BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Assertion Detection in Electronic Health Records

Ivana Trajanovska
Matriculation-Number: 885222

A thesis presented for the degree of
Master of Science

Supervised by: Prof. Dr.-Ing. habil. Alexander Löser
Assistant Supervisor: Prof. Dr. Felix Gers

Beuth Hochschule für Technik, Berlin
Data Science, Fachbereich VI
20.08.2020

Abstract Digitalized health systems produce vast amounts of Electronic Health Records (EHRs). Going over such data manually is tedious and time consuming. To provide a high-quality patient care, clinicians need to have better and faster access to crucial information, which will be in a summarized and interpretable format. Assertion Detection is one way of extracting medical entities from EHRs and detecting their association with the patients. Therefore, the aim of this thesis is to provide an end-to-end solution to solve the task of Assertion Detection. We propose our two-step solution based on a Named Entity Recognition (NER) tool and a pretrained BERT model, BioBERT + Discharge summaries. We fine-tune and evaluate our model on the i2b2 dataset on assertion detection. Our baseline outperforms the current state-of-the-art solution based on BiLSTM and Attention. Furthermore we label an additional batch of Discharge Summaries, Radiology Reports, Nurse Reports and Physician Letters from the MIMIC-III dataset, and show that the model can be transferable to different types of EHRs.

Contents

Chapter 1

Introduction

> "*Society is aging and healthcare costs keep rising. By digitizing the system, health services can be provided at lower cost and higher quality.*"

This is what the authors of (Hehner et al., 2020) say. As a result of the increasing number of digital data, studies have proved able to effectively solve challenging problems in Natural Language Processing (NLP). Using NLP for medical texts summarization, as well as information extraction is of a great importance in handling millions of data. Such tasks highly contribute to improving patient analysis, but can be tedious when done manually. Clinicians frequently discuss previous findings among colleagues, in order to use them in future treatments. Having that kind of information extracted and easily accessible simplifies their overall analysis. Assertion Detection is one way of extracting medical entities from EHRs and detecting their association with the patients. Therefore, the aim of this thesis is to propose an end-to-end solution for identifying assertions in Electronic Health Records (EHRs) and contribute to better patient care.

To solve such text-related tasks in the medical domain, our solution is based on a model which is pretrained on medical texts and has an additional knowledge of medical concepts. In recent years, language models such as BERT (Devlin et al., 2018) are gradually fine-tuned to different domains, so that they can perform better on texts which are different from Wikipedia articles. This includes fine-tuning on corpora from the medical domain, such as EHRs which are medical free-texts, and do not always follow a specific pattern. Moreover, they consist of medical concepts, which are not found in general-purpose texts, that the model should be able to detect. In addition, these kinds of models can be further fine-tuned to a specific downstream task without requiring a huge amount of data.

1.1 Motivation

The benefits of Assertion Detection The clinical information described in narrative reports is difficult for humans to access for clinical, teaching, or research purposes (Perera et al., 2013). To provide a high-quality patient care, clinicians need to have better and faster access to crucial information, which will be in a summarized and interpretable format. Moreover, clinicians gather information from diverse data sources and then have to communicate the findings with colleagues in order to provide a better care (Feblowitz et al., 2011). It is nearly impossible to go over all existing narrative reports and it might still not pay off, as important findings could be missed. Methods such as assertion detection bring valuable contribution to the process of information extraction and Cohort Analysis, which is "Type of medical research used to investigate the causes of disease and to establish links between risk factors and health outcomes " (Legg, 2018). Using assertion detection

for such a purpose leads to improved patient care (Bejan et al., 2013; Chen, 2019; Bhatia et al., 2019).

Knowledge extraction First important step in solving this task is knowledge extraction, i.e extracting the entities that are considered relevant (in our case diseases). This can be done with Named Entity Recognition (NER), which is extracting entities or terms from texts, no matter their source. However, extracting the entity in and of itself is not enough to derive a conclusion about how it affects a certain patient. This is the case because the entity would almost always be affected by its surrounding context (Chapman et al., 2001; Chen, 2019). For example the entity "pneumonia" in "no signs of pneumonia" is negated and should be considered as an absent condition when processing the patient's record. However, other proposed solutions on identifying the class of a given entity in an EHR often focus on either negation only (Chapman et al., 2001; Khandelwal and Sawant, 2020), or negation and uncertainty (Sergeeva et al., 2019b; Peng et al., 2017) (in our case absent and possible). In this thesis, we focus on three classes: Present, Possible and Absent, as they are most common, and, bring valuable information when doing Cohort Analysis. This decision is further discussed and justified in Chapter 2.

Language Models In the last few years, we have seen an immense progress in the field of deep learning. This includes improving machine learning models in the NLP field, that can process language very well, known as language models (Alammar, 2019). The most popular language models are based on Transformers (Vaswani et al., 2017) and BERT (Devlin et al., 2018) is one of them. These Transformer-based architectures and transfer learning are one of the biggest breakthroughs of our time and were shown to be relevant in solving various tasks. BERT is currently the main power behind Google Search. Google believe that this is "the biggest leap forward in the past five years, and one of the biggest leaps forward in the history of Search" (Nayak, 2019). This topic will be discussed in more details in Chapter 3.
Assertions are an attribute of the medical problem concepts that are markedin the c

1.2 Problem Definition

Before doing assertion detection on the data extracted from EHR (e.g Discharge summaries, Radiology or Nursing reports etc.), the first step is to process those medical records and have the medical entities (e.g. diseases) labeled. Then, assertion detection can be defined as follows: Given an entity in a medical text, identify its asserted class from the context. The assertion detection challenge was first introduced in 2010 by the authors of (Uzuner et al., 2011) where the term assertion is defined as:

> "*Assertions are an attribute of the medical problem concepts that are marked in the concept extraction task.*"

The proposed classes are: 1. Present, 2. Absent, 3. Possible, 4. Conditional, 5. Hypothetical, and 6. Not associated with the patient. In our work we focus on the first three classes, Present; Absent; and Possible. We come to this decision after consulting a specialist with a specialization in Internal Medicine and Nephrology, who advises us on using those, because they are: "most common and helpful classes in the process of patient information extraction".

1.3 Purpose and Research Question

The goal of our research is to come to an end-to-end solution which includes not merely assertion detection, but also a NER processing pipeline. We rely on our chosen model to surpass the current state-of-the-art (SOTA) solutions in assertion and negation detection, as well as to find a way for the model to generalize on data from a different distribution, for example Abstracts from BioScope, or medical records different than Discharge Summaries. The research question focuses on the possibility of generalization to other EHR and it tackles the issue of achieving better results than the existing solutions.

1.4 Approach and Methodology

Data Analysis The first step in our research is getting familiar with the data. We first look at sample discharge summaries to become acquainted with their structure. Most of the time the discharge summaries follow some specific pattern. Furthermore, we analyze the MIMIC-III dataset (Johnson et al., 2019, 2016b), more specifically Discharge summaries, Radiology and Nursing reports, and Physician Letter. All except the discharge summaries differ to some extent and bring novelty to our evaluation data. Additionally, we label that data and follow the same rules defined by the authors of (Uzuner et al., 2011). At the end we process our data and transform them in a representative format. A detailed explanation of this part follows in Chapter 3.

Architecture The scope of this research includes a final classification model for assertion detection, as well as a suitable NER model which recognizes medical entities such as diseases. We choose SciSpacy (Neumann et al., 2019) as the tool for NER. Further elaboration of our decision follows in Chapter 3. As our language model, we choose BioBERT + Discharge Summaries (Alsentzer et al., 2019), a fine-tuned BioBERT (Lee et al., 2019) on the MIMIC-III discharge summaries. Later in this thesis in Section 3.4.2 we justify our decision on the language model

1.5 Hypotheses

Our approach to constructing the given solution emerges from formulating the following hypotheses:

1.5.1 Research Question 1

The chosen fine-tuned model BioBERT + Discharge Summaries should surpass the current state-of-the-art models

The authors of (Devlin et al., 2018) show that fine-tuning BERT exhibits improvement on downstream tasks with limited amounts of training datasets for fine-tuning, which is a crucial property for transfer learning (Uran et al., 2019). So, by using specialized word embeddings from the pretrained BioBERT + Discharge summaries we can expect them to perform better than their general-purpose counterpart. Furthermore, we will compare it to the current state-of-the-art model on assertion detection, which is a BiLSTM with Attention (Chen, 2019).

1.5.2 Research Question 2

The model can be transferred to the same task on datasets coming from different distributions

Previous work (Dalianis and Skeppstedt, 2010; Khandelwal and Sawant, 2020) has focused on negation detection in medical texts. In their studies, they include corpora that come from a different distribution, like the BioScope papers and abstracts. In our research we would also like to tackle this problem and show that the model is easily extendable to other texts containing medical entities. Furthermore, we rely on similar results when testing on medical texts other than discharge summaries, like radiology reports.

1.6 Scope and Limitations

We will begin by defining our limitations. To our best knowledge there is only one public dataset available - the i2b2 dataset. Numerous research papers, which address the negation and uncertainty problem, commonly use the BioScope dataset of negation and uncertainty (Vincze et al., 2008). Originally, this dataset was composed of free-texts (radiology reports), biological full papers and biological paper abstracts. However, the radiology reports are currently not available, so we are limited to using the papers and abstracts from this dataset. Even if all the data was to be publicly accessible, we would still be limited to two classes only - absent and possible.
Another related dataset for this task is NegPar - A parallel corpus annotated for negation (Liu et al., 2018). Accessing this corpus is not free of charge, therefore that restricts us from using it in our research.
Over the past years, MIMIC-III data has become very popular and extensively used for different tasks in NLP. For instance, it was applied to fine-tuning BioBERT (Alsentzer et al., 2019). We therefore include it in our research as a supplementary dataset, considering that it was built upon discharge summaries, radiology and nurse reports, and physician records amongst others.

1.7 Outline

In Chapter 2 we present studies previous to ours and describe their methods and data they use. Furthermore we give a brief historical overview of word representations and language models. We focus on the Transformer architecture, whose encoder is the building block of our baseline. In Chapter 3 we will talk more about the architecture details, and the stages of our final solution. Additionally we will define our problem in more details and how we will assess it. We talk about the data and its structure, as well as the methods to process them. Next, we introduce our annotation process and guideline and the metric we use to evaluate the annotators. Furthermore, we will justify our decision of the building blocks of our end-to-end solution. In Chapter 4 we will describe our experimental environment, the frameworks and tools we use to train the model, as well as the libraries we used in the data processing step. Additionally, we talk about the importance of doing Hyperparameter Optimization, as well as the hyperparameters we will optimize. Next, in Chapter 5 we will explain the outcomes of the experiments we carry out. We will compare those to the current state-of-the-art solutions. Furthermore we perform an error analysis to identify the reasons for the misclassified samples. Additionally, we set up a human baseline and report the results from the experiments. Finally, in Chapter 6 we discuss our future work.

1.8 Summary

In this section, we gave a brief introduction of the problem that we are going to assess, and which challenges motivated us to focus our research on assertion detection in medical texts.

We dedicated one section on introducing the approach and methodology that we chose for solving the problem. In the hypotheses section we stated our two hypotheses that are the starting point of our research. The first hypothesis focused on surpassing the current state-of-the-art solutions whereas in the second hypothesis we stated our expectations for the model to be able to generalize on other type of medical texts. Furthermore, we explained our limitations and how they influenced the definition of our scope. Finally, we gave a short outline of the following chapters of this thesis.

Chapter 2

Background and Related Work

In this chapter we are going to give a detailed overview of existing solutions prior to ours, an end-to-end solution based on a NER tool and BioBERT + Discharge summaries . We are going to refer to not only assertion detection research, but we will reflect on some of the most important foundations in the task of solving negation in EHR. Moreover, as our proposed solution is utilizing the current state-of-the-art (SOTA) system in NLP, we dedicate a section to Language Models.

2.1 Rule-based approach and its limitations

Clinical free-texts somewhat follow some basic structure (Sergeeva et al., 2019b). The authors of (Chapman et al., 2001) approach the negation detection problem in clinical corpora, and they first conducted an analysis of the available reports and concluded the following:

> "*The narrative reports are limited to a handful of semantic types, including ndings, diseases, tests, drugs, etc., which are most often noun phrases rather than verbs, clauses, or sentences.*"

Moreover, they argue that a simple model as theirs could easily solve the problem of negation detection, without using sophisticated linguistic methodologies. This is possible when focusing on small amount of data with a limited set recognizable patterns. But as all other rule-based systems, it will not be able to generalize to unseen data to which the handcrafted rules can not apply (Sergeeva et al., 2019b).

2.1.1 NegEx

In 2001 (Chapman et al., 2001) collected around 2060 discharge summaries and stated their hypothesis:

> "*A relatively simple algorithm could produce reasonably accurate results.*"

Their model, NegEx, is based on 35 extracted negation phrases. Given that the model has sensitivity of 0 to data that does not contain any of their selected negation phrases, this is again a showcase of the weaknesses of rule-based models.

2.2 Neural Network approach

During the past two decades there has been a major shift in the world of neural networks. Although they were long set aside and out of popularity, Dr. Yoshua Bengio and a group of

researchers decided that they should do something about that and focused on transitioning the old neural networks to the deep neural networks that are used today (Faggella, 2019).

Deep neural networks have been long accepted in the NLP community and widely used for solving different tasks. Therefore, this section is dedicated to solutions and achievements based on such networks.

2.2.1 Convolutional Neural Networks (CNN)

The Convolutional Neural Network was originally presented by the authors of (LeCun et al., 1998). Their main idea was to propose a neural network architecture for handwritten and machine-printed character recognition in the 1990's which they called LeNet-5. The main building blocks of this kind of networks are a convolutional and a pooling layer. The primary concept of these networks is that in the convolutional layer there are filters of different window sizes that go over the input and detect certain patterns. Its power comes from the reusability of those filters, thus achieving better results with less trainable parameters.

Although the original purpose of CNN architectures was solving computer vision related problems, it was not long before people started using them for solving NLP tasks. It is well suited for detecting spatial substructures, and due to its characteristic for feature detection reusability, it is powerful in solving many text processing tasks (Rao and McMahan, 2019)

The authors of another paper on negation and speculation detection (Qian et al., 2016) offer a CNN-based solution with probabilistic weighted average pooling to address speculation and negation scope detection in medical texts. Their solution is based on using constituency parsing trees and dependency paths between the cues and the entities. They use the BioScope dataset and outperform the models in their benchmark comparison in the task of speculation detection with a F1 of 85.75%.

2.2.2 Long Short Term Memory Networks (LSTM)

One of the most used architectures in the NLP domain are the LSTM networks (Hochreiter and Schmidhuber, 1997). Their long-time popularity in NLP comes from their sequence dependency characteristic, as texts come in a sequence form and are full of semantic dependencies between words (Ruder, 2019b).

A recent paper (Sergeeva et al., 2019a), which is based on LSTMs, offers a solution for negation detection only. Their model consists of several layers - three of which are embedding layers, one bi-directional LSTM layer and a final output layer. Their solution demonstrates, like they say, "*a promising performance on a publicly available corpus referred to as BioScope*". Nevertheless, they focus on solving the negation detection problem by utilizing one dataset only.

