Natural Language Processing and Text Mining (Turning Unstructured Data into Structured)

Ayoub Bagheri, Anastasia Giachanou, Pablo Mosteiro and Suzan Verberne

Abstract

The integration of natural language processing (NLP) and text mining techniques has emerged as a key approach to harnessing the potential of unstructured clinical text data. This chapter discusses the challenges posed by clinical narratives and explores the need to transform them into structured formats for improved data accessibility and analysis. The chapter navigates through key concepts, including text pre-processing, text classification, text clustering, topic modeling, and advances in language models and transformers. It highlights the dynamic interplay between these techniques and their applications in tasks ranging from disease classification to extraction of side effects. In addition, the chapter acknowledges the importance of addressing bias and ensuring model explainability in the context of clinical prediction systems. By providing a comprehensive overview, the chapter offers insights into the synergy of NLP and text mining techniques in shaping the future of biomedical AI, ultimately leading to safer, more efficient, and more informed healthcare decisions.

Keywords

Natural language processing · Text mining · Clinical text · Text pre-processing · Language models · Text classification · Text clustering · Topic modeling · Explainability · Bias detection · Clinical NLP

1 Introduction

The field of biomedical artificial intelligence (AI) is undergoing a revolution. The widespread use of biomedical data sources next to electronic health records (EHR) systems provides a large amount of data in healthcare, leading to new areas for clinical research. These resources are rich in data with the potential to leverage applications that provide safer care, reduce medical errors, reduce healthcare expenditure, and enable providers to improve their productivity, quality and efficiency [1, 2]. A major portion of this data is inside free text in the form of physicians' notes, discharge summaries, and radiology reports among many other types of clinical narratives such as patient experiences. This clinical text follows the patient through the care procedures and documents the patient's complaints and symptoms, physical exam, diagnostic

A. Bagheri (✉) · A. Giachanou · P. Mosteiro
University Utrecht, Utrecht, Netherlands
e-mail: a.bagheri@uu.nl

S. Verberne
Leiden University, Leiden, Netherlands

F. W. Asselbergs et al. (eds.), *Clinical Applications of Artificial Intelligence in Real-World Data*,
https://doi.org/10.1007/978-3-031-36678-9_5

tests, conclusions, treatments, and outcomes of the treatment.

Free text in the clinical domain is unstructured information, which is difficult to process automatically. Despite many attempts to encode text in the form of structured data [3], free text continues to be used in EHRs. Additionally, clinical texts are packed with substantial amounts of abbreviations, special characters, stop words, and spelling errors. Therefore, natural language processing (NLP) and text mining techniques can be applied to create a more structured representation of a text, making its content more accessible for data science, machine learning and statistics, and for medical prediction models.

A widely accepted definition of text mining has been provided by Hearst [4], as "the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources". Text mining is about looking for patterns in text, in a similar way that data mining can be loosely described as looking for patterns in data. According to [5], NLP is one of the most widely used big data analytical techniques in healthcare, and is defined as "any computer-based algorithm that handles, augments, and transforms natural language so that it can be represented for computation" [6]. There is therefore often an overlap of the tasks, methods, and goals for text mining and NLP, and the concepts are sometimes used interchangeably. Fleuren and Alkema [7] describe clinical text mining as automated processing and analysis of text in relevant textual biources. Text mining typically involves a number of distinct phases including information retrieval, named entity recognition, information extraction and knowledge discovery. The first step concerns collecting and filtering relevant documents. After information retrieval, the resulting document collection can be analyzed by classification or clustering algorithms. As a last step, information extraction is performed to generate structured data from unstructured text.

Text mining and NLP techniques have been applied to numerous health applications involving text de-identification tools [8], clinical decision support systems [2], patient identification [9–12], disease classification [13–15], disease history [16], ICD10 classification [17], hospital readmission prediction [18], and chronic disease prediction [19].

Although those systems can now achieve high performance in various clinical prediction tasks, they come with some limitations. A common issue is related to whether there is any bias introduced in any step involved in learning process. This is important because we know that systems are trained on data which contain societal stereotypes, and can therefore learn to reproduce them in their predictions. Another limitation is that clinicians are reluctant to widely use those systems because, among other reasons, they do not understand the complicated processes on which the predictions are made. Those limitations have led to the necessity of systems that can produce explanations regarding their learning mechanism and decisions.

The successive sections of this chapter are organised as follows: Sect. 2 provides a gentle introduction on NLP and the common techniques when conducting biomedical and clinical text analysis. Subsequently, we discuss state-of-the-art pre-trained language models in Sect. 3, and NLP tasks and their challenges in healthcare in Sect. 4. Finally, we overview bias and explainability of NLP-based models for biomedical and clinical text in Sects. 5 and 6, respectively. We conclude the chapter with a summary and recommendations.

2 What Is Natural Language Processing

Natural language processing is an area of artificial intelligence concerned with the interactions between computers and human languages. There are many applications of NLP in specific domains, such as machine translation of legal documents, mental disease detection, news summarization, patent information retrieval, and so on.

2.1 Text Preprocessing

With the advancements of NLP, it is possible to develop methodologies and automate different natural language tasks. NLP tasks can be divided in document-level tasks (Sects. 2.2 and 2.3), sequence labelling tasks (Sect. 2.4), and sequence-to-sequence processing (not discussed in this chapter). There are two types of document-level tasks: *text classification* and *text clustering*. The former refers to tasks of adding labels from a pre-defined label set to a text. In other words, we are interested in classifying texts into pre-defined categories. Annotating a piece of text as expressing positive or negative sentiment or classifying an EHR regarding the patient's risk of disease are two text classification examples. Text clustering refers to automatically group textual documents into clusters based on their content similarity. In this case, there are no pre-defined categories. Topic clustering of textual documents is one example of such a task. In sequence labelling tasks, one label is added to each word in a text, to identify and extract specific relevant information such as named entities. Finally, in sequence-to-sequence tasks, both the input and the output is text, like in translation or summarization.

Text from natural language is often noisy and unstructured and needs to be pre-processed before it can be used in one of these tasks. Pre-processing transforms text into a consistent form that is readable from the machines. The most common steps are sentence segmentation, word tokenization, lowercasing, stemming or lemmatization, stop word removal, and spelling correction.

Here, we should note that an NLP system can involve some or all of those steps. The steps and the techniques that will be used depends on the data, the task and the method used. For example, social media posts contain special characters and emoticons and the NLP researcher can decide how to handle them, whereas domain specific stop words may be necessary when EHRs are analyzed. In addition, for sequence labelling it is important to keep capitalisation, punctuation and word order, while these aspects can be disregarded in text classification or clustering. Below we will briefly describe the most common steps, which are the sentence segmentation, tokenization and stemming/lemmatization.

Segmentation The NLP pipeline usually starts with the sentence segmentation that refers to divide the text into sentences. Although this looks like a trivial task, there are some challenges. For example, in social media texts users tend to use emoticons that are a combination of symbols including a period (.), question mark (?) or exclamation mark (!). Additionally, a period is used in many abbreviations (e.g., Mr.) that makes the sentence segmentation more challenging. Packages such as NLTK and Spacy can perform sentence segmentation for a range of languages.

Tokenization Tokenization is one of the core steps in pre-processing and refers to converting a sentence into tokens. Traditionally, tokens are words, punctuation marks, or numbers, but in some contexts subwords can be used as tokens (see Sect. 3.3). In some tasks, we can also add tokens that capture other type of information such as word order or part-of-speech tags (i.e., information that refers to the type such as noun, verb etc.).

Stemming and Lemmatization Both stemming and lemmatization aim to normalize the tokens that refer to the same base but appear in a different form in the text (e.g., disease and diseases). Stemming is based on a more heuristic process and cuts the ends of the words, whereas lemmatization is based on the morphological analysis of words, and aims to return the base of a word (known as the lemma). For example, stemming of the verb *saw* can result to no changes while lemmatization will return the base form of the word which is *see*.

2.2 Text Classification

Text classification is the task of assigning one or more predefined categories to documents based on their contents. Given a document d and a set of n_C class labels $C_L \in \{1, \ldots, n_C\}$, text classification tries to learn a classification function $f : D \rightarrow C_L$ that maps a set of documents to labels. Text classification can be implemented as

an automated process involving none or a small amount of interaction with expert users [20]. A general pipeline for a text classification system is illustrated in Fig. 1.

In binary text classification each document is assigned to either a specific predefined label or to the complement of that label (e.g. relevant or non-relevant). On the other hand, multi-class classification refers to the situation where each document is assigned a label from a set of n classes (where $n > 2$). Multi-label text classification refers to the case in which a document can be associated with more than one label. Text classification contains four different levels of scope that can be applied: (1) Document level, (2) Paragraph level, (3) Sentence level, and (4) Phrase level.

2.3 Text Clustering and Topic Modeling

With unsupervised learning such as clustering, there are no labeled examples to learn from, instead the goal is to find some structure or patterns in the input data [21]. Text clustering is an example of unsupervised learning, which aims to group texts or words according to some measure of similarity [22]. The goal of clustering is to identify the underlying structure of the observed data, such that there are a few clusters of points, each of which is internally coherent. Clustering algorithms assign each data point to a discrete cluster $c_i \in 1, 2, \ldots, K$.

Broadly speaking, clustering can be divided into subgroups; hard and soft clustering. Hard clustering groups the data in such a way that each item is assigned to one cluster, whereas in soft clustering one item can belong to multiple clusters. Topic modeling is a type of soft clustering [23, 24]. Topic modeling provides a convenient unsupervised way to analyze high-dimensional data such as text. It is a form of text analysis in which a collection is assumed to cover a set of topics; a topic is defined as a probability distribution over all words in the collection (some words being very prominent for the topic and other words not related to the topic) and each document is represented by a probability distribution over all topics (some topics being very prominent in the document, and other topics not covered).

There have been a number of topic modeling algorithms proposed in the literature. The most popular topic model is the Latent Dirichlet Allocation (LDA) that is a powerful generative latent topic model [23]. It applies unsupervised learning on texts to induce sets of associated words. LDA defines every topic as a distribution over the words of the vocabulary, and every document as a distribution over the topics.

