Cardiff University

School of Computer Science and Informatics

Deep Learning for Clinical Texts in Low-Data Regimes

Submitted in Partial Fulfillment of the Requirement for the Degree of Doctor of Philosophy

Daphné Anaïs Marylin Chopard

Supervisors:
Dr. Matthias Treder
Dr. Padraig Corcoran
Prof. Irena Spasić

March, 2023

Abstract

Electronic health records contain a wealth of valuable information for improving healthcare. There are, however, challenges associated with clinical text that prevent computers from maximising the utility of such information. While deep learning (DL) has emerged as a practical paradigm for dealing with the complexities of natural language, applying this class of machine learning algorithms to clinical text raises several research questions. First, we tackled the problem of data sparsity by looking into the task of adverse event detection. As these events are rare, examples thereof are lacking. To compensate for data scarcity, we leveraged large pre-trained language models (LMs) in combination with formally represented medical knowledge. We demonstrated that such a combination exhibits remarkable generalisation abilities despite the low availability of data. Second, we focused on the omnipresence of short forms in clinical texts. This typically leads to out-of-vocabulary problems, which motivates unlocking the underlying words. The novelty of our approach lies in its capacity to learn how to automatically expand short forms without resorting to external resources. Third, we investigated data augmentation to address the issue of data scarcity at its core. To the best of our knowledge, we were one of the firsts to investigate population-based augmentation for scheduling text data augmentation. Interestingly, little improvement was seen in fine-tuning large pre-trained LMs with the augmented data. We suggest that, as LMs proved able to cope well with small datasets, the need for data augmentation was made redundant. We conclude that DL approaches to clinical text mining should be developed by fine-tuning large LMs. One area where such models may struggle is the use of clinical short forms. Our method to automating their expansion fixes this issue. Together, these two approaches provide a blueprint for successfully developing DL approaches to clinical text mining in low-data regimes.

Acknowledgements

First and foremost, I would like to thank my supervisors, Prof Irena Spasić, Dr Matthias Treder, and Dr Padraig Corcoran for their guidance throughout this thesis. Irena, I'm extremely grateful that you gave me the opportunity to pursue a PhD under your supervision. Thank you for your flexibility, your understanding, and your invaluable help at every stage of this thesis. Matthias, thank you for your precious technical insights. And Padraig, thank you for accepting to join the supervision of this thesis at a later date.

I want to thank my partner Julien for his unwavering support through the ups and down. This thesis would not have been possible without his limitless patience, his invaluable insights, his unconditional love, and the countless great moments shared. I am also grateful for the support of my family and friends.

I also would like to thank my colleagues at Cardiff University for making these 4 years such a pleasant experience. A special thank you to Anastazia for some of the best memories I have of Cardiff. This PhD would not have been the same without you.

Lastly, I would like to express my gratitude to the School of Computer Science and Informatics at Cardiff University, who generously granted me a 3-year scholarship for pursuing my PhD.

Contents

List of Figures

List of Tables

List of Abbreviations

Adam adaptive moment estimation

ANN artificial neural network

BERT Bidirectional Encoder Representations from Transformers

biLM bidirectional language model

BLUE Biomedical Language Understanding Evaluation

BOW bag-of-words

CBOW continuous bag-of-words

char-CNN character-level convolutional neural network

CNN convolutional neural network

CRF conditional random field

CTR Centre for Trials Research

CTU clinical trial unit

CUI concept unique identifier

CV cross-validation

DL deep learning

DNN deep neural network

EDA easy data augmentation

EHR electronic health record

ELECTRA Efficiently Learning an Encoder that Classifies Token Replacements Accurately

ELMo Embeddings from Language Model

FDA Food and Drug Administration

FN False Negative

FP False Positive

GloVe Global Vectors

GLUE General Language Understanding Evaluation

GPT Generative Pre-Training *or* Generation from Pre-trained Transformers

GPU graphics processing unit

GRU gated recurrent unit

LM language model

LSTM long short-term memory

MLM masked language model

MLP multilayer perceptron

MNLI Multi-Genre Natural Language Inference Corpus

NER named-entity recognition

NHS National Health Service

NLI natural language inference

NLP natural language processing

NLU natural language understanding

NMT neural machine translation

NN neural network

NSP next sentence prediction

OOV out-of-vocabulary

P precision

PBA population-based augmentation

PBT population-based training

POS part-of-speech

PR precision-recall

QA question answering

R recall

RQ research question

RNN recurrent neural network

RO research objective

RoBERTa Robustly optimized Bidirectional Encoder Representations from Transformers (BERT) pre-training Approach

SAE serious adverse event

SBE Surrounding Based Embedding

SD standard deviation

Seq2Seq sequence-to-sequence

SGD stochastic gradient descent

SST Stanford Sentiment Treebank

TF-IDF term frequency-inverse document frequency

TN True Negative

TP True Positive

UAD Unsupervised Abbreviation Disambiguation

UK United Kingdom

ULMFiT Universal Language Model Fine-Tuning

UMLS Unified Medical Language System

US United States

WMD Word Mover's Distance

WSD word sense disambiguation

Chapter 1

Introduction

The deep learning (DL) revolution, which has happened in recent years (LeCun et al., 2015, Schmidhuber, 2015), has enabled rapid advances in many fields of computer science, especially in areas such as computer vision (Lin et al., 2014, Russakovsky et al., 2015), audio signal processing (Lee et al., 2009, Mohamed et al., 2011, Hinton et al., 2012) and natural language processing (NLP) (Chen et al., 2017, Yadav & Bethard, 2018, Zhang et al., 2018, Reddy et al., 2019). In particular, DL has proven to be a powerful modelling technique for supervised learning (LeCun et al., 2015)—a machine learning task that involves discovering the relationship between input and output based on a set of example pairs.

Unlike classical machine learning algorithms, which rely on a set of manually designed features for learning a mapping (Bengio et al., 2013), DL models can directly discover suitable representations from raw high-dimensional data. This capacity to learn implicit features in an end-to-end fashion during training makes these models more effective than their classical counterparts, as extracting a good set of high-level representations by hand is not always easy (LeCun et al., 2015).

Although the theoretical backbone of DL has been around for a long time, much of the recent breakthroughs were made possible by two essential developments (Sun et al., 2017)—the introduction of cheaper high-speed graphics processing units (GPUs) and the creation of large annotated datasets—which have alleviated the issues associated with the high number of parameters in DL models. Indeed, because they are highly parameterised, DL models are not only computationally expensive to train—lots of operations are required to update the many parameters, but they also have to rely on a large enough number of labelled examples to properly model the data. This characteristic has thus led to both a computational and a performance bottleneck. The former bottleneck has been overcome thanks to GPUs whose capacity to handle multiple computations simultaneously was successfully re-purposed to speed up the training and

has allowed scaling to large networks. At the same time, substantial work has been done to collect a considerable number of samples and annotate them, resulting in large reference datasets—such as ImageNet (Deng et al., 2009), which reduced the latter bottleneck and fostered the development of different DL models. Overall, these two factors have made possible the development and the training in an end-to-end fashion of large and powerful models and thereby lie at the root of subsequent success in DL research.

Over the past decade, the digital transformation happening in many aspects of modern life has led to an evergrowing amount of digitised data. However, in practice, most of these data are not readily available for training DL models (Bansal et al., 2021). In particular, in order to train models in supervised settings, input samples have to be explicitly labelled with the desired output, which requires added manual effort. Unfortunately, while deep neural networks (DNNs) offer a powerful solution to many supervised tasks, their success largely depends on the number of labelled examples available for training (LeCun et al., 2015).

One of the domains which generate large amounts of data daily is healthcare. In the United Kingdom (UK), for example, the National Health Service (NHS) has been putting a great deal of effort in recent years to slowly abandon the use of paper-based records for electronic ones to deliver more efficient and better healthcare (Klovig Skelton, 2022). In 2012 already, Metzger et al. (2012) reported that proportions of electronic health records (EHRs) compared to paper-based ones among the countries they surveyed ranged from 37% (Canada) to 97% (New Zealand). These numbers are most likely much higher nowadays. Such a worldwide effort has led to a high volume of digitised unstructured texts on top of more structured data—for instance, Swedish EHRs contain about 40% of free text (Dalianis et al., 2009). By nature, these unstructured texts contain rich patient information whose subtleties cannot be easily captured in rigidly structured machine-readable formats. Despite this, clinical texts remain less often exploited than the structured data of EHRs (Wang et al., 2018d) due to a lack of appropriate data processing tools (Dalianis, 2018). Yet the invaluable information they contain could be leveraged for diverse high-impact applications such as improving diagnosis, treatment, and prognostic.

This thesis focuses on DL for clinical text mining. Indeed, most of the information extraction tasks which lie at the heart of clinical text mining are well suited for supervised DL—whether it be information extraction within the text itself (such as named-entity recognition (NER) and relation extraction) or directly from the text as a whole (such as text classification) (Percha, 2021). Unfortunately, the data scarcity issue mentioned above is exacerbated in the clinical domain. A systematic literature review has revealed

that the size of clinical text datasets tends to be relatively small (Spasić et al., 2020). This problem mainly stems from the annotation bottleneck, as ground-truth labels are both time-consuming and expensive to obtain and often require the input of domain experts. Privacy concern is another factor that limits the creation of sizeable clinical text datasets. Since healthcare data contain sensitive information, they cannot easily be shared with external entities; every precaution must be taken to ensure that the data are adequately anonymised or deidentified before being used for machine learning applications. Moreover, privacy concerns have prevented the widespread use of crowdsourcing for annotating clinical data (Spasić et al., 2020), even though a study has shown that the annotations of untrained crowd workers—albeit less accurate than those of medical experts—could be satisfactory enough for model development (Cocos et al., 2017b).

Aside from the data scarcity issue, clinical text data contain domain-specific language that differs from the regular one. This means that NLP approaches developed for ordinary text often fail when applied to the medical domain (Ferraro et al., 2013). On the one hand, as medical notes act as a means of communication between healthcare professionals, they contain elements that are idiosyncratic to clinicians and institutions and are not easily accessible to other people, let alone machines (Lerner et al., 2000, Chapman et al., 2011). On the other hand, because clinical texts are written under considerable time pressure, they often lack clear structure. They contain incomplete sentences that not only omit certain words but are also filled with liberal word forms such as misspellings and non-standard abbreviations (Meystre et al., 2008, Leaman et al., 2015, Dalianis, 2018). As a result, clinical data can be very ambiguous from a computational point of view and difficult to process for models developed for non-medical applications.

1.1 Motivation

Data scarcity is one of the key obstacles to the development of DL models for clinical text mining, especially in the supervised setting. Spasić et al. (2020) have reported that some studies limit the amount of training data to as little as 0.002% of the available data precisely because of the annotation bottleneck. This is a critical issue since, as mentioned earlier, DL models rely on a large amount of training data to extract meaningful patterns that generalise well to unseen data. Indeed, as DNNs are highly parameterised, they sometimes manage to learn underlying noise patterns in the training data. As a result, they might achieve high performance on the training data, but that performance might drop when processing new examples. This effect is called overfitting and will be discussed more in Section 2.5.2.

Nevertheless, the annotation bottleneck and the privacy issues mentioned above are not the only factors preventing the creation of large clinical text datasets. Indeed, clinical applications are often concerned with rare occurrences, which means that the number of existing data related to the target of interest is by nature lower than those that are not. This characteristic further exacerbates the issue of small datasets. For example, when the goal of a task is to detect patients with a given condition, it can be expected that there exist fewer records of people diagnosed with this condition than the opposite. Similarly, when identifying rare events such as medication errors in clinical reports, there are likely fewer samples containing medication errors available than the other way around. This problem is not uncommon in the context of classification tasks, where a large fraction of instances might belong to a small subset of classes; it is referred to as the class imbalance problem (Rahman & Davis, 2013) and is typically addressed with re-sampling techniques—namely undersampling or oversampling—which rebalance the data distribution (Branco et al., 2016). However, these approaches are not always suitable for small datasets. In its simplest form, undersampling consists in removing samples from the majority class to match the number of examples of the minority class (He & Ma, 2013). If the size of the training dataset is small, this further reduces the number of training examples and makes it even harder for DL models to learn complex patterns. In contrast, oversampling—in its naive form—duplicates examples from the minority class (Yap et al., 2014), which, in the clinical setting, is often the class of interest. Yet, when the dataset is already small to begin with, the minority class does not only contain fewer samples by nature but also contains a small number of examples in absolute. As a result, oversampling often leads to overfitting since the model is likely to learn the specifics of the oversampled samples rather than the underlying patterns (Kotsiantis et al., 2006).

In addition to the smallness of clinical datasets, another element to consider is the clinical sublanguage, whose characteristics differ substantially from those of general English (Shao et al., 2020). In fact, the language used in clinical narratives is thought to constitute a distinct sublanguage with specific linguistic properties (Friedman et al., 2002), as based on the theory of sublanguages introduced by Harris (1991). Even within the clinical field, there exist different sublanguages depending on the medical discipline and the nature of the text (Patterson & Hurdle, 2011). One of the clinical sublanguage's specific properties is that it often omits information that can be easily inferred from the context (Shao et al., 2020). Overall, unlike ordinary texts, clinical texts tend to condense high-value information into a small number of characters (Leaman et al., 2015). This can be explained by the time constraints placed on healthcare professionals and the repetitive nature of the reporting task. For instance, the mention *no new lesion*

noted in a chest radiology report does not include the information that this observation was made by the radiologist nor that it was made about lung lesions. Thus, some information might not be explicitly contained within the text and must be unlocked by NLP approaches. In addition, omissions can lead to sentences that are grammatically incorrect as they are missing basic elements. For example, the subject *the patient* and the verb *has* can be left out of the sentence *The patient has lower back pain* without any loss of information as these two elements can easily be deduced from context. Allvin et al. (2011) noted that, in Finnish and Swedish clinical narratives, almost no sentence contained a subject and the verb was missing in about half of the sentences. This lack of grammatical structure can be an obstacle to the development of NLP algorithms, which are trained on and optimised for general language (e.g. (Mehrabi et al., 2015)). Similarly, characters are also often omitted from words to form abbreviations which are frequently used in clinical notes. The problem is that these abbreviations are often created ad hoc without following strict rules and, although their full form can often be inferred from the context, they are difficult to interpret without additional processing by NLP algorithms. In addition, abbreviations can be highly ambiguous (Holper et al., 2020): Liu et al. (2001) reported that about a third of the abbreviations in English clinical text could refer to multiple full forms. This means that short forms cannot be easily resolved to their full form with a simple mapping and that more elaborate strategies are needed to unlock the meaning hidden behind them.

Another feature of the clinical sublanguage is the idiosyncratic terminology it often adopts (Shao et al., 2020). Clinical text data contain many domain-specific terms, with each clinical subdomain using its own jargon (Patterson & Hurdle, 2011, Dalianis, 2018). In addition, there can exist different written forms for a unique medical notion (Lu et al., 2019). For instance, the terms "myocardial infarction", "heart attack" and "cardiovascular stroke" all refer to the same concept. This characteristic increases the difficulty of developing machine learning approaches for clinical text mining, as a concept could appear in a different form in the training examples than in the unseen data. In contrast, some words can mean other things when appearing in the general domain versus the clinical one. For example, the term *tab* has different meanings in ordinary English—such as a small object that is attached to something or a bill—than in the clinical setting, where it generally refers to the word *tablet* and is used to indicate medication dosage.

Further, the clinical sublanguage features the prevalence of compositionality, namely multi-word phrases that refer to domain-specific concepts such as *blood glucose* (Friedman et al., 2002). Because multi-word expressions are commonly used to express medical notions (He, 2016), they often end up being replaced by acronyms in written reports

(Friedman et al., 2002). However, when used, acronyms (e.g. 'HIV') obscure the corresponding multi-word phrase (e.g. 'human immunodeficiency virus') preventing individual words (e.g. 'virus') from being retrieved (Filimonov et al., 2022). As a result, NLP algorithms must be able to correctly interpret multi-word terms and harness the information hidden behind the corresponding acronym.

1.2 Research Hypothesis, Questions and Objectives

The main hypothesis of this thesis is that DL can successfully be applied for clinical text mining. As indicated earlier, some of the issues that make this problem challenging are (1) medical concept heterogeneity, (2) short-form expressions such as abbreviations and acronyms, and (3) small datasets and limited annotations. To tackle these issues, we suggest multiple strategies. The first strategy is to interpret data using a general language model (LM) pre-trained on a large dataset. The second strategy consists of interpreting data using domain-specific knowledge such as ontologies. The third strategy is to normalise data, that is to disambiguate short forms to unlock the meaning hidden behind them. Lastly, the fourth strategy consists of augmenting data with synthetic samples.

Our hypothesis along with the issues highlighted above naturally leads us to the three different research questions (RQs), one for each issue named above, which are detailed in the three subsections below.

1.2.1 Identification of Rare Events

The first RQ is the following:

RQ1. Can an effective DL strategy be developed to recognise references to rare clinical events?

Rare events are defined as events that occur much less frequently than non-event ones (King & Zeng, 2001). As mentioned earlier, a problem that is inherent to rare events is the lack of training data as, per definition, data that contain rare events are much less frequent than those that do not. To answer this question, we establish the following research objective (RO):

RO1/a. Using serious adverse events (SAEs) as a case study for rare events, combine the implicit semantic knowledge captured by pre-trained LMs with explicit domain

knowledge represented by an ontology to compensate for the lack of training data of sufficient size and diversity and map different expressions to single concepts.

An adverse event is defined by the United States (US) Food and Drug Administration (FDA) as "any undesirable experience associated with the use of a medical product in a patient" (US Food and Drug Administration, 2016). It is further qualified as "serious" if it results in a grave outcome such as death, hospitalisation or disability. By nature, SAEs are quintessential rare events in the medical domain but identifying them in clinical texts can be challenging. Indeed, adverse events cannot easily be extracted from text without taking the context into account to determine whether the event is caused by a medical product or is the consequence of an underlying condition for which the medical product was taken. Obviously, given the nature of SAEs, there is always a limited number of samples that contain such events, making this problem even more difficult for machine learning approaches. Moreover, adverse events—just like other medical terms—can be expressed in different ways. This means that additional consideration is needed to map every expression to unique concepts for large-scale analysis.

To tackle the detection of SAEs in clinical trial narratives, we choose to combine the first two strategies suggested above, namely to interpret data using both the implicit knowledge of a pre-trained LM and the explicit medical knowledge formally modelled by an ontology. We know that through the actions of pre-training and fine-tuning, it is possible to transfer knowledge from a large corpus to a downstream task for which the available data are limited (Vrbančič & Podgorelec, 2020). In particular, a pre-trained LM (i.e. a model that has been pre-trained on a LMling task) can provide an understanding of language (e.g. hierarchical relations, long-term dependencies or sentiment) which has been captured during pre-training (Howard & Ruder, 2018). To complement the implicit semantics contained in the continuous vector space and encapsulated by pre-trained LMs, we propose incorporating explicit semantics (through relationships between concepts) in a graph representation of a domain using the Unified Medical Language System (UMLS).

1.2.2 Short Forms: Abbreviations and Acronyms

The second RQ that arises from our main hypothesis and the issues identified above is as follows:

RQ2. Can an effective DL strategy be developed to normalise clinical text by automatically expanding short forms?

The use of abbreviations and acronyms is frequent in clinical text data and helps healthcare professionals cut down on time and ease a sometimes repetitive task. However, as mentioned earlier, acronyms hide the compositional meaning of multi-word expressions from NLP algorithms (Spasić, 2018) and abbreviations occlude the signification of the words they replace.

Indeed, the way most NLP models process text is by first mapping every input token to a representation that depends on a pre-defined vocabulary. For example, for traditional word embeddings, this vocabulary is comprised of every token in the training data, whereas for contextual word embeddings vocabularies are fixed and do not generally depend on the available data. Relying on a vocabulary to build word representation mainly helps avoid the curse of dimensionality that results from using very high dimensional representations.

However, this means that sometimes models will encounter words that are not contained in the vocabulary and therefore do not have a straightforward representation; such words are referred to as out-of-vocabulary (OOV) words. Most abbreviations, because they are often created ad hoc to save time and can vary depending on the person using them, are treated as OOV. Similarly, only frequent acronyms—such as *DNA* for example—can be considered as part of general English—acronyms in some cases have superseded their multi-word term counterpart—and could potentially be part of pre-defined vocabularies. In most cases, however, acronyms are also considered OOV words.

There exist multiple strategies to deal with OOV words. The most straightforward one consists in simply ignoring these words. Although this is a plausible solution in some cases—if the word is not common enough to be in the vocabulary, then it is less likely to be relevant to the model, in the case of clinical abbreviations and acronyms, we know that these words are most likely carrying important information as we know that unimportant words are often omitted in the clinical sublanguage. Another simple strategy is to add an extra word to the vocabulary and use that exact word to represent all OOV words, but, once again, this results in the loss of the information that is behind the short form. Alternatively, a separate word—with a random representation—can be attributed to each new OOV word. Even though this allows for representations that can be interpreted by the model differently for each word, it is unlikely to reflect the underlying meaning. All these approaches are commonly used with non-contextualised embeddings.

More recent models, such as BERT, rely on a vocabulary that contains both frequent words, subwords and individual characters and try to overcome that problem by breaking

OOV words into subwords or characters that are part of the vocabulary. This strategy can be very powerful; for example, if the form *reading* is not present in the BERT vocabulary but both the word *read* and suffix *##ing* are, the BERT tokenizer will separate it into the subwords *read* and *##ing*. This can help the model interpret this form as a variant of *read* that might be related to other forms that end with *-ing*. However, since abbreviations and acronyms are formed in a completely different way than full words, subwords can often not capture their full form and unlock their meaning. Alternatively, since LMs are pre-trained on the masked language modelling task, abbreviations could be replaced by a *[MASK]* token, and the model could be tasked with predicting its full form. However, this approach is impractical for the clinical sublanguage since many words are OOV and less essential words are omitted. In addition, this cannot be used to find the full form of acronyms, as the model only predicts a token for each *[MASK]*, whereas acronyms typically refer to multi-word expressions. Indeed, masked LMs learn to reconstruct a masked input sequence during pre-training and can rely on the full input to do so. As a result, they excel at classification tasks and are not well-suited for generating texts. In contrast, autoregressive LMs, which are pre-trained differently—namely in predicting the next token in a sequence (i.e. the classic language modelling task)—perform well on text generation tasks because of this very difference.

Another issue inherent to short forms is that they can be extremely ambiguous Liu et al. (2001), Li et al. (2015). For example, the abbreviation *cl* is used to refer to either the word *chlorine* or *clearance*, whereas the acronym *ms* is common for replacing the expressions *multiple sclerosis*, *mitral stenosis* or *myasthenic syndrome*. Consequently, by treating each short form as a separate word, we lose the rich information that is hidden behind it. That is why we suggest disambiguating abbreviations and acronyms and retrieving their full form prior to applying any DL algorithms.

This discussion highlights the need for a new approach to deal with this particular kind of OOV words and the ambiguity problem. However, as abbreviations and acronyms are almost never used with their definition in clinical text data and new abbreviations or acronyms can be coined at any time, there is a lack of annotated data to tackle that problem.

As a result, we identify two ROs to tackle this question:

RO2/a. Develop an approach to automatically expand abbreviations (i.e. short forms that expand to a one-word full form) to their full form.

RO2/b. Develop an approach to disambiguate acronyms automatically, that is, map them to the corresponding full form, which is a multi-word term.

1.2.3 Small Datasets and Limited Annotations

Up to this point, we have suggested combining pre-trained LMs—which capture language semantics—with explicit domain knowledge to help models interpret clinical text data when available data are limited. Similarly, we have given motivation for unlocking the evidence hidden behind short forms through disambiguation. We believe that these two elements can contribute to overcoming some of the challenges associated with applying DL to clinical text data. Nevertheless, it has been shown that, up to a certain point, the performance of DL models increases with the number of samples (Sun et al., 2017). Therefore we identify the following RQ:

RQ3. Can a label-preserving transformation of existing data improve the performance of DL approaches to text mining that are based on pre-trained LMs?

Applying label-preserving transformations to training data is a practical way of creating synthetic samples from real ones, a process called *data augmentation*. This approach has been very successful in the field of computer vision but has failed so far to become popular in NLP. In particular, very little research has been carried out to evaluate the effect of text data augmentation in conjunction with pre-trained LMs. Yet, when the amount of available data is limited, data augmentation helps increase the number of samples and can enable the use of more complex models, such as DL models, that must be trained on large datasets. As the data augmentation transforms that help at the beginning of training might be different than those that boost performance towards the end of the learning process, it is important to look at schedules of data augmentation operations (i.e. sequences of transformations applied at each epoch) rather than reyling on a fixed set of transforms throughout training.

To respond to this third RQ we define the two following objectives:

RO3/a. Develop a set of label-preserving transformations for text data.

RO3/b. Develop an approach that automatically chooses the best schedule of transformations for a given task and dataset.

1.3 Research Contributions

One of the challenges of clinical text data that we have identified is the lack of large annotated datasets. This issue is exacerbated in the context of rare events since these are, by nature infrequent. In addition, medical concepts can be expressed in different forms, which adds to the complexity of detecting patterns. Using the identification of SAEs as a case study for rare events, we propose combining the implicit semantic knowledge captured by the pre-trained LM BERT with the explicit domain knowledge of the UMLS to compensate for the lack of training data of sufficient size and diversity and the heterogeneity of medical concepts (Chopard et al., 2021a). This approach is presented in Chapter 3. Recent works view SAE detection as an NER task and tackle it using BERT. While this approach achieves an F1-score of 78.35% on our dataset, this performance drops to 63.35% when trying to map the detected events to unique concepts using the UMLS. In contrast, we are the first to introduce a method that directly uses the explicit knowledge domain to interpret the data by first extracting UMLS concepts with MetaMap before classifying them with BERT. Our approach reaches an F1-score of 80.80%. Note that this score is limited by the performance of MetaMap, which sometimes fails to identify all UMLS concepts. These results confirm that automated coding of adverse events described in the narrative section of SAE reports is feasible.

Another challenge of clinical text data—which can be traced back to the nature of the clinical sublanguage—is the prevalence of short forms which are often treated as OOV words by default resulting in the loss of important information that is hidden behind. However, because these short forms can be highly ambiguous, unlocking that information is not easy. Because abbreviations are often created ad hoc and can vary from the person using them, it is not suitable relying on a fixed abbreviation dictionary for disambiguation. Instead, we introduce a novel approach which first uses DL to model the structure of abbreviations and automatically find full-form candidates (Chopard & Spasić, 2019). Then, we propose using the Word Mover's Distance (WMD) to disambiguate between all candidates. This is, to the best of our knowledge, the first time WMD is used to that effect. The whole method is explained in detail in Chapter 4. The Siamese neural network (NN) developed as a novel way to extract full-form candidates achieves an F1-score of 78.57%, which is 3.36% more than the rule-based method against which it is compared. However, it offers a much higher recall (84.04% instead of 64.53%), meaning that the true full form is more likely to be extracted as a candidate when using our approach. Our suggestion to use the WMD to disambiguate between abbreviation candidates leads to F1-scores as high as 96.36%, corresponding to an improvement of between 4.3% and 57.4% in all metrics compared to the best of the

four benchmarks.

Unlike abbreviations—which stem from a single word from which letters have been omitted or replaced with other characters—acronyms refer to multiple words. They often result from the frequent use of multi-word terms in the clinical sublanguage. This difference motivates a different approach for this latter kind of short form. Chapter 5 describes this novel method which can create large training datasets for acronym disambiguation by automatically modifying scientific abstracts so as to simulate the usage of global acronyms. This chapter also shows how the created datasets can be used in a second phase to train a supervised approach for word sense disambiguation of acronyms (Filimonov et al., 2022). In particular, the model that we developed for acronym disambiguation achieved an F1-score of 94.7% (as opposed to 60.7% for the naive frequency baseline) when trained on data that were simulated and annotated using our approach.

The success of DL models often depends on having a large enough set of training data. Therefore the question remains whether using strategies such as those suggested above is enough or whether creating synthetic data from existing ones can further help the development of DL algorithms on clinical text data. To this end, we looked at the effect of searching for a schedule label-preserving transformations on well-known language understanding tasks (Chopard et al., 2021b). This study, which is the first attempt to find augmentation schedules for text data automatically, is presented in Chapter 6. The findings of this work reveal that combining data augmentation transforms with a pre-trained LM yields very inconsistent improvement of performance if any. We hypothesise that the language understanding captured by pre-trained LMs is powerful enough to solve tasks with limited data and that augmenting the data with label-preserving transforms is too noisy to be beneficial.

A summarised list of all contributions can be found as follows:

- **Chopard, D.** and Spasić, I. A deep learning approach to self-expansion of abbreviations based on morphology and context distance. In *International Conference on Statistical Language and Speech Processing*, pp. 71–82. Springer, 2019.

- **Chopard, D.**, Treder, M. S., Corcoran, P., Ahmed, N., Johnson, C., Busse, M., and Spasić, I. Text mining of adverse events in clinical trials: Deep learning approach. *JMIR Medical Informatics*, 9(12):e28632, 2021a.

- **Chopard, D.**, Treder, M. S., and Spasić, I. Learning data augmentation schedules for natural language processing. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pp. 89–102, 2021b

- Filimonov, M., **Chopard, D.**, and Spasić, I. Simulation and annotation of global acronyms. *Bioinformatics*, 38(11):3136–3138, 2022

1.4 Thesis Structure

The content of this thesis is summarised in Figure 1.1. The chapters in the remainder of this thesis are organised as follows:

- **Chapter 2** introduces the foundations of DL for text data and, more specifically, for clinical text data in order to help the understanding of lower-level technical details in subsequent chapters.
- **Chapter 3** tackles SAE detection as a case study for rare occurrences. After a review of related work on this task, the use of the transformer architecture BERT and the UMLS for the interpretation of data from clinical trials is investigated. This study shows the importance of combining pre-trained LMs and explicit domain knowledge when clinical text data are very limited.
- **Chapter 4** reviews existing methods for expanding abbreviations—i.e. short forms that refer to a single word—a crucial task when dealing with clinical text data where characters are often omitted from words for efficiency. In addition, it introduces to that effect a novel method that first uses a Siamese network to learn abbreviation formation patterns before disambiguating between expansion candidates using the WMD.
- **Chapter 5** introduces an approach that extracts acronyms from scientific abstracts and builds on that to create an annotated dataset that simulates global acronyms similar to clinical text data, where acronyms are frequent and rarely explicitly defined. This chapter also presents a supervised method which leverages pre-trained LMs to successfully disambiguate between acronyms, using the simulated abstracts for training as a way to unlock the information hidden behind.
- **Chapter 6** investigates the use of transformative data augmentation on the performance of pre-trained LMs for common language understanding tasks. More precisely, a method is used to automatically discover schedules of text augmentation and draw conclusions on the relevance of data augmentation for fine-tuning pre-trained LMs.
- **Chapter 7** presents the conclusions of this thesis and offers avenues of research for future works.

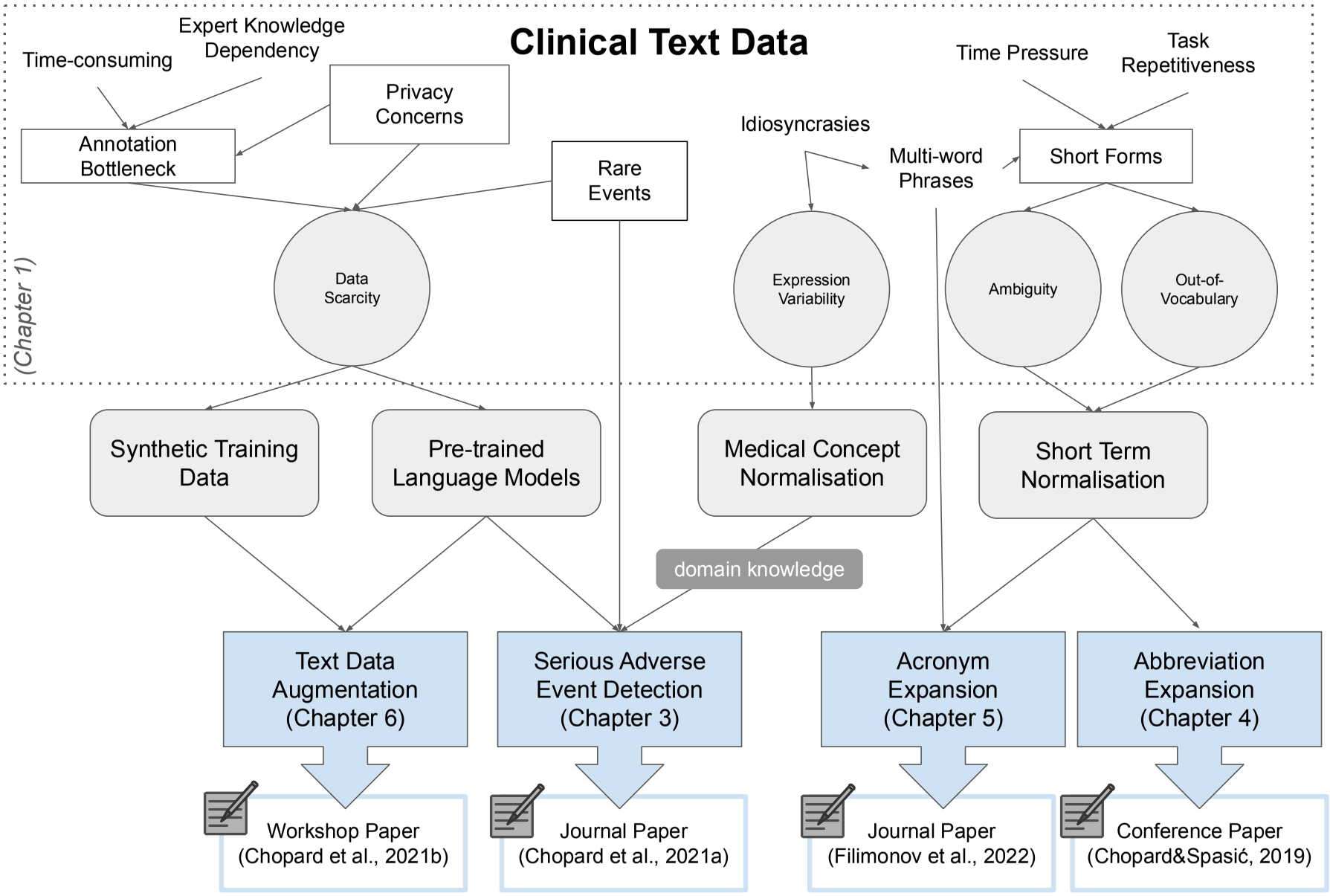


Figure 1.1: Overview of the thesis.

Chapter 2

The Foundations of Deep Learning in Clinical Text Mining

Deep learning is a sub-field of machine learning concerned with artificial neural networks (ANNs), a set of algorithms loosely inspired by the functioning of the animal brain. Like classical machine learning algorithms, ANNs use data to learn to perform tasks without being explicitly programmed for it. However, unlike the former, ANNs do not rely on manually engineered features for training but instead extract them directly from the raw input, limiting human intervention in the learning process. This key characteristic, among others, is thought to be responsible for the recent success of DL in many areas, such as computer vision, speech recognition and NLP.

In practice, a deep feed-forward ANN in its quintessential form consists of an input layer, one or more hidden layers and an output layer. Such an ANN is often referred to as a multilayer perceptron (MLP). The units of the network input layer hold the raw input. Examples of text input representations will be discussed later in this chapter. Then, these units are fed to the next layer of neurons—the first hidden layer—according to predefined connections. Each neuron in that layer takes a weighted sum of its input to produce an output using a non-linear function, the so-called activation function. The resulting value is then used as input for the next hidden layer and so on until the last layer of the network—the output layer—is reached. As the data progress through the layers, they are progressively transformed into more complex and abstract representations. Finally, the output layer maps the values of the last hidden layer—which contains the highest level of features—directly to a final output value whose interpretation depends on the task that needs to be performed. For example, if the task at hand is text classification, the output would be a vector of scores, each corresponding to the probability that the input text belongs to the corresponding class. The learning process then consists

in repeatedly getting the output of the network for each input in a set of samples and adjusting the weights of the network accordingly to minimise a measure of the difference between that value (or vector of values) and the true output.

Overall, it has been shown that feed-forward ANNs are universal function approximators (Cybenko, 1989, Hornik et al., 1989, Leshno et al., 1993, Pinkus, 1999). In other words, any function can be approximated by a feedforward ANN. Thus, the purpose of the learning process of deep ANNs is to use data to learn the network weights that correspond to the best function approximation for the task at hand. Since finding the underlying mapping between a model's input and output is at the core of machine learning, this theoretical foundation explains why DL has been so groundbreaking.

In the rest of this chapter, we delve more deeply into the specifics of DL in the context of text data and, more specifically, clinical text data. The goal is to provide the theoretical foundations needed to understand the remainder of this thesis. First, in Section 2.1, we briefly introduce clinical text data. Afterwards, Section 2.2 summarises the most well-known kinds of input representations for texts. Then, in Section 2.3 we look into the NN architectures that are most widely adopted when dealing with text data. Finally, in Section 2.4 we discuss the principles of the learning process and how they can be impacted by a lack of data that is inherent to clinical text data.

2.1 Clinical Text Data

The data that is at the centre of our interest in this work is clinical text data. In the introduction, we discussed the challenges associated with this kind of data. In this section, we focus on the properties of this data source.

It has been estimated that 80% of the data is in an unstructured format (Murdoch & Detsky, 2013). As a result, it is not sufficient to rely solely on structured data when trying to extract information. For example, Rannikmäe et al. (2020) reported that for UK Biobank participants, 42% of the cases were coded as "unspecified stroke" even though the stroke subtype could be found in the unstructured part of the EHR in 99% of the cases. Similarly, free texts allow for more comprehensive descriptions, which cannot easily be captured in structured data Jensen et al. (2017).

In the UK, EHRs typically contain different types of free text such as GP clinic notes, hospital reports (e.g. radiology and pathology reports), and discharge summaries (Ford et al., 2021). However, any kind of clinical text is relevant to clinical text mining. For example, as it will be seen in Chapter 3, clinical trial reports constitute an important source of information.

Depending on the types of clinical text and their intended use, the quality of the data can vary. On the one hand, patient records, which are mainly used by a small group of people for mnemonic reasons, might contain more noisiness. In contrast, discharge summaries are usually well-written and well-structured as they are intended for a larger audience. In between those two kinds of narratives, pathology and scan reports are semi-structured, and often contain fewer spelling mistakes than patient records (Dalianis, 2018). On the other end of the spectrum, scientific articles from the medical domain are very well written and should contain, in principle, no spelling mistakes.

As mentioned in Chapter 1, one of the challenges of clinical text data is privacy. In order to be used for research, data from the medical domain must first be deidentified as they contain sensitive information. As a result, the amount of data is often limited. Despite this, there exist a few publicly available datasets for clinical text mining. The main one is probably MIMIC (Medical Information Mart for intensive care) which consists of deidentified data collected over more than 10 years at the intensive care units of Beth Israel Deaconess Medical Center (Johnson et al., 2016). Different versions of the dataset have been released over the years (Johnson et al., 2020). On top of free-text clinical notes, such as discharge summaries, progress notes and radiology reports, the dataset contains, for example, vital information, democratic information, medications, and interventions. The MIMIC dataset has often been leveraged to train models for automatic ICD coding (Huang et al., 2019a, Li et al., 2018). Other important sources of publicly available datasets for clinical text mining have been released for the purpose of community challenges. In 2006, the i2b2 (Informatics for Integrating Biology to the Bedside) project provided a dataset of 1000 patient records labelled with to the patient's smoking category (Uzuner et al., 2006). Similarly, the CLEF eHealth evaluation lab has gathered datasets for automatic clinical coding of diagnoses for multiple languages (Suominen et al., 2018, Kelly et al., 2019, Goeuriot et al., 2020). Although all these datasets have been invaluable for research in clinical text mining, they unite most of the research efforts on a subset of clinical tasks for which annotations are available. In addition, it is still unclear, whether models developed on data from a specific institution translate well to different institutions Spasić et al. (2020).

2.2 Input Representation

As mentioned in the introduction of this chapter, one of the key characteristics of ANNs is that they are capable of extracting higher-level features directly from raw input. In the context of computer vision, the pixel intensity values in an image—or a sequence of images if one deals with videos—are often used as input representation (LeCun et al.,

2015). In speech recognition, the input audio could, for instance, be represented as a sequence of amplitude values throughout time. However, this representation does not take into account the rate of the signal. Alternatively, the signal could be transformed into the frequency domain so that the input consists of the amplitude at each frequency instead. Yet this representation is not ideal as it completely ignores the time component. This is why, in practice, audio waves are often first converted to spectrograms—which are visual representations of the signal frequencies through time, with each pixel intensity corresponding to the amplitude—before being fed to an ANN (Wyse, 2017). This allows for a richer representation while still keeping the amount of feature engineering to a minimum. Overall, even though NNs can deal with raw input, it is important to provide them with informative enough data to better exploit their potential for learning.

Similarly, when dealing with text data, it is important to find a meaningful input representation that is as simple and compact as possible but still captures the richness of language. However, unlike images and sounds, texts consist of discrete meaningful units (i.e. words). When combined in a specific way, this finite set of symbolic units can form sentences to express an infinite variety of meanings (Studdert-Kennedy, 2005). The sparse and discrete nature of language increases the difficulty of finding a suitable continuous representation.

In the remainder of this section, we describe the most common forms of input representation for text data and discuss their perks and drawbacks. In addition, we provide examples of clinical text mining applications using these representations.

2.2.1 Discrete Representations

Vector representations for text can be split into two main categories—localist or distributed—depending on whether each component of the vector has a designated meaning or whether they all participate in the representation of multiple concepts (Hinton, 1984). In this first subsection, we discuss the former kinds of representations, also called *discrete representations*, where each word is treated as an atomic unit.

One-Hot Encoding

The most trivial form of discrete text representation consists of a sequence of one-hot vectors whose size corresponds to the number of words in the vocabulary. Each coordinate corresponds to a word in the vocabulary. All values within a vector are set to zero apart from the one corresponding to the given word, which is set to one. This representation is referred to as *one-hot encoding*.