2.3 Studies on Assertion Detection

Bidirectional LSTM with Attention On the task of Assertion Detection, the authors of (Chen, 2019) outperform the then state-of-the-art systems on the Assertion task, with F1 of 0.95 on the Present class, 0.93 on the Absent class and 0.64 on the Possible class. They train their own embeddings on different corpora and achieve good results. They implement a Bidirectional LSTM with Attention.

Multi-Label Assertion Detection The authors of (Ambati et al., 2020) solve the task of multi-label assertion, and they focus on assigning multiple labels on one entity. However, they use texts other than i2b2, since this corpus does not have multiple label annotations.

2010 i2b2 challenge on Assertion Detection The authors of (de Bruijn et al., 2011) proposed the best model on the task of assertion detection in 2010. Their solution consists of large sparse binary vectors as word representation, and a SVMmulticlass model. Their overall F1 score is 93.62%.

Conditional Softmax Shared Decoder The authors of (Bhatia et al., 2019) proposed their Conditional Softmax Shared Decoder, which is based on sharing one decoder architecture for both entity extraction and assertion detection. However, in their study they focus on the Absent class only and achieve a F1 score of 90.5%.

2.4 Language Models

The fuel of NLP is natural language, represented as sequences of words and characters. Languages have many complexities that people take for granted. Words are created and used by humans, in a way that is understandable to us only. To a computer, they are only sequences of characters that do not have any inherent meaning. As a consequence, a need arises to represent words in numerical format (Zhang et al., 2016; Bengio et al., 2003).

One-hot encoding Since words are a form of nominal variables, one intuition is to encode them as discrete symbols. This concept is equivalent to *one-hot encoding*, a feature extraction method, where every word is represented as a sparse vector. The vector size is equivalent to the defined vocabulary and each index is pointing at a different word. It consists of all zeroes except one, the position of the word we want to encode. However, one-hot encoding is appropriate for categorical data where no relationship exists between categories, because every vector is orthogonal to all others, and there are no similarities between them (Zheng and Casari, 2018). Moreover, these vectors are huge in size, and yet so little expressive.

Word embeddings Word embedding, or as is known in computational linguistics, a distributional semantic model, roots from the early 1950s. It is often associated with Firth's famous saying (Firth, 1957) "*You shall know a word by the company it keeps*". (Zhang et al., 2016). The strength of word embeddings comes from their reflection to words as low-dimensional, continuous, dense vectors by taking into account how often words occur in similar contexts. Furthermore, word embeddings are able to encode semantic and syntactic similarity making words comparable to one-another in the high-dimensional vector space (Zhang et al., 2016; Sumbler, 2018). Word embeddings can be (Jurafsky and Martin, 2019):

- Count-based, such as Latent Semantic Analysis (LSA) (Deerwester et al., 1990) which are SVD decomposition over co-occurence matrix to reduce to lower-dimensional space
- Prediction-based, that predict the context of a word to then learn the low-dimensional word vector representation

Word2vec The problem with one-hot encodings is that there is no similarity between vectors, because all word vectors are orthogonal to each other. The intuition for solving this problem is based on finding an approach where all words will be represented as vectors, which encode their similarity. This was solved by the authors of (Mikolov et al., 2013) by building a simple model, whose purpose was to predict between every word and its context words. It takes words as inputs and produces a vector space model, built from distributed representations for every word of the vocabulary. They implement two algorithms: Skip-gram and CBOW.
Skip-Gram The intuition for the Skip-gram model is: Having an arbitrary word from a sentence, how likely is to find the other vocabulary words nearby the input word? Therefore, the input is a one-hot encoding of the chosen word, whereas the output is a probability distribution of all words in the vocabulary.
CBOW With this approach, the authors take a window of surrounding words and predict the likelihood for every word of the vocabulary to appear within the context.

Contextualized word representation The limitation of word embeddings such as word2vec is that they provide only one representation of the word, regardless of its context. Words can have considerable number of meanings, or syntactic behavior, and that should not be disregarded. To solve that problem, a new concept was represented, called contextualized word vectors. The idea behind it is that words will get different word representations, depending on their context (Manning, 2019).

Language-Model-Augmented Sequence Tagger (TagLM) Among the first approaches in creating contextualized word representations was introduced by the authors of TagLM (Peters et al., 2017), whose idea is based on a semi-supervised model (Manning, 2019). They gathered plenty unlabeled data, trained an embedding model, such as word2vec, but at the same time trained a BiLSTM[1] language model. At the top they placed a supervised model, such as a Part-Of-Speech (POS) tagger. At the end, instead of using the fixed word2vec embedding for a sample input only, they also used the output of the pretrained language model, then concatenated it with the hidden state from the supervised model, and got one kind of a context-dependent representation. They achieved best results on the CoNLL 2003 NER task, with a F1 of 91.93%, by 1.06% better than the then best model (Manning, 2019).

ELMo: Embeddings from Language Models Not so long after TagLM was published, a new model for contextual embeddings was introduced (Peters et al., 2018). They were aiming for something more compact that would be easier to train on less powerful hardware, therefore more obtainable for people. What they did differently among other things is that they use two BilSTM layers. Also, they use character CNNs to build word representations, thereby reducing the number of parameters that need to be stored. The most important feature they introduce is their concatenation of all hidden states within the language model stack, which showed an improvement in solving different tasks. With these improvements they achieved a bit better results on the the CoNLL 2003 NER task, with a F1 of 92.22%, by 0.29% better than TagLM. However, the popularity of this model came from its ability to solve any NLP task, and they proved it by surpassing by then the

[1] Bidirectional LSTM

SOTA solutions of tasks such SQuAD[2], SNLI[3], SRL[4] amongst others (Manning, 2019).

ULMfit: Universal Language Model Fine-Tuning for Text Classification The same year, 2018, the creators of ULMfit, Howard and Ruder published their paper (Howard and Ruder, 2018), introducing a same idea of transfer learning for any NLP task. Their architecture is built upon a deep language model, based on AWD-LSTM[5] (Merity et al., 2017), and trained on a big unsupervised corpus. The idea was to fine-tune the model on a specific domain, and then to implement a specific task layer. With this model, NLP got its big moment, and transfer learning was possible probably as well as in Computer Vision (Alammar, 2018b).

Attention The idea of attention, to some extent, is to mimic how humans pay visual attention to different parts of an image, or a sentence, in order to understand the sense of it. This concept was introduced by the authors of (Graves et al., 2014), that built the Neural Turing Machine (NTM). The model architecture consists of a neural network with an external memory. The memory in NTM is finite, so using attention helped them to store relevant information.
The idea of attention was soon a reaction to the then well-known Sequence-to-sequence (seq2seq) (Sutskever et al., 2014) models. The whole idea of seq2seq models was to solve problems that do not have an input of specific length, neither a specific length output. They transform an input sequence to another, output sequence, both of which have arbitrary lengths.
One example of such task is machine translation. A seq2seq model is built on an encoder-decoder architecture. The encoder concatenates the input sentence word-by-word in a contextualized vector, which is then used as an input to the decoder component. For both components, recurrent neural networks were used, such as LSTMs or GRUs (Alammar, 2018c). The biggest problem with this model was that the context vector at the end would eventually forget words at the beginning of a very long sequence.
The authors of (Bahdanau et al., 2015) introduced the concept of attention and attention weights in Neural Machine Translation (NMT), whose purpose was to perform a linear combination of the encoded input vectors, in this case all hidden states from the encoder, which are weighted by these attention weights. In other words, the attention layer consists of weights for all hidden states of the encoder, which are passed to the decoder, all together with the hidden states from the encoder. Furthermore, the decoder also consists of an attention vector, so that in every decoding time step it considers all received hidden states, where every state is strongly associated with a specific word from the input sequence. Next, the decoder gives each hidden state a score, which will help the model understand which states are less relevant to the current word (Alammar, 2018c).
There are various proposals for calculating the attention score. For example, the authors of (Bahdanau et al., 2015) propose the following calculation in Equation 2.1, where s_t is the hidden state of the decoder, h_i is the concatenated hidden state from the encoder, and both $\mathbf{v}_a$ and $\mathbf{W}_a$ are weight matrices that are learned by the alignment model. An alignment model assigns some score to each pair of input and output, based on how well they match. (Weng, 2018)

$$\text{score}\left(s_t, \boldsymbol{h}_i\right) = \mathbf{v}_a^{\top} \tanh\left(\mathbf{W}_a\left[\boldsymbol{s}_t; \boldsymbol{h}_i\right]\right) \tag{2.1}$$

[2]Stanford Question Answering Dataset, https://nlp.stanford.edu/pubs/rajpurkar2016squad.pdf
[3]Stanford Natural Language Inference, https://nlp.stanford.edu/projects/snli/
[4]Semantic Role Labeling https://www.aclweb.org/anthology/W12-4501.pdf
[5]ASGD Weight-Dropped LSTM

Other forms of attention were also introduced in other papers (Graves et al., 2014; Luong et al., 2015; Vaswani et al., 2017).

Transformers The main problem with training models such as UMLfit or ELMo was that it was not possible to train them faster, and build even bigger models, and the computation is time-dependent. That means that there is no possibility for parallel computation. Another motivation for getting rid of the RNNs, which seq2seq models relied on, was focusing on Attention, as it provides access to any state of the recurrent model. That led to the idea of the transformer architectures, introduced by the authors of (Vaswani et al., 2017). The model was built on a complex encoder and a complex decoder that work non-recurrently, in solving the task of machine translation. In this architecture, there is no concept of timestamp and all words in a sentence are being processed simultaneously.
Encoder Block At the beginning, input embeddings are fed to the network, these can be even some already pretrained embeddings. This way words are mapped into vectors. A single word can have different meanings depending on how it is used. To overcome this problem, they introduce positional encoders, which have information about the distances between words in a sequence. They use *sin* and *cos* functions in order to calculate the vectors. After this stage, the word encoders have a positional information (context) as well. These encoders are passed to the encoder block, first in the Multi-Head Attention layer, and then in the Feed Forward layer.
Decoder Block In the decoder block, the same original sentence is passed to an embedding layer, and a positional encoding is added as well. The encodings are passed to the decoder block which has a Masked Multi-Head Attention, a Multi-Head Attention and a Feed Forward layer. Next the attention vectors from the Masked Attention layer which uses all words from the input sentence, but only the previous words of the output sentence, together with the Attention vectors from the encoder block are passed to the second Attention layer, which determines how related each word is with the rest of the sentence. These are then forwarded to the Feed Forward layer, then another Feed Forward layer with the same size as the out vocabulary, after which follows a Softmax layer, predicting the next word.
Attention The attention function they use in the Attention layers is a Scaled Dot-Product as defined in Equation 2.2. First they define a Key (K), Value (V) and Query (Q) vectors for every word, which help to calculate its attention scores. However, in the Multi-Head Attention Layer, instead of vectors, they use matrices, which improve the performance of the model in a way that it can focus on different positions, because otherwise the vector could have too much attention on itself (Alammar, 2018a). In their paper (Vaswani et al., 2017) they justify the need of a Scaled Dot-Product, by introducing $\cancel{p}\frac{1}{\sqrt{d_k}}$, where d_k is the size of the vectors. They argue that for large values of d_k the dot product is larger in magnitude which results in the softmax function having small gradients.

$$\text{Attention } (Q, K, V) = \text{softmax}\left(\frac{QK^T}{\cancel{p}\sqrt{d_k}}\right)V \tag{2.2}$$

BERT (Bidirectional Encoder Representations from Transformers) An important feature of the Transformer model is that both its block, the encoder and the decoder, separately have some underlying understanding of language (Uszkoreit, 2017). Models like BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020) are built on this idea and they utilize only one part of the traditional Transformer. As they explain it in its name, BERT is built on the encoder block from the Transformer. In their original paper, the authors of (Devlin et al., 2018) published two versions of BERT,

BERT$_{BASE}$, built from 12 encoder blocks and BERT$_{LARGE}$ which is built on 24 encoder blocks. They use WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary, which means some words are even split in smaller chunks. The strength of this model comes from its bidirectional architecture, which in contrast to BiLSTMs uses the bidirectional feature simultaneously at every step. This, however, allows the model to see all words from both left and right, and therefore it will not make sense to predict the next word. Nevertheless, the authors of (Devlin et al., 2018) came up with two different training tasks.

- **Masked language model** is the first task that the authors of BERT adopt, which was first introduced in the early 1950s (Taylor, 1953). More specifically, this task includes masking 15% of the words, so when the model is evaluated, the cost function uses only those predictions. The masking process on the 15% is divided three ways: 80% of the words are replaced with the [MASK] token, 10% of the words are replaced by a random word, and the last 10% of the words are kept unchanged, as they say, "to bias the representation towards the actual observed word."

- **Two-sentence Tasks** The authors proposed this task to make the model better at handling different sentences. They setup a training set containing pairs of sentences, where 50% of the cases the second sentence was actually found to be after the first sentence in a text, and the rest of the cases, the second sentence was some random choice. They define it as follows: Given two sentences, sentence A and B, is B likely to appear right after A? In this task, a [CLS] token is added at the beginning of the first sentence and at the end of each sentence there is a [SEP] token so the model can distinguish between those two.

Therefore, BERT can be reused for different downstream tasks, with just adding a classification layer on top of it. There are many different fine-tuned versions of BERT, in different domains, such as SciBERT: A Pretrained Language Model for Scientific Text (Beltagy et al., 2019), ClinicalBERT (Alsentzer et al., 2019), BioBERT (Lee et al., 2019) amongst others.

2.5 Summary

In this section we talked about previous work on assertion detection, as well as negation detection tasks. We started by explaining the early approaches that implement rule-based models to solve these tasks, as well as their limitations. We mentioned several research studies which focus on attention and how we can relate to their scores. We briefly explained some neural network architectures, such as LSTMs and CNNs, as well as solutions on negation and speculation detection and we showed their achievements.
Furthermore, we talked about the natural language complexities and how they were overcome gradually by incremental solutions. We started explaining the beginning of word representations as one-hot encodings, sparse binary vectors, as well as the need for expressiveness and better feature representation. We introduced word embeddings as well as the initial solutions for creating similarity between words, like there is in natural language. We presented the early stage of word embeddings with the word2vec model. We talked about the weaknesses of those models and the need for context in word embeddings. We presented the contextualized word embeddings solution in a timeline, beginning with TagLM. We explained ELMo and ULMfit, models which were intended to solve different tasks, thus to be able to do transfer learning. We emphasized the Attention mechanism and how it helped improving results for long sequences. The Transformer model followed with its novel architecture, replacing the recurrent models with all words being processed

simultaneously. We talked about BERT, a model built on transformers and how it can be fine-tuned to other downstream tasks.

