Universiteit Leiden

# Master Computer Science

Clinical Temporal Relation Extraction: Towards a Patient's Timeline Creation


| | |
|---|---|
| Name: | Francesco Bovo |
| Student ID: | s2264951 |
| Date: | September 22, 2020 |
| Specialisation: | Data Science |
| 1st supervisor: | Dr. Suzan Verberne |
| 2nd supervisor: | Veysel Kocaman |


Master's Thesis in Computer Science


Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

**Abstract.** In the clinical Natural Language Processing (NLP) domain, Temporal Relation Extraction is a crucial task for understanding how events are ordered in a clinical text. Electronic Health Records (EHRs) contain valuable information regarding events that happened to a patient. The information in these medical records typically refers to events that happened in the past such as diseases, treatments and tests, as well as to present conditions of the patient. The ultimate goal of Temporal Relation Extraction in clinical narratives is to create a patient's clinical timeline and represent its detailed clinical history by discovering the link between temporal events and meaningful clinical named entities from the medical records. In this thesis, we implemented and evaluated an NLP pipeline that is able to pre-process clinical narratives, extract relevant named entities, namely problem, test and treatment entities, and relate them to specific temporal events. Specifically, in our experiments, for the Named Entity Recognition task we trained a Char – BiLSTM – CNN model, using BioWordVec embeddings, that was able to achieve an F1-score of 86.54% on the 2010 i2b2 NLP data set, attaining similar results to the state-of-the-art implementation by Zhang et al. [50]. The model considered for the Temporal Relation Extraction task, instead, was a ClinicalBERT pre-trained language model combined with a 1-d CNN for fine-tuning, based on Chen et al. [8] model architecture. Experiments on the merged 2012 i2b2 temporal relation extraction corpus revealed that the approach proposed attains an accuracy of 83.19% on the test set, proving that the model is able to effectively relate clinical entities to temporal events.

**Keywords:** Named Entity Recognition · Clinical Temporal Relation Extraction · Natural Language Processing.

# Acknowledgment

Firstly, I would like to thank my thesis supervisors Dr Suzan Verberne and Veysel Kocaman for giving me the chance to work on this interesting topic, providing incredible amounts of feedback and aiding me in general. Veysel came up with this extremely interesting thesis project and I'm happy he chose me to implement it in the first place, I'm grateful for the time and passion with which he supported me during this journey. Suzan encouraged me through the whole process with professionalism and kindness, always allowing me experimenting and taking up leadership. I'm thankful to her for believing in me even if this research area was rather new for me.
My thanks also go out to my peers at LIACS for their constructive criticism and for their help and feedback during my work there. Additionally, I would like to thank my family, Sirianna, Marino, Chiara and Filippo for supporting and trusting me with patience during my years of study. Lastly, I'm very grateful to the members of Tribe: Riley, Bianca, Laura and Sai for constantly believing in me and in my work.

# Table of Contents


1 Introduction . . . 3
Problem Statement & Thesis Structure . . . 4
2 Background . . . 5
2.1 Definitions and core concepts . . . 5
2.2 BERT . . . 6
Masked Language Model . . . 6
Next Sentence Prediction (NSP) . . . 7
BioBERT & ClinicalBERT . . . 8
2.3 Word Embeddings . . . 8
One-hot Representation . . . 8
TF-IDF Representation . . . 8
GloVe Representation . . . 9
Word2Vec Representation . . . 10
SparkNLP Library . . . 10
2.4 Related Work . . . 11
Clinical Named Entity Recognition . . . 11
Temporal Relation Extraction . . . 14
Context understanding models . . . 16
3 Methods . . . 17
3.1 Data sets . . . 18
Mimic-III. . . . 18
i2b2 2010. . . . 18
i2b2 2012. . . . 19
3.2 Approach . . . 20
Pre-processing . . . 20
Named Entity Recognition . . . 22
Temporal Relation Extraction . . . 23
Evaluation Method . . . 24
3.3 System Setup . . . 25
Distributed system setup . . . 25
Deployment and ML-Ops . . . 26
Hardware and Software . . . 26
4 Results . . . 26
4.1 Named Entity Recognition . . . 26
4.2 Temporal Relation Extraction . . . 30
5 Discussion . . . 33
6 Conclusion & Further Research . . . 37

**References** . . . 39
A Appendix - Python Information . . . 43
B Implementations availability . . . 43
C Further visualisation and results for both NER and TRE tasks . . . 43

# 1 Introduction

Clinical narratives from medical records contain valuable information regarding events that happened to a patient. However, most of these occurrences, such as the history of a patients illness or the effectiveness of a test for a particular disease are only significant when considered in a timeline that is relevant to the patient. Answering and interpreting questions such as "Is the treatment effective for patient X?" or "What is the progression of patient Ys illness?" can prove to be very challenging for a physician and might require further investigation. However, if we consider relative temporal relations between occurrences, solving these questions becomes much easier. Temporal Information Extraction used to process patient information in clinical narratives can positively contribute to the process of making accurate decisions in vital patient care tasks such as forecasting the effects of therapies, preventing the spread of a disease or diagnosing the nature of a medical condition [4]. In many cases, constructing a reliable and accurate timeline can facilitate physicians during the challenging decision-making process and help increase medical accuracy. In the hypothesis of a situation where, for instance, a patient with a chronic disease is taken to the ER, the physician, to be able to make an informed decision, would have to check over the history of the patient and manually verify their medical records to identify what type of treatment is most suited for this patient. Fundamental information can be hard to retrieve without any delay, but with a system able to extract crucial information and generate a detailed timeline this task can be facilitated.

Prior attempts in this research area led to promising results. In 2013, Nikfarjam et al. [33] implemented an hybrid system that applies machine learning together with a mechanism based on graph inference that was able to extract different components for separate types of temporal relations. More recently, in 2018, Leeuwenberg and Moens [27] proposed a novel archetype. Compared to earlier work where temporal relations needed to be predicted as an intermediate step to construct a timeline, the authors of this paper directly used the start and end-points predictions for events from the text.

Currently, Temporal Relation Extraction focuses on three main phases: (1) a named entity recognition phase in which events and their attributes are extracted as well as a temporal expression such as dates and durations, (2) a relation extraction phase which focuses on finding temporal links between the extracted entities. And (3) the timeline phases which concentrates in constructing a timeline from the extracted temporal links, if they are temporally consistent. In this thesis, the focus is in on building an NLP pipeline that would be able to pre-process narrative text from medical records, extract clinical entities and temporal expressions and finally find the relationship between these events. More specifically, the following contributions are made:

- Pre-processed and transformed raw clinical notes into training set following the CoNLL-2003 file format with an accuracy close to the licensed version in SparkNLP.
- Trained a Biomedical Named Entity Recognition model to extract `problem`, `treatment` and `test` entities from clinical narratives with an F1-score close to the current state-of-the-art implementation by Zhang et al. [50].
- Proposed a Temporal Relation Extraction model based on Chen et al. [8] BERT + 1-d CNN model architecture, to which different versions of BERT have

been implemented, namely BioBERT and ClinicalBERT pre-trained models that were able to achieve interesting results on the i2b2 2012 data set.

It is important to notice that Electronic Health Records (EHRs) incorporate different types of information, from physical assessments, admission notes, present complaints to physical examinations and discharge reports, among many. However, for this work, we only focused on the extraction of temporal relations from raw text fields in EHRs. Another important note is that both Biomedical Named Entity Recognition and Temporal Relation Extraction play a relevant role in this project since the first one is the tool used to extract both events and temporal expressions in a clinical setting while the second one is used to extract and classify the relationship between those entities. An exhaustive explanation of both these Information Extraction tasks is carefully unfolded in the following sections.

**Problem Statement & Thesis Structure** In this thesis, we will attempt to develop and evaluate an NLP pipeline that will be able to extract clinical events and relate them to specific temporal entities. Note, therefore, that this thesis focuses on the overall implementation of a data pipeline that starts with pre-processing raw data from clinical narratives, extracts specific medical entities and finally links them to temporal events. The thesis at hand, in particular, will attempt to answer the following questions:

1. What is the performance of a Named Entity Recognition method trained to specifically extract clinical entities such as treatments, tests and diseases from medical narratives?
2. To what extent is Temporal Relation Extraction able to efficiently categorise temporal relations between the extracted clinical entities? Specifically, to what degree are BERT methods more suitable for this task compared to other feature-based methods?

The layout of the thesis is as follows: Section 2 firstly introduces some of the preliminary terms and concepts that allow the reader to better understand the content. A detailed explanation of the model's architecture is given combined with a description of the word embeddings and libraries used in the implementation. The focus then passes onto related work regarding clinical named entity recognition, how temporal relation extraction has been implemented in the last few years and lastly some recent work on context understanding models and why recently they received large popularity.
Section 3 dives into the methods used in this thesis. An overview of the different data sets and approaches used together with an explanation of the evaluation metrics are given in detail. Additionally, the system setup with regards to the hardware and software are explained with also a deep focus on the distributed system setup used and future deployment of the models.
Section 4 presents the results of the multitude of experiments performed. For each type of experiment and each type of model, the time and metric results are conferred, summarised. Moreover, a deep dive into the experimental results is given by aggregating the different experimental results and discussing their practical implications and limitations, according to the evaluation criteria. With a focus on the limitations of the approaches proposed, Section 5 discusses the results of the experiments and summarise our findings.
Lastly, section 6 gives final remarks, presents shortcoming of the thesis as well as possible future research, and concludes the thesis.

# 2 Background

This section will cover the definitions related to text mining and, more generally, NLP, explaining more in detail the core concepts around Named Entity Recognition and text processing as well as text classification. A focus on recent achievements and related work in clinical NER, relation extraction and context understanding models will also be given.

## 2.1 Definitions and core concepts

Text mining is the process of parsing unstructured or semi-structured text and processing it to derive relevant information. One of the main objectives, therefore, of text mining is to extract meaningful information from text-based data by learning trends and patterns in the data. This task is pursued with the help of several statistical methodologies and models such as statistical language modelling or topic modelling, among many.

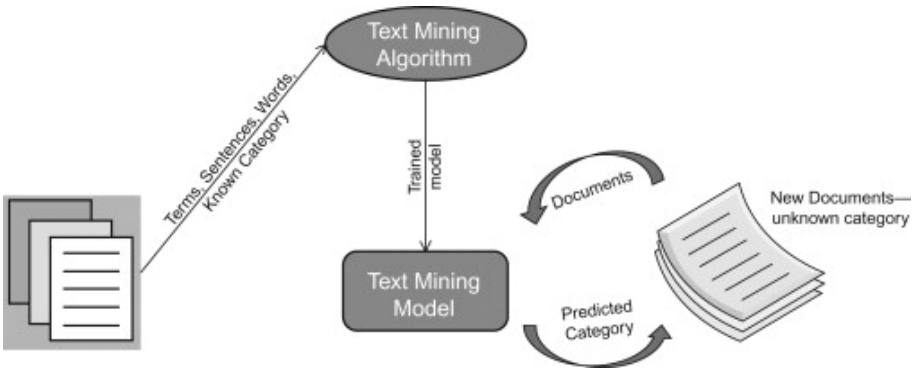



Fig. 1: Text mining process

As a field of artificial intelligence, Natural language processing (NLP) is a task that thrives to interpret and translate human language into meaningful information for machines. Its objective, as well as one of the main achievements, is not only for machines to understand natural language but also to translate highly unstructured data into structured sources for further analysis. Text data can be very hard to analyse, mostly because languages have different structures and most of the time even different versions (i.e. dialects). Moreover, text can contain mistakes, abbreviations or punctuation as well as different syntax rules and terms, therefore, structuring such highly unstructured data can be challenging for a machine. Generally, NLP tasks, such as topic modelling, sentiment analysis, machine translation and text to speech conversion, attempt to analyse and interpret the relationships between parts of text to understand the meaning of words when put together.
Named entity recognition (NER) is an information extraction task that thrives to recognise and classify entities from unstructured text into predefined categories. We refer to an entity as a single token/word or a chunk of words that together refer to the same named entity. For instance, if we consider the following sentence "Amsterdam is the capital of the Netherlands.", an NER model would identify

and extract entities such as "Amsterdam" and "Netherlands" and classify them as "city" and "country", respectively. Although the task in this example might seem of minor importance, NER use cases are getting more and more common across industries. For instance, media corporations use NER to determine the subject of a body of text or to extract similar articles based on the categorised entities; NER is also used by companies that thrive to improve customer experience by automatically classifying user requests and complaints and reduce response time; another important application is in the healthcare domain where NER can quickly parse and extract relevant information from diagnoses and medical reports.

## 2.2 BERT

In the last year, BERT models have become very popular among different NLP tasks such as Question Answering and NER, among many. Its fame is notorious not only for the state-of-the-art performances achieved in several NLP tasks but also for its conceptually simple model architecture. BERT [14] framework consists of the combination of two sequential tasks, namely pre-training and fine-tuning. The former refers to the process of training the BERT model on unlabelled data over multiple pre-training tasks, while the latter refers to the fine-tuning process of all of the parameters using labelled data from the downstream task. Therefore, it comes without saying that one of the main innovation of BERT is the use of two unsupervised tasks to pre-train the BERT model, instead of using the traditional unidirectional language models. These two unsupervised tasks are referred to as Masked Language Models (MLM) and Next Sentence Representation (NSP).

*Masked Language Model* The main idea behind MLM is that the model masks a percentage of the tokens at random from the input data before feeding it to BERT. The MLM's objective is to predict the original masked word's vocabulary ID solely based on the context. Therefore, its objective enables the representation to merge the right and the left context, allowing to pre-train a deep bidirectional Transformer which would not be possible to pre-train with a unidirectional language model. The concepts of MLM can be observed in Figure 2. The Figure depicts an example of how masked language model predicts the output words. As explained, a percentage of the input words is masked and the word embeddings are fed to the Transformer encoder that produces the output vectors. On top of the encoder outputs which are transformed into the vocabulary dimension by multiplying them with the embedding matrix, a classification layer is added. Finally, with the softmax activation function, the probabilities of each word are calculated.

An important note to consider is that the slower convergence of the model compared to more traditional directional models is based on the fact that only the prediction of the masked words is taken into consideration by BERT loss function while the non-masked word predictions are ignored.