This representation has many advantages: it is easy to obtain, can capture word

order, and assigns each word a unique vector. Unfortunately, this representation also has multiple drawbacks. First, it is not scalable since the size of the vector increases linearly with the size of the vocabulary. Similarly, the size of the sequence representation—which is equal to the size of the vocabulary times the number of words in the text—varies for input sequences of different lengths; this is an issue because all representations must be of the same size when training a DL model. In the same vein, one-hot encoding representations do not have any way of handling OOV words (i.e. words that were not part of the corpus used to create the vectoriser) other than retraining the model from scratch with a vocabulary of extended size. Thus, every time this representation needs to accomodate a new word that is not part of the vocabulary, the dimensions of all existing vectors must be increased accordingly. Second, it neither captures the relationships between words nor their semantics: vectors are orthogonal to one another, and, as a result, similar words are as far from each other in the vector space as dissimilar words are. Third, this type of representation merely reflects the presence or absence of words, not their importance. Finally, because the corresponding matrix is highly sparse and high-dimensional, one-hot encoding is computationally inefficient. For these reasons, it is rarely used as such to represent text. Instead, this encoding is often leveraged to generate more suitable representations (see Section 2.2.2 for example).

Bag of Words

The motivation behind the so-called bag-of-words (BOW) representation is that similar documents contain similar words. Thus, this representation takes into account the frequency of words and represents each text sequence as a single vector with one entry for each word in the vocabulary whose value directly relates to that word's frequency. As a result, similar documents are assigned a closer vector representation than dissimilar ones.

There exist different variants of BOW depending on how word weights are computed. The most basic one is *CountVectorize*, where each vector entry is equal to the number of occurrences of the corresponding word in the text sample. The most popular alternative is term frequency-inverse document frequency (TF-IDF) which relies on the statistic of the same name. It is computed as the product of the term frequency—i.e. the relative frequency of the word in a document—and the inverse document frequency—i.e. the logarithm of the quotient of the total number of documents divided and the number of documents containing that word. In more simple terms, the TF-IDF of a word can be thought as a measure of the amount of information provided by this word. This statistic offers a way of penalising very frequent and very rare words in order to capture the importance of words within a text.

In clinical text mining, Dessi et al. (2020) successfully leveraged BOW representations based on TF-IDF with a MLP for identifying morbidity types in clinical notes. Nevertheless, when investigating two separate classification tasks—namely smoking status and proximal femur fractures classification from clinical notes and radiology reports, respectively—Wang et al. (2019b) found that the TF-IDF BOW representations were significantly outperformed by more complex ones, such as word embeddings introduced below in Section 2.2.2.

Overall, compared to one-hot encodings, BOW representations offer the advantage of being of the same size for each text sequence, irrespective of its length. However, even though these vectors reduce the input dimensionality to the size of the vocabulary, they can still be sparse as the number of words in a document is often much smaller than the number of words in a vocabulary. In addition, since every word occurrence is collapsed into a single vector, this representation loses any information about the word order and, as a result, two sentences of opposite meanings (e.g. "the cat eats the dog" and "the dog eats the cat") have the same representation. A variant of BOW which is designed to account to the word order is bag-of-n-grams. This approach counts the number of times each n-gram appears in a text. An n-gram is a common concept in NLP and is defined a sequence of n consecutive words. Thus, it is a generalisation of the BOW model which represents texts with 1-grams (often called unigrams). Using n-grams with larger n allows to capture more precise information about the text. However, the larger the n, the higher the number of n-grams in a text and the less frequent each n-gram. As a result, the size of the vocabulary increases which leads to more sparsity. Therefore, even though n-grams with a large n are more informative, they can become inefficient.

In addition—as is the case with any discrete representation—the vector size is directly proportional to the vocabulary size, making BOW representations poorly scalable. Other drawbacks include the inability to capture any semantic similarity and most importantly to handle OOV words. As we know, this is an important issue when dealing with small datasets. Indeed, any classifier trained using the BOW representation will not be able to leverage words that were not previously encountered in the training data. The smaller the number of training samples, the less likely the classifier will have seen them during training.

2.2.2 Distributed Representations

Discrete representations have the disadvantage of being sparse and high-dimensional. In contrast, distributed representations are, by nature, dense and low-dimensional as each component of the vector representation can contribute to the meaning of multiple

words (Hinton, 1984). Overall, they have been shown to offer good internal representations that capture important features of the task domain (Rumelhart et al., 1986).

Bengio et al. (2000) were among the first to introduce—in the context of statistical language modelling—distributed representation of words in the form of real-valued word feature vectors. These representations, which are now commonly referred to as *word embeddings* or *word vectors*, were successfully trained as part of a one-hidden layer feed-forward NN that aimed to predict the next word in a sequence.

Following their work, Collobert & Weston (2008) have demonstrated the utility of using word vectors that were pre-trained in an unsupervised setting as input representations to improve the performance of NNs on downstream tasks. Their experiments also showed that this approach generates embeddings that capture syntactic and semantic information in the sense that they cluster—syntactically or semantically—similar words in the vector space.

Since then, various methods have been developed with the sole purpose of learning word embeddings from large corpora. Because this new paradigm decouples the learning of word representations from that of the task, one can leverage shallow architectures which are computationally cheaper to train compared to networks that learn word embeddings implicitly as a by-product of solving a task. While there exist multiple variants of models designed for learning word representations, they are generally based on the same three fundamental components—the embedding layer, the intermediate layer and the softmax layer—that were introduced with the very first model Bengio et al. (2000). More specifically, the role of the embedding layer is to look up the embedding of a word in a (trainable) lookup table, the so-called embedding matrix, based on its index in the vocabulary. The intermediate layer (there may be several of them) relies on non-linearities to transform the input into more abstract representations. Finally, the softmax layer outputs a probability distribution over the vocabulary words.

In this section, we discuss the most popular models for generating word embeddings.

Word2Vec

One of the most well-known models for training word embeddings is the so-called *Word2Vec* model, which was developed by Mikolov et al. (2013a) and popularised the idea of using pre-trained word vectors as input representation for NNs. Word2Vec utilises either of two model architectures (the continuous bag-of-words model or the skip-gram model) to learn word embeddings in an unsupervised manner. Both architectures rely on a similar shallow NN, but each is trained on a different self-supervised task.

Continuous bag-of-words (CBOW) The CBOW model is tasked with predicting a target word given its surrounding context. However, the implicit goal of the model is to learn an embedding matrix V where each row i $(0 \leq i \leq N-1)$ is a d-dimensional vector containing the embedding of word i from a vocabulary of size N. The arbitrary hyperparameter d is often referred to as *embedding dimension* and defines the size of the embedding space, which also corresponds to the number of units in the input layer.

The learning process unfolds as follows: at the very beginning of training, two weights matrices $V \in \mathbb{R}^{d \times N}$ and $U \in \mathbb{R}^{N \times d}$ are randomly initialized. The matrix V contains the weights between the input layer and the hidden layer and will eventually encode in its rows the representations of the input words—i.e. the words when they appear in the context. Similarly, the matrix U contains the weights that connect the hidden layer to the output layer and embeds in its columns the vector representations of the output words, i.e. the words as targets. At each time step t, the network is given a window of n words around the target word w_t. These context words are first mapped individually to their embedding—i.e. their one-hot vector is multiplied with the weight matrix V to extract the corresponding row—before being averaged. Note that any order information amongst context words is thereby discarded. The resulting d-dimensional vector is then multiplied with the matrix U producing a vector of size N which, in turn, is transformed into a probability distribution over the vocabulary through a softmax function. The prediction can then be compared to the true label—the one-hot embedding of the target word w_t—and the matrix weights can be updated accordingly through backpropagation and stochastic gradient descent (well-known learning processes which will be explained in more detail in Sections 2.4.4 and 2.4.4). Even though the model ends up learning two independent representations for each word (depending on whether it is a context word or a target word), only the word vectors encoded by the matrix V are eventually used as word embeddings in downstream tasks.

Skip-Gram This skip-gram model is similar to the CBOW model but is trained with the reversed goal (i.e. the input and the output are swapped): given a target word, it must predict words in its surrounding context. Thus, in this setting, the weight matrix between the input and the hidden layer encodes words as target words, whereas the other matrix contains the embeddings of words as context words. Since there is a single input vector, it is no longer necessary to average after mapping the one-hot encoded input to the embedding matrix. Instead, the intermediate state vector at the hidden layer is simply the target word's embedding. The model then takes this representation and generates n score vectors—one for each word in a window of size n around the target word, which are then turned into probabilities using a softmax func-

tion. These probabilities are then compared to the true probabilities (i.e. the one-hot vectors of the actual context words) and, once again, the weights of the matrices U and V are adjusted via stochastic gradient descent to minimize the error. In this case, it is the matrix containing the target word vectors that can be used as an embedding matrix.

Overall, even though the two methods are mirrored versions of each other, they offer different perks and drawbacks. For example, the CBOW method converges much faster as it is trained with an easier task—at each time step a single prediction is made and evaluated—than the skip-gram model which requires a prediction for each word in its context. Mikolov et al. (2013a) reported that the CBOW model could be trained in less than a day as opposed to almost three days for the skip-gram model. However, the latter is less prone to overfit frequent words and can in general better represent rare words, as it relies on single-word inputs. Similarly, the skip-gram approach yields better results when trained with smaller datasets compared to the CBOW model.

By leveraging the context of words to learn continuous vector representations, the Word2Vec models draw on the concept of distribution similarity—i.e. the idea that the meaning of a word can be inferred from the context in which it appears—and the distributional hypothesis—i.e. when two words occur in a similar context, they have a similar meaning (Harris, 1954, Firth, 1957). Since, by design, the models must output similar word vectors for words that appear in similar contexts, words with similar meanings will have similar representations. Thus, they will be closer together in the vector space than words that are semantically different. Most surprisingly, word embeddings trained with Word2Vec have been shown to exhibit interesting characteristics (Mikolov et al., 2013b, Levy & Goldberg, 2014). Namely, there exist linear relationships amongst word vectors in the continuous space. As a result, analogies of the form "w_a is to w_b as $w_{a'}$ is to $w_{b'}$" can be made using addition and subtractions: $w_b - w_a + w_{a'} \approx w_{b'}$. The most famous example of such an analogy is "man is to king as woman is to ...?" with $w_{king} - w_{man} + w_{woman} \approx w_{queen}$ yielding the word "queen". In fact, the existence of linear analogical relationships among these kinds of embeddings was later proven mathematically by Allen & Hospedales (2019). The models are not only able to capture semantic analogies (e.g. "Paris - France + Rome = Italy"), but also syntactic analogies (e.g. "bad - worst + big = biggest" or "dancing - danced + going = went").

Muneeb et al. (2015) and Chiu et al. (2016) reported, among others, that skip-gram outperformed CBOW for biomedical tasks. The former study investigated semantic relatedness of biomedical terms, whereas the latter tackled biomedical NER. In the clinical text mining literature, Word2vec embeddings obtained with the skip-gram model

have for instance been combined with other features for a clinical concept extraction task (De Vine et al., 2015). Similarly, in order to achieve higher effectiveness in clinical information extraction, Kholghi et al. (2016) have used—on top of hand-crafted features—word embeddings generated from a clinical corpus using the skip-gram model. Word2vec skip-gram embeddings were also found to yield better performance than other embeddings such as CBOW, GloVe and FastText (see Sections 2.2.2 and 2.2.2 below) on a mortality prediction task (Krishnan, 2019). On the other hand, Luo (2017) experimented with word2vec embeddings that were pre-trained on either the general domain or clinical notes from MIMIC-III and reported that the latter word vectors outperformed the former ones on a clinical relation classification task, thereby demonstrating the importance of domain-specific representations. In the same vein, (Patel et al., 2017) proposed a modified version of word2vec embeddings that includes domain-specific information for the task of medical coding. They showed that these modified embeddings consistently performed better than the original ones. Later, Wang et al. (2019b) showed that weakly-supervised ML models that used word2vec skip-gram embeddings outperformed those that used TF-IDF features on two different clinical text classification tasks (smoking status classification and hip fracture classification). Word2vec embeddings have also been utilised in information retrieval for biomedical query expansion (Wang et al., 2017). Other examples of using word2vec embeddings in clinical text mining include clinical abbreviation disambiguation (Xu et al., 2015), clinical NER (Wu et al., 2015b), predicting unplanned readmission from hospital patient records (Nguyen et al., 2016), adverse drug events identification from clinical notes (Henriksson et al., 2015) and phenotype classification from discharge summaries (Gehrmann et al., 2018).

Global Vectors (GloVe)

Another popular algorithm for learning word vector representations is GloVe which was developed at Stanford University by Pennington et al. (2014). Rather than being based on a language modelling task as Word2Vec, this model draws on matrix factorisation to generate word embeddings. More specifically, the model adds global statistics into the process by taking into account the word-to-word co-occurrence matrix over the entire vocabulary which contains in each entry X_{ij} the number of times word j appeared beside word i in the corpus. From the co-occurrence matrix, one can easily compute the probability of a word j appearing in the context of another word i—i.e. $P(j|i)$—by dividing the value X_{ij} by the sum of the values in row X_i, namely $P_{ji} = P(j|i) = X_{ij}/X_i$ with $X_i = \sum_k X_{ik}$. The model relies on ratios of occurrence probabilities to capture semantic information between words. The use of ratios helps discriminate between words that are relevant and those that are not. More specifically, the ratio $\frac{P_{ik}}{P_{jk}}$ takes on a large

value (i.e. $>> 1$) if word k is related to word i but not to word j. Similarly, the value of the ratio will be small (i.e. $<< 1$) if k occurs more times in the context of word j than in the context of word i. In contrast, if word k occurs frequently or rarely in both contexts, then the value of the ratio will be close to 1. Based on this observation, the following training objective (see Section 2.4.3 is defined:

$$J = \sum_{i,j} f(X_{ij})(w_i^T \tilde{w}_j - \log X_{ij})^2 \tag{2.1}$$

In other words, the model tries to minimise the difference between the dot product of the word vectors and the logarithm of the words' probability of co-occurrence. For computational efficiency, this difference is weighted by the function $f(x)$ so that smaller weights are given to words that are further apart. The authors propose using the following weighting function:

$$f(x) = \begin{cases} (x/x_{\text{max}})^\alpha & \text{if } x < x_{\text{max}} \\ 1 & \text{otherwise} \end{cases} \tag{2.2}$$

GloVe is faster to train than Word2Vec but requires more memory for the computation of the co-occurrence matrix. However, since this matrix only needs to be computed once upfront GloVe— unlike Word2Vec—does not need to be retrained from scratch if the embedding dimension changes, it can re-use the co-occurrence matrix and only modify the dimensionality of the factorisation. As mentioned earlier, GloVe also incorporates global statistics into the learning process. In fact, the authors have shown that GloVe is equivalent to the Skip-gram model but with added global statistics (as opposed to local statistics only as captured by a context window).

Because the GloVe model is designed to force linear relationships between words based on the co-occurrence matrix, analogies can be drawn from the resulting embedding vectors similarly to Word2Vec.

GloVe embeddings have, for instance, been used for clinical information extraction and relation extraction tasks (Wang et al., 2018c). Ong et al. (2020) reported GloVe representation supported the best performance for radiology report classification when compared to BOW and TF-IDF. Similarly, GloVe embeddings were also shown to achieve higher performance compared to word2vec embeddings on detecting comorbidity relationships in clinical notes (Beam et al., 2019). However, they were outperformed by word2vec on other types of relationships. Bhatia et al. (2019) have used GloVe word embeddings on top of character and tag embeddings to extract negative medical findings

in clinical reports. On the other hand, Dingwall & Potts (2018) introduced an extension of GloVe for fitting a new vocabulary of words. They added an extra term to the training objective that encourages the learned embeddings of words that were already present in the GloVe vocabulary to be as close as possible to existing embeddings. As the training objective is retrofitting, all representations must be retrained each time a new word is encountered. This novel model was shown to outperform the original GloVe embeddings and those pre-trained on clinical notes for a clinical sequence labelling task.

One drawback shared by both GloVe and Word2Vec is their inability to represent words that are not part of the vocabulary. Solutions to this problem include ignoring OOV words, replacing them with the same token (e.g. <UNK>) whose embedding can be learned during training, or initializing them from a uniform distribution. Nevertheless, all these options completely discard the meaning of OOV words.

FastText

In addition to being unable to handle OOV words, Word2Vec and GloVe fail to take into consideration the internal structure of words. A solution to model the morphological structure of words suggested by Bojanowski et al. (2017) is to divide them into subwords before feeding into an NN. This model is referred to as *FastText*. More precisely, words are broken into character n-grams and the sum of the vector representations of these n-grams is used as word representation. For instance, if we choose $n = 3$ the word *patient* is represented as the sum of the vector representations of the tri-grams '<pa', 'pat', 'ati', 'tie', 'ien', 'ent', and 'nt>'. Note that the symbols '<' and '>' are added at the beginning and the end of the word to differentiate prefixes and suffixes from other subwords. Vector representations for the individual n-grams are obtained using the skip-gram algorithm introduced earlier but trained on windows of n-grams instead of windows of words.

As FastText embeddings encode morphological information about words, they can represent OOV words as well as misspelt and rare words provided that the n-grams composing these words are present in the training data. One disadvantage of FastText is that is computationally more expensive than the previous models as it operates at a finer granularity. Similarly, it requires more memory. In particular, both these issues become more important as the corpus size increases.

In the clinical domain, FastText has for example been used to represent medical concepts in an event prediction task (Lu et al., 2020). In addition, Yang et al. (2019) showed that FastText embeddings outperformed Word2Vec embeddings on a clinical note de-identification task. More surprisingly, they reported that FastText embeddings

trained on the Common Crawl corpus—a collection of web pages—yielded slightly better performance than those trained on clinical notes. In contrast, Romanov & Shivade (2018) observed, in the context of medical natural language inference (NLI), that FastText embeddings trained on clinical data or FastText embeddings trained on Wikipedia articles and fine-tuned on clinical data achieved higher scores than those only trained on Wikipedia or PubMed abstracts. However, the authors noticed that FastText embeddings trained on Wikipedia performed worse than GloVe embeddings trained on the Common Crawl corpus.

One of the key issues of context-free distributed representations such as Word2Vec, GloVe and FastText is that each word is assigned a unique vector representation. Thus, it is impossible to differentiate between homonyms in the vector space. For example, the word *"cold"* takes on different meanings when used in the sentence *"Patient started feeling pain after taking a cold shower."* and in the sentence *"Patient suffers from a cold."*: the first one refers to a low temperature, whereas the second one refers to an illness. However, the word embedding models presented above collapse the two meanings into a single representation and fail to capture the polysemous nature of words.

2.2.3 Contextualised Word Embeddings and Pre-trained Language Models

Context-sensitive representations that capture the contextual meaning of words have been suggested as an alternative to address the inability of context-free word embeddings to differentiate between homonyms. Hence, for instance, the word *"mole"* used as *"a unit of measurement"* and as *"a disorder that affects the soft tissue"* will have distinct representations in the word-embedding space. In this section, we discuss these approaches in more detail.

Embeddings from Language Model

One of the first methods to generate context-sensitive word representations was ELMo (Embeddings from Language Model) (Peters et al., 2018). ELMo representations are functions of the entire input sequence and are obtained using a two-layer bidirectional language model (biLM). Each layer in the biLM consists of separate forward and backward passes. First, the sequence of words (e.g. a sentence) is fed to the first layer of the model using a character-level convolutional neural network (char-CNN) (Kim et al., 2016) (See Section 2.3.1 for details on CNNs). This allows embeddings to take into account the substructure of words and to accommodate for unknown words easily, thereby

eliminating the problem of OOV words. The input is then processed in each direction by two separate long short-term memorys (LSTMs)—a well-known NN architecture that is detailed in Subsection 2.3.2. The objective of the forward LSTM is to predict the next word in the sequence starting with the first one, while the backward LSTM must do the opposite and predict the previous word starting with the last one. The outputs of the two LSTMs—each of which contains an intermediate representation of the sequence—is then fed to the corresponding LSTM in the second layer of the biLM. The second layer does the same as the first one and produces another intermediate representation. Eventually, the final ELMo representation is computed as the weighted sum of the char-CNN representation and the two intermediate word representations. The coefficient values are task-specific. Indeed, the authors show in their paper that each of the three representations encodes different information, with lower layers capturing better syntactic information and higher layers semantic information. Therefore the importance of each representation depends on the task to solve.

Even though the authors suggest first fine-tuning the biLM on domain-specific data to improve the performance on the downstream task, the biLM weights are eventually fixed during training. More specifically, they propose freezing the weights and concatenating the resulting ELMo representations with other representations (e.g. Word2Vec) before incorporating them into existing neural NLP architectures without further modification.

Si et al. (2019) compared the performance of ELMo embeddings to context-free word embeddings, such as Word2Vec and GloVe, on four different clinical concept extraction datasets. Their results indicated, on one hand, that using ELMo embeddings improved performance in most cases and, on the other hand, that training the ELMo embeddings on a clinical dataset significantly helped downstream tasks. Similarly, a study by Zhu et al. (2018) showed that domain-specific ELMo embeddings (namely, generated using clinical reports and clinically relevant Wikipedia articles) produced better results for a clinical concept extraction task. In particular, this approach outperformed several existing benchmarks on the same dataset. In the same vein, Jin et al. (2019a) evaluated BioELMo—an ELMo model trained on PubMED abstracts—on both a biomedical NER and a biomedical NLI task. They found this model to be, in most cases, better than its general domain counterpart. Another instance of a domain-specific ELMo for biomedical NER was introduced by Sheikhshab et al. (2018). In contrast, Peng et al. (2019) used a biomedical ELMo on five tasks and ten different datasets from the clinical domain and found that it outperformed the state-of-the-art approaches in three instances only.

Universal Language Model Fine-tuning

A significant shift in how word embeddings are integrated into NNs followed the work of Howard & Ruder (2018). In their paper, the authors suggested a straightforward way to make inductive transfer learning—i.e. the idea of transferring knowledge from a source task to a different target task with labels only available in the target task—possible for NLP. The authors were inspired by the success of this approach in the field of computer vision, where models are not trained from scratch but rather first pre-trained on large datasets, such as ImageNet (Deng et al., 2009), before being fine-tuned on a downstream task (Donahue et al., 2014). More specifically, Howard & Ruder (2018) introduced a new model called Universal Language Model Fine-Tuning (ULMFiT) consisting of a single architecture—a 3-layer LSTM—that is first pre-trained on a large unlabelled corpus from the general domain through a LM task and is then fine-tuned on target task data. Eventually, two feed-forward layers are added to the pre-trained LM which can then be fine-tuned on the target task classifier. As the model learns general properties of language during the pre-training of the LM, the function of the fine-tuning phase is simply to adapt the LM to the specificity of the target data. Hence, this allows for fast convergence and robust representations even with small annotated datasets. In fact, the same pre-trained LM can be fine-tuned on any NLP classification task without the need to modify the underlying architecture. Therefore, ULMFiT offers a new paradigm: relying on pre-trained LMs instead of pre-trained word embeddings to represent natural language. This shift has minimised the dependency on manually annotated datasets. A labelled dataset is required only for fine-tuning on the downstream task and is no longer used to model the language itself. This means that its size does not need to be as big. Consequently, the manual annotation of datasets can become a less arduous task without necessarily sacrificing the performance. Indeed, even smaller datasets can lead to great generalisation performance thanks to the general nature of the underlying LM (Howard & Ruder, 2018).

Examples of uses of ULMFiT in the clinical domain include the classification of patients' smoking status from clinical narratives (Hirvonen et al., 2021), adverse drug responses in tweets (Dirkson & Verberne, 2019), and benign and malignant breast biopsies from pathology reports (Issa et al., 2021). The first two studies reported choosing to use ULMFiT as a classification framework to take advantage of transfer learning because of a lack of annotated data. Issa et al. (2021) compared this model to more recent pre-trained LMs but found that they all yielded similar performance.

Generative Pre-Training

To overcome the limitations of the LSTM architecture used by Howard & Ruder (2018) in their ULMFiT model—such as poor handling of long-term dependencies—Radford et al. suggested an approach called Generative Pre-Training *or* Generation from Pre-trained Transformers (GPT). It replaces the LSTM in the LM with a variant of the transformer architecture (Vaswani et al., 2017) consisting of a multi-layer transformer decoder. Indeed, the transformer architecture—which is explained in more detail in Section 2.3.4—relies on attention mechanisms which makes it possible to process the whole input simultaneously instead of sequentially. GPT was shown to improve the state-of-the-art on 9 out of 12 datasets and to outperform the LSTM-based pre-trained LM on all but one dataset. The input sequences are encoded using byte pair encoding (BPE), a subword-based tokenisation that iteratively looks for the most frequent pair of characters and merges them. This in turn enables the model to handle OOV words. More scaled-up versions of the GPT model have been released in subsequent years: GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020). Even though the GPT LM was originally evaluated on supervised tasks, GPT models have since then been mainly praised for their capacity to generate high quality texts that stand out by their fluency and lexical diversity which makes them hard to distinguish from human-written texts. Consequently, they are now mainly used for that purpose (Zhang et al., 2020b, Dale, 2021).

GPT models have been leveraged in the clinical domain in multiple instances. For example, (Van et al., 2020) investigated the use of pre-trained LMs such as GPT-2 to autocomplete text simplifications in the medical domain. In addition, Li et al. (2021) have used GPT-2 to generate synthetic clinical notes for training a NER model in an attempt to overcome the privacy concern of real clinical data and achieved similar performance as when using the real notes. Finally, Logé et al. (2021) have assessed the capacity of GPT-2 and GPT-3 to answer medical questions related to patient pain dosage.

Bidirectional Encoder Representations from Transformers

One of the shortcomings of the GPT model is its unidirectional nature. This limitation was addressed by Devlin et al. (2019) who proposed a clever way of making an LM truly and deeply bidirectional: they introduced a masked language model (MLM) objective where random words are replaced with a mask and have to be predicted. The architecture is called BERT and consists of transformer encoders blocks as opposed to the decoders block from the GPT model. In addition to the MLM task, BERT is also

pre-trained on a next sentence prediction (NSP) task. During training, BERT is given a pair of sentences from the corpus and must predict whether the second sentence directly follows the first one in the original text. BERT uses WordPiece tokenisation to obtain subword units by applying a greedy segmentation algorithm to minimise the number of WordPieces in the training corpus (Wu et al., 2016b). This implies that the model may be able to leverage the word morphology of unknown words when dealing with previously unseen data. Overall, BERT was shown to outperform existing approaches on all natural language understanding (NLU) tasks from the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018a) as well as question answering (QA) tasks with little modification. Overall, BERT is the model that has been the most successful due to its exceptional performance and the open-source release of large pre-trained models that can be easily fine-tuned on a wide array of tasks.

The success of BERT has led to multiple powerful variants. First, to lower the computational requirements, Lan et al. (2019) introduced **ALBERT**. This lite version of BERT uses both cross-layer parameter sharing and factorised embedding parametrisation to reduce the number of parameters by 18. As a result, training and inference are sped up by almost 2 while achieving only slightly worse performance. Second, **RoBERTa** (Robustly optimized BERT pre-training Approach) was suggested by Liu et al. (2019). It was shown to outperform BERT on all GLUE tasks. This model differs from the $\text{BERT}_{\text{BASE}}$ model on four points: First, the static MLM task is replaced by a dynamic one. Instead of relying on the same masking pattern at each epoch, different parts of the sentence are masked at different epochs. Second, the NSP task objective is removed from the training procedure. Third, RoBERTa is trained on an about ten times larger dataset, which includes the Common Crawl-News and the Open WebText datasets on top of the original BookCorpus and English Wikipedia datasets. Finally, larger batches of 8'000 samples instead of 256 are used to train the model. A third popular variant of BERT is **ELECTRA** (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) (Clark et al., 2019). Similar to RoBERTa, this variant does not rely on the NSP task for training. In addition, a Replaced Token Detection (RTD) task is used in place of the MLM task. Rather than being masked, tokens are replaced by another one. The model must then determine whether each token has been altered or is the original one. ELECTRA was shown to outperform state-of-the-art approaches with less pre-training time.

There also exist multiple domain-specific versions of BERT which have been pre-trained on clinical or biomedical data. For example, **BioBERT** uses the original $\text{BERT}_{\text{BASE}}$ architecture (initialized with weights from BERT) which they train on a collection of PubMed abstracts and PMC articles (Lee et al., 2020). Even though the medical ter-

minology differs from general English, the authors choose to use the same WordPiece vocabulary as the original model. This new pre-trained LM leads to improved performance on three different biomedical tasks: NER, QA and relation extraction. Similarly, **ClinicalBERT** is a version of $\text{BERT}_{\text{BASE}}$ that has been pre-trained on clinical notes from the MIMIC-III database Johnson et al. (2016) and has been developed for predicting hospital readmission (Huang et al., 2019b). Variants of ClinicalBERT have also been introduced by Alsentzer et al. (2019). Finally, **SciBERT** refers to a variant of BERT that has been pre-trained on scientific texts from both the computer science domain (18%) and the broad biomedical domain (82%) using a domain-specific WordPiece vocabulary (Beltagy et al., 2019). The authors show that SciBERT outperforms its classical counterpart $\text{BERT}_{\text{BASE}}$ on bio-medical tasks. It also achieves higher performance than BioBERT on multiple tasks even though BioBERT is trained on a larger biomedical dataset.

To assess the performance of BERT and ELMo models in the biomedical domain, Peng et al. (2019) introduced the Biomedical Language Understanding Evaluation (BLUE) benchmark, a collection of five tasks on ten different clinical and biomedical datasets. The best results were achieved by a $\text{BERT}_{\text{BASE}}$ model that was pre-trained on both PubMed abstracts and clinical data from the MIMIC-II dataset. Similarly, Ji et al. (2020) evaluated the effectiveness of the pre-trained BERT, BioBERT and ClinicalBERT models for biomedical entity normalisation and reported an increase in accuracy of up to 1.17% compared to existing state-of-the-art models. Finally, Liu et al. (2021) compared those three models in the context of clinical trial NER and concluded that the BioBERT model was the best model for extracting information from clinical trials.

2.3 Neural Network Architectures

Up until this point, we have only discussed NN architectures that consist of an input layer, one or more hidden layers and an output layer, all of which are fully connected (i.e. each node in a layer is connected to every node in the previous and subsequent layer). However, there exist other types of networks, some of which are very popular for NLP applications and which we will now shortly introduce.

2.3.1 Convolutional Neural Networks

Even though convolutional neural networks (CNNs) were originally proposed to process input images, they have also found some success for text data (Kim, 2014). The core building block of a CNN is the convolutional layer. This layer applies the same lin-

ear transformation—a convolutional filter—to each part of the input through a sliding window. Two-dimensional convolutions are usually applied to two-dimensional inputs such as images. One-dimensional convolutions are used for text data as they are one-dimensional data. A convolutional layer is generally followed by a so-called pooling layer which takes as input the feature map generated by the convolutional layer and reduces its dimensionality. For text classification, Chen (2015) suggested applying several convolutions of different sizes, each followed by a non-linear activation and a global pooling. Vector representations of the data are then obtained by concatenating the results of each convolution and can be used for classification. Alternatively, multi-layered convolutions (i.e. stacks of convolution and pooling blocks) can be used. This architecture is well suited for longer inputs, such as models processing input at the character level instead of the word level (Zhang et al., 2015). CNNs have also been suggested for language modelling (Pham et al., 2016). In contrast to text classification models, CNN LMs do not use pooling as they need to keep positional information intact. The main characteristic of CNNs is that they can extract features that are somewhat invariant to shifts in space, this allows the network to capture important local patterns (LeCun et al., 1998, Er et al., 2016). When the input is text, the convolutional layer looks at predictive n-grams and in a stack of multiple convolutional layers each can identify a different range of n-grams (Goldberg, 2017).

CNNs have the advantage of being more efficient than fully connected layers because the same filter reuses the same weights across the entire input (LeCun et al., 1998). However, even though they have shown good performance in text classification tasks, CNNs perform less well on other tasks that require the understanding of longer-range dependencies such as QA (Minaee et al., 2021).

Amongst others, CNNs have been used in clinical text mining for assigning medical subject headings to biomedical articles (Rios & Kavuluru, 2015), extracting clinical relations (Sahu et al., 2016, Raj et al., 2017, He et al., 2019), and disambiguating abbreviation senses in clinical narratives (Joopudi et al., 2018). An elaborated CNN with attention has also been leveraged for classifying radiology reports according to their degree of severity (Shin et al., 2017).

2.3.2 Recurrent Neural Networks

A popular class of NNs that are well suited for text data are recurrent neural networks (RNNs) (Rumelhart et al., 1986). Indeed, this architecture offers an effortless way to handle sequences of data, such as sequences of words, as the output of the hidden layer is fed back to itself in a recurrent manner. This means that the network can access information about previous parts of the sequence. Another advantage of RNNs is that

they can process a sequence of arbitrary length in a fixed-sized vector. In particular, since network weights are shared across time, the model size stays identical irrespective of input size (Goodfellow et al., 2016). These key characteristics make RNNs a good choice for processing text sequences. However, in practice, this kind of architecture is difficult to train and struggles to capture long-term dependencies (Bengio et al., 1994). Indeed, during training, the NNs are updated by comparing at each iteration the current output of the network with the true output and computing how fast each weight needs to be modified to gradually reduce that error. In practice, the backpropagation algorithm (Rumelhart et al., 1986)—which will be explained in Section 2.4.4—offers a convenient way to calculate the gradients with respect to the network weights and propagate them backwards through the network from the last layer through the first. However, this method involves successive gradient multiplication at each layer which means that if the gradients are small (less than 1) they will keep getting smaller the more layers there are and will not be able to reach earlier elements in the sequence. This is called the *vanishing gradient problem* (Bengio et al., 1994, Pascanu et al., 2013) and is an important issue in RNNs because the number of layers increases with the length of the input sequence.

Long short-term memory

The most popular variant of RNNs is LSTM which was introduced by Hochreiter & Schmidhuber (1997). This gating-based architecture improves on recurrent connections by offering more elaborate memory cells which are designed to address the issue of vanishing gradients.

An LSTM cell is composed of a cell state along with three different gates which together regulate what information goes in and out of the cell. The first gate, the *forget gate*, looks at the previous hidden state and the current element of the input sequence and decides what information should be forgotten from the cell state. Then, the second gate regulates how much each element of the cell state should be updated with new information. This is the role of the *update gate*. Finally, the *output gate* regulates what parts of the cell state should be forwarded to the following LSTM cell.

Even though LSTMs can tackle the vanishing gradient problem, they fail to eliminate it completely, which means that their capacity to handle long-range dependencies is still limited (DiPietro & Hager, 2020). Another drawback of LSTMs is that they require lots of resources to get trained (Hou et al., 2019). Because they contain a high number of parameters, LSTMs are prone to overfitting and must thus rely on large datasets for training (Bashar et al., 2020).

Gated Recurrent Unit

A simpler alternative to LSTMs cells for handling the vanishing gradient problem in RNNs is offered by gated recurrent units (GRUs) (Cho et al., 2014a). In GRUs, the cell state and the hidden state are merged. Similarly, a single *update gate* is used instead of both a forget and an input gate.

As mentioned earlier, RNNs are particularly well suited for processing text sequences. Therefore, it is no surprise that RNNs have been extensively used for clinical text mining. Examples of applications include the detection of disease and medication information from EHR notes (Jagannatha & Yu, 2016), clinical entity recognition from patient reports (Liu et al., 2017), clinical NER (Wu et al., 2017), and the classification of chest radiology reports for the presence of pulmonary embolism (Banerjee et al., 2019).

2.3.3 Encoder-Decoder Networks

Encoder-decoder networks—also called sequence-to-sequence (Seq2Seq) networks—are designed to handle input and output sequences of variable length. As its name suggests, this architecture consists of two main components: an encoder and a decoder. An encoder maps an input of variable length to a fixed-sized internal representation. A decoder does the opposite and takes that fixed-sized internal representation and outputs a sequence of variable length. Encoders and decoders can take different forms. For example, Sutskever et al. (2014) suggested using a pair of multilayered LSTMs as encoder and decoder, whereas Cho et al. (2014b) proposed using a pair of RNNs with GRUs cells instead.

In the clinical domain, Du et al. (2019) have used a Seq2Seq model to extract symptoms and their statuses (i.e., experienced/not experienced) from clinical conversations. Indeed, since symptoms are sometimes not described explicitly by patients and must rather be inferred from context, it is beneficial to formulate this problem as generating a list of symptoms and their statuses from a chunk of conversation instead of a span-attribute tagging task. Alternatively, Gharebagh et al. (2020) have used an LSTM-based encoder-decoder to summarise radiology reports.

Attention Mechanisms

The encoder-decoder architecture has two main problems. On one hand, it is difficult to compress all relevant information in a fixed-sized internal representation. On the other hand, the decoder has to extract all information for a single representation, which can be challenging. To address this issue, (Bahdanau et al., 2015) introduced the

attention mechanism. Instead of producing a fixed-internal representation, the encoder generates a separate encoder state for each input token. Then, at each time step during the decoding, a different representation is computed: an attention score between the current decoder state and each encoder state. This representation is immediately fed to the decoder to produce the next output at that time step. The attention score encodes the degree to which each element in the input sequence is relevant to the current decoding step. From this, attention weights can be computed by applying a soft-max function to the attention scores. Finally, attention outputs are obtained by multiplying encoder states and attention weights. This product is then fed to the decoder. Overall, this model allows the decoder to attend to specific parts of the input sequence when producing the next output.

The attention mechanism has, among others, been incorporated into a NER system to extract adverse drug reactions (Pandey et al., 2017). Similarly, Gligorijevic et al. (2018) introduced a model that uses the attention mechanism to predict the number of resources needed by patients when admitted to a hospital in order to improve patient prioritization. Finally, the attention mechanism has been leveraged for medical text categorisation (Qing et al., 2019).

2.3.4 Transformer

One of the biggest breakthroughs of the last decade in NLP is the Transformer architecture (Vaswani et al., 2017) which offers a new paradigm for processing sequences. Instead of recurrence, this architecture leverages only attention mechanisms to learn global dependencies between input and output. More specifically, the Transformer consists of encoder and decoder blocks that rely on self-attention, which means attention within a sequence itself rather than relative to another sequence. To compute attention outputs for self-attention, each element in the sequence receives three different representations: query q, key k and value v. The query is the vector representation of the input element that is looking at all other elements. The keys are the representations of the elements looked at by the query.They are used to compute the attention weights. Finally, the value is the representation that gives the information to the elements that need it and is used to compute attention outputs.

$$Attention(q, k, v) = softmax(\frac{qk^T}{\sqrt{d_k}})v \tag{2.3}$$

Note that the self-attention for the decoder is slightly different than for the encoder. Indeed, since the whole sequence is fed at the same time, future elements in the sequence have to be masked out so that the decoder does not know what comes next during

the generation process. Finally, to keep information about the order of the sequence, a positional embedding is added to each element. Vaswani et al. (2017) suggest using fixed positional embeddings (instead of learned ones) with each dimension corresponding to a sinusoid of different wavelengths. Because the whole input is processed at once instead of sequentially, the learning process can be parallelized which reduces the training time compared to RNNs and enables training on larger datasets. In addition to its greatly improved training speed, the Transformer was shown to improve the state-of-the-art on machine translation tasks (Vaswani et al., 2017) which has paved the way for its application to other tasks. In particular, as mentioned in Section 2.2.3, Transformer-based architectures have revolutionised the way text is processed by NNs through pre-trained LMs.

In the context of clinical text mining, transformer-based models have been used to measure semantic similarity for sentence pairs from clinical notes (Yang et al., 2020b), predict hospital readmissions (Amin-Nejad et al., 2020), extract lymph nodes from radiology reports (Peng et al., 2020) or extract clinical concepts (Yang et al., 2020a).

2.4 Training

To train NNs in a supervised setting, one relies on data pairs of input and output. The input is fed to the network, which processes it and produces an output according to the current value of its weights. This predicted output is compared to the true output using a cost function and the weights of the network are adjusted to minimise the observed error. Learning stops when the error rate cannot be reduced any further with new observations. The whole process is explained in more detail in the remainder of this section.

2.4.1 Training Data

To develop a DNN, as for any machine learning algorithm, it is important to take data into careful consideration. Indeed, data are not only essential to train the network and find its optimal weights but also necessary to estimate its performance on unseen data (see Section 2.5). Thus, it is common practice, when working on a new application, to split the data into three disjoint sets: a training set, a development set (also called validation set) and a test set. The training set is used exclusively, as its name suggests, to train the network and find an optimal set of weights. The development set acts as an indicator of out-of-sample performance and can be used to evaluate different hyperparameters or different architectures for example. Finally, the test set should

only be used a single time, once the architecture has been fixed, all hyperparameters have been set, and the training is complete. Its role is to give a good estimation of the generalisation performance of the network, that is the performance outside of the training data. In practice, when the data are very limited, one can resort to strategies to get an estimation of the generalisation error without setting aside samples for the development set. One popular approach is cross-validation (CV) (Allen, 1974, Stone, 1974, 1977). One of most well-known variant of CV is k-fold CV. It consists of splitting the data into k sets of equal size called folds. Then, $k-1$ of these sets are used to train the network and the remaining one to evaluate its performance on unseen data. This process is repeated k times, each time leaving a different fold out for evaluation. Of course, this means that the model needs to be re-trained from scratch for each fold of the k folds, which is computationally expensive. Nevertheless, this method still provides a good enough approximation of the out-of-sample error.

2.4.2 Learning Process

NNs are universal function approximators, meaning that any function can in theory be approximated by an NN (Cybenko, 1989, Leshno et al., 1993, Pinkus, 1999). Therefore, the goal of the learning process is to find the set of network weights that can best map the input data to the desired output. The first step towards this goal is to define a measure of how well the network performs throughout the learning process. This measure can then be used to adjust the weights of the network until an optimal set of weights has been found. In the remainder of this section, these ideas will be discussed in more detail.

2.4.3 Loss function

The goal of the optimisation process is to find the weights that best approximate the function that maps the input of the network to the desired output. In practice, this is done by defining an objective function—often referred to as cost function or loss function—that measures how well the network predicts the output given the input and the value of its weights. The goal of the learning process is to iteratively modify the network weights to minimise the value of the objective function.

The loss function can take different forms depending on the task that needs to be solved by the network. Here we introduce the most common and widely used objective for optimising classification NNs: the **cross-entropy loss function**. For binary

classification problems, the cross-entropy is given as follows:

$$L(w; x_i; y_i) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)),$$

where $y_i \in \{0, 1\}$ denotes the true label of sample i and $\hat{y}_i \in [0, 1]$ the prediction made by the network for input i, that is a probability value between 0 and 1. Overall, this loss function penalises predictions that are close to the either of the two extremes (i.e. 0 or 1), thus indicating high confidence, but are in fact incorrect.