LDA specifies a probabilistic procedure by which documents can be generated. Figure 2 shows a text generation process by a topic model. Topic 1 and topic 2 shown in the figure have different word distributions so that they can constitute documents by choosing the words which have different importance degree to the topic. Document 1 and document 3 are generated by the respective random sampling of topic 1 and topic 2. But, topic 1 and topic 2 generate document 2 according to the mixture of their different topic distributions. Here, the numbers at the right side of a word are its belonging topic numbers and, the word is obtained by the random sampling of the numbered topic.

LDA uses a K-dimensional latent random variable which obeys the Dirichlet distribution to represent the topic mixture ratio of the document, which simulates the generation process of the document. Let K be the multinomial topic distributions for the dataset containing V elements each, where V is the number of terms in the dataset. Let β_i represent the multinomial for the i-th topic, where the size of β_i is V. Given these distributions, the LDA generative process is as follows:

Algorithm 1: Generative process in LDA

```
for each document do
    (a) Randomly choose a K-dimensional multinomial distribution
    over topics
    for each word in the document do
        (i) Probabilistically draw β_j from the distribution over topics
        obtained in (a)
        (ii) Probabilistically draw one of the V words from β_j
    end
end
```

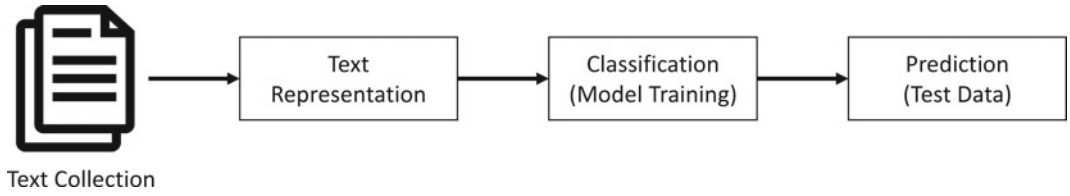

Fig. 1 The general pipeline of a text classification system

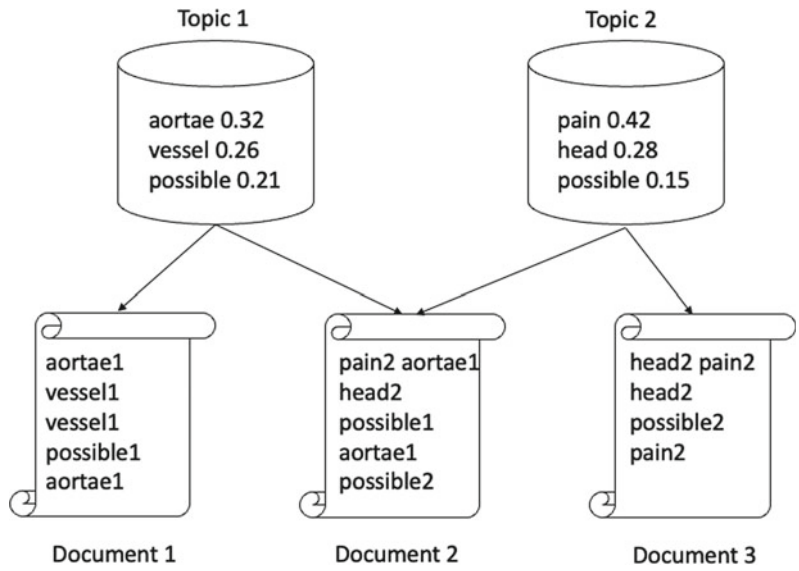

Fig. 2 The generative process of topic modeling

LDA emphasizes that documents contain multiple topics. For instance, a discharge letter might have words drawn from the topic related to the patient's symptoms and words drawn from the topic related to the patient's treatment. LDA uses sampling from the Dirichlet distribution to generate a text with the specific topic multinomial distribution, where the text is usually composed of some latent topics. And then, these topics are sampled repeatedly to generate each word for the document. Thus, the latent topics can be seen as the probability distribution of the words in the LDA model. And, each document is expressed as the random mixture of these latent topics according to the specific proportion.

The goal of LDA is to automatically discover the topics from a collection of documents. Standard statistical techniques can be used to invert the generative process of LDA, thus inferring the set of topics that were responsible for generating a collection of documents. The exact inference in LDA is generally intractable, therefore approximate inference algorithms are needed for posterior estimation. The most common approaches that are used for approximate inference are expectation-maximization, Gibbs sampling and variational method [25].

LDA has been applied in the health domain as well. Duarte et al. [26] applied LDA on a collection of electronic health records and showed that some topics occur more often in the deceased patients, like renal diseases, and others (e.g., diabetes) appear more often in the discharge collection. Li et al. [27] used LDA to cluster patient diagnostics groups from Rochester Epidemiology Projects (REP) that contains medical records. In their study, they identified 20 topics that could almost be connected with some group of diseases. However, they also observed that the same diagnosis code group might fall into different topics. LDA has not only been used to extract topics, but also as an alternative way to represent the documents [28].

LDA is accessible to work with, thanks to the implementation of the model in packages such as gensim.[1] There are a few challenges for the user though: First, the topics are unlabeled so a human has to assign labels to the topics to make them quickly interpretable. Second, LDA is not deterministic; in multiple runs it will give multiple different outputs. Third, the number of topics needs to be determined beforehand, e.g. through optimizing the model for topic coherence [29].

2.4 Information Extraction

As discussed in Sect. 2.2, in text classification tasks, labels are assigned to a text as a whole (a whole document, paragraph, or sentence). In information extraction tasks on the other hand, labels are assigned to each token in the text. The token labels identify tokens as being part of a relevant term, typically an entity such as a name. The task of identifying entities in text is called *Named Entity Recognition*. Machine learning tasks that learn to assign a label to each token are called *sequence labelling* tasks.

In sequence labelling, word order is important, because subsequent words might together form an entity (e.g. 'New York', 'breast cancer'), and words in the context of the entity words can give information about the presence of an entity. Take for example the sentence "Since taking Gleevec, the patient has peripheral edema". Even without ever having seen the word Gleevec, you can deduce from its context that it is a medication name. Apart from word order and context, capitalisation and punctuation are relevant in sequence labelling tasks: names are often capitalised, and punctuation such as bracketing sometimes provides information about the presence of an entity or the relation between two entities. These characteristics set information extraction tasks apart from text classification tasks, despite both being supervised learning tasks.

When creating labelled data for sequence labelling, words and word groups are marked in

[1] https://radimrehurek.com/gensim/.

Table 1 Example of IOB labelling with one medication name and one adverse drug reaction (ADR)

Since	Taking	Gleevec	,	The	Patient	Has	Peripheral	Edema
O	O	B-MED	O	O	O	O	B-ADR	I-ADR

annotation tools such as doccano[2] and inception.[3] These annotations are then converted to a file format with one label per token. The common token labelling scheme for named entity recognition is *IOB labelling*, in which each token gets one of three labels: 'I' if the token is inside an entity; 'O' if it is outside an entity; 'B' if it is the first token of an entity. The B and I labels have a suffix, indicating their type. Table 1 gives an example of IOB labelling for one sentence. Here, B-MED indicates the first word of the medication name, B-ADR the beginning of the adverse drug reaction (ADR), and I-ADR the subsequent word of a the ADR entity.

Based on token-level labelled data, sequence labelling models can be trained that take a vector representation for each token as input and learn the output label. For sequence labelling, we need machine learning models that take the context of tokens into account. The most commonly used feature-based sequence labelling model is Conditional Random Fields (CRF).[4] Since around 2016, CRF was typically used on top of a neural sequence model, Bi-LSTM [30]. LSTMs (Long Short-Term Memory models) are recurrent neural networks. These are neural network models that, instead of classifying each token independently, use the learned representations of the previous words for learning the label of the current token. Bi-LSTM-CRFs were the state of the art for named entity recognition for some years, before they were superseded by transformer-based models (see Sect. 3.1).

In addition to named entity recognition, *relation extraction* is often relevant: we not only want to identify medications and ADRs, but also which ADR is related to which medication. Another prominent relation extraction task in the biomedical domain is the relation between genes, proteins and diseases. Information extraction methods rely on co-occurrence of entities, both for unsupervised or supervised labelling. In supervised labelling, co-occurrence is combined with representations of the entities and their context to decide for a pair of entities whether or not there is a relation between them. An overview of methods is provided by Nasar et al. [31].

2.5 Text Representations

As introduced in Sect. 2.1, the first step of the NLP pipeline is to prepare the raw text into a representation that can be used for further processing. We have introduced classification, clustering and extraction tasks. In this subsection we will explain commonly used text representations: how to represent texts in a form that can be used as input to machine learning models.

2.5.1 Bag-of-Word Models

To perform text classification and after the text pre-processing, the question is how to represent each text document [22, 32]. A document can be seen as an observation in the dataset, e.g. a patient discharge letter in a collection of discharge summaries, or a chest x-ray report. A common approach is to use vector models of a co-occurrence matrix. A co-occurrence matrix is a way of representing how often words co-occur. An example of such co-occurrence matrices is a document-term matrix, in which each row represents a document from the dataset and each matrix column represents a word in the vocabulary of the dataset. Table 2 shows a small selection from a document-term matrix of radiology reports showing the occurrence of seven words in five documents.

[2] https://doccano.github.io/doccano/.

[3] https://inception-project.github.io/.

[4] A tutorial with a description of features for named entity recognition can be found on https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html.

Table 2 Document-term matrix

Document	Abnormalities	Aortae	Possible	Nicotine	Pain	Thoracic
1	1	0	1	0	0	0
2	1	1	1	1	0	0
3	1	0	0	0	1	0
4	1	0	0	0	0	0
5	1	0	0	1	0	1

In Table 2, each document is represented as a vector of word counts. This representation is often called a *bag-of-words*, because it includes only information about the count of each word, and not the order in which the words appear. With the bag-of-words representation, we are ignoring grammar and order of the words. Yet the bag-of-words model is surprisingly effective for text classification [22].

There are three commonly used bag-of-words representations of text data, corresponding to the $binary$, the TF, and the $TFiDF$ model. A binary representation model corresponds to whether or not a word is present in the document. In some applications, such as finding frequently co-occurring groups of k words, it is sufficient to use a binary representation. However, it may lead to the loss of information because it does not contain the frequencies of the words [32].

