UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN BIOINGEGNERIA E BIOINFORMATICA
XXX CICLO - 2017

INFORMATION EXTRACTION FROM MEDICAL REPORTS IN THE ITALIAN LANGUAGE FOR CLINICAL TIMELINES RECONSTRUCTION

PhD Thesis by
NATALIA VIANI

Advisor:
Prof. Lucia Sacchi

PhD Program Chair:
Prof. Riccardo Bellazzi

Abstract (Italiano)

Nel corso degli ultimi anni, l'impiego di cartelle cliniche elettroniche si sta largamente diffondendo, costituendo una sempre più grande fonte di informazioni rilevanti sia per la cura del paziente che per la ricerca in campo biomedico. Nonostante gli sforzi profusi nella raccolta di dati strutturati, che possono essere facilmente consultati ed interrogati, molte informazioni sono disponibili soltanto in forma di testo libero. Per questo motivo, sviluppare sistemi per l'estrazione automatica di informazioni rilevanti dai testi clinici è fondamentale. Inoltre, riassumere tutti i dati disponibili per un paziente – che potrebbero essere sparsi in diversi documenti testuali – rappresenta un obiettivo essenziale.

Nell'ambito dei sistemi di estrazione di informazioni cliniche, sono stati sviluppati diversi strumenti, soprattutto per l'analisi di testi scritti in lingua inglese. Tuttavia, l'attività di ricerca per lingue diverse dall'inglese è ancora limitata. In questa tesi, diverse tecniche di estrazione di informazioni e alcuni metodi di sintesi sono stati applicati all'analisi di referti medici scritti in italiano. Per questa lingua, non è facile avere a disposizione risorse condivise per l'estrazione di informazioni cliniche. In questo lavoro, è stato utilizzato un corpus di referti di cardiologia molecolare come dataset principale per lo sviluppo di metodi di analisi. Inoltre, per permettere la realizzazione e la validazione di diverse soluzioni, un sottoinsieme di questo corpus è stato *annotato*, identificando manualmente le informazioni da estrarre dai testi. Per facilitare quest'attività di annotazione, sono state sviluppate delle linee guida specifiche.

Per accedere alla conoscenza contenuta nei referti medici testuali, un primo passo riguarda l'identificazione di eventi clinici. Nel campo del *natural language processing*, questo compito è spesso svolto mediante l'uso di metodi di apprendimento supervisionato. In questa attività di ricerca, sono stati sfruttati due diversi approcci per l'estrazione di eventi dal testo. Innanzitutto, è stato utilizzato un approccio semplice ma efficace basato su ricerca tramite dizionari esterni. In secondo luogo, è stata investigata un'applicazione delle reti neurali ricorrenti.

Nei testi clinici, gli eventi sono inoltre spesso menzionati in associazione con attributi di interesse medico, che devono essere estratti per caratterizzare l'evento stesso. In questa tesi, è stato utilizzato un approccio guidato da ontologie per identificare gli attributi degli eventi contenuti nei referti di cardiologia molecolare. In particolare, è stata sviluppata manualmente un'ontologia specifica del dominio, contenente

tutti gli eventi rilevanti con i relativi attributi. Come *gold standard* per la fase di validazione, è stato sfruttato un database ospedaliero, che registra la maggior parte delle informazioni scritte nei referti.

Un altro compito importante per una corretta ricostruzione delle storie cliniche dei pazienti è l'associazione di un tempo specifico ad ogni evento estratto dal testo. A questo scopo, un primo passo è dato dall'identificazione delle espressioni temporali contenute nel testo stesso. In questa attività di ricerca, due sistemi esistenti per l'estrazione di informazioni temporali sono stati riadattati all'analisi dei testi clinici. Il primo sistema è basato su un insieme di regole scritte a mano, mentre il secondo sfrutta una grammatica formale.

Per processare ogni documento, i tre step illustrati (estrazione di eventi, attributi ed espressioni temporali) sono stati aggregati in una pipeline. Come nota importante, per ciascun evento ed espressione temporale identificati nel testo, la pipeline estrae anche alcune proprietà di interesse (e.g., la polarità dell'evento). Tra queste proprietà, viene ricavata la relazione (*DocTimeRel*) tra ciascun evento e la data di creazione del documento. Sulle basi di questa relazione, ogni evento viene poi associato ad un tempo di riferimento (la data di creazione del documento o un'altra espressione temporale), attraverso l'applicazione di un insieme di regole costruite manualmente.

Oltre a processare singoli referti medici, il sistema sviluppato in quest'attività di ricerca è in grado di sintetizzare le informazioni contenute in documenti diversi ma riferiti ad uno stesso paziente. In questo caso, la pipeline per l'estrazione di informazioni viene inizialmente impiegata per processare i singoli documenti riguardanti quel paziente. In seguito, il sistema costruisce e visualizza una timeline con tutti gli eventi estratti, sfruttando l'informazione data dal DocTimeRel e le associazioni evento-tempo di riferimento.

Per quanto riguarda la fase di validazione, la pipeline sviluppata ha ottenuto buoni risultati sul corpus italiano considerato. Partendo dall'estrazione di eventi, il classificatore basato su reti neurali ricorrenti ha mostrato una buona performance. In particolare, combinando questo metodo con la ricerca basata su dizionari esterni, la pipeline ha ottenuto i risultati migliori su tutti gli esperimenti condotti. In merito all'estrazione di attributi, l'approccio guidato da ontologie ha ottenuto buoni risultati, con accuratezze elevate per la maggior parte degli eventi clinici. Inoltre, è stata verificata la possibilità di adattare questo step all'analisi di un'altra lingua (i.e., inglese), con risultati promettenti. In modo simile, l'ontologia è stata adattata all'analisi di un altro dominio clinico (i.e., oncologia), portando alla realizzazione di un sistema di estrazione con buone prestazioni. Infine, per quanto riguarda l'estrazione di espressioni temporali, sono stati ottenuti buoni risultati riadattando opportunamente i due sistemi considerati.

In merito alla validazione dell'attività di sintesi, è stata condotta una valutazione preliminare analizzando la timeline ricostruita per un singolo paziente. Quest'analisi ha mostrato come il sistema abbia le potenzialità per essere impiegato come un efficace strumento per esaminare le storie

cliniche dei pazienti, riducendo il tempo richiesto per accedere a grandi quantità di dati.

In conclusione, i risultati ottenuti indicano che gli approcci indagati possono rappresentare una buona strategia per estrarre informazioni da testi clinici scritti in lingue diverse dall'inglese. Per quanto riguarda le applicazioni pratiche della ricerca condotta, si sta attualmente lavorando all'integrazione in due archivi di ricerca basati sul sistema i2b2 delle informazioni estratte tramite la pipeline sviluppata.

Abstract (English)

Electronic health records have been widely adopted over the years, representing a great source of valuable information for both patient care and biomedical research. Despite the efforts put into collecting structured data, which can be easily accessed and queried, a lot of information is available only in the form of free text. For this reason, developing systems that automatically extract relevant information from clinical narratives is essential. In addition, summarizing all the data related to one single patient – maybe scattered across multiple textual documents – represents an essential task.

In the field of clinical information extraction, several systems have been developed, especially for the analysis of texts written in English. However, the related research for non-English languages is still limited. In this research activity, information extraction techniques and summarization methods were applied to the analysis of medical reports written in Italian. For this language, shared resources for clinical information extraction are not easily available. In this work, a corpus of molecular cardiology reports was considered as the main dataset for methods development. Moreover, to enable the design and the evaluation of different approaches, a subset of this corpus was *annotated* by manually identifying the information to be extracted from the texts. To enable this annotation task, specific guidelines were developed.

To access the knowledge included in textual medical reports, a first step involves the identification of clinical events. In the natural language processing community, this task is often addressed by using supervised methods. In this research activity, two different approaches were exploited to perform event extraction. First, a simple, yet effective approach based on dictionary lookup was used. Second, an application of recurrent neural networks was investigated.

In clinical texts, events are often mentioned together with relevant attributes that have to be extracted to characterize the event itself. In this thesis, an ontology-driven approach was used to identify events' attributes in the molecular cardiology reports. In particular, a domain-specific ontology was manually developed, including all the relevant events with their associated attributes. As the gold standard for the evaluation phase, a hospital database, which stores most of the information written in the reports, was exploited.

As another important task, to correctly reconstruct patients' clinical histories, it is necessary to assign a specific time to each event extracted from the text. To this end, the identification of temporal expressions is a

first, mandatory step. In this research activity, two existing systems for temporal information extraction were adapted to the analysis of clinical narratives. The first system relies on a set of hand-written rules, while the second one makes use of a formal grammar.

To process each document, the three illustrated steps (event, attribute, and temporal expression extraction) were aggregated into a pipeline. As an important remark, for each event and temporal expression identified in the text, the pipeline extracts a few properties of interest (e.g., the event's polarity), too. Among these properties, the temporal relation between each event and the document creation time is computed (*DocTimeRel*). On the basis of this relation, each event is further linked to a reference time (either the document creation time or another time expression) by applying a set of hand-crafted rules.

Besides processing single medical reports, the system developed in this research activity is able to summarize multiple documents referred to the same patient. In this case, the information extraction pipeline is initially run on all the documents belonging to that patient. Then, the system builds and visualizes a timeline of all the extracted events, exploiting the DocTimeRel information and the event-time links.

As regards the system's evaluation, the overall information extraction pipeline performed well on the considered Italian cardiology corpus. Starting from the event extraction task, the recurrent neural network classifier achieved a good performance. In particular, by combining this method with the dictionary lookup approach, the pipeline obtained the best results across all experiments. With respect to the attribute extraction task, the proposed ontology-driven approach performed well, with high accuracies for most clinical events. In addition, the possibility to adapt this step to the analysis of another language (i.e., English) was assessed, with promising results. In a similar way, the developed ontology was adapted to the analysis of another clinical domain (i.e., oncology), leading to a well-performing extraction system. Finally, as regards the extraction of time expressions, good results were achieved by properly adapting the two investigated systems.

With respect to the summarization task, a preliminary evaluation was conducted by analyzing the timeline reconstructed for one single patient. This preliminary validation showed that the summarization system could serve as an effective tool for reviewing patients' clinical histories, reducing the time needed to access large amounts of data.

In conclusion, the obtained results indicate that the investigated approaches can be a good strategy to extract information from clinical texts in non-English languages. As regards the practical applications of the conducted research, the information extracted through the developed pipeline is currently being integrated into two research repositories based on the i2b2 system.

Contents

Chapter 1

Introduction

1.1. Clinical natural language processing

Thanks to the rapid adoption of information technologies in the clinical setting, the amount of patient-related information available in electronic form is growing incredibly fast. Enabling timely access to this clinical information is of paramount importance for several reasons, above all to improve patient care and to facilitate knowledge discovery. For example, the data included in electronic health records can be effectively reused to strengthen evidence-based medicine and to evaluate the quality of healthcare.

Although electronic health records facilitate the storage of structured data, which can be queried and processed in an automatic way, they also include much information in the form of narrative text. Despite the availability of this rich textual content, performing manual inspections to draw meaningful conclusions is expensive in terms of time and resources. On the other hand, automatically performing queries to access information of interest is not straightforward, due to the unstructured nature of the information.

For the above mentioned reasons, developing automatic natural language processing (NLP) techniques represents a necessary step towards the full exploitation of all available clinical data [1]. Following the definition by Hirschberg et al., natural language processing can be described as follows [2]:

> *<<Natural language processing is the subfield of computer science concerned with using computational techniques to learn, understand, and produce human language content.>>*

As regards the clinical domain, by applying NLP techniques on natural language narratives, all portions of medical records can be transformed to structured data, which can be later processed and examined in an automatic way.

As a main task in clinical NLP, information extraction (IE) deals with the automatic identification of a predefined set of concepts in a specific domain [3]. For example, to record the outcomes of patient visits and diagnostic procedures, physicians write medical reports containing relevant findings, diagnostic conclusions, and therapeutic considerations. To be accessed and reused for monitoring and decision making, these concepts have to be extracted in a reliable way, also taking into consideration the context in which the single mention occurs [1]. For instance, negated findings (e.g., "no evidence of ventricular arrhythmias") or hypothetical diagnoses (e.g., "suspected Brugada Syndrome") should be treated in a different way with respect to affirmed facts. As another aspect to be considered, it is important to capture possible relations between a certain concept and other relevant elements in the text. For example, whenever a medication name occurs in the text, it would be important to search for possible related dosages and frequencies of assumption.