When a sample can belong to one of many (more than 2) classes, i.e. in multi-class classification tasks, the *categorical cross-entropy* is used instead:

$$L(w; x_i; y_i) = -\frac{1}{N}\sum_{i=1}^{N} log(\hat{y}_{i,c[i]})$$

2.4.4 Optimisation

When training an NN, the goal is to find the network weights that minimise the loss function. This is known as the process of optimisation. The most popular approaches to this problem are gradient-based methods, which iteratively estimate the loss function over the training set, compute the gradient of the weights with respect to the loss and adjust the weights in the opposite direction of the gradient. There exist different optimisation approaches depending on how the update is defined and how the gradient of the cost function with respect to the network weights is computed (Goldberg, 2017).

Stochastic Gradient Descent

SGD is an iterative optimisation method based on gradient computation. Unlike batch gradient descent—the vanilla version of the gradient optimisation algorithm which calculates for one update the gradient over the entire training dataset—stochastic gradient descent (SGD) computes a stochastic approximation of the gradient using a subset of the train data (a so-called mini-batch). As this approximation helps speed up iterations, SGD is more computationally efficient, especially for high-dimensional problems, than its classical counterpart.

More specifically, the update looks as follows at each time step of the optimisation process:

$$w = w - \eta \cdot \nabla_w L(w; x_{i:i+n}; y_{i:i+n}) \quad , \tag{2.4}$$

where L is the loss function to minimise, w denote the weights of the NNs and η is a

hyper-parameter, the so-called **learning rate**.

Selecting a good value for the learning rate can be challenging. If the value is set too high, the optimisation algorithm will struggle to converge: it might bounce around the minimum without ever reaching it or it might even diverge. However, if the value is set too low, convergence will be extremely slow. In addition, it might be more beneficial to perform a larger update for important features that are less frequent rather than applying the same learning rate to all parameter updates. To deal with these issues an adaptive learning rate is sometimes used instead of a constant one Duchi et al. (2011), Tieleman et al. (2012), Kingma & Ba (2015), Dozat (2016).

Adam

Adam (adaptive moment estimation) is probably the most popular optimisation approach that uses an adaptive learning rate (Kingma & Ba, 2015). As its name suggests, Adam relies on the computation of estimates of the gradient moments for updating network weights. The first moment (i.e. the mean) estimate m_t and the second moment estimate v_t (i.e. the uncentred variance) which correspond to the exponential moving average of gradients and squared gradients. They can be computed as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_w L(w_{t,i}) \tag{2.5}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_w L(w_{t,i}))^2 \tag{2.6}$$

$\nabla_w L(w_{t,i})$ denotes the partial derivative of the loss function with respect to parameter w_i at time t. Based on these moments, the Adam update rule is the following:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \quad ,$$

where $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ and $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$ are the corrected versions of the first and second moment estimates respectively which offset their bias towards zero. The authors suggest using $\beta_1 = 0.9$, $\beta_2 = 0.999$ to control the decay rates and setting $\epsilon = 10^{-8}$.

Backpropagation

The so-called backpropagation algorithm was introduced in 1986 by Rumelhart et al. (1986) as an effective way of computing the gradients of the loss function—i.e. the partial derivative with respect to the network weights. The algorithm works as follows. First, the loss function is computed based on the true output and the network's predicted output given a set of training data (forward propagation). Then, the loss is backpropagated through the network layer after layer, starting from the output layer

all the way back to the input layer using the chain rule to get the gradient of the loss with respect to each weight. At each layer, the values of the weights in this layer are updated using the gradients. The idea of going backwards makes it possible to reuse the intermediate terms of the chain rule, instead of recomputing them at each layer. Because it is efficient, this algorithm enables the use of gradient-based algorithms, such as SGD, to optimise the weights of DNNs.

2.5 Evaluation

In this section, we discuss the evaluation of DL models, including popular measures of performance and issues that might arise and might impact generalisation error.

2.5.1 Performance Measures

An important structure for assessing the performance of binary classification models is the *confusion matrix*. It summarises the number of correctly and incorrectly classified samples: rows refer to true class labels whereas columns refer to predicted class labels. Thus, the matrix contains, in each entry respectively:

- the number of True Positives (TPs), i.e. the number of positive samples that have been correctly classified,
- the number of False Negatives (FNs), i.e. the number of positive samples that have been incorrectly classified as negative,
- the number of False Positives (FPs), i.e. the number of negative samples that have been incorrectly classified as positive, and finally
- the number of True Negatives (TNs), i.e. the number of negative samples that have been correctly classified.

The confusion matrix is illustrated in Figure 2.1.

The confusion matrix can then be used to calculate an array of evaluation measures. A commonly used metric for evaluating classification models is the **accuracy**. It corresponds to the fraction of correctly classified samples and can be computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

Three other metrics frequently used to evaluate machine learning classification models are **precision (P)**, **recall (R)** and **F1-score (F1)**. The precision is defined as the

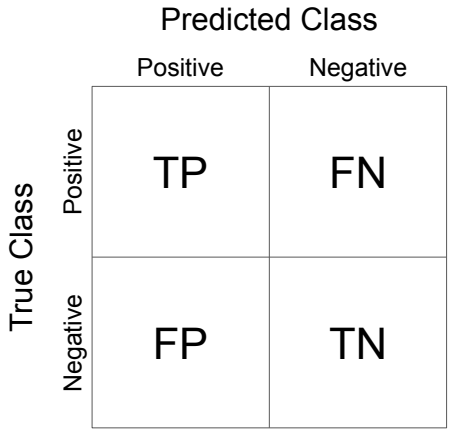


Figure 2.1: The confusion matrix: a popular representation of a binary classification model's performance.

number of correctly identified instances divided by the total number of instances identified. In contrast, the recall is computed as the number of correctly identified instances divided by the total number of correct instances. Finally, the F1-score is defined as the harmonic mean between the precision and the recall. Formally, this is equivalent to

$$\text{Precision} = P = \frac{TP}{TP + FP} \tag{2.7}$$

$$\text{Recall} = R = \frac{TP}{TP + FN} \tag{2.8}$$

$$\text{F1} = \frac{2PR}{P + R} \quad . \tag{2.9}$$

2.5.2 Overfitting

In Section 2.4.1, we have mentioned the importance of relying on disjoint sets of data for training, development and evaluation. Indeed, in order to best estimate the performance of a model, it should eventually be evaluated independently on data that had not been seen during training and development. One of the common problems when training DL models is that if the model is too complex and the number of training examples is too small, the network might be able to learn to fit idiosyncracies in the training data perfectly rather than more general patterns, which means that it will achieve great performance on samples used for training but low performance on unseen data. Such a model is referred to as overfitting.

In general, in machine learning, the problem of finding a model that explains the data well can be viewed as the problem of balancing the model's variance error and

bias error, a concept called *bias-variance trade-off*. The variance captures the model's variability to changes in the training data, whereas the bias reflects the model's capacity to learn from the training data. Therefore, models with high bias make too simplistic assumptions about the data and, thus, fail to accurately capture the patterns underlying the data. Such models achieve high training error, and their validation error is of the same magnitude. Conversely, models with high variance pay too close attention to the data and are unable to generalise well to unseen data: small fluctuations in the training data lead to drastically different predictions. Such models show lower training error but much higher validation error. Overall, overfitting is the characteristic of models with high variance and low bias, whereas models with high bias but low variance are said to be *underfitting*.

An optimal model should have low variance and low bias. Yet, as the name "bias-variance trade-off" conveys, reducing the bias often increases the variance and vice versa. A common strategy for reducing the bias error consists in increasing the model size, as a higher number of parameters helps to learn more complex patterns. However, if the model has too many parameters, its variance error might increase as the model can learn to fit the training data perfectly by also learning underlying noise patterns that are specific to the training examples. In contrast, lower variance can be achieved by adding regularisation during training. Regularisation is a set of techniques designed to reduce the generalisation error. One of the most common forms of regularisation for DNNs is dropout Srivastava et al. (2014), a method which drops hidden units at random during training to prevent the network from relying too heavily on any hidden unit. Nevertheless, too much regularisation inevitably increases the model's bias error as the model complexity decreases Gao (2021).

One easy way to counteract the increase in variance error when increasing the model size is to also increase the size of the training dataset. A higher the number of training samples helps decrease the variance. However, getting more annotated examples is often expensive, time-consuming and challenging. This is particularly true for clinical text data, as mentioned in Chapter 1. Hence, a common approach consists in generating synthetic training samples that are different from real training data but follow the same distribution. This approach, which is called *data augmentation*, will be investigated in Chapter 6.

2.5.3 Conclusion

In this chapter, we proposed a brief overview of DL techniques for NLP along with examples of how these approaches have been used in the context of clinical text mining. In the following chapters, we look at some practical applications of DL approaches

in more details: we will start with the identification of rare clinical events, followed by the automatic disambiguation of abbreviations and acronyms, and finish with the augmentation of training samples. Whereas, in this chapter, we have only touched upon possible applications, the following chapters will contain a more extensive section that highlights related work specific to the application discussed.

Chapter 3

Identification of Rare Events

The issue of small datasets, which is common in the clinical domain, is amplified when dealing with rare medical events. Yet the success of DL models largely depends on the identification and analysis of repetitive patterns that occur within statistical significance thresholds. The recognition of such patterns is further exacerbated by the variation of the language used by healthcare professionals who often use different terms to refer to the same concept. Indeed, the most commonly found type of term variants in the English scientific corpus are semantic variations (Jacquemin, 2001) (e.g. *benign tumors* vs *benign neoplasms*).

In order to test our main hypothesis—namely that DL can be successfully applied for clinical text mining, we have identified the first RQ: Is it possible to develop an effective DL strategy to recognise references to rare clinical events? To answer that question, we have chosen SAEs as a case study for rare clinical events and defined the following RO: to combine the implicit semantic knowledge captured by pre-trained LMs with explicit domain knowledge to compensate for the limited size and diversity of clinical text datasets and the diversity of medical terminology.

In this chapter, we tackle the identification of SAEs within the narrative section of clinical trial report forms. To that end, we use real-life data from the Centre for Trials Research (CTR) in Wales. As expected, these data are extremely limited with a total of 286 documents consisting of 37 tokens on average.

There are two main ways of formulating the problem of SAE identification and their mapping to unique concept. First, a NER model can be trained to identify mentions of SAEs within the text. Then, to make these mentions amenable to statistical analysis, they need to be normalised to a unique concept that represents the corresponding SAE. Alternatively, clinical events—SAE or not—can first be extracted from text using a predefined list of concepts and their synonyms. Then, a classifier can be trained to differentiate between SAEs and other types of events. We posit that the latter approach

is better suited for small datasets as it breaks down the complexity of the problem into two simpler tasks: (1) the identification of all clinical events, for which explicitly available domain knowledge can be leveraged and (2) the classification of events into SAEs depending on the context, for which the implicit language understanding knowledge of a context-aware pre-trained LM, such as BERT, can be exploited. In this chapter, we compare the two approaches.

The work detailed in this chapter has been published in the journal JMIR Medical Informatics under the title *Text mining of adverse events in clinical trials: Deep learning approach* (Chopard et al., 2021a).

3.1 Background

Modern health care is associated with increased costs and broad-reaching variations in care and outcomes across the global population. The provision of evidence-based health care is a critical priority for users, providers, and policy makers alike. The systematic and high-quality conduct of clinical trials is critical for the development of clinical guidance to inform evidence-based practice. Pharmacovigilance and safety reporting are among the most important aspects of the conduct of clinical trials. This is relevant to all clinical trials in which the benefit or harm must be fully established before any intervention or medicinal product is adopted.

Pharmacovigilance and safety reporting provide the basis for ensuring clinical trial participant safety and good research practice. It involves processes for monitoring the use of medicines or interventions in clinical trials. It has a critical role in the identification of previously unrecognised adverse events or changes in the patterns of adverse events. It is also relevant to the assessment of the risks and benefits of medicines or interventions to determine what action, if any, is needed to improve their safe use.

As mentioned in Chapter 1, an adverse event is any untoward medical occurrence in a participant to whom a medicinal product has been administered, including occurrences that are not necessarily caused by or related to the administered product. An SAE is any untoward medical occurrence that, at any dose, results in death, is life-threatening, requires inpatient hospitalisation or causes prolongation of existing hospitalisation, results in persistent or significant disability or incapacity, or comprises a congenital anomaly or birth defect. Early detection of unknown adverse events, reactions, interactions, and an increase in the frequency of (known) adverse events is a key element of the pharmacovigilance and safety process. Provision of up-to-date information on adverse events to health care professionals, researchers, and regulatory bodies contributes to the assessment of benefit, harm, effectiveness, and risk of the intervention, thus advancing

their safe, rational, and more effective (including cost-effective) use. In multicenter non-commercial clinical trials conducted in the United Kingdom, the SAE reporting requirements are detailed in the trial protocol, and the principal investigators at NHS sites are responsible for reporting SAEs to the coordinating clinical trial unit (CTU) for an assessment of the seriousness, causality, and expectedness as delegated by the clinical trial sponsor. An SAE report includes an event term and additional signs and symptoms in a narrative. The narrative is reported by a physician during their medical assessment of the event. The report is then reviewed by a central CTU reviewer to assess any potential causal relationship with the trial drug. Each narrative is reviewed as a single report. The narratives are typically received from sites as paper records. These are logged electronically in the safety databases by the CTU pharmacovigilance team for the relevant national competent authorities (e.g. the UK Medicines and Health Care Products Regulatory Agency or European Medicines Agency). The reports are searchable on request and subject to appropriate regulatory permissions. There is now a clear recognition of the potential for artificial intelligence in safety case management to identify relationships and signals (US Food and Drug Administration, 2018). Although these approaches may be implemented in commercial settings and within competent authorities, such methods for classifying and categorising data are not yet standardised or explicit across non-commercial pharmacovigilance settings.

It is possible that the narrative contains additional adverse events or toxicities that are not coded as additional events and are captured in the narrative only. However, there is no mechanism for the detection of safety signals across individual reports or individual trials and, thus, there is no possibility for early detection of worrying trends. This is particularly the case for toxicities for which reconciliation with the clinical database would be advantageous. Such a tool would facilitate the cross-checking of toxicities recorded in the narrative of the SAE form with those recorded in the trial database, which is currently only feasible if automated. Although these approaches may be used in commercial trial settings, they would not always be used in the public domain simply because of the nature of the drug licensing pathway. This study seeks to use text mining to automatically identify and code adverse events from the narrative sections of SAE reports in clinical trials of investigational medicinal products coordinated by a non-commercial CTU, with the aim of unlocking narrative evidence for further statistical analysis. Although such an analysis is beyond the scope of this study, it would serve to monitor the patterns of adverse events at the cohort level rather than singular adverse events. Owing to their narrative nature, such an analysis cannot be conducted directly on the content of SAE reports.

3.2 Related Work

Chapter 2 presented a review of DL in the context of clinical text mining. In this section, we focus specifically to the problem at hand, namely the identification of SAEs in the narrative section of clinical trial reports, and review how others tackled this problem, not necessarily using DL but also more classical approaches.

Text mining has been used to identify adverse events from a variety of data sources, including spontaneous reporting systems, medical literature, EHRs, and user-generated content on the internet (Wong et al., 2018). The problem of mining adverse events in text has been approached from different angles. Most commonly, it has been defined as a text classification problem, where a piece of text, either an entire document or its part (e.g. an individual sentence), is mapped to one or more predefined class that correspond to a type of adverse event or its property. Some approaches target a specific adverse event such as anaphylaxis and perform simple binary classification with respect to the presence of the event considered (Botsis et al., 2011). Other examples target a range of drugs and use documents that mention them to train a binary classifier with respect to their safety, using an existing watch list of drugs that have an active safety alert posted on the US FDA website (Chee et al., 2011).

In terms of semantics, adverse events are compatible with signs and symptoms. When a dictionary-based method is used to extract such instances, a binary classifier is needed to differentiate between the signs and symptoms that correspond to adverse events and those associated with the underlying diagnosis (Botsis et al., 2012). Along similar lines, when an adverse event is associated with medication, a system is needed to support safety evaluators in identifying reports that may demonstrate causal relationships with the suspect medications. To this end, it has been shown that a binary classifier can be trained to successfully differentiate between two causality categories: certain, probable, or possible versus unlikely or unassessable (Han et al., 2017). Multifaceted classification can be performed to identify additional properties of an adverse event, for example, temporal (historical or present), categorical (assertive, hypothetical, retrospective, or a general discussion), and contextual (deduced or explicitly stated) (Iqbal et al., 2017).

Alternatively, the problem of identifying adverse events can be defined as that of information extraction (Roberts et al., 2017). More specifically, we can differentiate between entity and relationship extraction. Here, the goal of entity extraction is to identify a text sequence that describes an adverse event. Therefore, it can also be viewed as a sequence labelling problem (Nikfarjam et al., 2015, Cocos et al., 2017a, Fan et al., 2021). In addition, the text sequence can be mapped to a relevant dictionary such as the

Medical Dictionary for Regulatory Activities (Duke & Friedlin, 2010, Combi et al., 2018) or the UMLS (Nikfarjam et al., 2015, Emadzadeh et al., 2017). Such normalisation of named entities to standardised identifiers is especially relevant when processing text originating from social media, whose language tends to be highly colloquial (Chee et al., 2011, Nikfarjam et al., 2015, Cocos et al., 2017a, Combi et al., 2018, Emadzadeh et al., 2017, Nikfarjam & Gonzalez, 2011, Sarker & Gonzalez, 2015, Liu et al., 2016).

When multiple medicines are considered, two types of named entities need to be extracted—medicines and adverse events—and additional reasoning needs to be performed to extract a relationship between the two (Iqbal et al., 2017, Liu et al., 2016, Wang et al., 2009). Further statistical analysis can be applied to such pairs to measure the strength of such associations (Wang et al., 2009). Information of interest can be extracted using pattern-matching approaches, where patterns are typically modelled using regular expressions (Iqbal et al., 2017, Duke & Friedlin, 2010, Skentzos et al., 2011). Alternatively, frequent patterns of language for expressing opinions about medications can be learned automatically using association rule mining by considering sentences as transactions and the words in a sentence as items in the transactions (Nikfarjam & Gonzalez, 2011).

Specific methods chosen to mine adverse events from text depend on the way the text mining problem is posed. Typical approaches chosen for text classification include rule-based methods (Botsis et al., 2011, Iqbal et al., 2017, Emadzadeh et al., 2017, Hazlehurst et al., 2009) and supervised machine learning (Botsis et al., 2011, Chee et al., 2011, Botsis et al., 2012, Han et al., 2017, Sarker & Gonzalez, 2015, Negi et al., 2019). A range of machine learning methods has been used, including naive Bayes, support vector machines, random forests, maximum entropy, and logistic regression. On occasion, ensemble learning has been used to improve classification performance by integrating multiple models using methods such as bagging, majority voting, weighted averaging, and stacked generalisation (Chee et al., 2011, Liu et al., 2016, Negi et al., 2019). The different types of lexical, syntactic, and semantic features have been used by the classification algorithms. Lexical features include n-grams (Chee et al., 2011, Sarker & Gonzalez, 2015), context windows (Liu et al., 2016), and lexicon matches (Sarker & Gonzalez, 2015). Typically, syntactic features include part-of-speech (POS) tags, negation, syntactic dependencies, and syntactic functions (Sarker & Gonzalez, 2015, Liu et al., 2016, Negi et al., 2019). Semantic features are either based on external sources such as the UMLS, PubChem, or DrugBank (Sarker & Gonzalez, 2015, Liu et al., 2016, Hazlehurst et al., 2009, Wang et al., 2019a) or manually engineered (Chee et al., 2011, Botsis et al., 2012, Han et al., 2017, Iqbal et al., 2017). Other used features were based on sentiment polarities (Chee et al., 2011, Sarker & Gonzalez, 2015) and

topic modelling (Sarker & Gonzalez, 2015). A few examples of using feature selection methods include bi-normal separation (Chee et al., 2011) and information gain (Liu et al., 2016).

Finally, approaches chosen to address adverse event mining as a sequence labelling problem include conditional random fields (CRFs) (Nikfarjam et al., 2015, Tao et al., 2017) and, more recently, NNs (Negi et al., 2019, Wang et al., 2019a), including RNNs (Cocos et al., 2017a) and LSTMs (Cocos & Masino, 2017), which outperformed CRFs. For best results, a bidirectional LSTM is combined with CRF Fan et al. (2021), Belousov et al. (2017), Dandala et al. (2017), Gu et al. (2017a,b), Xu et al. (2017). Most approaches used word embeddings, which represent words as meaningful real-valued vectors of configurable dimensions learned automatically from a large corpus based on their co-occurrence using methods such as Word2Vec (Wang et al., 2019a, Gu et al., 2017a), fastText (Cocos & Masino, 2017), and GloVe (Pawar et al., 2017)). As mentioned in Chapter 1, word-embedding models generate a single embedding for each word, thus conflating homonyms in the corresponding vector space, unlike transformers such as BERT which captures contextual relationships in a bidirectional way to contextualise the embedding of any given word based on the surrounding words. For example, BERT has been used to model adverse event extraction as an NER task (Fan et al., 2021, Du et al., 2021). The topics of word embedding and BERT, in particular, will be revisited later in this chapter in the context of motivating and describing our own approach to this problem.

The after-the-fact nature of text data collected from sources such as spontaneous reporting systems, medical literature, EHRs, and social media naturally gives rise to postmarketing surveillance applications (Wong et al., 2018, Luo et al., 2017). However, pharmacovigilance starts by collecting safety information derived from randomised controlled trials. Our review of text mining applications related to the identification of adverse events revealed that this source of data was underrepresented. This study addresses this gap by using SAE report forms collected during clinical trials as the primary source of data. Given that each trial focuses on a specific medicinal product, the problem is somewhat simplified as the need to extract information about the product itself is obviated. This also makes it more natural to define it as a multi-label text classification problem rather than an information extraction problem. Using the UMLS as our classification scheme, the main aim is to map each document to a set of coded adverse events. The main difficulty of the problem lies in differentiating between signs and symptoms associated with the underlying condition and those that represent adverse events. The fact that both types of references to signs and symptoms can be found within a single SAE report, often within the same sentence, renders a BOW approach

unsuitable. Instead, we opt for a DL approach. Instead of LSTM approaches, which seem to dominate in our review of the related work, we opt for transformers, which tend to outperform RNNs on a variety of natural language processing tasks.

3.3 Methodology

3.3.1 Data Provenance

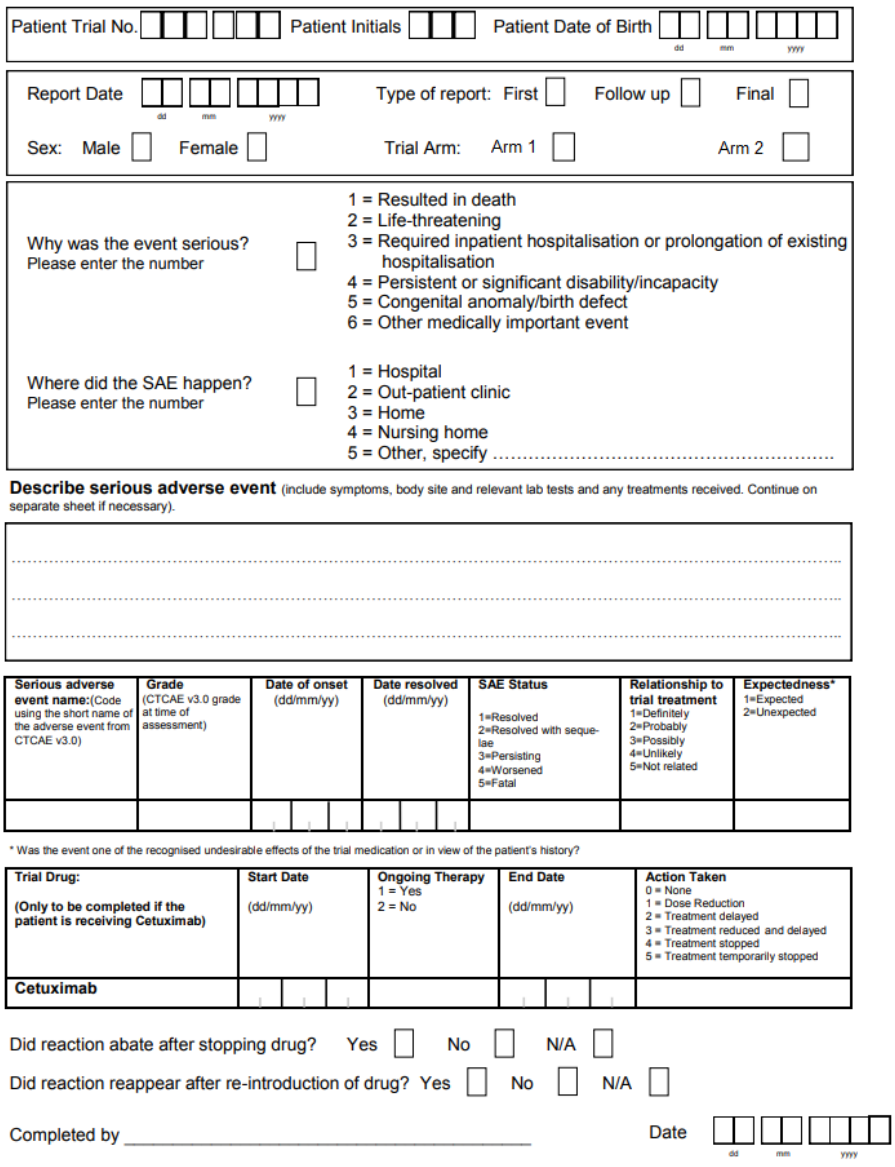
Data were provided by the CTR, the largest group of academic (noncommercial) clinical trial staff in Wales. Their portfolio of work includes drug trials and complex interventions, mechanisms of disease and treatments, cohort studies, and informing policy and practice in partnerships with researchers across the United Kingdom and worldwide. Across all these trials, standard procedures are put in place to monitor and manage safety reporting and SAE in line with the regulatory requirements for research.

Clinical trials SAE report forms (Figure 3.1) are completed by research nurses and physicians at hospital or clinical trial sites and submitted as PDF documents to the CTR central safety team for management and processing. They contain data on the SAE and a narrative description of the event. The narrative is used by the reviewer to help assess causal relationships with the trial drug but is not entered into the trial database and is not used in any analysis of the events. Completed SAE reports are then sent for review by a physician and, depending on the outcome of the review, are logged in the safety databases for the regulatory authorities, ethics committees, and drug companies.

Although narratives in noncommercial settings, such as CTR, can be digitised, this does not currently take place at the point of initial SAE reporting, as electronic data capture for the SAE report is associated with additional regulatory challenges, primarily because of the requirement for signature verification by a physician and a contemporaneous changelog. Clinical trial staff reviewing SAE reports are, thus, unable to systematically analyse the information provided in the narrative, missing an opportunity to identify the trends and potential safety signals. If the text mining approach were to identify additional safety events and signals not detected through standard reporting, processes could be altered to improve work practices at the level of a noncommercial CTU pharmacovigilance team. This study aims to assess the feasibility of text mining in the context of such an analysis. The findings could affect the way regulatory narratives are reviewed and analysed, for example, noncompliances or audit findings.

3.3.2 Data Collection

Data were collected from 6 ongoing clinical trials, as described in Table 3.1.



Patient Trial No. ☐☐☐☐☐ Patient Initials ☐☐☐ Patient Date of Birth ☐☐ ☐☐ ☐☐☐☐ (dd mm yyyy)

Report Date ☐☐ ☐☐ ☐☐☐☐ (dd mm yyyy) Type of report: First ☐ Follow up ☐ Final ☐

Sex: Male ☐ Female ☐ Trial Arm: Arm 1 ☐ Arm 2 ☐

Why was the event serious? Please enter the number ☐

1 = Resulted in death
2 = Life-threatening
3 = Required inpatient hospitalisation or prolongation of existing hospitalisation
4 = Persistent or significant disability/incapacity
5 = Congenital anomaly/birth defect
6 = Other medically important event

Where did the SAE happen? Please enter the number ☐

1 = Hospital
2 = Out-patient clinic
3 = Home
4 = Nursing home
5 = Other, specify .....

Describe serious adverse event (include symptoms, body site and relevant lab tests and any treatments received. Continue on separate sheet if necessary).

.....
.....
.....

Serious adverse event name:(Code using the short name of the adverse event from CTCAE v3.0)	**Grade** (CTCAE v3.0 grade at time of assessment)	**Date of onset** (dd/mm/yy)	**Date resolved** (dd/mm/yy)	**SAE Status** 1=Resolved 2=Resolved with sequelae 3=Persisting 4=Worsened 5=Fatal	**Relationship to trial treatment** 1=Definitely 2=Probably 3=Possibly 4=Unlikely 5=Not related	**Expectedness*** 1=Expected 2=Unexpected
		/ /	/ /			

* Was the event one of the recognised undesirable effects of the trial medication or in view of the patient's history?

Trial Drug: **(Only to be completed if the patient is receiving Cetuximab)**	**Start Date** (dd/mm/yy)	**Ongoing Therapy** 1 = Yes 2 = No	**End Date** (dd/mm/yy)	**Action Taken** 0 = None 1 = Dose Reduction 2 = Treatment delayed 3 = Treatment reduced and delayed 4 = Treatment stopped 5 = Treatment temporarily stopped
Cetuximab	/ /		/ /	

Did reaction abate after stopping drug? Yes ☐ No ☐ N/A ☐

Did reaction reappear after re-introduction of drug? Yes ☐ No ☐ N/A ☐

Completed by _____ Date ☐☐ ☐☐ ☐☐☐☐ (dd mm yyyy)

Figure 3.1: An SAE reporting form. CTCAE: Common Terminology Criteria for Adverse Events; N/A: Not Applicable.

Ethical review and approval were waived for this study as this study involved the use of secondary SAE data that were fully deidentified. All involved trials were conducted according to the guidelines of the Declaration of Helsinki and approved by the relevant research ethics committees. All chief investigators from these trials were consulted, and sponsor agreement was obtained for the use of the data in this secondary research study. Participant consent was also waived for the reasons stated above.

A subset of SAE reports was sampled randomly from each trial, giving a total of 286 reports. Phases 1 and 2 were early phases with a smaller number of participants. The fewer numbers of reported SAEs were a function of the smaller numbers of participants compared with phase 3; hence there were variations in the number of documents across the 6 trials.

The original SAE reports were pseudoanonymised at the point of extraction from the system by obscuring any links between the patient and their individual records. The narrative sections of the SAE reports were then transcribed and saved as Microsoft Word documents. The transcription process was extended to include deidentification by obscuring any personally identifiable information in a way that minimises the risk of unintended disclosure of the identity of individuals and information about them. The transcribed documents were an average of 37 (standard deviation (SD) 24) tokens long.

3.3.3 Data Annotation

The aim of this task was to annotate adverse events in the transcribed versions of the SAE report forms. For the purpose of this task, we clarify the definition of an SAE by the FDA mentioned in Chapter 1 (US Food and Drug Administration, 2016) and define an adverse event as any unfavourable or unintended disease, sign, or symptom (including an abnormal laboratory finding) that is temporally associated with the use of a medical treatment or procedure, which may or may not be considered related to the medical treatment or procedure. Such an event could be related to the intervention, dose, route of administration, or patient or caused by an interaction with another drug or procedure.

The annotation guidelines prescribed the scope of the annotation task as follows: (1) focus only on adverse events that have occurred in the present or past, that is, ignore hypothetical or future events; (2) annotate the entire phrase that describes an adverse event; and (3) if the same adverse event were mentioned multiple times, then annotate every mention. The annotation process was based on the following instructions: (1) identify an adverse event that is mentioned in the narrative, (2) select the text that describes the adverse event, and (3) highlight the selected text. The text editing operations were performed using Microsoft Word, which was preferred over a specifically

Table 3.1: Clinical trials from which data were collected. MRI: magnetic resonance imaging.

ID	Description	Documents, n
Trial-1	A phase 2 study of neoadjuvant chemotherapy given before short-course preoperative radiotherapy as treatment for patients with MRI-staged operable rectal cancer at high risk of metastatic relapse	5
Trial-2	A phase 1b/2 randomised placebo-controlled trial in postmenopausal women with advanced breast cancer previously treated with drug A	7
Trial-3	A randomised phase 3 clinical trial investigating the effect of drug B added to standard therapy in patients with lung cancer	131
Trial-4	Study of chemoradiotherapy in esophageal cancer, plus or minus drug C	34
Trial-5	A phase 1/2 single-arm trial to evaluate combination drugs for the treatment of advanced cancers, including first-line treatment of patients with advanced transitional cell carcinoma of the urothelium	3
Trial-6	A randomised phase 3, open-label, multicenter, parallel group clinical trial to evaluate and compare the efficacy, safety profile, and tolerability of oral drug X versus intravenous drug Y in the treatment of patients with breast cancer and bone metastases	106

Annotator A	Annotator B
Post oxaliplatin dose patient began to tremor – patient has a history of tremor when feeling nervous. Vomited x 1 . Admitted for observation as patient lives alone. Tremor now resolved.	Post oxaliplatin dose patient began to tremor – patient has a history of tremor when feeling nervous. Vomited x 1 . Admitted for observation as patient lives alone. Tremor now resolved.

Figure 3.2: An SAE report annotated independently by two annotators. Annotations are highlighted in yellow.

designed annotation tool such as BRAT or Bionotate (Neves & Leser, 2014) because of zero installation and training overhead. Microsoft Word supports the bulk selection of text based on its formatting. This functionality was used to export highlighted text as stand-off annotations, which were later used to calculate the inter-annotator agreement.

A total of two annotators independently annotated all the documents. Figure 3.2 provides an example. Here, both annotators annotated two mentions of tremor but did not annotate the historical mention of tremor as it was not temporally associated with the use of the medical treatment that was the subject of the given clinical trial. Further, one reviewer failed to annotate vomiting, leading to disagreement, which was later resolved through discussion. To identify all such cases, we compared all annotations automatically and measured the inter-annotator agreement.

The two annotators labelled SAEs as phrases, which were sequences of words whose total number, together with their start and end positions, were not prefixed. Comparing the inter-annotator agreement at the token level, as suggested by Tomanek & Hahn (2009), was not entirely appropriate for two reasons. First, the annotators labelled phrases as sequences of tokens instead of labelling the tokens individually. Therefore, such an approach approximated the original annotation task. More importantly, the number of negative cases (i.e. the tokens that had not been annotated) would inevitably be much larger than the number of positive cases, thus skewing the data. The lack of a well-defined number of negative cases prevented the use of traditional inter-annotator agreement measures such as Cohen's kappa statistic (Deleger et al., 2012). A common way of quantifying inter-annotator agreement in such circumstances is to use information retrieval performance measures instead (Hripcsak & Rothschild, 2005). By treating one annotator's annotations as the gold standard and the other one's as predictions, we calculated the numbers of TPs, FPs, and TNs, as shown in the confusion matrix (Table 3.2). When these values were combined to calculate the F1-score, it no longer mattered which annotator was considered the gold standard as this measure is symmetrical.

Table 3.2: Agreement between two annotators. (TP=True positives, FP=False Positives, FN=False Negatives, N/A=Not applicable)

	Gold positive	Gold negative
Predicted positive	TP = 744	FP = 50
Predicted negative	FN = 98	N/Ad

These values can then be used to calculate the precision, recall and F1-score as follows, using the formulas defined in Chapter 2:

$$P = \frac{TP}{TP + FP} = \frac{744}{744 + 50} = 0.9370 = 93.70\% \tag{3.1}$$

$$R = \frac{TP}{TP + FN} = \frac{744}{744 + 98} = 0.8836 = 88.36\% \tag{3.2}$$

$$F1 = \frac{2PR}{P + R} = 0.9095 = 90.95\% \tag{3.3}$$

An advantage of using information retrieval performance measures to estimate interannotator agreement is that their values can later be used to gauge a system against human-like performance. At $F1 = 90.95\%$, the interannotator agreement was found to be relatively high. A total of 148 disagreements were resolved through discussions to establish the ground truth. As part of the discussions, the agreed annotations of adverse events were coded manually against the UMLS, which integrates multiple terminologies, classifications, and coding standards in an attempt to support the interoperability between biomedical information systems, including EHRs (Bodenreider, 2004). The MetaThesaurus Browser, a web-based search interface, was used to query the UMLS for each annotation to identify the corresponding concept (Figure 3.3). This searching procedure involved checking concept definitions to make sure that the chosen concept matched the sense of the adverse event annotation. Each concept in the UMLS is assigned a concept unique identifier (CUI), which was used to code the corresponding annotation (see Figure 3.4 for examples). Subsequently, the CUI codes were extracted, duplicates were removed, and the remaining CUIs were used as class labels for each document. Table 3.3 provides a statistical summary of the annotated dataset, which contains a total of 995 class labels.

3.3.4 Problem Representation

The aim of this study was to automate the identification of adverse events described in the narrative section of the SAE reports. This goal was cast as a text classification problem. Given a document and classification scheme, the system should label the

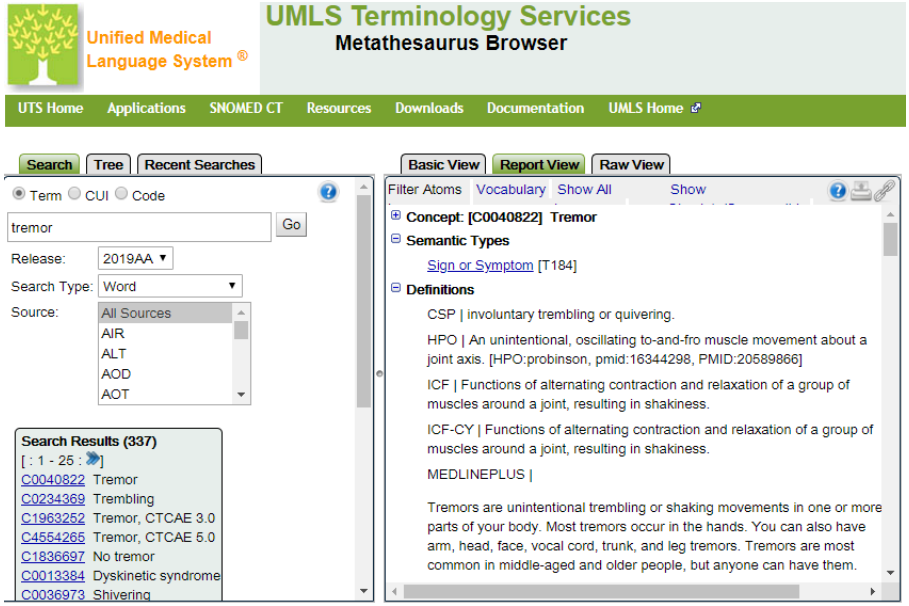


Figure 3.3: Metathesaurus browser search results.

Document	Labels
Post oxaliplatin dose patient began to tremor C0040822 – patient has a history of tremor when feeling nervous. Vomited C0042963 x 1 . Admitted for observation as patient lives alone. Tremor C0040822 now resolved.	C0040822 C0042963

Figure 3.4: Coding of documents against the Unified Medical Language System.

Table 3.3: Statistical properties of the annotated dataset.

Statistical properties	Document length (in tokens)	Annotations	Class labels
Values, minimum	2	1	1
Values, maximum	223	20	19
Values, median	31	3	3
Values, mean (SD)	36.71 (23.77)	3.76 (2.46)	3.48 (2.18)

document with the relevant classes from the given scheme. In our case, the document was an SAE report, a classification scheme was the set of concepts encompassed by the UMLS, and their CUIs were used as class labels. The second column in Figure 3.4 provides an example of the expected output.

To identify the possible adverse events mentioned in a document, the first step involved looking for concepts of the relevant semantic types. In our approach, the UMLS dictionary lookup was restricted to six manually selected semantic types: disease or syndrome, finding, injury or poisoning, neoplastic process, pathological function, and sign or symptom. Some of their mentions could be in the context of medical history and, therefore, not necessarily constitute an adverse event. To differentiate between the two types of mentions, we formulated a binary classification task at the concept level: *given a context, does a specific UMLS concept constitute an adverse event?* Figure 3.5 provides different references to the concept of pleural effusion. For example, the first three references do not constitute adverse events. The first and third mentions of pleural effusion refer to medical history, whereas the second mention is negated. The remaining three mentions of pleural effusion refer to the cause of hospital admissions that prompted SAE reporting.

The practical implementation of such problem representations started with linguistic preprocessing, which was originally developed to support cohort selection from hospital discharge summaries adapted for this study (Spasić et al., 2019). This module involved text segmentation and basic string operations, such as lowercasing, fully expanding enclitics and special characters, replacing a selected subset of words and phrases with their representatives, and, in particular, replacing acronyms and abbreviations with their full forms. Finally, the preprocessed documents were analysed using MetaMap (Aronson & Lang, 2010), a highly configurable dictionary lookup software, to find mentions of UMLS concepts from the six semantic types listed above. MetaMap first uses the SPECIALIST (McCray et al., 1994) parser to get all noun phrases from the text. Second, it generates a list of variants for all the phrases using a table lookup. Third, it builds a set of candidates by extracting all strings containing one of the variants in the UMLS Metathesaurus. Finally, an evaluation function is used to evaluate each of these candidates against the input text, assign them a score, and select the best candidate. Figure 3.6 illustrates a portion of the UMLS dictionary and how it was matched against the input text. As the figure illustrates, a single document might contain multiple adverse events. To support the classification of one adverse event candidate at a time, a separate copy of the given document was saved for each candidate. Each copy anchored a single concept, which may have had multiple occurrences, by marking them up in line. In addition, the text was further regularised by replacing all the concepts with their

Adverse event candidates	Status
Patient admitted from outpatient clinic with empyema of right lung. Patient currently receiving carbo/alimpta chemotherapy. Previous pleural effusion C4012196 . Had pleurex catheter inserted previously for drainage of malignant pleural effusion C4012196 .	☒
Chest x-ray showed no new lesion, no pleural effusion C4012196 or pneumothorax and history of smoking.	☒
Patient was in hospital being treated for pleural effusion C4012196 and during his stay became increasingly short of breath. CT scan showed pulmonary embolism.	☒
Admitted feeling unwell, dry mouth and constipation with high calcium levels 3,83ml/L with left sided pleural effusion C4012196 and slight confusion.	☑
Admitted to hospital on DD:MM:YY following experiencing increasing shortness of breath, at rest, for one week. On admission to hospital, anxious and complaining of chest pain. Had pleural effusion C4012196 .	☑
Admitted to hospital with a pleural effusion C4012196 . Treated and fluid drained.	☑

Figure 3.5: Adverse event identification as a binary classification task. CT: computed tomography

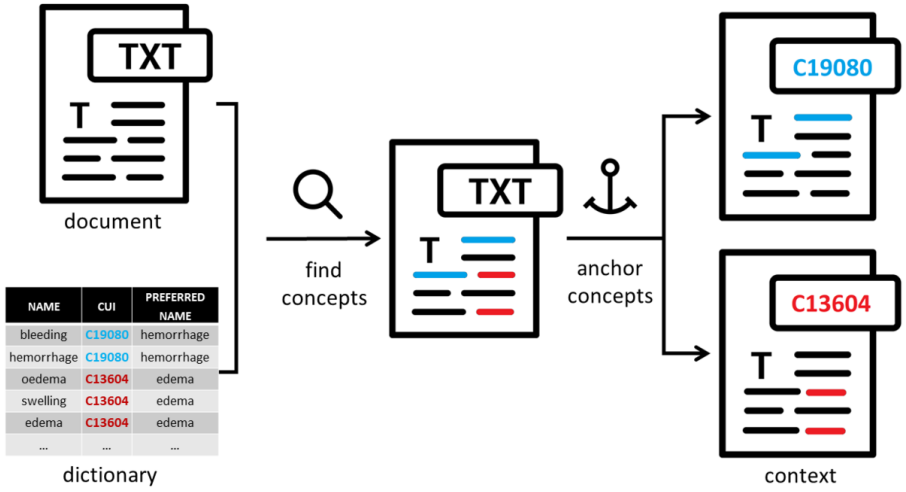


Figure 3.6: Identification of potential adverse event mentions. CUI: concept unique identifier

preferred names. Concept anchoring provided a simple, uniform representation of the potential adverse events, which enabled us to train a single binary classifier based on the context surrounding the anchors.