The most basic form of frequency-based text feature extraction is TF. TF stands for the term frequency. In this method, each word is mapped to its number of occurrences in the text. However, this approach is limited by the fact that particular words (e.g., patient in a health application) that are commonly used in the language may dominate such representations. Most representations of text use normalized frequencies of the words. One approach is the $TFiDF$, where iDF stands for the inverse document frequency. The mathematical representation of the weight of the term t in the document d by TFiDF is given in:

$$TFiDF(d,t) = TF(d,t)log\left(\frac{N}{DF(t)}\right) \quad (1)$$

where $TF(d,t)$ is the frequency of the term t in document d, N is the number of documents and $DF(t)$ is the number of documents containing the term t. Although TFiDF tries to overcome the problem of common words in the document, it still suffers from the fact that it cannot account for the order of the words and the similarity between them in the document since each word is independently presented. Another issue with TFiDF is that even though it removes common words, it might decrease the performance by increasing the frequencies of misspellings that were not properly handled at the pre-processing step [20, 22].

2.5.2 Word Embeddings

There is a quote by Firth [33], denoting that "words occurring in similar contexts tend to have similar meanings". It outlines the idea in NLP that a statistical approach, that considers how words and phrases are used in text documents, might replicate the human notions of semantic similarity. This idea is known as the distributional hypothesis.

Word embeddings are dense vector representations of words. The embeddings vector space has much lower dimensionality than the sparse bag-of-words vector space (100–400 as opposed to tens of thousands). In the embeddings space, words that are more similar (semantically and syntactically) are closer to each other than non-similar words. In other words, embeddings are a distributional semantics representation of words. Embeddings can be learning with several algorithms. The most common algorithm is called word2vec and is a neural network-based model. Word2vec [34, 35] includes two main algorithms: continuous bag-of-words (CBOW) and skip-gram.

1. CBOW: Predicting target word from contexts. This model tries to predict the tth word, w_t, in a sentence using a window of width C around the word. Therefore, the context words $w_{t-C}, w_{t-C+1}, \ldots, w_{t-1}, w_{t+1}, \ldots,$

w_{t+C-1}, w_{t+C} are at the input layer of the neural network model to predict the target word w_t.

2. Skip-gram: Predicting contexts from target word.
   This model is the opposite of the CBOW model. The target word is at the input layer, and the context words are on the output layer.

Continuous Bag-of-Words The CBOW model is similar to a feed-forward neural network, where the hidden layer is removed and the projection layer is shared for all words. The model architecture is shown in Fig. 3.

The model receives as input context words and seeks to predict the target word w_t by minimizing the CBOW loss function:

$$L_{\text{CBOW}} = -\frac{1}{|C|}\sum_{t=1}^{|C|}\log P(w_t|w_{t-C},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+C})$$

$P(w_t|w_{t-C},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+C})$ is computed using the softmax function:

$$P(w_t|w_{t-C},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+C}) = \frac{\exp(\hat{x}_t^{\mathrm{T}} x_s)}{\sum_{i=1}^{|V|}\exp(\hat{x}_i^{\mathrm{T}} x_s)}$$

where x_i and $\hat{x}_i$ are the word and context word embeddings of word w_i respectively. x_s is the sum of the word embeddings of the words $w_{t-C},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+C}$, and V is the vocabulary of the text dataset.

Mikolov et al. [34] called the CBOW model a bag-of-words because the order of the context words does not influence the projection. It is also called continuous, because rather than conditioning on the words themselves, we condition on a continuous vector constructed from the word embeddings.

Skip-Gram The skip-gram model is similar to CBOW, but instead of predicting a word based on the context, the context is predicted from the word. More precisely, the skip-gram architecture can be seen as a neural network without a hidden layer. It uses each word as input to the network to predict words within a certain range before and after that word (context size). This yields to the loss function:

$$L_{\text{Skip-Gram}} = -\frac{1}{|C|}\sum_{t=1}^{|C|}\sum_{-C\le j\le C, j\neq 0}\log P(w_{t+j}|w_t)$$

$P(w_{t+j}|w_t)$ is computed using the softmax function:

$$P(w_{t+j}|w_t) = \frac{\exp(\hat{x}_{t+j}^{\mathrm{T}} x_t)}{\sum_{i=1}^{|V|}\exp(\hat{x}_i^{\mathrm{T}} x_t)}$$

The skip-gram architecture is shown in Fig. 3. In this architecture, each word is generated multiple times; each time it is conditioned only on a single word. Increasing the context size in the skip-gram model increases the computational complexity, but it also improves quality of the resulting word vectors.

By training the word2vec model on this language modelling task (predicting words in context), the weights on the nodes in the neural network are continuously adapted in such a way that more similar words have more similar vector representations than less similar words. After training, the hidden layer of the network is stored as a dense vector representation for each word in the vocabulary. In the resulting vector space, closeness of words represents their similarity.

3 Pre-trained Language Models

As explained in the previous section, word embeddings are rich language representations: a dense vector for each term in the vocabulary. They are useful for word similarity applications, but if we want to use word embeddings models for the purpose of document representation instead, we need to go from word representations to document representations. One option is to combine the embeddings of all words in the document (e.g. by averaging), or to use a model such as doc2vec [36], which adds a document indicator to an embedding vector to learn document embeddings. Either way, these

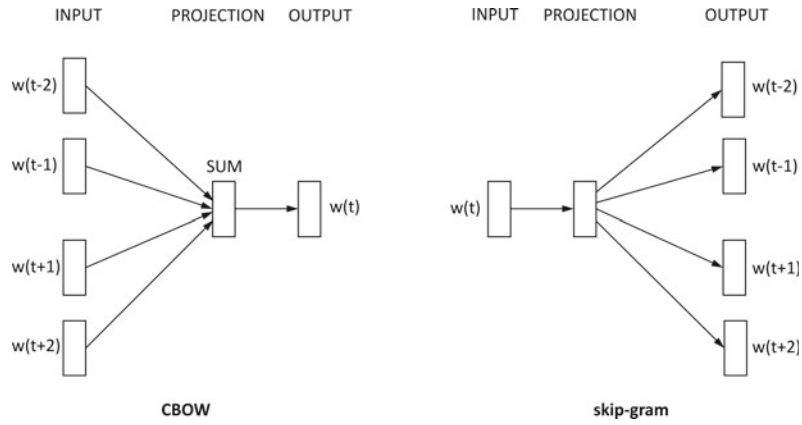

Fig. 3 Model architectures for the CBOW and the skip-gram model [34]

embeddings models are *static* in nature; they can be used as the input to a predictive model but are not updated during training.

A big leap forward in text representations for NLP was made by the introduction of pre-trained language models in the form of *dynamic* embeddings. These embeddings models can be directly used in supervised learning tasks by adding a classification layer on top of the embeddings architecture. During the supervised learning, the full network—including the input embeddings—is updated. This gave rise to the potential of *transfer learning* for text data [37]. Transfer learning is the principle of training a model on a large dataset and then transferring the learned parameters and finetuning them to a more specific, smaller dataset. Until 2018 transfer learning was possible for image data [38], not for text. Transfer learning is further described in Sect. 3.2. First, the next subsection will introduce BERT (Bidirectional Encoder Representations from Transformers), the most popular type of embeddings model in recent NLP.

3.1 Transformers and BERT

In 2017, a research team from Google introduced a new, powerful architecture for sequence-to-sequence learning: the transformer [39]. A transformer is an encoder-decoder architecture: in the encoder part it creates embeddings from input text; in the decoder part it generates text from the stored embeddings.

The core of the transformer architecture is the *self-attention mechanism* [40]. Prior architectures for sequential data (recurrent neural networks such as LSTMs) process text as a sequence: left-to-right and right-to-left. This makes them inefficient because parallellization of the process on a computer cluster is not possible. The self-attention mechanism computes the relation between each pair of input words, thus processing the whole input in parallel. As a result, the context that is taken into account by a transformer is much larger (i.e. the complete input) than in an LSTM (see Sect. 2.4), which has to be trained strictly sequentially (token by token). The longer context in transformer models makes long-distance linguistic relationships possible. This is necessary for language understanding tasks. For example, in the sentence "My lectures, taught in lecture hall 1 to computer science master students on Wednesday mornings at 9 a.m., are about Text Mining", the verb *are* has *my lectures* as subject. With long-distance attention, transformer models can process this correctly—evidenced by the correct translation of the sentence by Google Translate. A disadvantage of self-attention is that it is memory-heavy: since it computes the relation (dot-product) between the embeddings vectors of each pair of words in the input, the computational complexity is quadratic to the number of tokens in the input.

The consequence is that training transformer models required high-memory GPUs.

A year after the introduction of the transformer, BERT was introduced: Bidirectional Encoder Representations from Transformers [41].[5] BERT is a transformer model with only an encoder part. This means that it serves to convert text to embeddings.[6] BERT was designed for transfer learning, which is further explained in the next subsection.

3.2 Transfer Learning: Pre-training and Fine-Tuning

BERT models are trained in two stages: the model is pre-trained on a large—huge[7]—unlabeled text collection and then fine-tuned with a much smaller amount of labelled data to any supervised NLP task. BERT uses almost the same architecture for pre-training and fine-tuning: the dynamic embeddings vectors learned during pre-training are updated during fine-tuning.

The pre-training stage is *self-supervised*, following the same language modelling principles as static word embeddings without any labelled data. In BERT, two language modelling tasks are used during pre-training: Masked Language Modelling and Sentence Prediction. Masked Language Modelling is the task of predicting words based on their context. A proportion (typically 15%) of all tokens is replaced by the token [MASK] and while processing the text collection the model tries to predict what the word in place of the [MASK] token is. The second pre-training task, Sentence Prediction, takes place in parallel with Masked Language Modelling: based on the current sentence, the model tries to predict which of two alternatives is the next sentence. The goal is to learn relations between sentences, which is valuable for tasks such as question answering. Huge amounts of text data are needed to pre-train a BERT model, but thanks to the developers and the research community, pre-trained BERT models are shared for re-use by others. The largest repository of transformers, Hugging Face, contains almost 100,000 models, of which almost 10,000 BERT models for over 150 languages at the time of writing.[8]

Once pre-trained, the embeddings can be fine-tuned using labelled data to a supervised learning task. This can be a classification task (e.g. clinical code prediction, sentiment classification) or a sequence labelling task (e.g. named entity recognition). The last layer of the model defines the loss function and the labels that the model learns to predict.[9]

3.3 BERT Models in the Health Domain

BERT proved to be highly effective for many NLP tasks, outperforming state-of-the-art models. BBecause of its popularity and effectiveness, researchers have trained and released BERT models for specific domains. Generally speaking, there are three strategies for creating a domain-specific model: (1) pre-training a model from scratch on domain-specific data; (2) further pre-training an existing, generic, BERT model by adding domain-specific data to it; (3) no domain-specific pre-training, but only fine-tuning a generic model to a domain-specific task. The first strategy requires a huge amount of data and advanced computational resources (high-memory GPU cores) and is not a realistic choice for most researchers. The second strategy is therefore more common. In both the second and third strategy, the vocabulary of the original model is kept, as a result of which some of the domain-specific terms are not in the model's vocabulary and will be split in sub-words by the tokenizer.