A single medical report intrinsically reflects a certain time in the patient's history. However, it also contains references to both previous facts (i.e., that happened in the past), or scheduled events (i.e., that will happen in the future). Therefore, extracting information from medical reports requires addressing two complementary tasks. First, all the relevant concepts related to the patient's history need to be identified. From now on, these concepts will be referred to as *events*. As a second step, each event has to be contextualized from the temporal point of view, as the clinical information included in an electronic record is only significant in a certain temporal context [4]. To analyze the temporal aspects of narratives, the extraction and the normalization of the temporal expressions included in reports is required. By linking the events and the temporal expressions found in the text, it becomes possible to build a system that automatically "understands" not only *if* a certain event happened, but also *when* the event took place.

From a practical point of view, clinical IE techniques allow automatically processing patient-related narratives to search for relevant data. In the clinical setting, as a result of multiple ambulatory visits and hospital stays, many documents can be produced over time for each patient. In light of this consideration, another important step to support patient monitoring and decision making consists in clinical summarization. From the work by Feblowitz et al. [5]:

> *<<Clinical summarization can be defined as the act of collecting, distilling, and synthesizing patient information for the purpose of facilitating any of a wide range of clinical tasks. >>*

Collecting and synthesizing the information included in free text, indeed, can effectively enhance the process of reviewing and making sense of all available data points.

1.2. Motivation and objectives

As pointed out in the previous section, there is a well-recognized need to access the huge amount of clinical information locked in free-text. As a matter of fact, many approaches have been proposed in the literature to perform clinical IE [6]. However, the majority of published works concern the English language, and advances in other languages are still limited mostly due to the lack of shared resources [7]. As clinical notes are always written in the institution local language, though, developing techniques that are able to process a specific language is crucial. Moreover, considering that clinical narratives frequently include jargon, abbreviations, and specialty-specific phrases, targeting NLP techniques to the analysis of this kind of text is needed.

Starting from this observation, this thesis is focused on the problem of extracting relevant information from textual medical reports written in the Italian language. The final aim is to build a system that is able to extract and visualize a clinical timeline of events starting from multiple patient documents.

The main research questions that motivated this thesis can be summarized as follows:

- What are the NLP challenges specific for the Italian medical language, and how can they be addressed?
- Is it possible to convert Italian medical reports into structured information that can be queried and examined in an automatic way?
- Is it possible to guide the clinical IE process to preserve some semantic relations inside the text?
- Is it possible to effectively summarize the information included in multiple medical reports, visualizing extracted data in a smart way?

To answer these questions, this dissertation presents methodologies that analyze texts written in Italian at different levels, extracting both clinical and temporal information. In particular, an IE approach relying on a domain ontology has been defined in order to extract clinical information in such a way that it verifies some predefined semantic relations.

From a general point of view, exploring the applicability of IE approaches to clinical text written in Italian represents a significant contribution to research on clinical NLP for non-English languages.

As a main clinical case to support the development of both IE methods and summarization approaches, the techniques explored in this thesis have been used to process documents in the molecular cardiology domain. As it will be explained throughout the dissertation, the extension to other clinical domains is investigated as well.

1.3. Relevance in the clinical setting

In the clinical practice, the implications of building a clinical IE system that is able to process text by extracting structured data are multi-fold. First, this kind of system would help clinicians to access unstructured information at the time of need, avoiding key-word search and manual inspection. As a main consequence, the time spent into such a demanding and error-prone activity would be reduced, providing a significant support to patient care. As an additional advantage, transforming clinical notes into structured data could facilitate improvements in data quality, allowing monitoring errors and missing data in a computer-aided way [8].

As regards clinical summarization, developing a system that reconstructs synthetic clinical histories could be important for two main reasons. On the one hand, displaying information belonging to the same patient on a single temporal line could facilitate the process of reviewing and making sense of multiple data points. On the other hand, comparing the clinical histories of different patients would allow searching for recurrent patterns, which could lead to interesting conclusions.

1.4. Dissertation outline

This dissertation is organized as follows:

Chapter 2 The second chapter provides the dissertation background. As this research activity focuses on extracting information from clinical narratives, this chapter formalizes the IE problem in this domain, presenting the related relevant literature. First, a few basic definitions are given. Then, the main IE methods found in the literature are presented. Besides describing the revised works from a methodological point of view, a few existing architectures and systems, which are useful to contextualize the conducted research activity, are illustrated.

Given that this dissertation investigates the applicability of IE approaches to Italian clinical text, relevant work in this language is presented. First, an overview of the main issues to be addressed is given. Then, the approaches that have been proposed to address these issues are described.

In the last section of the chapter, the problem of clinical summarization is considered. Specifically, a few relevant works on unstructured data summarization are described.

Chapter 3 The third chapter describes the methodological approaches proposed in this work for clinical and temporal IE in the Italian language. First of all, the main corpus used for the development of IE techniques is presented, i.e., a corpus of medical reports belonging to the molecular cardiology domain. To allow evaluating the proposed approaches, a subset

of this corpus was manually analyzed, identifying the information to be automatically extracted. In Chapter 3, this annotation process is described as well.

After providing an overview of the considered dataset, Chapter 3 presents the complete developed IE pipeline. To describe the methodologies proposed for each extraction step, a separate section is available (e.g., extraction of clinical concepts, extraction of temporal information). Finally, the approach used for summarizing the information extracted from multiple patient reports is described.

Chapter 4 The fourth chapter presents the main results of this research activity. Each section contains both the results obtained through the proposed evaluations, and a discussion including a comparison to the related literature. The chapter focuses on the work conducted on the main cardiology dataset: for each of the pipeline steps, a separate section is available. In the last section, the results of the summarization task are shown.

Chapter 5 The fifth chapter describes a multilingual extension of the developed IE pipeline, considering texts in the English language. Then, the chapter discusses the extension of the IE task to a different domain (i.e., oncology), highlighting the main differences to be addressed.

In the last part of Chapter 5, the exploitation of the proposed IE pipeline in two real clinical settings is discussed.

Chapter 6 The last chapter presents the main conclusions of this research activity, providing a summary of the conducted work, and highlighting its main novelties and contributions. In addition, the possible future directions are outlined.

Chapter 2

Background

This chapter provides the background material and the literature review for IE from clinical narratives. Section 2.1 provides a few basic definitions and presents approaches for clinical and temporal IE from free text, with an emphasis on the first task. Section 2.2 illustrates a few NLP architectures and IE systems that have been explored as part of this research activity. Section 2.3 describes relevant work in clinical and temporal IE for the Italian language. Finally, Section 2.4 outlines the problem of clinical summarization starting from multiple texts referring to the same patient.

2.1. Information extraction approaches

In this section, the problem of extracting information from a given *corpus* of documents is presented.

The section starts by introducing a few basic definitions that will be used throughout the dissertation. Then, it describes the most common preprocessing steps that convert the input corpus into a format suitable for subsequent analyses. Finally, the main methods found in literature to extract clinical and temporal information are presented.

2.1.1. Basic definitions

Unlike information retrieval, which deals with finding relevant documents in a certain collection, IE focuses on identifying predefined types of information in the text. Among IE subtasks, named entity recognition (NER) deals with the identification and the classification of relevant entities into predefined categories (e.g., persons, organizations, and locations). In clinical and temporal IE, which are the focus of this

dissertation, different extraction tasks are performed besides named entity recognition.

Clinical IE deals with identifying clinical information inside the text. In this case, the concepts to be identified are represented by diseases, medications, diagnostic procedures, and other relevant mentions. Besides extracting these concepts, clinical IE involves searching for additional related information of interest. First, for each identified concept, a few contextual properties are usually considered. For example, to determine the role of an extracted mention within the clinical narrative, it would be important to capture both its polarity (negative vs positive mention) and uncertainty level (e.g., concepts mentioned with some degree of uncertainty). Moreover, as clinical concepts are often mentioned together with a set of related attributes, extracting these attributes and their values would be important to fully identify all event-related information. For example, drugs prescriptions could be related to dosages and frequencies, while diagnostic procedures could be linked to their results.

Temporal IE aims to analyze the temporal structure of the text, and substantially requires three different steps: (i) the identification of relevant events, (ii) the identification of temporal expressions (e.g., "today", "4 pm"), which are typically referred to as *TIMEXes*, and (iii) the identification of temporal relations between entity pairs (Event-Event, Event-TIMEX, or TIMEX-TIMEX relations). A formal description of these steps can be found in the TimeML specification language, a set of rules that describe temporal IE in the general domain for the English language [9]. As defined by TimeML, entity properties can be extracted for both event mentions and temporal expressions. With respect to TIMEX entities, the *value* property is used to convert temporal expressions to standardized values, which allows ordering events on the same temporal line.

When applying temporal IE to the clinical domain, the events of interest can be defined as those that are relevant to the patient's clinical timeline [10,11]. As regards the temporal expressions, although most TIMEXes are common across different domains, further particular cases should be considered (e.g., the "post-operative" adjective conveys a temporal meaning).

It is important to point out that, to enable the development and the evaluation of an IE system, one first crucial step consists in manually *annotating* the information of interest inside the text [7]. This annotation process is required to define the entities, the properties, and possibly the relations to be extracted, thus creating a gold standard to be used as reference for the evaluation. Hence, to create a reliable and consistent annotated corpus, the development of clear annotation guidelines is essential.

In Figure 2.1, a high-level schema for extracting information from clinical narratives is depicted. On the left, the main steps that are performed on the input text are shown: preprocessing (i.e., an initial

elaboration of the input), clinical IE (e.g., extraction of events and their attributes), and temporal IE (e.g., extraction of temporal expressions and temporal links). On the right-hand side, the manual annotation of the input text is reported. As shown in the figure, the so produced gold standard can be exploited both for developing IE methods ("knowledge-based approaches" and "machine learning methods" arrows) and for the final evaluation of the system.

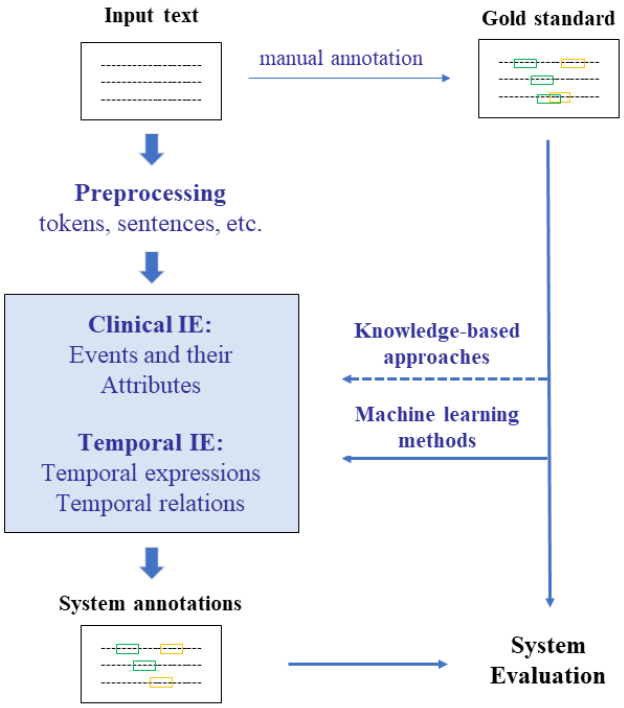

Figure 2.1: High-level schema for IE from clinical text.

Although the IE systems that have been proposed over the years rely on a variety of different approaches, most methodologies fall into one of two classes: knowledge-based techniques and machine learning methods. In the following, the use of these approaches, both separately and in combination, will be described in detail. For each approach, references to both clinical and temporal IE systems will be provided, with a focus on the former problem.

2.1.2. Preprocessing methods

In an IE pipeline, it is frequent to start analyzing the text by applying a set of preprocessing techniques, such as morphological and syntactic analyses. Figure 2.2 shows the main preprocessing steps that can be used to prepare the text for subsequent IE tasks.

Figure 2.2: Main preprocessing steps.

