	(stage) C3	2		UNSURED SENSE	1
C4	cervical (level) 4	261	OTC	over the counter	469
	(complement) component 4	231		ornithine transcarbamoylase	31
	cluster of differentiation 4:CD4	6	PA	posterior-anterior	212
	(PO Box) C4	1		pulmonary artery	138
CA	cancer	391		physician associates	83
	carbohydrate antigen	105		physician assistant	61
	California	2		UNSURED SENSE	2
	UNSURED SENSE	2		tissue plasminogen activator:TPA	2
CDI	Children's Depression Inventory	270		pulmonary auscultation	1
	center for diagnostic imaging	225		pulmonary embolus:PE	1

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
CDI	clean, dry, intact	3	PAC	premature atrial contraction	275
	UNSURED SENSE	2		physician assistant certification	137
CEA	carcinoembryonic antigen	444		post anesthesia care	47
	carotid endarterectomy	53		picture archiving communication	25
	UNSURED SENSE	1		patient-controlled analgesia:PCA	7
	cancer:CA	1		UNSURED SENSE	4
	cerebrovascular accident:CVA	1		(drug) PAC	2
CR	controlled release	453		prostate-specific antigen:PSA	1
	cardiorespiratory	28		pulmonary arterial concentration	1
	complete remission	16		pulmonary artery catheter	1
	C-reactive	1	PCP	Pneumocystis jiroveci pneumonia	294
	closed reduction	1		primary care physician	111
	creatinine	1		phencyclidine	93
CTA	clear to auscultation	396		UNSURED SENSE	1
	computed tomographic angiography	100		patient-controlled analgesia:PCA	1
	UNSURED SENSE	2	PD	peritoneal dialysis	409
	cerebellopontine angle:CPA	1		posterior descending	34
	creatine phosphokinase:CPK	1		police department	14
CVA	cerebrovascular accident	278		phosphate dehydrogenase	9
	costovertebral angle	222		pancreatic duct	8
CVP	central venous pressure	436		(device) PD	6
	cyclophosphamide, vincristine, prednisone	62		UNSURED SENSE	6
	cardiovascular pulmonary	2		(drug) PD	3
	chorionic villus sampling	457		prism diopter	3

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
CVS	cardiovascular system	41	PD	pulmonary embolus:PE	3
	customer, value, service	2		Parkinson disease	1
DC	discontinue	282		dorsalis pedis:DP	1
	direct current	152		patent ductus	1
	District of Columbia	31		personality disorder	1
	discharge	31		purified protein derivative:PPD	1
	(diltiazem) DC	1	PDA	posterior descending artery	361
	(drug) DC	1		patent ductus arteriosus	138
	deceased donor:DD	1		patient-controlled analgesia:PCA	1
	direct and consensual	1	PE	pulmonary embolus	408
DIP	distal interphalangeal	462		pressure equalization	89
	desquamative interstitial pneumonia	36		UNSURED SENSE	2
	dipropionate	2		pleural effusion	1
DM	dextromethorphan	286	PM	afternoon	423
	diabetes mellitus	209		physical medicine and rehabilitation:PMR	74
	UNSURED SENSE	3		UNSURED SENSE	2
	NAME	1		metacarpophalangeal:MP	1
	medical doctor:MD	1	PR	pr interval	252
DT	diphtheria-tetanus	336		per rectum	141
	delirium tremens	129		progesterone receptor	88
	dorsalis pedis:DP	23		pulmonary regurgitation	12
	UNSURED SENSE	4		UNSURED SENSE	4
	(drug) DT	3		(drug) PR	2
	deep vein thrombosis:DVT	3		pulse rate	1
	doppler echo:DE	1	PT	physical therapy	455
	physical therapy:PT	1		prothrombin time	22
EC	enteric-coated	439		posterior tibial	21
	enterocutaneous	45		UNSURED SENSE	1
	UNSURED SENSE	11		prothrombin	1

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
EC	epirubicin	2	RA	right atrium	394
	extensor carpi	2		rheumatoid arthritis	66
	MISTAKE:EZ PAP	1		room air	36
ER	emergency room	448		UNSURED SENSE	3
	extended release	34		retinoic acid	1
	estrogen receptor	18	RT	radiation therapy	336
ES	extra strength	469		respiratory therapy	149
	enhanced sensitivity	14		retrograde tachycardia	7
	ejection fraction:EF	8		NAME	2
	UNSURED SENSE	7		UNSURED SENSE	2
	(drug) ES	1		respiratory therapist	2
	erythrocyte sedimentation rate:ESR	1		(drug) RT	1
ET	enterostomal therapy	289		right	1
	endotracheal	200	SA	slow acting/sustained action	373
	electrophysiology:EP	6		sinuatrial	88
	UNSURED SENSE	1		UNSURED SENSE	29
	elective termination	1		saturation	4
	electroconvulsive therapy:ECT	1		MISTAKE:Oncotype DX	2
	enterocutaneous:EC	1		sinus arrest	2
	pressure equalization:PE	1		American Society of Anesthesiologists:ASA	1
FSH	follicle-stimulating hormone	265		methicillin-susceptible Staphylococcus aureus	1
	Fairview Southdale Hospital	231	SBP	spontaneous bacterial peritonitis	417
	fascioscapulohumeral muscular dystrophy	4		systolic blood pressure	83
GT	gastrostomy tube	446	SMA	superior mesenteric artery	353
	glutamyl transpeptidase	30		sequential multiple autoanalyzer	84
	gutta	16		spinal muscular atrophy	56