Chapter 3

Methodology

3.1 Problem Definition

Assertion detection is considered to be a significant and challenging task in clinical NLP (Chen, 2019). Many different studies are focused on detecting negated entities only (Bhatia et al., 2019; Khandelwal and Sawant, 2020). Most of them are not attempting to assert whether the disease might be present. Also, many of them first detect the negation in a sentence, and then identify which entities are affected by it (Bhatia et al., 2019; Sergeeva et al., 2019b; Khandelwal and Sawant, 2020). Others focus on detecting the entity first, and then identify its label from the context (Wang et al., 2016/04; Rumeng et al., 2018; Chen, 2019; Ambati et al., 2020). We adopt the second approach because of the following reasons:

- The model in (Chen, 2019) holds the highest scores in the task of assertion detection, and it is tested on all assertion classes, making it easier for us to do a direct comparison.
- The last step of this research is setting up an API. There will be options to test the endpoint on raw medical texts only, as well as already NER annotated data. The architecture of the final solution will enable running tests on both annotated and raw data.

Therefore, we are building an end-to-end solution, which focuses on both information extraction and classification. Finally, we define our approach towards solving the task of assertion detection:

> Given an entity in a medical text, identify its asserted class from the context.

This implies that our solution will be separated in two steps:

1. Given a paragraph, detect the entities it contains.
   e.g. She showed signs of pneumonia, but has no pain →pneumonia, pain

   For this purpose a Named Entity Recognition model will be used
2. Having the entities marked in a paragraph, identify their asserted class.
   e.g. She showed signs of pneumonia, but no [UNUSED1] pain [UNUSED1] →ABSENT

   For this purpose a classification model will be used.

Our final model should be an end-to-end solution, along with an interface that will yield the assigned class to each extracted entity, as shown on the figure below Figure 3.1.

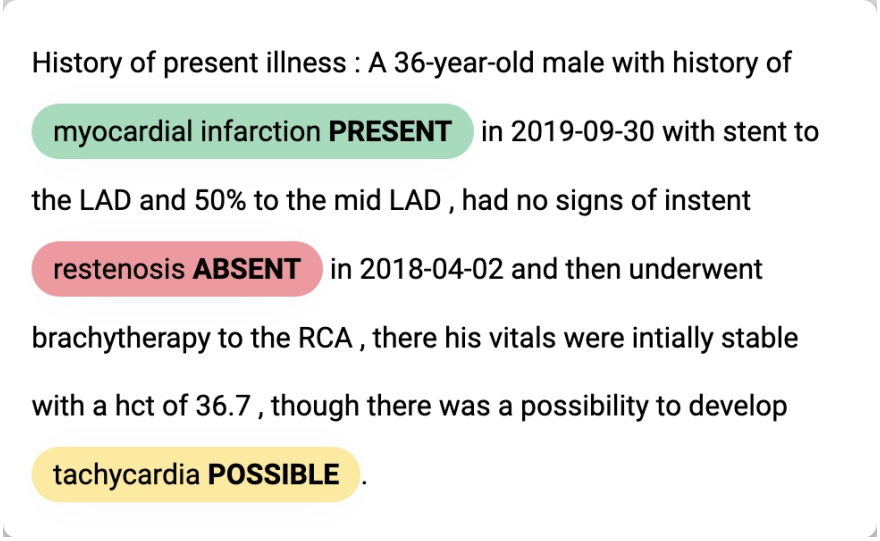

Figure 3.1: Example of the final model output. All medical entities (diseases) are extracted and labeled as Present, Absent or Possible.

3.1.1 Named Entity Recognition (NER)

The term "Named Entity" was first introduced by the authors of (Grishman and Sundheim, 1996) in the Sixth Message Understanding Conference (MUC-6), which until that point of time was still focusing on the task of Information Extraction (IE). The idea of IE was to extract knowledge from raw texts. The concept of NER came only naturally after they recognized the need to define name units such as people, organization and location names. This identification of entities was introduced as one of the important sub-tasks of information extraction that seeks to locate and classify named entities. Ever since, numerous studies tried to solve the task of Named Entity Recognition, while working on different shared tasks. For this purpose, both supervised and unsupervised methods have been used, with the latter trying to solve the limitations of the former (Ghiasvand and Kate, 2018).

NER Techniques There are multiple techniques when solving the problem of named entity recognition (Roldós and Wolff, 2020). The most popular among all are:

- **Lexicon Approach** The idea of the lexicon approach is that it is based on an already defined ontology which will be used as a static source of already existing, labeled entities. The NER task is then executed in such a manner that the model seeks for exact entities in the raw text data. The problem with this method is that it will not identify unseen entities (entities that do not exist in the ontology).
- **Rule-Based Systems** Solving the NER task while using Rule-Based methods is focusing on defining a limited set of grammatical rules which would later be applied to raw texts in order to extract the entities of interest. The main problem with Rule-Based systems is that it is not possible to generalize to unseen data, especially data from another domain. In order to extend the capacity of a Rule-Based System, one should manually add more rules to it.
- **Machine Learning-Based Systems** Machine learning models are another alternative to other methods, as a solution for better generalization. The only drawback of these models is, to get them running and yield good results, one should have a

large amount of already labeled data.

– **Hybrid Approach** A higher-level precision model would be a mixture of both a Machine Learning (ML) model and a Rule-Based System. The ML model will be trained on a set of labeled data, and then fine-tuned with a series of hand-crafted rules. These types of models produce a better precision (Roldós and Wolff, 2020).

CoNLL 2003 NER The first challenge was presented by the authors of the paper (Tjong Kim Sang and De Meulder, 2003), known as the CoNLL 2003 NER task. They limit the set of named entities to PERSON; LOCATION; and ORGANIZATION. The current SOTA solution achieved on this task is presented by the authors of (Baevski et al., 2019) with a F1 of 93.5%.

There are also other challenges on this topic which more or less focus on the same general entities, such as WNUT2017 Shared Task on Novel and Emerging Entity Recognition (Derczynski et al., 2017), or OntoNotes: A Large Training Corpus for Enhanced Processing (Weischedel et al., 2011).

Medical Named Entity Recognition Although the first epochs of solving the task of NER were mostly focused on general concepts, the necessity for applying the same idea on identifying medical concepts soon became crucial. When thinking of using supervised learning methods, the set of entities in the medical domain is distinctively different than the one in the general-domain NER. The alternate method is using an unsupervised learning strategy for that purpose, which relies on already defined dictionaries of known named entities, so the model could match its findings. Research has shown that applying the same unsupervised existing general-text NER methods on medical corpora is not straight forward as it would intuitively seem (Ghiasvand and Kate, 2018).

There are two arguments supporting this statement:

1. The general-domain entities usually don't have any linguistic variations, and not only that, but medical entities can sometimes be mentioned in several different ways.

2. Medical entities can sometimes contain multiple tokens, which can then also be separated into multiple entities, making these sub-entities falling into a different category.

There are numerous tasks on this topic. For instance, on ShARe/CLEF[1] and BC5CDR-disease[2], (Peng et al., 2019) have the current state-of-the-art solution with a F1 score of 79.2% and 86.8% respectively. On GENIA[3], the authors of (Li et al., 2020) have a F1 score of 83.8%. The authors of (Arnold et al., 2016) achieve a F1 of 93% on the CoNLL2003 Task, and F1 of 89% on the GENIA dataset, whereas the authors of (Neumann et al., 2019) state that they have competitive baselines for 5 of the 9 datasets they evaluate on.

3.1.2 Classification

We define the second step of this research as a classification problem. A simple classification problem is described as a mapping function which expects an already defined input, $\mathbf{X}$, in one or more dimensional space and maps it to a certain category, from a predefined set of possible values, $\mathbf{Y}$ (James et al., 2013). In neural networks, in order to be able

[1] Clinical free-text notes from the MIMIC-II database (Suominen et al., 2013)
[2] Data from BioCreative V chemical-disease relation task (Wei et al., 2016)
[3] An Annotated Research Abstract Corpus in Molecular Biology Domain (Ohta et al., 2002)

to predict among target classes, an activation function is set at the end of the model, in order to predict for the most probable class. In our solution, we use the Softmax activation function, which computes the probability distribution on all possible outputs. It produces an output which ranges between 0 and 1, where all probabilities sum up to 1. The calculation for every prediction is defined in Equation 3.1. This activation function is used in multi-class problems, where the predicted class is the one with highest probability (Nwankpa et al., 2018). Nevertheless, our solution strongly relies on an existing pretrained language model, which we will discuss in Section 3.4.2.

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3.1}$$

3.2 Data

In this section we are going to discuss the data that is the foundation of this thesis. The primary corpus that we use is the i2b2 dataset on assertion detection (Uzuner et al., 2011). Since the choice of labeled data is limited, we expand our scope, and label an additional batch of data from the MIMIC-III corpus (Johnson et al., 2016b). We are going to give a thorough overview of the data structure of each dataset, and a final overview of the expectations of this work.

3.2.1 2010 i2b2/VA challenge on assertions

In the scope of the 2010 i2b2/VA Workshop(Uzuner et al., 2011), three challenges were introduced, one of which is the 2010 i2b2/VA challenge on assertions. In this thesis, we are going to focus on the challenge of assertions only. This is the only corpus being used in the task of assertion detection because it is the only publicly available dataset on this particular task. The corpus was provided by the i2b2 tranSMART Foundation, a non-profit foundation developing an open-source community around the i2b2, tranSMART and OpenBEL translational research platforms. The data was gathered from Partners Healthcare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center.

Data Description

The data from this came in a free-text form as discharge summaries. Discharge summaries are "*clinical reports prepared by a health professional at the conclusion of a hospital stay or series of treatments*" (Kamalodeen, 2020). Although the discharge summaries come from different sources, they follow some common pattern. Data exploration showed that the information most relevant to this task are found in the following sections: *History of Present Illness, Past Medical History, Impression, Chief Complaint, Imaging studies, Findings, etc.*

Discharge summaries The following example is a sample text from one discharge summary. In order to retain the privacy of the patients and the data, the paragraph is altered.

> CHIEF COMPLAINT AND HISTORY OF PRESENT ILLNESS: Pt. 111 is a 45-year-old female with squamous cell carcinoma of the top of mouth (stage T2 N0) that was biopsied by her dentist. Pathology was reviewed revealing invasive cell carcinoma. The possibility of metastatic carcinoma could not be excluded. She presented on 2018-01-23 for resection. She was admitted on

2018-01-23 following her surgery. She underwent a joint procedure by Otolaryngology.

Labels The labels for the discharge summaries came in separate files - each label file referencing the original discharge summary. Each line in the label files represents a single label for a particular entity:

c="nad" 42:2 42:2 *jj* t="problem" *jj* a="absent"

c="bilateral dvt" 3:18 3:19 *jj* t="problem" *jj* a="present"

c="atherosclerotic" 59:13 59:13 *jj* t="problem" *jj* a="possible"

The "c" label references the entity, after which follows a **beginning line : beginning token location**, and an **ending line: ending token location**, as one entity can be mentioned many times in different contexts.

The "t" label identifies the entity's class, in our case, we only focus on "problem", in our case, a disease.

Finally, "a" is representing the entity's asserted class.

Class Definition and Distribution The entities in this corpus are distributed within six classes – present, absent, possible, hypothetical, conditional, and not associated with patient. We will give a brief definition of each class:

- Present – includes all problems that are present in a patient. It is the default class and is assigned when none of the other assertion classes fit.
- Absent – indicates that a specific medical problem doesn't exist in a patient, or a problem that a patient had, but no longer does.
- Possible – asserts that there is a possibility that a patient has a specific medical problem, but that there is an uncertainty to some extent.
- Conditional – a medical problem in such a context is asserted as present under certain conditions.
- Hypothetical – entities found in this particular context are considered as a possible condition that the patient may develop
- Not associated with patient – the mention of a medical problem is not considered to be associated with the patient, but with someone else, e.g. a person from her family.

In Table 3.1 we show the class distribution of all identified entities within the discharge summaries in the dataset. The class distribution is quite skewed, which means the data is imbalanced, with *Present* being the majority class.

Present	Absent	Possible	Hypothetical	Conditional	Not associated with patient
21064	6144	1418	1367	274	236

Table 3.1: Class distribution - 2010 i2b2/VA challenge on assertions

After a discussion of the necessity of each class in solving this task, a professional in the field advises us to limit our set of classes to *Present, Absent, and Possible*. His argument

is that for doing such an analysis on patient records, these three classes are most common, and, bring the most useful information from such records. Finally, our chosen subset of data looks as follows:

Present	Absent	Possible
21064	6144	1418

Table 3.2: Class distribution on Present, Absent and Possible - Subset of the original data from 2010 i2b2/VA challenge on assertions

3.2.2 MIMIC-III

MIMIC-III (Medical Information Mart for Intensive Care - III), a freely accessible critical care database containing anonymized health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. It consists of data associated with more than fifty-thousand hospital admissions for adult patients, and 7870 neonates admitted between 2001 and 2008 (Johnson et al., 2016b). This version is an update of the MIMIC-II database (Johnson et al., 2016a). The database consists of different types of reports, such as discharge summaries, radiology and nurse reports, and physician letters amongst others. This dataset has been used extensively and is quite popular in the medical society. A lot of research questions have been explored thanks to the information this dataset contains. For instance, the authors of (Bashar et al., 2019) try to detect atrial fibrillation in ICU patients. Another example is from the authors of (Yang et al., 2020) who try to predict mortality in patients with a sepsis-associated encephalopathy. Altogether, there is a significant number of research studies which tackle similar problems, and it shows that the authors of (Johnson et al., 2016b) made an important contribution to the community.

We include part of the annotated data in one of our experiments, to see if adding more samples to the minority class will improve the performance of the model.

3.2.3 BioScope

The BioScope corpus (Szarvas et al., 2008) is a freely available source for research on handling negation and uncertainty in biomedical texts. The corpus contains three forms of texts: medical free texts, biological full papers and biological scientific abstracts. In the area of negation detection it is the most widely used corpus with a current best F1 score of 97.87 (Britto and Khandelwal, 2020). The data labels consist of negation and speculation classes only, in our case absent and possible, and excludes the present class within the context.

Availability Numerous other studies try to improve on the task of negation detection based on this corpus. We also include it in our experiments. One important matter that we work on is the imbalanced distribution of the classes in the original i2b2 dataset. Therefore, by including the BioScope corpus in our training data we will have a less skewed class distribution. The main problem when requesting this corpus is that the medical free texts are no longer publicly available, hence we are limited on using the biological full papers and scientific abstracts.

Data Representation The data is available in an xml[4] format as shown in Listing 3.1. Each document is divided in multiple parts, and each part consists of several sentences. Each single sentence is annotated within the sentence tag, but not every one of those contains some label inside. The entities within the sentence are not explicitly labeled, but given are the scope of the negation/speculation cue, marked with a xscope tag, and the cue itself, placed within a cue tag.

```
<DocumentPart type="Text">
    <sentence id="S1.5">Since a number of microbial genomes have
        been completely sequenced to date, it is tempting to ask
        <xcope id="X1.5.1">
            <cue type="speculation" ref="X1.5.1">whether</cue>
            the 23rd amino acid is left undiscovered in these genomes
        </xcope>.
    </sentence>
</DocumentPart>
```

Listing 3.1: Example annotated sentence from the BioScope biological full papers

	Clinical	Full Paper	Abstract
Documents	1954	9	1273
Sentences	6383	2624	11872
Negation sentences	6.6%	13.76%	13.45%
Negation cues	871	404	1757
Speculation sentences	13.4%	22.29%	17.69%
Speculation cues	1137	783	2691

Table 3.3: Distribution of samples per document type & classes per document type - BioScope (Szarvas et al., 2008)

Data Distribution In Table 3.3 we provide a clear representation of the document types and the distribution between both classes in each source (Szarvas et al., 2008). The ratio of negation and speculation sentences within the texts is rather small, especially in the clinical summaries. In total there are around 5000 samples available from this corpus.