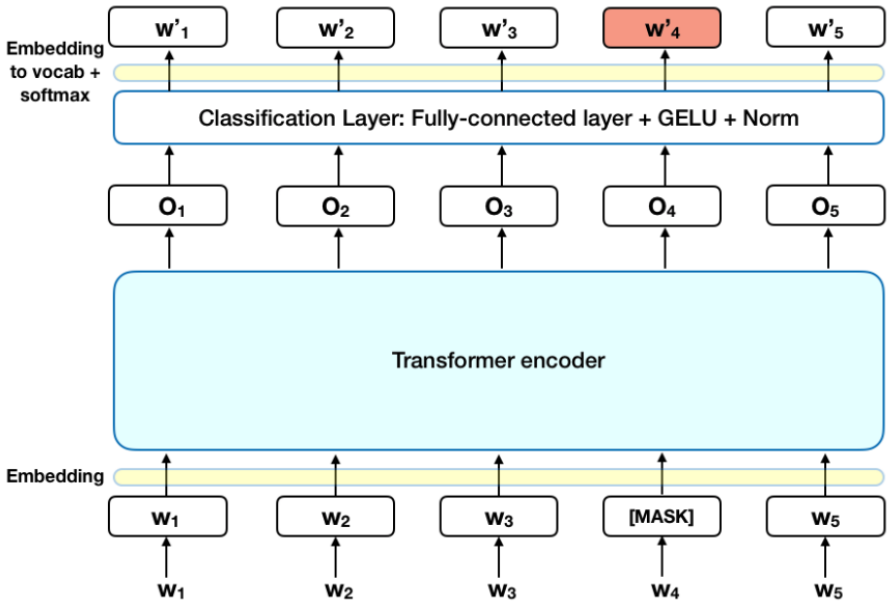



Fig. 2: Bert Masked Language Model [18].

*Next Sentence Prediction (NSP)* Language modelling doesn't directly capture the relationship between two sentences that many downstream NLP tasks, such as Question Answering, are based on. With the Next Sentence Prediction, as the task's name suggests, the model receives as input pairs of sentences `A` and `B` and learns to predict if sentence `B` in the pair is the actual sentence that follows sentence `A` in the original document. When training, 50% of the inputs are a pair where sentence `B` is the actual subsequent sentence in the original document, while 50% of the other inputs are a pair where a sentence from the corpus is chosen at random as sentence `B`. The main assumption in the NSP task is that the random second sentence will not be connected with the first one. Before entering the model, however, the input is processed in order to facilitate the model discriminating between the two sentences. A representation of this process is shown in Figure 3, where sentence `A` refers to the first sentence, while sentence `B` refers to the second sentence. The input tokens are processed in the following way:

- A special classification token (`[CLS]`) is added at the beginning of sentence `A`, while at the end of every sentence a separation token (`[SEP]`) is added.
- A learned sentence embedding is added to each token stating if it belongs to sentence `A` or `B`.
- The position in the sequence is then indicated by adding a positional embedding to each token. In summary, the final input representation of every token is assembled by adding the corresponding token, segmentation and positional embeddings.

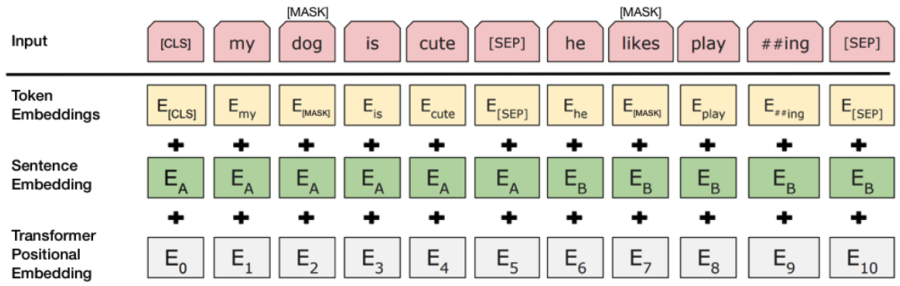



Fig. 3: BERT input representation [14].

**BioBERT & ClinicalBERT** Less than a year after BERT publication, different versions of BERT model started to be proposed. BERT proved to be very effective in several NLP tasks, however, it has its limitations when it comes to domain-specific tasks, mostly because it is pre-trained on only general domain corpora. The rapid increase in the volume of biomedical literature suggested that a domain-specific language model needs to be pre-trained in order to benefit numerous biomedical NLP research tasks. For this reason, Lee et al. [26] released BioBERT, a biomedical language representation model pre-trained on large-scale biomedical corpora. While BERT was pre-trained on general domain corpora, namely BookCorpus and English Wikipedia, BioBERT is pre-trained using PubMed abstracts and PMC full-text articles. In their publication, the authors demonstrated that, when pre-trained on biomedical corpora, BioBERT significantly outperforms BERT in several biomedical text mining tasks. On a similar note, Alsentzer et al. [2] released ClinicalBERT, which is a pre-trained model on more than 2 millions clinical notes from the MIMIC-III v1.4 database, improving not only general domain results but also BioBERT results on 2 established clinical NER tasks and one medical NLI task. The motivation behind the need of specialised clinical BERT models is that clinical narratives, such as physician notes, have different linguistic characteristics from other more general and non-clinical biomedical narratives.

## 2.3 Word Embeddings

*One-hot Representation* The idea behind word embeddings is to capture as much of the semantical, morphological and contextual information as possible from a word. Also known as Count Vectorization, one-hot word embeddings is a rather trivial and naive way to represent a word as a vector. It consists of assigning 1 to only one element that corresponds to the word, and 0 to all the other elements. The generally high vocabulary size, which represents the vector dimensionality, is responsible for the sparse representation of the words. Moreover, one-hot embeddings are not able to catch any relationship between words as well as context, and therefore it is generally considered a naive embeddings representation.

*TF-IDF Representation* *Term frequency-inverse document frequency*, abbreviated tf-idf, is a statistical measure often used in NLP. The main idea behind tf-idf is to measure the relevance of a specific term to a document in an ensemble of corpora. This measure of relevance proportionally increases to the frequency of that word in the document, however, this effect is counterbalanced by the number of documents containing that word. In a large ensemble of documents, words such as an, on, at, is, identified as stop-words, occur very often, however they don't carry a lot of useful

information. Because of their high occurrence in many documents, the vectors of these stop-words are not-so-sparse and some trivial encoding representation, such as one-hot word embeddings, consider these words as terms carrying a lot of information. In various information retrieval and text mining applications, tf-idf is used as stop-words filtering.
Tf-idf is calculated in the following way:

$$tfidf(term, document) = tf(term, document) \cdot idf(term)$$

The first part of the multiplication referring to "term frequency" is calculated as the ratio between the occurrences of that term in the document and the total number of words in that document.

$$tf(term, document) = \frac{n_i}{\sum_{k=1}^{V} n_k}$$

The second part of the multiplication referring to "inverse document frequency" is calculated as the logarithmic ratio between the total number of documents and the number of documents in which the term appears.

$$idf(term) = log\frac{N}{n_t}$$

This way by combining these two quantities a measure of how relevant a term is to a particular document can be used as a vector representation of a word.

*GloVe Representation* GloVe is a word embedding method based on unsupervised learning. To obtain a vector representation of words, the model, implemented by Pennington et al. [36], is trained on aggregated global word-word co-occurrence statistics from a given corpus. Specifically, the entire given corpus is parsed on a single instance and used to populate the matrix with statistics that represent how frequently words co-occur with one another. The GloVe model architecture is a log-bilinear model with a weighted least-squares objective. The choice of this implementation is based on the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. To explain this concept, we can consider the following example in Figure 4. The probabilities of *solid* given *ice* and *water* given *ice* are higher than the probabilities of *gas* given *ice* and *fashion* given *ice*, meaning that the word *ice* co-occurs more frequently with the words *solid* and *water* and more infrequently with *gas* and *fashion*, as expected. The same idea is followed for the probability of word *k* given the word *steam*, however, the ratio of these probabilities behaves differently. High values of this ratio, typically much larger than 1, indicate a high correlation with properties specific to *ice*, while small values, typically much lower than 1, indicate a low correlation with properties specific of *steam*. Values of this ratio very close to 1 indicate a low correlation of the word with both target words in the ratio. In this way, the ratio of probabilities is able to encode some sort of meaning associated with the target words.

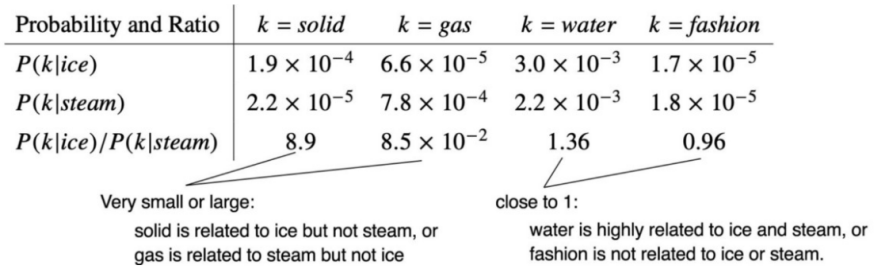



Fig. 4: Example of co-occurrence probabilities for target words *ice* and *steam* with other words from the corpus [36].

*Word2Vec Representation* Word2Vec is a method developed by Thomas Mikolov in 2013 for computing word embeddings. It consists of a two-layer neural networks and it is trained to reconstruct linguistic contexts of words. The strength of Word2Vec is its ability to both transform input text into low-dimensional vector representations and to express the semantic similarity of words or sentences. This way, the model encodes two important qualities. First, it is able to understand semantic similarity relations and, second, it understands linear translation relations in the vector space. For instance, if we consider the vector representation of the word "king", subtract the vector representation of the word "man" and add the vector representation of the word "woman", we will see that the resulting vector is a lot closer to the vector of the word "queen" than to any other.

To compute word embeddings, Word2Vec uses two different architectures, namely and Skip-gram Continuous Bag-Of-Words. Continuous Bag-Of-Words (CBOW) training object is to use context words in a sentence to predict target words. CBOW performs better with smaller data sets because it considers a whole context as one instance by smoothing over a lot of the distributional information. Skip-gram architecture, instead, is the exact opposite of CBOW. It uses target words to predict surrounding context words. Statistically, Skip-gram performs better with larger data sets because it considers each target-context pair as a new instance. For example, training a skip-gram model to learn word embeddings from a 100 billion words corpus can take up to one day with an optimised single-machine implementation.

**SparkNLP Library** SparkNLP is an open-source NLP library implemented by John Snow Labs and built on top of Apache Spark and Apache Spark ML. This library has been awarded for its ability to deliver a unified, scalable and high-accuracy solution for real production use. It's important to notice that, generally, an NLP pipeline consists of a sub-task of a bigger data processing pipeline. For instance, considering named entity recognition tasks, they involve first transforming the training data, then apply NLP annotators, train the model, evaluating the results either with cross-validation or by splitting train and test sets and finally hyperparameter estimation. This is what sparkNLP provides, an end-to-end solution from text pre-processing to the final prediction to help the researcher in all steps of solving a data science problem with NLP. Several common NLP tasks such as stemming, tokenization, sentiment analysis, POS tagging and NER are covered by this library.

For this thesis, the focus goes on the annotators available in this library. SparkNLP offers two kinds of annotators [22]: (1) Annotator Approaches, which represent Spark ML Estimators and require a training stage that is performed by means of the *fit()* function on the input data; (2) Annotator Models, which are spark models or Transformers, meaning they have a *transform()* function which take a dataset and add to it a column with the result of the annotation. To train the NER model in SparkNLP, therefore, the Annotator Approach was used and, specifically, the NerDLApproach framework. This framework is a Char – BiLSTM – CNN architecture, introduced by Chiu and Nichols [9], which achieved state-of-the-art performance in both the CoNLL-2003 shared task and OntoNotes 5.0 data set, exceeding systems that employ more complex feature engineering. Their main contribution consisted of presenting a hybrid model based on bi-directional LSTMs and CNNs that is able to learn both word- and character-level features, eliminating the need for most feature engineering. An overview of this hybrid architecture is represented in Figure 5. The CNN component is used to produce the character-level features. Specifically, for each word, a new feature vector is extracted from character-level feature vectors (i.e. character embedding) by means of a convolution and a max pooling layer. These new feature vectors are then concatenated and fed first into a forward LSTM network and then into a backward LSTM network. A linear layer and a log-softmax layer are then applied to decode into log-probabilities the output of each network for each tag category. The final output is produced by simply adding together these two vectors.

## 2.4 Related Work

In many clinical and public health informatics applications, it is extremely important to extract entities from clinical notes or reports and find the relationships between them in order to gain information and knowledge. NLP has shown to be a huge driver of success in areas of clinical research such as drug repositioning [13], protein research [11] or information extraction from Electronic Health Records (EHRs), among many. Consequently, bioinformatics has seen an increasing number of applications in text mining and information retrieval. A considerable amount of research studies have been conducted in named entity recognition as well as relation extraction in the last years and these studies suggest different methods. Here, a deep explanation of recent achievements in clinical NER, relation extraction and context understanding models is given.