Text tokenization is one of the first steps to be performed on the input text. In this phase, the text is divided into *tokens*, i.e., linguistic units such as words, punctuations, and numbers. Although segmenting the text into tokens may seem straightforward, a few challenges must be considered. For example, deciding how to treat a hyphenated term (e.g., "patient-controlled") is not a trivial task, as either one or two tokens could be created, depending on the context. As a related issue, correctly identifying tokens that include punctuation marks, such as dates (e.g., "05/10/90") and times ("3:00"), could be challenging.

Another task closely linked to text tokenization is *sentence segmentation*, which deals with identifying sentence boundaries in the text. Also in this case, performing a correct segmentation requires addressing a few issues, above all the ambiguity of punctuation marks. For example, the presence of a period does not always denote the end of one sentence: such punctuation marks can be found in both abbreviations (e.g., "Dr.") and numbers ("2.5").

Once the text is segmented into tokens, a common processing step is represented by *morphological analysis*. This analysis is needed to perform lemmatization, a task that consists in assigning to each word its base form, called lemma (e.g., "patient" for "patients", or "complain" for "complained"). As lemmatization converts many possible variants to the same form, it allows making the content of the text uniform. For those languages, such as Italian, that have a high degree of inflection, working on lemmas can be useful to facilitate some subsequent NLP tasks, such as looking up terms in a dictionary. As another important processing step, *Part-of-speech (POS) tagging* assigns to each token the corresponding POS tag, i.e., a grammatical class such as noun, verb, and adjective. Among other tasks, POS tagging can be useful to disambiguate between two different word meanings. For example, the word "discharge" as a noun can refer to a bodily emission, while the same word as a verb can indicate a release from the hospital [7].

To obtain a shallow syntactic analysis of the text, tokens can be grouped into predefined constituents, such as noun phrases (e.g., "the old woman") and verb phrases (e.g., "was discharged"). This process, known as *text chunking*, can be useful to restrict subsequent analyses to predefined types of phrases. For example, to look for named entities inside the text, restricting the search to noun phrases could represent a good choice.

To perform each of the described steps, different methods can be used. In general, the approaches proposed in literature can rely either on general linguistic knowledge or on the specific corpus features.

To accomplish text tokenization and sentence segmentation, the most common approaches are based on regular expressions, i.e., standard notations used to specify text strings [12]. An example of regular expression is given by

```
\d\d\-\d\d\-\d\d\d\d
```

This expression identifies all the strings that represent a date in the format DD-MM-YYYY (2 digits for the day, 2 digits for the month, and 4 digits for the year). Besides regular expressions, resources like lexicons and predefined word lists (e.g., abbreviations, acronyms) can be used.

Another step often heavily relying on regular expressions is morphological analysis [13]. Also in this case, language-specific lexicons can be exploited. An example for the Italian language is given by the *Morph-It* lexicon, that includes word forms and lemmas obtained by processing a corpus with more than 3 million words [14].

As regards the POS tagging task, the most common approaches are based on either hand-crafted rules or statistical models [12]. While rule-based taggers use hand-written rules to disambiguate between two different POS tags, statistical taggers use a training corpus to compute the probability of a word having a certain tag in relation to the specific context (i.e., based on the adjacent words). It is clear that for this second type of algorithms, the choice of the training corpus is crucial to determine the tagging performance. For example, a POS tagger trained on a general domain corpus may not perform well on corpora belonging to other domains.

Finally, to address the text chunking problem, a variety of different approaches have been proposed. Among others, rule-based systems, memory-based systems, and statistical systems (e.g., hidden Markov models, maximum entropy models) have been explored [15].

2.1.3. Information extraction using knowledge-based approaches

As previously mentioned, both clinical and temporal IE require identifying domain-specific entities within the text. To this end, pattern matching techniques, such as dictionary lookup and rule-based approaches, provide a simple, yet effective solution. Dictionary lookup consists in searching for dictionary entries inside the text, thus basically performing a string matching. Rule-based techniques require defining a set of structural or grammatical rules that encompass more elaborate information.

Dictionary lookup approaches. As regards clinical IE, the availability of shared terminologies in the biomedical field greatly supports the creation of systems that exploit these resources, especially for the English language. For example, Pakhomov et al. developed a system that identifies relevant entities in clinical notes by using a set of shared medical dictionaries [16]. As another example, the MedEx system [17] extracts medication names and signature information from free text by exploiting RxNorm [18], a normalized naming system for generic and branded drugs.

Among available biomedical resources, the Unified Medical Language System (UMLS) is a particularly rich compendium, developed by the National Library of Medicine (NLM), that integrates and distributes terminology, classification and coding standards in the biomedical field [19]. The UMLS Metathesaurus is one of the three knowledge sources available within the UMLS, and gathers terms and codes from many vocabularies, including RxNorm, ICD-10-CM [20], and SNOMED CT [21]. In the UMLS Metathesaurus, terms are univocally identified by a Concept Unique Identifier (CUI) and organized into semantic categories. As this vocabulary database not only is very large, but also identifies useful relationships between the concepts, it has been widely used to develop automatic systems for extracting entities from clinical text. Among these systems, MetaMap was developed by the NLM itself to allow searching for UMLS Metathesaurus entries in text [22]. A detailed description of this system is provided in Section 2.2.3.

Besides exploiting public resources, it is possible to develop *ad hoc* vocabularies that include entities of interest for a specific application. As a well-known example, the MedLEE system extracts and encodes clinical information in textual patient reports, relying on hand-crafted lexicons [23]. Zhou et al. developed a tool that extracts information from clinical text by using both standard and institution-specific terminologies [24]. Carrell et al. created a custom dictionary to process electronic clinical notes for women with breast cancers [25].

Rule-based approaches. To identify complex information inside the text, such as drug regimens (e.g., dosage, frequency of assumption) or test results (e.g., heart rate, blood pressure), dictionary lookup alone is not a good choice, as all possible variants should be listed in the dictionary. To capture this complex information, it can be useful to rely on regular expressions and/or rules that combine different elements, including lexicon entries. For example, in the MedEx system, a dictionary lookup tagger and a regular expression tagger are used in combination to identify drug regimens [17]. While the first tagger relies on lexicon files to extract drug names and forms, the second tagger exploits regular expressions to identify information expressed through patterns, such as the prescription frequency (e.g., “q4h”, “q6h”). In addition, the system uses predefined context rules to determine the appropriate semantic categories for ambiguous terms. As another example, in a recent work by Patterson et al. a combination of dictionary lookup, rules, and patterns is used to extract echocardiogram

measurements [26]. In this case, the authors created a domain-specific lexicon to look for measurement names (e.g., "left ventricular ejection fraction"), and used regular expressions to identify measurement values. These extracted names and values were then linked through manually defined patterns (e.g., *term* + *separating string* + *value* + *unit*).

As stated earlier, in a plain dictionary-lookup approach, concepts included in specific dictionaries are searched for in the texts. As an additional way to guide the IE process, it is possible to rely on domain ontologies, too [27]. In the clinical domain, using an ontology that includes information on concepts and their semantic relations can be helpful to extract complex information [28]. For example, an ontology-driven approach could facilitate the task of extracting clinical concepts together with related attributes and their values. Spasić et al. proposed an ontology-driven system to extract information on findings and anatomical regions from magnetic resonance imaging (MRI) reports written in English [29]. The developed ontology was used to guide and constrain text analysis, while language processing was modeled through a set of sophisticated lexico-semantic rules. Mykowiecka et al. developed a rule-based system that extracts information from Polish clinical texts to fill in template forms [30]. To specify the information to be extracted, a domain ontology was designed, and manually translated into typed feature structures (TFSs) i.e., sets of attribute-value pairs. To extract information, these TFSs were combined by manually written grammar rules. In another work, Toepfer et al. created a system that extracts objects (mostly body parts), attributes, and values from German clinical texts [31]. To formalize these concepts, a domain ontology was developed and refined by domain experts, in a semi-automatic and iterative way. In this ontology, the most fundamental kinds of entries are *variants*, which specify lexical expressions referring to concepts, in form of either a string or a regular expression. At each iteration, the expert accepted or rejected the annotations that were automatically extracted by the system on the basis of the ontology, thus refining the ontology itself.

With respect to temporal IE, using lexicons and rules can be an effective way to both identify (i.e., finding boundaries) and normalize (i.e., converting to a standard format) temporal expressions. As a representative example, the HeidelTime system relies on regular expressions for the identification process, and uses knowledge resources and linguistic clues to normalize the extracted expressions to a standard format [32]. TimeNorm is another example of rule-based system performing temporal normalization [33]. In this case, the system exploits a synchronous context free grammar (SCFG [34]) that maps the language used in the text to formal operators for time manipulation.

2.1.4. Information extraction using machine learning methods

From a general point of view, *machine learning* techniques enable computers to automatically learn how to solve specific tasks, without being explicitly programmed. In supervised machine learning, the system is presented with example inputs and expected output labels, and aims to learn how to map each input to the corresponding label. Conversely, in unsupervised machine learning, no labels are given instead, letting the system finding regularities in the input data by itself. Therefore, in order to solve those NLP tasks that can be addressed as a classification problem (i.e., where each input must be assigned the corresponding label or class) supervised machine learning can be exploited. An example of classification problem in NLP is represented by entity recognition. As previously mentioned, the goal of this task is to identify predefined entities in the text. To address this problem, the beginning-inside-outside (BIO) classification schema is commonly used. In this schema, the input data are represented by sequences of tokens, while the corresponding outputs are given by B, I, O labels denoting the inclusion (B, I) or the exclusion (O) of each token in an entity. For example, in a NER task focused on the identification of persons, the sentence "*Barack Obama was president*" would be translated to the following sequence: "*B I O O*". In this case, the B and the I labels represent the beginning and the inside of the *Barack Obama* entity, respectively.

Annotated corpora and competitions. As a matter of fact, developing supervised classifiers requires the availability of large datasets to "learn from". This is true also for NLP classification problems: to enable the development of effective classifiers, the most important first step is the creation of large, reliably-annotated corpora [35].

For the English language, a few corpora have been collected and annotated to support advances in supervised clinical IE (e.g., MiPACQ [36], ShARe [37]). Most of these corpora have been then used to organize IE competitions, thus enabling the development and the evaluation of various supervised approaches (e.g., 2010 i2b2 challenge [38], 2013 ShARe/CLEF eHealth task [39], SemEval-2015 task [40]). As an example of clinical IE competition, the SemEval-2015 task "Analysis of clinical text" focused on entity recognition and template slot filling for clinical texts [40]. In this case, the ShARe corpus, consisting of 531 de-identified clinical notes annotated with disorder mentions and a set of relevant properties (e.g., negation, uncertainty), was used as the main dataset.

Regarding temporal IE from clinical narratives, annotated resources have been created, too. The Informatics for Integrating Biology and the Bedside (i2b2) project created a temporally annotated corpus consisting of 310 de-identified summaries, all annotated with clinical and temporal information [10]. The developed corpus was used for organizing the 2012 i2b2 Challenge, which involved the identification of events, TIMEXes, and a subset of the TimeML temporal links inside the texts [41]. Styler IV et al. developed a formal specification for annotating temporal information in

clinical text, extending the TimeML guidelines to the clinical domain [11]. This formalization effort resulted in the creation of the THYME corpus, which consists of 1254 de-identified notes annotated with clinical and temporal information. This corpus has been exploited in different NLP challenges (Clinical TempEval competitions [42–44]). Among the most recent ones, the 2016 Clinical TempEval task required participants to extract events, TIMEXes, and their properties from the texts [44]. As regards temporal links, two kinds of relations were considered for the competition: (i) relations between each event and the document creation time, and (ii) relations between an event or a TIMEX and a *narrative container*, i.e., a time span that is central to the discourse.

Supervised extraction methods. In the described clinical and temporal IE competitions, most state of the art solutions involved supervised methods. As previously mentioned, the SemEval-2015 task "Analysis of clinical text" involved two different assignments [40]. For the first task, i.e., disorder span recognition and UMLS/SNOMED-CT normalization, most teams used supervised approaches based on conditional random fields (CRFs [45]). For the second task, which consisted in identifying nine properties for the extracted disorders, different classifiers were built for each property. Referring to a clinical/temporal IE challenge, the 2016 Clinical TempEval task consisted in extracting events, TIMEXes, and temporal links from clinical narratives [44]. Also in this case, all the state of the art solutions included supervised approaches. The best performing systems relied on structured learning models, namely support vector machines (SVMs [46]) and CRFs. It is important to point out that the temporal relation tasks were regarded as the most difficult ones.