3.3.5 Classification Rationale

The binary task formulation itself—*given a context, does a specific UMLS concept constitute an adverse event?*—indicates two main types of involved features: extrinsic (context) and intrinsic (concept). Extrinsic features may include the number of mentions within a document, the position within a document, and other words within a fixed-size window. When combined with gold standard annotations, machine learning can be used to discover how to differentiate between positive and negative contexts without having to manually describe the patterns of positive and negative use. For example, by considering the co-occurring words (see Figure 3.7 for examples) and the corresponding annotations, a simple NN can learn to use words such as previous and have as negative and positive modifiers, respectively. By considering a wider context, more complex patterns such as *admitted to hospital* with and *known to have* (see Figure 3.8 for examples) would start to emerge as positive and negative contexts, respectively. Traditionally, such patterns were observed using corpus linguistics methods, which were engineered manually and encoded formally as regular expressions (Spasić et al., 2010). In recent times, NNs are used to automatically capture both short- and long-range dependencies. Similarly, lexical morphology could be explored in an NN approach to learn the patterns of subwords within a concept's name, which were positively or negatively correlated with adverse

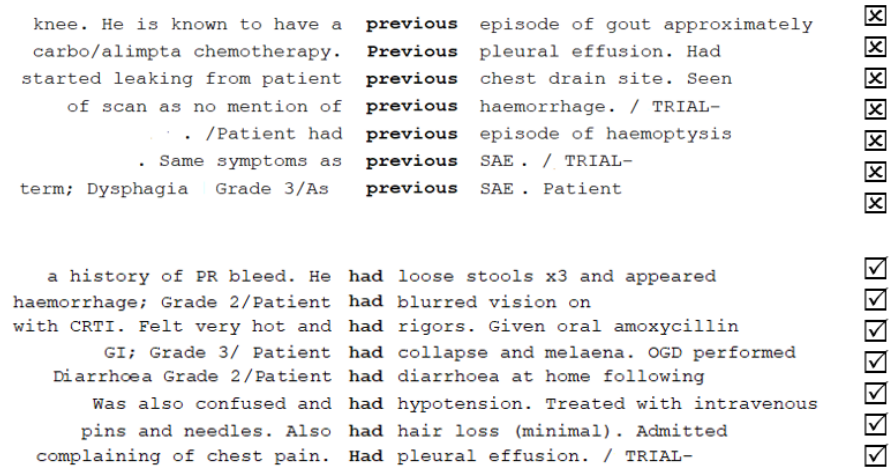


Figure 3.7: Observing the patterns of positive and negative modifiers. CRTI: common respiratory tract infection; GI: gastrointestinal; OGD: oesophagogastroduodenoscopy; PR: per rectum; SAE: serious adverse event.

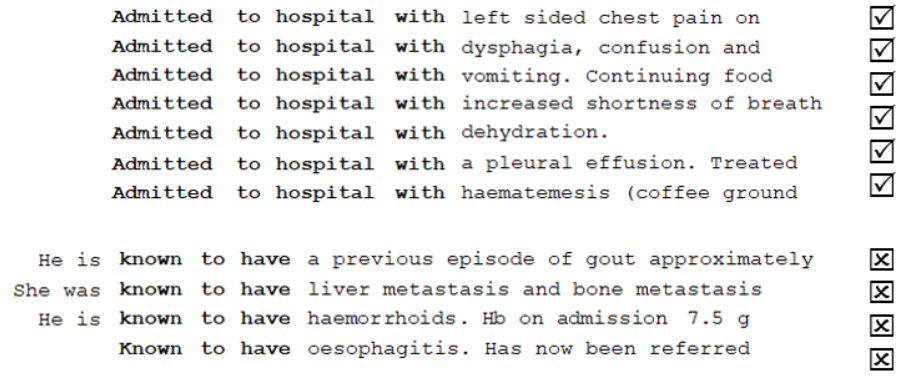


Figure 3.8: Observing more complex patterns of positive and negative use. Hb: hemoglobin.

events. For example, it is reasonable to expect that any concept identified as a potential adverse event that contains the word *chronic* (e.g. *chronic obstructive airway disease* or *chronic infection*) is more likely to refer to a process than a single event. Similarly, any concept whose name contains a word *loss* (e.g. *loss of appetite* or *hair loss*) is more likely to be an adverse event. The words themselves can be analysed for affixes. For example, the prefix hypo- (low or below normal) can be used to increase the likelihood of concepts such as *hypocalcemia* or *orthostatic hypotension* corresponding to adverse events. Similarly, the suffix -*emia* (presence in the blood) can be used to identify concepts such as *cerebrovascular ischemia* or *hyperkalemia* as strong candidates for adverse events. Again, no prior medical knowledge is required to embed such features into NNs, which consider inputs and outputs simultaneously to support end-to-end learning and, hence, bypasses manual feature engineering.

3.3.6 Classification Model

The MLM task was one of the two tasks on which BERT was trained simultaneously. The second task was the NSP task. In addition to [MASK], BERT uses two other special tokens for fine-tuning and specific task training: (1) a classification token [CLS], which indicates the beginning of a sequence and is commonly used for classification tasks (the output associated with this token is used for the NSP task); and (2) a sequence delimiter token [SEP], which indicates the end of a segment.

The embedding layer shown in Figure 3.9 illustrates the input format that BERT expects. Each token's vocabulary identifier is mapped to a token embedding that is learned during training. Next, a binary vector is used to differentiate between two text segments, typically sentences. The type of segment depends on a specific task, for example, in QA both question, and the reference text could be appended and separated by a special delimiter token [SEP]. In our model, we chose the anchored concept as one segment and its context (i.e. the whole document) as another. The binary vector was mapped to a segment embedding using a lookup table, which was learned during training. Finally, local token positions were mapped to positional embeddings using a lookup table, which was updated during training.

The three types of embeddings were added and fed into the pre-trained $\text{BERT}_{\text{BASE}}$ model, which comprises 12 layers of transformer encoders, each having a hidden size of 768 and 12 attention heads. Each layer produces a token-specific output, which can be used as its (contextualised) embedding. Similar to binary classification tasks described in (Devlin et al., 2019), the final transformer output corresponding to the special [CLS] token was taken as an aggregate problem representation, that is, pooled output, and passed on to the classification layer after a 0.1 dropout, which was used to reduce

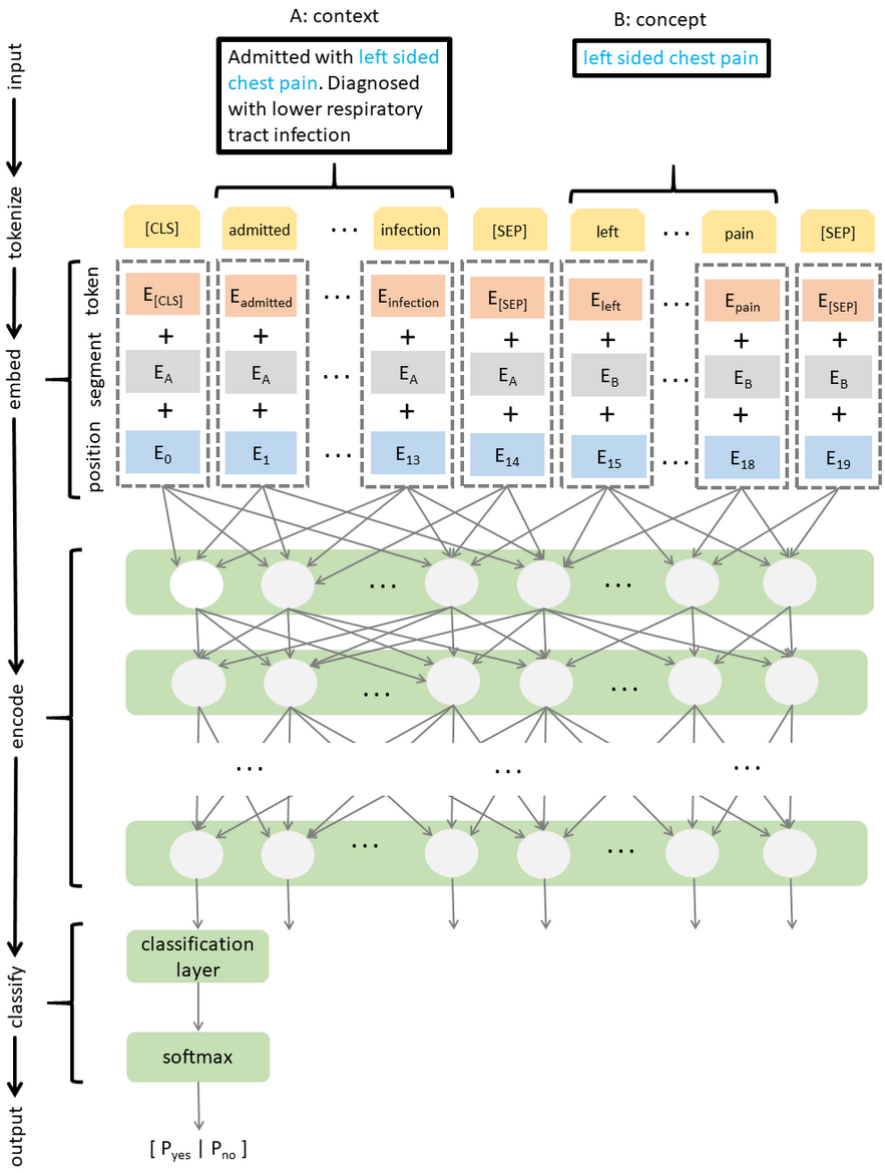


Figure 3.9: Architecture based on BERT for classification of adverse events. CLS: classification token; SEP: sequence delimiter token.

overfitting.

The classification layer reduced the size of the pooled output from 768 to 2, which corresponds to the log-odds (or logits) of the classification output with respect to the question of whether the given concept was an adverse event or not. In contrast to the network up to that point, the classification layer was not pre-trained. Instead, the corresponding weights were learned during BERT fine-tuning. As suggested in the study by Devlin et al. (2019), the weights were initialized using a truncated normal distribution with mean 0 (SD 0.02). A softmax function was then applied to obtain the probability distribution of the two classes. The loss function (softmax cross entropy between the logits and the class labels) was optimised using the Adam optimiser with an initial learning rate of 2×10^{-5} , which was chosen without any fine-tuning, based on the values suggested in the study by Devlin et al. (2019).

The classification model was trained for 8 epochs. This hyperparameter was preselected without any tuning. In each epoch, the training data were looped over in batches of 8 samples. The batch size was limited by memory. All other parameters were kept identical to those in the original $\text{BERT}_{\text{BASE}}$ uncased model, including the clip norm of 1.0, and linear warmup (100 warmup steps with linear decay of learning rate). The system was implemented in TensorFlow (Abadi et al., 2016), an open-source software library for machine learning, with a particular focus on training and inference of DNNs, using the GeForce RTX 2080 (Nvidia Corp) graphics processing unit to accelerate DL.

3.4 Results

During preprocessing, MetaMap was used to extract adverse event candidates. MetaMap failed to extract a total of 118 adverse events from the ground truth. Therefore, these instances automatically constituted FNs. By looking at the concepts that MetaMap did not correctly identify, we noticed two main patterns of errors. The first one is concepts related to test results. For example, unsurprisingly, MetaMap did not map the phrase "oxygen saturation 92% on 15 liters of oxygen" to the concept "Oxygen saturation below reference range". The other primary source of failed extraction is inflexions or phrase variants. For example, the phrases "was constipated", "shaking", and "reduced appetite" were not matched to the concepts "constipation", "tremor", and "decrease in appetite", respectively. This source of error is more surprising as part of the MetaMap algorithm is explicitly defined to generate those alternate forms and use them for matching. Therefore, one could parse a lemmatised version of the text to reduce the number of FNs. Doing so would ensure that inflexions, such as "fatigued", are transformed to their base root form (in this case ", fatigue") and therefore are correctly identified by

MetaMap. The remaining 1021 adverse event candidates extracted by MetaMap were passed on to the BERT-based classification model shown in Figure 3.9. To understand the performance of the BERT classifier, we first focused only on these 995 adverse event candidates before amalgamating them with 118 FNs. Of the 995 candidates, 659 (66.2%) were positive instances (i.e. regarded as adverse events in the ground truth), and 336 (33.8%) were negative instances (i.e. not regarded as adverse events in the ground truth).

We performed 10 independent 5-fold CVs to evaluate the performance of the classification model. In other words, during each CV, 20% of the documents were held out for evaluation, whereas the remaining 80% were used for training, and this was done 5 times in a row, each time using a different fold for evaluation. More specifically, for each of the 10 independent runs, we did the following: The 286 unique document identifiers were first shuffled randomly and then split into 5 folds. Remember that each document may have contained multiple adverse event candidates, and a separate copy was created for each candidate during preprocessing. All copies of the same document shared the same document identifier; hence, there was no overlap of data across the folds. As the splitting was done by document irrespective of the number of events they contained, the actual number of samples (i.e. potential adverse events identified by MetaMap) in each fold may vary. We looped over the folds, each time using a different fold for evaluation and the remaining 4 folds for training. Each time, we measured precision, recall, and F1-scores. Once each of the 5 folds was used for evaluation, we calculated the mean values obtained for each evaluation measure. Finally, these values were averaged over 10 independent runs.

The same CV process was applied to the baseline approach. Remember that the goal of our system was to code adverse events against the UMLS; therefore, a UMLS lookup was inevitable. The lookup itself could be performed as the first step to identify an adverse event candidate (and code it at the same time) and then classify it. Alternatively, it could be performed as the last step to code an adverse event, which was first extracted from free text. In the former approach, we were dealing with a binary classification problem where it needed to be determined whether a given UMLS concept was an adverse event or not. In the latter approach, we were dealing with a sequence labelling problem where the boundaries of a token sequence that referred to an adverse event needed to be determined. This is how Du et al. (2021) approached the extraction of adverse events from safety reports by framing it as the NER problem and fine-tuning BERT for this task. We reimplemented and cross-validated their approach on our dataset to establish the baseline. Although the authors originally used BERT for biomedical text mining (BioBERT) (Lee et al., 2020), we replaced it with BERT in our

Parameters	Baseline approach: NER (BERT) + concept extraction (MetaMap)	Our approach: concept extraction (MetaMap) + classification (BERT)
Precision	57.15 (0.76)	86.38 (0.57)
Recall	71.16 (0.96)	76.04 (1.21)
F1-score	63.35 (0.72)	80.80 (0.71)

Table 3.4: Evaluation results. Both the mean and the SD over a 5-fold CV are reported (in [%]).

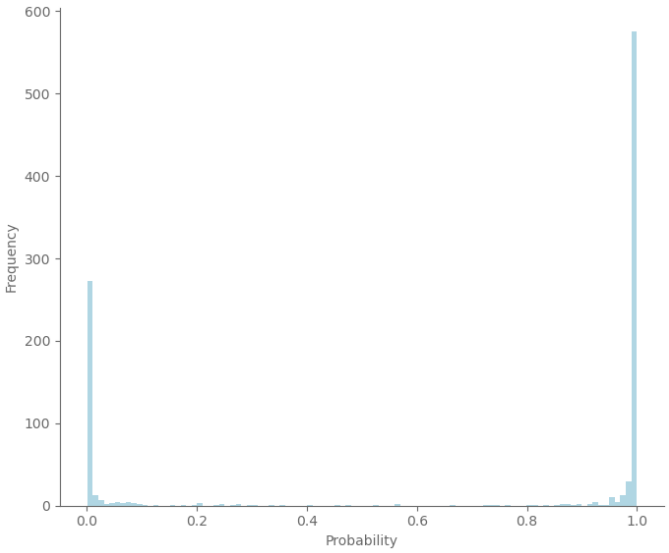


Figure 3.10: Distribution of prediction probabilities for all folds in a CV experiment.

experiments to make their approach directly comparable with ours. The results achieved by the two contrasting approaches are presented in Table 3.4.

Despite the similarities in the underlying technologies, we can observe a notable difference in the performance of the two approaches, most prominently in terms of precision, where we can see an improvement of approximately 30 percent points over the baseline. A detailed analysis of this phenomenon is provided in the Discussion section. In this section, we proceed to describe the results achieved using our own approach.

Figure 3.10 displays the distribution of the prediction probabilities. The histogram combines the predictions from all folds used for CV. We can observe that most prediction probabilities are concentrated around the two extremes, 0 and 1, which suggests that the classification model is able to make clear-cut decisions, as it does not depend on a specific threshold.

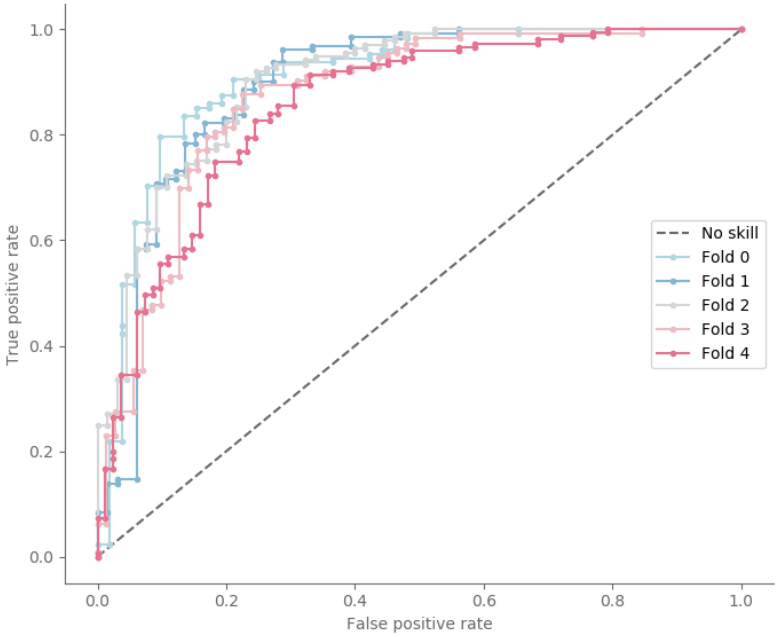


Figure 3.11: Receiver operating characteristic curve for each fold in a CV experiment.

In Figure 3.11, we used receiver operating characteristic curves to illustrate the diagnostic ability of the classification model. A separate curve was provided for each of the 5 folds used for CV. The plot shows the TP rate versus the FP rate at each classification threshold. The solid-coloured lines correspond to the model's performance, whereas the gray dashed line represents the performance of a classifier with no skill, that is, the one that always predicts the majority class. An ideal model would result in a curve that bows toward the coordinate (1,0). With its curve consistently lying close to the top-left corner, our model demonstrated very good classification performance. We summarised the receiver operating characteristic results by calculating the area under the curve to measure the ability of our model to distinguish between the 2 classes, with higher values indicating better performance. With an overall mean score of 87.89% (SD 1.01%) and a range between 0 and 1, our model was clearly able to distinguish between adverse events and underlying conditions 87.79% of the time on average.

Finally, to account for the class imbalance, we also looked at the precision-recall (PR) curve shown in Figure 3.12. Again, the solid-coloured lines correspond to our model's performance, whereas the gray dashed horizontal line corresponds to a model with no skill, that is, a model whose precision is equal to the proportion of positive samples. The PR curve of our model was relatively close to that of an ideal model, whose curve would bow toward the coordinate $(1, 1)$. In comparison to a no skill model, which would achieve a PR area under the curve score of 65.33%, our model reached a high score

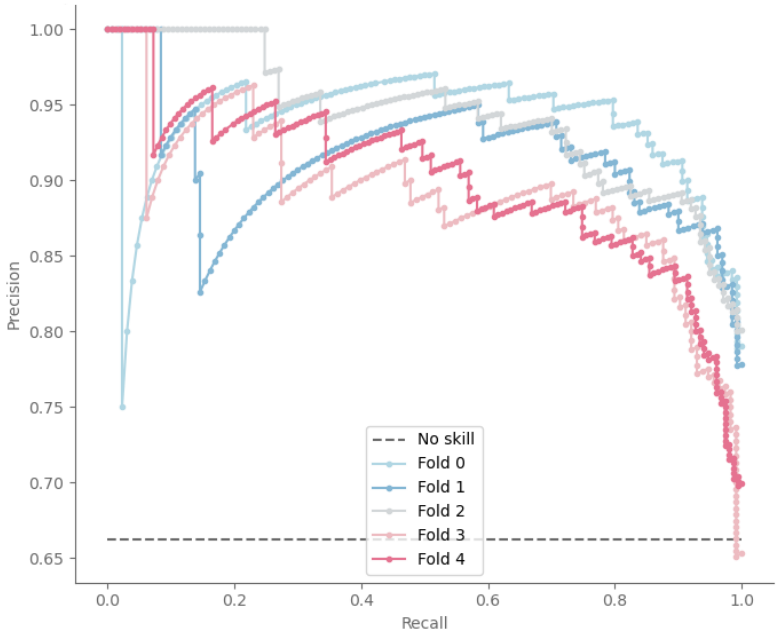


Figure 3.12: Precision-recall curve for each fold in a CV experiment.

of 91.08% (SD 1.03%), demonstrating its ability to correctly classify adverse events despite the class imbalance.

3.5 Discussion

3.5.1 Principal Findings

Previously, we provided details on calculating the interannotator agreement using precision, recall, and F1-score. When a system is evaluated against the ground truth, the corresponding values establishthe human performance baseline, which in this case were $P = 93.70\%$, $R = 88.36\%$, and $F1 = 90.95\%$. If we compare these values against the results provided in Table 3.4, we can observe a 10.15 percent points difference in the F1-score. In particular, we notice that the system's recall is 10.34 percent points lower than its precision. There are two potential sources of type 2 errors in the system. Remember that the system first uses MetaMap to identify potential adverse events, which are then classified by BERT as positive or negative. Both components can give rise to FN results. First, any adverse event that MetaMap failed to forward to BERT would have been automatically counted as a FN. Second, any adverse event that MetaMap did supply to BERT for further classification could have still ended in a FN. MetaMap is a predefined rule-based system, and as such, its performance within our system is limited by external factors. BERT, on the other hand, has been trained for a specific

Parameters	NER (BERT)	Classification (BERT)
Precision	74.84 (0.66)	86.51 (0.53)
Recall	82.37 (0.86)	89.74 (1.04)
F1-score	78.35 (0.53)	88.02 (0.44)

Table 3.5: BERT performance. Both mean and SD over 5-fold CV are reported (in [%])

task using the dataset described here. Therefore, it is worth focusing specifically on its classification performance.

To evaluate how well BERT learned to classify adverse events, we removed those FNs from the ground truth that were never actually classified by BERT because of MetaMap failing to identify them in the first place. Table 3.5 provides the CV results for BERT's performance alone. We observe that the classification performance alone is much closer to the human performance baseline, lagging behind the F1-score by only 2.93 percent points.

If we now compare BERT's classification performance given in Table 3.5 with the overall system performance given in Table 3.4, we can see that the precision is almost identical (86.38 vs 86.51), whereas the recall differs by 13.70 percent points (76.04 vs 89.74). Hence, we can conclude that the recall of the overall system is primarily limited by MetaMap's performance, which naturally raises the question of whether its use as a preprocessing step within our system was appropriate. The baseline method uses MetaMap as the postprocessing step; therefore, we investigated the extent of its effect on the overall performance by singling out BERT's performance on the NER task, which was evaluated using the exact matching of phrases annotated in the ground truth. If we compare the first column of Table 3.5 with the second column of Table 3.4, we can observe that without MetaMap, BERT can certainly achieve higher recall (82.37% vs 76.04%) when it is allowed to determine the phrase boundaries on its own rather than having them prescribed by MetaMap.

Although such an approach is unarguably more flexible, it can also have a negative impact when the goal of the system is to code adverse events rather than only recognise their mentions in the text. If the phrase boundaries are not correctly detected as part of the NER task, then searching the UMLS using an incorrectly extracted phrase may provide an incorrect code. Consider, for example, two adverse events, respiratory tract infection (whose code in the UMLS is C0035243) and urinary tract infection (whose code is C0042029). Suppose that a system failed to correctly identify their boundaries, for example, by suggesting tract infection in both cases. The UMLS has no concept referring to tract infection; therefore, MetaMap would at best suggest infection

(whose code is C3714514) as the closest concept matching the given search term, thus incorrectly coding both respiratory tract infection and urinary tract infection, resulting in 2 FNs (labeled C0035243 and C0042029 in the ground truth) and 2 FPs (both labeled C3714514 by the system). On the other hand, MetaMap can be configured to recognise the longest phrases from relevant semantic types and, in that way, impose tighter control of the process, reducing the number of both FPs and FNs. Although MetaMap may limit the recall, it does play an important role in controlling the precision in our proposed approach, as the results in Table 3.4 clearly depict. Nonetheless, MetaMap could benefit from revising its rule-based dictionary lookup approach in light of the new advances in text mining and,in particular, DL approaches to bring its performance in line with the state of the art.

Focusing on BERT's performance alone in Table 3.5, we can see that it performs better on the binary classification task than the NER task. This is not surprising, as the sequence labelling task is inherently more complex than binary classification. This is because of the number of possible sequences growing exponentially with the length of a document. In particular, the performance gap is bound to widen when training the corresponding models on a relatively small dataset, as is the case in this study. Having less than 300 annotated documents available, we can see from Table 3.5 that BERT's performance on the classification task is in the high 80s across all metrics, whereas its performance on the NER task is in the high 70s overall. This again justifies our choice to run BERT after MetaMap rather than the other way around. Going back to the BERT's classification performance provided in Table 3.5, while examining the misclassified examples, we noticed some patterns. Some simple negation patterns were not captured by the classifier. For example, in the document containing the sentence "Chest X-ray showed no new lesion, no pleural effusion disorder or pneumothorax and history of smoking," both pleural effusion disorder and pneumothorax were misclassified as adverse events. Similarly, in the document with the sentence "admitted with right scaptula/back pain, no chest pain or dyspnea," both chest pain and dyspnea were misclassified as adverse events.

This finding is in line with the current evidence that neural models struggle to generalise negation to out-of-sample datasets, even within the same domain (Grivas et al., 2020). The generalisability of negation remains a challenge, as none of the factors considered, including the annotation guidelines, the amount of data available, and their lexical and syntactic properties, fully explained the poor performance (Wu et al., 2014). Empirical evidence suggests that the use of domain-specific embeddings such as BioBERT (Lee et al., 2020) may improve negation detection (Zavala et al., 2020). BERT can also be fine-tuned to support the negation detection task in clinical

text (Zavala et al., 2020, Lin et al., 2020); however, this requires data to be annotated specifically for this task. Nonetheless, manual adaptation, be it rule modification or in-domain data annotation, remains a recommended strategy for optimising performance in clinical NLP (Wu et al., 2014). Rule-based systems for negation detection such as ConText (Harkema et al., 2009) seem to transfer well within a domain (Sykes et al., 2021). Therefore, the simplest and most effective way of addressing negation as the source of errors in our proposed framework would be to use the ConText algorithm (Harkema et al., 2009) to detect negated contexts and automatically exclude them from further consideration.

Some words, such as the word *decreasing*, can have the opposite effect depending on the context in which it is used. For example, *decreased mobility* implies a negative effect, whereas *decreased pain* implies a positive effect and not an adverse event. The system was not able to differentiate between such contexts. This could be remedied by incorporating domain knowledge about candidate adverse events. Alternatively, with a larger training dataset, these properties could be learned directly from the data.

Finally, the classification model struggled when a given concept was used in multiple contexts. For example, for the concept infection in the document extract "admitted to hospital with lower respiratory tract infection [...] not commenced chemotherapy related infection," the model misinterpreted the latter mention as a negated one and, consequently, misclassified this adverse event.

3.6 Conclusion

This study established the feasibility of automated coding of adverse events described in the narrative section of the SAE reports. This, in turn, enables statistical analysis of adverse events and the patterns of such events so that any correlations with the use of medicines can be estimated in a timely fashion. We demonstrated how an existing DL architecture trained on a relatively small dataset can be used to build similar tools rapidly. Moreover, the evaluation results show that such tools also perform with high accuracy. This performance can be attributed to the choice of the method. BERT is already pre-trained on a large unlabelled corpus, which allows it to be fine-tuned on a small, labelled corpus for a specialised task. As mentioned in the introduction of this thesis, this is particularly relevant for clinical text mining applications, where the data annotation bottleneck has been identified as one of the key obstacles to machine learning approaches for clinical text mining (Spasić et al., 2020).

Unfortunately, the clinical trial data are still mainly handwritten, which means that they cannot be immediately processed in the way proposed in this study. There are

two ways in which this issue can be addressed. We can work with the stakeholders to change the policy on the means of collecting information on SAEs, for example, by transcribing the notes when they reach the safety and pharmacovigilance teams in the central trial unit, by requiring them to be typed, or by using some combination of these two approaches.

Alternatively, we can propose to develop methods to digitise handwritten notes automatically using tools such as Transkribus (Kahle et al., 2017), which have been designed to digitise historical documents and allow the training of specific text recognition models. This would have a great potential for impact on safety by digitising and mining legacy data from previous trials, where some medicinal products may have already reached the market, thus exposing the population to previously overlooked safety concerns. Currently, these issues prevent a systematic analysis of the information provided in the narrative of SAE reports, hence missing an opportunity to identify potential safety signals.

Overall, these findings show that an effective DL strategy can indeed be developed to recognise references to rare clinical events in unstructured text, thereby answering our first research question (**RQ1**). By using SAEs as a case study for rare events, we have managed to combine the implicit semantic knowledge captured by pre-trained LMs and the explicit knowledge of the medical domain encapsulated in the UMLS to compensate for term variations and the lack of data when training a DNN. This is the first step towards supporting our main hypothesis which states that DL can be successfully applied for clinical text mining.

Chapter 4

Word Sense Disambiguation of Abbreviations

Abbreviations are defined in this work as short forms whose full form consists of a single word. As mentioned in Chapter 1, the prevalence of abbreviations in healthcare reports is one of the main characteristics of the clinical sublanguage (Friedman et al., 2002). Indeed, clinical texts are often written under high time pressure and short forms offer an efficient way to write text (Dalianis, 2018). Unfortunately, as abbreviations are often created ad hoc by healthcare professionals, they are not typically defined in formal vocabularies. As a result, they are treated as unknown words by NLP algorithms which leads to a loss of information (Lu et al., 2019). Since we know that clinical texts condense information in a small number of characters and omit anything that is not indispensable, one can assume that every sequence of characters present in these texts is essential (Leaman et al., 2015). Similarly, if some word is used frequently and is too important to be left out, it will likely be shortened to speed up the reporting process. Consequently, unlocking the meaning hidden behind short forms is crucial to text understanding and information extraction. However, many abbreviations are highly ambiguous and cannot be resolved to their full form in a straightforward way. For instance, studies by Liu et al. (2001, 2002) showed that 33% of abbreviations in English clinical text and 81% of abbreviations contained in MEDLINE abstracts can correspond to different words depending on the context. Because the clinical language keeps evolving (Shao et al., 2020), dictionary-based approaches are not sufficient to address this issue that hinders the development of DL models for clinical text mining. Pakhomov et al. (2005) noted, for example, that the number of full forms relating to a single abbreviation has increased in successive versions of the UMLS. This constant evolution calls for the development of an automatic approach that can easily adapt to new short forms. These observations bring us to our second RQ: Can an

effective DL strategy be developed to normalise clinical text by automatically expanding short forms. This question in turn leads us to the following RO: to develop an approach to automatically expand abbreviations to their full form. This approach is detailed in the remainder of this chapter. Here we deal specifically with abbreviations as short forms. Later, in Chapter 6, we will tackle acronyms, which represent a different category of short forms.

The work described here has been published in the paper *A Deep Learning Approach to Self-Expansion of Abbreviations Based on Morphology and Context Distance*, which was presented at the 7th International Conference on Statistical Language and Speech Processing (SLSP) (Chopard & Spasić, 2019).

4.1 Background

In recent years, text data has become ubiquitous in many critical fields. For example, it is nowadays standard practice for medical practitioners to write and rely on electronic reports when taking care of patients. As narratives are an important source of information, this growth has been accompanied by a surge in NLP applications, such as information retrieval and topic modelling. While NLP systems have displayed stellar performance on numerous tasks, they rely most of the time on clean and normalised data due to the tasks' complexity.

However, as actual text data can rarely be found in canonical form, transforming text data into a unique standard representation—also called text normalisation—is a key aspect of NLP pipelines. Some documents might for instance contain uncommon tokens that cannot be directly recognised by a standard NLP system and must first be resolved. The normalisation of short forms (e.g. contractions) is particularly critical in any field that involves the regular and rapid writing of reports, such as aviation or healthcare, where such forms are frequently used to speed up the writing or to ease a repetitive task. The word underlying a short form is hidden and therefore inaccessible to NLP applications, thus skewing their performance (Spasić, 2018). This ambiguity inevitably leads to a loss of information which weakens the system's understanding of language.

Liu et al. (2001) revealed in a study conducted in 2001 that among the 163,666 short forms they retrieved from the UMLS, 33.1% of them referred to multiple full forms. Similarly, Li et al. (2015) reported that the 379,918 short forms which could be found on the website *AcronymFinder.com* had in average 12.5 corresponding full form. Furthermore, they noted that 37 new short forms were added daily to the website. These observations further highlight the need for an automatic method for short form

expansion that is highly adaptable, preferably unsupervised and domain-independent.

Short forms can be divided into two categories: those that refer to a single word (e.g. *PT* for *patient*) and those that refer to multiple words (e.g. *DOB* for *date of birth*). (Note that, although not completely accurate from a linguistic point of view, in the remainder of this work we will refer to the former as *abbreviations* and to the latter as *acronyms* for the sake of simplicity.) The way abbreviations and acronyms relate to full forms is intrinsically different, due to their distinctive nature. Furthermore, while acronyms tend to follow pre-defined rules, new abbreviations are often created spontaneously. Consequently, we believe that these two types of short forms should be considered independently.

4.2 Related work

Whereas Chapter 2 provided a broad review of DL with application to clinical text mining, this section focuses on works related to the specific topic of abbreviation expansion.

NLP applications often rely on external lexical resources to expand the short forms prior to text analysis. However, short forms may correspond to multiple long forms (Li et al., 2015), which implies that word sense disambiguation (WSD) is a required as part of pre-processing. Unlike acronyms, which are often standardised within a domain, authors often create ad hoc abbreviations, which may not be encoded in existing lexicons.

When they are included in specialised biomedical terminologies, it is has been shown that simple techniques, such as bag-of-words, combined with majority sense prevalence were effective in practice despite an expectation that sophisticated techniques based on biomedical terminologies, semantic types, POS and language modelling and machine learning approaches would be necessary (Moon et al., 2015). The ShARe/CLEF eHealth 2013 challenge (Mowery et al., 2016) created a reference standard of clinical short forms normalised to the UMLS (Bodenreider, 2004). The challenge evaluated the accuracy of normalising short forms compared to a majority sense baseline approach, which ranged from 43% to 72%. In line with findings suggested in (Moon et al., 2015), a majority sense baseline approach achieved the second-best performance. Nonetheless, machine learning approaches to clinical abbreviation recognition and disambiguation was found to be as effective with F1-score over 75% (Wu et al., 2016a). However, this study focused on 1,000 most frequent abbreviations in a corpus used for evaluation, which makes it possible to successfully translate their distribution into a classification model. This also means that the given approach may not necessarily work with ad hoc abbreviations. Another problem with using supervised machine learning methods for abbreviation disambiguation in clinical texts is associated with the acquisition of training data. Manually

annotating abbreviations and their senses in a large corpus is time-consuming, labour-intensive and error-prone. In addition, the learnt model may not be transferable across domains, which supervised learning impractical for this particular text-mining problem. With accuracy up to 90%, semi-supervised classification algorithms proved to be a viable alternative for abbreviation disambiguation (Finley et al., 2016). Moreover, an F1-score of 95% could be reached by using an unsupervised approach (Kreuzthaler et al., 2016), which avoids the need to retrain a classification model or use bespoke feature engineering, which makes the approach domain independent. It also proved to be robust with respect to ad hoc abbreviations. Word embeddings provide an alternative way to represent the meaning of clinical abbreviations. Three different methods for deriving word embeddings from a large unlabelled clinical corpus have been evaluated (Xu et al., 2015). Adding word embeddings as additional features to be used by supervised learning methods improved their performance on the clinical abbreviation disambiguation task.

More recently, DL approaches for abbreviation expansion have been developed. For example, Jin et al. (2019b) addressed the problem of biomedical abbreviation expansion by training abbreviation-specific bidirectional LSTM classifiers. They encoded sentence tokens with BioELMo representations. In contrast, Wen et al. (2020) showed that pre-training common NN architectures (e.g. biLSTM, LSTM with attention, and Transformer) on an abbreviation disambiguation task before fine-tuning these models on downstream tasks improved performance on the latter. To that end, they introduced a new biomedical dataset that was automatically generated from PubMED abstracts and in which abbreviations had been artificially introduced through reverse substitutions (Skreta et al., 2020).

All of the above mentioned systems, post-process clinical notes long after clinicians originally created them. The results show that post-processing clinical abbreviation cannot yet guarantee 100% accuracy in their identification and disambiguation. With this problem in mind, a system for real-time clinical abbreviation recognition and disambiguation has been implemented (Wu et al., 2015a). The system interacts with an author during note generation asking them to verify correct abbreviation senses suggested automatically by the system. With the accuracy of 89% and the processing time ranging from 0.630 to 1.649 milliseconds, the system incurred around 5% of total document entry time, which demonstrated the feasibility of integrating a real-time abbreviation recognition and disambiguation module with clinical documentation systems.

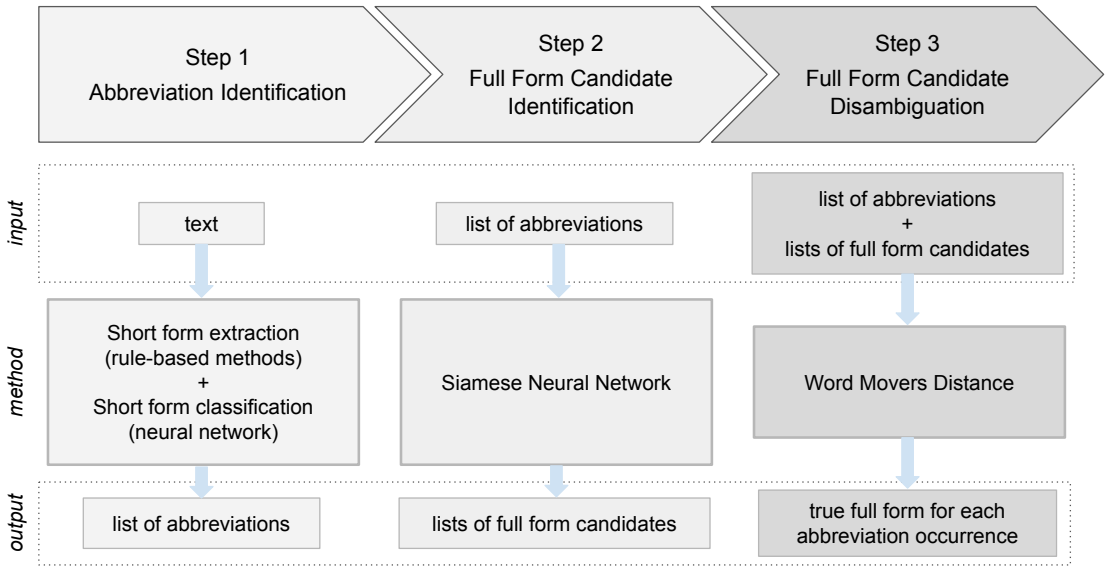


Figure 4.1: Pipeline overview.

4.3 Methodology

Our pipeline consists of three successive steps. The first step identifies short forms contained in a document based on a set of rules and determines the ones referring to single words (abbreviations) rather than multiple words (acronyms) using an NN. Then, the second step extracts, for each abbreviation, a set of full form candidates with a Siamese NN. Finally, the third and last step of our pipeline take advantage the *Word Mover's Distance* (WMD) to disambiguate each abbreviation based on the distance between the full form and the short form's context words in the word embedding space. The whole pipeline is illustrated in Figure 4.1.

4.3.1 Step 1: Abbreviation Identification

Let us assume we have a document D for which we would like to automatically resolve all abbreviations. As mentioned above, we must initially identify all abbreviations in the document: all short forms contained in the text are first extracted, then we discard all acronyms.

To extract all short forms the text is first tokenized, and we gradually discard tokens that cannot be short forms. To begin with, all tokens that are recognised as English words are discarded. For the sake of this work, abbreviations that are identical to existing English words—such as the short form *tab* for the word *tablet*—are assumed to always appear immediately followed by a period so that the two together can be identified as a single token. Indeed, in such case, the use of a full point is standard practice to mark

abbreviations to avoid any confusion.

Secondly, all tokens containing less than 2 or more than 6 characters are rejected. An abbreviation should consist of at least two characters as a single character is extremely ambiguous. Moreover, since abbreviations are short per definition, we set a strict upper limit of 6 characters. This threshold is identical to the one used in previous works on abbreviations (Liu et al., 2001, Xu et al., 2007). This helps the system discard unknown words that are not abbreviations (e.g. misspelled words).