**Clinical Named Entity Recognition** As previously introduced, Named Entity Recognition (NER) is a text mining and information extraction task that parses unstructured text attempting to recognise and classify named entities into predefined categories. Compared with document or sentence level classification tasks, NER normally makes classifications on word or even character level, giving each word or character a category label that denotes whether it is part of a target named entity or not. In the biomedical text mining area, the application of NER is a widely discussed and studied topic, which aims to distinguish entities such as diseases, proteins or genes from text in each clinical document. These detected named entities will then be available for further statistic analysis or relation extraction task to offer evidence, resources or just to give inspiration for biomedical research.
After the first results of Biomedical NER, the NLP community started to rapidly create high-quality and structured data sets. Many of these labelled data sets have

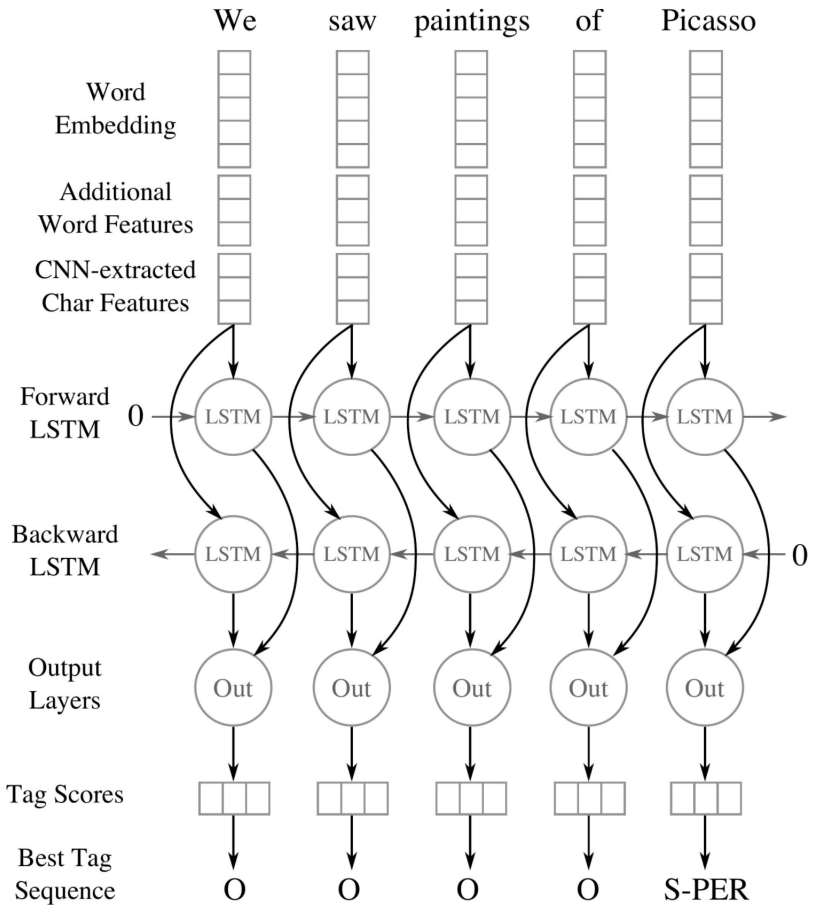



Fig. 5: Unfolded representation of the Char – BiLSTM – CNN architecture for named entity recognition. The CNN component extracts a fixed length feature vector from character-level features. For each word, these vectors are concatenated and fed to the BiLSTM network and then to the output layers [9].

been organised as part of several NER shared tasks. In 2014, the GermEval NER Shared Task made licensed German data with NER annotations available to the public, aiming for a remarkable advancement in the German NER task and for a deeper representation of named entities. Recently, in 2019 the Balto-Slavic Natural Language Processing (BSNLP) Shared Task at ACL 2019 [37] centred around multilingual named entity recognition (NER) in Slavic languages and was composed by several subtasks such as recognition, lemmatization, and entity linking.

Research in the clinical field often requires detailed patients information documented in clinical narratives. Clinical NER [46], specifically, is a fundamental NLP task that thrives to recognise and extract specific entities of interest such as diseases, medications and symptoms from medical records. Researchers have developed and applied computational models in general clinical NLP systems where in most of the cases, such as for MedLEE [15], KnowledgeMap [12] and MetaMap [3], their method is a rule-based one relying on existing medical vocabularies for

NER. The clinical NLP community has also organised multiple challenges to examine the performances of state-of-the-art methods. Most of the top-performed systems are primarily based on supervised machine learning models with manually defined features. In order to further improve accuracy, several strategies have been explored by researchers within the current infrastructure of conventional machine learning models, such as ensemble models, which stack multiple machine learning methods, unsupervised features generated using clustering algorithms (i.e., Brown clustering [6]), hybrid systems, and domain adaptation to leverage labelled corpora from other domains.

Many machine learning models have been applied in clinical research, including Structured Support Vector Machines (SSVMs) [41], Maximum Entropy (ME) and Conditional Random Fields (CRFs) [23]. Many top-ranked NER systems applied the CRFs model, which is the most popular solution among conventional machine learning algorithms. A typical state-of-the-art clinical NER system usually utilises features from different linguistic levels, including orthographic information, syntactic information (e.g. CHUNK tags, Part-Of-Speech tags), word n-grams, and semantic information (e.g., the UMLS concept unique identifier). Some hybrid models further leverage the concepts and semantic types from the existing clinical NLP systems namely MetaMap, cTAKES [38]. To further improve the performance, researchers have also utilised ensemble methods to combine different machine learning models, such as re-ranking. More recently, researchers have also started to examine the unsupervised features derived from large volumes of unlabelled corpora, such as the word clusters generated using Brown clustering [6] and random indexing. Thanks to the in-depth and continuous hard work on clinical NLP, performances of clinical NER have improved considerably, while at the same time targeting some obstacles that would impede further improvement, such as:

- **Fragile feature representation.** In many cases, feature representation, such as bag-of-word, for clinical NER is fragile due to sparsity problem. Many times, entities that have related concepts are not recognised as similar using bag-of-words feature representation, therefore, there is a need for more robust feature representations [44].
- **Time-consuming and handcrafted feature engineering.** Conventionally, in machine learning solutions, the extraction of features heavily depends on humans while the machine can only handle the parameter optimisation supervised by the gold-standard annotations. Human feature engineering can bring several issues, such as incomplete feature or repeated in many complex features and feature combinations. For these reasons, automatic feature learning algorithms are increasing in popularity and need as they would release the researcher from the time-consuming feature engineering.
- **Lack of long-term dependencies.** It has been observed in many experiments [10] that the system prediction errors have reported false negatives caused by the lack of long-term dependencies. It goes without saying that the clinical NER system needs a better architecture to capture long-term dependencies from clinical texts.

In the last decade, increasing efforts have been put to explore a new emerging technology, deep learning, to improve the current clinical NLP systems. Deep learning is a sub-domain of machine learning that uses deep architectures to learn high-level feature representations. Currently, deep neural networks are commonly used as the unique deep architecture for high-level feature learning. Deep learning models introduced word embedding as a critical technique to train densely-valued vector representation of words to replace the fragile bag-of-word representation. Each word in the vocabulary is represented by a row of the matrix while each column is associated with a latent feature. The input word sequence can be transformed into a vector by concatenating the corresponding word vectors from the embedding matrix. By automatically learning high-level features automatically, deep neural network architectures can release researchers from time-consuming feature engineering.

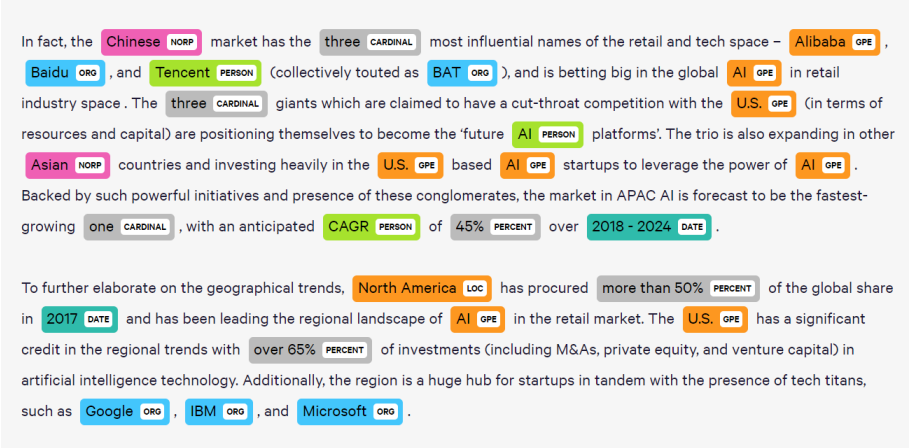



Fig. 6: Example of Named Entity Recognition on text data.

Figure 6 shows an example of how Named Entity Recognition works on general text data. It locates entities that can variate from a person to a more domain-specific term in an unstructured or semi-structured text. For example, in the sentence *"In fact, the Chinese markets has the three most influential names of the retail and tech space [...] of 45% over 2018-2024."*, four different entities have been recognised: Chinese - Nationality, three - Cardinal number, 45% - Percentage, 2018-2024 - Date. It is easily understandable, from this example, the power that NER has in enabling machines to understand context in text data.

**Temporal Relation Extraction** Information Extraction (IE) represents one of the major achievements in NLP. General relation extraction, particularly, aims to discover and extract relationships between events that are present in plain text. In the clinical domain, specifically, relation extraction has drawn a remarkable amount of attention from NLP researcher given the important role that plays in applications such as creating patient's clinical timeline, predicting disease given the history of the patient, among many. In their work, Alimova et al. [1] proposed a machine learning model with a novel set of features, namely BioSentVec embeddings and

knowledge-based features, with the aim to identify, in a medical record, the relationship between drug entities and their attributes. These features were tested systematically on the impact with standard distance and word-based features on both the MADE 2018 and the i2b2 2018 benchmark data sets, resulting in improving the F1-score by 3.5% on the MADE corpus.

Magge et al. [31] took it one step further and proposed a natural language processing pipeline consisting of an NER model to identify and extract 9 medical entities from clinical narratives and a random forests classifier to classify 7 types of relations between entities. Their method focused on using bidirectional LSTM units coupled with a CRF classifier at the output layer. As explained by the authors, this model has achieved interesting results in a variety of sequence tagging and chunking tasks. The most important part of their work regarding relation extraction can be divided into two parts. Firstly, they use a binary classifier to filter out entity pairs based on their types such that only entity pairs with possible relations between them are selected. They then use, as inputs to a random forests classifier, features extracted from the two entities and their contexts to identify the type of relationship between them.

Temporal Relation Extraction (TRE) is a task in Information Extraction that thrives to identify and extract tokens or chunks corresponding to temporal intervals to determine the temporal relations between them. The entities extracted consist of temporal expressions such as datetime events, eventualities, or auxiliary signals that support the interpretation of an entity or relation. The relations extracted consist of temporal links (TLinks) which describes the order of events and times. As Galvan et al. [16] explain, TRE has proven to be one of the most challenging tasks in NLP. Nowadays, it covers an important active area of research where the ultimate goal is to be able to create a patient's clinical timeline and represent its detailed clinical history. Thanks to the extensive adoption of structured Electronic Health Records (EHRs) there has been a significant increase of research studies in the NLP community, raising hopes that clinical narratives can be used to improve and sometimes solve information extraction challenges in real-world settings. On the same line, recently several corpora, from clinical narratives to more domain-general text, have been annotated as part of shared tasks with the aim of implementing systems able to accurately build timelines based on the events described in the text. Several information extraction tasks can benefit from the implementation of these narrative timelines, namely question answering [48], text summarization, clinical outcomes prediction [28], and the identification of temporal patterns [51], among many.

Research in the area of temporal relation extraction has been led by several shared challenges such as TempEval shared tasks [43] but in recent years, the target domain has been shifted to the clinical domain [5]. Results from clinical TempEval shared challenges, however, showed that temporal relation extraction remains one of the most difficult tasks. Taking as an example the best-ranked system in 2016 Clinical TempEval, UTHealth [24], using an end-to-end system based on linear and structural Hidden Markov Model, showed a significant gap of 0.25 when compared to human performance even with gold-standard entity annotations.

Even though the reasons for the discrepancy of the results between named entity recognition and temporal relation predictions are still not fully discovered, it is clear that the complexity of temporal representation in natural language processing represents the main cause of the low performance on temporal relation tasks. As Galvan et al. [16] investigated, "Tense and aspect are the two grammatical means

to express the notion of time in English but little has been discussed about the latter on clinical text.".
Recent work in temporal relation extraction by Lin et al. [29] described a method that can automatically generate more high-quality training occurrences. Their approach is to semantically expand gold medical events based on the Unified Medical Language System (UMLS). For instance, if we consider the following sentence "*Last week the patient suffered from severe back pain*", with a relation between *severe back pain* and *last week*, this method is able to automatically create additional training occurrences by expanding the relation to the words in the event. Therefore, the relation between *last week* and *severe back pain* will automatically generate three additional relations of the same type where the second arguments are *pain* and *back pain*. The authors demonstrate several advantages of this method, such as the ability to generalise between-argument signals in a more effective way as well as having a robust mechanism of data augmentation. As a consequence, their method reached state-of-the-art results, achieving a two points improvement over the best system for the Clinical TempEval 2016 challenge.

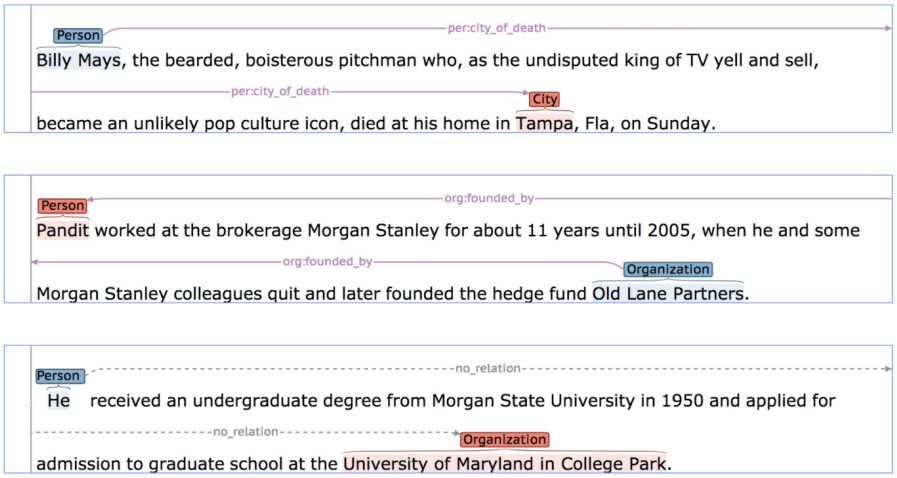



Fig. 7: Example of Relation Extraction from TACRED data set.

In Figure 7 an example of relation extraction is shown from the TACRED data set. TACRED is a large data set specifically for relation extraction tasks that consists of 106,264 examples built from the corpus used in the yearly TAC KBP challenges. As the example depicts, the data set provides annotations such as object mentions, the spans of the subject, the types of the mentions, the relationship held between the entities as well as no_relation label.

**Context understanding models** One of the main goals of NLP is trying to find the meaning of a sentence or text. The process of understanding the context of a corpus by answering to questions like *Who is the subject? What is the subject talking about? How do they feel? And why?* is known as Context Analysis. In NLP, context analysis involves breaking down unstructured text data to help the machine understand the context or, more specifically, to extract information such

as n-grams, noun phrases, themes, and facets present within sentences. Extracting knowledge around the sentiment of a sentence doesn't fully describe the context of a sentence and that is where theme extraction and context determination come into play. Recent years have seen a huge hype around language representation models.