BERT and other transformer models use a tailored tokenization method, called WordPiece [42]. The principle is that the vocabulary size (number of terms) is pre-given and fixed,

[5] The preprint was released in 2018; the paper published in a conference in 2019.

[6] A text generation transformer such as GPT-2 is decoder-only, generating text from embeddings.

[7] Typically, the whole wikipedia and a large book corpus.

[8] https://huggingface.co/models?search=bert.

[9] Hugging Face has example code available for fine-tuning: https://huggingface.co/docs/transformers/training.

typically at 30,000. While pre-training, WordPiece optimizes the coverage of the vocabulary of the collection using 30,000 terms. Words that are relatively frequent will become a term on their own, while words that are infrequent are split into more frequent subtokens. This splitting is not necessarily linguistically motivated. The authors of the BioBERT paper [43] give the example of *Immunoglobulin* that is tokenized by WordPiece as `I ##mm ##uno ##g ##lo ##bul ##in`, the hashes indicating that the tokens are subwords.

BioBERT was the first BERT model in the biomedical domain. BioBERT was pre-trained on PubMed Abstracts and PMC Full-text articles together with the English Wikipedia and BooksCorpus. In the paper it was shown to be successful on biomedical NLP tasks in 15 datasets for three types of tasks: named entity recognition (e.g. extracting disease names), relation extraction (e.g. extracting the relation between genes and diseases), and question answering [43]. Later, more biomedical models followed, specifically Clinical BERT [44], pretrained on the MIMIC-III data.

It became common in the past years to not only release pre-trained models on Huggingface, but also models that have been fine-tuned to a specific task, for example named entity recognition[10] or sentiment classification[11] [45]. This is valuable for users who don't have the computational resources or labelled data to fine-tune a model themselves. In addition, these models can also serve as a starting point for more specific fine-tuning tasks. For example, one could re-use a BioBERT model that was fine-tuned for named entity recognition of diseases, and use it either as-is ('zero-shot use') to label an unlabelled collection with disease names, or fine-tune it further to another set of labelled data for disease recognition.[12]

A challenge when extracting biomedical entities in text (e.g. diseases, medications, side effects), is that the extracted entities need to be normalized for spelling errors and other variations: there are multiple ways to refer to the same entity, e.g. because of the difference between specialist and layman language. The common approach to entity normalization is *ontology linking*: connecting a mention in a text (e.g. "cannot sleep") to a concept in a medical term base (e.g. *insomnia*). Medical terminologies, of which the most commonly used in the clinical domain is SNOMED CT, can be huge, with tens of thousands different labels. A model linking entities from the text to the SNOMED terminology needs to be able to connect terms it has not seen during training time to labels from this huge label space. A BERT model fine-tuned for this particular task is SapBERT [46].

4 NLP Tasks and Challenges in Healthcare

Text data are abundant in the health and biomedical domain. There exist a large variety of text data types from which information extraction could be valuable, ranging from scientific literature to health social media. In this section we will discuss issues related to data privacy, existing datasets and applications of NLP in the health and biomedical domain.

4.1 Data Privacy

Healthcare information exchange can benefit both healthcare providers and patients. Healthcare data are universally considered sensitive data and are subject to particularly strict rules to be protected from unauthorized access. Because of privacy concerns, healthcare organizations have been extremely reluctant to allow access to care data for researchers from outside the associated institutions. Such restricted access to data has hindered collaboration and information exchange among research groups. Because of the recent introduction of technologies such as

[10] e.g. https://huggingface.co/raynardj/ner-disease-ncbi-bionlp-bc5cdr-pubmed.

[11] e.g. https://huggingface.co/raynardj/ner-disease-ncbi-bionlp-bc5cdr-pubmed.

[12] It is good to be aware of the distinction between *cased* and *uncased* models. Cased models have been pre-trained with capitalisation preferred, while uncased models have all capitals removed.

differential privacy [47, 48], federated learning [49], synthetic data generation [50] and text de-identification (text anonymization) [51], we expect the increase in data sharing, facilitating collaboration, and external validity of analysis using integrated data of multiple healthcare organizations. The extent of data sharing required for widespread adoption of data science and specifically natural language processing technologies across health systems will require extensive collaborative efforts.

Clinical **text de-identification** is one of the easiest methods enables collaborative research while protecting patient privacy and confidentiality; however, concerns persist about the reduction in the utility of the de-identified text for information extraction and natural language processing tasks. On the other hand, growing interest in **synthetic data** has stimulated development and advancement of a large variety of deep learning-based models for a wide range of applications including healthcare.

Federated learning enables collaborative model training, while training data remains distributed over many clients, minimizing data exposure. On the contrary, **differential privacy** is a system for publicly sharing information about a dataset by describing the patterns of groups within the dataset while withholding information about individuals in the dataset.

4.2 Biomedical Data Sources and Their Challenges

Scientific papers and patents. In their 2015 paper, Fleuren and Alkema [7] show the strong increase of the number of scientific publications between 1994, 2004 and 2014. We can only imagine how much this increase has progressed since then. Scientific papers are challenging for NLP techniques because they are long, often stored as PDF with headers, footers, captions, mid-sentence line endings, potential encoding issues, containing figures and tables, and technical language. Similarly challenging to process are patent documents; the amount of biomedical and biotechnical patents is large. Patents are a rich source of information, but also long, multilingual, and with technical and legal language [52].

Electronic Health Records (EHRs).EHRs receive a substantial amount of research in biomedical NLP [53]. The text data in EHRs, consisting of doctor notes and letters, provide rich information in addition to the structured data in the records, and therefore are promising sources for mining biomedical knowledge (see Sect. 4.3 for some key examples). The use of patient health records brings challenges related to pre-processing: doctor notes are written under time pressure, and contain typos and doctor-specific abbreviations. For example, the word *patient* is abbreviated by one doctor to 'pnt', by the second doctor to 'pt' and by the third even to 'p'. Another challenge for the use EHRs is data privacy: the anonymization of text data is challenging [8]. Recently, some work has addressed the potential of generating artificial EHR text for use in benchmarking contexts [54]. This direction is promising, and can be expanded upon in the near future with the fast improving quality of large generative language models such as Generative Pre-trained Transformers (GPT) [55].

Health social media. A more freely available source of patient experiences is health social media [56]: information shared on general platforms such as Twitter, and Reddit, but also disease-specific discussion forums in patient support groups. These data are direct personal accounts of experiences, without filtering through a questionnaire or interview. This makes the data potentially rich, but also noisy—not all information in the patient accounts is necessarily correct and of high-quality. Like with EHRs, the use of health social media data poses challenges with pre-processing and normalization, such as spelling errors and the use of medical language by laymen [57], and with data privacy. Under the GDPR, medical information shared online, also on a public channel, is considered personal information and should be handled with care.

An anonymous alternative source of patient experiences are the patient surveys conducted by hospitals. These surveys are not asking for specific medical and personal information, but for cus-

tomer satisfaction aspects: how did patients experience their stay and what can be improved [58]. These data are less privacy sensitive and therefore easier to use, but also less rich in content and can only be used to analyze general trends of patient satisfaction [59].

4.3 Tasks and Applications

NLP tasks in the biomedical domain directly relate to the data sources that are available. We will discuss tasks related to the three types of data sources described in the previous subsection.

Scientific papers and patents. For the purpose of biomedical scientific research, mining knowledge from large bodies of biomedical papers is relevant, because individual papers only address one topic at the time, and the amount of papers published is large. Fleuren and Alkema [7] describe biomedical text mining task for scientific publications: starting with information retrieval to select the topically relevant papers from a large collection, followed by named entity recognition, relation extraction, knowledge discovery, and visualization. The most commonly addressed named entity recognition task is the extraction of diseases, genes and protein names from scientific tasks. Fleuren and Alkema [7] list benchmark tasks that have helped advancing the methods development for named entity recognition. The task of gene, protein, disease extraction can be expanded from scientific papers to patents, thereby also expanding from English-only to multiple languages [60].

NLP technology can also support the task of systematic reviewing of scientific publications, typically performed by clinical librarians or medical scholars [61]. Systematic reviewing is a challenging task, even for trained users, who compose long Boolean queries to select relevant papers to the topic of their review [62]. Text classification models can help the process of paper selection, but since the task is high-recall—the user cannot miss any relevant paper—should always be conducted in interaction with the human expert. Techniques such as Continuous Active Learning [63] allow for this interaction.

Electronic Health Records (EHRs). In the past two decades, biomedical NLP research has largely aimed at development of predictive models for EHRs [64]. Predictive models are classification tasks for the purpose of predicting future events. Past records are used as training data. Examples of such tasks are the prediction of clinical risks [65], the prediction of diagnosis codes based on free-text notes [66], the prediction of a patient's time to death for general practitioners [67], the prediction of hospital admissions in emergency departments [68], and the prediction of re-admissions after discharge [69].

Challenges in some clinical prediction tasks are huge label spaces: the ICD-10 coding system, used to code a patient's diagnosis, has tens of thousands of codes.[13] When training a machine learning model, the codes that are frequent in the training data will be well represented by the model and easy to predict, while the rare diseases have not sufficient training data to be correctly predicted in the test data. A second challenge is bringing the developed models to the clinical practice. Before a hospital takes the step to involve machine learning and NLP in the clinical workflow, the developed applications need to be evaluated in an end-to-end setting with user involvement. The models are typically aimed to not replace the human expert (the doctor or the clinical information specialist), but to assist them in making the right decisions. One example application in the hospital context is to discover misclassifications or inconsistencies in previously coded data [70, 71]. Another application is to use the machine learning model to make suggestions in an interactive task context, e.g. suggest the most likely diagnosis code based on the text typed by the doctor or coder [72].