Although they must be discarded eventually, locations and names are rarely part of an English dictionary and are therefore still retained by the system at this point. To deal with this, a NER system is applied to the original document to classify the remaining tokens. Those that are labelled as *PERSON* or *LOCATION* are not retained any further.

After this simple processing, we obtain a list of short forms that includes both abbreviations and acronyms. As mentioned in Chapter 1, in this work we differentiate between abbreviations and acronyms because of their difference in nature and, in this chapter, we develop a method tailored to abbreviations exclusively. In order to model morphological differences between the two, we develop a DNN that learns to distinguish between abbreviations and acronyms. Using a deep architecture rather than other rule-based or machine learning methods has the advantage of obviating the need for any manual features, giving more flexibility to the model to accommodate any type of short forms.

More specifically, the DNN takes as input a sequence of characters which is first processed by fully-connected layers for representation learning. Then, an RNN sequentially reads the improved representation for structure learning. Finally, a soft-max layer predicts whether the sequence is an abbreviation or an acronym (i.e. refers to a single or multiple words). The architecture is displayed in Figure 4.2.

At the end of this first step, the system yields a set A of abbreviations, namely

$$A = \{\mathrm{w} \in D \mid \mathrm{w} \text{ is an abbreviation}\}$$

.

4.3.2 Step 2: Full Form Candidates Identification

In the second step, a set of full form candidates Φ_α must be identified for each abbreviation $\alpha \in A$. Based on the assumption that characters appear in the exact same order in both forms, a simple rule-based solution could consist in searching for all words within the document that have as a subset all the characters of contained in the abbreviation. However, many short forms such as *xmas* and *x-mas* for the word *Christmas*

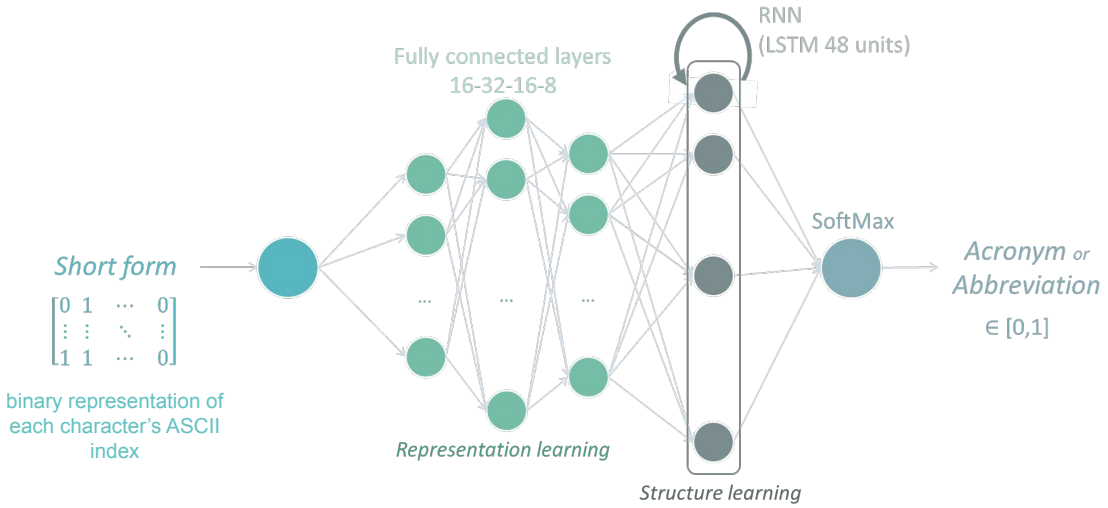


Figure 4.2: NN architecture to differentiate between abbreviations and acronyms.

contain characters that are not present in their full form. Terada *et al.* chose to address this manually by not considering specific characters such as *X* (Terada et al., 2004). Unfortunately, this limits the system to follow manually engineered rules. Therefore, for maximal flexibility, we instead take advantage of a DL architecture to select full form candidates.

We design a Siamese RNN (Mueller & Thyagarajan, 2016) to learn how full forms relate to short forms. Its architecture is depicted in Figure 4.3. Every abbreviation $\alpha \in A$ and every word w in the document D that is recognised as an English word are first encoded as a sequence of characters. Then, one by one, each abbreviation is fed along with one word to the network. The two are first processed by multiple fully-connected layers for representation learning. Each improved representation is then fed to one of two independent RNNs: one that processes the sequence corresponding to the short form and one that processes the full form. Finally, the output of the two RNNs are compared by the network which must decide whether this word could potentially be referred to by the short form or not. If so, the word is added to the list of full form candidates Φ for that abbreviation.

Instead of feeding only corpus words, we could feed every English word contained in the dictionary along with each abbreviation to get a set of full-form candidates that is more comprehensive.

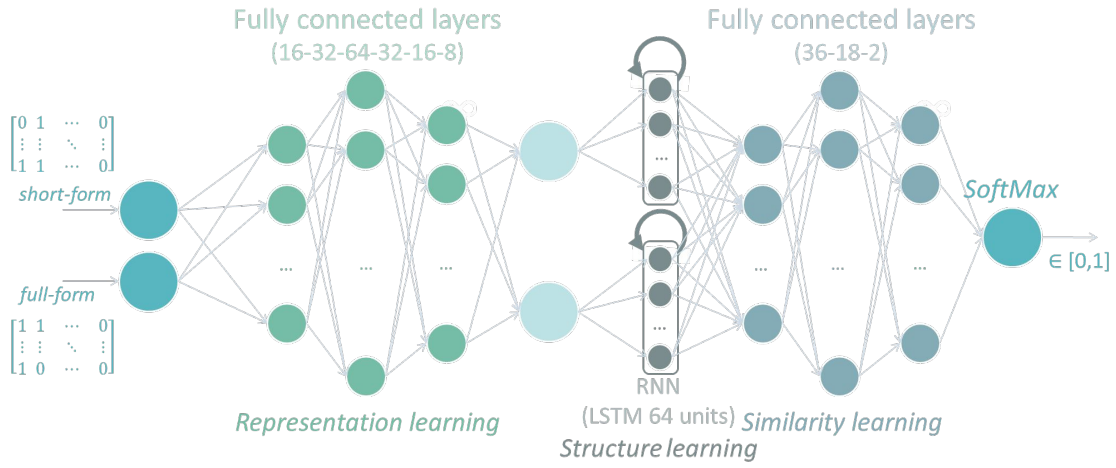


Figure 4.3: Siamese RNN to select a set of full form candidates.

4.3.3 Step 3: Abbreviation Disambiguation

In order to determine the right full form for each abbreviation $\alpha \in A$ the system must select the best of all full form candidates Φ_α. We rely on the assumption that short forms and their corresponding true full forms share a similar context in order to disambiguate each abbreviation and find the most appropriate of all full form candidates.

To compare the context of an abbreviation and its full form candidates, we propose to use the WMD, a measure that was developed by Kusner et al. (2015) to assess similarity between two documents. It takes advantage of the semantic properties inherent to word embeddings to match documents that have a similar meaning, although they consist of very different words. First, each word i is represented by a d-dimensional word embeddings $\mathbf{x}_i$. The use of pre-trained word vectors allows the model to take advantage of the linear properties of continuous space word representations (Mikolov et al., 2013b) without the need to train it on the chosen corpus. For each document D, the n-dimensional normalised BOW vector is denoted as $\mathbf{f}^D$ with entries $f_i^D = c_i / \sum_{j=1}^n c_j$, where c_i is the number of occurrences of word i in document D, and n the vocabulary size. Finally, let $\mathbf{T} \in \mathbb{R}^{n \times n}$ be the transport matrix whose entries $\mathbf{T}_{ij}$ denote how much of f_i^D should travel to $f_j^{D'}$, where D' is a another document. Then the WMD minimises the following linear optimisation problem:

$$\min_{\mathbf{T} \geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad \text{subject to} \quad \sum_{i=1}^{n} \mathbf{T}_{ij} = f_j^{D'}, \ \sum_{j=1}^{n} \mathbf{T}_{ij} = f_i^D \ \ \forall i, j \tag{4.1}$$

which is an instance of the well-studied earth mover's distance problem for which many efficient solutions already exist (Monge, 1781, Rubner et al., 1998, Wolsey & Nemhauser,

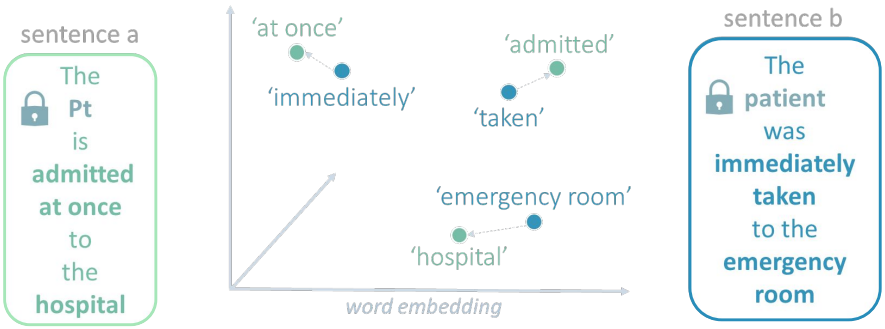


Figure 4.4: WMD for abbreviation disambiguation. The minimum cumulative distance between non-stop words in the context of target abbreviation *Pt* and of full form candidate *patient* is computed.

2014, Ling & Okada, 2007, Pele & Werman, 2009).

Figure 4.4 illustrates how this measure is used to determine the full forms that are semantically close to target short form. In this example, the abbreviation *Pt* used in the sentence *"The Pt was immediately taken to the emergency room"* refers to the word *patient*, which can be found in the sentence *"The patient is admitted at once to the hospital."*. Although the two sentences do not share a single representative word, they are semantically very close. To minimise the cumulative distance between the two sets of context words $c_1 = \{\text{admitted, at once, hospital}\}$ and $c_2 = \{\text{immediately, taken, emergency room}\}$, the words *admitted* and *taken* are matched together, and so are the words *at once* and *immediately*, and the words *hospital* and *emergency room*, because their respective word embeddings lie close together in the word space. Since the context words for the short form *Pt* have a similar meaning as the context words for the candidate *patient*, the WMD between the two sentences will be small.

For each of the abbreviations α identified in the first step, we have filtered—in the second step—a set Φ_α of potential full form candidates. Let S(w) be the set of all sentences in document D containing word w, i.e. $\mathsf{S(w)} = \{s \in D | w \in s\}$. We must determine the best full form candidate ϕ^*_α that abbreviation α refers to in sentence s_α. We compute for each candidate $\phi \in \Phi_\alpha$ the WMD between the sentence the abbreviation appears in (i.e. s_α) and each sentence containing this candidate (i.e. $\mathsf{S}(\phi)$). The individual WMD are then summed up and averaged to yield a disambiguation score $\sigma(\phi)$:

$$\sigma(\phi) = \sum_{\tilde{s} \in \mathsf{S}(\phi)} \frac{\mathsf{WMD}(\mathsf{s}_\alpha, \tilde{s})}{|\mathsf{S}(\phi)|}. \tag{4.2}$$

Eventually, we consider the full form with smallest WMD as being the best one, namely

$\phi^*_\alpha = \min_{\phi \in \Phi_\alpha}(\sigma(\phi))$.

4.4 Results and Discussion

4.4.1 Step 1: Abbreviation Identification

Our pipeline separates the abbreviation identification step into two subsequent parts: short forms identification and abbreviation-acronym differentiation.

We first evaluate the former on discharge summaries from the 2009 i2b2 medication challenge (Uzuner et al., 2010) that we have manually annotated to this end. We use the python library *NLTK* (Bird et al., 2009) both for word tokenization and to find non-standard English words. More specifically, we build an English dictionary that combines all words found in the Brown, Reuters, and Words corpora. To label entities, we use the Stanford NER (Finkel et al., 2005). We succesfully identify short forms with an average F1-score of 62.20% with high precision (92.18%) but low recall (52.90%). The low recall can be explained by the presence of numerous short forms that are common enough to be part of an English vocabulary (e.g. *MD*, *Dr.*, *mg*, *cm*, *tel*, *ID*) and, as a result, are discarded by our method. Such short forms would likely be discarded if using a different dictionary, or could easily be resolved using a standard abbreviation dictionary. When ignoring them, recall improves drastically and our approach achieves an average F1-score of **91.04%** (Precision: 95.20%, Recall: 87.38%).

Second, the network that we developed for abbreviation-acronym differentiation is trained and evaluated using distinct samples from the 32,048 unique short forms of the CARD framework (Wu et al., 2016a). The network takes as input a short form x_i and outputs a probability score $y_i \in [0, 1]$, where 0 denotes that the short form refers to multiple words (acronyms) and 1 denotes that the short form refers to a single word (abbreviation). Each short form is represented as a 6×8 matrix where each row corresponds to the binary representation of one character based on its Unicode code point—6 being the highest number of characters in a sequence. Standard characters have code point value at most 256, which allows a compact representation in only 8 dimensions. As a comparison, a one-hot encoding would require around 100 dimensions, depending on the actual number of characters in the corpus. Since short forms can sometimes refer to either an abbreviation or an acronym depending on the context, we assign each of them a label between 0 and 1 which accounts for this versatility. More precisely, the label is computed as a weighted sum of the nature of all possible expansions:

$$y_i = \frac{\sum_{Abb(x_i)}}{\sum_{Abb(x_i)} + \sum_{Acr(x_i)}}, \tag{4.3}$$

where $Abb(x_i)$ and $Acr(x_i)$ are the sets of full forms of x_i that are single word and multiple words respectively. The deep architecture consists of 4 fully connected layers (with 16, 32, 16 and 8 notes respectively) for representation learning followed by an RNN consisting of an LSTM cell with 48 units for morphology learning. Finally, a softmax layer outputs the probability score. The network is optimised using the Adam optimiser (Kingma & Ba, 2015) to minimise the cross-entropy loss. The network is trained on 80% of the data, while the test set, which is used to determine the final out-of-sample performance, is composed of 20% of the short forms. Due to the high imbalance of the dataset (only 19.15% of them are abbreviations), we oversample samples from this smaller class so that there are roughly the same number of examples in both classes. Since the goal is to retrieve all abbreviations, recall is the most important measure for this task. On test set we achieve a recall of **72%** (F1-score of 59%, Precision 49%). These scores could be improved by training the network on a larger dataset. However, we can easily set a minimum threshold for disambiguation, which will eventually reject short forms that do not have any full form with context close enough therefore discarding any remaining acronyms.

4.4.2 Step 2: Full Form Candidates Identification

To evaluate the second step of our pipeline, we use the 31,922 abbreviations from the CARD dataset. We assign the label $y_i = 1$ to each pair $x_i = (shortform, fullform)$ contained in the dataset. Negative examples are created by randomly selecting a short form from the dataset and pairing it with another randomly selected full form. If the pair is not already in the dataset, we assign it the label $y_i = 0$ and add it to the set of samples. We repeat this process until we have as many negative as positive samples. We train the Siamese RNN on 70% of the data whereas 30% are left out for testing. The deep architecture consists of 6 fully-connected layers (with 16, 32, 64, 32, 16, and 8 nodes respectively) for representation learning followed by two independent RNN—one for the abbreviation, one for the candidate. Each RNN consists of an LSTM cell with 64 units. The output of each network are then compared by stacking them and feeding them to 3 successive fully-connected layers consisting of 36, 18, and 2 nodes respectively. Prediction is achieved through a final softmax layer. We use dropout to prevent overfitting.

For comparison we implement the simple rule-based baseline suggested by Terada et al. (2004). It is based on the assumptions that the short form always contain less characters than the full form and the characters contained in the short form is a subset in the same order of the characters contained in a potential full form candidate (except for "X", "-", "/"). The result of the comparison is displayed in Table 4.1. Our approach

Table 4.1: Comparison between our DL-based full form selection approach against a rule-based baseline [%].

Method	Prec	Rec	F1
Baseline (Terada et al., 2004)	**90.12**	64.53	75.21
Ours (Siamese RNN)	75.21	**84.04**	**78.57**

achieves a slightly higher F1-score than the baseline, but with much higher recall. Once again, recall is the most important measure as it is crucial to select the true full form as part of the candidates, whereas FPs will be naturally dealt with in the final step.

The network handles around 5'000 samples per second for inference, which means that it needs 8 seconds for a corpus of 40'000 words and less than $40s$ for the entire Oxford English dictionary (i.e. 171'476 words).

4.4.3 Step 3: Abbreviation Disambiguation

Due to the lack of well-established benchmarks for abbreviation disambiguation, we evaluate our approach on a subset of the MSH WSD dataset, which mostly contains acronyms. We believe that although acronyms and abbreviations have different morphological properties, disambiguation is similar. Hence we here reproduce an experiment first conducted by Prokofyev et al. (2013) and then later replicated by Li et al. (2015) and Ciosici et al. (2019) respectively. The subset of MSH WSD—a dataset of abstracts from the biomedical domain created by Jimeno-Yepes et al. (2011)—selected by Li *et al.* consists of 11'272 abstracts which contain a total of 69 ambiguous short forms each having in average 2.1 full form candidates.

To assess the performance of our approach, we compare it with the same methods as Ciosici et al. (2019). First, a simple baseline called *FREQUENCY* which, as its name implies, simply selects the most frequent full-form candidate. Clearly, such an approach completely disregards any context information and relies purely on corpus statistics for determining the best candidate. Second, the SBE model, a word embedding-based model developed by Li et al. (2015), which first computes word embeddings of abbreviations by summing the word embeddings of the words in a window around the abbreviation before disambiguating between full form candidates using the cosine similarity. Similarly, *Distr. Sim.* is an approach introduced by Charbonnier and Wartena (Charbonnier & Wartena, 2018), which relies on word embeddings to build weighted average vectors of the context that are compared using the cosine similarity. Finally, the last benchmark we compare with is the Unsupervised Abbreviation Disambiguation (UAD) method developed by Ciosici et al. (2019), which deals with disambiguation as a word

Table 4.2: Comparison between our disambiguation approach against different benchmarks (Surrounding Based Embedding (SBE) (Li et al., 2015), Distr. Sim. (Charbonnier & Wartena, 2018), UAD (Ciosici et al., 2019)) on a subset of the MSH WSD dataset [%].

		Weighted			Macro		
Method	Acc	Prec	Rec	F1	Prec	Rec	F1
FREQUENCY	54.14	30.04	54.14	38.46	25.55	46.34	32.79
SBE	82.48	83.07	82.48	82.53	82.18	82.16	81.87
Distr. Sim.	80.19	80.87	80.19	80.25	79.90	80.12	79.71
UAD	90.62	92.28	90.62	90.66	91.35	91.36	90.59
ours (WMD)	**96.36**	**96.38**	**96.36**	**96.36**	**95.65**	**96.97**	**96.27**

prediction task. For more details on all benchmarks' implementation, please refer to the work of Ciosici et al. (2019).

We use 300-dimensional word embeddings pre-trained on a subset of the Google News corpus, which contains about 100 billion words. Our scores are computed as an average of a 3-fold CV, similar to the implementation of the benchmarks. Table 4.2 illustrates the remarkable performance of the WMD compared to other benchmarks for short-form disambiguation. The WMD improves the score on every metric by a margin of at least 4.1%. It is worth noticing our approach, although it exhibits great performance, comes with an important drawback. It needs to rely on a corpus that contains sentences that include the long-forms. In practice, this might not always be easy to acquire, especially when the long forms are domain-specific.

4.4.4 Discussion

As our method consists of three independent steps, its performance is always bounded by the previous step. Here, the step with the lowest performance is the second one, which involves identifying full-form candidates. Since we want to ensure we extract the true candidate, the important measure in this step is recall, which is 84.04%. In order to boost recall further, one could lower the threshold for selecting candidates. A drawback is that the more candidates we need to process in the third step, the slower it is.

In general, because our method is not end-to-end, each step can be designed in a specific way to address the difficulty of this step. However, it is known that developing end-to-end solutions that limit human input is often powerful and one of the reasons behind the remarkable success of deep learning. Thus, it would be interesting to investigate such approaches for comparison.

4.5 Conclusion

In this study, we introduced a domain-independent approach to matching abbreviations to their full form as part of text normalisation. Unlike the vast majority of existing approaches to abbreviation expansion, which use an external lexicon in order to interpret an abbreviation and match it to its full form, we extract full forms from the corpus itself. This approach is based on the assumption that a full form is actually used elsewhere in the corpus. The likelihood of such an event increases with the size of a corpus, which makes the approach most suitable for large-scale text mining applications. However, since our Siamese RNN is able to select full-form candidates from the whole English dictionary in a short amount of time, one could use publicly available resources (such as Wikipedia) to find context for the candidates that are not part of the corpus. Thus, our approach is still very relevant for low-data regimes as often found in clinical settings.

An advantage of using a corpus instead of a lexicon, which is typically domain-specific, is that it makes our approach domain-independent. It also avoids the need for maintaining an external lexicon while making our approach robust with respect to ad hoc abbreviations. However, one may argue that our approach still uses an external lexicon. Indeed, we do use an external lexicon to train an RNN to model the morphological differences between acronyms and abbreviations. Assuming that these morphological properties are universal across the language rather than specific to a domain, then once trained on any representative lexicon, the model itself is readily reusable across domains and does not require to be re-trained. The same can be said about the second RNN, which models the morphological principles of word abbreviation. Once trained on any abbreviation lexicon, the model can be used to expand abbreviations that were not present in the training data (i.e. the lexicon). Lastly, the DL approach taken avoids the need for manual feature engineering, while the novel use of existing lexicons avoids the need for manual annotation of training data.

Finally, the use of DL to constrain the search space of possible matches based on the morphological structure of both abbreviations and full forms paves the way for more sophisticated approaches that can be utilised to analyse their contexts. We used the WMD to leverage pre-trained word embeddings to measure semantic compatibility between abbreviations and full forms based on the assumption that both are used in similar contexts.

We have evaluated the different steps of our approach and achieved F1-scores of 91.04%, 78.57% and 96.36%, respectively. These results are in line with those reported by other state-of-the-art methods.

In this chapter, we have demonstrated the feasibility of developing an approach to automatically expand abbreviations to their full form (**RO2/a**). This provides a partial answer to our second research question (**RQ2**). Indeed, as touched upon earlier, abbreviations are not the only kind of short forms prevalent in clinical text data. Clinical reports also contain acronyms, short forms that refer to multi-word phrases. Therefore, acronyms require the extraction of sequences of words, which adds another layer of complexity to the problem of finding the corresponding full form. For this reason, we handled the expansion of the two kinds of short forms as separate problems. In the next chapter, we will turn our attention to the problem of automatic acronym expansion in our quest to address the second research question (**RQ2**): Can an effective DL strategy be developed to normalise clinical text by automatically expanding short forms?.

Chapter 5

Word Sense Disambiguation of Global Acronyms

Acronyms represent another category of short forms prevalent in the clinical sublanguage. Like abbreviations, they make the reporting task more efficient (Dalianis, 2018). The main difference between abbreviations and acronyms is the number of words in their long-forms. Whereas the former refer to a single word, the latter correspond to more than one word. The actual number of words in the acronym long forms may vary, which exacerbates the problem of finding long forms. Not all acronyms are initialisms, in which each character corresponds to the initial letter of a word, e.g. *High Blood Pressure (HBP)*. Several characters from an acronym may be mapped to a single word, often indicating the initial letter of a morpheme, e.g. *AtheroSclerotic CardioVascular Disease (ASCVD)*. In addition, some words such as determiners and prepositions are often ignored when constructing an acronym, e.g. *Activities of Daily Living (ADL)*, but not always, e.g. *Shortness Of Breath (SOB)*. As a result, it is impossible to determine the length of an acronym's long form based on the number of characters in the acronym itself. In contrast, abbreviations—which we tackled in the previous chapter—are contractions of single words that are often created ad hoc by removing some letters from words.

This crucial difference between abbreviations and acronyms motivates the development of a different approach for expanding the latter. Nonetheless, such an approach still addresses the same RQ about the feasibility of a DL strategy for normalising short forms in clinical texts. The problem of acronym expansion in clinical texts is more challenging than the same problem in other types of narratives, e.g. scientific documents. Indeed, scientific writing conventions prescribe that acronyms should be explicitly defined when used for the first time in a text by writing the full form followed by the acronym—written in uppercase—within parentheses (Doumont et al., 2010). By con-

trast, global acronyms, such as the ones used in clinical narratives, appear in documents without their definitions. Global acronyms are widely accepted as preferred synonyms of predominant domain-specific concepts, e.g. *Deoxyribonucleic Acid (DNA)* (Yu et al., 2002). As such, they are described in relevant domain dictionaries (Stedman, 2005, Bodenreider, 2004). However, shorter acronyms tend to be ambiguous (Andersson et al., 2017, Moon et al., 2014) and, therefore, they may have multiple entries in such dictionaries. For example, Diabetes Mellitus, Dystrophia Myotonica, Doctor of Medicine and Drosophila Melanogaster all share the same acronym, DM. Automatic recognition of global acronyms usually entails mapping to a correct entry in an external dictionary, comparable to a WSD problem Agirre & Stevenson (2007). The novelty of our approach to this task is an attempt to forego an external dictionary altogether in order to make the approach more portable. The automatic expansion of global acronyms comes with two main challenges: the ambiguity of acronyms and a lack of annotated data for training disambiguation models. As before, this lack of training data is associated with privacy safeguards and the annotation bottleneck. To address the problem of suitable training data, we suggest a method of simulating the usage of clinical acronyms using biomedical abstracts. We use this method to create a large annotated dataset, which we then take advantage of to train a DL model for the disambiguation of global acronyms.

Some aspects of the work presented in this chapter have been published in the journal Bioinformatics under the following titles (Filimonov et al., 2022): *Simulation and annotation of global acronyms*. This work was a joint effort of the co-authors with contributions from Dr Maxim Filimonov who implemented the web-based acronym simulation platform and Daphné Chopard who solely developed the acronym disambiguation module. In the given publication, the module was used as a case study to practically demonstrate the utility of the acronym simulation system for the purpose of training supervised machine learning approaches to acronym disambiguation. Hence, the article's focus was on the performance of acronym disambiguation. In this chapter we focus on the methodological aspects of the proposed acronym disambiguation approach.

5.1 Background

Acronyms are systematic abbreviations of frequently mentioned words and phrases. Their formation follows special capitalisation and blending patterns (Fandrych, 2008). They are introduced primarily to support the efficiency of written communication in terms of time and space. From the reading perspective, familiarity (prior experience processing a given stimulus), rather than orthographic regularity, plays a critical role in rapidly translating an acronym from percept to meaning (Laszlo & Federmeier, 2007).

Therefore, the use of globally accepted acronyms should pose no major difficulties in specialist communication between domain experts. Indeed, clinical narratives feature extensive use of acronyms, which, unlike their counterparts in formal scientific writing, are not defined explicitly in documents that refer to them. However, when the content of such documents needs to be analysed automatically, their use can hinder the performance of NLP algorithms (Moon et al., 2014, Spasić et al., 2019). For example, when retrieving information from EHRs, the use of acronyms (e.g. 'HIV') obscures the corresponding phrase (e.g. 'human immunodeficiency virus') whose words (e.g. 'virus') cannot be indexed by a search engine and, hence, cannot be retrieved. On the other hand, the highly polysemous nature of acronyms (e.g. 'MRS' can be interpreted as 'magnetic resonance spectroscopy', 'Melkersson-Rosenthal syndrome' or a courtesy title prefixed to the name of a married woman) may result in retrieval of irrelevant documents. These problems can be resolved by automatically mapping acronyms to correct interpretation in an external dictionary (Bodenreider, 2004) based on their context of use. This may be viewed as a WSD problem (Agirre & Stevenson, 2007), which is commonly approached by supervised machine learning. Supervised methods are trained using a set of manually annotated examples. Among other factors, the performance of machine learning models and the significance of test results both depend on the size of the dataset used for training and testing, respectively. A recent systematic review of clinical text data in machine learning revealed the data annotation bottleneck as one of the key obstacles to machine learning approaches to clinical NLP (Spasić et al., 2020). The need to preserve patient privacy further narrows down this bottleneck by removing crowdsourcing as a viable option for annotation. For synthetic data, crowdsourcing remains an option, but it becomes an expensive commodity due to the medical expertise required.

5.2 Methods

We present, in this chapter, a method for the automatic expansion of acronyms in clinical narratives. A high-level overview of the approach is provided in Figure 5.1.

To eliminate the manual data annotation bottleneck, we look at the possibility of generating an annotated dataset automatically. More specifically, we suggest a novel application of existing NLP methods on scientific abstracts to simulate the clinical narrative style of acronym usage and annotate them automatically with the correct senses. This, in turn, enables the creation of large datasets that can be used to train supervised approaches to WSD of biomedical acronyms.

We use this method to generate a large training dataset to train a DL model for

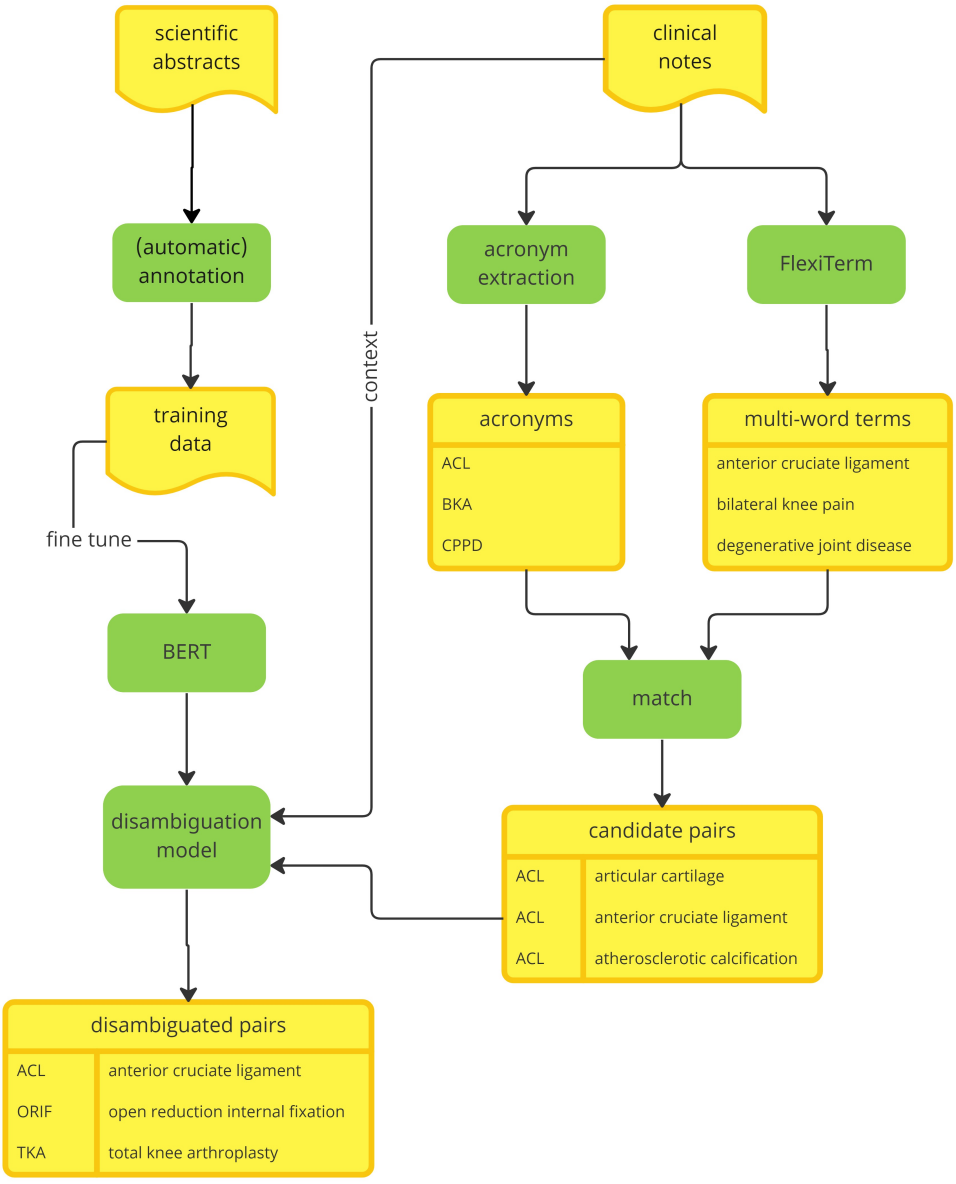


Figure 5.1: Flowchart of our approach to acronym disambiguation.

acronym disambiguation. More precisely, we fine-tune a pre-trained BERT model to disambiguate an acronym occurrence based on its context. The sole function of the model is the binary classification of a candidate pair linking an acronym and a candidate long form. Its output is positive if they are compatible based on the context of the short form and negative otherwise.

To fully automate the expansion of acronyms, we need a method to extract such pairs from the text. We retrieve potential acronyms using a simple heuristic based on their orthographic properties. Independently, we extract multi-word terms from text based on their linguistic and statistical properties using an existing method called FlexiTerm (Spasić et al., 2013).

Having retrieved both acronyms and multi-word terms, they are matched using their internal properties. In a nutshell, the characters from an acronym are aligned against each term to select a list of potential long forms. All candidate pairs of acronyms and their potential long forms are then disambiguated using the previously trained DL model based on the context of the sentence in which the acronym is mentioned. The final result is a lexicon of acronyms mapped to their senses. Speaking of lexicons, we would like to emphasise the fact that the method itself does not rely on any external lexicons. In that aspect, our method departs from the traditional approaches, thus providing a novel solution to the problem of expansion of clinical acronyms.

The following sections provide further details about each stage of the proposed algorithm:

1. Simulation and annotation of global acronyms
2. Disambiguation of global acronyms
3. Expansion of global acronyms

5.3 Simulation and Annotation of Global Acronyms

The prevalence of acronyms in biomedical domains (Liu et al., 2002) gave rise to the proliferation of methods that extract acronym definitions from texts. Most of these methods focus on biomedical literature and have been evaluated on abstracts (Ao & Takagi, 2005, Chang et al., 2002, Gaudan et al., 2005, Liu & Friedman, 2002, Okazaki & Ananiadou, 2006, Pustejovsky et al., 2001, Schwartz & Hearst, 2002, Sohn et al., 2008, Wren & Garner, 2002, Yu et al., 2003, 2007). They rely on scientific writing conventions, which prescribe that all acronyms need to be defined the first time they are mentioned in a document by specifying the full form followed by the acronym, written within parentheses in uppercase. These conventions are modelled by pattern-matching

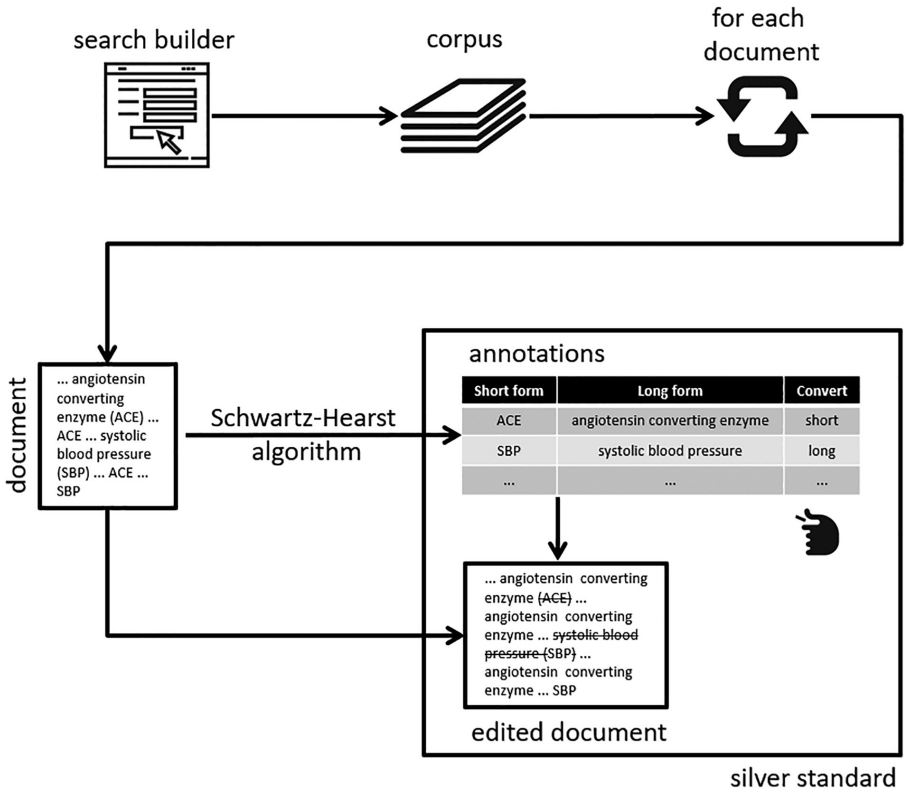


Figure 5.2: System design for dataset creation.

rules to identify potential acronym definitions followed by heuristic alignment of the acronym against its full form. The Schwartz-Hearst algorithm for identifying acronyms by Schwartz & Hearst (2002), which performs at 96% precision and 82% recall, is by far the most referenced method of its kind. It can be embedded easily into NLP algorithms to support more complex tasks, e.g. multi-word term recognition (Spasić, 2018, 2021). Similarly, it forms the backbone of the system described here (Figure 5.2).

As a first step to the simulation of global acronyms, we apply the Schwartz-Hearst algorithm to recognise acronym definitions in scientific abstracts. As a result, an acronym is linked to its full form, i.e. sense. To artificially simulate the clinical narrative style of acronym usage, we need to remove these definitions. More precisely, we use the definitions to re-write the corresponding document using either an acronym or its full form consistently throughout the document. As a result, an acronym will never occur alongside its long form within the same document. Therefore, physical proximity can no longer be used as a feature to link acronyms to their long form. This characteristic, in turn, creates an opportunity to learn how to link global acronyms to their long forms based solely on their morphology and the context in which they are used.

The training dataset for this task is created as follows. Given a document, the acronym simulation system first chooses between an acronym and its full form randomly

and then replaces all occurrences of the chosen item with its counterpart. Whenever the full form is removed from the text, it is retained as the sense annotation and can, therefore, be used to train machine learning approaches to WSD. By repeating this process on many documents, a large training dataset can be created automatically.

The above-described system has been implemented as a web-based application. The main source of data is MEDLINE, a bibliographic database of scientific articles covering the fields relevant to clinical applications, including medicine, nursing, pharmacy, dentistry and healthcare (MEDLINE, 2021). The choice of this particular database ensures that there will be a large lexical overlap between the training data and clinical narratives.

MEDLINE is publicly available online and provides free access to the abstracts of the articles it indexes. These abstracts are used as free-text documents to assemble a corpus (Figure 5.2) whose topical coverage is constrained by the search query. This constraint fosters the "one sense per discourse" hypothesis (Gale et al., 1992). For example, in a corpus retrieved using the search terms 'knee' and 'MRI', it is reasonable to assume that any mention of the acronym 'ACL' would have a single meaning, which would be that of 'anterior cruciate ligament'. This assumption is in line with expectations from a thematic corpus of clinical narratives. For example, a corpus of knee MRI reports would be expected to uphold a "single sense per discourse" hypothesis.

The search is performed using PubMed (2021), a search engine designed specifically to retrieve information from MEDLINE. A search query, which follows PubMed's syntax to combine field names, MeSH terms, keywords and Boolean operators, is obtained from a user using the front-end of our web-based application. It is then passed onto PubMed using its API to retrieve the corresponding documents. The API limits the number of results returned to 500 per query, so the application may take some time to complete the search in multiple batches. To prevent unnecessary processing and to save time, users have the option to limit the number of results themselves depending on their needs.

Once retrieved, all documents are processed as described in Figure 5.2. In addition, all long forms are used to search the UMLS (2021) to obtain their unique concept identifier. This helps unify different long forms of the same acronym. For example, if 'DM2' is linked to 'diabetes mellitus type 2', 'diabetes mellitus type II' and 'type two diabetes mellitus', then the corresponding sense will be the same as all three long forms have the same concept identifier, C0011860. The UMLS is searched using its own API, which has a rate limit of 20 requests per second per IP address. Consequently, the web application may take some time to process all the long forms.

All information is managed in a MongoDB database. Once the corpus has been processed, it can be downloaded together with the sense inventory in a simple JSON

format, ready to be used to train an acronym disambiguation system.

The code for the simulation of global acronyms is made publicly available[1].

Although MEDLINE offers a convenient source of data for simulating global acronyms, it comes with the drawback of consisting of scientific writings. Therefore, we will investigate in Section 5.5 how well the method developed in the next section (5.4) works on clinical notes since, as underlined in Chapter 2, both data sources have very different properties.

5.4 Disambiguation of Global Acronyms

To collect abstracts relevant to clinical applications, we create a PubMed query using the keyword 'clinical' and the suffix 'logy' to refer to various clinical domains (e.g. gastroenterology) while excluding certain keywords (e.g. biology). All abstracts are retrieved and annotated automatically by the system. All non-ambiguous acronyms, i.e. those mapped to a single long form uniquely identified in the UMLS, are discarded, leaving a total of 963 ambiguous acronyms with a mean of 4.09 potential long forms per acronym (minimum 2, maximum 27, median 3).

Using the above training data, we develop a system for global acronym disambiguation. The model is described in this section. We formulate the problem of acronym disambiguation as that of binary classification: given an acronym in its context and a potential expansion candidate, the model determines whether or not the candidate corresponds to the true full form. In practice, acronyms can be found in a text using an NER model trained to that effect or simple rules as suggested in Chapter 4. Below we discuss an approach for finding multi-term expansion candidates.

As we choose to tackle a binary classification task, we must compile both positive samples—i.e. where the expansion candidate is the true expansion—and negative samples—i.e. when it is not—to obtain a suitable training dataset. Using the simulated data detailed above, we proceed as follows. First, we extract every sentence containing an acronym from the set of all retrieved abstracts. This first step is straightforward as acronyms are indicated by a tag in the simulated data. We save each sentence along with the acronym and its true long form. Then, we loop through these sentences to extract a dictionary that contains for each unique acronym a list of all true expansion candidates. Similar to acronyms, true expansion candidates are easily extracted from the simulated data as they are saved along with the corresponding short forms. Afterwards, for each sentence saved, we create two samples: a positive example which corresponds to the original triplet *(sentence, acronym, true long form)* and a negative

[1] https://github.com/ispasic/acronimity

example, where the true long form is replaced at random by one of the other expansion candidates according to the dictionary created in the previous step. By proceeding in this manner, we obtain a balanced dataset which consists of as many positive examples as negative ones.