Devlin et al. [14] proposed a conceptually simple but empirically powerful language representation model called BERT: Bidirectional Encoder Representations from Transformers. BERT is a fine-tuning based approach which is designed to pre-train, on unlabelled text, deep bidirectional representations by jointly conditioning on both left and right context in all layers. The significant results of BERT, such as the 7.7 points absolute improvement on the GLUE score or the 5.1 points absolute improvement on the SQuAD v2.0 question answering, served as a breakthrough in the context understanding domain. In later years different versions of pre-trained BERT models started to become publicly available. Alsentzer et al. [2] addressed the need to release pre-trained BERT models for clinical text data, particularly for discharge summaries and for generic clinical text. In their work, releasing ClinicalBERT, they demonstrated that with these domain-specific models the performance in 2 clinical NER shared tasks, namely i2b2 2010 and i2b2 2012 shared task, improved significantly. Similarly, Lee et al. [26] released BioBERT, a pre-trained biomedical language representation model based on BERT but specifically meant for biomedical text mining. BioBERT proved to outperform state-of-the-art models on three representative biomedical text mining tasks, namely biomedical relation extraction, biomedical NER and biomedical question answering.

# 3 Methods

The methods section is divided into three subsections: data sets, approach and system setup, each with their own subsections. The data sets subsection describes what data sets have been used in the experiments together with an overview of the shared challenges related to them. In the approach subsection, the workflow is explained in detail. The main components of the NER and temporal relation extraction systems are illustrated in Figure 8. Firstly, the focus is on one of the most important tasks in Natural Language Processing, namely pre-processing, where all the processes of how the data is cleaned and formatted are discussed. Secondly, the NER models training for entities and temporal events are accurately described. Lastly, the architecture of the temporal relation extraction model is exhaustively covered for all the versions used. Moreover, the parameters of the models, and their values for the experiments, used to train the models are explained in detail. Following the explanation of the parameters as well as an overview of the evaluation methods is the system setup subsection where the cluster implementation to run the experiments is described together with some critical thoughts about model deployment and ML-Ops. Additionally, the hardware used to run the experiments as well as the software used for the implementations of the techniques are mentioned.

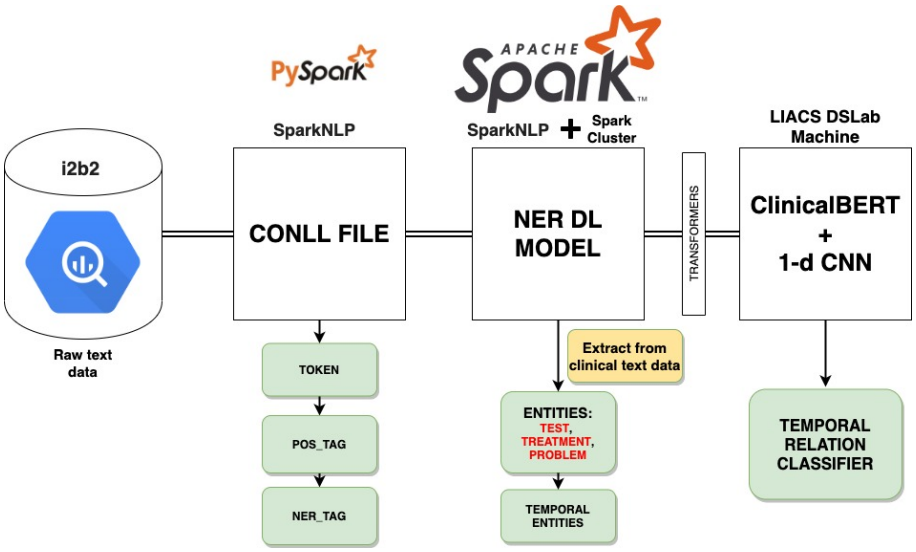



Fig. 8: High-level view of the represented structure.

## 3.1 Data sets

**Mimic-III.** Mimic-iii is a large, relational database maintained by PhysioNet [17] that contains de-identified health data from more than 40 thousand patients admitted to the care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 [17]. This database encloses 26 tables among which we focus on the NOTEEVENTS table, with 2,083,180 records, that incorporates clinical notes of patients (i.e. Echo reports, ECG reports, radiology reports).

**i2b2 2010.** This data set is the result of the 2010 i2b2/VA Challenge [42]. This shared task challenge encloses three sub-tasks that thrive to:

1. Extract medical annotations, namely problem, treatment and test entities.
2. Classify assertions made on medical problems.
3. Find relations between the extracted medical annotations.

The data available for this shared task consists of discharge summaries from two hospitals, namely Partners HealthCare and Beth Israel Deaconess Medical Center (MIMIC-II Database), and discharge summaries and progress notes from University of Pittsburgh Medical Center. To ensure the anonymity and the privacy of the patients, a de-identification algorithm, that removes all specific individual identifiers, has been run on all records, while the annotation task has been manually carried out by experts. In this thesis, the main focus is on the first sub-task, namely the extraction of medical problem, test, and treatment annotations.

| Data sets | | Notes | Entities | Entity types |
|---|---|---|---|---|
| i2b2 2010 | Training set<br>Test set | 349<br>477 | 27,837<br>45,009 | Problem, Treatment, Test |
| i2b2 2012 | Training set<br>Test set | 190<br>120 | 16,468<br>13,594 | Events, TLINKs |
| MIMIC-III | N/A | 2,083,180 | N/A | N/A |

Table 1: Descriptive statistics on the i2b2 2010, i2b2 2012 and MIMIC-III data sets.

**i2b2 2012.** Similarly to i2b2 2010, this data set is the result of the i2b2 2012 Challenge [39]. This data set consists of 310 de-identified discharge summaries resulting in over 178,000 tokens, with three main types of annotations, namely clinical events (EVENTs), temporal relations (TLINKs) and temporal expression (TIMEX3s). Moreover, it contains on average 87 events, 176 temporal relations and 12 temporal expressions per discharge summary. In this corpus, eight types of TLINKs between events and temporal expressions were annotated: `DURING`, `AFTER`, `BEGUN_BY`, `SIMULTANEOUS`, `OVERLAP`, `BEFORE`, `ENDED_BY` and `BEFORE_OVERLAP`. A sample text of a patient's report with the annotated events and expressions is represented in Figure 9. For the annotation of this data set, few considerations have been taken, as explained by Sun et al. [39]. The annotation task for the 2012 i2b2 data set has been carried out by eight annotators with a medical background. Each clinical narrative has firstly been annotated by two annotators and secondly, it has been adjudicated by a third annotator. This agreement, called the inter-annotator agreement, has been used to define the quality of the annotation, therefore a low level of the agreement would mean that not all the annotators agreed with the annotation type. The analysis performed by Sun et al. [39] demonstrated that the inter-agreement on some temporal links between events and temporal expression was low. In response to this observation, as suggested by authors of the paper, the eight temporal links types were merged in the following way: `AFTER` and `BEGUN_BY` were merged as `AFTER`; `DURING`, `OVERLAP`, and `SIMULTANEOUS` were merged as `OVERLAP` and `BEFORE`, `BEFORE_OVERLAP`, and `ENDED_BY` were merged as `BEFORE`. Table 2 presents the details of the merged data set.

| | Training set | Test set |
|---|---|---|
| Temporal expressions | 2,366 | 1,820 |
| Events | 16,468 | 13,594 |
| Temporal relations | 33,635 | 27,736 |
| BEFORE | 17,513 | 15,113 |
| OVERLAP | 12,823 | 9,894 |
| AFTER | 3,207 | 2,729 |
| Unlabelled | 92 | 0 |

Table 2: Summary of the merged i2b2 2012 temporal relation corpus.

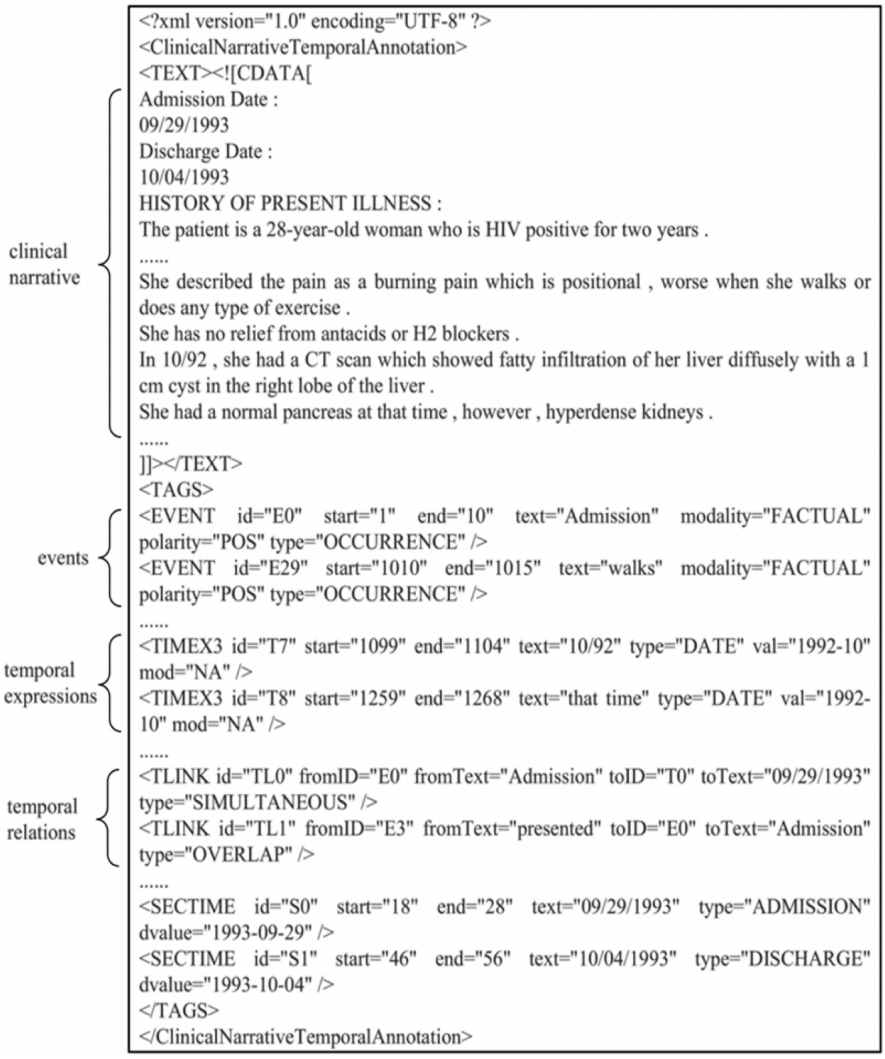



Fig. 9: Sample text from the i2b2 2012 data set

## 3.2 Approach

This section represents one of the most important steps for the implementation of this thesis project. First, the raw data is pre-processed. This step is performed both for the NER and TRE tasks. During this step, the clinical notes are transformed into meaningful training data following the CoNLL file format to be fed to the NER model and the TRE model. Second, the cleaned and structured training data set is used to train the NER model in SparkNLP to extract clinical entities, namely *Problem*, *Treatment* and *Test*, as well as date entities. Lastly, the clinical notes are used to train a temporal relation extraction model which would be able to link temporal events to the clinical entities extracted at the previous stage.

**Pre-processing** In order to train a Named Entity Recognition DL annotator on SparkNLP, it is fundamental to transform the raw clinical notes into the CoNLL file format. CoNLL refers to Conference on Natural Language Learning which is

a SIGNLL's yearly shared task where different challenges are tackled. In NLP, the CoNLL file format is a way of representing corpus with one word per line and with each word containing multiple tab-separated columns with information about the word, such as its POS-tag, Chunk-tag, NER-tag, Lemmatized form, among many. There are many different versions of CoNLL file formats, however, for this thesis project, the CoNLL-2003 file format has been used to transform the data set before proceeding with training the model. The CoNLL-2003 file format contains four tab-separated columns, as previously explained. Each word is placed on a single line and sentences are separated by an empty line. The item on the first tab for each line is a single word, followed by a part-of-speech (POS) tag on the second tab, a syntactic chunk tag on the third tab, which was not used in this project, and the named entity (NER) tag on the last tab. Moreover, the NER tags follow the IOB2 tagging format, otherwise known as Inside-Outside-Beginning format. The B- prefix of the NER tag refers to the beginning of a chunk, the I- prefix refers to an entity that is inside a chunk, while the O tag refers to a token belonging to no chunk. This whole process of formatting the training data set, following the CoNLL style, was implemented by a SparkNLP pipeline that first extracts sentences from text using SentenceDetector annotator, identified tokens by means of a tokenizer and normalized them following a regex pattern that would remove all the non-relevant characters from text such as `[@\'^\&$\#~_+]`. The pre-trained PerceptronModel `pos_anc` is used in the SparkNLP pipeline to annotate the Part-of-speech (POS) tag for every token. The last column of our CoNLL file consists of the NER-tags. The i2b2 2010 data set already provides the annotation file in which the three entities of interest, namely *Problem*, *Treatment* and *Test*, are annotated. Therefore, to populate the last column of the CoNLL file, the NER-tags from the annotation file have been merged with the single-line tokens in the CoNLL file. An illustration of this process is shown in Figure 10, where the clinical note on the left is processed to create the CoNLL file on the right with the above-mentioned annotations as columns. This way the raw clinical narratives are pre-processed and transformed into the so-called CoNLL training file, which is used to train the NER DL model at the next stage.


Fig. 10: Pre-processing of the annotated i2b2 2010 data set (**left**) into the CoNLL training file format (**right**).

For the Temporal Relation Extraction task, however, the pre-processing step on the i2b2 2012 data set was rather straightforward. Firstly, the clinical narrative files, for which a sample is given in Figure 9, are parsed and the temporal links (TLINKS) together with events and temporal expression tags are retrieved. Secondly, the output file, that will be used to train the TRE model, is generated and consists of four tab-separated components: first, an ID that identifies the entry, followed by the target tags such as entities for which a relationship exists. The third tab consists of the original text of the clinical note, while the fourth tab defines the type of relation between the target tags, categorised as the label.