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
GT	gamma-glutamyltransferase:GGT	5		smooth muscle antibody	3
	glucose tolerance	2		UNSURED SENSE	2
	guttae:GGT	1		smooth muscle actin	2
IA	(stage) IA	275	SS	single strength	439
	intraarterial	176		UNSURED SENSE	57
	Iowa	19		sickle cell genotype SS	4
	(grade) IA	11	T1	tumor stage 1	198
	(status) IA	5		thoracic (level) 1	194
	(type) IA	5		T1 (MRI)	103
	UNSURED SENSE	4		UNSURED SENSE	3
	(class) IA	3		term 1	1
	transient ischemic attack:TIA	2		type 1 (diabetes mellitus)	1
IB	(stage) IB	472	T2	T2 (MRI)	227
	(grade) IB	8		tumor stage 2	166
	(status) IB	8		thoracic (level) 2	97
	international baccalaureate	5		UNSURED SENSE	7
	(cycle) IB	2		S2 (heart sound):S2	1
	(type) IB	2		T2 Nodes	1
	UNSURED SENSE	1		term 2	1
	interferon beta	1	T3	triiodothyronine	268
	intravenous:IV	1		tumor stage 3	156
IM	intramuscular	461		thoracic (level) 3	65
	intramedullary	38		UNSURED SENSE	5
	UNSURED SENSE	1		T3 (ECG pattern)	4
IR	interventional radiology	394		term 3	2
	immediate-release	102	T4	thyroxine	424
	internal rotation	2		thoracic (level) 4	41
	UNSURED SENSE	1		tumor stage 4	35
	infrared	1	US	United States	402
IT	GENERAL ENGLISH	225		ultrasound	94
	information technology	103		GENERAL ENGLISH	3
	intrathecal	58		UNSURED SENSE	1

A description of clinical abbreviations and their senses distribution from the UMN data set(continued).

Abb.	Expansion	No.	Abb.	Expansion	No.
IT	ischial tuberosity	48		vincristine adriamycin and dexamethasone	396
	iliotibial	40	VAD	ventricular assist device	87
	intertrochanteric	14		vascular access device	13
	UNSURED SENSE	6		UNSURED SENSE	3
	(drug) IT	2		video-assisted thoracic surgery:VATS	1
	idiopathic thrombocytopenic purpura:ITP	1	VBG	vertical banded gastroplasty	299
	immature-to-total neutrophil	1		venous blood gas	201
	inspiratory time	1	FISH	fluorescent in situ hybridization	449
	pravastatin evaluation and infection therapy	1		GENERAL ENGLISH	51

APPENDIX C

The top ten most frequent acronyms founded in the SCiAD data set.

Acronym	Expansion	No.	Total
CNN	citation nearest neighbour	13	2,929
	complicated neural networks	1	
	condensed nearest neighbor	33	
	convolutional neural network	2,579	
RNN	random neural networks	305	1,370
	recurrent neural network	1,064	
	reverse nearest neighbour	1	
FEC	federal election candidate	2	1,038
	forward error correction	1,036	
RL	reinforcement learning	822	936
	relative location	31	
	representation learning	4	
	resource limitations	4	
	restrained lloyd	41	
	robot learning	15	
	robust locomotion	19	
CT	class table	15	878
	computed tomography	842	
	conditional training	1	
	confidential transactions	2	
	constraint theory	14	
	contributor trust	2	
	coordinated turn	1	
	crowd trust	1	
ML	machine learning	475	589
	malware landscape	11	
	maximum likelihood	76	
	model logic	22	
	mortar luminance	5	
GP	gaussian process	418	499
	geometric programming	81	

The top ten most frequent acronyms founded in the SCiAD data set (continued).

Acronym	Expansion	No.	Total
IP	image preprocessing	2	481
	inductive programming	32	
	integer programming	8	
	intellectual property	364	
	intercept probability	22	
	internet protocol	52	
	inverse proportion	1	
DL	deep learning	229	473
	depth loss	8	
	description length	8	
	description logics	93	
	dice loss	1	
	distributed ledger	33	
	dogleg	10	
	downlink	91	
RF	radio frequency	153	469
	random forest	298	
	register file	3	
	regression forest	8	
	regression function	7	