The supervised methods used to identify entities and properties use a variety of different features. Among the most common ones, lexical and morphological features (e.g., bag of words, word orthographic forms), as well as syntactic aspects (e.g., POS tag, phrase chunks) are usually exploited. The quality of these features is highly dependent on the performance of the preceding processing steps. As another useful set of features, the inclusion of concepts from external resources (e.g., medical dictionaries) can be exploited, too. Finally, word representation features have been recently used for their ability to automatically capture useful morpho-syntactic aspects [47,48]. Going a little bit into the details, a word representation is a mathematical object associated with each word, often a vector [49]. These word vectors can be learnt on large unlabeled corpora through different techniques (e.g., word embeddings [50]), and represent the actual features that can be used in a supervised learning context.

Supervised approaches like CRFs and SVMs usually rely on complex features to be adequately trained. As an alternative approach that allows reducing the annotation effort, neural network architectures have raised increasing interest in the NLP community [51]. Neural networks are powerful machine learning models based on connected computation units, called *nodes*, which receive scalar inputs (each associated to a weight) and

produce scalar outputs. To solve a generic IE task, a neural network model receives a list of tokens as inputs (more precisely, the corresponding word representations would be used), and returns specific labels as outputs. In the network training phase, the set of weights which better describe the relation between the inputs and the outputs are automatically learnt. As a main advantage, neural network models do not need language-specific preprocessing or manually engineered features. Thanks to these characteristics, neural network models have been successfully applied to many entity recognition tasks [52–55] on different domains and languages [56–58]. In the clinical domain, Li and Huang investigated neural networks to identify event spans and their properties from clinical notes and pathology reports written in English [56]. As regards the Italian language, Bonadiman et al. proposed a neural network to predict tags for entity recognition in the general domain [57].

2.1.4.1. Recurrent neural networks for entity recognition

Among the supervised approaches available for clinical IE, this research activity investigated the application of neural network models to the task of event extraction. This section provides an introduction to the considered models, with a particular focus on recurrent architectures.

As mentioned, any entity recognition task, including the identification of clinical concepts, can be treated as a classification problem. More specifically, recognizing entities formed by multiple tokens represents a *sequence labelling problem*: a sequence of tokens must be transcribed to a sequence of labels indicating the inclusion of the tokens in the predefined entities. As it will be explained in the following, there exists a class of neural networks, called *recurrent neural networks* (RNNs), which are specialized for processing and classifying input data in the form of sequences. Thanks to this characteristic, RNNs are particularly suitable to solve entity recognition tasks [53–55,57]. In the following paragraphs, an introduction to neural networks models and RNNs is provided.

Neural networks. Inspired by the brain's functioning, neural networks are machine learning models consisting of computation units resembling the brain's neurons. These units, or nodes, receive scalar inputs and produce scalar outputs, and they are connected to each other by means of weighted links. The network is activated by providing an input to some of the nodes, and this activation then spreads from one node to the other along the weighted connections [59].

Although many types of neural network have been proposed over the years, a substantial distinction exists between feed-forward neural networks, whose connections do not form cycles, and RNNs, which are characterized by cyclic connections.

The most popular example of feed-forward neural network is the *multilayer perceptron*, shown in Figure 2.3. In this network, nodes are

organized in layers, with connections going from one layer to the next. An input layer receives the input data, and an output layer produces the network output. All layers in between are referred to as *hidden layers*. Neural network models including a high number of hidden layers are typically referred to as *deep* neural networks.

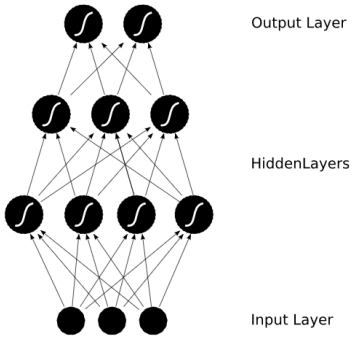

Figure 2.3: Multilayer perceptron [59].

During the network activation, each node in the network calculates a weighted sum of its input units, and applies a non-linearity (called *activation function*) to this sum. Considering a node h with I inputs x_i, each associated to a weight w_{ih}, the weighted sum of inputs is given by a_h. The final output b_h is obtained by applying the activation function (f_h) to a_h [59]:

$$a_h = \sum_{i=1}^{I} w_{ih} x_i$$

$$b_h = f_h(a_h)$$

The obtained output of the unit (b_h) will serve as input to the subsequent layer's nodes in the network.

The parameters of the model (i.e., the set of weights w_{ih}) are estimated during the network training phase: an optimization algorithm computes the best set of values to model the input-output relations over the training set. For neural network models, the stochastic gradient descent algorithm (or one of its variants) is usually exploited [60].

For some classes of applications, the described structure does not seem to be adequate. For example, in case of a sequential input (such as the words of a sentence), the feed-forward neural networks would treat each item as an independent input, without memory of the previous one. For this kind of problems, RNN have been proposed. As a main difference with respect to feed-forward neural networks, RNNs allow cyclical connections, too. So, while feed-forward architectures are intended to map from one

single input vector to one output vector, an RNN architecture takes as input an ordered *list of vectors* (e.g., they can be interpreted as inputs at different time steps) and returns an ordered list of vectors as the output. In Figure 2.4, a simple RNN with only one hidden layer is depicted: at each time step, the hidden layer receives activations from both the current external input and the hidden layer activations from the previous time step [59].

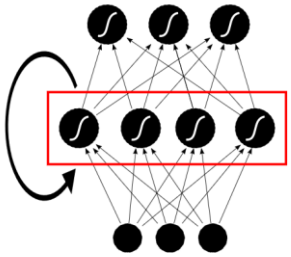

Figure 2.4: Recurrent neural network [59].

Going into the mathematical details, an RNN takes as input a sequence of vectors $\mathbf{x_1}$, ..., $\mathbf{x_N}$ as well as an initial state $\mathbf{s_0}$, and returns a list of state vectors $\mathbf{s_1}$, ..., $\mathbf{s_N}$ together with a sequence of output vectors $\mathbf{y_1}$, ..., $\mathbf{y_N}$ [51]. The $\mathbf{x_n}$ vectors are given as inputs to the network in a sequential way: at the n-th time step, $\mathbf{s_n}$ and $\mathbf{y_n}$ represent the state and the output of the network after processing the inputs $\mathbf{x_1}$, ..., $\mathbf{x_n}$. The state vector $\mathbf{s_n}$ is a function R of the current input ($\mathbf{x_n}$) and the previous state ($\mathbf{s_{n-1}}$), while the output vector $\mathbf{y_n}$ is a (typically non-linear) function O of the corresponding state vector ($\mathbf{s_n}$):

$$s_n = R(s_{n-1}, x_n)$$

$$y_n = O(s_n)$$

As for feed-forward neural networks, both R and O depend on a set of parameters θ, which represent the network's weights.

Figure 2.5 provides a graphical representation of an RNN, "unrolled" over time. In this case, a sequence of five input vectors is considered, resulting in a neural network in which the same parameters (θ) are shared across all time steps, since on each input the same classification task is performed.

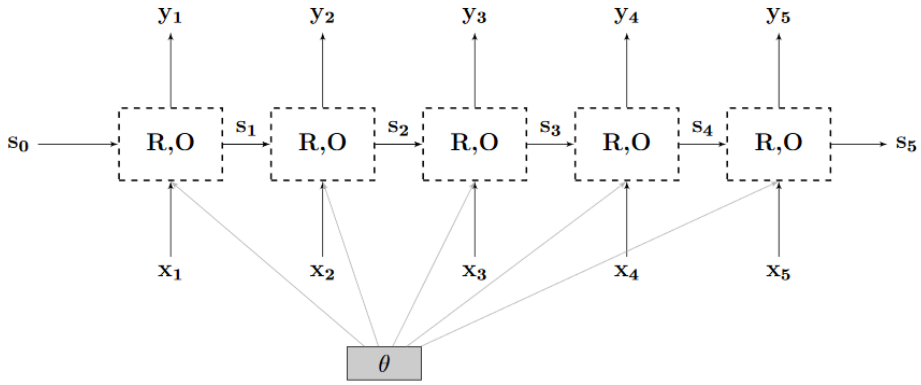

Figure 2.5: "Unrolled" recurrent neural network [51].

As it can be noticed from Figure 2.5, an unrolled RNN is a deep neural network. As a main consequence, the parameters of the model can be learned in the same way as for feed-forward networks. In particular, for a given input sequence, it is sufficient to train the unrolled network by applying an optimization algorithm.

Word embeddings. As previously explained, neural network models are used to process numeric inputs, and require the computation of weighted sums. For this reason, strings are not a good choice to represent the input of a neural network. Therefore, to apply this class of methods to the analysis of natural language, words are usually converted into real-valued vectors called *embeddings* [50].

Embeddings can be defined as *vector representations* of words. In an unsupervised framework, these vectors are learnt from a large corpus of documents through different techniques. Most of these methods rely on the *distributional hypothesis*, which states that words occurring in the same contexts tend to have similar meanings [61]. In this framework, given a target word, its context is defined as the set of surrounding words (e.g., the 2 preceding and the 2 following words). Among the models used for learning embeddings, the *continuous bag-of-words* (CBOW) and the *skip-gram* models are the most widespread. Both of them involve the definition of a probabilistic classification problem: given a large corpus, the first model predicts target words starting from their context, while the second one operates the other way around [62]. In a CBOW model, for example, each word in the input context is a one-hot encoded vector indicating its position inside the vocabulary (only one element is 1, and all the others are 0), while the output is a vector where each element represents the posterior probability of that word being the target word. To compute the probability of each target word given the context, words are converted via lookup tables to vectors with lower dimensionality with respect to the vocabulary size: these vectors are the searched numerical representations of the words (i.e., the embeddings). The embeddings learnt in this way were shown to

effectively capture the syntactic and semantic features of words, and can be used to support many NLP tasks [63].

Word embeddings can be used also in a supervised NLP classification task based on neural networks. In this case, *embedding layers* can be used to convert each word to a fixed-size embedding vector, that will serve as input to the main neural network structure. In this framework, the word embeddings are learned during the training of the overall network, designed to solve the specific supervised task (e.g., the classification of input tokens with B, I, or O labels). In this sense, these embeddings are different from the ones that are learnt in an unsupervised way on a large corpus, for example through the CBOW or the skip-gram models. However, it has been shown that initializing embeddings with vectors pre-trained in an unsupervised way can be helpful to build a neural network for a supervised task, as it provides a "good" starting point that guides the network training [64].

2.1.5. Combined approaches

For extracting domain-specific entities and their properties, both pattern-matching approaches and supervised machine learning methods present advantages and drawbacks.

Approaches that exploit external lexicons and rules are a good choice to identify concepts that are included in standardized resources (e.g., drug names), or extract information that follows well defined patterns (e.g., drug dosages). However, these approaches present two main disadvantages. On the one hand, textual reports could contain relevant mentions that are not found in external resources. On the other hand, creating well performing rules requires a considerable manual effort.

As regards supervised machine learning, two main advantages can be pointed out. First, supervised classifiers can learn to effectively identify non-standard entities in the text, without requiring much human intuition. Second, in case of generalizable enough classifiers, the models developed for one application can be easily retrained and reused on a new different domain. Also in this case, though, a crucial drawback must be considered. To obtain well performing and generalizable classifiers, large annotated training corpora are needed, and building such corpora is expensive in terms of both time and resources.

To combine the advantages of these two classes of approaches, it is possible to develop systems that integrate multiple modules, each relying on a different methodology. One of the most popular systems of this kind is cTAKES, an NLP pipeline for the extraction of information from clinical free-text [65]. As it will be detailed in Section 2.2.3, cTAKES allows performing several tasks through different NLP modules, which can be customized using both dictionaries and machine learning.

Another example of hybrid system for clinical IE was proposed by Wang et al., who focused on the problem of extracting disorder concepts

from clinical narratives [66]. In this case, the developed system includes three extraction components, one based on supervised machine learning, and the others based on pattern matching. The first component is an SVM classifier that uses different kinds of features, such as bag of words and orthographic features, to identify disorder mentions in the text. The second component is a rule-based annotator that corrects the errors performed by the SVM classifier. Finally, the third extraction component relies on the MetaMap system to identify concepts not included in the training data.