3.3 Data Annotation

One of the goals of this thesis is to include a novel corpus in the task of assertion detection. The authors of (Bhatia et al., 2019) have already done this by including their proprietary dataset and achieved good results on both i2b2 and their own corpus. We think that it is of great importance to have a shared dataset on a specific task, so that the results can be comparable. For this particular purpose, we decide to do manual annotation on a chosen subset from the MIMIC-III corpus.

[4]The Extensible Markup Language (XML) is a simple text-based format for representing structured information: documents, data, con guration etc.

3.3.1 Annotation Setup

In the process of data annotation, two individuals were involved. Before doing the annotation, the following was defined:

- Subset of the most representative medical records
- Annotation Guidelines, the agreement on how the data is going to be annotated
- A tool for data annotation
- The annotators

Subset of the most representative medical records Before doing manual annotation, we have to choose a meaningful subset of the MIMIC-III corpus that would bring significant variety to our data. This task can be solved by using Active Learning (Settles, 2012), but given the limited time and the scope of this thesis, using this technique was not feasible. Therefore we do a manual comparison on some of the chosen records and make this decision ourselves.
While going over the existing records, at the beginning we conclude that they follow some common pattern. The choice of records is done in two epochs. First, we decide for the Discharge Summaries, as our main idea is to bring more samples to the training set. Nevertheless, after analyzing other records such as Physician Letters, Radiology Reports, and Nurse Letters we annotate a second batch, containing records from all three sources. These are the main findings which we find challenging and consider as a reason for the additional annotations:

- Those clinical texts tend to have shorter, unstructured paragraphs, which can be hard to understand
- Many of the entities are often listed one after another and do not follow a specific syntactic structure

Annotation Guidelines In order to keep the consistency of the annotation rules defined by the authors of (Uzuner et al., 2011), we decide to use their very same Rule Book, defined as follows:

The entity for each sample is **bold**, and the class is obtained by the context around it.

1. Present: problems associated with the patient can be present. This is the default category for medical problems and it contains those that do not fit the definition of any of the other assertion category.
    - the wound was noted to be clean with **mild serous drainage**
    - history of **chest pain**
    - patient had a **stroke**
    - the patient experienced a drop in **hematocrit**
    - the patient has had **increasing weight gain**
    - He has **pneumonia**
2. Absent: the note asserts that the problem does not exist in the patient. This category also includes mentions where it is stated that the patient HAD a problem, but no longer does.

- patient denies **pain**
- no **fever**
- no history of **diabetes**
- No **pneumonia** was suspected
- History inconsistent with **stroke**
- his **dyspnea** resolved
- elevated **enzymes** resolved

3. Possible: the note asserts that the patient may have a problem, but there is uncertainty expressed in the note.

   - This is very likely to be an **asthma exacerbation**.
   - Doctors suspect an **infection of the lungs**.
   - The patient came in to rule out **pneumonia**.
   - Questionable / small chance of **pneumonia**.
   - **Pneumonia** is possible / probable
   - Suspicion of **pneumonia**
   - We are unable to determine whether she has **leukemia**.
   - It is possible / likely / thought / unlikely that she has **pneumonia**
   - We suspect this is not **pneumonia**
   - this is probably not **cancer**
   - **pneumonia** unlikely

Data Annotation tool We choose a data annotation tool which will preferably be free of charge. As a consequence, we choose Doccano (Nakayama et al., 2018), an open source annotation tool. This tool provides different kinds of features for text classification, sequence labeling and sequence to sequence tasks. One can create labeled data for NER, text summarization, sentiment analysis etc. We set it up on our DATEXIS[5] Kubernetes cluster, so that it is accessible for every annotator.

Annotators Our group of annotators consists of two people, a Software engineer and a Data Scientist, who have no experience in data annotation in the medical domain. Both of them annotate the data separately.

3.3.2 Annotation Evaluation

The crucial point in the process of data annotation is to have all annotators follow and agree on the provided annotation rules. For that purpose we select a suitable measure, the Inter-Annotator Agreement (IAA), a criterion of how well multiple annotators can agree on the same annotation for a certain category (Bruijn, 2020). This very evident description of the IAA is explaining everything we need to know about the measure itself:

> "*IAA is a measurement of how clear the annotation guidelines are, how uniformly the annotators understood it, and how reproducible the annotation task is.*"

[5] The research group Database Systems and Text-based Information Systems at Beuth University of Applied Sciences Berlin

As Accuracy and F1 score do not take into account the expected chance agreements, kind of agreements that are likely to occur when people annotate data, but are not happening as a result of the defined annotation guidelines. For that reason we choose the Cohen's kappa coefficient, a statistic which measures the IAA for qualitative (categorical) items (McHugh, 2012).
Cohen's kappa measures the agreement between two annotators who each classify N items into C mutually exclusive categories. The Cohen's kappa coefficient k is defined in Equation 3.2, where p_o is the actual observed agreement, and p_e represents chance agreement.

$$k = \frac{p_o \quad p_e}{1 \quad p_e} = 1 \quad \frac{1 \quad p_e}{1 \quad p_e} \tag{3.2}$$

Cohen's kappa can range from -1 to +1. In this case, 0 is the amount of agreement that can be expected from a random change, and 1 represents the perfect agreement between the annotators. This score was interpreted by Cohen, and he explains it as the level of agreement between the annotators, as well as what the score means towards the reliability of the data. These interpretations and thresholds are defined in Table 3.4. A a score between 0.8 and 0.9 is already a strong level of agreement, whereas a score higher than 0.9 is almost perfect. The Cohen's kappa coefficient from our annotation, based on ten shared documents between the annotators was 0.847, which according to Cohen is a strong level of agreement. This also means that about 64 - 81% of the data are reliable, but given that our annotators have never annotated such clinical texts, the score can differ if experienced people from the field did the annotation instead (Ng, 2017a).

Value of Kappa	Level of Agreement	% of Data that are Reliable
0 - 20	None	0 - 4%
.21 - .39	Minimal	4 - 15%
.40 - .59	Weak	15 - 35%
.60 - .79	Moderate	35 - 63%
.80 - .90	Strong	64 - 81%
Above .90	Almost Perfect	82 - 100%

Table 3.4: Cohen's kappa interpretation

3.3.3 Annotated data

In Table 3.5 we present the final number of samples we labeled. These samples are result from 50 labeled Discharge summaries, and 10 from every other document type.

Dataset	Present	Absent	Possible
Discharge Summaries	2613	980	250
Physician Letters	204	66	34
Nurse Letters	293	59	14
Radiology Reports	249	130	40

Table 3.5: Class distribution on Present, Absent and Possible after labeling a subset of the MIMIC-III dataset

3.4 Model Architecture and Pipeline Setup

In this section we will talk about the building blocks of our end-to-end solution. First we will explain the method we use to decide for the Named Entity Recognition tool. Furthermore we will inquire into BioBERT + Discharge Summaries, and will give a justification of our choice.

3.4.1 Named Entity Recognition tool

Given the two steps nature of this solution, as our first assignment we applied measurements to two different NER tools, in order to decide for the one which will perform better on the annotated data from the i2b2 dataset. We choose two NER models, TeXoo (Arnold et al., 2016) and scispaCy (Neumann et al., 2019). TeXoo has the highest F1 of 93% on the CoNLL2003 Task, and F1 of 89% on the GENIA dataset, whereas the authors of (Neumann et al., 2019) state that "Overall, we find that the scispaCy models are competitive baselines for 5 of the 9 datasets.".

Evaluation Metric To compare both models we choose the following metric, defined by the authors of (Cornolti et al., 2013), also used by (Ling et al., 2015) and (Arnold et al., 2016). The equations below are cited in a summarized from (Arnold et al., 2016). Let D be the set of annotated documents where the gold standard is defined as $G = fG_d \; j \; d \; 2 \; Dg$, and $N = jGj$ is the number of total examples. Each entity in G is defined with a start position b, and an end position e. From each model, we define a set of predicted entities $P = fP_d \; j \; d \; 2 \; Dg$. To compare the predicted values to the gold standard mentions, we introduce the following calculations in Equation 3.3 for tp, fp, tn and fn, explained later in Chapter 5, which calculate the number of correctly labeled samples, and the number of misclassified samples for each class.

$$\begin{aligned} tp_d &= jfp \; 2 \; P_d \, j \; 9g \; 2 \; G_d : m(p,g)gj \\ fp_d &= jfp \; 2 \; P_d \, j \; \nexists g \; 2 \; G_d : m(p,g)gj \\ tn_d &= jfp \; \cancel{2} \; P_d \, j \; \nexists g \; 2 \; G_d : m(p,g)gj \\ fn_d &= jfp \; 2 \; G_d \, j \; \nexists g \; 2 \; P_d : m(g,p)gj \end{aligned} \tag{3.3}$$

Where the method m is a calculation of the weak annotation match defined in Equation 3.4:

$$\begin{aligned} m : (p,g) \; \not\exists \; & (b_p \quad b_g \quad e_p)\_ \\ & (b_p \quad e_g \quad e_p)\_ \\ & (b_g \quad b_p \quad e_g)\_ \\ & (b_g \quad e_p \quad e_g) \end{aligned} \tag{3.4}$$

In the following equation, we present the measurements of micro-averaged precision ($Prec$), recall (Rec) and NER-style F1 score:

$$\begin{aligned} \text{Prec} &= \frac{\sum_{d2D} tp_d}{\sum_{d2D} (tp_d + fp_d)} \\ Rec &= \frac{\sum_{d2D} tp_d}{\sum_{d2D} (tp_d + fn_d)} \\ F1 &= \frac{2 \;\; \text{Prec} \;\; \text{Rec}}{\text{Prec} + \text{Rec}} \end{aligned} \tag{3.5}$$

Finally, we present the results of the NER-style F1 score after running both models on a subset of documents from the i2b2 dataset. The TeXoo model has an F1 score of 46%, whereas the score from running scispaCy is 69%, as shown on Table 3.6. Therefore we decide to use scispaCy as the NER tool for the purpose of this research.

TeXoo	scispaCy
0.46	**0.69**

Table 3.6: F1 scores on TeXoo and scispaCy, after evaluating both on the same portion of labeled data from the i2b2 dataset

3.4.2 Language Model

A few points are important in selecting our language model, which will be the base of our solution.

- It is preferably fine-tuned on medical texts
- It works well on thirty-thousand samples

In general BERT has been been found to outperform models such as ELMo or non-contextual embeddings on many tasks, including those from the medical domain (Alsentzer et al., 2019). Regarding the data size, in their original paper, the authors of (Devlin et al., 2018), perform some downstream tasks on BERT and show that it works well with datasets similar in size to the i2b2 corpus we have. We first look at BioBERT (Lee et al., 2019), trained on PubMed Abstracts and PMC Full-text articles, and ClinicalBERT (Alsentzer et al., 2019), where they also offer several variations, for example, they retrain BioBERT on MIMIC data. Furthermore, the authors of (Alsentzer et al., 2019) not only have a pretrained BERT model on medical data, but on the discharge summaries from the MIMIC-III corpus, which they call BioBERT + Discharge summaries. This is important as the model has already seen data from a similar distribution to the i2b2 dataset. In Chapter 5, we try out some initial experiments to justify our decision on the model we choose, which is BioBERT + Discharge summaries. The pretrained model we use is based on BioBERT$_{BASE}$, and supports sequences of maximum 512 tokens.

3.5 Summary

In this chapter we defined the problem we want to solve and our approach to solving it. We elaborated the choice of the concepts implemented in our end-to-end solution. We proposed a two-steps solution that will consist of a Named Entity Recognition (NER) task in its first step, and a classification model in the second step. The selection of the corpora, i2b2 dataset on assertion detection; BioScope; the different records from MIMIC-III, was clearly explained, as well as the need for bringing novelty in solving this task. We explained the annotation process of the MIMIC-III corpus. As our evaluation metric we used Cohen's kappa and we interpreted results of our annotation evaluation. Furthermore we explained the model architecture and the components that built it. We chose scispaCy to be our NER tool, as it outperformed TeXoo by 23%. As our language model we chose the pretrained BioBERT + Discharge summaries. In the following Chapter 4 we will explain the experimental environment, the tools we used, the data processing as well as the training during the downstream task.

Chapter 4

Implementation

In this chapter we will focus on explaining the environment of our experiment, the dependencies we used and the technologies our model is based on. Next, we will explain the process of Hyperparameter Optimization, and the methods for data processing.

4.1 Experimental Environment

In the experiment implementation, specifically the model implementation we first used PyTorch (Paszke et al., 2019) along with several pre-trained models from the Hugging Face library (Wolf et al., 2019). PyTorch is a Python package that provides two high-level features, tensor computation with strong GPU acceleration and deep neural networks built on a tape-based autograd system. Hugging Face is a library with over thousands of pretrained models in 100+ languages and deep interoperability between PyTorch & TensorFlow.

FARM FARM (Deepset-Ai) is a library that facilitates faster setup when working with Transformer models. One should only specify several important features in their configuration file, choose a compatible BERT model from Hugging Face and run the model. Their architecture is built from Adaptive Model, containing the Language Model and the Prediction Head(s), which is a simple classification or regression head, and a Data Processing part, made from a Processor and Data Silo. Their main features include:

- Simplified fine-tuning
- Speed
- Modular design of language models and prediction heads
- Combining prediction heads for multitask learning
- Support custom datasets with their Processor class
- Powerful experiment tracking & execution
- Checkpointing & Caching to resume training and reduce costs with spot instances
- Simple deployment

To see the progress of our model, the experiment tracking and execution feature was certainly helpful.

Cluster setup Models such as BERT are too large to be trained on a laptop or a working station. Therefore, we train the model on the DATEXIS[1] Kubernetes cluster. The cluster itself has the following gpus as part of its inventory: NVIDIA's Tesla - k80, p100 and v100. Given the amount of data we had, a k80 was a sufficient resource for us. All experiments were carried out reasonably quickly. In order to have the models running on the Kubernetes cluster, we create Docker images, which are pushed to the DATEXIS registry, and also Kubernetes configuration files to allocate the required resources and get the containers running.

4.2 Data Processing

We use the Pandas library (McKinney, 2010) to do the data processing. It provides fast, flexible, and expressive data structures making the data storing, analysis and querying easier. The data processing part somewhat differs for every corpus, which shows how different types of medical free-text corpora follow diverse patterns and writing styles.

i2b2 The data in this dataset is separated in label and text files. We begin the data processing by going over each line in the label files, which follow the structure defined in Section 3.2.1, and detect the original sentence within the text files. In order not to lose any relevant information about the section, or to identify ordered lists that usually point out that a disease is present, we use our own custom processing regular expressions, so that the paragraphs would be neither too long, nor too scarce. Our processing method results with a csv[2] format of the data, with entity-marked sentences and their respective label.