As mentioned in Chapter 2, in order to get a trustworthy estimate of the performance of a system it is important to set aside a fraction of the data—a test set—prior to doing any kind of analysis or training. Similarly, it is often beneficial to hold back another portion of the data—a validation set—to get an unbiased evaluation of a model fit throughout its development, including estimating a good set of hyperparameters. Consequently, for each of the 963 ambiguous acronyms, a total of 1000 samples (or 10% if less than 10'000 samples are available) are drawn out randomly and reserved for validation. We repeat this process a second time to obtain a test set. A total of 16'130'782 remaining samples are retained for training. Separating the dataset with respect to each acronym helps ensure that each acronym in the test dataset is seen during training prior to inference.

Next, the disambiguation of an acronym against the potential long forms is performed using a transformer-based architecture similar to that of Huang et al. (2019c). In that work, the classifier version of BERT (Devlin et al., 2019)—which can be used to model the relationship of a pair of text sequences—is fine-tuned for WSD. More specifically, the model is fed two sequences: (i) a word within its context and (ii) that word followed by one of its definitions. The network is trained to determine whether or not the definition in the second sequence fits in the context of the first sentence. Because short form disambiguation is a special case of WSD, we draw on this idea for developing our classifier. Similar to Huang et al. (2019c) we fine-tune BERT but, in our case, we are modelling a relationship between (i) a sentence that contains an acronym and (ii) the acronym itself together with its potential long form. The network is illustrated in Figure 5.3. It uses the $\text{BERT}_{\text{BASE}}$ uncased model for classification and the implementation from HuggingFace. The network was pre-trained on BookCorpus and English Wikipedia. Like the original paper, the network is fine-tuned with the Adam optimiser with a linear warmup. A gradient clipping of 1.0 is used to prevent the gradients from exploding.

It is worth noting that a similar strategy was developed by Pan et al. (2021) as the winning solution for the second SDU@AAAI-21 shared task on acronym disambiguation in English scientific papers (Veyseh et al., 2021). In their approach, the authors suggested using the long form as a first sequence and the acronym within its context as the second sequence. In order to single out the acronym and directing the model's attention to it, they wrapped the short form using two special tokens (<start> and

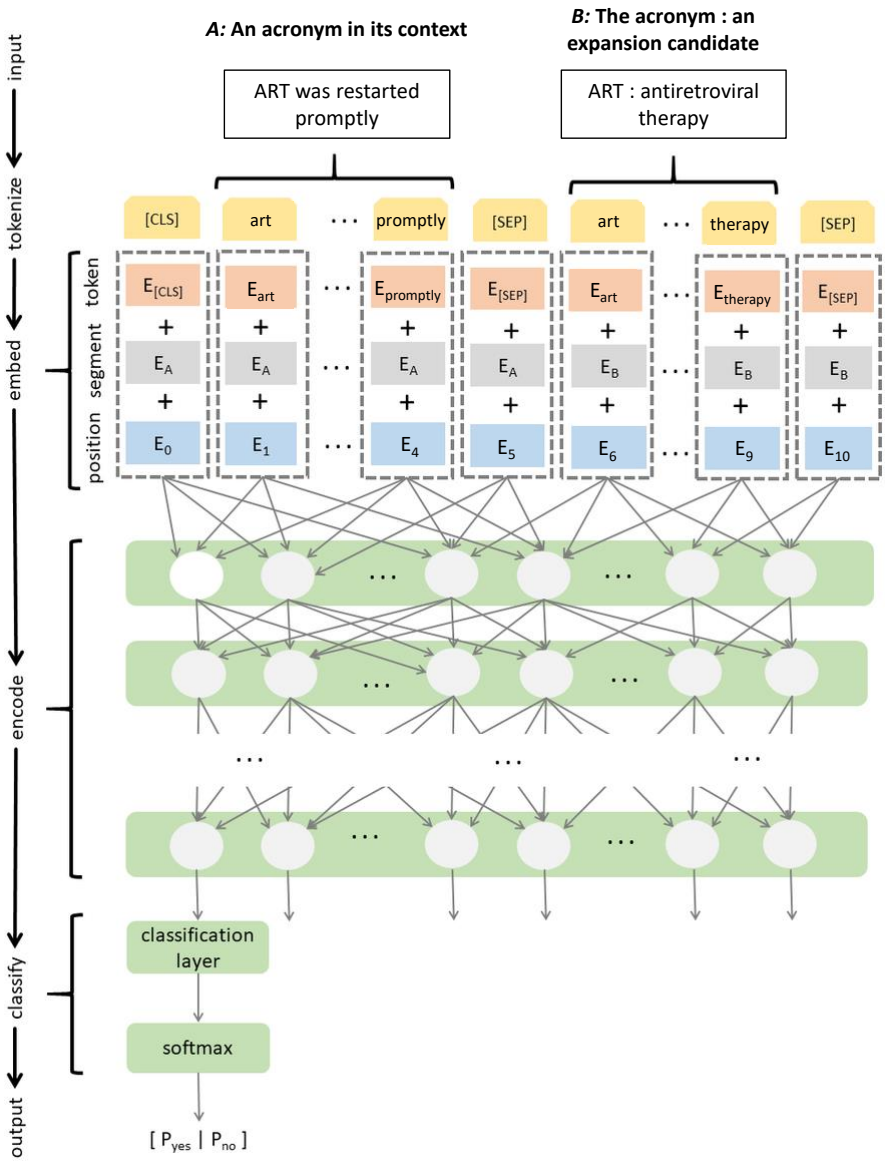


Figure 5.3: Diagram of BERT-based architecture for acronym disambiguation.

Table 5.1: Comparison between our transformer-based classifier against a naïve frequency baseline [%].

Method	Acc	Prec	Rec	F1
Frequency baseline	64.96	69.15	54.02	60.65
Ours (BERT classifier)	**94.62**	**94.45**	**94.89**	**94.67**

<end>). In order to find the best candidate at inference time, one can select the expansion candidate that was assigned the highest probability.

We train the transformer-based classifier to recognise the correct long form and evaluate it on the test dataset. Results can be found in Table 5.1. The accuracy, precision, recall, and F1 score achieved are 94.62%, 94.45%, 94.89% and 94.67%, respectively. To get a better sense of these results, we compare this performance to that of a naïve baseline classifier. This benchmark—which is based on the most frequently occurring long form—achieves accuracy, precision, recall, and F1 score of 64.96%, 69.15%, 54.02% and 60.65%, respectively. These results show that the performance achieved by our classifier is not an artefact of the acronym distribution in the dataset. On the contrary, our model outperforms the frequency approach by at least 0.253, which shows that it successfully identified patterns for acronym disambiguation.

The proposed method is easy and efficient. In addition, it does not rely on external resources for learning to find the best expansion candidate. Its main drawback comes from its supervised training. While nothing prevents the model from being fed at inference short forms and long forms that were not seen during training, one can expect that the model will not be able to correctly classify them if these were not present in the training set. Indeed, the model does not take into account the context of long forms. In theory, as we have introduced, at the beginning of this chapter, a method for simulating clinical acronyms from biomedical abstracts, one could use as many of these data to train a model to make sure that every short form and long form has been seen prior. In practice, however, it might be computationally too expensive to train a model with such a large number of samples. The potential lack of generalisation to new short form and long form pairs will be examined more thoroughly in Section 5.5 below. In comparison, the disambiguation method based on WMD that was introduced in Chapter 4 is fully unsupervised and relies on the context of long forms. Yet, because of that, it often requires external resources in order to get the necessary sentences that contain long forms.

5.5 Expansion of Global Acronyms

The purpose of this section is twofold. First, it aims to demonstrate the effectiveness of the acronym disambiguation module trained on a corpus of biomedical abstracts using a corpus of clinical narratives. Second, it extends the functionality of acronym disambiguation to include the extraction of potential acronyms and the corresponding long-form candidates automatically from a given corpus. To assemble a corpus of clinical narratives, we use MIMIC-III. This large, freely available database comprises de-identified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 (Goldberger et al., 2000). The choice of a subset of these data was motivated by the availability of local expertise needed to interpret the results. Specifically, we had easy access to the team who developed the TRAK ontology (Button et al., 2013), which defines standard care for the rehabilitation of knee conditions, and who previously applied this ontology to support text mining of knee radiology reports (Spasić et al., 2015). We identified 2'609 knee radiology reports in MIMIC-III. For each report, we extracted its main body. Section titles were pinpointed using a set of simple regular expressions and removed from further consideration. The main reason for discarding section headings was the subsequent recognition of potential acronyms. Namely, both are written mainly using uppercase characters. To recognise potential acronyms, we used a simple heuristic, which extracts tokens that satisfy the following conditions:

1. The length of the token is at least 2 and at most 10 characters.
2. The token has to be tagged as a noun.
3. The first character of the token has to be a capital letter.
4. The number of letters has to be greater than the number of digits.
5. The number of uppercase letters has to be greater than the number of lowercase letters.
6. The token has to occur at least 10 times in the corpus.

Using this heuristic, we extracted a total of 26 potential acronyms, all of which were coincidentally uppercased. The full list of these acronyms, together with their interpretation, is provided in Table 5.2. LF is a special token used in MIMIC-III to indicate missing information that had been removed to anonymise the data. Nonetheless, it is retained as a potential acronym to really challenge the disambiguation model.

Acronym	True Long Form
ACL	Anterior Cruciate Ligament
AM	Ante Meridiem
AML	Angiomyolipoma
AP	Anteroposterior
BKA	Below Knee Amputation
CPPD	Calcium Pyrophosphate Deposition
CT	Computed Tomography
DJD	Degenerative Joint Disease
IV	Intravenous
LCL	Lateral Collateral Ligament
LF	N/A
MCL	Medial Collateral Ligament
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MVA	Motor Vehicle Accident
MVC	Motor Vehicle Collision
OA	Osteoarthritis
OR	Operating Room
ORIF	Open Reduction Internal Fixation
PCL	Posterior Cruciate Ligament
PD	Posterior Duplication
PM	Post Meridiem
SI	Sacroiliac
STIR	Short Tau Inversion Recovery
TKA	Total Knee Arthroplasty
TKR	Total Knee Replacement

Table 5.2: List of acronyms and their correct full form.

As we mentioned before, in this project, we define acronyms as short forms of multi-word terms. As seen in Table 5.2, three of the acronyms extracted in the previous step are short forms for a single word, e.g. osteoarthritis (OA). Again, this puts the disambiguation module to the test, as we want to establish whether the acronym would be incorrectly matched to another multi-word term. To find potential long forms for the given acronyms, we applied a method called FlexiTerm, which was originally developed to recognise multi-word terms (Spasić et al., 2013). Even though it was subsequently extended to recognise acronyms as multi-word terms (Spasić, 2018), we did not use this option in order to test acronym disambiguation with our model. The list of multi-word terms was obtained using the latest implementation of FlexiTerm (Spasić, 2021). It consisted of 2,079 multi-word terms, each linked to multiple variants (e.g. plural, hyphenation, etc.). Potential acronyms were matched to multi-word terms using a simple heuristic:

1. Both the acronym and the multi-word term have to start with the same letter.
2. All characters from the acronym have to occur in the same order within the multi-word term.
3. The acronym cannot occur as a token within the long form.
4. The last character from the acronym has to occur in the last token of the multi-word term.
5. The Damerau–Levenshtein distance between the acronym and the initialism constructed out of the multi-word term is less than 2.

The matching procedure was deliberately loose so as to extract as many potential long forms in order to test the disambiguation model. As a result, each acronym was mapped to 4.19 multi-word terms with a standard deviation of 2.86 (see Figure 5.4 for distribution).

To be able to disambiguate an acronym, we need its context of use. Therefore, for each acronym, we extracted a list of all sentences that mention that acronym. Each candidate long form of a given acronym was then classified against each of its sentences. In practical terms, a sentence represents the first sequence that is fed into the BERT-based disambiguation module as described in the previous section. The second sequence consists of the acronym followed by a colon and the long-form candidate. To evaluate the disambiguation module on clinical data, each pair was labelled as either true or false depending on whether the long form was correct or not in accordance with the gold standard provided in Table 5.2. The ground truth was created by manually

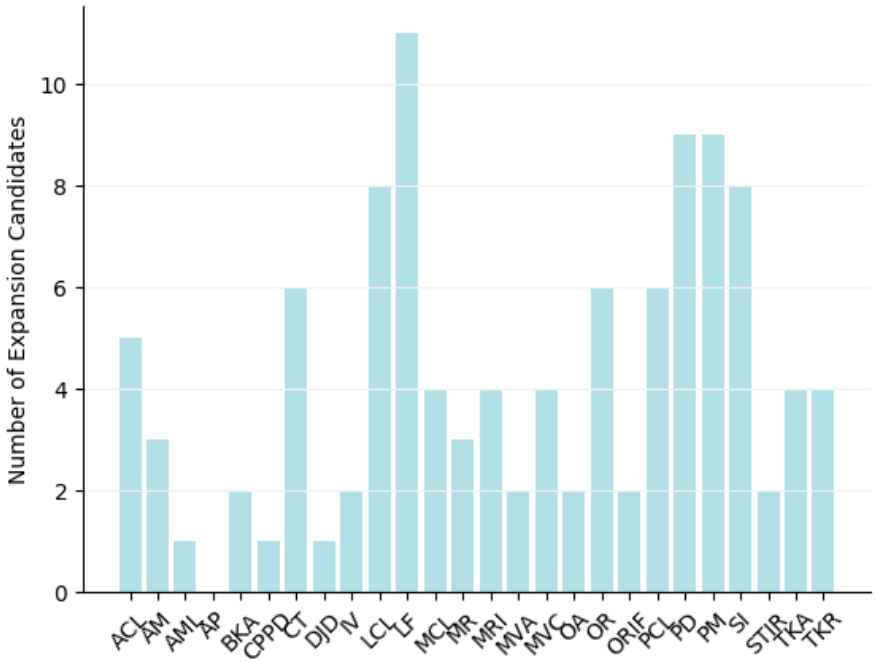


Figure 5.4: Histogram of number of expansion candidates per acronym.

labelling multi-word terms suggested as long forms by comparing their meaning against the interpretation given in Table 5.2. The corpus-specific ground truth is provided in Table 5.3. Note that some acronyms have no corresponding long forms in the ground truth. For example, SI is not considered an acronym as we assume that an acronym always corresponds to a multi-word term. Even when an acronym corresponds to a multi-word term, it may not occur in the corpus, in which case it will have no long form in the ground truth. This is one of the drawbacks of using Flexiterm for identifying multi-word candidate terms, the full-form candidate has to be present in the corpus, which might not always be the case. To alleviate this issue, one could rely on external resources, preferably from the same domain, to extract candidates.

Next, the unlabeled data were passed through the BERT-based disambiguation model to classify each pair. In effect, this classified each acronym occurrence separately. The corresponding results are provided in the first row of Table 5.7.

For each acronym, the classifications of its occurrences were aggregated to classify its long forms. A long form was accepted as a positive one only if it was classified as such in at least 75% of the acronym occurrences. Otherwise, the long form counts as a negative prediction. Thus, for each pair containing an acronym and its potential long form, we obtain a single prediction. The corresponding results are provided in the

Acronym	Expansion Candidates	Ground-Truth	Prediction
ACL	amorphous calcification anterior cruciate ligament appreciable change articular cartilage atherosclerotic calcification	anterior cruciate ligament	**anterior cruciate ligament**
AM	ankle mortise anterior midline assessment with mri	None	None
AML	anterior midline	None	None
AP	None	None	None
BKA	bilateral total knee arthroplasties bilateral knee pain	None	bilateral total knee arthroplasties bilateral knee pain
CPPD	calcium pyrophosphate deposition disease	calcium pyrophosphate deposition disease	**calcium pyrophosphate deposition disease**
CT	cartilage thinning complete tear complex tear cortical thickening cross table lateral cross table	None	None
DJD	degenerative joint disease	degenerative joint disease	**degenerative joint disease**
IV	intraoperative fluoroscopic views intraoperative views	None	intraoperative fluoroscopic views intraoperative views

Table 5.3: Predictions (part 1/4)

Acronym	Expansion Candidates	Ground-Truth	Prediction
LCL	lateral collateral ligament lateral collateral ligament complex lateral condyle linear calcification lucent fracture line lucent lesion lucent line lytic lesion	lateral collateral ligament lateral collateral ligament complex	**lateral collateral ligament** lucent fracture line **lateral collateral ligament complex** lateral condyle lucent lesion
LF	lateral facet lateral film lateral patellar facet left ankle fracture left distal femur left femur left fibula left foot left knee followup left proximal fibula lytic focus	None	left ankle fracture left foot
MCL	medial collateral medial collateral ligament medial condyle mild chondrocalcinosis	medial collateral medial collateral ligament	**medial collateral** **medial collateral ligament** medial condyle
MR	mild periosteal reaction medial retinaculum median ridge	None	None
MRI	medial retinaculum mild irregularity minimal irregularity multiplanar multisequence images	None	**multiplanar multisequence images**

Table 5.4: Predictions (part 2/4)

Acronym	Expansion Candidates	Ground-Truth	Prediction
MVA	mild varus angulation motor vehicle accident	motor vehicle accident	mild varus angulation **motor vehicle accident**
MVC	mild degenerative change minimal degenerative change motor vehicle accident motor vehicle collision	motor vehicle accident motor vehicle collision	**motor vehicle accident** **motor vehicle collision**
OA	obliquity of ap osseous abnormality	None	osseous abnormality
OR	oblique radiographs open reduction operating room operative report osseous remodeling osteophytic reactions	operating room	**operating room**
ORIF	open reduction internal fixation operative report for full	open reduction internal fixation	**open reduction internal fixation**
PCL	patellar cartilage periprosthetic loosening periprosthetic lucency posterior cruciate ligament proximal calf proximal medial collateral ligament	posterior cruciate ligament	**posterior cruciate ligament** proximal medial collateral ligament patellar cartilage
PD	posttraumatic deformities patchy demineralization proton density proximal femoral diaphysis proximal fibular diaphysis proximal diaphysis proximal tibial diaphysis posterior displacement posterior doctor	None	posterior displacement

Table 5.5: Predictions (part 3/4)

Acronym	Expansion Candidates	Ground-Truth	Prediction
PM	packing material patient motion popliteus muscle prior mri proximal fibular metaphysis proximal metadiaphysis proximal metaphysis proximal tibial metaphysis proximal tibial metadiaphysis	None	None
SI	spot fluoroscopic images sagittal stir images single image stir images skeletal injury soft tissue injury signal intensity soft tissue irregularity	None	soft tissue injury
STIR	soft tissue injury soft tissue irregularity	None	None
TKA	total knee arthroplasty total knee replacement total left knee arthroplasty total right knee arthroplasty	total knee arthroplasty total knee replacement total left knee arthroplasty total right knee arthroplasty	**total knee arthroplasty** **total left knee arthroplasty** **total right knee arthroplasty** **total knee replacement**
TKR	total knee arthroplasty total knee prosthesis total knee replacement total right knee replacement	total knee arthroplasty total knee prosthesis total knee replacement total right knee replacement	**total knee arthroplasty** **total knee prosthesis** **total knee replacement** **total right knee replacement**

Table 5.6: Predictions (part 4/4)

Table 5.7: Performance of our acronym disambiguation model on clinical notes [%].

Evaluation method	Acc	Prec	Rec	F1
Acronym occurrence	75.80	47.52	93.63	63.05
Acronym	84.40	55.26	100	71.19

second row of Table 5.7. What stands out the most is the high recall of 100%. This high value indicates that the model does not yield any FN. In other words, all true long forms were correctly classified by the model. However, as shown by the precision of 55.25%, the model misclassified some long forms that do not fit the context of the acronym.

Detailed predictions can be found in Tables 5.3, 5.4, 5.5, and 5.6. The first column contains the acronyms identified in the clinical notes, whereas the second column shows the expansion candidates extracted from the notes using FlexiTerm. The third column indicates which of these expansion candidates should be classified as True by the model. Finally, the fourth and last column contains the candidates predicted as True by our classifier in at least 75% of the acronym occurrences. True positives are indicated in bold.

Among the 26 acronyms found in the clinical notes, 9 were not part of the dataset that we created to train the BERT model from PUBMed abstracts, namely AM, AML, CPPD, DJD, LF, MVC, PM, SI, STIR. Despite this, the classifier correctly handled 6 of them (i.e. two thirds) by not matching them to any incorrectly suggested long forms. This indicates that the model is able to exploit the words in the long form to disambiguate an acronym based on its context. The three acronyms AM, AML and PM were all correctly assigned none of the suggested long forms. In addition, the true candidate was correctly classified as positive for the two acronyms CPPD and DJD. Finally, the system correctly identified the two true expansions of MVC. The model only failed when handling the acronyms LF, SI, STIR. In all three cases, the model classified some long forms as positive even though none of the expansion candidates fitted the context of the acronym. In future work, this can be addressed by taking into account the context of the long for in addition to that of an acronym.

5.6 Conclusion

We described a web-based application that uses a corpus of scientific abstracts to simulate the clinical narrative style of acronym usage and annotate them automatically with

the correct senses, which in turn can be used to train supervised approaches to WSD of biomedical acronyms. It helps navigate the problems associated with patient privacy and manual annotation overhead associated with the use of clinical text data in machine learning (Spasić et al., 2020). Even though the application can be used to create large training datasets automatically, it is relatively slow due to limitations associated with the use of external APIs. However, this is not a major issue as the acquisition of training data is seen as batch processing rather than real-time processing.

Overall, this study shows that, even though acronym datasets are limited, it is possible to artificially create such datasets. The simulated data can eventually be used to successfully train powerful acronym expansion DL models. In addition, this work demonstrates that the implicit knowledge captured by pre-trained LMs can be leveraged for acronym expansion. In particular, we show that this approach can be applied to the disambiguation of acronyms in clinical notes, even though the global acronyms were simulated with medical scientific writings. More specifically, we approach the issue of data scarcity in two ways: on the one hand, we use a large available corpus to create a dataset that mirrors the behaviour of acronyms in clinical text. On the other hand, we transfer knowledge from pre-trained LMs to help the classifier understand the context of acronyms to find the best expansion candidate. These results conclude the second part of our second research objective (**RO2/b**) and address our second research question (**RQ2**) by providing an effective DL strategy to normalise clinical text by automatically expanding acronyms.

Chapter 6

Data Augmentation

One main problem of clinical datasets is scarcity. Reasons for a lack of labelled training data include the annotation bottleneck—labeling data is time-consuming, expensive and challenging—as well as privacy concerns that prevent data from sharing. In Chapter 2, we explained how when provided with too few training examples, DL models can often fit the training data too well, meaning that they even manage to learn the underlying noise patterns resulting in models that do no generalise well on unseen data. This effect is called *overfitting*. In contrast, relying on a large number of training samples can help DL models learn invariances in the data. For example, an image of a car always represents a car whether it is facing left or right; this can only be learned by an NN, if it is supplied with a sufficient number of examples of cars facing both right and left.

Therefore, a natural way to help a model learn specific invariances when the number of samples is limited is to explicitly provide training data that reflect those invariances. To return to the previous example, if we know that the essence of a car is invariant to the way it is facing, we can flip all images in our training set before adding them to the training data (with the same label as the original images) to ensure that the DNN is provided with examples of cars facing both left and right irrespective of the way they were facing initially. This process is referred to as *data augmentation*.

Even though very popular in the domain of computer vision, data augmentation struggles to gain popularity in the field of NLP. This can be explained in part by the difficulty of finding easy label-preserving transformations for text data. Indeed, recall that language consists of symbols and that any change in these symbols (e.g., inverting letters within a word) or among these symbols (e.g., inverting words within a sentence) can lead to symbols—or collections of symbols—that do not bear any meaning. In contrast, images consist of pixels and one can easily change all the pixels—e.g., by applying a greyscale filter—without modifying what the image represents.

In addition, most of the efforts that have been put into developing data augmentation techniques for NLP do not take into account recent advances and the advent of transformers. Consequently, it is still unclear whether the language understanding captured by pre-trained LM can still benefit from learning language invariances when fine-tuned on a task with limited training data.

In this chapter we thus try to answer our third and last RQ: Can a label-preserving transformation of existing text data improve the performance of DL approaches to text mining that are based on pre-trained LMs?. As mentioned in Chapter 1 this RQ lead us to two ROs. The first objective is to develop a set of label-preserving transformation for text data. The second objective is to develop an approach that automatically chooses the best schedule of transformations for a given task and dataset—both of which, we will tackle here.

The outcome of this RQ has been published under the title *Learning Data Augmentation Schedules for Natural Language Processing* at the Workshop on Insights from Negative Results in NLP at the Conference on Empirical Methods in Natural Language Processing (EMNLP) (Chopard et al., 2021b).

6.1 Background

In recent years, data augmentation has become an integral part of many successful DL systems, especially in the fields of computer vision and speech processing (Krizhevsky et al., 2012, Jaitly & Hinton, 2013, Hannun et al., 2014, Ko et al., 2015). Traditionally, data augmentation approaches take the form of label-preserving transforms that can be applied to the training datasets to expand their size and diversity. The idea of generating synthetic samples that share the same underlying distribution as the original data is often considered a pragmatic solution to the shortage of annotated data, and has been shown to reduce overfitting and improve generalisation performance (Shorten & Khoshgoftaar, 2019). However, despite a sound theoretical foundation (Dao et al., 2019), this paradigm has not yet translated to consistent and substantial improvement in NLP (Longpre et al., 2020).

The inability of NLP models to consistently benefit from data augmentations can be partially attributed to the general difficulty of finding a good combination of transforms and determining their respective set of optimal hyperparameters (Ratner et al., 2017), a problem that is exacerbated in the context of text data. Indeed, since the complexity of language makes text highly sensitive to any transformations, data augmentations are often tailored to a specific task or dataset and are only shown to be successful in specific settings.

In this study, we investigate whether automatically searching for an optimal augmentation *schedule* from a wide range of transformations can alleviate some of the shortcomings encountered when applying data augmentations to NLP.

This endeavour follows the recent success of automated augmentation strategies in computer vision (Cubuk et al., 2019, Ho et al., 2019, Cubuk et al., 2020). In doing so, we extend the efforts to understand the limits of data augmentation in NLP.

6.2 Related Work

This section presents a review of related work in data augmentation for NLP and in automated data augmentation in general. In contrast, Chapter 2 offered a more general related work on DL in the context of clinical text data.

Although there exist recent surveys (Feng et al., 2021, Shorten et al., 2021) that offer a comprehensive review of related work, they do not provide a comparative analysis of the different data augmentation approaches and of their effect on learning performance.

In general, the literature lacks general comparative studies that encompass the variety of tasks and datasets in NLP. Indeed, most of the existing text data augmentation studies either focus on a single approach in a specific setting or compare a small set of techniques on a specific task and dataset (Giridhara. et al., 2019, Marivate & Sefara, 2020). In addition, many of these comparisons have been conducted before the widespread adoption of contextualised representations. Recently, Longpre et al. (2020) showed that, despite careful calibration, data augmentation yielded little to no improvement when applied to pre-trained transformers even in low data regimes. While their comparative analysis is conducted on various classification tasks, it focuses on a limited set of augmentation strategies that are applied independently and whose hyperparameters are optimised via a random search.

To assess the effectiveness of data augmentation on pre-trained transformers—our third RQ, we investigate in this chapter the paradigm of learning data augmentation strategies from data in the context NLP. The idea is to leverage the training data to automatically discover an optimal combination of augmentations and their hyperparameters at each epoch of the fine-tuning. First, we define a search space that consists of a variety of transformations before relying on the training data to learn an optimal schedule of probabilities and magnitudes for each augmentation. This schedule is later used to boost performance during fine-tuning.

In this section, we review a variety of data augmentations and, for each category of transformations, we highlight a subset of augmentations that is representative of the

category and that will constitute our search space. We focus on *transformative* methods which apply a label-preserving transformation to existing data—rather than generative methods which create entirely new instances using generative models. Indeed their simplicity of use and their low computational cost make them good candidates for a wide deployment In the last decade, we witnessed a widespread adoption of continuous vector representations of words, which can be easily fed to DNN architectures. As a result, label-preserving transforms have been developed not only at the lexical level (i.e. words) but also at the latent semantic level (i.e. embeddings). This distinction is emphasised throughout this section.

6.2.1 Word Replacement-based Augmentation

A commonly used form of data augmentation in NLP is word replacement. At the lexical level, the most common approach consists in randomly replacing words with their synonyms (Zhang et al., 2015, Mueller & Thyagarajan, 2016, Vosoughi et al., 2016). Some variants include replacement based on other lexical relationships such as hypernymy (Navigli & Velardi, 2003) or simply words from the vocabulary which can be sampled either uniformly at random (Wang et al., 2018b) or from a distribution that takes into account word similarity (Cheng et al., 2018). Another popular approach consists in using an LM for replacement:while Fadaee et al. (2017) take advantage of an LM to replace common words by rarer words, Ratner et al. (2017) select words to swap based on multiple pre-defined transformations functions that depend on the relative location of entities of interest and POS tags. Similarly, Kolomiyets et al. (2011) leverage the Latent Words Language Model (LWLM) (Deschacht et al., 2012) to replace headwords—which is the assumed position of temporal trigger words. Because these transformations do not ensure the preservation of the sample class, Kobayashi (2018) suggested conditioning a bidirectional LM on the labels, an idea later revisited by Wu et al. (2019) who replaced the LM with a conditional BERT.

At the latent semantic level, word replacement amounts to randomly replacing its embedding with some other vector. For instance, Wang & Yang (2015) choose the k-nearest-neighbour—as measured with cosine similarity—in the embedding vocabulary as a replacement for each word. A similar strategy was later adopted by Vijayaraghavan et al. (2016) who selected words to replace and their replacement (ordered by cosine similarity) with geometric distributions. The same metric is also used by Zhang et al. (2019) to replace words from a class with the most analogue word in another class based on the linguistic regularities observed in word embeddings (Mikolov et al., 2013b).

In this study, as replacement methods, we select both **synonym** and **hypernym** replacement as well as **contextual augmentation** at the word level and **nearest neigh-**

bour at the embedding level.

6.2.2 Noising-based Augmentation

A simple yet effective form of augmentation that is often applied to images and audio samples is data noising. Not surprisingly, this type of data augmentation can also be found in NLP despite the discrete nature of text data.

In its simplest form, data noising when applied to text consists in inserting, swapping or deleting words at random (Wei & Zou, 2019). More generally, the process of ignoring a fraction of the input words is often referred to as *word dropout* (Iyyer et al., 2015) and can take multiple forms.For example, each input token can be retained with a given probability while the rest are entirely discarded (Iyyer et al., 2015, Dai & Le, 2015). Similarly, words may be completely dropped from the vocabulary, as suggested by Zhang et al. (2016) who set random dimensions of the input vector—which contains word frequencies—to zero. Alternative implementations of word dropout include replacing, with a given probability, the word embeddings of the input tokens with the embedding of the generic "unknown word" token "UNK" (Bowman et al., 2016) or with the underscore token "_" (also called *blank noising*) (Xie et al., 2017).

Sometimes, word replacement can also be thought of as a form of noising. For instance, replacing words at random with other words from the vocabulary introduces noise into the data.Variants include choosing replacements based on word frequencies (Xie et al., 2017) or on the cosine distance between the respective word embeddings (Cheng et al., 2018).

In contrast, at the distributed representation level, this type of augmentation often takes the form of added noise to the embeddings. Possible noising schemes include Gaussian noise (Kumar et al., 2016, Cheng et al., 2018), uniform noise (Kim et al., 2019), Bernoulli noise and adversarial noise (Zhang & Yang, 2018). Typically, noising is applied to every word embedding, but it can also be applied only to selected ones (Kim et al., 2019). For instance, in the context of spoken language understanding, Kim et al. (2019) choose to add noise to slot values only in order to account for their diversity ("open-vocabulary"). Alternatively, as with word dropout, noise can be incorporated into the training by discarding, across all words, some embedding dimensions with a predefined probability (Dai & Le, 2015).

Noising strategies used in this study are based on **random deletion**, **random swap** and **random insertion** at the sentence level as well as **Gaussian noise** and **uniform noise** at the feature level.

6.2.3 Back-translation

Back-translation provides a way for neural machine translation (NMT) systems to leverage monolingual data to increase the amount of parallel data (Sennrich et al., 2016, Edunov et al., 2018). For instance, Fadaee & Monz (2018) sample monolingual data containing difficult-to-predict words in the target language (German) and back-translate these to the source language (English) before adding the resulting pairs to the training data to boost the performance of a NMT system. This approach has also been leveraged for paraphrasing. For instance, Wieting et al. (2017) translate the foreign language side of parallel data to English and use the produced translation as a paraphrase for the English side of the bitext; the pairs are then use to learn paraphrastic sentence embeddings.

In the same vein, back-translation can be applied twice in a row (e.g. from English to another language and back to English) to generate new data points without the need for parallel corpora and can therefore find applications as a task-agnostic augmentation in other tasks such as text classification (Luque & Pérez, 2018, Aroyehun & Gelbukh, 2018), paraphrase generation (Mallinson et al., 2017) and QA (Yu et al., 2018a). For instance, Mallinson et al. (2017) propose a paraphrasing model that takes a source English sentence, generates k translations in an intermediate language, which are then combined and back-translated into English to produce a single paraphrase. Similarly, in the context of QA, Yu et al. (2018b) produce k foreign language translations of the document containing the answer, which are in turn translated back into English so as to obtain k^2 document paraphrases from which the answers must then be extracted.

Here, we consider a wide range of intermediate languages to include **back-translation** into our search space

6.2.4 Automated Data Augmentation

In NLP, little effort has been put into developing strategies that can, given a task and a dataset, learn an optimal subset of data augmentation operations and their hyperparameters (Shorten et al., 2021). Yet, this idea has been very successful in computer vision and has led multiple approaches to achieve state-of-the-art results on various datasets (Cubuk et al., 2019, Ho et al., 2019, Cubuk et al., 2020). For instance, in the context of image classification, Cubuk et al. (2019) have proposed AutoAugment a procedure that automatically searches for optimal augmentation policies using reinforcement learning. Later, population-based augmentation (PBA) (Ho et al., 2019)—an algorithm that views the data augmentation selection and calibration problem as a hyperparameter search and can thus leverage the population-based training (PBT) method (Jaderberg

et al., 2017) to find an optimal transformation schedule in an efficient way—was introduced as a more cost-effective yet competitive alternative to AutoAugment. Their approach was shown to match the performance of the former and achieved state-of-the art results on some datasets while requiring far fewer computational resources. Finally, Cubuk et al. (2020) introduced the RandAugment method which tackles some of the issues arising from these previous works.

In this study, we adapt the PBA framework to NLP in an attempt to learn an optimal schedule of data augmentation operations with optimised hyperparameters.

6.3 Methodology

In this section, we discuss the mechanism behind the PBA algorithm in more details and define our hyperparameter space. We also explain how we conduct the data augmentation search and the overall training procedure.

6.3.1 Population-Based Augmentation

Given a hyperparameter search space that consists of data augmentation operations along with their probability level (i.e. likelihood of being applied) and their magnitude level (i.e. strength with which they are applied), PBA works as follows: during a predefined number of epochs, k child models of identical architecture are trained in parallel for the task at hand on a given dataset. Periodically, the training is momentarily interrupted, all models are evaluated on a validation set and an "exploit-and-explore" procedure takes place. First, the worst-performing models (bottom 25%) copy the weights and hyperparameters of the best-performing models (top 25%) (**exploit**), then the hyperparameters are either slightly perturbed or uniformly resampled from all possible values (**explore**). At that point, training can continue as before until the next exploit-and-explore procedure. At the end of the training, a data augmentation policy schedule is extracted from the hyperparameters of the best performing child model. The obtained schedule can then be used to train from scratch a different model on the same task and the same dataset. The PBT algorithm which underlines this process is illustrated in Figure 6.1.

6.3.2 Hyperparameter Space

The hyperparameter space consists of 10 data augmentation operations highlighted in the previous section with associated probability and magnitude. For most replacement methods, the magnitude level can be thought of as a percentage of tokens on which

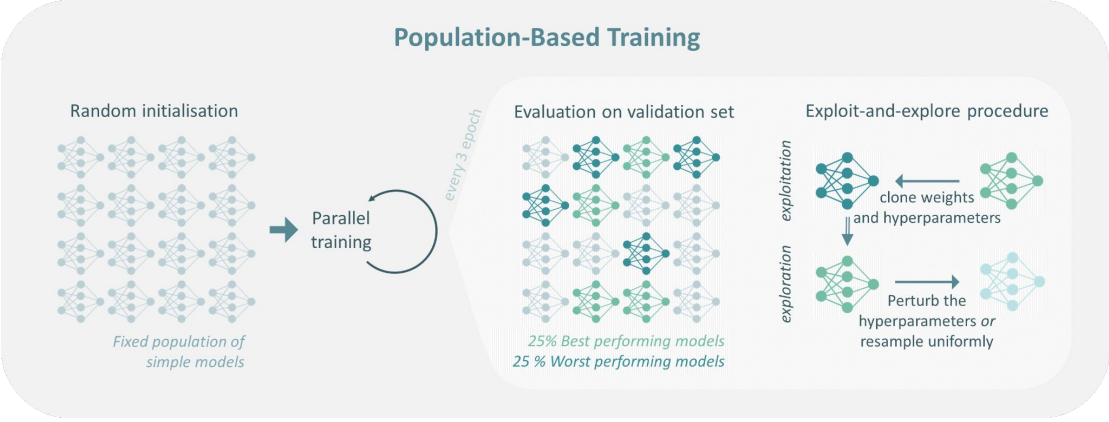


Figure 6.1: The basic principles of the PBT algorithm.

the transformation is applied and, for noising transforms, it corresponds to the amount of noising. In the context of back-translation, however, magnitude relates to the quality of the translation according to BLEU3 scores (Aiken, 2019). The remainder of this sections contains details concerning the implementation of these augmentations as well as how exactly magnitude is defined for each one of them.

As a reminder we consider the following 11 data augmentation operations:

- Synonym replacement
- Hypernym replacement
- Nearest neighbour
- Contextual augmentation
- Random insertion of a vocabulary word
- Random insertion of synonym
- Random deletion
- Random swap
- Gaussian noising
- Uniform noising
- Back-translation

The search space consists of augmentation operations with associated probability and magnitude. More specifically, this can be represented as a vector of 11 tuples (o_i, p_i, m_i) (i.e. one tuple for each transform). During the training, up to two data augmentation operations o_i are drawn uniformly at random for each training sample and applied with probability p_i and magnitude m_i. As suggested by Ho et al. (2019), we set the number of operations to $0, 1$ and 2 with probabilities $0.2, 0.3$ and 0.5 respectively. The operations

are applied in the same order in which they are drawn. However, the two embedding-level noising operations (i.e. Gaussian noising and uniform noising) are always applied after the other augmentations since they must be applied in the middle of the graph, after the representation layers, whereas the other augmentations are applied directly on the input of the NN.

To allow for a smooth parametrisation of the search space with large coverage, probabilities and magnitudes can take any values between 0 and 1: $p, m \in [0, 1]$. This is different from the original PBA algorithm where the parameters are limited to discrete values. The magnitude level, which represents the intensity with which each operation is applied, is scaled down differently to fit the different operations. Maximal magnitude values are chosen so as to allow for a wide enough array of impactful values and their specific values for each augmentation are indicated in the corresponding paragraphs.

All transformation operations, along with their implementation, are detailed below. Moreover, transformations that are applied directly on the input rather than on the embeddings are illustrated in Table 6.1.

Synonym Replacement The implementation follows the one suggested by the authors of the easy data augmentation (EDA) techniques (Wei & Zou, 2019) and uses the provided codes. First, the number of words that are replaced with one of their synonyms is determined as $n_i = \lfloor \hat{m}_i * |s_i| \rfloor$, with $\hat{m}$ the magnitude level scaled down between 0 and 0.25. Then stop-words are removed from the sample. While the number of words replaced is lower than n_i, one word is selected uniformly at random among the words that have not been replaced yet and is replaced with one of its synonyms. Synonyms are retrieved with WordNet (Miller, 1998). Note that since many words have multiple meanings, it is not rare that the chosen synonym carries a different meaning than the original word.

Hypernym Replacement The process for hypernym replacement is identical to that of synonym replacement in all respect except that hypernyms instead of synonyms are extracted using WordNet.

Nearest Neighbour At the beginning of each search (and at the beginning of the final training), we feed every training sample into the pre-trained BERT and use the contextualised representation of each token to build a k-d tree. Note that this process is one-time only and is tailored to the train set. When applied to sample s_i, the *nearest-neighbour* operation first tokenizes the sample into WordPiece tokens using the BERT tokenizer before computing the number of tokens that will be replaced as follows:

	original sample ($m = 0$)	$m = 0.5$	$m = 1$
Synonym Replacement	a pretentious and ultimately empty examination of a sick and evil woman	a pretentious and ultimately empty examination of a sick and **malefic** woman	a **ostentatious** and ultimately empty **interrogatory** of a **retch** and evil woman
Hypernym Replacement	just another disjointed , fairly predictable psychological thriller	just another disjointed , fairly predictable psychological **adventure story**	just another **part** , fairly predictable psychological **heroic tale**
Nearest Neighbour	tormented by his heritage , using his storytelling ability to honor the many faceless victims	tormented by his heritage , using his storytelling ability to honor the many faceless **emotions**	tormented by his **fate** , using his **cinematic** ability to honor the many **unexplainable** victims
Contextual Augmentation	small - budget	**thin** - budget	**barely running** budget
Random Insertion (vocabulary)	actually manages to bring something new into the mix	actually manages *johns* to bring something new into the mix	*lest* actually manages to *goaltender* bring something new into the mix
Random Insertion (synonym)	is a gorgeous film - vivid with color , music and life	is a gorgeous film *brilliant* - vivid with color , music and life	*liveliness* is a gorgeous film - *medicine* vivid with color *exist* , music and life
Random Deletion	an adventurous young talent who finds his inspiration on the fringes of the american underground	an adventurous talent who finds his on the fringes of american underground	an young finds his on the of underground
Random Swap	the movie is widely seen and debated with appropriate ferocity and thoughtfulness	the movie seen widely is and debated with appropriate ferocity and thoughtfulness	movie the is widely seen ferocity thoughtfulness with appropriate and and debated
Back-translation	does n't know what it wants to be	Do not know what he wants to be	I don't know what you want to be

Table 6.1: Overview of the data augmentation transforms from our search space that operate directly on the input. This shows the outcome when the transformations are applied on samples from the Stanford Sentiment Treebank (SST)-2 dataset with three different magnitude levels $m = 0, 0.5, 1$. Where relevant, tokens that have been replaced are highlighted in bold. In addition, newly inserted tokens are italicized whereas tokens that have changed places are underlined.