With respect to temporal IE applied to the clinical domain, an example of hybrid system was proposed by Kovačević et al. as a contribution to the 2012 i2b2 Challenge [67]. To extract both event mentions and temporal expressions, the authors combined rules and machine learning techniques. For extracting all types of events, suitable CRF classifiers were developed. For one specific event type (*Clinical Department*), the CRF module was also combined with a manually-curated dictionary. To extract temporal expressions, the authors combined 65 manually engineered rules and a CRF module based on token-level features (e.g., lexical and domain features). Finally, for the normalization task, a rule-based approach was exploited.

2.2. Information extraction systems

In this section, a few existing architectures and systems, which are useful to contextualize the conducted research activity, are illustrated:

- General architectures: GATE, UIMA
- Preprocessing tools: Stanford CoreNLP, Apache OpenNLP, TextPro
- Clinical IE systems: MetaMap, cTAKES

2.2.1. General architectures

GATE. GATE is an open-source infrastructure for developing software components that process human language [68]. As an architecture, it allows defining the structure of an NLP system that combines many of these components. To facilitate software development, GATE also provides a set of building blocks that can be reused, extended, and customized.

Thanks to its flexibility, the GATE architecture has been used to develop several NLP applications. For example, it has been recently exploited to perform a real-time semantic analysis of social media content [69]. Regarding the biomedical domain, Cunningham et al. have described three well-performing systems that leverage on the GATE architecture [70]. The first one contributes to gene-disease association studies by using a method called Adjusting Association Priors with Text (AdAPT). This method searches research paper abstracts for prior knowledge on single nucleotide polymorphisms, thus facilitating the discovery of gene-disease associations.

The second system was developed to extract information from a large mental health case register in the UK. The system extracts the results of a cognitive ability test (the Mini Mental State Examination) from textual medical records, identifying assessments, scores and dates. Finally, the third system uses GATE's ontology tools and a data repository to search for drug-related information over patent data. It performs a semantic annotation of patents, identifying references to drugs and additional related information (e.g., ingredients, organizations, dosages and routes of administration).

UIMA. UIMA is a software architecture for developing unstructured information management applications [71]. These applications analyze large volumes of unstructured data, such as textual documents or images, to discover relevant knowledge. As a main feature, UIMA allows decomposing applications into modules, each with a different role. With respect to language processing, it is possible to build a pipeline that takes in input a collection of textual documents, and outputs the same documents annotated with the relevant information found by each component (*Annotator*). Given that UIMA enables to easily develop, customize, and aggregate different Annotators, which may be independent or rely on previous annotations, many NLP tools have been implemented by exploiting this architecture. For example, the already mentioned cTAKES system is built upon UIMA, and has a modular structure that allows to reuse any UIMA compatible component [65]. As regards other well-known systems for IE, both MetaMap [22] and HeidelTime [32] are available as UIMA components, allowing them to be easily integrated in existing applications. As a final example in biomedical NLP, the *BioNLP UIMA Component Repository* maintains several annotators used in biomedical text processing as UIMA components [72]. Specifically, this repository includes components for gene identification, biomedical term recognition and mutation identification, as well as general preprocessing components (tokenization, sentence detection, and semantic parsing).

2.2.2. Preprocessing tools

Stanford CoreNLP. Stanford CoreNLP provides tools to perform many NLP tasks [73]. At the token level, it allows obtaining the base forms of words, their POS tags, and whether they are named entities (e.g., companies, people). At the sentence level, it identifies phrases and syntactic dependencies, indicating which noun phrases refer to the same entities. The basic Stanford CoreNLP distribution provides model files to analyze well-written English texts. However, packaged models are available also for Arabic, Chinese, French, German, and Spanish.

Apache OpenNLP. Apache OpenNLP is another tool that supports the most common NLP tasks [74]. Among others, it performs tokenization,

sentence segmentation, POS tagging, named entity extraction, chunking, and parsing. Also in this case, most OpenNLP models were developed for the English language. For the most basic tasks, however, pre-trained models are available also for Danish, German, Swedish, Dutch, Spanish, and Portuguese.

TextPro. As regards the Italian language, TextPro is a well-known suite of modular NLP tools that perform different tasks [75]. The TextPro pipeline includes modules for tokenization, sentence segmentation, morphological analysis, POS tagging, lemmatization, text chunking, NER, and syntactic analysis. TextPro is available for both the Italian and the English languages, and the different modules have been evaluated in several shared tasks (e.g., EVALITA [76]).

2.2.3. Systems for clinical information extraction

MetaMap. The MetaMap tool identifies UMLS concepts in the text by performing dictionary lookup and other elaborate processing [22]. The structure of MetaMap is depicted in Figure 2.6.

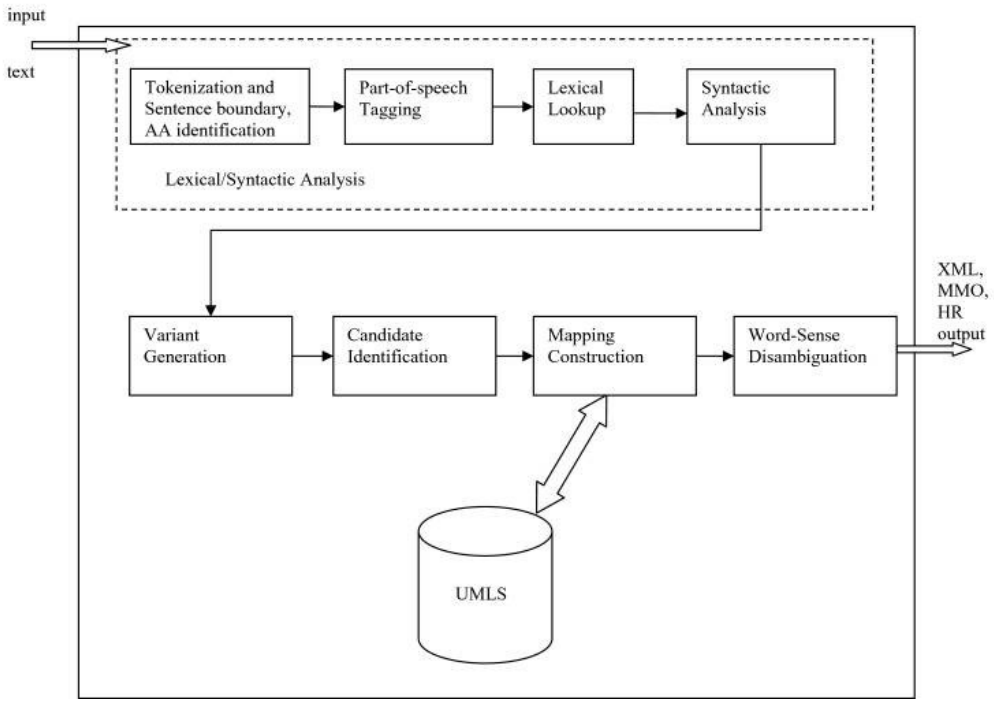

Figure 2.6: MetaMap system diagram [22]. HR, human readable; MMO, MetaMap machine output; UMLS, unified medical language system.

First, the input text undergoes lexical and syntactic analysis through four basic steps: text segmentation (i.e., tokenization, sentence splitting, and identification of acronyms/abbreviations), POS tagging, lexical lookup on

the SPECIALIST lexicon (i.e., an UMLS lexicon that includes syntactic, morphological, and orthographic information about words), and a syntactic analysis that identifies phrases and their lexical heads, i.e., the word that determines the syntactic category of that phrase. The output of this processing is a list of phrases, that are subsequently analyzed by four components:

1. Variant generation: the variants of all phrase words are computed (e.g., spelling variants, acronyms);
2. Candidate identification: based on the generated variants, the possible Metathesaurus strings (*candidates*) are identified. Each candidate is then evaluated according to how well it matches the input text;
3. Mapping construction: the extracted candidates are combined, and the results are evaluated to produce the best match for the input text;
4. Word sense disambiguation (optional): when multiple UMLS mappings are regarded as possible, surrounding tokens (i.e., the context) are used to determine the preferred one.

The MetaMap system has been used for extracting different kinds of information from clinical text, such as for recognizing specimens and their findings [77], or for problem extraction [78]. As reported by Aronson et al., this system has evolved significantly over the years, and has been used by many groups in the biomedical informatics community.

Apache cTAKES. The Apache cTAKES system extracts relevant information from clinical narratives by exploiting a variety of different approaches, including both pattern matching and machine learning techniques [65]. As already mentioned, this system relies on the UIMA architecture, and consists of several modules, each with a specific role. In Figure 2.7, the components for the most recent version (currently, Apache cTAKES 4.0) are shown.

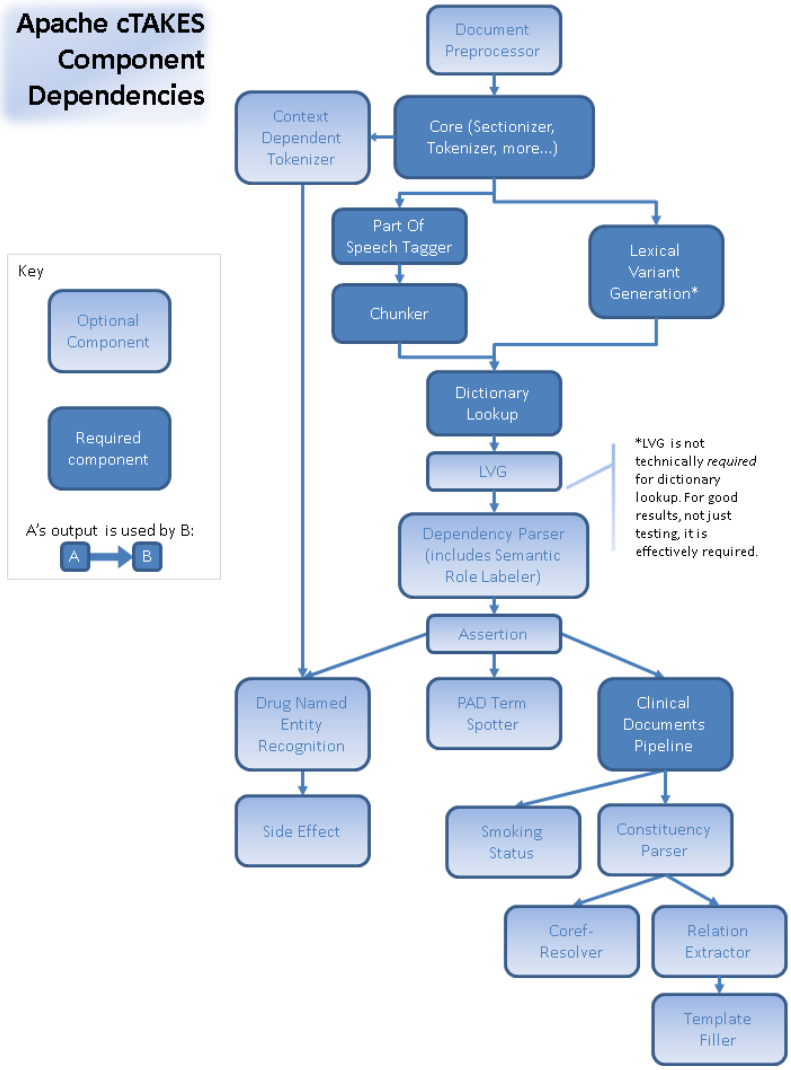

Figure 2.7: Apache cTAKES components [79].

First, the textual input undergoes the usual preprocessing steps (e.g., tokenization, sentence segmentation, POS tagging, chunking). To obtain token annotations that depend on surrounding tokens, it is possible to use a context dependent tokenizer. To generate the lexical variants of words, the SPECIALIST tool is exploited.

Once the text is preprocessed, different components for clinical IE can be used:

1. The Dictionary Lookup component identifies terms in text and normalizes them according to a specific terminology (e.g., UMLS, Snomed-CT, RxNorm).
2. The Dependency Parser component provides syntactic information about sentences. Instead of finding standard phrases (e.g., noun phrases, verbal phrases), it looks for other types of dependencies between words.