MIMIC-III At the beginning we prepare the data for the manual labeling and mark the diseases in the selected texts. Similarly to the i2b2 data, the labeled texts are processed in such manner that we get paragraphs that retain all useful information and have a sufficient input length (with number of tokens less than 512). We store the output data in csv files.

BioScope This dataset is available in a different format, as xml files. We process them with a xml parsing library. We first extract the sentences containing some kind of negation or speculation scope, then transform those into regular sentences (omitting all xml tags), and mark the already labeled entity. We then assign the annotated label to every sentence.

Input sequences There is a proposed input standard by the authors of (Devlin et al., 2018) for passing sequences into BERT. The input structure it task-dependent, and for the classification task it looks as follows: At the beginning of the sentence, the [CLS] (classification token) needs to be added, and then follows the rest of the sentence to be classified. The final hidden state from the model which corresponds to the [CLS] token is then used as an aggregate sequence representation for the classification task. However, our task is not really a sequence classification problem. It does start with a sequence as an input, but somewhere in the middle we have to mark the entity for which the classifier decides its predicted class. Considering that we did not find a similar problem nor a solution on how to treat such an input, we use one of the [UNUSEDxxx] BERT tokens to mark the entity in the sentence, based on a thread comment[3] from one of the contributors of (Devlin et al., 2018). These tokens, as he says, "*were not used and they*

[1] The research group Database Systems and Text-based Information Systems at Beuth University of Applied Sciences Berlin

[2] comma-separated value

[3] https://github.com/google-research/bert/issues/9#issuecomment-434796704

are e ectively randomly initialized", and allow you to treat words that are relevant only in your context. Therefore, at the end all samples followed this structure (the choice of [unused1] is arbitrary, as all unused tokens are randomly initialized):

[**CLS**] However, the test results were negative for [**unused1**] COVID-19 [**unused1**]

4.3 Fine-tuning and Hyperparameter Optimization (HPO)

As our next step we fine-tune our baseline on the i2b2 data on assertion detection, so it will perform better on that specific task (Devlin et al., 2018). We first define the classification model on top of BioBERT + Discharge summaries. As BERT is already a very complex model, the usual choice is implementing a simple feed forward layer on top of it. This is enough to show some initial results. For any further improvement we are implementing Hyperparameter Optimization (HPO).

HPO When training a machine learning model, there are parameters that the model will learn in this process, as for example weights and biases of a neural network, but there are also parameters that can't be learned during the training process (Zheng, 2015). Instead these have to be predefined when building the model architecture. These are known as hyperparameters. Parameters such as learning rate, optimizer, and batch size amongst others are considered to be hyperparameters. For that purpose, researchers need to explore different combinations of hyperparameters in order to achieve better results. HPO can be considered as the final step of model design and the first step of training a neural network (Yu and Zhu, 2020). There are several ways of doing a hyperparameter search. Grid Search, Random Search and Bayesian Optimization are the most popular among all others. Random Search is a good alternative to the Grid Search method, which performs an exhaustive searching through a manually specified subset of the hyperparameter space. This can often be time and resource consuming, and not always feasible. However, given the complexity of the task and the DATEXIS cluster resources, we choose Grid Search as a HPO method.

Automated hyperparameter optimization The process of automated HPO has the following benefits (Feurer and Hutter, 2019):

- It drastically reduces human effort
- It helps improving problem-specific machine learning algorithms
- It contributes to reproducibility of scientific research

As our HPO framework we choose Tune (Liaw et al., 2018), a Python library for experiment execution and hyperparameter tuning.

Hyperparameters We are optimizing on the following hyperparameters: learning rate (lr), training epochs, and batch size, which are proposed by the authors of the original BERT paper (Devlin et al., 2018).

4.4 Assertion detection app

As our final solution we set up an endpoint which can be accessed and tested on real examples. The expected input is any raw medical text that contains diseases. There is an additional (optional) field to insert NER annotations for the raw text, which can be used instead of ScispaCy. In Figure 4.1 we showcase the output from our end-to-end solution.

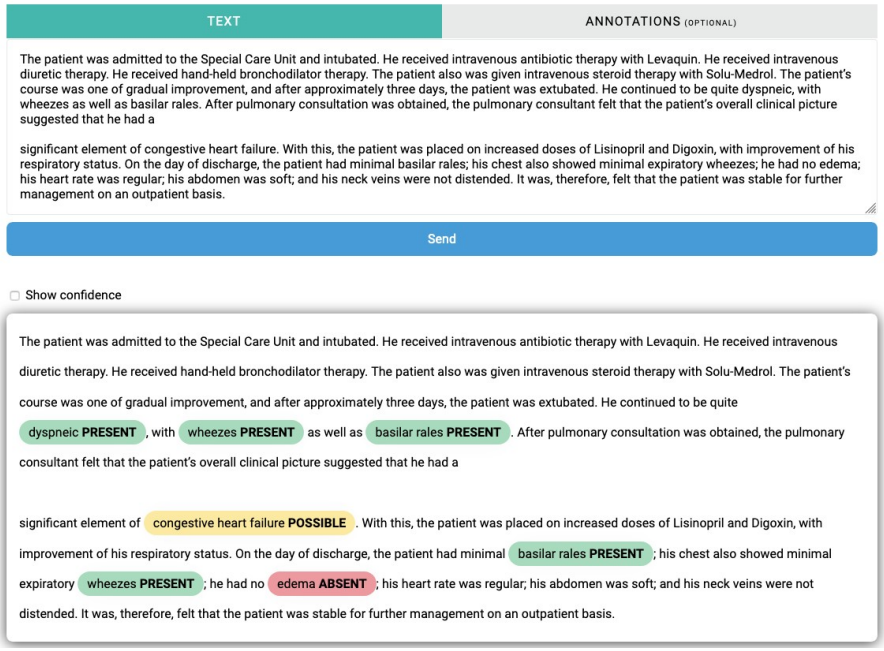

Figure 4.1: Overview of the assertion detection endpoint. It shows an example of a raw medical text, which is processed and labeled

4.5 Summary

In this section we first described our experimental environment. Furthermore, we elaborated on our decisions to use FARM to train the models and what are the benefits of the framework. We further elaborated on our data processing steps, the challenge of choosing the right input structure for our problem, and the decision of an entity-marking token. We talked about the importance of doing Hyperparameter Optimization (HPO) as well as the hyperparametrs we will optimize. At the end, we showcased the final version of our end-to-end solution.

Chapter 5

Evaluation

In the following section we first recall on our Hypotheses from Chapter 1. Next, a definition of the chosen evaluation metric will follow. Furthermore, we will elaborate the results and compare them to a human baseline. There we will explain the necessity of a human baseline and the possibility of improving the model by using other methods. Finally, we will demonstrate the results yielded from all different test datasets and conclude the outcome of our hypotheses. At the end we will have a discussion about the limitations, and suggest improvements on the existing model.

5.1 Hypotheses

Our primary hypothesis is expecting that our model, based on BioBERT + Discharge Summaries will surpass the current state-of-the-art models (Chen, 2019; Bhatia et al., 2019). The second hypothesis is focused on expectations for the model to be able to generalize on other types of medical texts.

5.2 Evaluation Metrics

When evaluating the outcomes of a classification task, one has to do a contingency table, or also known as a confusion matrix, as shown on Figure 5.1, which shows the number of correct and incorrect predicted samples, per class. These are called: true positive (TP) equivalent with hit; true negative (TN) equivalent with correct rejection; false positive (FP) equivalent with false alarm, Type I error; and false negative (FN) equivalent with miss, Type II error. (James et al., 2013)

Precision, defined in Equation 5.1, is a quantitative measure for: What proportion of positive identifications was actually correct? (James et al., 2013)

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

Whereas recall, defined in Equation 5.2, answers the following question: What proportion of actual positives was identified correctly? (James et al., 2013)

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

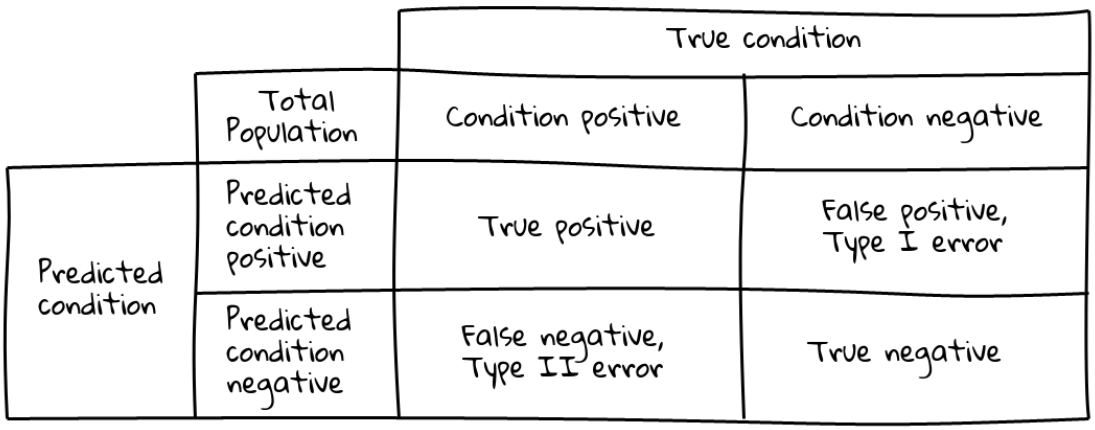

Figure 5.1: Confusion Matrix

F1, as defined in Equation 5.3, is a combined measure from both precision and recall, or more precisely, F1 score represents the balance between precision and recall (Ng, 2017b).

$$F1 = 2 \quad \frac{Precision \quad Recall}{Precision + Recall} \tag{5.3}$$

In classification, some of the most common metrics for model evaluation are recall, precision, and F1. In his book, Machine Learning Yearning (Ng, 2017b), Ng specifies that when working on a task such as classification, it is very important to have one metric when evaluating our models:

> "*Having a single-number evaluation metric such as accuracy allows you to sort all your models according to their performance on this metric, and quickly decide what is working best.*"

Following this practice we decide to use the F1 measure as our evaluation metric, more specifically the macro-average F1, as we are dealing with an imbalanced dataset and the macro-average considers all classes as equal when calculating the overall score, as it does not take into account the size behind the minority class scores (Zhang and Luo, 2018).

5.3 Results

A few remarks are important in this section:

- All initial experiments are done on the i2b2 data on assertion
- For any further improvements of the model, we use part of the MIMIC-III labeled data
- The transfer-learning tests are carried out on MIMIC-III and BioScope
- The classes are mapped into numbers where 0 replaces **Present**, 1 represents **Absent** and 2 is a replacement for **Possible**

First iteration of tests To prove that BioBERT + Discharge Summaries (Alsentzer et al., 2019) is a better fit than Clinical BERT (Alsentzer et al., 2019) in solving our problem, our first set of experiments includes running both models on the same set of data, the i2b2 corpus, using the same, arbitrary parameters in both trials.

We use the following set of parameters, stated in Table 5.1, which are some of the recommendations from the authors of (Devlin et al., 2018), when fine-tuning BERT.

Learning rate	Batch Size	Weighted CrossEntropyLoss
5e-5	16	True

Table 5.1: Initial set of training parameters

	precision	recall	f1 score	support
0	0.9893	0.9163	0.9514	1314
1	0.9438	0.9782	0.9607	412
2	0.5027	0.9029	0.6458	103
accuracy			0.9295	1829
macro avg	0.8119	0.9325	0.8526	1829
weighted avg	0.9517	0.9295	0.9363	1829

Listing 5.1: Clinical BERT results. Trained and tested on i2b2 data on assertion, using the parameters from Table 5.1

	precision	recall	f1 score	support
0	0.9807	0.9665	0.9736	1314
1	0.9924	0.9466	0.9689	412
2	0.6596	0.9029	0.7623	103
accuracy			0.9584	1829
macro avg	0.8775	0.9387	0.9016	1829
weighted avg	0.9652	0.9584	0.9606	1829

Listing 5.2: BioBERT + Discharge Summaries results. Trained and tested on i2b2 data on assertion, using the parameters from Table 5.1

As shown on Listing 5.1 and Listing 5.2, BioBERT + Discharge Summaries outperforms Clinical BERT, especially in the minority (Possible) class. Based on these outcomes, all further experiments use BioBERT + Discharge Summaries. Furthermore, from these sets of trials we notice that the dev and train errors are very close. There is no high variance in the model, and that helps us when doing Hyper Parameter Optimization, for example when deciding whether to use regularization techniques (Ng, 2017a).

5.3.1 Hyper Parameter Optimization (HPO)

We explained the necessity of going over the process of Hyper Parameter Optimization in Section 4.3. The Grid Search is based on the following parameters and respective values, shown in Table 5.2, as recommended by the authors of (Devlin et al., 2018). We use Early Stopping in order to overcome possible overfitting.

Learning Rate	1e-5, 2e-5, 5e-5
Batch Size	16, 32
Weighted CrossEntropy	True, False
Epochs	2, 3

Table 5.2: Set of Hyperparameters for HPO

First set of results From the Grid Search we get the best set of hyperparameters, as shown on Table 5.3. Our expectations were that using Weighted CrossEntropyLoss would help the model to handle the minority class better, but the results seem to differ.

Learning rate	Batch Size	Weighted CrossEntropyLoss	Epochs
1e-5	32	False	2

Table 5.3: Best set of parameters from Table 5.2 in HPO of BioBERT + Discharge Summaries

Finally, we show the best results from our second trial of tests. In Listing 5.3 we show the output from the test phase of the model, which outperforms the first BioBERT + Discharge summaries trial (results in Listing 5.2), trained with an arbitrary set of hyper parameters. We see a bigger improvement in the F1 of the minority (Possible) class, as well as a slight improvement in the other two classes.

```
              precision    recall  f1-score   support

           0     0.9877    0.9795    0.9836      1314
           1     0.9832    0.9927    0.9879       412
           2     0.8091    0.8641    0.8357       103
    accuracy                         0.9759      1829
   macro avg     0.9267    0.9454    0.9357      1829
weighted avg     0.9766    0.9759    0.9762      1829
```

Listing 5.3: Test results from the best trained model in the process of HPO

Again, the test results do not differ by much with those from the dev set, and we show that on Figure 5.2, where it is clearly demonstrated that at one point the model achieves a good result at the 420th batch, and does an early stopping at batch 690.

First step of improvements In Table 3.2 we show that the i2b2 dataset is imbalanced. The Present class covers almost 72% of the data, the Absent class represents around 22.5% and the Possible class covers only 5.5% of the data. It is a real challenge to achieve good results on all three classes. However, the results that BERT achieves are satisfactory in this clinical setting, as it handles imbalanced data pretty well, which also shown by the authors of (Tayyar Madabushi et al., 2019). To improve results on the minority class, we add more samples from the Possible class. In this regard, we take a subset (rows representing the Possible class) of our MIMIC-III Discharge Summaries labeled samples and add it to the training set. We use the same set of hyperparameters that outperformed all other in the HPO process. The training set is updated with 250 additional samples.