$n_i = \lfloor \hat{m}_i \cdot |s_i| \rceil$. Here, $|s_i|$ corresponds to the number of WordPiece tokens in sample s_i and $\hat{m}_i = 0.25 \cdot m_i$ is the scaled-down level of magnitude, which has a maximum value

of 0.25 so that at most 25% of the tokens are replaced. Then, n_i WordPiece tokens at drawn uniformly at random. For each one of them, the 10 tokens with the nearest embeddings (in the context of sample s_i) are retrieved and one of them is selected for replacement using a geometric distribution with parameter $q = 0.5$. A geometric distribution ensures that the nearest neighbours have a higher chance to be selected as a replacement than the more distant ones. The implementation is based on the one provided by Dale (2020).

Contextual Augmentation The *contextual augmentation* transform replaces words by these predicted by an LM conditioned on the labels. In this study, we use the implementation provided by Wu et al. (2019) which uses BERT as a conditional MLM. At the beginning of the search, the model is fine-tuned on the training data on a task that applies extra label-conditional constraint to the traditional MLM objective. Once fine-tuned, the model can be used to infer masked words given a label. When applied to sample s_i, this operation replaces $n_i = \lfloor m_i \cdot |s_i| \rceil$ of the tokens with a mask, where $|s_i|$ is the number of tokens in the sample after applying the BERT tokenizer. Then, along with its label, the masked sample is fed to the fine-tuned conditional model which infers a vocabulary word for each of the masked tokens. These predictions are used as a replacement.

Random Insertion When applied to a sample s_i, the *random insertion* operation randomly adds a token to the sample. The number of tokens to insert n_i is set to a fraction of the length of s_i, namely $n_i = \lfloor \hat{m}_i \cdot |s_i| \rceil$, where $|s_i|$ is the number of tokens in s_i and $\hat{m}_i = 0.25 \cdot m_i$ is the scaled-down magnitude which ensures the number of inserted tokens does not exceed 25% of the original number of tokens and, by extension, that the new sample contains at most 20% of randomly inserted tokens. We include two independent variants: each inserted token is either a synonym of one of the tokens (selected uniformly at random) in s_i as suggested by Wei & Zou (2019) as part of their EDA techniques or is sampled uniformly at random from a subset of the BERT vocabulary. Note that we only consider words between index 1996 and 29611 of the vocabulary to exclude special and unused tokens as well as punctuation, digits and tokens with non-English characters. We also ignore tokens that start with "##" the special characters used to indicate a trailing WordPiece token. The position for the insertion of the new token in s_i is chosen uniformly at a random. The implementation uses the codes provided by Wei & Zou (2019).

Random Deletion The *random deletion* operation removes a fraction of the tokens from the sample. Each token is discarded with probability q_i, where $q_i = \hat{m}_i = 0.25 \cdot m_i$ which is the magnitude level scaled down between 0 and 0.25 to guarantee that at most half of the tokens are removed. This allows a wide range of values around the original intensity parameter of 0.1 suggested by Wei & Zou (2019) whose implementation we use.

Random Swap This augmentation swaps any two words from the sample s_i at random n_i times in a row, where $n_i = \lfloor \hat{m}_i * |s_i| \rfloor$ and $|s_i|$ is the number of tokens in sample s_i. The magnitude parameter m_i is scaled down to have a maximum value of 0.25 to ensure that at most 50% of the words are swapped. Once again, we rely on the implementation provided by Wei & Zou (2019).

Gaussian Noising The *Gaussian noising* operation is not applied on the input sequences but rather directly on the contextualised word representations. Let w_{ij} be the embedding of word j in sample s_i. Then, each embedding in the sample is transformed as follows:

$$\hat{w}_{ij} = w_{ij} + e_j, \quad e_{jk} \sim \mathcal{N}(0, \sigma^2) \quad , \tag{6.1}$$

where $\sigma = m_i$ and e_j is a vector of the length of d (embedding dimension) with elements e_{jk} normally distributed with mean 0 and standard deviation m_i.

Uniform Noising Similarly to Gaussian noising, the *uniform noising* operation is applied directly on the contextualised embeddings. More specifically,

$$\hat{w}_{ij} = w_{ij} + e_j, \quad e_{jk} \sim \mathcal{U}(-m_i, m_i) \quad . \tag{6.2}$$

Once again, e_{jk} $(0 \leq k < d)$ are the elements of noise vector e_j uniformly distributed over the half-open interval $[-m_i, m_i]$ and d is the dimension of the embeddings.

Back-translation The *back-translation* operation first translates the sample s_i to an intermediate language before translating the intermediate translation back to English. To allow us to incorporate this transform into the search space, we relate the magnitude level with the quality of the translation: when back-translation is applied with a low magnitude, the intermediate language used is one that achieves a high BLEU3 score according to Aiken (2019). Similarly, high magnitude settings back-translate samples through a language with a poor BLEU3 score. Table 6.2 summarises the languages that can be chosen for each level of magnitude. To generate translations, we use the python

m_i	Intermediate Languages
1	Portuguese, Italian, French, Czech, Swedish
2	Dutch, Maltese, Polish, Romanian, Russian
3	Afrikaans, Belarusian, Slovak, Danish, Indonesian
4	German, Albanian, Bulgarian, Japanese, Spanish
5	Chinese, Croatian, Finnish, Latvian, Arabic, Malaysian
6	Greek, Korean, Norwegian, Serbian, Turkish, Welsh
7	Galician, Icelandic, Slovenian, Vietnamese
8	Catalan, Estonian, Filipino, Hungarian, Swahili
9	Irish, Thai, Hebrew, Ukrainian, Persian
10	Lithuanian, Macedonian, Yiddish, Hindi

Table 6.2: The intermediate translation languages for each magnitude level m_i. They are separated according to inverse BLEU3 scores.

library *Googletrans* which uses the Google Translate Ajax API to make calls.

6.3.3 Search

The search is conducted on 48 epochs using around 20% of the N data points for training and the rest for validation. Both the child models and the final model follow the original uncased BERT base architecture suggested by (Devlin et al., 2019). The learning rate is chosen so as to slow down the fine-tuning without affecting the performance.

At the beginning of training, all probability and magnitude hyperparameters are set to 0. This follows suggestions by Ho et al. (2019) who postulate that little to no augmentation is needed at the beginning of the training, since the model only starts to overfit later on, and the data should increasingly become more diverse throughout the training. Because the complexity of the BERT models lies in the contextual representation layers which are already pre-trained and the task-specific layer that needs to be fine-tuned is rather simple, we keep the architecture identical both for the child models and for the final model. A key difference between applying PBA to image classification and applying PBA to NLP tasks with pre-trained BERT model is that in the former settings models are commonly trained for hundreds of epochs, whereas only two to four epochs of fine-tuning are sufficient in the latter settings. Consequently, the original strategy of running a search for 160 or 200 epochs (depending on the model and dataset) while having exploit-and-explore procedures take place after every 3 epochs is not feasible for NLP tasks with a pre-trained BERT model. Hence we modify the learning rate to slow down the fine-tuning process. More specifically, we look through a small grid search for a learning rate that can replicate the performance achieved when using the original parameters (Devlin et al., 2019) but on a larger number of epochs. Thus, by reducing the learning rate, we find a way to carry out the search over a total

of 48 epochs. The search is conducted on 16 child models that are trained in parallel. At the beginning of the search, approximately 80% of the training data are set aside to form the validation set, which will be used to periodically assess the performance of the child models. The remaining training data are used to optimise the networks. After each epoch (instead of 3 originally), the exploit-and-explore procedure takes place where the 4 worst performing (on the validation set) child models copy the weights and the parameters of the 4 best performing child models. At the end of the 48 epochs, the augmentation schedule of the model with the highest performance (on the validation set) is extracted.

6.3.4 Train

To train the final model, the train and validation data are grouped to form the final training set. Training is conducted using the same learning rate and the same number of epochs as during the search and uses the discovered schedule for augmentation. At the end of the training, the performance of the trained model is evaluated on an independent test set.

6.4 Experiments and Results

In this section, we present the details of our experiments and discuss the results. Our implementation builds on the original PBA codes (Ho et al., 2019). We make the NLP adaptation of this framework publicly available[1].

6.4.1 Datasets

To conduct our experiments, we use two of the datasets suggested by Longpre et al. (2020) in their comparative analysis, namely: SST-2 (Socher et al., 2013) and Multi-Genre Natural Language Inference Corpus (MNLI) (Williams et al., 2018). The former is a corpus of movie reviews used for sentiment analysis: single sentences from movie reviews taken from the SST (Socher et al., 2013) dataset have to be classified as positive or negative (i.e. binary classification). Performance is evaluated using accuracy. In contrast, MNLI (Williams et al., 2018) is a natural language inference corpus which consists of sentences pairs—a hypothesis sentence and a premise sentence—of miscellaneous domains that have to be classified as *entailment*, *contradiction* or *neutral*. It is therefore a non-binary classification task of sentence pairs. Similar to Longpre et al. (2020)'s work, we focus on low resource settings and use only up to 9k samples for training and

[1] https://github.com/chopardda/LDAS-NLP

validation. This allows us to use for each task a subsample of the original training set for training and validation (since the training set is larger than 9k samples) and the original validation set for testing. More specifically, we take $N = \{1500, 2000, 3000, 9000\}$ samples from the original train sets to constitute our train-validation sets and as test sets, we use the *full* original validation sets which—unlike the original test sets—contain publicly available ground-truths.

6.4.2 Implementation details

The implementation parameters can be found in Table 6.3.

	Parameters		
	epochs	learning rate	batch size
SST-2	48	1e-5	32
MNLI	48	5e-5	32

Table 6.3: Implementation details of both the child models and the final model (both with and without augmentation). The model used is the uncased BERT base model with a classification layer.

6.4.3 Results and Discussion

The main results can be found in Table 6.4 and Table 6.5.

The first row corresponds to the performance on the test set when training the model on all N samples without any augmentation. The second row in the table contains the result of the evaluation on the test set of the model trained on all N samples using the discovered schedules (i.e. for each data size, the schedule of the child model with the highest validation accuracy at the end of the search is used to train the final model). The same slowed-down learning rate and the same extended number of epochs are used across all experiments to allow for a fair comparison.

Overall, the improvements yielded by the optimised data augmentation schedules are inconsistent and unsubstantial (below 0.8%). Even though the incorporation of transforms has a small positive impact on the SST-2 dataset, it has the opposite effect on MNLI (i.e. the scores plummet by as much as 1.39%). A possible reason for these poor results might be due to the difference in settings between our experiments and the ones in the PBA study. Indeed, our search is conducted on 48 epochs as opposed to the 160 to 200 epochs suggested for image classification tasks and the exploit-and-explore procedure takes place after each epoch rather than after every 3 epochs. In addition,

	Accuracy [%]			
	$N = 1500$	2000	3000	FULL
No Augm	88.22 ±0.69	88.11 ±0.41	88.76 ±0.46	90.49 ±0.49
Schedule	88.64 ±0.47	88.60 ±0.68	89.56 ±0.40	90.91 ±0.43
Difference	*+0.42*	*+0.49*	*+0.80*	*+0.42*

Table 6.4: SST-2 test dataset.

	Mis-matched Accuracy [%]			
	$N = 1500$	2000	3000	FULL
No Augm	65.73 ±1.12	67.60 ±2.21	69.67 ±0.79	74.26 ±0.35
Schedule	64.78 ±1.20	66.21 ±0.77	68.71 ±0.49	73.95 ±0.47
Difference	*-0.95*	*-1.39*	*-0.96*	*-0.31*

Table 6.5: MNLI test dataset.

Table 6.6: Performance on SST-2 and MNLI. The model is trained 10 times independently either without augmentation or with the augmentation schedule yielded by the search. Since a single search is conducted per value of N, the reported standard deviation measures the robustness of the training procedure for a single schedule. (For more details about the robustness of the augmentation search, see Section 6.4.4.)

the size of the training datasets is very different. The size of our largest experiment is roughly the same as the size of the smallest dataset in (Ho et al., 2019).

Surprisingly, our data-driven search seems unable to reproduce the performance boost reported by Longpre et al. (2020) on the MNLI dataset, even though the augmentations considered in their work are part of our search space. This might be explained by the way augmentations are applied. In our study, we transform each training sample by applying up to 2 transformations in a row with probability p and magnitude m. In contrast, Longpre et al. (2020) add $N \times \tau$ augmented samples to the training set with $\tau \in \{0.5, 1, 1.5, 2\}$, meaning that the original examples are provided along with their augmented counterparts at each iteration.

In general, it is expected that not every data augmentation operation is relevant for every task. For example, when the goal of a task is to determine whether a sentence is syntactically correct or not, changing a word with a synonym will not change the syntax

of the sentence, even if that synonym carries a different meaning than the original word. In contrast, swapping two words will most likely disturb the syntactic structure of the sentence and might change the ground-truth label. However, when considering a different task, the opposite might be true. For example, when dealing with a topic-modelling task, swapping words has no effect on the label, but replacing them with a synonym might change the topic and, therefore, the label. One could therefore expect that different augmentation operations will be chosen for different tasks but this trend was not observed in our experiments.

6.4.4 Search Robustness

Given the stochastic nature of the searching process, the discovered schedule is bound to differ from one run to another. So far, we have run a single search for each experiment setting. In this section, we investigate whether the limited effect of automatic augmentation on the model performance may be caused by the stochasticity of the search. To that end, we run 10 independent searches on the SST-2 dataset with $N = 1500$ and use each of the 10 discovered schedules to train a separate model. All the network hyperparameters are kept the same as in the previous section. Overall, the standard deviation over 10 independent schedules is 0.55%, which indicates that the performance of the training is robust across searches. Thus, the poor results observed in the previous section cannot be explained by the variability of schedules.

However, a closer look at these 10 individual schedules—as illustrated in Figure 6.2—reveals that the chosen augmentation hyperparameters are very different from one run to another and that the search does not seems to favour any particular set of augmentation transforms. Note that, for a more compact representation, the product of the magnitude and the probability hyperparameters through the epochs is shown for each schedule. In Figure 6.5 the average magnitude and probability parameters over the 10 schedules at each epoch is displayed.

These plots allow us to realise that, while the parameters generally increase throughout the epochs, the magnitudes and probabilities of each transform have a similar value. Although some operations (e.g. Random Swap) have slightly higher average parameters than others, we can see that no augmentation transform clearly dominates the others. This variation in the optimal set of hyperparameters for each independent search may indicate that, in this setting, data augmentation acts more as a regulariser rather than a way to learn invariance properties and that, as a result, any kind of augmentation transform has a similar effect on performance. In view of these findings, it would be interesting to explore whether relying on a greater number of child models during the search could potentially yield less disparate schedules and improve the overall quality of

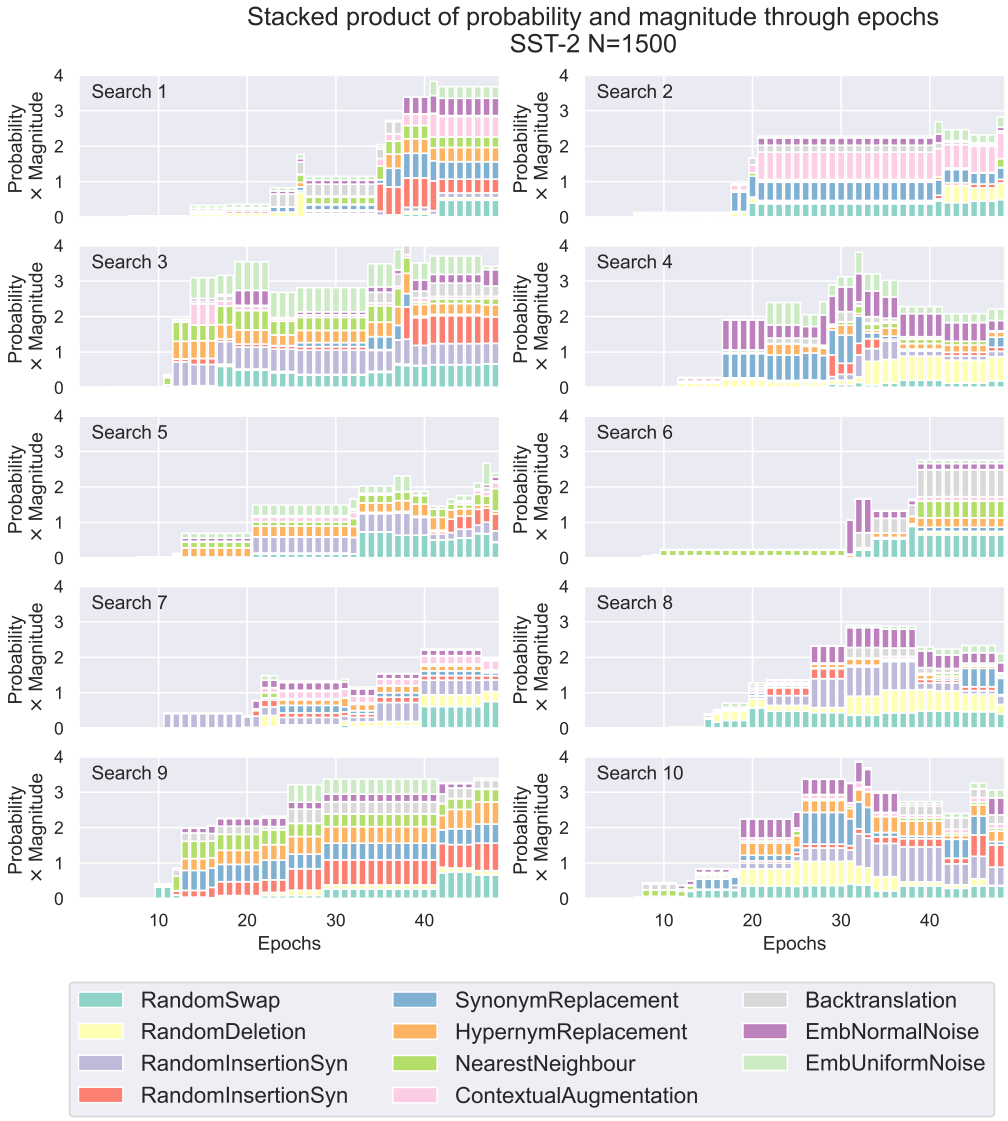
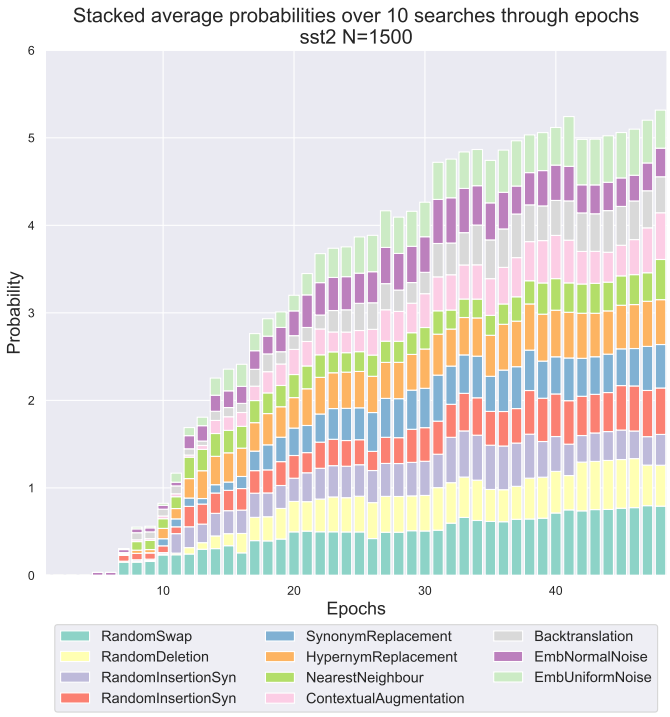
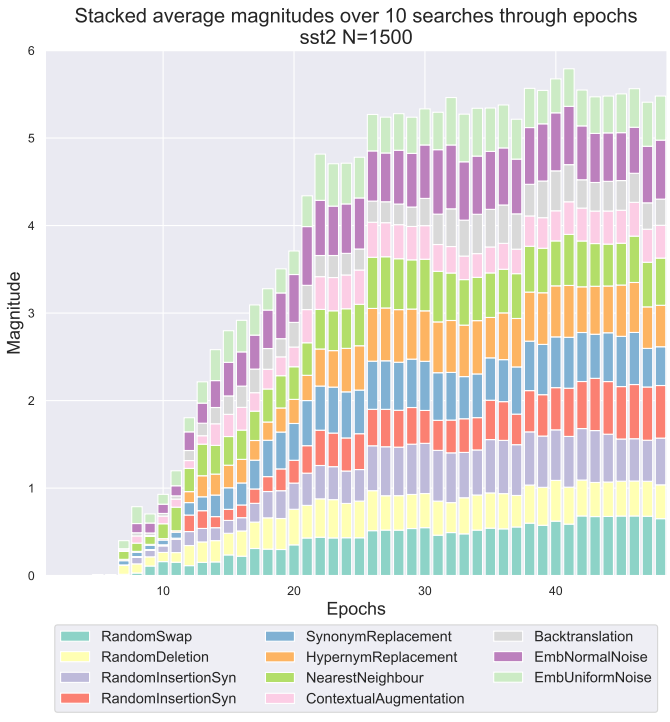


Figure 6.2: The schedules yielded by 10 independent searches on SST-2 with $N=1500$ samples (using 250 for training and 1250 for validation during the search). The height of each bar corresponds to the product of the probability and the magnitude parameters at each epoch.



1

Figure 6.3: Average probabilities.



1

Figure 6.4: Average magnitudes.

Figure 6.5: The average probability and magnitude values for the schedules yielded by 10 independent searches on SST-2 with $N{=}1500$ samples. The height of each bar corresponds to the average probability and magnitude parameters at each epoch over the 10 schedules.

the search.

6.4.5 Validation size

As mentioned earlier, the limited impact of automatic data augmentation scheduling in our settings might be due to the small number of samples available for each experiment. In particular, one of the drawbacks of PBA is that a large portion of training data (approximately 80% as suggested by Ho et al. (2019)) has to be set aside to form a validation set that is used during the search to find optimal hyperparameters. For example, at $N{=}1500$ only 250 examples are used to learn the network weights during the search while the remaining 1250 samples are used for hyperparameter selection. As a result of this discrepancy, the selected data augmentation might be relevant when only 250 data points are available for training but less effective when learning with 1500 samples as is ultimately the case.

Overall, the main idea behind using a large validation set is to choose augmentation hyperparameters that do not overfit the validation samples and thus generalise well to unseen data. However, in our case, since the total number of available samples N is small in all experiments, this implies that the size of the training set will be extremely limited. This might hinder the learning process (with too few training examples it can be difficult to learn the optimal network weights) or make the choice of augmentation hyperparameters irrelevant for larger training sets (there is no guarantee that the augmentation chosen for the small train set with will also help when ultimately training with both the training and the validation set). Thus, there exists a clear trade-off between the size of the two sets: while a large validation set can allow for better optimisation of the augmentation hyperparameters, a larger training set allows for better optimisation of the network weights which, in turn, has an impact on the quality of the augmentation hyperparameters evaluation.

In this section, we investigate whether the poor results observed in Table 6.6 can be attributed to the ratio chosen to split the available data into a train and a validation set. To that end, we run the search on the SST-2 dataset using different ratios to divide the $N{=}1500$ samples at hand. The results reported in Table 6.7 suggest that using different proportions of train and validation examples does not affect the effectiveness of the augmentation schedule in this setting. In fact, the performance remains the same even though the model is trained with schedules that were optimised using very different split ratios. This might be explained by the fact that both the train and the validation sets are too small to find optimal augmentation hyperparameters irrespective of the chosen split ratio. Alternatively, it is possible that the chosen dataset can simply not benefit from augmentation because of its nature. To verify this hypothesis, it would

TRAIN/VAL	250/1250	750/750	1250/250
ACCURACY	88.64 ±0.47	88.64 ±0.62	88.76 ±0.59

Table 6.7: Performance on the SST-2 test set of the model trained on N=1500 samples with the schedule discovered using different proportions of validation and training sets. For each split ratio, the model is trained 10 times using the schedule yielded by a single search. The mean accuracy and standard deviation are reported.

be interesting to extend this analysis to a wider range of datasets, including actual low-resource datasets.

6.5 Conclusion

In this chapter, we have presented the first work that investigates the automatic search of data augmentation schedules for text data. However, the results suggest that augmentation schedules and data-driven parameter search do not provide a consistent and straightforward way to improve the performance of NLP models that use pre-trained transformers. There are a few possible explanations for this phenomenon. First, the overall setup of the PBA approach (e.g. the need for large validation sets) might not be well suited for low-data regimes in NLP. A second but more likely reason is that transformers are already pre-trained on huge datasets and their representations may already be invariant to many of the transformations that are encoded into the data augmentation. A systematic investigation into the latter hypothesis is required, which, if proven, would show that data augmentation may be redundant when opting to use transformers to implement NLP solutions. A final reason might be that the search space we consider only contains transformative data augmentation techniques and omits generative ones, even though the latter have started to show some promising results.

The findings made in this chapter suggest that pre-trained LMs do not benefit from simple data augmentation during fine-tuning, thereby offering a negative answer to our third RQ. Even though this is a negative findings, it has got significant implications. It indicates that pre-trained LMs already provide a comprehensive understanding of language on their own. This confirms the findings from Chapter 3, where the use of pre-trained LMs enabled an understanding of context that was effective enough to differentiate serious adverse events from other underlying symptoms.

Chapter 7

Conclusion and Future Work

In this thesis, we investigated the problem of leveraging DL techniques for clinical text mining despite the various challenges associated with these kinds of data including data scarcity, frequent use of abbreviations and their ambiguity, OOV words and term variation. This chapter summarises this thesis' main contributions, discusses how we tackled the RQs, and proposes new avenues for extending this research.

RQ1. Can an effective DL strategy be developed to recognise references to rare clinical events?

As highlighted throughout this work, the issue of data scarcity arises as a consequence of privacy safeguards and the manual annotation bottleneck. This problem is further exacerbated in clinical applications such as clinical trials that are concerned with rare events, of which, by their definition, there are fewer instances. This problem was addressed by the first RQ in Chapter 3, where we used SAEs as a case study for the issue of rare clinical events. In addition to their rarity, the task of SAE detection is further complicated by the fact that SAEs are a subset of signs and symptoms, whose mentions are plentiful in clinical narratives. Therefore, the internal characteristics of SAEs, which can be viewed as named entities, are bound to be insufficient for distinguishing signs and symptoms monitored by a given clinical trial from those arising from factors. These factors are typically related to the underlying disease that is the subject of the clinical trial. As an additional challenge, any given sign or symptom may represent an SAE in some contexts but not in others. Therefore, any system aiming to detect SAEs must be able to utilise context as the main source of features.

In addition to their ambiguity, SAEs are also plagued by the problem of term variation, which is quite common in the biomedical domain (Krauthammer & Nenadic, 2004, Zheng et al., 2015, D'Souza & Ng, 2015). For example, the terms "heart attack" and "myocardial infarct" are equivalent. When the training data are scarce, the distributional

hypothesis, which assumes that there is a correlation between distributional similarity and meaning similarity, is bound to be of little practical value. If the synonymy of terms is ignored, then any subsequent statistical analysis of the corresponding concepts, in this case SAEs, is bound to be skewed. To deal with this issue, we leveraged an ontology as an explicit representation of domain knowledge to simplify the problem of SAE detection. As we have already mentioned above, SAEs are a subclass of signs and symptoms, which are defined as a separate class in the UMLS, the largest ontology of its kind. It can be used to focus the attention of the system on these particular instances in free text, allowing it to concentrate its effort on the analysis of their contexts, which as we said above represent the main source of features for the classification of SAEs. Therefore, we formulated the given problem as a binary classification task. All signs and symptoms are first identified and mapped to unique UMLS concepts. Then, the classifier determines which concepts refer to SAEs and which do not.

Given the rarity of SAEs, the issue of data scarcity still persists. The binary classifier would, therefore, struggle to generalise when trained on such a small dataset. Large pre-trained LMs already generalise word properties by building on their distribution in large corpora. By fine-tuning a large LM to perform binary classification, we can leverage its generalisability when using a small training dataset. Along these lines, we fine-tuned BERT to differentiate between the UMLS concepts which refer to SAEs and those which do not.

To test these hypotheses, we evaluated the effectiveness of this combination of a top-down approach based on an ontology (representing explicit knowledge about the domain) and a bottom-up approach based on a large LM (representing implicit knowledge about the language). We used a very small real-world clinical trial dataset of less than 300 documents in total. Nevertheless, we successfully trained a system to recognise the minority class of SAEs among the majority of all other signs and symptoms. The key research contribution here represents a successful integration of two fundamentally different approaches to word semantics that allow a system to make generalisations of instances available in a training dataset despite its small size.

RQ2. Can an effective DL strategy be developed to normalise clinical text by automatically expanding short forms?

The prevalence of short forms is another key characteristic of clinical texts. It is the result of time pressure and repetitiveness associated with the reporting task. We differentiate between two types of short forms, abbreviations and acronyms as they differ in their formation patterns. The corresponding approaches were described separately in two chapters, Chapter 4 and Chapter 5. Short forms present two important challenges to

DL approaches to clinical text mining. First, they may not be present in the vocabulary of a corpus used for training or an external vocabulary as they are often ad hoc in nature. Second, the smaller the number of characters in a short form, the higher the probability that it will collide with another short form, which introduces the problem of ambiguity.

As we have seen in RQ1, large LMs can be used to compensate for the lack of sufficiently large training data. In other words, they are pre-trained on large corpora, which have much greater coverage of the language measured by the size of the corresponding vocabulary. That means that even though a particular word is not present in the clinical dataset, its representation (i.e. embedding) may still be available in the pre-trained LM. However, this often does not apply to abbreviations as they may be specific to an institution or even an individual (Moon et al., 2015). To a certain extent, LMs can cope with this issue by inferring the embedding of an unknown word (in this case abbreviation) from its context. Still, this does not help text interpretability from the human perspective, which would prefer the abbreviation to be expanded to its long form rather over its vector in the embedding space. This leads us to the main problem approached in Chapter 4, which focuses on finding abbreviations and mapping them explicitly to their long forms. Our approach is based on a hypothesis that the long form (i.e. the original nonabbreviated word) is likely to occur elsewhere in a corpus. The problem can therefore be framed as the problem of finding the long form within the corpus. In addition to the similarity between an abbreviation and its long form (e.g. shared characters and the order in which they appear), this approach lends itself to checking whether these two forms are semantically similar by comparing their contexts. Specifically, we used the WMD as it can compare two contexts (e.g. sentences) in a meaningful way even when they have no words in common. The problem of finding plausible candidates for long forms to compare against was approached by using a DL approach. Specifically, a Siamese NN was chosen because of its ability to handle unseen classes (Koch et al., 2015, Huang et al., 2022), which enables it to correctly interpret new abbreviations and expansion candidates after training. In addition, Siamese NNs focus on learning representations by relying on similarity information (Neculoiu et al., 2016). By learning semantic similarity, such network constrains the number of long form candidates to be compared using more costly WMD. The key research contribution here represents a novel approach to abbreviation expansion that departs from the traditional way in which this problem is tackled using a domain specific dictionary to find plausible interpretations and a labelled corpus to supervise the process. This makes our approach more robust especially against OOV abbreviations. Moreover, our approach can be easily ported to other domains as it does not depend on a domain-specific dictionary.

We dedicated a separate chapter to another type of short forms. In Chapter 5

we described our approach to acronym disambiguation. Because of data scarcity, it is often difficult to have access to large clinical datasets for training disambiguation models. In contrast, biomedical abstracts, which are known to contain many acronyms, are widely available. In particular, in these abstracts acronyms and their full form are explicitly defined contrary to clinical texts. Consequently, we suggested leveraging this characteristic to simulate a dataset of clinical acronyms from biomedical abstracts. This dataset can then be used to train classifier to disambiguate between expansion candidates. One important component for finding the best expansion candidate is to correctly interpret the context that the acronym appears in. This is why, similar to Chapter 3, we proposed using a pre-trained LM as a classifier. Indeed, our experiments have shown that such architecture was able to understand the context of clinical events. Because we relied on a large number of simulated samples for training our network, we retrieved expansion candidates directly from the training examples. Alternatively, we suggested retrieving expansion candidates using Flexiterm in order to make this step unsupervised. Indeed, Flexiterm offers a way of automatically extracting group of words, which makes it very relevant for acronyms. The whole approach was validated on a dataset of clinical notes. Our contribution was to introduce a suitable strategy to face the problem of data scarcity by finding a way of generating data and use these data for acronym disambiguation. By doing so, we addressed the final part of the second RQ. Indeed, the approach we proposed makes possible the automatic expansion of acronyms despite a lack of existing datasets.

Overall, we have demonstrated the possibility of developing effective DL strategies to expand all short forms automatically.

RQ3. Can a label-preserving transformation of existing data improve the performance of DL approaches to text mining that are based on pre-trained LMs?

In Chapter 3, we suggested using knowledge from pre-trained LMs as a way of address the issue of data scarcity. Alternatively, we can try to augment the original training dataset by creating new samples by artificially varying the existing ones. Here, synthetic samples should follow the same distribution as the existing data at hand. Even though this approach in already very popular in the field of computer vision, its suitability for NLP applications remains underexplored. This leads us to the third and final RQ in which we investigate the potential of data augmentation for pre-trained LM representations.

More specifically, in Chapter 6, we leveraged PBA to find data augmentation schedules for a pre-trained LM-based classifier. To that end, we chose two common NLP tasks which we aimed to solve with an extremely limited amount of data. To the best

of our knowledge, this represents the first systematic approach of automatically scheduling augmentation for text data. Unfortunately, the results revealed that label-preserving transformations of text data did not consistently improve the performance of pre-trained LMs on this set of NLP tasks. The most plausible explanation for this negative finding is that the types of invariance incorporated by data augmentation may have already been captured by pre-trained LMs. In other words, large pre-trained models already capture the asymptotic behaviour of the underlying NNs leaving little room for further improvement. Another possible explanation is that fine-tuning on small datasets might be instable (Zhang et al., 2020a). Despite this negative finding, our approach may still find practical applications using clinical text data. Many researchers are still using bag-of-words or context-free embedding approaches, which would certainly benefit from augmenting the training data.

7.1 Limitations and Future Work

Each chapter came with its own limitations specific to each question we were trying to address. For instance, the performance of the model that was developed in Chapter 3 to detect SAEs was bounded by the capacity of the MetaMap algorithm to detect all kinds of events. The number of FNs could hence be reduced by finding ways to increase the number of events detected by MetaMap. In addition, the model was unable to correctly deal with negations. This was most likely due to the lack of training data and, thus, we suggested dealing with all negations separately, on top of the classifier.

The abbreviation expansion approach introduced in Chapter 4 also had some drawbacks. For instance, one needed to feed each full form separately so that the Siamese NN could determine whether it was a suitable expansion candidate or not. A solution was to feed every word in the dictionary, but it could be interesting to investigate more efficient approaches. Similarly, when disambiguating between expansion candidates for an abbreviation in a given sentence, one computed the minimal distance between that sentence and a set of other sentences in which the full form appeared. This implies that one must have access to a corpus of sentences from which such a set of sentences can be extracted. Therefore, looking into approaches that do not rely on external knowledge could be an exciting avenue for research. Because we fine-tuned a pre-trained LM for binary classification to achieve acronym disambiguation in Chapter 5, we relied on a supervised setting. Neverthelessss, we showed that it did not prevent the system from handling short forms or candidates that were not seen in the training data. Overall, because we chose to handle abbreviations and acronyms separately we dismissed edge cases where a short-form could refer to either a single word or a group of words

depending on context (e.g. "pt" for "patient" and "physical therapy").

Finally, the negative insight obtained in Chapter 6 could be the result of the limitations that came with the PBA approach we adopted for discovering an augmentation schedule. The first limitation of PBA is that it was developed for computer vision tasks which require training for hundreds of epochs. In contrast, LM fine-tuning can be achieved in a few epochs only. It would be interesting to apply this approach to different settings which require longer fine-tuning to achieve convergence. Second, we chose to limit our search space to transformation operations, as these techniques are easy to apply and have proven popular in other fields. However, this does not imply that generative data augmentation approaches could not be extremely beneficial to create synthetic samples. This is definitely a path that needs exploring. Finally, another avenue of research could be to use the PBA algorithm to jointly find a set of transforms to apply not only during training but also at inference time as a kind of ensemble method.

On top of the limitations specific to each aspect investigated, this thesis comes with some limitations of its own. Given the wide scope of the main hypothesis, we chose to focus on three RQs which we felt were particularly relevant to the field of clinical text mining. However, other paths that could be worth exploring. For instance, one could research the topic of active learning to carefully select a few samples which, once labelled, could help drive the model's performance. This could help limit the annotation bottleneck as the model learns to focus on the most informative samples for learning. Another important research area which could help the development of DL models on clinical text data is few-shot learning. This corresponds to a variety of machine learning models which precisely aim to train NNs with a very limited number of samples. We explored this topic to some extent when we used a Siamese NN for finding abbreviation expansion candidates. Indeed, Siamese NNs rely on similarity to learn to discriminate between unseen classes. In our work, a Siamese NN was trained to learn how abbreviations related to their full form. This means that, in theory, the NN had the capacity to apply this pattern to any new abbreviation without the need for further training. However, this is only a small example of the many existing few-short learning techniques that should be delved into further.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. TensorFlow: a system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

Agirre, E. and Stevenson, M. Knowledge sources for wsd. In *Word Sense Disambiguation*, pp. 217–251. Springer, 2007.

Aiken, M. An updated evaluation of google translate accuracy. *Studies in linguistics and literature*, 3(3):253–260, 2019.

Allen, C. and Hospedales, T. Analogies explained: Towards understanding word embeddings. In *International Conference on Machine Learning*, pp. 223–231. PMLR, 2019.

Allen, D. M. The relationship between variable selection and data agumentation and a method for prediction. *technometrics*, 16(1):125–127, 1974.

Allvin, H., Carlsson, E., Dalianis, H., Danielsson-Ojala, R., Daudaravičius, V., Hassel, M., Kokkinakis, D., Lundgrén-Laine, H., Nilsson, G. H., Nytrø, Ø., et al. Characteristics of finnish and swedish intensive care nursing narratives: a comparative analysis to support the development of clinical language technologies. In *Journal of Biomedical Semantics*, volume 2, pp. 1–11. Springer, 2011.

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jindi, D., Naumann, T., and McDermott, M. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 72–78, 2019.

Amin-Nejad, A., Ive, J., and Velupillai, S. Exploring transformer text generation for medical dataset augmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4699–4708, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.578`.

Andersson, L., Hanbury, A., and Rauber, A. The portability of three types of text mining techniques into the patent text genre. In *Current Challenges in Patent Information Retrieval*, pp. 241–280. Springer, 2017.

Ao, H. and Takagi, T. ALICE: an algorithm to extract abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 12(5):576–586, 2005.

Aronson, A. R. and Lang, F.-M. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3): 229–236, 2010.

Aroyehun, S. T. and Gelbukh, A. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 90–97, 2018.

Bahdanau, D., Cho, K. H., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., Chapman, B., Amrhein, T., Mong, D., Rubin, D. L., et al. Comparative effectiveness of convolutional neural network (cnn) and recurrent neural network (rnn) architectures for radiology text report classification. *Artificial intelligence in medicine*, 97:79–88, 2019.

Bansal, M. A., Sharma, D. R., and Kathuria, D. M. A systematic review on data scarcity problem in deep learning: Solution and applications. *ACM Computing Surveys (CSUR)*, 2021.

Bashar, M. A., Nayak, R., and Suzor, N. Regularising lstm classifier by transfer learning for detecting misogynistic tweets with small training set. *Knowledge and Information Systems*, 62(10):4029–4054, 2020.

Beam, A. L., Kompa, B., Schmaltz, A., Fried, I., Weber, G., Palmer, N., Shi, X., Cai, T., and Kohane, I. S. Clinical concept embeddings learned from massive sources of multimodal medical data. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2020*, pp. 295–306. World Scientific, 2019.

Belousov, M., Milosevic, N., Dixon, W., and Nenadić, G. Extracting adverse drug reactions and their context using sequence labelling ensembles in tac2017. 2017.

Beltagy, I., Lo, K., and Cohan, A. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3615–3620, 2019.

Bengio, Y., Simard, P., and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *Advances in Neural Information Processing Systems*, 13, 2000.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Bhatia, P., Celikkaya, B., and Khalilia, M. Joint entity extraction and assertion detection for clinical text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 954–959, 2019.

Bird, S., Klein, E., and Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, 2009.

Bodenreider, O. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270, 2004.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5: 135–146, 2017.

Botsis, T., Nguyen, M. D., Woo, E. J., Markatou, M., and Ball, R. Text mining for the vaccine adverse event reporting system: medical text classification using informative feature selection. *Journal of the American Medical Informatics Association*, 18(5): 631–638, 2011.

Botsis, T., Buttolph, T., Nguyen, M. D., Winiecki, S., Woo, E. J., and Ball, R. Vaccine adverse event text mining system for extracting features from vaccine safety reports. *Journal of the American Medical Informatics Association*, 19(6):1011–1018, 2012.

Bowman, S., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning (CoNLL).*, 2016.

Branco, P., Torgo, L., and Ribeiro, R. P. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50, 2016.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Button, K., Van Deursen, R. W., Soldatova, L., and Spasić, I. Trak ontology: defining standard care for the rehabilitation of knee conditions. *Journal of Biomedical Informatics*, 46(4):615–625, 2013.

Chang, J. T., Schütze, H., and Altman, R. B. Creating an online dictionary of abbreviations from MEDLINE. *Journal of the American Medical Informatics Association*, 9 (6):612–620, 2002.

Chapman, W. W., Nadkarni, P. M., Hirschman, L., D'avolio, L. W., Savova, G. K., and Uzuner, O. Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions, 2011.

Charbonnier, J. and Wartena, C. Using word embeddings for unsupervised acronym disambiguation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2610–2619. Association for Computational Linguistics, 2018.

Chee, B. W., Berlin, R., and Schatz, B. Predicting adverse drug events from personal health messages. In *AMIA Annual Symposium Proceedings*, volume 2011, pp. 217. American Medical Informatics Association, 2011.

Chen, H., Liu, X., Yin, D., and Tang, J. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

Chen, Y. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

Cheng, Y., Tu, Z., Meng, F., Zhai, J., and Liu, Y. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1756–1766, 2018.

Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pp. 166–174, 2016.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, 2014a.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014b.

Chopard, D. and Spasić, I. A deep learning approach to self-expansion of abbreviations based on morphology and context distance. In *International Conference on Statistical Language and Speech Processing*, pp. 71–82. Springer, 2019.