**Named Entity Recognition** The next stage of the NLP pipeline after pre-processing consists of training the Named Entity Recognition model. Before starting to train the model it is important to identify which word embeddings will be used during the training process. Several pre-trained word embeddings were considered in these experiments. Initially, GloVe embeddings were used. GloVe [36], Global Vectors for Word Representation, is an unsupervised learning algorithm for obtaining vector representations for words. For this specific task, we considered GloVe embeddings with 100 dimensions pre-trained on 6 billion tokens and 400 thousand vocabularies. Further research, however, led to the application of BioWordVec [47] [7], pre-trained embeddings for biomedical words. BioWordVec is a 200-dimensional word vector computed with fastText and trained on PubMed and MIMIC-III clinical notes with a window size of 20, 0.05 learning rate, and a sampling threshold of 1e-4. Moreover, the pre-trained vector, publicly available, follows the word2vec bin format.
Following the selection of the word embeddings, it is time to train the NerDLApproach model in SparkNLP as explained in the SparkNLP Library section. While training, several functions are selected. Firstly, the input and output columns are provided, where the training data, word embeddings and the NER predictions are

respectively passed to the model. The label column provided refers to the ground truth for the supervised task, namely `TEST`, `PROBLEM` and `TREATMENT` labels. For training, the number of epochs considered in the experiments varies between 20 and 10, while the learning rate values range between 0.001 and 0.005. Lastly, the batch size examined for different configurations are 8 and 64. These values chosen for the experiments are based on previous heuristics and projects [21], where similar NER tasks have been addressed. Finally, a validation split parameter is set to 0.2 and 0.25, defining the proportion of training data set to be validated against the model on each epoch, while a path to the test data set is added for performance evaluation on unseen data. Moreover, after training the model and evaluating it on the validation set, the best model configuration was chosen to train the model on the entire training data, with validation split parameter set to 0. The performance evaluations, alongside the multiple implementations and function parameters considered, are furtherly explained in the Results section.
For the extraction of temporal expressions, such as dates, times of the day or durations, a date extraction pipeline was implemented in SparkNLP. After the tokenization of the document, each token was represented as a 100-dimensional word vector using GloVe pre-trained word embeddings model. The annotation of temporal expressions was then accomplished by means of `onto_100` pre-trained NER model, available in SparkNLP. Onto 100 is an NER model, trained with GloVe 100-dimensional word embeddings on the OntoNotes text corpus and it annotates text to find features such as organisations, name of people, places and dates, among many.

**Temporal Relation Extraction** For the Temporal Relation Extraction task we applied different versions of BERT + 1-d CNN classifier which was first introduced by Chen et al. [8]. In particular, from their work, we tested the architecture using different versions of BERT model, namely BioBERT and ClinicalBERT pre-trained models. In this work, in order to increase the performance of the 1d-CNN in clinical temporal relation extraction tasks, we apply the 1d-CNN classifier firstly presented by Kim [20] and improved by Chen et al. [8] to fine-tune the pre-trained versions of BERT. The core components of the 1d-CNN architecture are explained in four steps as follows:

i Firstly, the input layer converts variable-length medical relation documents into fixed-length vectors.
ii The convolution layer then uses a 1-d convolution to extract semantic features from the input vectors by means of multiple convolutional kernels.
iii A pooling layer, in which the most useful semantic features are selected by a max-overtime pooling operation;
iv Finally, the output layer uses a fully connected softmax classifier to concatenate and classify multiple features.

When training, back-propagation of the training error is used to fine-tune BERT's parameters, which are firstly restored from the pre-trained model.
Moreover, the pre-trained word embeddings used in 1d-CNN are PubMed and GloVe embeddings. When training the 1d-CNN, the input of the model is represented by a concatenation of the two entities of a relationship together with the text where they both occurred and the target TLINK type. As applied by the authors of the original paper [8], the activation function used consists of rectified linear units, filter windows of lengths 3, 4 and 5 with 100 feature maps each and a dropout rate

of 0.5. The fine-tuning process of 1d-CNN, using the pre-trained BioBERT model and ClinicalBERT model, is carried out by concatenating two entities in a relation as the first input sentence and the text where the two entities both occurred as the second input sentence. The BERT model used is an *uncased_L-12_H-768_A-12* while the BioBERT and ClinicalBERT models are *'cased_L-12_H-768_A-12'*, with 12 layers, 768 hidden nodes, 12 heads and 110M parameters.

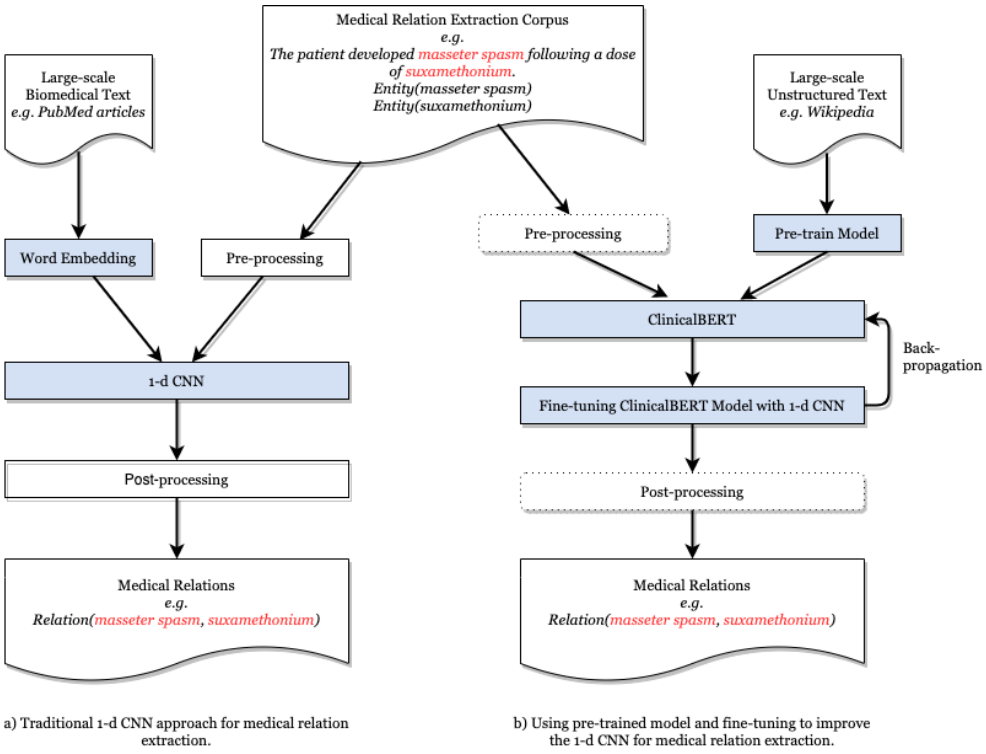



Fig. 11: TRE model's architecture based on Chen et al. [8] model framework to which different versions of BERT model have been applied and examined.

**Evaluation Method** The performance evaluation of the above-mentioned approaches is carried out using standard measures of Precision, Recall and F1-score. Precision is a metric that calculates the number of positive class predictions among the actual positive class. If the focus is on minimising the number of false positives than the precision metric is more appropriate.

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

Recall is a metric that quantifies the proportion of actual positives that were correctly identified in the data set. If the focus is on minimising the number of false negatives than the recall metric is more appropriate.

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

F1-score is a metric that balances the concerns of both precision and recall and it is defined as the harmonic mean of the models precision and recall. Alone, neither precision nor recall can give an overall performance of the model. An increase in precision would mean a decrease in recall, or alternately, a decrease in precision would mean an increase in recall. F1-score provides a trade-off between these two metrics and is able to express the overall performance of the model.

$$\textit{F1-score} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

These metrics will be used to evaluate the performances of both NER and temporal relation extraction tasks. For the experiments, the training process was performed on the training set, while the evaluations of the performances of the models were done using the held-out test set for both tasks. In this thesis, the process of automatic hyperparameter optimisation was not included as part of the implementation. The choice of different hyperparameter values, however, was mostly based on previous heuristics and practical work [21], where similar task were addressed.

## 3.3 System Setup

**Distributed system setup** Every day larger volumes of data are collected and processed and this gives rise to lack of computation power in processing, reading and manipulation of data. In this section, this issue is tackled while introducing several best practices in data privacy, distributed system development lifecycle and the use of Cloud Providers for this matter.
With the recent improvements in cluster-computing frameworks like Spark and Dask, the training, pre-processing and deployment purposes can be distributed more dynamically. In this thesis, the advantage of the Spark General Purpose Cluster framework was used to deploy on 5 nodes on a Virtual Private Network. Using such toolset, the computation was allowed to perform on more than 125 units of CPU and more than 150 GB of memory. Alongside the computation power, security, network connectivity and deployment of the trained model were carefully analysed. As expected, general-purpose frameworks do not provide such infrastructure, therefore Google Cloud Platform was used to host the data sets and create automated Machine learning pipelines. To make the most out of GCP, an isolated single Virtual Machine (VM) was created and the data sets migrated to Google Big Query. As one of the biggest advantages of cloud platforms, a Virtual Private Cloud network was enabled to increase security and isolation of the processes. This allowed accessing BigQuery data sets from the VM, as multiple services like Jupyterlab were hosted on it, to manipulate the data and have visual access to service. Although GCP itself provides several High-Performance service and cluster, using a self-managed Spark cluster seemed to be a better choice but connecting this to the self-contained VPC network was a challenge in which it was feasible to use SSH port-forwarding and Nginx to establish connectivity between these two networks. With this configuration, it was possible to take full advantage of a self-hosted Spark cluster and GCP services in a self-contained ecosystem which was fully privacy data compliant. In this journey for making the best out of distributed systems and dynamic toolsets like GCP, several questions and technical difficulties were raised for deployment and hosting of trained ML models. Developing ML applications in research labs or Institutes are rather straightforward as they are not designed for deployment in the industry.

**Deployment and ML-Ops** Every day more and more complex and computationally expensive models are being demystified and commoditised for public use. This opens the challenge of easy access to these models for consumer devices like mobile phones and IoT devices. In the past, organisations transpired Data Science source codes to more deployable languages like Java or C++. With the increase in complexity and size of the application, this seems impossible today. This section presents the best practices for taking an ML application from the development environment to public production with few clear steps. To remove run-time and project-level dependencies, containerising the source code with tools like Docker which can help remove barriers of platform and software incompatibility. One of the concerns regarding deployment for public use is scalability which can be fully managed by Cloud providers like AWS and GCP. These cloud providers can deploy and fully maintain the containers and publish an endpoint for consumers and developers to take advantage of. Creating an automated pipeline can enable the user to increase the number of deployments and a fasted route to the production stage. Alongside self-managed service from cloud providers, other tools like Kubernetes and Openfaas can be used for container orchestration and containers as a function service.

**Hardware and Software** The proposed NER and Temporal Relation Extraction were implemented in Python 3.7.1 64-bit using JupyterLab framework. The Python programming language was chosen due to the wide variety of packages available and support for Apache Spark and SparkNLP that allowed easy computations. The SparkNLP library [22] was a huge asset to pre-process the clinical notes and to perform which were used in the experiments of this thesis. Appendix A contains the packages used in these experiments as well as their (advised) versions.
The experiments performed on the NER task were executed on the Leiden University Spark Cluster `fs.dslc.liacs.nl`, with the setup explained in the previous distributed system setup section. The experiments performed on the TRE task, however, were implemented on using the Mithril server of the Data Science Lab at LIACS. This computing machine has 1 TB of RAM and 64 Intel Xeon E5-4667 v3 cores and it is specialised in running CPU-heavy programs.

# 4 Results

The results section contains the metric outcomes as well as the evaluation of the performances of the experiments performed on both NER and TRE tasks.

## 4.1 Named Entity Recognition

For this task, several parameters as well as word embeddings have been used in the experiments. Table 3 summarises the results of this task per configuration. Each one refers to the different set of functions used to train the NerDLApproach model, and their functionalities are extensively explained in the Approach section. Configuration A considers the maximum number of epochs set to 20, a learning rate set to 0.001, a batch size of 8. Configuration B, instead, considers the maximum number of epochs set to 10, a learning rate set to 0.001, a batch size of 64. Configuration C, finally, considers the maximum number of epochs set to 20, a learning rate set to 0.005, a batch size of 64. These different configurations are matched with different word

embeddings to perform the experiments, namely BioWordVec, GloVe_100d and Bert_base_cased embeddings. The results in Table 3 show that the best NER model is the combination of configuration A and BioWordVec embeddings, achieving an F1-score of 86.54%. The reason for choosing these specific batch size and learning rate values is purely heuristic and it's based on previous NER implementations in SparkNLP available in [21]. However, for further improvements, a more robust and efficient parametrization strategy is needed.

| Configuration | Embeddings | Precision | Recall | F1-score |
|---|---|---|---|---|
| A | BioWordVec | 87.64% | 85.48% | **86.54%** |
| B | BioWordVec | 83.34% | 84.16% | 83.74% |
| C | BioWordVec | 84.05% | 84.82% | 84.43% |
| A | GloVe_100d | 80.87% | 84.71% | 82.75% |
| C | GloVe_100d | 80.59% | 84.64% | 82.56% |
| B | Bert_base_cased | 82.12% | 84.52% | 83.30% |
| C | Bert_base_cased | 81.93% | 84.42% | 83.16% |

Table 3: Results for different configurations of the NER model on the i2b2 2010 test set. The result in red represents the lowest performance (F1-score) in this task given configuration and embeddings, while the result in **bold** represents the highest performance.