Health social media. Health social media data can be used for the extraction of structured information, such as side effects for medications [73, 74], but also for more social-emotional aspects of patients' well-being, such as patient empower-

[13] https://www.cdc.gov/nchs/icd/icd10.htm.

ment [75]. The most commonly addressed health-related task with social media is the extraction of adverse drug reactions (ADRs), for which high-quality benchmarks have been developed [76]. The extraction of ADRs is defined as an information extraction task consisting of three steps: (1) named entity recognition to identify medications and ADRs; (2) ontology linking to normalize the extracted ADR string (e.g. "cannot fall asleep") to the correct term in a medical database (e.g. *insomnia*); (3) relation extraction to identify that the mentioned ADR is indeed connected to the mentioned medication.

5 Bias and Fairness

In this chapter we have seen how we can apply artificial intelligence algorithms to extract information and insights from real-world clinical text data. These AI algorithms draw their insights and information by generalizing observations from their training data to new samples. Sometimes this generalization can be grounded on an incorrectly assessed correlation between an input feature and an effect. This is known as *bias* [77]. As an example, consider an image classifier that is trained to distinguish wolves from dogs. If the classifier decides something is a wolf (rather than a dog) based on the snow in the background [78], then it is biased because it is not the snow that makes a wolf a wolf. This classifier will struggle to distinguish dogs from wolves in scenarios where the background is not visible, or if a dog happens to be surrounded by snow.

A related but somewhat distinct concept is *fairness*: how well people who are similar to each other are treated similarly by an AI system [79]. To see how fairness relates to bias, consider the following example [80]. An AI system is trained to determine whether benzodiazepines should be prescribed to a psychiatric patient, on the basis of certain information about the patient. The training data would be annotated with real prescriptions from past data. Suppose that one of the pieces of information available to the AI system is the biological gender of the patient, and suppose further that there is a high correlation between biological gender and past prescriptions [81]. The AI system might use the correlation between gender and past prescriptions to inform future predictions. This is biased, because biological gender is not expected to have any impact on whether a patient should be prescribed benzodiazepines [82, 83]. It is also unfair, because by discriminating on biological gender, the system might be treating otherwise equal patients differently. For a real-world example, Singh et al [84] found that a predictive model for mortality risk failed to generalize from one hospital to another, and that this resulted in disparate impact for different races.

Bias and fairness in AI have garnered attention for several years [85, 86]. We will use the terms bias and unfairness interchangeably to describe a situation in which an AI system uses certain *protected attributes* [79] implicitly or explicitly for a purpose that is unrelated to the value of the protected attribute. Protected attributes vary by country and by domain, but they typically include gender, nationality, race, and age, among others [87]. The challenge can sometimes arise from the fact that these attributes can be correlated with other features in the dataset, so that removing the protected attribute from the features used in the AI system does not remove the bias [88].

In this section we will outline some of the causes of bias in AI applications for clinical text analyses, as well as how to measure and mitigate those biases. We will also highlight some of the challenges associated with the study of bias given the limitations imposed by real-world clinical data.

5.1 Bias in Clinical NLP

Bias can be introduced at multiple points in the AI pipeline for clinical applications. We will introduce four common ways in which bias can occur. First, *selection bias* can be present in the dataset used for training an algorithm due to a sampling problem [89]. A notable example is *healthcare*

access bias [77]: patients admitted to an institution do not necessarily represent the whole population they are drawn from. Therefore, using data from a single institution to draw insights about a population might be biased.

Second, bias can be intrinsically incorporated in the population, as in the case where more members of a protected group have a certain characteristic than non-members for historical reasons. Take the classical example of loan approvals presented in the introduction to this section. The correlation between ethnicity and postal code is due to social or historical reasons, and is not related to loan approval.

Third, bias can be caused by design choices in the AI system. For example, a clinician might decide to work on implementing a classifier to detect a sickness that only affects a subset of the population, while ignoring other sickness that affect another segment of the population [90].

Fourth, bias can also happen when systems trained on language varieties that are considered "standard" work less well on texts written by certain sociodemographic groups [91]. In the clinical practice, this could have a significant impact when designing models trained on texts written by patients from a given institution [92], as the application of these models on other institutions might lead to bias.

Bias can be dangerous for clinical NLP and text mining applications, but before we can do something about it, we must be able to identify bias. This can be complicated because it is not always clear whether bias should be removed. As an extreme example, consider an AI system trained to predict the probability of a (biologically) female patient becoming pregnant in the next three months based on reports written by doctors during general screenings. Suppose that the doctors are instructed to never write the age of the patient in the reports. They might, however, write other information that correlates with age. The AI system could then associate this information with the pregnancy status and use it to predict pregnancy. As a result, the AI system would "bias" its predictions against older women. As age can be considered a protected attribute, this could be considered unfair bias. In this case, however, there might be a medical reason why the prediction should be different for different ages.

Nevertheless, there are cases in which it is clear that bias should be mitigated if possible. As an example, consider an NLP system designed to predict a diagnosis from a written report. Suppose this NLP system is biased against a protected group, and that the illness the system tries to diagnose is potentially fatal. As a result, members of the protected group go undetected and die more often as a result of the sickness. This means that fewer patients come back for further treatment, and as a result there are fewer written reports about patients from the protected group to use as training data for newer models. This creates a feedback loop that results in the bias becoming even larger [93].

5.2 Bias Measurement

Bias can be measured using multiple metrics, depending on the specific details of the case. The very definition of bias is highly contested, with a recent review citing more than ten of them [93]. Listing all possible definitions is beyond the scope of this chapter, but we can sketch out two of them to give an idea of where differences in definitions come from. For illustrative purposes, consider a dataset containing patient records for white and black patients.[14] Suppose this dataset is annotated with *gold labels* representing whether the patient is diagnosed with a particular sickness or not. We want to train a binary classifier to predict this diagnosis in new non-annotated data: given a new patient record, the *predicted label* is *positive* if the model thinks the patient has the sickness, or *negative* if not. The *equal opportunity* definition of fairness requires that datapoints with a positive gold label have the same probability of being assigned a positive predicted label by the model; in other words: if we knew that a given patient has the sickness, the model should have the same probability of predicting *true positives* regardless of the race of the patients. The *equalized odds* definition requires exactly the same, and additionally

[14] In other words, we remove all records for patients who identify as belonging to any other race from the dataset for this example.

that all protected groups having a *negative* gold label should have the same probability of being (incorrectly) predicted as positive [94]; in other words: the model should have the same probability of predicting true positives *and false positives* regardless of the patient race.

Additionally, another question to be considered is whether we want *individual* fairness or *group* fairness. Individual fairness means that similar individuals get treated similarly. In the example above, this would mean that two patients with similar age, socioeconomic status, health status, etc., but of different races, should receive the same treatment by the model. Group fairness requires that each group gets treated similarly, so that the performance of the model is similar for each group. In the example above, this could mean that the accuracy of the model is the same for black and white patients. Individual fairness is very hard to implement, given that some kind of similarity metric needs to be defined.

Guidelines for selecting an appropriate bias measure depend on the specific use case [95]. As an example, suppose you are developing a system to help clinicians diagnose a disease. We assume that receiving a diagnosis is desirable, as it helps speed up treatment. As such, the designer of the system will prioritize minimizing the false negatives, to ensure no sick people go undetected. In that case, equal opportunity might be a better bias measure than equalized odds, as we are not so concerned with bias occurring in false positives. In a concrete example from the literature [96], a model trained to predict depression from clinical notes found a bias against patients of a given gender. They quantified the bias using the False Negative Rate Ratio (FNRR), i.e., the false negative rate for members of that gender divided by the false negative rate for other patients. The false negative rate is the fraction of patients with depression that were diagnosed by the model as not having the condition. They found the FNRR to be different from 1, which is the value expected if the classifier were fair. In practice, it's often not possible to satisfy multiple fairness metrics at the same time, therefore making it even more important to select one based on the domain.

An important remark to be made when it comes to measuring bias in clinical NLP applications is that clinical datasets are often heavily imbalanced. Often clinical NLP systems aim at extracting rare symptoms, detecting rare diseases, or predicting rare events. This should be taken into consideration when choosing a bias measure. For example, metrics emphasizing differences in the True Negative Rate are often inappropriate, as the True Negative Rate is usually very large due to the imbalanced nature of the dataset.

5.3 Bias Mitigation

Multiple bias mitigation techniques have been proposed [87] for machine learning applications. These can be classified as pre-processing, in-processing, or post-processing techniques. Pre-processing mitigation techniques attempt to debias by making modifications to the training dataset, such as applying different weights to sample from different protected groups. In-processing mitigation techniques attempt to debias by modifying the NLP and text mining algorithms; a popular example is the *prejudice remover* [97]. Finally, post-processing techniques attempt to debias by modifying the way predictions from the model are interpreted. As in the case of measuring bias, mitigating bias is also context-dependent, and the right tool should be chosen based on the domain and the task.

In recent literature, one study uses data augmentation to mitigate bias: they create new datapoints by swapping gender pronouns in the input documents, and find a difference in the fairness measures [96]. A recent survey outlines several more studies that used bias mitigation techniques [98]. As a complementary strategy, some argue that every dataset should be accompanied by a *data statements* providing enough information so that users can understand what biases might be present in the dataset [99].

6 Explainability

The advancements in AI and NLP with the emergence of deep learning approaches have led to systems with high predictive accuracy, which however, are based on very complex learning processes that are very difficult for users and researchers to understand. The difficulty to understand the internal logic and how those systems are reaching predictions is known as the *Black Box problem* and has led to an increasing interest of researchers to explainable AI (XAI) and interpretable AI.

Although the term XAI is mentioned already in a study published in 2004 [100], there is still no standarized technical definition. In literature, many times *transparency*, *explainability* and *interpretability* are used interchangeably [101]. Many researchers have already attempted to give formal definitions. Gilpin et al. [102] stated that both interpretability and fidelity are required to achieve explainability. According to Gilpin et al. *interpretability* refers to whether the explanation is understandable by humans, whereas *fidelity* refers to whether the explanation describes the method accurately. Based on that, Markus et al. [103] defined *explainability* as follows: *An AI system is explainable if the task model is intrinsically interpretable or if the non-interpretable task model is complemented with an interpretable and faithful explanation.* On the other hand, *transparency* has been defined as providing stakeholders with relevant information about how the model works that can include documentation of the training procedure and code releases [104].