Chopard, D., Treder, M. S., Corcoran, P., Ahmed, N., Johnson, C., Busse, M., and Spasić, I. Text mining of adverse events in clinical trials: Deep learning approach. *JMIR Medical Informatics*, 9(12):e28632, 2021a.

Chopard, D., Treder, M. S., and Spasić, I. Learning data augmentation schedules for natural language processing. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pp. 89–102, 2021b.

Ciosici, M. R., Sommer, T., and Assent, I. Unsupervised abbreviation disambiguation. *arXiv preprint arXiv:1904.00929*, 2019.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2019.

Cocos, A. and Masino, A. J. Combining rule-based and neural network systems for extracting adverse reactions from drug labels. In *TAC*, 2017.

Cocos, A., Fiks, A. G., and Masino, A. J. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *Journal of the American Medical Informatics Association*, 24(4):813–821, 2017a.

Cocos, A., Qian, T., Callison-Burch, C., and Masino, A. J. Crowd control: effectively utilizing unscreened crowd workers for biomedical data annotation. *Journal of biomedical informatics*, 69:86–92, 2017b.

Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

Combi, C., Zorzi, M., Pozzani, G., Arzenton, E., and Moretti, U. Normalizing spontaneous reports into MedDRA: Some experiments with MagiCoder. *IEEE Journal of Biomedical and Health Informatics*, 23(1):95–102, 2018.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 3079–3087. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning.pdf`.

Dale, D. GitHub repository, 2020. URL `https://gist.github.com/avidale/c6b19687d333655da483421880441950`.

Dale, R. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1):113–118, 2021.

Dalianis, H. *Clinical text mining: Secondary use of electronic patient records*. Springer Nature, 2018.

Dalianis, H., Hassel, M., and Velupillai, S. The stockholm EPR corpus-characteristics and some initial findings. *Proceedings of ISHIMR*, pp. 243–249, 2009.

Dandala, B., Mahajan, D., and Devarakonda, M. V. Ibm research system at tac 2017: Adverse drug reactions extraction from drug labels. In *TAC*, 2017.

Dao, T., Gu, A., Ratner, A., Smith, V., De Sa, C., and Ré, C. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pp. 1528–1537. PMLR, 2019.

De Vine, L., Kholghi, M., Zuccon, G., Sitbon, L., and Nguyen, A. Analysis of word embeddings and sequence features for clinical information extraction. In *Australasian Language Technology Association Workshop 2015: Proceedings of the Workshop*, pp. 21–30. Australasian Language Technology Association (ALTA), 2015.

Deleger, L., Li, Q., Lingren, T., Kaiser, M., Molnar, K., Stoutenborough, L., Kouril, M., Marsolo, K., Solti, I., et al. Building gold standard corpora for medical natural language processing tasks. In *AMIA Annual Symposium Proceedings*, volume 2012, pp. 144. American Medical Informatics Association, 2012.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deschacht, K., De Belder, J., and Moens, M.-F. The latent words language model. *Computer Speech & Language*, 26(5):384–409, 2012.

Dessi, D., Helaoui, R., Kumar, V., Reforgiato Recupero, D., and Riboni, D. Tf-idf vs word embeddings for morbidity identification in clinical notes: An initial study. In *1st Workshop on Smart Personal Health Interfaces, SmartPhil 2020*, volume 2596, pp. 1–12. CEUR-WS, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

Dingwall, N. and Potts, C. Mittens: an extension of glove for learning domain-specialized representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 212–217, 2018.

DiPietro, R. and Hager, G. D. Chapter 21 - deep learning: Rnns and lstm. In Zhou, S. K., Rueckert, D., and Fichtinger, G. (eds.), *Handbook of Medical Image Computing and Computer Assisted Intervention*, The Elsevier and MICCAI Society Book Series, pp. 503–519. Academic Press, 2020. ISBN 978-0-12-816176-0. doi: https://doi.org/10.1016/B978-0-12-816176-0.00026-0. URL https://www.sciencedirect.com/science/article/pii/B9780128161760000260.

Dirkson, A. and Verberne, S. Transfer learning for health-related Twitter data. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pp. 89–92, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3212. URL https://aclanthology.org/W19-3212.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

Doumont, J.-L., Grossenbacher, L., Matta, C., and Cham, J. English communication for scientists. *Cambridge, MA: NPG Education*, 2010.

Dozat, T. Incorporating nesterov momentum into adam. 2016.

Du, J., Xiang, Y., Sankaranarayanapillai, M., Zhang, M., Wang, J., Si, Y., Pham, H. A., Xu, H., Chen, Y., and Tao, C. Extracting postmarketing adverse events from safety reports in the vaccine adverse event reporting system (vaers) using deep learning. *Journal of the American Medical Informatics Association*, 28(7):1393–1400, 2021.

Du, N., Chen, K., Kannan, A., Tran, L., Chen, Y., and Shafran, I. Extracting symptoms and their status from clinical conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 915–925, 2019.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Duke, J. D. and Friedlin, J. Adessa: a real-time decision support service for delivery of semantically coded adverse drug event data. In *AMIA Annual symposium proceedings*, volume 2010, pp. 177. American Medical Informatics Association, 2010.

D'Souza, J. and Ng, V. Sieve-based entity linking for the biomedical domain. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 297–302, 2015.

Edunov, S., Ott, M., Auli, M., and Grangier, D. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, 2018.

Emadzadeh, E., Sarker, A., Nikfarjam, A., and Gonzalez, G. Hybrid semantic analysis for mapping adverse drug reaction mentions in tweets to medical terminology. In

AMIA Annual Symposium Proceedings, volume 2017, pp. 679. American Medical Informatics Association, 2017.

Er, M. J., Zhang, Y., Wang, N., and Pratama, M. Attention pooling-based convolutional neural network for sentence modelling. *Information Sciences*, 373:388–403, 2016.

Fadaee, M. and Monz, C. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 436–446, 2018.

Fadaee, M., Bisazza, A., and Monz, C. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 567–573, 2017.

Fan, Y., Zhou, S., Li, Y., and Zhang, R. Deep learning approaches for extracting adverse events and indications of dietary supplements from clinical text. *Journal of the American Medical Informatics Association*, 28(3):569–577, 2021.

Fandrych, I. Submorphemic elements in the formation of acronyms, blends and clippings. *Lexis. Journal in English Lexicology*, (2), 2008.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. A survey of data augmentation approaches for NLP. *Findings of ACL*, 2021.

Ferraro, J. P., Daumé III, H., DuVall, S. L., Chapman, W. W., Harkema, H., and Haug, P. J. Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. *Journal of the American Medical Informatics Association*, 20(5):931–939, 2013.

Filimonov, M., Chopard, D., and Spasić, I. Simulation and annotation of global acronyms. *Bioinformatics*, 38(11):3136–3138, 2022.

Finkel, J. R., Grenager, T., and Manning, C. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 363–370. Association for Computational Linguistics, 2005.

Finley, G. P., Pakhomov, S. V., McEwan, R., and Melton, G. B. Towards comprehensive clinical abbreviation disambiguation using machine-labeled training data. In *AMIA Annual Symposium Proceedings*, volume 2016, pp. 560. American Medical Informatics Association, 2016.

Firth, J. R. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

Ford, E., Curlewis, K., Squires, E., Griffiths, L. J., Stewart, R., and Jones, K. H. The potential of research drawing on clinical free text to bring benefits to patients in the united kingdom: A systematic review of the literature. *Frontiers in Digital Health*, 3: 606599, 2021.

Friedman, C., Kra, P., and Rzhetsky, A. Two biomedical sublanguages: a description based on the theories of zellig harris. *Journal of biomedical informatics*, 35(4):222–235, 2002.

Gale, W. A., Church, K., and Yarowsky, D. One sense per discourse. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

Gao, Y. Dynamic dnn nodes drop-out rate analysis. In *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pp. 691–695. IEEE, 2021.

Gaudan, S., Kirsch, H., and Rebholz-Schuhmann, D. Resolving abbreviations to their senses in MEDLINE. *Bioinformatics*, 21(18):3658–3664, 2005.

Gehrmann, S., Dernoncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., Foote Jr, J., Moseley, E. T., Grant, D. W., Tyler, P. D., et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PloS one*, 13(2):e0192360, 2018.

Gharebagh, S. S., Goharian, N., and Filice, R. Attend to medical ontologies: Content selection for clinical abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1899–1905, 2020.

Giridhara., P. K. B., Mishra., C., Venkataramana., R. K. M., Bukhari., S. S., and Dengel., A. A study of various text augmentation techniques for relation classification in free text. In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, pp. 360–367. INSTICC, SciTePress, 2019. ISBN 978-989-758-351-3. doi: 10.5220/0007311003600367.

Gligorijevic, D., Stojanovic, J., Satz, W., Stojkovic, I., Schreyer, K., Del Portal, D., and Obradovic, Z. Deep attention model for triage of emergency department patients. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 297–305. SIAM, 2018.

Goeuriot, L., Suominen, H., Kelly, L., Miranda-Escalada, A., Krallinger, M., Liu, Z., Pasi, G., Gonzalez Saez, G., Viviani, M., and Xu, C. Overview of the clef ehealth evaluation lab 2020. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11*, pp. 255–271. Springer, 2020.

Goldberg, Y. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309, 2017.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Grivas, A., Alex, B., Grover, C., Tobin, R., and Whiteley, W. Not a cute stroke: analysis of rule-and neural network-based information extraction systems for brain radiology reports. In *Proceedings of the 11th international workshop on health text mining and information analysis*, pp. 24–37, 2020.

Gu, X., Ding, C., Li, S., and Xu, W. Bupt-pris system for tac 2017 event nugget detection, event argument linking and adr tracks. In *TAC*, 2017a.

Gu, X., Ding, C., Li, S., and Xu, W. Bupt-pris system for tac 2017 event nugget detection, event argument linking and adr tracks. In *TAC*, 2017b.

Han, L., Ball, R., Pamer, C. A., Altman, R. B., and Proestel, S. Development of an automated assessment tool for medwatch reports in the fda adverse event reporting system. *Journal of the American Medical Informatics Association*, 24(5):913–920, 2017.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Harkema, H., Dowling, J. N., Thornblade, T., and Chapman, W. W. Context: an algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of biomedical informatics*, 42(5):839–851, 2009.

Harris, Z. S. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Harris, Z. S. Theory of language and information: a mathematical approach. 1991.

Hazlehurst, B., Naleway, A., and Mullooly, J. Detecting possible vaccine adverse events in clinical notes of the electronic medical record. *Vaccine*, 27(14):2077–2083, 2009.

He, B., Guan, Y., and Dai, R. Classifying medical relations in clinical text via convolutional neural networks. *Artificial intelligence in medicine*, 93:43–49, 2019.

He, H. and Ma, Y. Imbalanced learning: foundations, algorithms, and applications. 2013.

He, Y. Extracting topical phrases from clinical documents. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Henriksson, A., Kvist, M., Dalianis, H., and Duneld, M. Identifying adverse drug event information in clinical notes with distributional semantic representations of context. *Journal of biomedical informatics*, 57:333–349, 2015.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

Hinton, G. E. Distributed representations. 1984.

Hirvonen, E., Karlsson, A., Saaresranta, T., and Laitinen, T. Documentation of the patient's smoking status in common chronic diseases–analysis of medical narrative reports using the ulmfit based text classification. *European Clinical Respiratory Journal*, 8(1):2004664, 2021.

Ho, D., Liang, E., Chen, X., Stoica, I., and Abbeel, P. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pp. 2731–2741, 2019.

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

Holper, S., Barmanray, R., Colman, B., Yates, C. J., Liew, D., and Smallwood, D. Ambiguous medical abbreviation study: challenges and opportunities. *Internal medicine journal*, 50(9):1073–1078, 2020.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Hou, L., Zhu, J., Kwok, J., Gao, F., Qin, T., and Liu, T.-y. Normalization helps training of quantized lstm. *Advances in Neural Information Processing Systems*, 32, 2019.

Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, 2018.

Hripcsak, G. and Rothschild, A. S. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American medical informatics association*, 12(3): 296–298, 2005.

Huang, J., Osorio, C., and Sy, L. W. An empirical evaluation of deep learning for icd-9 code assignment using mimic-iii clinical notes. *Computer methods and programs in biomedicine*, 177:141–153, 2019a.

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

Huang, L., Sun, C., Qiu, X., and Huang, X.-J. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3509–3514, 2019c.

Huang, Y., Li, Y., Heyes, T., Jourjon, G., Cheng, A., Seneviratne, S., Thilakarathna, K., Webb, D., and Da Xu, R. Y. Task adaptive siamese neural networks for open-set recognition of encrypted network traffic with bidirectional dropout. *Pattern Recognition Letters*, 2022.

Iqbal, E., Mallah, R., Rhodes, D., Wu, H., Romero, A., Chang, N., Dzahini, O., Pandey, C., Broadbent, M., Stewart, R., et al. Adept, a semantically-enriched pipeline for extracting adverse drug events from free-text electronic health records. *PloS one*, 12 (11):e0187121, 2017.

Issa, T., Uminsky, D., Shaw, A., Makipour, R., and Filice, R. W. Toward automatic mammography auditing via universal language model fine tuning. In *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 215–222, 2021. doi: 10.1109/IRI51335.2021.00035.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pp. 1681–1691, 2015.

Jacquemin, C. *Spotting and discovering terms through natural language processing*. MIT press, 2001.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Jagannatha, A. N. and Yu, H. Bidirectional rnn for medical event detection in electronic health records. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2016, pp. 473. NIH Public Access, 2016.

Jaitly, N. and Hinton, G. E. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, 2013.

Jensen, K., Soguero-Ruiz, C., Oyvind Mikalsen, K., Lindsetmo, R.-O., Kouskoumvekaki, I., Girolami, M., Olav Skrovseth, S., and Augestad, K. M. Analysis of free text in electronic health records for identification of cancer patient trajectories. *Scientific reports*, 7(1):1–12, 2017.

Ji, Z., Wei, Q., and Xu, H. Bert-based ranking for biomedical entity normalization. *AMIA Summits on Translational Science Proceedings*, 2020:269, 2020.

Jimeno-Yepes, A. J., McInnes, B. T., and Aronson, A. R. Exploiting MeSH indexing in MEDLINE to generate a data set for word sense disambiguation. *BMC bioinformatics*, 12(1):223, 2011.

Jin, Q., Dhingra, B., Cohen, W., and Lu, X. Probing biomedical embeddings from language models. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pp. 82–89, 2019a.

Jin, Q., Liu, J., and Lu, X. Deep contextualized biomedical abbreviation expansion. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pp. 88–96, 2019b.

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, 2020.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Joopudi, V., Dandala, B., and Devarakonda, M. A convolutional route to abbreviation disambiguation in clinical text. *Journal of biomedical informatics*, 86:71–78, 2018.

Kahle, P., Colutto, S., Hackl, G., and Mühlberger, G. Transkribus-a service platform for transcription, recognition and retrieval of historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 4, pp. 19–24. IEEE, 2017.

Kelly, L., Suominen, H., Goeuriot, L., Neves, M., Kanoulas, E., Li, D., Azzopardi, L., Spijker, R., Zuccon, G., Scells, H., et al. Overview of the clef ehealth evaluation lab 2019. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings 10*, pp. 322–339. Springer, 2019.

Kholghi, M., De Vine, L., Sitbon, L., Zuccon, G., and Nguyen, A. The benefits of word embeddings features for active learning in clinical information extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pp. 25–34, 2016.

Kim, H.-Y., Roh, Y.-H., and Kim, Y.-G. Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 97–102, 2019.

Kim, Y. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: $10.3115/\mathrm{v1/D14\text{-}1181}$. URL `https://aclanthology.org/D14-1181`.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. Character-aware neural language models. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

King, G. and Zeng, L. Logistic regression in rare events data. *Political analysis*, 9(2): 137–163, 2001.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Klovig Skelton, S. Electronic patient records key to nhs digital transformation. *Computer Weekly*, 2022. URL `https://www.computerweekly.com/news/252514807/Electronic-Patient-Records-key-to-NHS-digital-transformation`.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Kobayashi, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 452–457, 2018.

Koch, G., Zemel, R., Salakhutdinov, R., et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, pp. 0. Lille, 2015.

Kolomiyets, O., Bethard, S., and Moens, M.-F. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 271–276. Association for Computational Linguistics, 2011.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al. Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, 30 (1):25–36, 2006.

Krauthammer, M. and Nenadic, G. Term identification in the biomedical literature. *Journal of biomedical informatics*, 37(6):512–526, 2004.

Kreuzthaler, M., Oleynik, M., Avian, A., and Schulz, S. Unsupervised abbreviation detection in clinical narratives. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pp. 91–98, 2016.

Krishnan, G. S. Evaluating the quality of word representation models for unstructured clinical text based icu mortality prediction. In *Proceedings of the 20th International Conference on Distributed Computing and Networking*, pp. 480–485, 2019.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pp. 1378–1387, 2016.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. From word embeddings to document distances. In *International Conference on Machine Learning*, pp. 957–966, 2015.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019.

Laszlo, S. and Federmeier, K. D. The acronym superiority effect. *Psychonomic Bulletin & Review*, 14(6):1158–1163, 2007.

Leaman, R., Khare, R., and Lu, Z. Challenges in clinical natural language processing for automated disorder normalization. *Journal of biomedical informatics*, 57:28–37, 2015.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lee, H., Pham, P., Largman, Y., and Ng, A. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in neural information processing systems*, 22, 2009.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Lerner, E. B., Jehle, D. V., Janicke, D. M., and Moscati, R. M. Medical communication: do our patients understand? *The American journal of emergency medicine*, 18(7): 764–766, 2000.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Levy, O. and Goldberg, Y. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pp. 171–180, 2014.

Li, C., Ji, L., and Yan, J. Acronym disambiguation using word embedding. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Li, J., Zhou, Y., Jiang, X., Natarajan, K., Pakhomov, S. V., Liu, H., and Xu, H. Are synthetic clinical notes useful for real natural language processing tasks: A case study on clinical entity recognition. *Journal of the American Medical Informatics Association*, 28(10):2193–2201, 2021.

Li, M., Fei, Z., Zeng, M., Wu, F.-X., Li, Y., Pan, Y., and Wang, J. Automated icd-9 coding via a deep learning approach. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(4):1193–1202, 2018.

Lin, C., Bethard, S., Dligach, D., Sadeque, F., Savova, G., and Miller, T. A. Does bert need domain adaptation for clinical negation detection? *Journal of the American Medical Informatics Association*, 27(4):584–591, 2020.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ling, H. and Okada, K. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE transactions on pattern analysis and machine intelligence*, 29(5):840–853, 2007.

Liu, H. and Friedman, C. Mining terminological knowledge in large biomedical corpora. In *Biocomputing 2003*, pp. 415–426. World Scientific, 2002.

Liu, H., Lussier, Y. A., and Friedman, C. A study of abbreviations in the UMLS. In *Proceedings of the AMIA Symposium*, pp. 393. American Medical Informatics Association, 2001.

Liu, H., Aronson, A. R., and Friedman, C. A study of abbreviations in MEDLINE abstracts. In *Proceedings of the AMIA Symposium*, pp. 464. American Medical Informatics Association, 2002.

Liu, J., Zhao, S., and Zhang, X. An ensemble method for extracting adverse drug events from social media. *Artificial intelligence in medicine*, 70:62–76, 2016.

Liu, X., Hersch, G. L., Khalil, I., and Devarakonda, M. Clinical trial information extraction with bert. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pp. 505–506. IEEE, 2021.

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

Liu, Z., Yang, M., Wang, X., Chen, Q., Tang, B., Wang, Z., and Xu, H. Entity recognition from clinical texts via recurrent neural network. *BMC medical informatics and decision making*, 17(2):53–61, 2017.

Logé, C., Ross, E., Dadey, D. Y. A., Jain, S., Saporta, A., Ng, A. Y., and Rajpurkar, P. Q-pain: A question answering dataset to measure social bias in pain management. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Longpre, S., Wang, Y., and DuBois, C. How effective is task-agnostic data augmentation for pretrained transformers? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 4401–4411, 2020.

Lu, M., Fang, Y., Yan, F., and Li, M. Incorporating domain knowledge into natural language inference on clinical texts. *IEEE Access*, 7:57623–57632, 2019.

Lu, W., Ma, L., Chen, H., Jiang, X., and Gong, M. A clinical prediction model in health time series data based on long short-term memory network optimized by fruit fly optimization algorithm. *IEEE Access*, 8:136014–136023, 2020.

Luo, Y. Recurrent neural networks for classifying relations in clinical notes. *Journal of biomedical informatics*, 72:85–95, 2017.

Luo, Y., Thompson, W. K., Herr, T. M., Zeng, Z., Berendsen, M. A., Jonnalagadda, S. R., Carson, M. B., and Starren, J. Natural language processing for ehr-based pharmacovigilance: a structured review. *Drug safety*, 40(11):1075–1089, 2017.

Luque, F. M. and Pérez, J. M. Atalaya at tass 2018: Sentiment analysis with tweet embeddings and data augmentation. In *TASS@ SEPLN*, pp. 29–35, 2018.

Mallinson, J., Sennrich, R., and Lapata, M. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 881–893, 2017.

Marivate, V. and Sefara, T. Improving short text classification through global augmentation methods. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 385–399. Springer, 2020.

McCray, A. T., Srinivasan, S., and Browne, A. C. Lexical methods for managing variation in biomedical terminologies. In *proceedings of the annual symposium on computer application in medical care*, pp. 235. American Medical Informatics Association, 1994.

MEDLINE. MEDLINE. `https://pubmed.ncbi.nlm.nih.gov/`, 2021.

Mehrabi, S., Krishnan, A., Sohn, S., Roch, A. M., Schmidt, H., Kesterson, J., Beesley, C., Dexter, P., Schmidt, C. M., Liu, H., et al. DEEPEN: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, 54:213–219, 2015.

Metzger, M.-H., Durand, T., Lallich, S., Salamon, R., and Castets, P. The use of regional platforms for managing electronic health records for the production of regional public health indicators in france. *BMC Medical Informatics and Decision Making*, 12(1):1–14, 2012.

Meystre, S. M., Savova, G. K., Kipper-Schuler, K. C., and Hurdle, J. F. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of medical informatics*, 17(01):128–144, 2008.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y. (eds.), *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL `http://arxiv.org/abs/1301.3781`.

Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N13-1090`.

Miller, G. A. *WordNet: An electronic lexical database*. MIT press, 1998.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40, 2021.

Mohamed, A.-r., Dahl, G. E., and Hinton, G. Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, 20(1):14–22, 2011.

Monge, G. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie royale des sciences de Paris*, 1781.

Moon, S., Pakhomov, S., Liu, N., Ryan, J. O., and Melton, G. B. 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, 2014.

Moon, S., McInnes, B., and Melton, G. B. Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain. *Healthcare informatics research*, 21(1):35–42, 2015.

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. Normalizing acronyms and abbreviations to aid patient understanding of clinical texts: ShARe/CLEF eHealth Challenge 2013, task 2. *Journal of biomedical semantics*, 7(1):43, 2016.

Mueller, J. and Thyagarajan, A. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Muneeb, T., Sahu, S., and Anand, A. Evaluating distributed word representations for capturing semantics of biomedical concepts. In *Proceedings of BioNLP 15*, pp. 158–163, 2015.

Murdoch, T. B. and Detsky, A. S. The inevitable application of big data to health care. *Jama*, 309(13):1351–1352, 2013.

Navigli, R. and Velardi, P. An analysis of ontology-based query expansion strategies. In *International Workshop & Tutorial on Adaptive Text Extraction and Mining held in conjunction with the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of*, pp. 42, 2003.

Neculoiu, P., Versteegh, M., and Rotaru, M. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 148–157, 2016.

Negi, K., Pavuri, A., Patel, L., and Jain, C. A novel method for drug-adverse event extraction using machine learning. *Informatics in Medicine Unlocked*, 17:100190, 2019.

Neves, M. and Leser, U. A survey on annotation tools for the biomedical literature. *Briefings in bioinformatics*, 15(2):327–340, 2014.

Nguyen, P., Tran, T., Wickramasinghe, N., and Venkatesh, S. Deepr: a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1): 22–30, 2016.

Nikfarjam, A. and Gonzalez, G. H. Pattern mining for extraction of mentions of adverse drug reactions from user comments. In *AMIA annual symposium proceedings*, volume 2011, pp. 1019. American Medical Informatics Association, 2011.

Nikfarjam, A., Sarker, A., O'connor, K., Ginn, R., and Gonzalez, G. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681, 2015.

Okazaki, N. and Ananiadou, S. A term recognition approach to acronym recognition. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 643–650, 2006.

Ong, C. J., Orfanoudaki, A., Zhang, R., Caprasse, F. P. M., Hutch, M., Ma, L., Fard, D., Balogun, O., Miller, M. I., Minnig, M., et al. Machine learning and natural language processing methods to identify ischemic stroke, acuity and location from radiology reports. *PloS one*, 15(6):e0234908, 2020.

Pakhomov, S., Pedersen, T., and Chute, C. G. Abbreviation and acronym disambiguation in clinical discourse. In *AMIA Annual Symposium Proceedings*, volume 2005, pp. 589. American Medical Informatics Association, 2005.

Pan, C., Song, B., Wang, S., and Luo, Z. BERT-based acronym disambiguation with multiple training strategies. 2021.

Pandey, C., Ibrahim, Z., Wu, H., Iqbal, E., and Dobson, R. Improving rnn with attention and embedding for adverse drug reactions. In *Proceedings of the 2017 International Conference on Digital Health*, pp. 67–71, 2017.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. PMLR, 2013.

Patel, K., Patel, D., Golakiya, M., Bhattacharyya, P., and Birari, N. Adapting pre-trained word embeddings for use in medical coding. In *BioNLP 2017*, pp. 302–306, 2017.

Patterson, O. and Hurdle, J. F. Document clustering of clinical narratives: a systematic study of clinical sublanguages. In *AMIA Annual Symposium Proceedings*, volume 2011, pp. 1099. American Medical Informatics Association, 2011.

Pawar, S., Palshikar, G. K., Bhattacharyya, P., Ramrakhiyani, N., Gupta, S., and Varma, V. Tcs research at tac 2017: Joint extraction of entities and relations from drug labels using an ensemble of neural networks. In *TAC*, 2017.

Pele, O. and Werman, M. Fast and robust earth mover's distances. In *12th International Conference on Computer Vision*, pp. 460–467. IEEE, 2009.

Peng, Y., Yan, S., and Lu, Z. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pp. 58–65, 2019.

Peng, Y., Lee, S., Elton, D. C., Shen, T., Tang, Y.-x., Chen, Q., Wang, S., Zhu, Y., Summers, R., and Lu, Z. Automatic recognition of abdominal lymph nodes from clinical text. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pp. 101–110, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.clinicalnlp-1.12. URL https://aclanthology.org/2020.clinicalnlp-1.12.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Percha, B. Modern clinical text mining: A guide and review. *Annual review of biomedical data science*, 4:165–187, 2021.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Pham, N.-Q., Kruszewski, G., and Boleda, G. Convolutional neural network language models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1153–1162, 2016.

Pinkus, A. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

Prokofyev, R., Demartini, G., Boyarsky, A., Ruchayskiy, O., and Cudré-Mauroux, P. Ontology-based word sense disambiguation for scientific literature. In *European conference on information retrieval*, pp. 594–605. Springer, 2013.

PubMed. Pubmed. `https://pubmed.ncbi.nlm.nih.gov/`, 2021.

Pustejovsky, J., Castano, J., Cochran, B., Kotecki, M., and Morrell, M. Automatic extraction of acronym-meaning pairs from MEDLINE databases. In *MEDINFO 2001*, pp. 371–375. IOS Press, 2001.

Qing, L., Linhong, W., and Xuehai, D. A novel neural network-based method for medical text classification. *Future Internet*, 11(12):255, 2019.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rahman, M. M. and Davis, D. N. Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2):224, 2013.

Raj, D., Sahu, S., and Anand, A. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)*, pp. 311–321, 2017.

Rannikmäe, K., Ngoh, K., Bush, K., Salman, R. A.-S., Doubal, F., Flaig, R., Henshall, D. E., Hutchison, A., Nolan, J., Osborne, S., et al. Accuracy of identifying incident stroke cases from linked health care data in uk biobank. *Neurology*, 95(6):e697–e707, 2020.

Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J., and Ré, C. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pp. 3236–3246, 2017.

Reddy, S., Chen, D., and Manning, C. D. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

Rios, A. and Kavuluru, R. Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In *Proceedings of the 6th ACM*

Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 258–267, 2015.

Roberts, K., Demner-Fushman, D., and Tonning, J. M. Overview of the tac 2017 adverse reaction extraction from drug labels track. In *TAC*, 2017.

Romanov, A. and Shivade, C. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1586–1596, 2018.

Rubner, Y., Tomasi, C., and Guibas, L. J. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision*, pp. 59–66. IEEE, 1998.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Sahu, S., Anand, A., Oruganty, K., and Gattu, M. Relation extraction from clinical texts using domain invariant convolutional neural network. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pp. 206–215, 2016.

Sarker, A. and Gonzalez, G. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53: 196–207, 2015.

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

Schwartz, A. S. and Hearst, M. A. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, pp. 451–462. World Scientific, 2002.

Sennrich, R., Haddow, B., and Birch, A. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, 2016.

Shao, Y., Divita, G., Workman, T. E., Redd, D., Garvin, J. H., and Zeng-Treitler, Q. Clinical sublanguage trend and usage analysis from a large clinical corpus. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 3837–3845. IEEE, 2020.

Sheikhshab, G., Birol, I., and Sarkar, A. In-domain context-aware token embeddings improve biomedical named entity recognition. In *Proceedings of the ninth international workshop on health text mining and information analysis*, pp. 160–164, 2018.

Shin, B., Chokshi, F. H., Lee, T., and Choi, J. D. Classification of radiology reports using neural attention models. In *2017 international joint conference on neural networks (IJCNN)*, pp. 4363–4370. IEEE, 2017.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. Text data augmentation for deep learning. *Journal of Big Data*, 8(1):1–34, 2021.

Si, Y., Wang, J., Xu, H., and Roberts, K. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26 (11):1297–1304, 2019.

Skentzos, S., Shubina, M., Plutzky, J., and Turchin, A. Structured vs. unstructured: factors affecting adverse drug reaction documentation in an emr repository. In *AMIA annual symposium proceedings*, volume 2011, pp. 1270. American Medical Informatics Association, 2011.

Skreta, M., Arbabi, A., Wang, J., and Brudno, M. Training without training data: Improving the generalizability of automated medical abbreviation disambiguation. In *Machine Learning for Health Workshop*, pp. 233–245. PMLR, 2020.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Sohn, S., Comeau, D. C., Kim, W., and Wilbur, W. J. Abbreviation definition identification based on automatic precision estimates. *BMC bioinformatics*, 9(1):1–10, 2008.

Spasić, I. Acronyms as an integral part of multi-word term recognition–a token of appreciation. *IEEE Access*, 6:8351–8363, 2018.

Spasić, I. Flexiterm: a more efficient implementation of flexible multi-word term recognition. *arXiv preprint arXiv:2110.06981*, 2021.

Spasić, I., Sarafraz, F., Keane, J. A., and Nenadić, G. Medication information extraction with linguistic pattern matching and semantic rules. *Journal of the American Medical Informatics Association*, 17(5):532–535, 2010.

Spasić, I., Greenwood, M., Preece, A., Francis, N., and Elwyn, G. Flexiterm: a flexible term recognition method. *Journal of biomedical semantics*, 4(1):1–15, 2013.

Spasić, I., Zhao, B., Jones, C. B., and Button, K. Kneetex: an ontology–driven system for information extraction from mri reports. *Journal of biomedical semantics*, 6(1): 1–26, 2015.

Spasić, I., Krzeminski, D., Corcoran, P., Balinsky, A., et al. Cohort selection for clinical trials from longitudinal patient records: text mining approach. *JMIR Medical Informatics*, 7(4):e15980, 2019.

Spasić, I., Nenadić, G., et al. Clinical text data in machine learning: systematic review. *JMIR medical informatics*, 8(3):e17984, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Stedman, T. L. *Stedman's medical dictionary for the health professions and nursing*. Lippincott Williams & Wilkins, 2005.

Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974.

Stone, M. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 44–47, 1977.

Studdert-Kennedy, M. How did language go discrete. *Language origins: Perspectives on evolution, ed. M. Tallerman*, pp. 48–67, 2005.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Suominen, H., Kelly, L., Goeuriot, L., Névéol, A., Ramadier, L., Robert, A., Kanoulas, E., Spijker, R., Azzopardi, L., Li, D., et al. Overview of the clef ehealth evaluation lab 2018. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction:*

9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings 9, pp. 286–301. Springer, 2018.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Sykes, D., Grivas, A., Grover, C., Tobin, R., Sudlow, C., Whiteley, W., McIntosh, A., Whalley, H., and Alex, B. Comparison of rule-based and neural network models for negation detection in radiology reports. *Natural Language Engineering*, 27(2): 203–224, 2021.

Tao, C., Lee, K., Filannino, M., Buchan, K., Lee, K., Arora, T. R., Liu, J., Farri, O., and Uzuner, Ö. Extracting and normalizing adverse drug reactions from drug labels. In *TAC*, 2017.

Terada, A., Tokunaga, T., and Tanaka, H. Automatic expansion of abbreviations by using context and character information. *Information processing & management*, 40 (1):31–45, 2004.

Tieleman, T., Hinton, G., et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Tomanek, K. and Hahn, U. Timed annotations—enhancing muc7 metadata by the time it takes to annotate named entities. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pp. 112–115, 2009.

UMLS. UMLS. `https://www.nlm.nih.gov/research/umls/`, 2021.

US Food and Drug Administration. What is a serious adverse event? `https://www.fda.gov/safety/reporting-serious-problems-fda/what-serious-adverse-event`, 2016.

US Food and Drug Administration. Data mining at fda - white paper. `https://www.fda.gov/science-research/data-mining/data-mining-fda-white-paper`, 2018. Accessed: 2021-12-11.

Uzuner, O., Szolovits, P., and Kohane, I. i2b2 workshop on natural language processing challenges for clinical records. In *Proceedings of the Fall Symposium of the American Medical Informatics Association*. Citeseer, 2006.

Uzuner, Ö., Solti, I., and Cadag, E. Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, 17(5):514–518, 2010.

Van, H., Kauchak, D., and Leroy, G. Automets: The autocomplete for medical text simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1424–1434, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veyseh, A. P. B., Dernoncourt, F., Nguyen, T. H., Chang, W., and Celi, L. A. Acronym identification and disambiguation shared tasks for scientific document understanding. In *SDU@AAAI*, 2021.

Vijayaraghavan, P., Sysoev, I., Vosoughi, S., and Roy, D. DeepStance at SemEval-2016 task 6: Detecting stance in tweets using character and word-level CNNs. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 413–419, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1067. URL https://www.aclweb.org/anthology/S16-1067.

Vosoughi, S., Vijayaraghavan, P., and Roy, D. Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 1041–1044. ACM, 2016.

Vrbančič, G. and Podgorelec, V. Transfer learning with adaptive fine-tuning. *IEEE Access*, 8:196197–196211, 2020.

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

Wang, C.-S., Lin, P.-J., Cheng, C.-L., Tai, S.-H., Yang, Y.-H. K., Chiang, J.-H., et al. Detecting potential adverse drug reactions using a deep neural network model. *Journal of medical Internet research*, 21(2):e11016, 2019a.

Wang, W. Y. and Yang, D. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2557–2563, 2015.

Wang, X., Hripcsak, G., Markatou, M., and Friedman, C. Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records:

a feasibility study. *Journal of the American Medical Informatics Association*, 16(3): 328–337, 2009.

Wang, X., Pham, H., Dai, Z., and Neubig, G. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 856–861, 2018b.

Wang, Y., Rastegar-Mojarad, M., Komandur-Elayavilli, R., and Liu, H. Leveraging word embeddings and medical entity extraction for biomedical dataset retrieval using unstructured texts. *Database*, 2017, 2017.

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

Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., et al. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49, 2018d.

Wang, Y., Sohn, S., Liu, S., Shen, F., Wang, L., Atkinson, E. J., Amin, S., and Liu, H. A clinical text classification paradigm using weak supervision and deep representation. *BMC medical informatics and decision making*, 19(1):1–13, 2019b.

Wei, J. and Zou, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6383–6389, 2019.

Wen, Z., Lu, X. H., and Reddy, S. Medal: Medical abbreviation disambiguation dataset for natural language understanding pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pp. 130–135, 2020.

Wieting, J., Mallinson, J., and Gimpel, K. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 274–285, 2017.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Wolsey, L. A. and Nemhauser, G. L. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

Wong, A., Plasek, J. M., Montecalvo, S. P., and Zhou, L. Natural language processing and its implications for the future of medication safety: a narrative review of recent advances and challenges. *Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy*, 38(8):822–841, 2018.

Wren, J. D. and Garner, H. R. 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, 2002.

Wu, S., Miller, T., Masanz, J., Coarr, M., Halgrim, S., Carrell, D., and Clark, C. Negation's not solved: generalizability versus optimizability in clinical natural language processing. *PloS one*, 9(11):e112774, 2014.

Wu, X., Lv, S., Zang, L., Han, J., and Hu, S. Conditional BERT contextual augmentation. In *International Conference on Computational Science*, pp. 84–95. Springer, 2019.

Wu, Y., Denny, J., Rosenbloom, S., Miller, R., Giuse, D., Song, M., and Xu, H. A preliminary study of clinical abbreviation disambiguation in real time. *Applied clinical informatics*, 6(02):364–374, 2015a.

Wu, Y., Xu, J., Jiang, M., Zhang, Y., and Xu, H. A study of neural word embeddings for named entity recognition in clinical text. In *AMIA annual symposium proceedings*, volume 2015, pp. 1326. American Medical Informatics Association, 2015b.

Wu, Y., Denny, J. C., Trent Rosenbloom, S., Miller, R. A., Giuse, D. A., Wang, L., Blanquicett, C., Soysal, E., Xu, J., and Xu, H. 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, 2016a.

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

Wu, Y., Jiang, M., Xu, J., Zhi, D., and Xu, H. Clinical named entity recognition using deep learning models. In *AMIA Annual Symposium Proceedings*, volume 2017, pp. 1812. American Medical Informatics Association, 2017.

Wyse, L. Audio spectrogram representations for processing with convolutional neural networks. In *Proceedings of the First International Conference on Deep Learning and Music*, pp. 37–41, 2017.

Xie, Z., Wang, S. I., Li, J., Lévy, D., Nie, A., Jurafsky, D., and Ng, A. Y. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL `https://openreview.net/forum?id=H1VyHY9gg`.

Xu, H., Stetson, P. D., and Friedman, C. A study of abbreviations in clinical notes. In *AMIA annual symposium proceedings*, volume 2007, pp. 821. American Medical Informatics Association, 2007.

Xu, J., Zhang, Y., Xu, H., et al. Clinical abbreviation disambiguation using neural word embeddings. *Proceedings of BioNLP 15*, pp. 171–176, 2015.

Xu, J., Lee, H.-J., Ji, Z., Wang, J., Wei, Q., and Xu, H. Uth_ccb system for adverse drug reaction extraction from drug labels at tac-adr 2017. In *TAC*, 2017.

Yadav, V. and Bethard, S. A survey on recent advances in named entity recognition from deep learning models. In *27th International Conference on Computational Linguistics, COLING 2018*, pp. 2145–2158. Association for Computational Linguistics (ACL), 2018.

Yang, X., Lyu, T., Li, Q., Lee, C.-Y., Bian, J., Hogan, W. R., and Wu, Y. A study of deep learning methods for de-identification of clinical notes in cross-institute settings. *BMC medical informatics and decision making*, 19(5):1–9, 2019.

Yang, X., Bian, J., Hogan, W. R., and Wu, Y. Clinical concept extraction using transformers. *Journal of the American Medical Informatics Association*, 27(12): 1935–1942, 10 2020a. ISSN 1527-974X. doi: 10.1093/jamia/ocaa189. URL `https://doi.org/10.1093/jamia/ocaa189`.

Yang, X., He, X., Zhang, H., Ma, Y., Bian, J., Wu, Y., et al. Measurement of semantic textual similarity in clinical texts: comparison of transformer-based models. *JMIR medical informatics*, 8(11):e19735, 2020b.

Yap, B. W., Rani, K. A., Rahman, H. A. A., Fong, S., Khairudin, Z., and Abdullah, N. N. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, pp. 13–22. Springer, 2014.

Yu, A. W., Dohan, D., Le, Q., Luong, T., Zhao, R., and Chen, K. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018a. URL `https://openreview.net/forum?id=B14TlG-RW`.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V. Qanet: Combining local convolution with global self-attention for reading comprehension. In *International Conference on Learning Representations*, 2018b.

Yu, H., Hripcsak, G., and Friedman, C. Mapping abbreviations to full forms in biomedical articles. *Journal of the American Medical Informatics Association*, 9(3):262–272, 2002.

Yu, H., Kim, W., Hatzivassiloglou, V., and Wilbur, W. J. Using MEDLINE as a knowledge source for disambiguating abbreviations and acronyms in full-text biomedical journal articles. *Journal of biomedical informatics*, 40(2):150–159, 2007.

Yu, Z., Tsuruoka, Y., and Tsujii, J. Automatic resolution of ambiguous abbreviations in biomedical texts using support vector machines and one sense per discourse hypothesis. In *Proceedings of the SIGIR*, volume 3, pp. 57–62. Citeseer, 2003.

Zavala, R. R., Martinez, P., et al. The impact of pretrained language models on negation and speculation detection in cross-lingual medical text: comparative study. *JMIR Medical Informatics*, 8(12):e18953, 2020.

Zhang, D. and Yang, Z. Word embedding perturbation for sentence classification. *arXiv preprint arXiv:1804.08166*, 2018.

Zhang, D., Luo, T., and Wang, D. Learning from LDA using deep neural networks. In *Natural Language Understanding and Intelligent Applications*, pp. 657–664. Springer, 2016.

Zhang, J., Lertvittayakumjorn, P., and Guo, Y. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1031–1040, 2019.

Zhang, L., Wang, S., and Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

Zhang, T., Wu, F., Katiyar, A., Weinberger, K. Q., and Artzi, Y. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*, 2020a.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657, 2015.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, W. B. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 270–278, 2020b.

Zheng, J. G., Howsmon, D., Zhang, B., Hahn, J., McGuinness, D., Hendler, J., and Ji, H. Entity linking for biomedical literature. *BMC medical informatics and decision making*, 15(1):1–9, 2015.

Zhu, H., Paschalidis, I. C., and Tahmasebi, A. M. Clinical concept extraction with contextual word embedding. In *NIPS Machine Learning for Health Workshop*, 2018.