3. The Assertion component identifies the contextual properties of an extracted concept (e.g., polarity, uncertainty).

Besides performing clinical IE at the general level, cTAKES allows addressing more specific tasks:

4. The Drug NER component identifies drug mentions and related attributes, such as dosage and format. The Side Effect component subsequently extracts asserted side effects, or sentences that possibly contain causative drugs and their side effects.
5. The PAD Term Spotter processes radiology notes to extract information specifically pertaining to lower limb Peripheral Artery Disease (PAD).
6. The Clinical Documents Pipeline processes clinical documents by combining all the components that are available in cTAKES. Once the execution of this pipeline is terminated, a few additional processing steps can be performed. In particular, the Smoking Status component can be used to determine patients smoking habits according to five possible "smoking" categories (past smoker, current smoker, smoker, non-smoker, and unknown). The Constituency Parser component instead identifies subphrases in the text, such as noun phrases and verb phrases. Once the constituency parse tree is obtained, two final steps can be performed: Coreference Resolution, which finds all the expressions that refer to the same entity, and Relation Extraction, which annotates predefined relations between different annotations.

As an interesting feature, cTAKES includes a temporal module that extracts events, temporal expressions, and their relations. The Event annotator uses a Begin-Inside-Outside (BIO) format to extract clinically relevant events. The temporal expressions annotator exploits various machine learning methods (e.g., CRFs, SVMs) to identify expressions that denote dates, times, frequencies and durations. For every event, an SVM-based annotator computes the temporal relation between the event itself and the document creation time. Finally, different SVM approaches are used for detecting relations between pair of entities (Event-TIMEX or Event-Event) occurring in the same sentence.

Thanks to the availability of several modules and the possibility to easily add extensions, cTAKES has been widely adopted by the clinical NLP community. In some cases, it has been used to extract the features needed for subsequent machine learning analyses [80,81]. Currently, the system is used by different hospitals (e.g., Boston Children's Hospital, Mayo Clinic) and academic institutions (e.g., Massachusetts Institute of Technology, University of Pittsburgh).

2.3. Related work on the Italian language

Despite the increasing research activity in clinical and temporal IE, advances in languages other than English are still limited, mainly due to the lack of shared resources and tools. This is true also for the Italian language, which is the focus of this thesis. In this section, the main challenges and the relevant literature related to IE from clinical text written in Italian are presented.

2.3.1. Main challenges

To process clinical narratives in the Italian language, one main issue is represented by the substantial lack of shared annotated corpora. The availability of a corpus annotated with the information of interest is essential for two reasons: not only it is needed for evaluation purposes, but it also enables the development of supervised machine learning approaches. As far as it is known, only two annotated corpora were developed for the extraction of clinical concepts in the Italian language, as it will be described in Section 2.3.2. However, these corpora are not publicly available and cannot be reused for further exploration of supervised techniques.

A second issue specific to the Italian language concerns the availability of shared medical dictionaries and terminologies. Although there are a few medical resources that can be exploited for clinical IE, their coverage is much smaller in comparison to their English counterpart. As an emblematic example, while the English version of the UMLS Metathesaurus currently gathers 131 sources, the Italian version of the same vocabulary only includes 6 different dictionaries.

As a final issue, despite the availability of preprocessing tools for the Italian language, the underlying models were mostly developed on general domain corpora. However, with respect to the general domain text, clinical narratives present additional challenges, such as the abundancy of abbreviations and acronyms, the presence of ungrammatical phrases, and the use of institution-specific jargon. Therefore, using preprocessing tools developed on a general domain corpus may not perform equally well when applied to clinical narratives.

2.3.2. Relevant literature

Unsupervised approaches for clinical IE. To assess the usability of IE tools that do not require annotated data, Chiaramello et al. applied the MetaMap system to clinical text written in Italian [82]. The main goal of the study was to determine whether a linguistic tool developed to process English text could be suitable to extract medical concepts in the Italian language. To adapt MetaMap for this purpose, the Italian version of the

UMLS Metathesaurus was used as source for candidate retrieval. For all other steps, the original English modules were exploited. To allow a systematic evaluation of the adapted system, the authors annotated 3462 unstructured sentences, taken from 100 clinical notes written in Italian, with mentions of medical concepts (ITA-TXT corpus). In the manual annotation process, each identified concept was assigned the corresponding CUI in the UMLS Metathesaurus. For comparison purposes, the authors also created an English corpus by means of automatic translation, and annotated this translated corpus as well (EN-TXT corpus). To compare the performance of the system on the two languages, the ITA-TXT corpus was processed with the Italian MetaMap system, while the EN-TXT corpus was processed with the original English system. The experiments conducted in this work led to two main results. First, by manually identifying terms and CUIs in the two corpora, the authors found that the Italian UMLS Metathesaurus has a smaller coverage in comparison to the English one: while 99% of concepts were found in the English UMLS Metathesaurus, only 91% of concepts were included in its Italian version. As a second interesting result, the authors obtained a better performance by running the original English MetaMap on the automatically translated corpus (EN-TXT) than by using the adapted system on the ITA-TXT corpus. In particular, the lack of the "variant generation step" for Italian was identified as the main source of annotation failures on the ITA-TXT dataset.

Another work not relying on annotated data was proposed by Alicante et al. [83]. The developed system extracts information from clinical records written in Italian by using only unsupervised methods. It includes two main components: first, relevant entities are extracted by using standard preprocessing tools and external dictionaries; then, unsupervised clustering methods are exploited to discover relations among the entities extracted from the whole dataset. As a first step, the input texts are preprocessed by the TextPro tool, which identifies sentences and tokens, including their POS tags and lemmatized forms. Then, entity extraction is performed through a pattern matching approach: predefined POS sequences are considered as candidates for dictionary lookup, which is then performed on the UMLS Metathesaurus and on an Italian list of pharmaceutical terms. Regarding the relation extraction step, the underlying hypothesis is that a potential relation could exists between all entity pairs occurring in the same sentence. These pairs are thus represented by ad hoc feature vectors, and clustering techniques are used to group similar pairs. To compute the feature vector for a certain entity pair, the features associated to the two involved entities are concatenated together: the entity type, a predefined list of n-grams (sequences of 1, 2 or 3 words) where at least one word belongs to the entity, and *barrier* features, which use POS tags to capture information about syntactic patterns inside the sentence. In the clustering phase, entity pairs belonging to the same clusters are assumed to have the same type of relation, while cluster having a small size are considered as representative for "non-relations". The proposed methods were used to

process 989 medical records written in Italian. As a qualitative assessment of the obtained results, it was noticed that the system identified clusters corresponding to possible relations, which were automatically labelled by using the most significant features.

Supervised approaches for clinical IE. In the Italian clinical NLP community, few works have dealt with corpora annotation for developing supervised IE techniques. In the work by Esuli et al., 500 mammography reports were annotated by two different annotators with segments belonging to one of 9 classes (e.g., mammography standardized code, technical info) [84]. Annotated segments not necessarily coincided with entire sentences (they could also cross sentence boundaries) and had an average length of 17.33 words. As a result of the annotation process, three corpora were created: two datasets were annotated by a single annotator, and the third was annotated by both. These corpora were then used to explore two IE approaches based on CRFs. The first approach consisted in a two-stage method using two taggers generated via a linear-chain CRFs learner [45]. The second approach was an ensemble method that combined standard LC-CRFs and the proposed two-stage method. As an interesting point, positional features were used to account for the position of a token inside the text. As a main result, the authors found that combining the two-stage method with standard linear-chain CRFs outperformed the traditional single-stage CRFs system.

In another work related to corpora annotation and supervised machine learning, Attardi et al. addressed the problem of extracting information from large scale records written in Italian [85,86]. To enable the development of supervised approaches, the authors created a corpus of 10000 sentences taken from a collection of 23695 medical reports written in Italian. Sentences were annotated with six different mentions: active ingredients, body parts, signs or symptoms, diseases or syndromes, drugs, and treatments [85]. In this case, the corpus was created by using automatic tools and manually correcting the obtained annotations. To identify measurements as well, a rule-based approach was used to create corresponding annotations in a semi-automatic way [86]. Besides recognizing relevant entities, the authors extracted two other kinds of information: relations between measurements and entities, and the presence of negations. To carry out the NER tasks, a customizable statistical sequence labeler was exploited. In particular, three different classifiers were built: one for body parts and treatments, one for other medical entities, and one for measurements. In the performed experiments, the best results were obtained by using a support vector classifier (with L2 regularization term). To perform relation extraction and identify negations, a subset of the corpus (10%) was annotated with this additional information. For both these tasks, features involving the parse trees of sentences were used, and suitable SVM-based classifiers were developed. As stated by the authors, although the obtained results were promising, a corpus extension would be needed to further assess the performance.

As a topic not directly related to the extraction of medical concepts, Gerevini et al. recently worked on the automatic classification of Italian radiological reports following a multilevel schema [87]. To develop supervised machine learning techniques, they manually annotated a corpus of 346 reports using five levels of classification (exam type, test result, lesion neoplastic nature, lesion site, and lesion type). The annotated corpus was used to run experiments with different learning algorithms (Naïve Bayes, SVMs, decision trees, random forests, and neural networks), leading to encouraging results.

In Table 2.1, the methodologies that have been proposed for processing clinical text in the Italian language are summarized. It is important to point out that, among the revised papers, only one work has dealt with the extraction of relations between concepts and related attributes (e.g., measurements) [86].

Table 2.1: Relevant literature on clinical text processing for the Italian language. NB: naïve bayes; CRF: conditional random field; SVM: support vector machine; DT: decision tree; RF: random forest; NN: neural network;

Paper	Task	Method	Dataset
Chiaramello et al. [82]	Clinical IE	Rule-based (MetaMap system)	3462 sentences
Alicante et al. [83]	Clinical IE and entity clustering	Rule-based and unsupervised ML (clustering)	989 medical records
Esuli et al. [84]	Clinical IE	Supervised ML (CRFs)	500 medical reports
Attardi et al. [85,86]	Clinical IE and relation extraction	Supervised ML (SVMs)	10000 sentences
Gerevini et al.[87]	Text classification	Supervised ML (NB, SVMs, DT, RF, NN)	346 reports

Temporal information extraction. As regards temporal IE in the Italian language, few challenges have been organized, all involving texts belonging to the general domain. The 2007 EVALITA task on temporal IE ("Temporal Expression Recognition and Normalization") required participants to recognize and normalize the temporal expressions included in 525 newspaper articles written in Italian [88]. A multilingual task within the SemEval 2010 conference (TempEval-2 [89]) involved temporal IE from newspaper articles written in Italian. The 2014 EVALITA task named "EValuation of Events aNd Temporal Information" extended the 2007 temporal IE task to the extraction of events and temporal relations [90]. This competition also provided the chance to test the Ita-TimeBank resource, a corpus of Italian texts annotated according to the It-TimeML guidelines (an adaptation of TimeML to the Italian language) [91].

The organization of competitions for the Italian language has greatly supported the development of systems for temporal IE in the general

domain. For example, Manfredi et al. adapted the HeidelTime system to the analysis of the Italian language by tackling the recognition of empty tags (i.e., temporal expressions without an explicit correspondence in the text, but that can be inferred from other expressions) and by tuning HeidelTime's Italian resources [92]. As another example, Mirza and Minard created an end-to-end system for temporal IE, mostly based on supervised machine learning approaches [93]. In the developed system, SVM classifiers were used for TIMEX recognition, event detection and classification, and temporal relation identification and classification. For normalizing each extracted TIMEX to a standard value, the TimeNorm tool was adapted to the Italian language.

Despite the increasing interest in applying temporal NLP techniques to Italian texts, most efforts have targeted the general domain. As far as it is known, only one work has focused on the development of temporally annotated resources for the biomedical domain [94]. However, the developed corpus is not currently available.

2.4. Clinical summarization approaches

Besides performing clinical and temporal IE on single patient documents, summarizing the individual information extracted from multiple sources can effectively enhance the process of reviewing longitudinal data [5].

The need for patient record summarization has been known for a long time [95]. Several tools for the automatic summarization of patient records have been proposed over the years [96]. In many cases, structured electronic health records are considered [97,98]. Nevertheless, a few tools deal with unstructured data as well [99–101]. In this section, a brief description of such tools is reported.