From the above definitions, it is evident that XAI and transparency are very important for AI and NLP systems developed for the clinical domain. XAI models in healthcare should align with clinicians' expectations and acquire their trust, increase the transparency of the system, assure results quality, and allow addressing fairness, and ethical concerns [105].

In this section we will outline some of the main methodologies that have been used for explainability of AI applications for clinical text analyses, and how they were evaluated. We will also highlight some of the challenges and limitations associated with the explainability in AI and NLP in clinical applications.

6.1 Methods for Explainability

One of the aims of a XAI model is to produce explanations regarding the system's process and outcome predictions. Those explanations can be categorized in two groups: local and global [106]. The *local* explanations refer to providing explanation on an individual prediction, whereas the global refers to the model's prediction process as a whole. The *global* explanations can either emerge from the prediction process (self-explaining) or after post-processing (post-hoc).

There are several well known techniques that can have been proposed to generate explanations. One of the most well known models is LIME (Local Interpretable Model-Agnostic Explanations) that focuses on local explanations [78]. LIME is based on surrogate models which are trained to approximate the predictions of the initial non-explainable model. Surrogate models can also be learned for global explanations [107]. Although XAI methods based on surrogate models became very popular, they have a main drawback which is that the original model and the learned surrogate models may have completely different ways to reach the predictions.

SHapley Additive exPlanation (SHAP) is another popular Explainable AI (XAI) model that can provide model-agnostic local explainability for different types of data [108]. SHAP is based on Shapley values, which is a concept popularly used in Game Theory and is applies additive feature importance.

Many researchers also tried to derive explanations using the importance scores of different features on the output predictions. This can be applied on manual features derived from traditional feature engineering [109], lexical features [66] or gradient-based methods such as DeepLIFT [110] or Grad-CAM [111]. In particular, DeepLIFT is designed to compute feature importance in feed-forward neural networks, whereas Grad-CAM uses the gradients of a target

concept flowing into the final convolutional layer and produces a coarse localization map highlighting the important regions for predicting the concept.

The extraction of weights from the attention mechanism is also a very popular way to enable feature-based explanations. Attention layers that can be added to most neural network architectures, indicate the parts that the network focuses. The package BERTviz[15] uses this premise to visualize the attention between input tokens, in particular between the [CLS] token—which has information for the prediction itself—to each of the input tokens. However, they have become a topic of debate on whether they can be used as a means of explanation or not. Jain and Wallace [112] claimed that there is no correlation between attention scores and other feature-important measures concluding that attention is not explanation. However, Wiegreffe and Pinter [113] proposed diagnostic tests to allow for meaningful interpretation of attention, but also showed that adversarial attention distributions could not achieve the performance of real model attention.

6.2 Evaluation of Explainability

One of the current challenges in XAI refers to their proper evaluation. It is important that the explainable models to be evaluated not only on their performance but also on the quality of the explanations. Taking into account that explainability is a relatively new field, there is still no agreement regarding a standarized evaluation of the XAI models.

One approach that has been applied, is to present an informal evaluation of the explanations and high level discussions of how some of the generated explanations agree with human intuition. In some cases explanations are even compared to other reference approaches [114] such as LIME.

A more formal way to evaluate an XAI approach is to use human evaluations that can quantify a system's performance [115]. The collected ground truth can be then compared with the generated explanations and state-of-the-art performance metrics such as Precision/Recall/F1 and BLUE scores can be calculated. Instead of collecting ground truth beforehand, an alternative evaluation approach is to ask humans to evaluate the explanations generated by the XAI system [66]. Although collecting human labels is a way to quantify the performance of those systems, they are not always of high quality. Also, humans have many biases that can be also reflected in the collected ground truth. Multiple annotators of diverse backgrounds and high inter-annotator agreement is a way to ensure the quality of the labels.

Attention based explanations have been also evaluated by more specific approaches. For example, Serrano and Smith [116] performed experiments in which they repeatedly set the maximal entry generated by the attention layer to zero. The idea behind this mechanism is that turning off those weights should lead to different explanations in the case that they actually explain the predictions.

One limitation of the current studies is the limited or even absent elaboration on what is being actually evaluated. Explanations can be evaluated from different angles such as fidelity and comprehensibility [117]. One exception is the study by Lertvittayakumjorn and Toni [118] who proposed human evaluation experiments targeting the following three goals: model behavior, model predictions and assist humans in investigating uncertain predictions.

6.3 Explainability in Clinical NLP Tasks

The widespread use of AI and NLP models into clinical practice have made transparency and explainability of critical importance, especially if we consider not only that practitioners usually work with complex sources of data [119] but also that incorrect predictions can lead to severe

[15] https://github.com/jessevig/bertviz.

consequences [120]. In order to build trust between clinicians and AI models, clinicians should be able to understand the logic of the system and detect cases in which the model gave incorrect or unexpected predictions.

There have been several attempts for XAI models for different prediction tasks in the medical domain ranging in the type of data they use [119, 121]. Some of those works focus on XAI models for text prediction tasks in the medical domain. The easiest and most straightforward way is to apply well known models such as LIME, SHAP and DeepLIFT to generate explanations. For example, Uddin et al. [122] proposed an RNN system for depression detection from text and applied LIME to generate explanations of the predictions. Caicedo-Torres and Gutierrez [123] applied SHAP to generate explanations of their proposed deep learning system that was trained to predict patient mortality inside the ICU based on free-medical notes. DeepLIFT that is designed to compute feature importance in feed-forward neural networks was used by Caicedo-Torres and Gutierrez [123] to find word embeddings that deemed as most important for survival and death prediction.

Combing convolution with attention has been proved efficient in different NLP tasks. To this end, Mullenbach et al. [66] applied attentional convolution to highlight the most relevant parts of the clinical text of each ICD code. Hu et al. [124] focused also on ICD classification and proposed SWAM which established the correspondences between the informative snippet and convolution filter. Blanco et al. [125] proposed a bidirectional Gated Recurrent Units (GRU) with attention mechanism that allowed to understand which fragment contributed the most in the cause of death prediction.

7 Summary and Recommendations

7.1 Clinical Natural Language Processing

As the amount of unstructured text narratives that biomedical and healthcare systems produce grows, so does the need to intelligently process it and extract different types of knowledge from it. In the future, with an active role of the health community, more clinical NLP-based expert systems will be deployed in practice to accurately recognize the knowledge within clinical text, and feed this knowledge automatically into patient daily care.

7.2 Transfer Learning in Health

In NLP as well as in many areas of machine learning, the standard way to train a model is to annotate a number of examples that are then provided to the model. Recent deep learning-based transfer learning methods and pre-trained language models have achieved remarkable successes on a wide range of NLP tasks. Given the lack of annotated datasets for training and benchmarking in clinical text mining, in the future, it is expected that the knowledge from related tasks or domains are combined. We also expect, for the NLP tasks in healthcare, more effective approaches that combine semi-supervised learning with transfer learning.

7.3 Bias and Fairness

Bias in NLP occurs when an algorithm or model exploits certain properties of texts to solve a task that is unrelated to those properties. *Fairness* is a requirement that machine learning models treat members of different *protected groups* equally. For our purpose, we consider an NLP or text mining model to be biased or unfair if it uses certain protected attributes implicitly or explicitly to solve a problem unrelated to those attributes. There are multiple definitions of bias, as well as multiple bias metrics, such as *equal opportunity*

and *equalized odds*. There are also several *mitigation* strategies that can be adopted to reduce the bias. The choice of a bias definition, a bias measure, and a bias mitigation strategy is dependent on the domain and the task, as different measures cannot be optimized simultaneously, and different tasks require different measures. Some work on bias measurement and mitigation has been done on the clinical NLP domain, but it is very much a nascent field, and no measure or mitigation strategy should be adopted without careful evaluation.

7.4 Explainability

In Sect. 6 we discussed what is XAI and the main methodologies that exist. In medical domain, XAI models aim to increase the trust of the practitioners and patients by providing transparent systems that are understandable by humans. Developing automated systems that could potentially take decisions for diagnosis and treatment is a multidisciplinary process. Models should be developed in collaboration with experts input from the appropriate areas. That will allow to understand domain-specific needs such as the purpose of the system, the need and level of required transparency and explainability. Additionally, the type of explanations should be decided considering not only the aspects of ethics and fairness, but also the limitations of the audience [126].

Another remaining challenge is related to the evaluation, a topic of a great discussion in the area. The majority of studies are using subjective measurements, such as user satisfaction, and researchers' intuition on the explanations [126]. From the previous studies, it is evident that there is an overall lack of validated and reliable evaluation metrics on which more work is needed. Zhou et al. [127] gave a summary of quantitative metrics for the evaluation of explainability aspects (i.e., clarity, broadness, parsimony, completeness, and soundness). In their study, they conclude that *the evaluation of ML explanations is a multidisciplinary research topic.* and that *It is also not possible to define an implementation of evaluation metrics, which can be applied to all explanation methods*.

References

1. Bagheri A. Text mining in healthcare: bringing structure to electronic health records. PhD thesis, Utrecht University; 2021.
2. Sutton RT, Pincock D, Baumgart DC, Sadowski DC, Fedorak RN, Kroeker KI. An overview of clinical decision support systems: Benefits, risks, and strategies for success. NPJ Digital Med. 2020;3(1):1–10.
3. Spasic I, Nenadic G, et al. Clinical text data in machine learning: systematic review. JMIR Med Inform. 2020;8(3): e17984.
4. Hearst MA. Untangling text data mining. In: Proceedings of the 37th annual meeting of the association for computational linguistics; 1999. p. 3–10.
5. Mehta N, Pandit A. Concurrence of big data analytics and healthcare: a systematic review. Int J Med Informa. 2018;114:57–65.
6. Yim W-W, Yetisgen M, Harris WP, Kwan SW. Natural language processing in oncology: a review. JAMA Oncol. 2016;2(6):797–804.
7. Fleuren WW, Alkema W. Application of text mining in the biomedical domain. Methods. 2015;74:97–106.
8. Menger V, Scheepers F, van Wijk L, Spruit M. DEDUCE: a pattern matching method for automatic de-identification of Dutch medical text. Telematics Inform. 2018;35(4):727–36.
9. Byrd R, Steinhubl S, Sun J, Ebadollahi S, Stewart W. Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records. Int J Med Inform. 2014;83(12):983–92.
10. Jamian L, Wheless L, Crofford LJ, Barnado A. Rule-based and machine learning algorithms identify patients with systemic sclerosis accurately in the electronic health record. Arthritis Res Ther. 2019;21(1):305.
11. Jonnalagadda S, Adupa A, Garg R, Corona-Cox J, Shah S. Text mining of the electronic health record: an information extraction approach for automated identification and subphenotyping of HFPEF patients for clinical trials. J Cardiovasc Transl Res. 2017;10(3):313–21.
12. Wu X, Zhao Y, Radev D, Malhotra A. Identification of patients with carotid stenosis using natural language processing. Eur Radiol. 2020;1–9.
13. Kocbek S, Cavedon L, Martinez D, Bain C, Mac Manus C, Haffari G, Zukerman I, Verspoor K. Text mining electronic hospital records to automatically classify admissions against disease: measuring the impact of linking data sources. J Biomed Inform. 2016;64:158–67.
14. Koopman B, Karimi S, Nguyen A, McGuire R, Muscatello D, Kemp M, Truran D, Zhang M, Thackway S. Automatic classification of diseases from free-text death certificates for real-time surveillance. BMC Med Inform Dec Making. 2015;15(1):53.