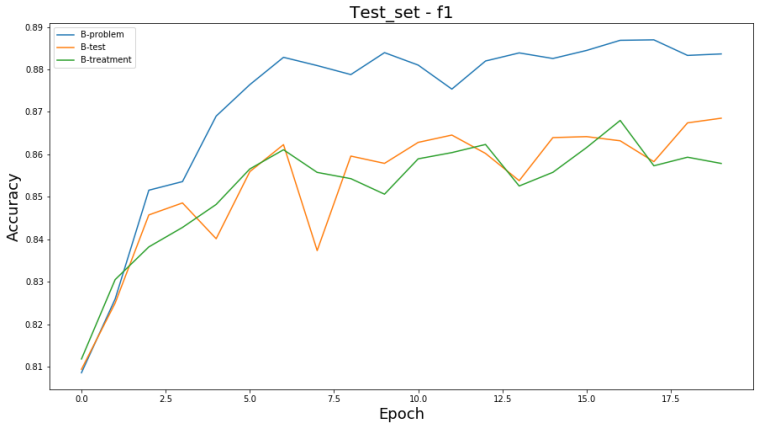
Considering the F1-score as the base evaluation method, these results show that, for all the experiments, the BioWordVec embeddings achieve better performances compared to the other word embeddings. The reason for this disparity can most certainly be attributed to the nature of these embeddings. BioWordVec is an embeddings vector pre-trained on PubMed and MIMIC-III clinical notes and therefore more suited for a clinical NER model, while GloVe and Bert embeddings are pre-trained in Gigaword and other large text such as Wikipedia. Moreover, as explained in the previous sections, context-free models such as GloVe or Word2Vec only generate a single embedding representation for each token in the vocabulary, so the token "book" would have the same vector of embeddings in "to book a hotel room" and "buying a book at the library". In contextual models, such as BERT, instead, each word is represented based on the other surrounding words, both on the left and on the right, in the sentence, making these models deeply bidirectional. Consequently, as expected, the results for NER model using GloVe_100d embeddings for both configurations proved to be not comparable with the other settings that achieved higher performances and, therefore, was not considered for the next task in this thesis.

| Approach | Feature | Precision | Recall | F1-score |
|---|---|---|---|---|
| Char-CNN-BiLSTM | BioWordVec Embedding | 87.64% | 85.48% | 86.54% |
| CharLM-BiLSTM-CRF [50] | Word Embedding | - | - | **88.13%** |
| RNN by Wu et al. [46] | Word Embedding | 85.33% | 86.56% | 85.94% |

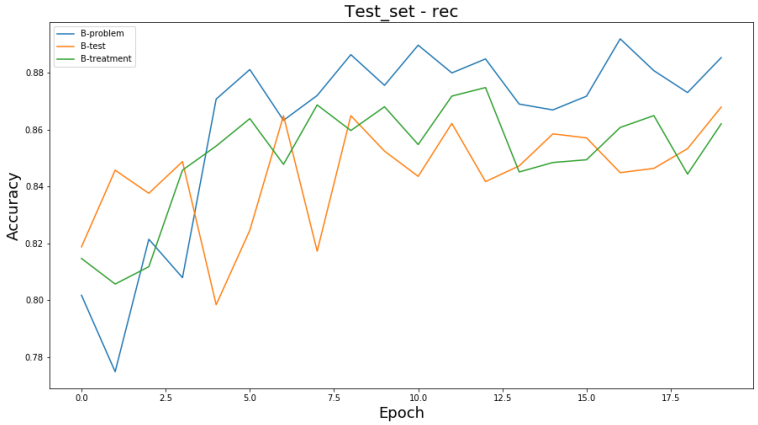
Table 4: Performance comparison between different approaches for the Named Entity Recognition task using the i2b2 2010 test data set.

As the benchmark for this task, we considered the recent results for clinical NER from Zhang et al. (2020) [50]. To tackle this task, the authors implemented a CharLM – BiLSTM – CRF based architecture. Specifically, a bidirectional LSTM character-level language model is trained to supplement the word representation in each sentence. At tagging time, the representations from these CharLMs at each word position are concatenated with a word embedding. The result is then fed into a standard one-layer Bi-LSTM sequence tagger with a CRF-based decoder. Through the use of this model, the authors achieved state-of-the-art results with an F1-score of 88.13% on the 2010 i2b2 test set. Table 4 depicts the comparison between the benchmark and the results from this work. With an F1-score of 86.54% the best NER model in this work, implemented using the Char – BiLSTM – CNN architecture, attained results close to the baseline. Another, less recent, benchmark considered is the work by Wu et al. (2017) [46]. Their implementation based on an RNN model using bi-direction LSTM neurons achieved an F1-score of 85.94% on the test set using only the word and character embedding. It is interesting to notice that the best NER model in this project managed to achieve similar results to the benchmarks even without the use of GPU machines, as instead considered in the RNN implementation [46].

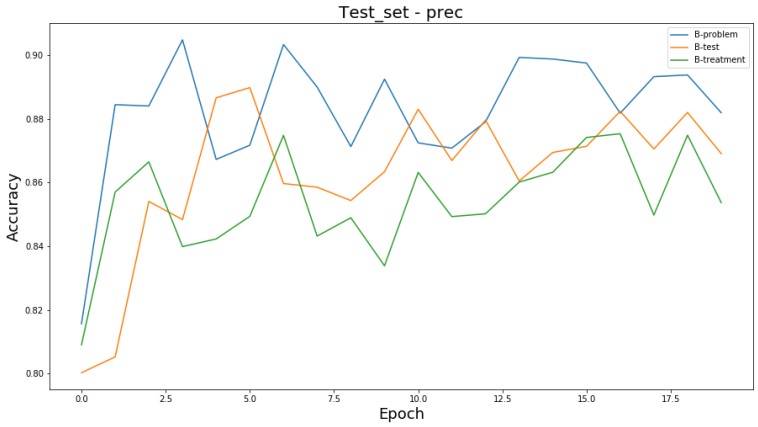
Furthermore, Figure 12 depicts the performance evaluation on the 2010 i2b2 test set of the best NER model for the three named entities considered: `problem`, `test` and `treatment`. As one can notice, for all three metrics used, the best NER model is able to achieve an accuracy of 85% already after 5 epochs on the test set, meaning that even with limited computational power to train it, the model can achieve relevant performances. Moreover, the F1-score (a) and the Recall (b) visualisations suggest that, between the three named entities, the `problem` entity achieves higher accuracy on the test set with an F1-score and a Recall of 88.37% and 88.54%, respectively. Between the three metrics, however, the `treatment` entity achieves the lowest accuracy with an F1-score of 85.79%, a Recall of 86.21% and a Precision of 85.37%. (For further visualisation on other NER models experimented see Appendix C).




(a) F1-score on the test set.




(b) Recall on the test set.




(c) Precision on the test set.

Fig. 12: Performances of the best NER model on the 3 named entities: `B-problem`, `B-test` and `B-treatment`.

## 4.2 Temporal Relation Extraction

Tables 5 to 7 depict the performance results on the merged i2b2 2012 data set for different implementations of the TRE task, namely BERT/BioBERT/ClinicalBERT + 1-d CNN. Table 5, in particular, shows the results using the BERT uncased pre-trained model. Although the performances for the `BEFORE` and `OVERLAP` labels are quite high with an F1-score of 86.68% and 80.73% respectively, the model performs poorly when predicting the `AFTER` label, with an F1-score of only 8.41%. These limited results suggest the need for a pre-trained language model specifically in the clinical domain. For this reason, experiments using BioBERT and ClinicalBERT have been conducted and the results on the merged i2b2 2012 data set are shown in Table 6 and 7. For both the implementations, the performance increased when compared to the one using BERT uncased model. Specifically, for the BioBERT implementation, the F1-score, for the `OVERLAP` label, increased by 0.94 percentage point, while for the `AFTER` label, increased by 25.63 percentage points. Overall, this model achieved an accuracy of 82.21% which proves the increase in performance using a language model pre-trained with clinical embeddings compared to models pre-trained with general English corpora such as English Wikipedia and Brown corpus. The last implementation in Table 7, however, depicts the best performance achieved for this task. The ClinicalBERT model, which is trained on 2 million clinical notes in the MIMIC-III database, outperforms the previous implementations in all the target labels considered. Particularly, for the `BEFORE` and `OVERLAP` labels it achieves an F1-score of 89.01% and 83.21%, outperforming the BioBERT implementation by 1.59 and 1.54 percentage points, respectively. For the `AFTER` label, the ClinicalBERT configuration reached an F1-score of 36.69%, increasing the score by 2.65 percentage points compared to BioBERT. More importantly, this model overall attained an accuracy of 83.19, exceeding the performances of all the previously considered implementation. Although these results seem to be promising for the accomplishment of this task, one should notice that the achievements for the `AFTER` label are still far below consideration and can't be taken into account when building a patient's timeline. One of the possible reasons for this difference in performance can be attributed to the data availability for this specific label when compared to the `BEFORE` and `OVERLAP` labels, however, further research must be conducted in order to investigate the issue.

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| BEFORE | 87.31% | 86.05% | 86.68% | 15113 |
| OVERLAP | 72.28% | 91.42% | 80.73% | 9894 |
| AFTER | 64.51% | 4.50% | 8.41% | 2729 |
| Accuracy | - | - | **80.28%** | 27736 |
| Macro avg | 75.38% | 60.78% | 59.22% | 27736 |
| Weighted avg | 78.69% | 80.26% | 77.15% | 27736 |

Table 5: Results using 1-d CNN + BERT uncased model on the merged i2b2 2012 data set.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| BEFORE | 88.03% | 86.82% | 87.42% | 15113 |
| OVERLAP | 76.43% | 87.69% | 81.67% | 9894 |
| AFTER | 59.41% | 23.85% | 34.04% | 2729 |
| Accuracy | - | - | **82.21%** | 27736 |
| Macro avg | 72.95% | 64.38% | 68.18% | 27736 |
| Weighted avg | 79.91% | 80.79% | 80.24% | 27736 |

Table 6: Results using 1-d CNN + BioBERT cased model on the merged i2b2 2012 data set.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| BEFORE | 87.73% | 90.32% | 89.01% | 15113 |
| OVERLAP | 78.29% | 88.79% | 83.21% | 9894 |
| AFTER | 62.38% | 25.99% | 36.69% | 2729 |
| Accuracy | - | - | **83.19%** | 27736 |
| Macro avg | 76.21% | 67.87% | 70.32% | 27736 |
| Weighted avg | 82.27% | 82.79% | 82.11% | 27736 |

Table 7: Results using 1-d CNN + ClinicalBERT cased model on the merged i2b2 2012 data set.

As the benchmark for the TRE task, we considered the results from the paper by Tang et al. [40], whose system for the temporal relation extraction task was ranked first at the 2012 i2b2 challenge. Table 8 depicts some performances from previous works [25] and the comparison between the benchmark and the results from this work. In their work, Tang et al. [40] implemented the temporal relation extraction task into two different ways: a TLINK-only system, where they used the gold standard events and temporal expressions, and an end-to-end system, where instead of the gold standards events they used system-generated events. Both these systems rely on the combination of a rule-based and machine learning based approach. Specifically, the architecture implemented includes conditional random field (CRF) and support vector machine (SVM) combined with rule-based pair selection. The TLINK-only system was able to achieve an F1-score of 69.32%, outranking all the other implementations presented at the challenge. When compared to the implementation in this thesis, however, it is clear that the deep learning based approach outperforms the feature-based machine learning one, in fact, the F1-score of ClinicalBERT combined with 1-d CNN is 13.87 percentage points higher than the benchmark. These results confirmed once again that, although deep learning approaches have been only recently considered for information extraction tasks, they can be extremely useful and help improving accuracy as well as releasing the researcher from complex feature engineering.

| Approach | Precision | Recall | F1-score |
|---|---|---|---|
| ClinicalBERT + 1-d CNN | - | - | **83.19%** |
| CRF+SVM+Rule based by Tang et al. [40] | 71.43% | 67.33% | 69.32% |
| SVM by Lee et al. [25] | 63.93% | 63.62% | 63.77% |

Table 8: Performance comparison between different approaches for the Temporal Relation Extraction task on the merged i2b2 2012 test data set.

Following the same approach as for the above-mentioned experiments, we tested the same architectures on the unmerged i2b2 2012 data set. Table 9 reports the results for BERT/BioBERT/ClinicalBERT + 1-d CNN implementations. As expected, these performances, compared to those achieved on the merged data set, are considerably lower, in fact, if we consider the 1-d CNN + ClinicalBERT implementation, the overall accuracy reached by the model is only 65.36%, with 17.8 percentage points less than the same implementation on the merged version of the data set. This is not a surprise since the data available per label are different between the two versions of the data set, making the experiments relatively uncomparable. However, it is worth to notice that the results on the unmerged data set resemble the same trend as for those on the merged version, where ClinicalBERT outperforms the other BERT implementations. (For a more detailed overview of the results per TLINK type, see Appendix C.)

| | | Precision | Recall | F1-score |
|---|---|---|---|---|
| BERT uncased | Accuracy | - | - | 64.37% |
| | Macro avg | 58.35% | 50.23% | 49.40% |
| | Weighted avg | 65.88% | 65.41% | 63.37% |
| BioBERT cased | Accuracy | - | - | 65.36% |
| | Macro avg | 53.46% | 49.84% | 50.48% |
| | Weighted avg | 64.46% | 64.79% | 64.49% |
| ClinicalBERT cased | Accuracy | - | - | **66.46%** |
| | Macro avg | 55.89% | 52.24% | 53.36% |
| | Weighted avg | 65.34% | 67.10% | 65.26% |

Table 9: Results using 1-d CNN combined with BERT, BioBERT and ClinicalBERT models on the unmerged i2b2 2012 data set. The result in **bold** refers to the best achievement for this task.

Among previous work for the TRE task on the unmerged 2012 i2b2 data set, we considered the LSTM-based implementation by Patel et al. [35]. In this LSTM-based architecture, the input vector consists of three merged embedding vectors, one word vector for the temporal expression tags, one word vector for the event tags and one sentence vector. This merged single vector is applied as input to the LSTM model, which consists of three layers. The first LSTM layer uses *tanh* activation function for mapping words with the corresponding sentences and lowers the

dimension of words. In the second layer (dense layer) *sigmoid* or *ReLU* activation function is used. The last layer uses the *softmax* activation function which lowers the dimension of the vector and yields the final output. With this implementation, the authors claimed to achieve an F1-score of 78% on the i2b2 2012 unmerged data set. Although the authors declared to achieve state-of-the-art results on the unmerged data set, some of their work and results remain arguable and unclear, such as the real number of classes used in the experiments as well as the temporal relation types definitions, which differ from the official i2b2 2012 documentation. Moreover, in their project, no open-source implementation was shared, leaving the results hardly reproducible. Hence, a real comparison between the work in this thesis and the LSTM-based implementation by Patel et al. [35] cannot be established.

# 5 Discussion

The extensive testing performed on various configurations and settings has shown relevant results and has demonstrated that these models can effectively contribute towards the ultimate goal of creating a patient's clinical timeline. However, this large amount of data collected from the experiments needs to be summarised and evaluated. Therefore, this section briefly discusses the results achieved for both the NER and TRE tasks, focusing on their applications in real-world settings as well as their limitations. Moreover, to better understand the insights from this work, some examples of the results achieved are given, followed by the main take-away.