CliniViewer is one of the first examples of summaries created using NLP [99]. In this system, multiple patient reports are first processed by using MedLEE, that creates structured XML outputs containing the extracted information. These outputs are then modified and merged by a Tree Generator, resulting into two different XML trees: a *conceptual* tree, that provides a summarized view of all the extracted concepts, and a *report* tree, that includes the original reports. These two trees can be visualized through a Tree Viewer interface, and the communication between them is obtained through a Communicator component. When a node is selected in one tree, the corresponding information on the other tree is highlighted. Thanks to this visualization strategy, it is possible to navigate the extracted information in a straightforward way.

Another example for NLP-based clinical summarization is given by Bashyam et al. [100]. The developed system extracts, structures, and presents the information included in multiple patient records. As a first step, an IE module uses NLP techniques to identify problems and findings, along with the associated properties and relationships. Then, a second module characterizes this extracted information along four dimensions:

time, space, existence, and causality. Finally, the obtained categorizations are used by a third module that displays the information in an integrated format. In this format, findings are organized into a problem list, that can be sorted based on anatomic location. Elements from the list can be selected to populate a timeline grid, where cells are color coded to show existential information at a certain time. By clicking on a specific cell, finally, it is possible to access available reports and images.

As a final, recent example, HARVEST is a longitudinal patient record summarizer that extracts content from patient notes and aggregates information from multiple care settings [101]. The system consists of two online processing modules: a distributed HL7 message and visit parsing module, and a web-based patient-specific visualization module. In the content extraction process, clinical notes are processed by extracting document structure elements (e.g., sections, sentences) and mentions of problems. In the visualization phase, the summarizer retrieves the data available for a single patient, and displays the extracted information on three different panels: a timeline, a problem cloud, and a note display panel. For each visit, the timeline includes a mark that indicates the visit type (e.g., clinic, inpatient), and an information bubble providing additional meta-information (e.g., visit date, attending physician, primary billing code). By using a slider, the user can select a certain time range on the timeline: the concepts and the notes available for this time range are used to populate the problem cloud and the note display panel, respectively. HARVEST was deployed at the New York Presbyterian Hospital in September 2013, and has been used by physicians as a support tool for reviewing patient data.

Chapter 3

Materials and methods

This chapter describes the methodological approaches proposed in this research activity, providing details on their implementation. Section 3.1 presents the main corpus considered for the development of IE techniques. Section 3.2 illustrates the manual annotation that was conducted on a subset of this corpus. Section 3.3 outlines the complete IE pipeline developed as part of this work. Section 3.4 describes the methodologies proposed for the extraction of clinical concepts (i.e., the events), while Section 3.5 presents an ontology-driven approach for the identification of their attributes. Section 3.6 illustrates the methods used for temporal expression identification and normalization. Section 3.7 discusses the solution proposed for identifying temporal links. In addition, it presents the approach used for summarizing the information extracted from multiple reports of the same patient. Finally, Section 3.8 describes the evaluation conducted for each IE step and for the timeline reconstruction task.

3.1. Main dataset

As a main clinical case to support the development of NLP techniques for the Italian language, a set of clinical texts belonging to the molecular cardiology domain was considered.

The documents used in this research activity were provided by the Molecular Cardiology Laboratories of the ICS Maugeri hospital in Pavia, Italy. This corpus is made up of medical reports belonging to patients with inherited arrhythmias, such as Long QT Syndrome and Brugada Syndrome. To overcome any privacy issues, all the considered reports are anonymized, i.e., they do not contain any sensitive patient data (e.g., names, addresses, telephone numbers). In Figure 3.1, as a descriptive statistic, the distribution of the most frequent diseases is reported.

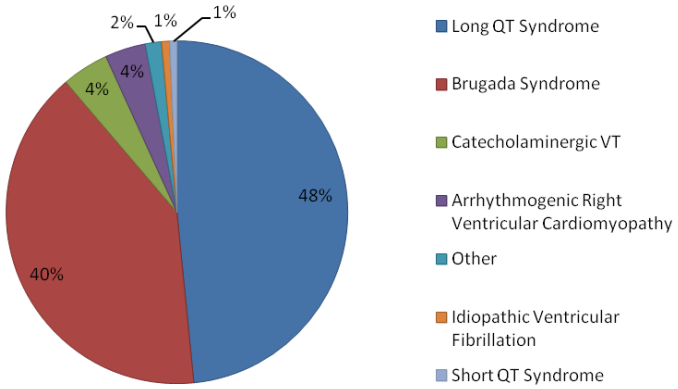

Figure 3.1: Distribution of diseases in the considered corpus.

To carry out this research activity, a set of 5432 reports was used. This dataset, which will be referred to as *CARDIO dataset*, was obtained after cleaning the original corpus to remove a few duplicate instances and those reports that did not include a specific date.

Reports in the CARDIO dataset cover a time span of 6 years (2010-2015), and each report contains the visit date. Since patients are followed over time, in several cases there is more than one report referred to the same patient.

Although the considered documents are not structured in a fixed way, most are organized in sections, including an anamnestic fitting, the family history, information on performed tests, and a conclusion with possible drug prescriptions. The anamnestic fitting provides a summary of the patient's clinical history, including past diagnostic procedures, found problems, and prescribed treatments. The family history section describes the relevant events involving the patient's relatives, such as cardiologic diseases and sudden deaths. The sections referring to diagnostic procedures (e.g., ECG section) provide a detailed description of the performed tests, including their results. In the conclusions section, the reached diagnostic considerations are reported, together with possible drug prescriptions and information on future follow-up visits.

In Figure 3.2, an example of one medical report in the CARDIO dataset is shown. The visit date, i.e., the document creation time, is written in the first line (*Date: Pavia, 22 February 2014*). In this example, the document is composed of the following 7 sections: anamnestic fitting, family history, physical examination, electrocardiogram test (ECG) results, effort stress test results, Holter ECG test (a 24-hour ECG test) results, and conclusions. As an example of one section describing a test, the ECG section provides the following relevant data: type of rhythm (*sinus rhythm*), measured heart rate (*57 bpm*), PR interval length (*156 msec*), QRS interval length (*106 msec*), QRS axis value (*40°*), QT interval length (*430 msec*), and corrected QT interval length (*425 msec*). The report conclusions contain one drug prescription, including the drug name with its dosage and format: "*CORGARD 80 mg (nadolol): 1 tablet in the morning*".

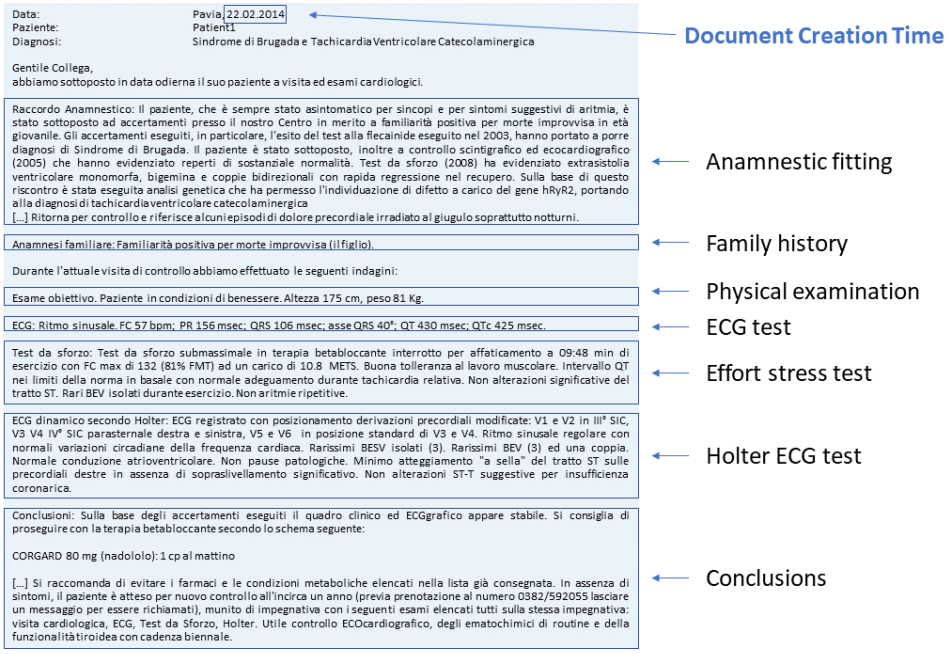

Figure 3.2: Example of one medical report in the CARDIO dataset. The document creation time and the 7 sections composing the report are highlighted.

At the moment, a selection of the data written in the textual reports is manually entered in a hospital system, named Transatlantic Registry of Inherited Arrhythmogenic Diseases (TRIAD) [102]. This system stores data on diagnoses, tests, prescriptions, and other relevant events in the field of genetic mutations and inherited arrhythmias.

3.2. Corpus annotation

As explained in Section 2.1.1, annotating a corpus with the information of interest is one important step to enable the development and the evaluation of an IE system. In the case of supervised machine learning, in fact, the availability of annotated data is essential for both training and testing the classification models. In this work, a subset of 75 documents were randomly selected from the CARDIO dataset, and manually annotated with the information of interest.

To annotate documents in an effective way, the first step is to formalize the IE problem, identifying the information to be automatically extracted. Once the extraction task is well defined, it is a good practice to define clear and exhaustive rules that serve as a guide for the annotation process. In this section, the manual annotation conducted as part of this work is described. The developed annotation guidelines are reported in Appendix 2.

The aim of this research is to ultimately build a system that retrieves and summarizes the events included in multiple unstructured reports, taking into account also the available temporal details. To this end, the annotation strategy was developed based on previous work in the field of temporal IE applied to clinical narratives. For defining the entities to be annotated, the THYME annotation guidelines were considered [11]. In particular, the definitions of the Event and the TIMEX tags were reused to define relevant clinical concepts and temporal expressions, respectively. With respect to temporal relations, identifying these links inside the text is regarded as a harder task than entity annotation: the set of all possible relations in a document is essentially quadratic to the number of events and time expressions, which makes it nearly impossible to annotate all temporal links by human means alone [103]. In the scope of this work, to avoid introducing a layer of complexity in the task definition, it was therefore decided not to annotate temporal relations.

Annotations were performed by exploiting Anafora, a web-based tool that enables multiple annotators to access documents remotely [104]. Anafora provides simple representations of the data used in the annotation process: annotation schemas (consisting of tag definitions) and performed annotations are stored as human-readable XML files. In addition, plaintext files are saved alongside annotation data. Thanks to these features, Anafora allows to easily administrate datasets and annotators, enabling the development of annotated corpora in a controlled way.

In Figure 3.3, the Anafora annotation interface is shown. In the proposed annotation schema, event annotations are marked in blue, while temporal expressions are represented in yellow. To annotate a new entity, it is necessary to select its boundaries inside the text. Subsequently, the related properties can be specified in the PROPERTY form.

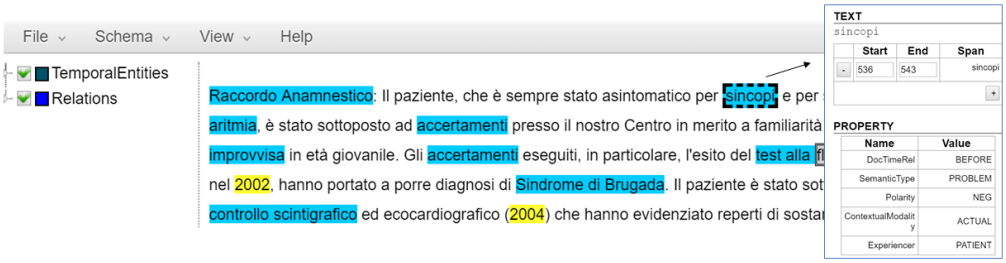

Figure 3.3: Anafora annotation interface.

3.2.1. Event annotation

As an initial step to determine the information of interest in the CARDIO dataset, a small set of 20 reports was selected to be manually reviewed and discussed with clinicians. In this phase, the relevant events to be extracted were identified, and classified into four *semantic types*:

- **Problems**, including diseases, disorders, and relevant health-related issues. As shown in Figure 3.1, the Long QT Syndrome and the Brugada Syndrome are the most prevalent diseases in the CARDIO dataset.
- **Tests**, referring to diagnostic procedures. In the molecular cardiology unit, the most commonly performed tests are the ECG, the Holter ECG, and the effort stress test.
- **Treatments,** mostly related to medication data. For cardiology patients, prescribed therapies are generally beta-blockers and/or anti-arrhythmic drugs.
- **Occurrences**, i.e., events that play a role in the patient's clinical history but are not included in the first three groups (e.g., "admission", "discharge").