15. Torii M, Fan J, Yang W, Lee T, Wiley M, Zisook D, Huang Y. Risk factor detection for heart disease by applying text analytics in electronic medical records. J Biomed Inform. 2015;58:S164-70.
16. Bagheri A, Sammani A, van der Heijden PG, Asselbergs FW, Oberski DL. Etm: Enrichment by topic modeling for automated clinical sentence classification to detect patients' disease history. J Intell Inform Syst. 2020;55(2):329–49.
17. Sammani A, Bagheri A, van der Heijden PG, Te Riele AS, Baas AF, Oosters C, Oberski D, Asselbergs FW. Automatic multilabel detection of icd10 codes in dutch cardiology discharge letters using neural networks. NPJ Dig Med. 2021;4(1):1–10.
18. Huang K, Altosaar J, Ranganath R. ClinicalBERT: modeling clinical notes and predicting hospital readmission; 2019. ArXiv preprint arXiv:1904.05342
19. Jonnagaddala J, Liaw S, Ray P, Kumar M, Chang N, Dai H. Coronary artery disease risk assessment from unstructured electronic health records using text mining. J Biomed Inf. 2015;58:S203-10.
20. Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text classification algorithms: a survey. Information. 2019;10(4):150.
21. Murphy KP. Machine learning: a probabilistic perspective. MIT Press; 2012.
22. Aggarwal C. Machine learning for text. Springer; 2018.
23. Blei D, Ng A, Jordan M. Latent Dirichlet allocation. J Mach Learn Res. 2003;3(1):993–1022.
24. Reed C. Latent dirichlet allocation: towards a deeper understanding. Available at obphio.us; 2012:1–13
25. Bagheri A, Saraee M, De Jong F. ADM-LDA: an aspect detection model based on topic modelling using the structure of review sentences. J Inform Sci. 2014;40(5):621–36.
26. Duarte D, Puerari I, Dal Bianco G, Lima JF. Exploratory analysis of electronic health records using topic modeling. J Inform Data Manage. 2020;11(2).
27. Li DC, Thermeau T, Chute C, Liu H. Discovering associations among diagnosis groups using topic modeling. AMIA Summits Transl Sci Proceed. 2014;2014:43.
28. Mosteiro P, Rijcken E, Zervanou K, Kaymak U, Scheepers F, Spruit M. Making sense of violence risk predictions using clinical notes. In: Huang Z, Siuly S, Wang H, Zhou R, Zhang Y, editors. Health information science. Cham: Springer International Publishing; 2020. p. 3–14.
29. Stevens K, Kegelmeyer P, Andrzejewski D, Buttler D. Exploring topic coherence over many models and many topics. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning; 2012. p. 952–61
30. Luo L, Yang Z, Yang P, Zhang Y, Wang L, Lin H, Wang J. An attention-based bilstm-crf approach to document-level chemical named entity recognition. Bioinformatics. 2018;34(8):1381–8.
31. Nasar Z, Jaffry SW, Malik MK. Named entity recognition and relation extraction: State-of-the-art. ACM Comput Surveys (CSUR). 2021;54(1):1–39.
32. Eisenstein J. Natural language processing; 2018.
33. Firth JR. A synopsis of linguistic theory, 1930–1955. Studies in Linguistic Analysis 1957.
34. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J, Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems; 2013. , p. 3111–9.
35. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space; 2013. ArXiv preprint arXiv:1301.3781.
36. Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning. PMLR; 2014, p. 1188–96
37. Ruder S. Neural transfer learning for natural language processing. PhD Thesis, NUI Galway; 2019.
38. Huh M, Agrawal P, Efros AA. What makes imagenet good for transfer learning? 2016. ArXiv preprint arXiv:1608.08614.
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inform Proc Syst. 2017;30.
40. Jurafsky D, Martin J. Speech and language processing: an introduction to speech recognition, computational linguistics and natural language processing, 3rd edn. Prentice Hall; 2021.
41. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, Volume 1 (Long and Short Papers); 2019. p. 4171–86.
42. Wu Y, Schuster M, Chen Z, Le QV, 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; 2016. ArXiv preprint arXiv:1609.08144.
43. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020;36(4):1234–40.
44. Alsentzer E, Murphy JR, Boag W, Weng W-H, Jin D, Naumann T, McDermott M. Publicly available clinical bert embeddings, 2019. ArXiv preprint arXiv:1904.03323
45. Barbieri F, Camacho-Collados J, Anke LE, Neves L. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. Find Assoc Comput Linguist: EMNLP. 2020;2020:1644–50.
46. Liu F, hareghi E, Meng Z, Basaldella M, Collier N. Self-alignment pretraining for biomedical entity representations. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, (Online), Association for Computational Linguistics; 2021. p. 4228–38

47. Dwork C, Roth A, et al. The algorithmic foundations of differential privacy. Found Trends® Theor Comput Sci. 2014;9(3–4):211–407.
48. Price WN, Cohen IG. Privacy in the age of medical big data. Nat Med. 2019;25(1):37–43.
49. Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: strategies for improving communication efficiency. In: NIPS Workshop; 2016.
50. Eigenschink P, Vamosi S, Vamosi R, Sun C, Reutterer T, Kalcher K. Deep generative models for synthetic data. ACM Comput Surv. 2021.
51. Obeid JS, Heider PM, Weeda ER, Matuskowitz AJ, Carr CM, Gagnon K, Crawford T, Meystre SM. Impact of de-identification on clinical text classification using traditional and deep learning classifiers. Stud Health Tech Inf. 2019;264:283.
52. Verberne S, D'hondt E, Oostdijk N, Koster C, Quantifying the challenges in parsing patent claims. In: Proceedings of the 1st international workshop on advances in patent information retrieval at ECIR 2010; 2010. p. 14–21
53. Johnson AE, Pollard TJ, Shen L, Lehman L-WH, Feng M, Ghassemi M, Moody B, Szolovits P, Anthony Celi L, Mark RG. Mimic-iii, a freely accessible critical care database. Sci. Data 2016;3(1):1–9.
54. Libbi CA, Trienes J, Trieschnigg D, Seifert C. Generating synthetic training data for supervised de-identification of electronic health records. Fut Internet. 2021;13(5):136.
55. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al. Language models are few-shot learners. Adv Neural Inform Proc Syst. 2020;33:1877–901.
56. Weissenbacher D, Banda J, Davydova V, Zavala DE, Sánchez LG, Ge Y, Guo Y, Klein A, Krallinger M, Leddin M, et al. Overview of the seventh social media mining for health applications (# smm4h) shared tasks at coling 2022. In: Proceedings of the seventh workshop on social media mining for health applications, workshop and shared task; 2022. p. 221–41.
57. Dirkson A, Verberne S, Sarker A, Kraaij W. Data-driven lexical normalization for medical social media. Multimodal Technol Inter. 2019;3(3):60.
58. van Buchem MM, Neve OM, Kant IM, Steyerberg EW, Boosman H, Hensen EF. Analyzing patient experiences using natural language processing: development and validation of the artificial intelligence patient reported experience measure (aiprem). BMC Med Inf Dec Mak. 2022;22(1):1–11.
59. Bozik M. Aspect-based sentiment analysis on dutch patient experience survey data. Master's thesis, Master Computer Science, LIACS, Leiden University; 2022.
60. Hu Y, Verberne S. Named entity recognition for chinese biomedical patents. In: Proceedings of the 28th international conference on computational linguistics; 2020. p. 627–37.
61. Scells H, Zuccon G, Koopman B, Deacon A, Azzopardi L, Geva S. A test collection for evaluating retrieval of studies for inclusion in systematic reviews. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval; 2017. p. 1237–40.
62. Scells H, Zuccon G, Koopman B. A comparison of automatic Boolean query formulation for systematic reviews. Inf Retrieval J. 2021;24(1):3–28.
63. Cormack GV, Grossman MR. Scalability of continuous active learning for reliable high-recall text classification. In: Proceedings of the 25th ACM international on conference on information and knowledge management; 2016. , p. 1039–48.
64. Paxton C, Niculescu-Mizil A, Saria S. Developing predictive models using electronic medical records: challenges and pitfalls. In: AMIA annual symposium proceedings, American medical informatics association, vol 2013. 2013, p. 1109.
65. Goldstein BA, Navar AM, Pencina MJ, Ioannidis J. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. J Am Med Inf Assoc. 2017;24(1):198–208.
66. Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable prediction of medical codes from clinical text. 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. 1101–1111, Association for Computational Linguistics, June 2018.
67. Beeksma M, Verberne S, van den Bosch A, Das E, Hendrickx I, Groenewoud S. Predicting life expectancy with a long short-term memory recurrent neural network using electronic medical records. BMC Med Inf Decision Mak. 2019;19(1):1–15.
68. Lucini FR, Fogliatto FS, da Silveira GJ, Neyeloff JL, Anzanello MJ, Kuchenbecker RS, Schaan BD. Text mining approach to predict hospital admissions using early medical records from the emergency department. Int J Med Inf. 2017;100:1–8.
69. Huang Y, Talwar A, Chatterjee S, Aparasu RR. Application of machine learning in predicting hospital readmissions: a scoping review of the literature. BMC Med Res Methodol. 2021;21(1):1–14.
70. De Lusignan S, Khunti K, Belsey J, Hattersley A, Van Vlymen J, Gallagher H, Millett C, Hague N, Tomson C, Harris K, et al. A method of identifying and correcting miscoding, misclassification and misdiagnosis in diabetes: a pilot and validation study of routinely collected data. Diabetic Med. 2010;27(2):203–9.
71. Tate AR, Martin AG, Ali A, Cassell JA. Using free text information to explore how and when gps code a diagnosis of ovarian cancer: an observational study using primary care records of patients with ovarian cancer. BMJ Open. 2011;1(1): e000025.