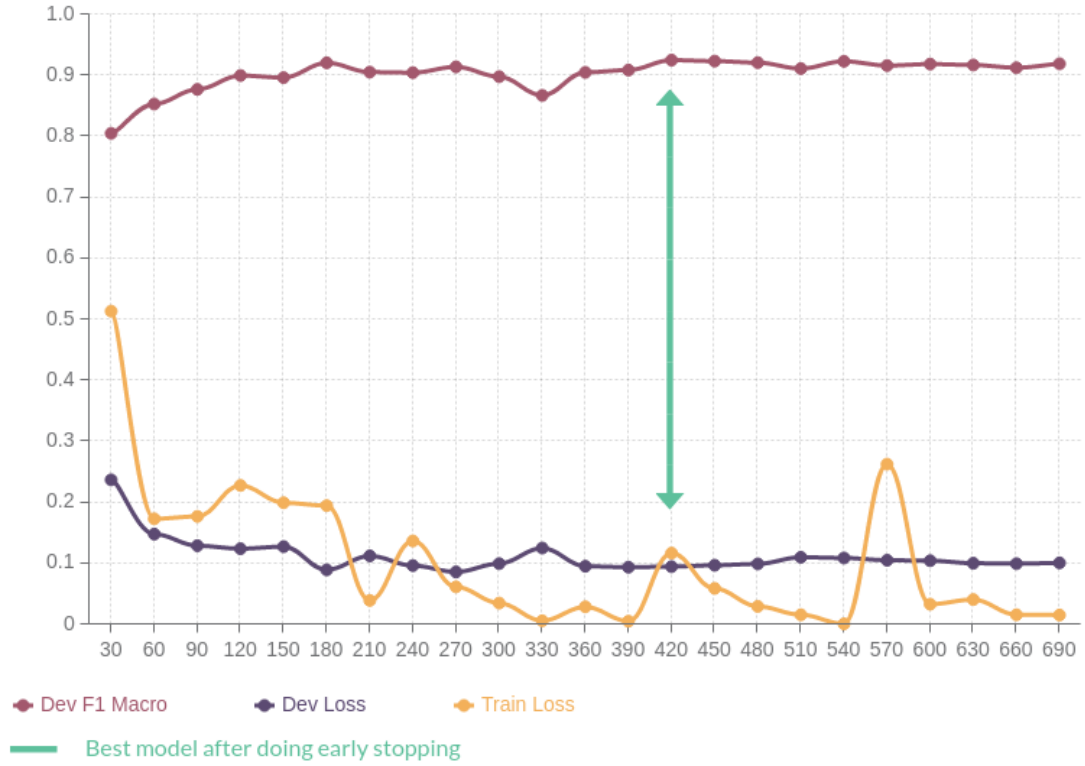

Figure 5.2: Model training with the best set of hyper parameters, shown on Table 5.3. On the X axis we represent the number of current (accumulative) batches, whereas the Y axis is shared between Train Loss, Dev Loss and Dev F1 Macro. At batch #690, the model does an early stopping and uses the best saved model, at batch #420

	precision	recall	f1 score	support
0	0.9855	0.9825	0.9840	1314
1	0.9926	0.9806	0.9866	412
2	0.7946	0.8641	0.8279	103
accuracy			0.9754	1829
macro avg	0.9243	0.9424	0.9328	1829
weighted avg	0.9764	0.9754	0.9758	1829

Listing 5.4: Test results after adding more data from the Possible class, gathered from the labeled MIMIC-III data, to the i2b2 training set

In Listing 5.4 we present the final test scores after we tried to improve the model. And in Table 5.4 we give a clear comparison of the F1 scores between both trials. Surprisingly the model did worse on the Possible class. We analyze this further in Section 5.5 and discuss why our expectations were not fulfilled. For any further experiments we will use the model trained on the i2b2 dataset only.

Class	F1 - i2b2	F1 - i2b2 + MIMIC-III
0	0.9836	**0.9840**
1	**0.9879**	0.9866
2	**0.8357**	0.8279

Table 5.4: Comparison of BioBERT + Discharge Summaries trained on the original i2b2 dataset, and after adding samples of the Possible class from a subset of our labeled MIMIC-III - Discharge Summaries

Comparison with current state-of-the-art models Next, we compare our model to the current state-of-the art solution as well as the best paper from the 2010 i2b2/VA task on assertion detection (Uzuner et al., 2011). We prove that BioBERT + Discharge summaries outperforms the so far best results, especially in the Absent and Possible case. Having that said, we confirm our first hypothesis, as defined in Section 1.5.1.

Model	F1		
	Present	Absent	Possible
Conditional Softmax Shared Decoder (Bhatia et al., 2019)	-	0.905	-
Bidirectional LSTM with Attention (Chen, 2019)	0.950	0.927	0.637
BioBERT + Discharge Summaries (ours)	**0.984**	**0.988**	**0.836**

Table 5.5: Test set performance (measuring F1) on 2010 i2b2/VA. Our baseline outperforms current state-of-the art solution, as well as the best paper from the challenge

5.4 Human Baseline

As the results from our model are already satisfactory in comparison to other models, and considering that we already tried out some minor improvement on the minority class, we set up another, human baseline, to see if there is room for improvement. The human baseline is a group of two people, tested separately, one of which is a Software Engineer, and the other, a Data Scientist.
In this regard we defined our labeling test set, which counts 20% of the original test set. The goal is to find the F1 score for each annotator individually, and to check their Cohen's Kappa score. Both annotators are already familiar with the Annotation Guideline from Section 3.3.1.

Cohen's Kappa This score is equal to 0.7382, which according to the Cohen's interpretation, this is a moderate level of agreement. The percentage of the data that is reliable is in the range of 35–63% (McHugh, 2012). Furthermore, we present the overlap between annotators, on each class separately. This results in 88% overlap on samples from the Present class, 75% overlap on Absent samples, and 72% overlap on samples labeled as Possible.

Annotators' test scores The annotators' separate F1 scores are evidently lower than the F1 scores that our Baseline yielded, as shown on Table 5.6. To understand the source of the large disparities we look further into the original i2b2 data. There are notable inconsistencies between the annotators there, as well as deviations from the Annotation Guideline. This will be looked into in more details in Section 5.5.

	F1 Score		
	Annotator 1	Annotator 2	Our Baseline
0	0.941	0.937	**0.98**
1	0.915	0.846	**0.993**
2	0.625	0.6	**0.717**

Table 5.6: F1 scores comparison of Annotator 1, Annotator 2 and our Baseline, BioBERT + Discharge Summaries, on 20% of the i2b2 test data

The right Human Baseline Nevertheless, this raises the question of what a proper Human Baseline is. According to Ng (Ng, 2017a), when evaluating the model, we try to reach the Bayes Optimal Error, which is the theoretical optimal level of performance, even better than a human performance. But in practice this is practically impossible, as there is almost always noise in the data, which will probably decrease the accuracy of the model. A human-level error is a proxy Bayes error, but yet not all human baselines will yield the same error. Therefore, in a medical sense, more types of human baselines can be defined:

(a) Typical human, an untrained person that does not work in the medical field

(b) Typical doctor, which is a less experienced doctor

(c) Experienced doctor

(d) Group of experienced doctors

All of them will gradually be slightly better than the previous person's performance, and also will be closer to the optimal Bayes error. In such scenarios it is expected that the group of experienced doctors will have the lowest possible error of all, by bringing their expertise and finding the right consensus. Therefore, we treat our annotators as the first type of human baseline, which is a typical human. We expect that a group of experienced doctors will outperform the scores in Table 5.6. This is also discussed later in the Chapter 6, as a part of our future work.

5.5 Error Analysis

In order to understand where the model goes wrong, we do a quantitative and qualitative error analysis. This is a common and preferred practice in many research studies, like the authors of (van Aken et al., 2018) that did a profound error analysis in their research and gave helpful insights about the data. In this error analysis we elaborate the weaknesses of the model which are highly important in a medical setting. In Figure 5.3 see a normalized version of the confusion matrix which depicts the False Negatives and False Positives, class by class. It becomes clear where the model does most of its mistakes, most of the false negatives are found in the intersection between Possible as true label, and Present as predicted label.
We are surprised to find out that most of the times the mistakes that the model did were because of irregularities in the labeled data. After doing a thorough analysis, sample by sample, we come to a general conclusion and divide our findings in the following categories:

– **Typos** Although not so common, there are a few misclassified samples that we believe are mistaken by the model because of mistyped key words such as **appeas** or **probalbe**. These kinds of errors are only around 5% of the analyzed data. In many cases, these words are decisive for the Possible class, which were correctly labeled by the annotators of the i2b2 dataset, but were confused by the model.

Figure 5.3: Confusion Matrix of the results previously presented in Listing 5.3. The values are normalized so it is easier to see which portion of the data was wrongly labeled by the model. We see that the diagonal is where the highest values are found. The biggest confusions that the model does are the Possible class False Negatives in respect to the Present class.

- **Wrongly labeled test samples** Another problem that we identify is that there are samples in the test set that are wrongly labeled. Words such as **likely** and **concerning for**, used to indicate uncertainty of a disease; or **not**, **no** and **resolved** always used as negation cues were found among samples labeled as Present, which clearly were not. Around 45% of the misclassified samples fall into this category.
- **Overall labeling inconsistency** Most of the misclassifications (around 35%) that the model makes are as a result of inconsistent labeling that we mostly found in the training data. There are clearly disagreements among the annotators, and there are examples to prove that. For example, words and phrases such as **appeared to be**, **concerning for** and **consistent with** were all labeled differently in the train data, sometimes as Present, and other times as Possible. For some of these misclassifications, the model also had a lower confidence in its decision due to the inconsistent labels in training time.
- **Model weaknesses** All other cases of misclassification (15%) can be attributed to the model itself. We recognize some examples that have longer dependencies that the model might have problem detecting as shown on the following example, where may is the cue indicating that the underlined entity should be marked as possible.

  Example: *May be either viral or secondary to resolving abdominal pain with resultant hematoma.*

  There are other cases as well. For example there were samples that did not consist of any phrase indicating Absent or Possible, but the model treated them as such. This is shown in the following example that the model considers as Absent.

  Example: *His hospital course was remarkable for ruling in for pneumonia.*

After analyzing a large portion of the data, our main conclusion is that:

- Although there was a general Annotation Guideline, annotators still had trouble distinguishing between classes when some exact samples were not found in it.

– The annotators responsible for the annotation of i2b2 dataset had more trouble than our annotators in deciding what *Possible* means. That resulted in confusing the model when predicting that class, and when adding our labeled data to the training set.

– We agree that most of the samples in the Present and Absent class follow some similar pattern, but it is not easy to come to a general agreement for the Possible class.

5.6 Evaluation on MIMIC-III

In Section 3.3.1 we spoke about which datasets we decided to annotate in order to see how will the model generalize on unseen data, other than i2b2. In this section test scores from all different labeled data will follow. Furthermore, we will discuss the results and do a short error analysis. We use our best model, trained on i2b2 only. We will give a short overview of each evaluation trial, and then give a general error analysis on all four datasets.

Discharge Summaries In Listing 5.5 we show the test results from evaluating the model on the labeled discharge summaries from MIMIC-III. The model is doing well on the larger classes, but performs really poorly on the Possible class. We do a further error analysis on these results in Section 5.6.1 to find out where did the model go wrong.

	precision	recall	f1 score	support
0	0.9528	0.9499	0.9513	2613
1	0.9148	0.9643	0.9389	980
2	0.7024	0.5760	0.6330	250
accuracy			0.9292	3843
macro avg	0.8567	0.8301	0.8411	3843
weighted avg	0.9268	0.9292	0.9274	3843

Listing 5.5: Test results from the MIMIC - III Discharge Summaries

Physician letters In Listing 5.6 we present the results from the evaluation on physician letters. The model has lower scores on all classes, especially on the Possible class. Although only 304 samples are labeled, the proportion of Possible samples is rather large. This gives us an inside view on our annotators' idea of that class, which is slightly broader when comparing it to the i2b2 annotators. The annotators found these letters most challenging as they are very unstructured, and consist of many listed entities in the middle of the document that are sometimes hard to follow.

	precision	recall	f1 score	support
0	0.8955	0.9657	0.9292	204
1	0.9062	0.8788	0.8923	66
2	0.8000	0.4706	0.5926	34
accuracy			0.8914	304
macro avg	0.8672	0.7717	0.8047	304
weighted avg	0.8871	0.8914	0.8836	304

Listing 5.6: Test results from the MIMIC - III Physician Letters

Nurse letters In general all types of medical texts were similar to some extent. However, we, as annotators found nurse letters to be most comparable to the discharge summaries, as they were better in structure and writing, and we find them second most convenient type of medical texts to annotate, which results in an easier annotation process. In Listing 5.7, we show some improvement on both Present and Absent class, but the model is still struggling on the Possible class. Again, there are more samples from that class, in comparison with the i2b2 data.

	precision	recall	f1 score	support
0	0.9757	0.9590	0.9673	293
1	0.8852	0.9153	0.9000	59
2	0.6471	0.7857	0.7097	14
accuracy			0.9454	366
macro avg	0.8360	0.8867	0.8590	366
weighted avg	0.9485	0.9454	0.9466	366

Listing 5.7: Test results from the MIMIC - III Nurse letters

Radiology reports The results on Radiology Reports as shown in Listing 5.8 were good on both Present and Absent class. However, the model shows weaknesses in predicting the Possible class.

	precision	recall	f1 score	support
0	0.9444	0.9558	0.9501	249
1	0.9921	0.9615	0.9766	130
2	0.6829	0.7000	0.6914	40
accuracy			0.9332	419
macro avg	0.8731	0.8725	0.8727	419
weighted avg	0.9343	0.9332	0.9336	419

Listing 5.8: Test results from the MIMIC - III Radiology Reports

5.6.1 General overview on errors - MIMIC-III

In this section we are going to elaborate our findings in the error analysis that we did on samples from MIMIC - III. We present a mutual analysis for all documents at once because the errors are common among all test results.

Again, we can divide our key findings in several categories:

- **Data processing** There are around 20% of the analyzed samples which the model got right, but were not meant to be classified as such. Those samples were originally part of a larger context, but important information was lost in the processing part and key words such as **no**, or **not** found were cut out of the context. We find this problem to be most common in the physician letters, which were also most challenging in the annotation process as they have very unstructured form. Therefore, it is a challenge to find a good processing function for such letters.
- **Annotators' mistakes** Around 55% of the misclassified samples are due to wrong labels. There are sentences which contain a question mark in front of the entities,

and our annotators considered those as typos. Many of the misclassfications are then because the annotators labeled those data wrong. There are also obvious wrong labels, such as **possibly consistent with**, which is clearly a sample from the Possible class, and we found it to be labeled as Present. We conclude that the majority of the problems come from the Possible class, which is now expected, because it was clear in the human baseline that the annotators struggled as well.