Fig. 13: Example of the extraction of `problem`, `test` and `treatment` entities on unseen clinical text using the best NER model.

For the NER task, the model implemented demonstrated to achieve interesting results on the 2010 i2b2 test data set, attaining similar performance to the benchmark considered (F1-score of 86.54% for the proposed model and F1-score of 88.13% for the benchmark considered). Figure 13 shows the application of the best NER model on an unseen clinical narrative. The highlighted text in red refers to `problem` entities, while blue and green highlighted text refer to `test` and `treatment` entities, respectively. As one can see, the effectiveness of the model in discriminating domain-specific entities is remarkable even with unseen clinical terminology.

Considering this task alone, we see a number of possible practical applications that this system can contribute to, from supporting physicians in extracting key entities to categorising entire clinical narratives and establishing new solid backgrounds for further NER research. The ability for a system to accurately extract this information from large medical text represents not only an important achievement in clinical NER but also a fundamental intermediate step for the success of further challenges, such as Temporal Relation Extraction in this work. Nonetheless, this task implemented has its limitations in real-world settings. An accuracy of about 87% on a domain-specific problem could be considered a significant end goal. However, from a medical point of view, these results are far not sufficient for a full autonomous deployment of the model in production since a margin of about 13% error in accuracy is still not acceptable, especially when it involves the health of a patient. Nevertheless, even though the performance of the model are not sufficient for a completely autonomous system, it can certainly provide support to the difficult decision-making process of a physician and positively contribute to the further development of clinical NER systems.

For the TRE task, the model considered was based on a 1-d CNN model combined with ClinicalBERT. The model showed good results on the 2012 i2b2 temporal relation extraction corpus, being able to classify the three TLINK types, `BEFORE`, `OVERLAP` and `AFTER`, with an overall F1-score of 83.19% on the test set.

| | | **PREDICTION** | | | **Total** |
|---|---|---|---|---|---|
| | | BEFORE | OVERLAP | AFTER | n = 27736 |
| **ACTUAL** | BEFORE | 13530 | 1321 | 262 | 15113 |
| | OVERLAP | 933 | 8778 | 183 | 9894 |
| | AFTER | 911 | 1105 | 713 | 2729 |

Table 10: Confusion matrix for the three TLINK types from the 1-d CNN + ClinicalBERT classification on the 2012 i2b2 data set. Rows represent the actual classes, while columns represent the predicted classes.

To better understand the results from this task, Table 10 depicts the confusion matrix for the classification of the three TLINK types using the above-mentioned model. As one might notice, the classification performance varies between TLINK type. With 90% and 89% of true positives respectively, the `BEFORE` and `OVERLAP` labels are fairly straightforward for the model to correctly classify, whereas, the `AFTER` label, with only 26% of correct classifications, seems to be more challenging for the model to categorise, representing one of the limitations of this task. One

of the reasons for this low performance on the AFTER label can be tied to the unbalance of data between labels used in the training process, however, further investigation is needed to address this issue.

In order to have an idea of the implications of this task for the practical application as well as gain some insights from the results, we consider as an example the following medical narrative extract from the 2012 i2b2 data set.

Fig. 14: Clinical narrative extract from the 2012 i2b2 data set. The yellow-marked text refers to clinical entities, namely TEST, TREATMENT and PROBLEM, while the green-marked text relates to temporal expressions.

*Admission Date : 2015-08-10 Discharge Date : 2015-08-15 [...] The patient was taken to the Operating Room the day of admission . He underwent a left bronchoscopy and lower lobectomy of the left side. [...] The patient was then transferred to the CSRU, intubated in stable condition. [...] The patient remained to be stable on the floor with subsequent monitoring of hematocrit within a normal stable range. He is successfully extubated on 08-12 and hypertension transferred to the floor in stable condition where the Pain Service is managing his epidural with very good effect and he is tolerating a regular p.o. diet and is making adequate amount of urine. His recovery was essentially unremarkable. His epidural was successfully discontinued and he is discharged to home on 08-15 with instructions to follow-up with Dr. Rodriguez in the office within the next one to two weeks. He is discharged to home with pain medication which is percocet.*

Figure 15 represents the constructed timeline of some of the events extracted from the above clinical narrative and their relations to the temporal expressions. To have a better understanding of the timeline structure, Table 11 shows the extracted entities and the relation types classified by the model referring to the example at hand. Specifically, for each clinical entity and temporal expression pair, the model assigns the relation type of BEFORE if the first entity refers to a period of time preceding the second entity, OVERLAP if the first entity refers to a period of time overlapping with the second entity and AFTER if the first entity refers to a period of time following the second entity. For instance, reading Figure 15 from left to right, the relation between the TEST entities left bronchoscopy, lower lobectomy and intubated and the temporal expression the day of the admission, corresponding to the 2015-08-10, is classified by the model as OVERLAP. Similarly, the following entities, such as extubated and epidural are successfully related to 08-12 and 08-15 with TLINK types OVERLAP and BEFORE, respectively. Finally, the TREATMENT entity percocet is related to the temporal expression 08-15 with a type AFTER, since this medication was prescribed after the patient was discharged on the 08-15.

| Entity 1 Type | Chunk 1 | Entity 2 Type | Chunk 2 | Relation Type |
|---|---|---|---|---|
| TEST | `left bronchoscopy` | DATE | `the day of admission` | OVERLAP |
| TEST | `lower lobectomy` | DATE | `the day of admission` | OVERLAP |
| TEST | `intubated` | DATE | `the day of admission` | OVERLAP |
| TEST | `intubated` | DATE | `08-12` | BEFORE |
| TEST | `extubated` | DATE | `08-12` | OVERLAP |
| TEST | `epidural` | DATE | `08-15` | OVERLAP |
| TREATMENT | `percocet` | DATE | `08-15` | AFTER |

Table 11: Classification of the relation type (TLINK) between extracted entities from the text in Figure 14. Columns 1 to 4 represent the entity types followed by the corresponding text. The last column defines the classified relation types.

This example, in summary, represents the important implications that this task has in the real-world implementations. The ability of a model to accurately discriminate the chronological relation between entities and temporal expression can have a large number of applications in the medical field and can certainly help physicians not only on difficult decision-making tasks but also in making accurate predictions based on the patient's chronological history. However, although this model achieved important results on the 2012 i2b2 data set, it also bears some limitations. As for the NER task, also in this case even with a model achieving an accuracy of 83.19% on unseen data, the model is not fully autonomous to deploy it in complex systems such as the healthcare one, but, as previously mentioned, it can definitely assist physicians in many daily situations. It is also important to notice that this timeline was constructed from the predictions of the model, however, for the model to physically build the timeline from the categorised relations, further implementations, that were not possible in this work due to time constraints, are needed.

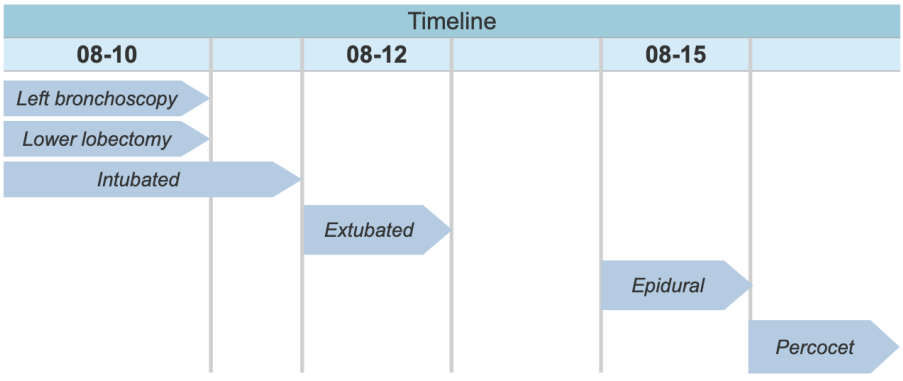



Fig. 15: Example of a constructed timeline with the temporal events and clinical entities extracted from the above-given medical narrative.

# 6 Conclusion & Further Research

Information Extraction, and specifically Temporal Relation Extraction from a large bulk of raw clinical text data, represents a challenging and crucial task in the biomedical research area. Temporal Information Extraction used to process patient information in clinical narratives can positively contribute to the process of making accurate decisions in vital patient care tasks such as forecasting the effects of therapies, preventing the spread of a disease or diagnosing the nature of a medical condition. Moreover, constructing a reliable and accurate timeline can facilitate physicians during the process of making important decisions and help increase medical accuracy.
In this thesis, we attempted to develop and evaluate an NLP pipeline that is able to pre-process raw clinical narratives, extract clinical named entities and relate them to specific temporal entities. Specifically, we firstly showed that our CoNLL parser is able to pre-process and transform raw clinical notes into training set following the CoNLL-2003 file format with an accuracy close to the licensed version in SparkNLP. Secondly, we tried to address the following research questions:

1. What is the performance of a Named Entity Recognition method trained to specifically extract clinical entities such as treatments, tests and diseases from medical narratives?

From the results of our experiments, we demonstrated that the trained Named Entity Recognition model is able to extract `problem`, `treatment` and `test` entities from clinical narratives with an F1-score of 86.54%, close to the state-of-the-art implementation by Zhang et al. [50]. Moreover, we discussed the implications of this work in practical applications and we showed some examples of the results achieved, followed by main insights.

2. To what extent is Temporal Relation Extraction able to efficiently categorise temporal relations between the extracted clinical entities? Specifically, to what degree are BERT methods more suitable for this task compared to other feature-based methods?

To address this research question, we implemented a Temporal Relation Extraction model based on Chen et al. [8] BERT + 1-d CNN model architecture, to which different versions of BERT were applied. We demonstrated that 1-d CNN combined with ClinicalBERT pre-trained model achieved an F1-score of 83.19% on the merged 2012 i2b2 data set and an F1-score of 66.46% on the unmerged 2012 i2b2 data set. These results showcase the ability of this NLP pipeline to effectively process raw medical narratives, extract fundamental clinical entities with a performance close to state-of-the-art systems and accurately relate them to temporal events. Moreover, for the TRE task in clinical text, we proved the effectiveness of BERT models compared to previous feature-based methods, showing once again the potentiality of deep learning based approaches. Until recently, the majority of NLP problems have been addressed with shallow machine learning models and hand-crafted, time-consuming features. However, new projects and research in NLP, such as recent achievements of BERT models or studies like this work, are revealing positive prospects for employment of deep learning methods which proved to achieve superior results in several NLP tasks compared to more traditional machine learning methods such as logistic regression or SVM.

Future research could look into the optimisation and improvements of part of the tasks implemented for this paper. First, as for the pre-processing task, a closer look should be given to the implementation of the CoNLL training file which perhaps represents one of the most important processes for the achievement of higher performances. Specifically, a more efficient procedure, that was not possible here due to time constraints, should be implemented in the token-tags matching task in order to potentially scale up this project to production. Furthermore, for the NER task, an interesting improvement would be to apply different word embeddings when training the NER model. Recently, SparkNLP, on its latest 2.5.0 release, published new word embeddings specifically for biomedical and clinical words, such as `biobert_clinical_base_cased` and `biobert_pubmed_pmc_base_cased` among many, which can significantly help improve the model accuracy.

# References


[1] Alimova, I., Tutubalina, E.: Multiple features for clinical relation extraction: A machine learning approach. Journal of Biomedical Informatics **103**, 103382 (2020). https://doi.org/https://doi.org/10.1016/j.jbi.2020.103382, http://www.sciencedirect.com/science/article/pii/S1532046420300095

[2] Alsentzer, E., Murphy, J.R., Boag, W., Weng, W.H., Jin, D., Naumann, T., McDermott, M.B.A.: Publicly available clinical bert embeddings (2019)

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

[4] Augusto, J.C.: Temporal reasoning for decision support in medicine. Artificial intelligence in medicine **33**(1), 1–24 (2005)

[5] Bethard, S., Savova, G., Palmer, M., Pustejovsky, J.: Semeval-2017 task 12: Clinical tempeval. pp. 565–572 (01 2017). https://doi.org/10.18653/v1/S17-2093

[6] Brown, P., Dellapietra, V., Souza, P., Lai, J., Mercer, R.: Class-based n-gram models of natural language. Computational Linguistics **18**, 467–479 (01 1992)

[7] Chen, Q., Peng, Y., Lu, Z.: Biosentvec: creating sentence embeddings for biomedical texts (2018)

[8] Chen, T., Wu, M., Li, H.: A general approach for improving deep learning-based medical relation extraction using a pre-trained model and fine-tuning. Database : the journal of biological databases and curation **2019** (01 2019). https://doi.org/10.1093/database/baz116

[9] Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional lstm-cnns (2015)

[10] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of machine learning research **12**(ARTICLE), 2493–2537 (2011)

[11] Corporation, H.P.: Survey of natural language processing techniques in bioinformatics. Computational and Mathematical Methods in Medicine **2015**, 674296 (2015). https://doi.org/10.1155/2015/674296, https://doi.org/10.1155/2015/674296

[12] Denny, J.C., Irani, P.R., Wehbe, F.H., Smithers, J.D., Spickard III, A.: The knowledgemap project: development of a concept-based medical school curriculum database. In: AMIA Annual Symposium Proceedings. vol. 2003, p. 195. American Medical Informatics Association (2003)

[13] Deshpande, T., Butte, A.: Exploiting drug-disease relationships for computational drug repositioning. Briefings in bioinformatics **12**, 303–11 (06 2011). https://doi.org/10.1093/bib/bbr013

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

[15] Friedman, C.: Towards a comprehensive medical language processing system: methods and issues. In: Proceedings of the AMIA annual fall symposium. p. 595. American Medical Informatics Association (1997)

[16] Galvan, D., Okazaki, N., Matsuda, K., Inui, K.: Investigating the challenges of temporal relation extraction from clinical text. In: Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis. pp. 55–64. Association for Computational Linguistics, Brussels, Belgium