72. Zhou L, Cheng C, Ou D, Huang H. Construction of a semi-automatic icd-10 coding system. BMC Med Inf Decision Mak. 2020;20(1):1–12.
73. Magge A, Tutubalina E, Miftahutdinov Z, Alimova I, Dirkson A, Verberne S, Weissenbacher D, Gonzalez-Hernandez G. Deepademiner: a deep learning pharmacovigilance pipeline for extraction and normalization of adverse drug event mentions on twitter. J Am Med Inf Assoc. 2021;28(10):2184–92.
74. Dirkson A, Verberne S, Kraaij W, van Oortmerssen G, Gelderblom H. Automated gathering of real-world data from online patient forums can complement pharmacovigilance for rare cancers. Sci Rep. 2022;12(1):1–9.
75. Verberne S, Batenburg A, Sanders R, van Eenbergen M, Das E, Lambooij MS. Analyzing empowerment processes among cancer patients in an online community: a text mining approach. JMIR Cancer. 2019;5(1): e9887.
76. Gurulingappa H, Rajput AM, Roberts A, Fluck J, Hofmann-Apitius M, Toldo L. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. J Biomed Inf. 2012;45(5):885–92.
77. Delgado-Rodríguez M, Llorca J. Bias J Epidemiol Commun Health. 2004;58:635–41.
78. Ribeiro MT, Singh S, Guestrin C. Why should i trust you? explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 1135–44.
79. d'Alessandro B, O'Neil C, LaGatta T. Conscientious classification: A data scientist's guide to discrimination-aware classification. Big Data. 2017;5(2):120–34.
80. Mosteiro P, Kuiper J, Masthoff J, Scheepers F, Spruit M. Bias discovery in machine learning models for mental health. Information 2022;13(5).
81. Olfson M, King M, Schoenbaum M. Benzodiazepine Use in the United States. JAMA Psychiatry. 2015;72(2):136–42.
82. Federatie Medisch Specialisten. Angststoornissen. 2010. https://richtlijnendatabase.nl/richtlijn/angststoornissen/gegeneraliseerde_angststoornis_gas/farmacotherapie_bij_gas/enzodiazepine_gegeneraliseerde_angststoornis.html. (Accessed 18 Nov 2021)
83. Vinkers CH, Tijdink JK, Luykx JJ, Vis R. Kiezen voor de juiste benzodiazepine. Ned Tijdschr Geneeskd. 2012;156:A4900.
84. Singh H, Mhasawade V, Chunara R. Generalizability challenges of mortality risk prediction models: a retrospective analysis on a multi-center database. medRxiv (2021).
85. Baer T. Understand, manage, and prevent algorithmic bias. Berkeley, CA, USA: Apress; 2019.
86. Barocas S, Selbst AD. Big data's disparate impact. California Law Rev. 2016;104(3):671–732.
87. Bellamy RK, Dey K, Hind M, Hoffman SC, Houde S, Kannan K, Lohia P, Martino J, Mehta S, Mojsilović A, et al. Ai fairness 360: an extensible toolkit for detecting and mitigating algorithmic bias. IBM J Res Dev. 2019;63(4/5):4–1.
88. Lang WW, Nakamura LI. A model of redlining. J Urban Econ. 1993;33(2):223–34.
89. Ellenberg JH. Selection bias in observational and experimental studies. Statistics in Med. 1994;13:557–567. Place: England.
90. Geneviève LD, Martani A, Shaw D, Elger BS, Wangmo T. Structural racism in precision medicine: Leaving no one behind. Bmc Med Ethics. 2020;21(1):17.
91. Blodgett SL, Barocas S, Daumé III H, Wallach H. Language (Technology) is power: a critical survey of "Bias" in NLP. In: Proceedings of the 58th annual meeting of the association for computational linguistics, (Online) Association for Computational Linguistics, 2020, p. 5454–76.
92. Spruit M, Verkleij S, de Schepper K, Scheepers F. Exploring language markers of mental health in psychiatric stories. Appl Sci. 2022;12(4).
93. Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A. A survey on bias and fairness in machine learning. ACM Comput Surv. 2021;54.
94. Hardt M, Price E, Price E, Srebro N. Equality of opportunity in supervised learning. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R. editors. Advances in neural information processing systems vol. 29, Curran Associates, Inc., 2016.
95. Saleiro P, Kuester B, Hinkson L, London J, Stevens A, Anisfeld A, Rodolfa KT, Ghani R. Aequitas: a bias and fairness audit toolkit, 2018.
96. Sogancioglu G, Kaya H. The effects of gender bias in word embeddings on depression prediction. In: Empowering communities: a participatory approach to AI for mental health, NeurIPS'22 Workshops, 2022.
97. Kamishima T, Akaho S, Asoh H, Sakuma J. Fairness-aware classifier with prejudice remover regularizer. In: Flach PA, De Bie T, Cristianini N, editors. Machine learning and knowledge discovery in databases. Springer, Berlin Heidelberg: Berlin, Heidelberg; 2012. p. 35–50.
98. Meng C, Trinh L, Xu N, Enouen J, Liu Y. Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. Sci Rep. 2022;12(1):1–28.
99. Bender EM, Friedman B. Data statements for natural language processing: toward mitigating system bias and enabling better science. In: Transactions of the association for computational linguistics, vol. 6; 2018. p. 587–604.
100. van Lent M, Fisher W, Mancuso M. An explainable artificial intelligence system for small-unit tactical behavior. In: Proceedings of the 16th conference on innovative applications of artifical intelligence, IAAI'04. AAAI Press; 2004. p. 900–7.
101. Miller T. Explanation in artificial intelligence: insights from the social sciences. Artif Intell. 2019;267:1–38.

102. Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L. Explaining explanations: an overview of interpretability of machine learning. In,. IEEE 5th international conference on data science and advanced analytics (DSAA). IEEE. 2018;2018:80–9.
103. Markus AF, Kors JA, Rijnbeek PR. The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies. J Biomed Informat. 2021;113: 103655.
104. Bhatt U, Xiang A, Sharma S, Weller A, Taly A, Jia Y, Ghosh J, Puri R, Moura JM, Eckersley P. Explainable machine learning in deployment. In: Proceedings of the 2020 conference on fairness, accountability, and transparency; 2020. p. 648–57.
105. Ahmad MA, Teredesai A, Eckert C. Interpretable machine learning in healthcare. In: 2018 IEEE international conference on healthcare informatics (ICHI); 2018. p. 447–7.
106. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE Access. 2018;6:52138–60.
107. Liu N, Huang X, Li J, Hu X. On interpretation of network embedding via taxonomy induction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '18. Association for Computing Machinery; 2018. p. 1812–20
108. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Adv Neural Inf Proc Syst. 2017;30
109. Rojas JC, Carey KA, Edelson DP, Venable LR, Howell MD, Churpek MM. Predicting intensive care unit readmission with machine learning using electronic health record data. Ann Am Thoracic Soc. 2018;15(7):846–53.
110. Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: Proceedings of the 34th international conference on machine learning, vol. 70, ICML'17, JMLR.org; 2017, p. 3145–53.
111. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision; 2017, p. 618–26.
112. Jain S, Wallace BC. Attention is not explanation. In: Proceedings of NAACL-HLT; 2019, pp. 3543–56.
113. Wiegreffe S, Pinter Y. Attention is not not explanation. 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); 2019. p. 11–20.
114. Ross AS, Hughes MC, Doshi-Velez F. Right for the right reasons: training differentiable models by constraining their explanations. In: Proceedings of the 26th international joint conference on artificial intelligence, IJCAI'17, AAAI Press; 2017, p. 2662–70.
115. Rajani NF, McCann B, Xiong C, Socher R. Explain yourself! leveraging language models for commonsense reasoning. In: Proceedings of the 57th annual meeting of the association for computational linguistics; 2019, p. 4932–42.
116. Serrano S, Smith NA. Is attention interpretable? In: Proceedings of the 57th annual meeting of the association for computational linguistics; 2019, p. 2931–51.
117. Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: A survey on methods and metrics. Electronics. 2019;8(8):832.
118. Lertvittayakumjorn P, Toni F. Human-grounded evaluations of explanation methods for text classification. 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); 2019, p. 5195–205.
119. Holzinger A, Biemann C, Pattichis CS, Kell DB. What do we need to build explainable ai systems for the medical domain? 2017. ArXiv preprint arXiv:1712.09923.
120. Cabitza F, Rasoini R, Gensini GF. Unintended consequences of machine learning in medicine. Jama. 2017;318(6):517–8.
121. Payrovnaziri SN, Chen Z, Rengifo-Moreno P, Miller T, Bian J, Chen JH, Liu X, He Z. Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. J Am Med Inf Assoc. 2020;27(7):1173–85.
122. Uddin MZ, Dysthe KK, Følstad A, Brandtzaeg PB. Deep learning for prediction of depressive symptoms in a large textual dataset. Neural Comput Appl. 2022;34(1):721–44.
123. Caicedo-Torres W, Gutierrez J. Iseeu2: Visually interpretable mortality prediction inside the icu using deep learning and free-text medical notes. Expert Syst Appl. 2022;202: 117190.
124. Hu S, Teng F, Huang L, Yan J, Zhang H. An explainable cnn approach for medical codes prediction from clinical text. BMC Med Inf Decis Mak. 2021;21(9):1–12.
125. Blanco A, Pérez A, Casillas A, Cobos D. Extracting cause of death from verbal autopsy with deep learning interpretable methods. IEEE J Biomed Health Inf. 2020;25(4):1315–25.
126. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Inf Fusion. 2020;58:82–115.
127. Zhou J, Gandomi AH, Chen F, Holzinger A. Evaluating the quality of machine learning explanations: a survey on methods and metrics. Electronics. 2021;10(5):593.