– **Labeling disagreements** Only around 7% of the misclassifications are found in samples where **unlikely** is the decisive word. Our annotators treat it as Possible, whereas in the i2b2 dataset we found it to be labeled as Absent.

– **Model Weaknesses** Nearly 18% of the mistakes are due to confusions that the model makes. We classify those as follows:

Key phrases One type of misclassification is when the model misses key phrases. For example, the model fails to identify **probably** when it was long before the entity, or examples that do not consist of cues indicating absence or possibility, but it tSreats it as such. The model also failed to miss **neither** as a negation. There were cases which we also find challenging such as the underlined entity in: *no hydrocephalus, subarachnoid hemorrhage, no fracture*. However, even in such cases, the model had lower confidence.

Cue following an entity Another case of misclassifications was when the decisive cue was found after the entity, such as: *His rash on the right hand was examined further and is now resolved* was treated as positive.

5.7 Evaluation on BioScope

The BioScope dataset consists of already labeled data which is available in a xml format. We do not do any further processing, just extraction of the cues and scopes, where the scopes are significantly long, and sometimes even the cue is within the whole scope. Therefore, the final structure of this dataset is quite different than the so far used medical corpora. We examine the data and find many samples whose class we are not sure of. A representative example is the following sentence, where the underlined part is the whole entity within the scope, and there is also the cue, in this case **predict**.

Example: *However, it can predict interacting protein pairs with a posterior odds ratio above 1.0 when used in combination with any single module in group A*

Nevertheless, as we do not find any guidance on how to process the data, nor have seen some other authors doing it, we decide to leave the samples as they are and use them in their original format.

Next, we run the experiment on the BioScope data, which consists of Absent and Possible samples and got the results presented in Listing 5.9. Other studies (Dalianis and Skeppstedt, 2010; Khandelwal and Sawant, 2020) managed to overcome the complexity of this dataset and had better results. The authors of (Khandelwal and Sawant, 2020) had an average F1 of 93.46% on the Absent (Negation) class. However, an important note is that they trained and tested their model on the BioScope data.

```
              precision    recall  f1-score   support
```

1	0.8645	0.8255	0.8446	1840
2	0.9617	0.4286	0.5930	2697
accuracy			0.5897	4538
macro avg	0.6090	0.7514	0.4796	4538
weighted avg	0.9221	0.5897	0.6949	4538

Listing 5.9: Test results from BioScope Abstracts and Papers

Unseen patterns The model is making mistakes (24% of all analyzed samples) on unseen speculation (Possible) cues such as **hypothesise** or **raises the question**, as well as on negation (Absent) cues such as **instead**, **presence or absence** and **cannot** amongst others.

Disagreement between annotators Around 10% of the analyzed samples fall in this category. There are samples whose key words are **apparent** and **assumed** which are labeled as Present in the i2b2 training data, but do not seem as such in the examples. We also found samples where **estimated** was the decisive word for the Present class, but were labeled as Possible.

Model weaknesses There are cases of known phrases that are missed by the model (56% of the analyzed samples), such as **may**, **would** and **was not**. However, usually a very long entity was found after such phrases, which was probably unexpected for the model. For example the following sequence is treated as an entity: *false positive rate of available computational and high-throughput experimental interaction datasets is as high as 90%.*

Not recognizing non-medical entities We tested some of the misclassifications which do not seem hard for the model. For example, we took the following sample from the BioScope dataset: *If the N- and C-terminal parts of an iORF have distinct but closely arranged BLAST hits in other genomes, it strongly suggests the iORF is actually two adjacent genes*, which the model predicted as present and replaced the underlined part with a disease. By replacing the original entity with the disease, the model predicted the corrected class. This happened with around 10% of the analyzed data, and gives us some insight about the model, that it is not easily transferable to data other than diseases.

Having the results from this dataset, our general conclusion is that the problem lies in both the complexity of the BioScope data, and the disparities between medical reports and scientific papers and abstracts. Regardless the different key phrases that are decisive for the predicted class, the entities in the BioScope dataset contain ten tokens on average, which might be challenging for the model. Another important finding is that the model can make different predictions if the entities are altered with diseases.

There are also similar findings from the authors of (Clark et al., 2011) that implement cue detection in their solution. For this purpose they used the annotated BioScope cues. However, while testing their model, they concluded that the cue detection task did not contribute that much to achieving better results as the BioScope data is significantly different than the i2b2 corpus.

5.8 Discussion

During the evaluation process we learned some essential insights, which will be helpful in our future research. At first we thought that the medical records have obvious patterns, so it was no surprise that many rule-based systems achieved good results. Nevertheless, the first iteration of error analysis already revealed some inconsistencies in the labels and we agreed that the data was not as simple as it may seemed before. We proved this by setting up our human baseline, where two annotators labeled 20% of the data, and were far from the baseline. Again, we proved that we were wrong regarding the simplicity of this data. Therefore, our main take-away notes from these experiments are:

Annotators agreement We showed how this point it most crucial when assembling data. We tried it for ourselves and realized that sometimes language can be perceived differently, therefore even for humans there needs to be more training examples in order to be able to generalize.

Model capability Regarding the MIMIC-III dataset, the model showed to be robust, because it achieved an overall good results on unseen data, which also came from another source, and were labeled by different annotators. Although when it came to generalizing on general texts, the BioScope corpus, it had trouble recognizing novel Absent and Possible phrases. Such problems were detected by others as well (Feblowitz et al., 2011), and therefore, we conclude they are due to the different text styles of the BioScope dataset.

Adding more data This experiment caused the model to be less accurate when predicting the Possible class, because of the disparities that our batch of annotated data brought. In general the model managed to achieve good results.

Possible improvement Another thing we learned from the error analysis is that the label inconsistencies were not at random (Ng, 2017a). One suggestion for solving such a problem is eliminating the samples that were wrongly labeled, but we did not consider this, as the size of the Possible class samples will be much smaller, and we already deal with a small proportion of the dataset. The biggest improvement of all is rethinking the labeling process and including more experts from the field as annotators and supervisors. This is discussed in more details in Chapter 6.

Model reliablity The most important words that can be said to a patient are "*You do not have cancer*"(Harvey, 2017). This means that, once it is said, the observer is as certain about it as feasibly possible. Nevertheless, like machine learning models, doctors can make mistakes as well, and making a mistake in such case can be catastrophic for patients. In order for our model to avoid such scenarios, it should maximize its True Positives (where positive refers to the Present class in this research) and be as certain as possible about the Absent and Possible predictions. That is measured with the Recall of the Positive class (in this research the Present class) and therefore should be maximized. Our model yields a **0.9795** Recall of the Present class which is the highest of all existing solutions. In our error analysis we found out that the model was making mistakes which were due to longer dependencies. These were not common, but such mistakes should not be allowed as the predictions may result in misleading analysis and wrong interpretations. Nevertheless, these examples are only a few and will be reviewed in our future steps. Also, the high Recall (**0.9927**) of the Absent class also shows the capability of the model not to mistake such samples for another class. This is crucial in comparing results to similar patients, as the model will provide more precise information in the process of deciding for common treatments. In general, the overall F1 score of our model (0.9357), shows that it

outperforms current state-of-the-art solutions, and it can contribute more in the Cohort Analysis. Finally, here are some techniques that will pursue clinical professionals to trust our model (McCaw, 2019):

- **Testing** We use millions of drugs, whose biochemical effects are not fully understood, but are still accepted because they passed some randomized clinical trials and received FDA approval. Similarly, we should allow our model to undergo rigorous tests in order to gain more trust. This can be done once the Annotation Guideline is approved by experts so that appropriate tests can be constructed.

- **Boundary conditions** This method is based on specifying a set of boundary conditions and rules which the data should fulfill in order to assure that the model will be consistent and certain about its predictions. In our case these rules include reduced inconsistencies in the annotated data, and constructing a strong definition of what an instance from the Possible class should look like.

- **Explainability** For most of the misclassifications made by the model we state in our error analysis that those mistakes are due to inconsistent labeling. Moreover, adding our additional labeled data to the original i2b2 dataset showed that our definition of the Possible class somewhat differs from the i2b2 annotators. Again, as long as this problem exists among humans, it is only acceptable for the model incapability to generalize on that class. We elaborate the currently inexplicable misconceptions of the model in Chapter 6.

5.9 Summary

We began this chapter with a recall on our two hypotheses. First, we did our first trial of experiments, to compare Clinical BERT and BioBERT + Discharge Summaries on same set of parameters to show that BioBERT + Discharge Summaries is more suitable for this task. Next, we did a Hyperparameter Optimization and chose the best model. We showed that our Baseline outperforms the current state-of-the-art solutions. It achieved an overall macro F1 score of 0.9357. The Present class had a F1 of 0.9836, the Absent class had a F1 score of 0.9879, whereas the Possible class resulted in a F1 of 0.8357. We determined the weakness of the model, which results in lower scores on the Possible (minority) class. We trained the model again by adding more labeled data of the MIMIC-III corpus. However, we did not see any improvement. Next, we did an experiment with a Human Baseline, where two annotators were tested on 20% of the test data. The annotators had low level of agreement, as well as lower scores than our baseline. Next, we tested the model on the MIMIC-III data and the BioScope corpus. It showed good results on the MIMIC-III data, but lower performance on the BioScope set. From our error analysis we concluded that most of the wrongly labeled samples are a consequence of the inconsistent labeling we detected in the i2b2 dataset. We also detected some typos, long dependencies that we assume the model was having trouble with, and the rest of the incorrectly labeled data were mistakes made by the model. We further elaborated on the explainability of our results and how applying tests and a clear annotation guideline will help the model to improve. We talked about setting boundary conditions in the future so that the model can be more certain about its decisions, and therefore can be trusted even more.

Chapter 6

Conclusion

6.1 Summary

The goal of this thesis was to present an end-to-end solution in solving the task of assertion detection. First, in our introduction we stated our motivation and the challenges we want to solve, which would help experts in doing Cohort Analysis and providing better patient care. We defined our problem as follows:

Given an entity in a medical text, identify its asserted class from the context.

The assertion classes that we focus on are *Present*, *Absent* and *Possible*. Next, we defined the building blocks of our solution, which are ScispaCy, a Named Entity Recognition (NER) tool, and BioBERT + Discharge Summaries, a fine-tuned language model which was trained on medical corpora and Discharge summaries. Next we stated our two hypotheses, one of which stated that we expect our solution to surpass the current state-of-the-art models, and the other was about our expectations that the model should be easy transferable to other medical texts.
In the Background and Related work chapter we talked about solutions previous to ours. We first mentioned that a lot of the solutions are focused on either Absent or Possible samples and many of them ignore all three classes that we focus on. We explained how rule-based models were widely used and what are their limitations. We continued with a brief history of word representations and language models. We also talked about the machine learning models based on different architectures, such as Convolutional Neural Networks (CNNs) and Long Short Term Memory Networks (LSTMs). We listed the solutions which are considered to tackle this problem at best and their scores as well.
Next, in the Methodology chapter we outlined the definition of the problem we want to solve and how we decided to approach it. We proposed a two-steps solution that consists of a Named Entity Recognition (NER) task in its first step, and a classification model in the second step. We defined the selection of the corpora that we trained and tested our model on: i2b2 dataset on assertion detection, BioScope, and clinical texts from the MIMIC-III corpus. We explained the need for new data and our decision to annotate part of the MIMIC corpus as well as the annotation process. Two annotators were labeling the data. Their level of agreement, measured by the Cohen's kappa score was 0.76.
Furthermore, we introduced our working environment, the cluster and the resources we used to run our experiments, we talked about FARM and Tune, the two frameworks we used to train and tune our models.
Finally, in the evaluation section we presented our final results. We first did a Hyperparameter Optimization and chose the best model. We showed that our Baseline outperforms the current state-of-the-art solutions. It achieves an overall macro F1 score of 0.9357. The Present class had a F1 of 0.9836, the Absent class had a F1 score of 0.9879, whereas the

Possible class resulted in a F1 of 0.8357. We noticed the weakness of the model on the minority, Possible, class and added more from our labeled data to it. Those were 250 samples from the Possible class of the MIMIC-III Discharge summaries. However, we did not see any improvement on the Possible class. Furthermore, we defined our human baseline, consisted of two different annotators and tested them on part of the i2b2 test data. Both annotators seemed to fail on the Possible class the most. This was an important insight as we realized that the Possible class is challenging to the annotators as well. Next, we tested the model on the MIMIC-III newly labeled data, as well as on the BioScope corpus. Our baseline showed good results on MIMIC-III, but had problems detecting certain patterns in the BioScope dataset.
We did an error analysis and concluded the following: Most of the misclassifications the model did were due to inconsistent labeling of the i2b2 dataset, which was also a problem for the authors of (Clark et al., 2011). There were many similar examples that were labeled differently, therefore the model had struggled when predicting for similar samples. We further elaborated on the explainability of our results and why should the model be trusted. We talked about setting boundary conditions in the future so the model can be more certain about its decisions.

6.2 Future work

Although BioBERT + Discharge summaries, which we fine-tuned for the purpose of the Assertion Detection task, showed good results and outperformed the current state-of-the-art solutions, there is still room for improvement, and the following points are what we consider to implement in the future.

Adding more layers The authors of (Liu, 2019) show that by adding an additional layer at the beginning of BERT, in their case an Interval Segment Layer which distinguishes between sentences in a document, can improve its overall performance. Our idea is to add an additional encoding layer which will emphasize key patterns and words that are decisive for the asserted class of the entities.

Syntactic dependency The Transformer architecture consists of a positional encoding layer at the beginning to compute the linear distance between words. However, the syntactic dependency in the language should not be omitted. Research shows that the attention heads in BERT track some kind of syntactic dependencies between tokens (Clark et al., 2019; Htut et al., 2019; Voita et al., 2019). However, as the authors of (Qian et al., 2016) showed, including constituency and dependency parsing as an additional encoding layer can improve the performance of the model, we hypothesize this can also be a helpful insight to BERT. This should be important as the model will have an additional information about the dependency between the entities and cues that are decisive for the entities' classes. That might help the model in identifying some long dependencies which it sometimes failed to do so.

Including experts We strongly encourage the inclusion of medical professionals in the annotation guideline and process as one of our main findings from this research is that the model cannot learn properly due to inconsistent labeled data. That way the model will be more confident in its weak predictions.

Interpretability We showed that we can explain most of the errors in our error analysis, which are due to inconsistent labeling. However, it is of great importance to be able to interpret the decisions that BERT makes and what happens within its layers. This has

been tackled in many studies (van Aken et al., 2019; Htut et al., 2019; Voita et al., 2019) and it has been shown that certain patterns can be found in those layers. As our next step, we will use LIT (Language Interpretability Tool) (Tenney et al., 2020) which is a visual, interactive model-understanding tool for NLP models. We expect to benefit from it to answer questions about the decisions of the model and its consistency.

Bibliography

J. Alammar. The illustrated transformer, Jun 2018a. URL https://jalammar.github.io/illustrated-transformer/.

J. Alammar. The illustrated bert, elmo, and co. (how nlp cracked transfer learning), Dec 2018b. URL https://jalammar.github.io/illustrated-bert/.