(Oct 2018). https://doi.org/10.18653/v1/W18-5607, https://www.aclweb.org/anthology/W18-5607
[17] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P., Mark, R., Mietus, J., Moody, G., Peng, C.K., Stanley, H.: Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals. Circulation **101**, E215–20 (07 2000). https://doi.org/10.1161/01.CIR.101.23.e215
[18] Horev, R.: Bert explained: State of the art language model for nlp, https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
[19] Hunter, J.D.: Matplotlib: A 2d graphics environment. Computing in science & engineering **9**(3), 90–95 (2007)
[20] Kim, Y.: Convolutional neural networks for sentence classification (2014)
[21] Kocaman, V.: Named Entity Recognition (NER) with BERT in Spark NLP. https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/blogposts/3.NER_with_BERT.ipynb
[22] Labs, J.S.: Spark nlp: State of the art natural language processing. https://nlp.johnsnowlabs.com/docs/en/concepts, accessed: 2020-01-20
[23] Lafferty, J., Mccallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pp. 282–289 (01 2001)
[24] Lee, H.J., Xu, H., Wang, J., Zhang, Y., Moon, S., Xu, J., Wu, Y.: Uthealth at semeval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. pp. 1292–1297 (01 2016). https://doi.org/10.18653/v1/S16-1201
[25] Lee, H.J., Zhang, Y., Jiang, M., Xu, J., Tao, C., Xu, H.: Identifying direct temporal relations between time and events from clinical notes. BMC Medical Informatics and Decision Making **18** (07 2018). https://doi.org/10.1186/s12911-018-0627-5
[26] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics (Sep 2019). https://doi.org/10.1093/bioinformatics/btz682, http://dx.doi.org/10.1093/bioinformatics/btz682
[27] Leeuwenberg, A., Moens, M.F.: Temporal information extraction by predicting relative time-lines (2018)
[28] Li, X., Zhu, D., Levy, P.: Predicting clinical outcomes with patient stratification via deep mixture neural networks. AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science **2020**, 367–376 (05 2020)
[29] Lin, C., Miller, T., Dligach, D., Bethard, S., Savova, G.: Improving temporal relation extraction with training instance augmentation. In: Proceedings of the 15th Workshop on Biomedical Natural Language Processing. pp. 108–113. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/W16-2914, https://www.aclweb.org/anthology/W16-2914
[30] Loper, E., Bird, S.: Nltk: The natural language toolkit. In: In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics (2002)
[31] Magge, A., Scotch, M., Gonzalez-Hernandez, G.: Clinical ner and relation extraction using bi-char-lstms and random forest classifiers. In: Liu, F., Ja-

gannatha, A., Yu, H. (eds.) Proceedings of the 1st International Workshop on Medication and Adverse Drug Event Detection. Proceedings of Machine Learning Research, vol. 90, pp. 25–30. PMLR (04 May 2018), http://proceedings.mlr.press/v90/magge18a.html

[32] McKinney, W., et al.: Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. vol. 445, pp. 51–56. Austin, TX (2010)

[33] Nikfarjam, A., Emadzadeh, E., Gonzalez, G.: Towards generating a patient's timeline: Extracting temporal relationships from clinical notes. Journal of Biomedical Informatics **46**, S40 – S47 (2013). https://doi.org/https://doi.org/10.1016/j.jbi.2013.11.001, http://www.sciencedirect.com/science/article/pii/S1532046413001779, 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data

[34] Oliphant, T.E.: A guide to NumPy, vol. 1. Trelgol Publishing USA (2006)

[35] Patel, R., Tanwani, S.: Temporal relation identification from clinical text using lstm based deep learning model. International Journal of Research and Analytical Reviews (IJRAR) (2018)

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

[37] Piskorski, J., Laskova, L., Marciczuk, M., Pivovarova, L., Priban, P., Steinberger, J., Yangarber, R.: The second cross-lingual challenge on recognition, normalization, classification, and linking of named entities across slavic languages. pp. 63–74 (01 2019). https://doi.org/10.18653/v1/W19-3709

[38] Savova, G., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Schuler, K., Chute, C.: Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. Journal of the American Medical Informatics Association : JAMIA **17 5**, 507–13 (2010)

[39] Sun, W., Rumshisky, A., Uzuner, O.: Evaluating temporal relations in clinical text: 2012 i2b2 challenge. Journal of the American Medical Informatics Association : JAMIA **20** (04 2013). https://doi.org/10.1136/amiajnl-2013-001628

[40] Tang, B., Wu, Y., Jiang, M., Chen, Y., Denny, J.C., Xu, H.: A hybrid system for temporal information extraction from clinical text. Journal of the American Medical Informatics Association **20**(5), 828–835 (04 2013). https://doi.org/10.1136/amiajnl-2013-001635, https://doi.org/10.1136/amiajnl-2013-001635

[41] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research **6**, 1453–1484 (09 2005)

[42] Uzuner, .z., South, B.R., Shen, S., DuVall, S.L.: 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. Journal of the American Medical Informatics Association **18**(5), 552–556 (06 2011). https://doi.org/10.1136/amiajnl-2011-000203, https://doi.org/10.1136/amiajnl-2011-000203

[43] UzZaman, N., Llorens, H., Allen, J., Derczynski, L., Verhagen, M., Pustejovsky, J.: Tempeval-3: Evaluating events, time expressions, and temporal relations. arXiv (06 2012)

[44] Wang, M., Manning, C.D.: Effect of non-linear deep architecture in sequence labeling. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing. pp. 1285–1291 (2013)

[45] Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D.C., Augspurger, T., Halchenko, Y., Cole, J.B., Warmenhoven, J., de Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M., Meyer, K., Miles, A., Ram, Y., Yarkoni, T., Williams, M.L., Evans, C., Fitzgerald, C., Brian, Fonnesbeck, C., Lee, A., Qalieh, A.: mwaskom/seaborn: v0.8.1 (september 2017) (Sep 2017). https://doi.org/10.5281/zenodo.883859, https://doi.org/10.5281/zenodo.883859

[46] Wu, Y., Jiang, M., Xu, J., Zhi, D., Xu, H.: Clinical named entity recognition using deep learning models. AMIA ... Annual Symposium proceedings. AMIA Symposium **2017**, 1812–1819 (2017)

[47] Yijia, Z., Chen, Q., Yang, Z., Lin, H., lu, Z.: Biowordvec, improving biomedical word embeddings with subword information and mesh. Scientific Data **6** (12 2019). https://doi.org/10.1038/s41597-019-0055-0

[48] Yu, M., Yin, W., Hasan, K.S., dos Santos, C., Xiang, B., Zhou, B.: Improved neural relation detection for knowledge base question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 571–581. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1053, https://www.aclweb.org/anthology/P17-1053

[49] Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., Ghodsi, A., Gonzalez, J., Shenker, S., Stoica, I.: Apache spark: A unified engine for big data processing. Commun. ACM **59**(11), 5665 (Oct 2016). https://doi.org/10.1145/2934664, https://doi.org/10.1145/2934664

[50] Zhang, Y., Zhang, Y., Qi, P., Manning, C.D., Langlotz, C.: Biomedical and clinical english model packages in the stanza python nlp library. ArXiv **abs/2007.14640** (2020)

[51] Zhou, L., Hripcsak, G.: Temporal reasoning with medical data - a review with emphasis on medical natural language processing. Journal of Biomedical Informatics **40**(2), 183 – 202 (2007). https://doi.org/https://doi.org/10.1016/j.jbi.2006.12.009, http://www.sciencedirect.com/science/article/pii/S1532046407000032

# A Appendix - Python Information

In order to make use of the implemented techniques the version of Python has to be 3.7.1 or higher. Setups with Python versions below 3.7.1 were not tested.
The following Python libraries were used with their recommended versions:

1. PySpark 2.4.5 [49]
2. Spark 2.4.5 [49]
3. SparkNLP 2.4.3 [22]
4. Numpy 1.17.2 [34]
5. Pandas 0.25.1 [32]
6. Nltk 3.4.5 [30]

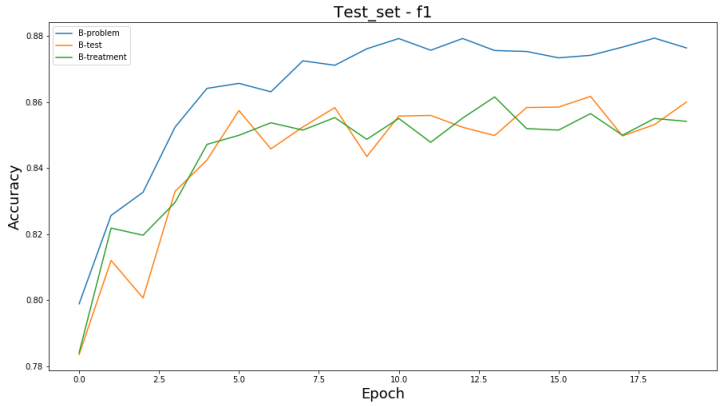
Though not necessary for making use of the NER and Temporal Relation Extraction algorithms, the following libraries are useful for plotting and creating tables:

1. Matplotlib 3.1.1 [19]
2. Seaborn 0.10.1 [45]

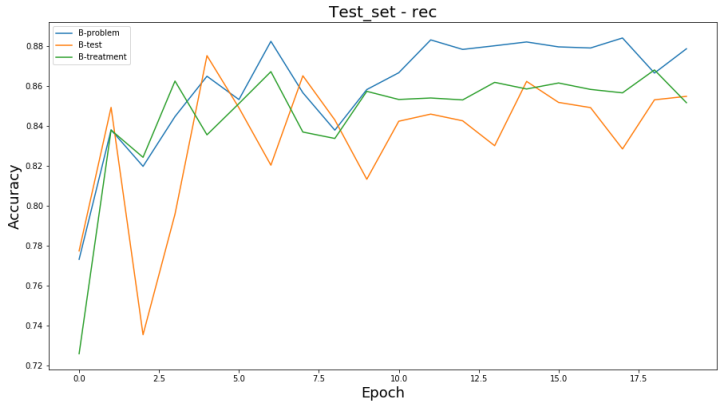
# B Implementations availability

The python implementations of the CoNLL parser, NER model training and Temporal Relation Extraction tool and additional functions are made available and can be found here: b Page Link
Instructions on how to run the python scripts as well as a documentation of all the available functions are included.

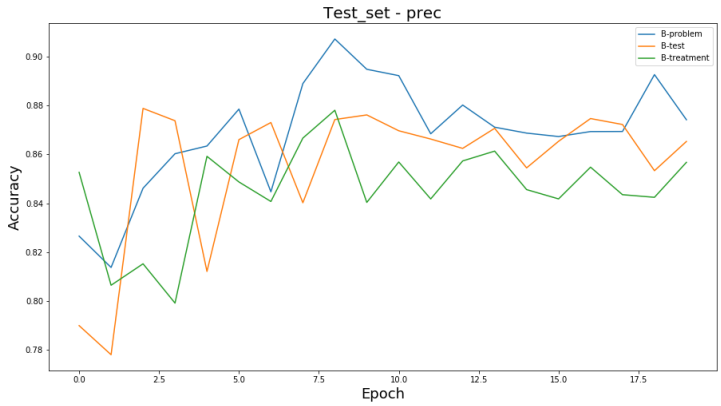
# C Further visualisation and results for both NER and TRE tasks




(a) F1-score on the test set for the 3 named entities: `B-problem`, `B-test` and `B-treatment`.




(b) Recall on the test set for the 3 named entities: `B-problem`, `B-test` and `B-treatment`.




(c) Precision on the test set for the 3 named entities: `B-problem`, `B-test` and `B-treatment`.

Fig. 16: Performances of the model with same parameters as the best NER model, however, trained with batch size of 64

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| ENDED_BY | 38.34% | 48.41% | 42.73% | 688 |
| BEFORE_OVERLAP | 59.64% | 37.08% | 45.73% | 3636 |
| BEFORE | 80.45% | 86.89% | 83.55% | 10789 |
| BEGUN_BY | 76.03% | 10.45% | 18.37% | 788 |
| DURING | 47.34% | 69.48% | 56.31% | 875 |
| OVERLAP | 47.34% | 69.48% | 56.31% | 4877 |
| AFTER | 40.48% | 24.37% | 30.42% | 1941 |
| SIMULTANEOUS | 72.89% | 56.12% | 63.42% | 4142 |
| Accuracy | - | - | **64.37%** | 27736 |
| Macro avg | 58.35% | 50.23% | 49.40% | 27736 |
| Weighted avg | 65.88% | 65.41% | 63.37% | 27736 |

Table 12: Results using 1-d CNN + BERT uncased model on the unmerged i2b2 2012 data set.

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| ENDED_BY | 38.19% | 25.88% | 30.85% | 688 |
| BEFORE_OVERLAP | 63.39% | 38.72% | 48.07% | 3636 |
| BEFORE | 78.76% | 89.28% | 83.69% | 10789 |
| BEGUN_BY | 34.96% | 45.31% | 39.47% | 788 |
| DURING | 56.38% | 51.87% | 54.03% | 875 |
| OVERLAP | 50.45% | 55.68% | 52.94% | 4877 |
| AFTER | 39.45% | 21.87% | 28.14% | 1941 |
| SIMULTANEOUS | 63.68% | 69.08% | 66.27% | 4142 |
| Accuracy | - | - | **65.36%** | 27736 |
| Macro avg | 53.46% | 49.84% | 50.48% | 27736 |
| Weighted avg | 64.46% | 64.79% | 64.49% | 27736 |

Table 13: Results using 1-d CNN + BioBERT cased model on the unmerged i2b2 2012 data set.

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| ENDED_BY | 42.38% | 42.01% | 42.19% | 688 |
| BEFORE_OVERLAP | 61.43% | 39.78% | 48.28% | 3636 |
| BEFORE | 78.45% | 92.78% | 85.02% | 10789 |
| BEGUN_BY | 50.31% | 38.69% | 43.74% | 788 |
| DURING | 54.78% | 61.45% | 57.92% | 875 |
| OVERLAP | 54.15% | 52.61% | 53.37% | 4877 |
| AFTER | 50.16% | 18.95% | 27.51% | 1941 |
| SIMULTANEOUS | 62.87% | 70.21% | 66.34% | 4142 |
| Accuracy | - | - | **66.46%** | 27736 |
| Macro avg | 55.89% | 52.24% | 53.36% | 27736 |
| Weighted avg | 65.34% | 67.10% | 65.26% | 27736 |

Table 14: Results using 1-d CNN + ClinicalBERT cased model on the unmerged i2b2 2012 data set.