J. Alammar. Visualizing a neural machine translation model (mechanics of seq2seq models with attention), May 2018c. URL https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/.

J. Alammar. A visual guide to using bert for the first time, Nov 2019. URL https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/.

E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-1909. URL https://www.aclweb.org/anthology/W19-1909.

R. Ambati, A. Hanifi, R. Vunikili, P. Sharma, and D. Farri. Assertion detection in multi-label clinical text using scope localization, 05 2020.

S. Arnold, F. Gers, T. Kilias, and A. Löser. Robust named entity recognition in idiosyncratic domains. Aug 2016.

A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli. Cloze-driven pretraining of self-attention networks. 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)*, pages 5360–5369, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1539. URL https://www.aclweb.org/anthology/D19-1539.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

S. K. Bashar, E. Ding, D. Albuquerque, M. Winter, S. Binici, A. Walkey, D. Mcmanus, and K. Chon. Atrial fibrillation detection in icu patients: A pilot study on mimic iii data *. volume 2019, pages 298–301, 07 2019. doi: 10.1109/EMBC.2019.8856496.

C. A. Bejan, L. Vanderwende, F. Xia, and M. Yetisgen-Yildiz. Assertion modeling and its role in clinical phenotype identification. *Journal of Biomedical Informatics*, 46(1): 68–74, 2013. doi: 10.1016/j.jbi.2012.09.001.

I. Beltagy, K. Lo, and A. Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

Processing (EMNLP-IJCNLP), pages 3615–3620, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL https://www.aclweb.org/anthology/D19-1371.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, Mar. 2003. ISSN 1532-4435.

P. Bhatia, B. Celikkaya, and M. Khalilia. Joint entity extraction and assertion detection for clinical text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 954–959, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1091. URL https://www.aclweb.org/anthology/P19-1091.

B. Britto and A. Khandelwal. Resolving the scope of speculation and negation using transformer-based architectures, 01 2020.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. 2020.

L. d. Bruijn. Inter-annotator agreement (iaa), Jul 2020. URL https://towardsdatascience.com/inter-annotator-agreement-2f46c6d37bf3.

W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34 5:301–10, 2001.

L. Chen. Attention-based deep learning system for negation and assertion detection in clinical notes. *International Journal of Arti cial Intelligence and Applications*, 10:1–9, 2019.

C. Clark, J. S. Aberdeen, M. Coarr, D. Tresner-Kirsch, B. Wellner, A. Yeh, and L. Hirschman. Determining assertion status for medical problems in clinical records. 2011.

K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does bert look at? an analysis of bert's attention. *ArXiv*, abs/1906.04341, 2019.

M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 249–260, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488411. URL https://doi.org/10.1145/2488388.2488411.

H. Dalianis and M. Skeppstedt. Creating and evaluating a consensus for negated and speculative words in a Swedish clinical corpus. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 5–13, Uppsala, Sweden, July 2010. University of Antwerp. URL https://www.aclweb.org/anthology/W10-3102.

B. de Bruijn, C. Cherry, S. Kiritchenko, J. Martin, and X. Zhu. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association*, 18(5):557–562, sep 2011. doi: 10.1136/amiajnl-2011-000150. URL https://doi.org/10.1136%2Famiajnl-2011-000150.

Deepset-Ai. deepset-ai/farm. URL https://github.com/deepset-ai/FARM.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4418. URL https://www.aclweb.org/anthology/W17-4418.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

D. Faggella. The rise of neural networks and deep learning in our everyday lives - a conversation with yoshua bengio, Feb 2019. URL https://emerj.com/ai-podcast-interviews/the-rise-of-neural-networks-and-deep-learning-in-our-everyday-lives-a-conversation-with-yoshua-bengio/.

J. C. Feblowitz, A. Wright, H. Singh, L. Samal, and D. F. Sittig. Summarization of clinical information: A conceptual model. *J. of Biomedical Informatics*, 44(4):688–699, Aug. 2011. ISSN 1532-0464. doi: 10.1016/j.jbi.2011.03.008. URL https://doi.org/10.1016/j.jbi.2011.03.008.

M. Feurer and F. Hutter. Hyperparameter optimization. pages 3–38. Springer, 2019.

J. R. Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.

O. Ghiasvand and R. Kate. Learning for clinical named entity recognition without manual annotations. *Informatics in Medicine Unlocked*, 13, 10 2018. doi: 10.1016/j.imu.2018.10.011.

A. Graves, G. Wayne, and I. Danihelka. Neural turing machines, 2014. URL http://arxiv.org/abs/1410.5401. cite arxiv:1410.5401.

R. Grishman and B. Sundheim. Message understanding conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL https://www.aclweb.org/anthology/C96-1079.

D. H. Harvey. How data scientists can convince doctors that ai works, Oct 2017. URL https://www.healthcare.digital/single-post/2017/10/05/How-data-scientists-can-convince-doctors-that-AI-works.

S. Hehner, S. Biesdorf, and M. Möller. Digitizing healthcare–opportunities for germany, Mar 2020. URL https://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/digitizing-healthcare-opportunities-for-germany#.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9: 1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *ACL*. Association for Computational Linguistics, 2018. URL http://arxiv.org/abs/1801.06146.

P. M. Htut, J. Phang, S. Bordia, and S. R. Bowman. Do attention heads in bert track syntactic dependencies? *ArXiv*, abs/1911.12246, 2019.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL https://faculty.marshall.usc.edu/gareth-james/ISL/.

A. Johnson, T. Pollard, and R. Mark. Mimic-iii clinical database, Sep 2016a. URL https://physionet.org/content/mimiciii/1.4/.

A. Johnson, T. Pollard, and R. Mark. Mimic-iii clinical database demo (version 1.4), Apr 2019. URL http://dx.doi.org/10.13026/C2XW26.

A. E. W. Johnson, T. J. Pollard, L. Shen, L. wei H. Lehman, M. Feng, M. M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scienti c Data*, 3, 2016b.

D. Jurafsky and J. H. Martin. Speech and language processing, Oct 2019. URL https://web.stanford.edu/~jurafsky/slp3/.

S. Kamalodeen. How to write a discharge summary: Discharge letter, Jul 2020. URL https://geekymedics.com/how-to-write-a-discharge-summary/.

A. Khandelwal and S. Sawant. Negbert: A transfer learning approach for negation detection and scope resolution. *ArXiv*, abs/1911.04211, 2020.

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

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz682. URL https://doi.org/10.1093/bioinformatics/btz682.

T. Legg. Cohort study: Finding causes, examples, and limitations, Feb 2018. URL https://www.medicalnewstoday.com/articles/281703.

X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li. A unified mrc framework for named entity recognition. In *ACL*, 2020.

R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

X. Ling, S. Singh, and D. S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015. doi: 10.1162/tacl_a_00141. URL https://www.aclweb.org/anthology/Q15-1023.

Q. Liu, F. Fancellu, and B. Webber. NegPar: A parallel corpus annotated for negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1547.

Y. Liu. Fine-tune bert for extractive summarization. *ArXiv*, abs/1903.10318, 2019.

T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://www.aclweb.org/anthology/D15-1166.

C. Manning. Contextual representations, 2019. https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture13-contextual-representations.pdf.

B. McCaw. The Batch: Google AI Explains Itself, Neural Net Fights Bias, AI Demoralizes Champions, Solar Power Heats Up, Dec 2019. URL https://blog.deeplearning.ai/blog/google-ai-explains-itself-neural-net-fights-bias-ai-demoralizes-champions-solar-power-heats-up.

M. McHugh. Interrater reliability: The kappa statistic. *Biochemia medica : casopis Hrvatskoga drustva medicinskih biokemicara / HDMB*, 22:276–82, 10 2012. doi: 10.11613/BM.2012.031.

W. McKinney. Data structures for statistical computing in python. In S. van der Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

S. Merity, N. Keskar, and R. Socher. Regularizing and optimizing lstm language models. 08 2017.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi, and X. Liang. doccano: Text annotation tool for human, 2018. URL https://github.com/doccano/doccano. Software available from https://github.com/doccano/doccano.

P. Nayak. Understanding searches better than ever before, Oct 2019. URL https://www.blog.google/products/search/search-language-understanding-bert/.

M. Neumann, D. King, I. Beltagy, and W. Ammar. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5034. URL https://www.aclweb.org/anthology/W19-5034.

A. Ng. Structuring machine learning projects, August 2017a. URL https://www.coursera.org/learn/machine-learning-projects.

A. Ng. *Machine Learning Yearning*. Online Draft, 2017b. URL http://www.mlyearning.org/,/bib/ng/ng2017mlyearning/Ng_MLY01_13.pdf.

C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *ArXiv*, abs/1811.03378, 2018.

T. Ohta, Y. Tateisi, and J.-D. Kim. The genia corpus: an annotated research abstract corpus in molecular biology domain. pages 82–86, 01 2002.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Y. Peng, X. Wang, L. Lu, M. Bagheri, R. Summers, and Z. Lu. Negbio: a high-performance tool for negation and uncertainty detection in radiology reports. *AMIA Summits on Translational Science Proceedings*, 2017, 12 2017.

Y. Peng, S. Yan, and Z. Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5006. URL https://www.aclweb.org/anthology/W19-5006.

S. Perera, A. Sheth, K. Thirunarayan, S. Nair, and N. Shah. Challenges in understanding clinical notes: Why nlp engines fall short and where background knowledge can help. In *Proceedings of the 2013 International Workshop on Data Management & Analytics for Healthcare*, DARE '13, page 21–26, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324250. doi: 10.1145/2512410.2512427. URL https://doi.org/10.1145/2512410.2512427.

M. Peters, W. Ammar, C. Bhagavatula, and R. Power. Semi-supervised sequence tagging with bidirectional language models. 04 2017.

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

Z. Qian, P. Li, Q. Zhu, G. Zhou, Z. Luo, and W. Luo. Speculation and negation scope detection via convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 815–825, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1078. URL https://www.aclweb.org/anthology/D16-1078.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

D. Rao and B. McMahan, editors. *Natural Language Processing with PyTorch*. O'Reilly, 2019.

I. Roldós and R. Wolff. Named entity recognition: Concept, guide and tools, Apr 2020. URL https://monkeylearn.com/blog/named-entity-recognition/.

L. Rumeng, J. Abhyuday N, and Y. Hong. A hybrid neural network model for joint prediction of presence and period assertions of medical events in clinical notes, Apr 2018. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977733/.

E. Sergeeva, H. Zhu, P. Prinsen, and A. Tahmasebi. Negation scope detection in clinical notes and scientific abstracts: A feature-enriched lstm-based approach. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2019:212–221, 05 2019a.

E. Sergeeva, H. Zhu, A. Tahmasebi, and P. Szolovits. Neural token representations and negation and speculation scope detection in biomedical and general domain text. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 178–187, Hong Kong, Nov. 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-6221. URL https://www.aclweb.org/anthology/D19-6221.

B. Settles. *Active Learning*. Morgan & Claypool Publishers, 2012. ISBN 1608457257.

P. Sumbler. A brief history of word embeddings, Sep 2018. URL https://www.gavagai.io/text-analytics/a-brief-history-of-word-embeddings/.

H. Suominen, S. Salanterä, S. Velupillai, W. Chapman, G. Savova, N. Elhadad, S. Pradhan, B. South, D. Mowery, G. Jones, J. Leveling, L. Kelly, L. Goeuriot, D. Martinez, and G. Zuccon. Overview of the share/clef ehealth evaluation lab 2013. 01 2013. doi: 10.1007/978-3-642-40802-1_24.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

G. Szarvas, V. Vincze, R. Farkas, and J. Csirik. The bioscope corpus: Annotation for negation, uncertainty and their scope in biomedical texts. *BioNLP*, pages 38–45, 07 2008.

W. L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953. doi: 10.1177/107769905303000401. URL https://doi.org/10.1177/107769905303000401.

H. Tayyar Madabushi, E. Kochkina, and M. Castelle. Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5018. URL https://www.aclweb.org/anthology/D19-5018.

I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, and A. Yuan. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models, 2020.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://www.aclweb.org/anthology/W03-0419.

A. Uran, C. van Gemeren, R. Diepen, R. Chavarriaga, and J. d. R. Millan. Applying transfer learning to deep learned models for eeg analysis. 07 2019.

J. Uszkoreit. Transformer: A novel neural network architecture for language understanding, Aug 2017. URL https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html.

O. Uzuner, B. South, S. Shen, and S. DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA*, 18:552–6, 06 2011. doi: 10.1136/amiajnl-2011-000203.

B. van Aken, J. Risch, R. Krestel, and A. Löser. Challenges for toxic comment classification: An in-depth error analysis. 10 2018. doi: 10.18653/v1/W18-5105.

B. van Aken, B. Winter, A. Löser, and F. A. Gers. How does bert answer questions?: A layer-wise analysis of transformer representations. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9 (11):S9, Nov 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-S11-S9. URL https://doi.org/10.1186/1471-2105-9-S11-S9.

E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *ArXiv*, abs/1905.09418, 2019.

J. Wang, Y. Peng, B. Liu, Z. Wu, L. Deng, and T. Jiang. Extracting clinical entities and their assertions from chinese electronic medical records based on machine learning. In *Proceedings of the 2016 3rd International Conference on Materials Engineering, Manufacturing Technology and Control*, pages 1503–1508. Atlantis Press, 2016/04. ISBN 978-94-6252-173-5. doi: https://doi.org/10.2991/icmemtc-16.2016.290. URL https://doi.org/10.2991/icmemtc-16.2016.290.

C.-H. Wei, Y. Peng, R. Leaman, A. P. Davis, C. J. Mattingly, J. Li, T. C. Wiegers, and Z. Lu. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database*, 2016, 03 2016. ISSN 1758-0463. doi: 10.1093/database/baw032. URL https://doi.org/10.1093/database/baw032. baw032.

R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. *OntoNotes: A Large Training Corpus for Enhanced Processing*. 01 2011.

L. Weng. Attention? attention!, Jun 2018. URL https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. S. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016.

Y. Yang, S. Liang, J. Geng, Q. Wang, P. Wang, Y. Cao, R. Li, G. Gao, and l. Lihong. Development of a nomogram to predict 30-day mortality of patients with sepsis-associated encephalopathy: a retrospective cohort study. *Journal of Intensive Care*, 8, 12 2020. doi: 10.1186/s40560-020-00459-y.

T. Yu and H. Zhu. Hyper-parameter optimization: A review of algorithms and applications. *ArXiv*, abs/2003.05689, 2020.

Y. Zhang, M. M. Rahman, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. Nguyen, D. Xu, B. Wallace, and M. Lease. Neural information retrieval: A literature review. 11 2016.

Z. Zhang and L. Luo. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, Accepted, 10 2018. doi: 10.3233/SW-180338.

A. Zheng. *Evaluating Machine Learning Models: A Beginner's Guide to Key Concepts and Pitfalls*. O'Reilly Media, 2015. ISBN 9781491932469. URL `https://books.google.de/books?id=OFhauwEACAAJ`.

A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 1st edition, 2018. ISBN 1491953241.