To annotate Problems, Tests and Treatments, the UMLS Metathesaurus was used as a guide [105]: the UMLS entries belonging to one of these 3 defined semantic types were considered as the main events to be annotated. For instance, expressions that could be traced back in the UMLS ontology to the "Pharmacologic Substance" or to the "Therapeutic or Preventive Procedure" semantic types were annotated as Treatments. As regards occurrences, a list of relevant events to be searched for was manually created. For this group only, also single verbs were regarded as candidates for event annotation (e.g., "discharged"). As an important remark, overlapping events, such as "Test with Flecainide" (Test) and "Flecainide" (Treatment), were annotated in a few specific cases, i.e., when it was important to maintain information about both events.

For each identified concept, both its boundaries and the selected semantic type were annotated. From the THYME annotation guidelines, three additional properties were captured:

- The *DocTimeRel*, which is the relation of the event to the document creation time. This property encompasses the temporal aspects of the event and has four possible values: OVERLAP, BEFORE, BEFORE/OVERLAP, or AFTER.
- The *polarity* of the event, which can be either POSITIVE or NEGATIVE. Events are usually regarded as negative when they did not happen, or were found not to be true (e.g., "the patient did not experience syncopal episodes").
- The *contextual modality* of the event, which can take one of four values: ACTUAL, HEDGED, HYPOTHETICAL, or GENERIC. Actual events are those having already happened or being scheduled. Hedged events are concepts mentioned with any sort of hedging (e.g., "Suspicious for X"). Hypothetical events are those that might happen in the future, without certainty (e.g., "If X happens, then…"). Generic events are concepts mentioned in a general sense and should not appear on the patient's clinical timeline (e.g., "In all patients with X…").

As events in the CARDIO dataset are often referred to family members (e.g., arrhythmias can be inherited diseases), the *experiencer* of each event was annotated, too. As proposed by Harkema et al., this property can take two possible values: PATIENT, when the event is experienced by the patient himself, or OTHER, when the event is experienced by any other individual [106].

Figure 3.4 shows an example of the described annotations on a portion of text taken from the CARDIO dataset. In this case, three events are highlighted: *sudden death*, which happened to the patient's brother in the past ("before" DocTimeRel and "other" experiencer), *syncopes*, referring to the patient's previous history but in a negated way since he did not suffer from this problem ("before" DocTimeRel and "negative" polarity), and *ECG test*, that was performed on the patient during the current visit ("actual" DocTimeRel).

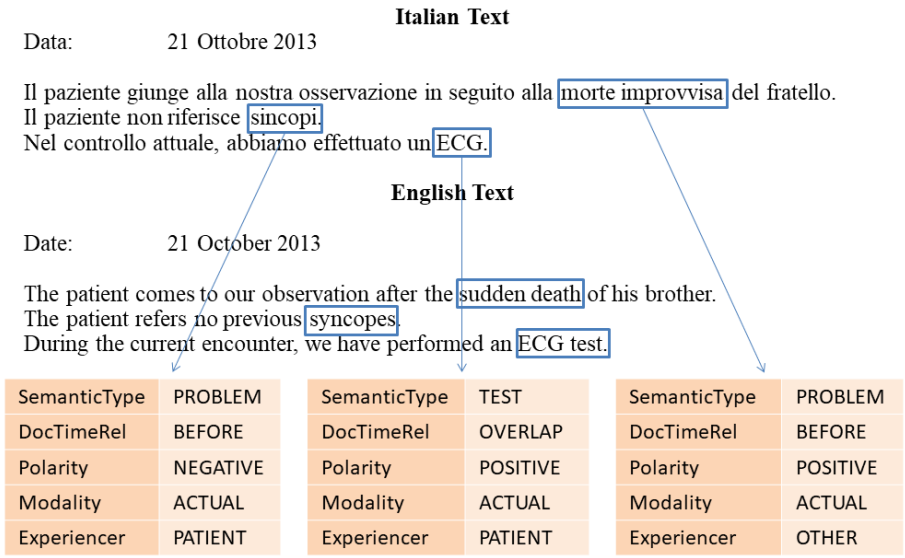

Figure 3.4: Example of Event annotations.

3.2.2. Temporal expression annotation

To address the annotation of temporal expressions for the Italian language, the IT-TimeML guidelines were used for defining the TIMEXes and the properties to be annotated [91].

As a first step in the guideline adaptation process, a small set of 20 reports was selected for manual review, in a similar way as for the Event extraction task. Then, the IT-TimeML annotation rules were carefully read, with the aim to determine the definitions (i.e., entities and properties) that were most useful to capture the temporal information included in the considered documents. Afterwards, a few domain-specific rules were manually developed, dealing with cases not included in the original guidelines. As an important remark, this high-level methodology, though

developed on the cardiology use case, could be easily reused to annotate other corpora including temporal information.

In the IT-TimeML guidelines, a TIMEX entity can be defined as a reference to time. Examples might be phrases like "the 24th of September", "tomorrow", "one month", and "twice a day". To adapt the guidelines to the CARDIO dataset, the following TIMEX properties were considered:

- The *type* property, that classifies the temporal expression into one of four types: DATE, TIME, DURATION, or SET. Dates are temporal expressions describing calendar units (e.g., "1985", "tomorrow"). Times are used to refer to certain times of the day, even if in an indefinite way (e.g., "4 pm", "Saturday night"). Durations denote periods of time not pointing to any specific area in the temporal axis. (e.g., "three days"). Sets are used to describe reoccurring temporal expressions (e.g., "twice a day").
- The *value* property, that assigns to the temporal expression a normalized value representing a calendar date (e.g., "2015-06-04"), a clock time (e.g., "2015-06-04T08.45"), or a special format for durations (e.g., "P1D" for a duration of one day).
- The *mod* property (optional), that is used to signal the presence of certain modifiers. In this research activity, this property was used only to denote approximate temporal expressions (e.g., "about one year").
- The *quant* and the *freq* properties (optional), that are used in conjunction with temporal expressions classified with type SET. *Quant* is a piece of text representing a quantifier (e.g., "every"), whereas *freq* is expressed as an integer and a time granularity (e.g., "2X" for "twice a day").

As previously mentioned, considering that clinical texts might contain additional temporal expressions with respect to the general domain, a few adaptations were required. In particular, specific rules were written for the annotation of drug prescription times and frequencies (e.g., the latin word "*die*" is used to express the "daily" concept).

In Figure 3.5, an example of the TIMEX annotations is depicted. The reported textual portion includes three temporal expressions: the visit date, representing the document creation time ("*21 October 2013*"), a SET temporal expression, related to an event that occurs every year ("*annually*"), and another relevant date ("*1997*"), for which only the year is specified. For these three TIMEXes, the optional properties *mod*, *quant*, and *freq* do not take specific values.

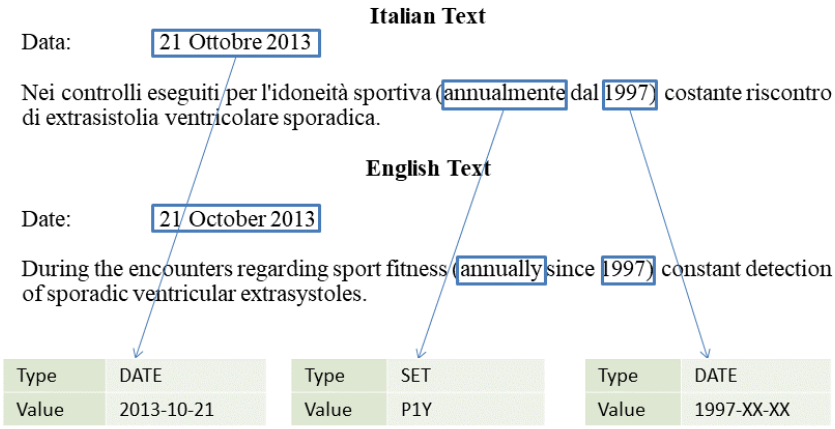

Figure 3.5: Example of TIMEX annotations.

3.3. Information extraction pipeline

As the main contribution of this research activity, a pipeline for the analysis of Italian clinical text was designed and implemented. For each step in the pipeline, one or more annotators were developed using the UIMA architecture [71].

In Figure 3.6, the implemented IE modules are depicted.

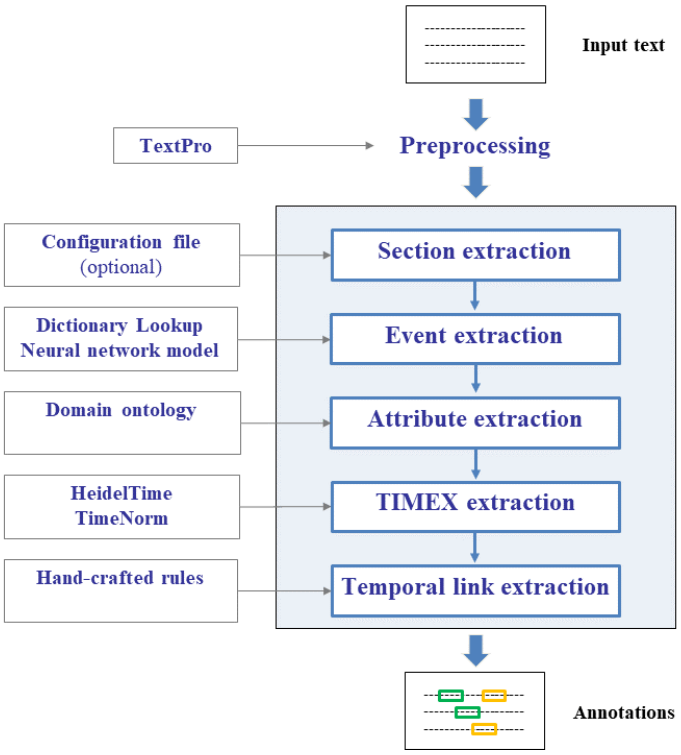

Figure 3.6: Information extraction pipeline.

First, the TextPro tool is exploited to carry out standard preprocessing tasks [75]. In particular, text tokenization, sentence splitting, lemmatization, and POS tagging are performed. Then, preprocessed texts are given as inputs to the pipeline for clinical and temporal information extraction. As an important remark, the document creation time is identified by using a regular expression representing a date (e.g., "\d\d\.\d\d\.\d\d\d\d"). In particular, an external configuration file can be used to specify whether this date should be found at the beginning or at the end of the documents.

The first step in the pipeline involves the identification of sections. This is done by using an optional configuration file containing possible names for sections, such as "*raccordo anamnestico*" (anamnestic fitting). Starting from these names, a Section Annotator constructs a simple *section pattern*, to be searched for in the document. Essentially, this pattern consists of one section name followed by one of four predefined characters: colon, new line, full stop, or comma. If the configuration file is not provided to the Section Annotator, no sections are identified in the text.

The second IE step regards the identification of relevant events and their properties. To address event extraction, two different approaches were developed: the first approach is based on dictionary lookup (Event Annotator and cTAKES Dictionary Lookup Annotator), while the second one relies on a neural network model (Supervised Event Annotator). In these approaches, event semantic types are extracted alongside with the events themselves. The polarity, the modality, and the experiencer properties are then determined by means of an algorithm named ConText [106]. To classify the DocTimeRel of each event, a supervised method based on SVMs was developed (DocTimeRel Annotator).

After events are extracted, the third IE step deals with the identification of additional attributes that can be found in the text. Examples of such attributes are given by drug regimens and test results. To define these attributes and their relationships to relevant events, a domain ontology was designed. To link each event to the corresponding attributes in the text, a rule-based approach was implemented (Attribute Annotator), using the developed ontology.

Moving to temporal IE, the fourth step in the pipeline involves the identification and the normalization of temporal expressions. For extracting TIMEXes from the text, the HeidelTime system [32] was exploited, after performing an extension to the biomedical domain (HeidelTime Annotator). On the other hand, to normalize each temporal expression to a standardized format, both the HeidelTime and the TimeNorm [33] tools were explored (HeidelTime Annotator and TimeNorm Annotator).

Once relevant events and temporal expressions have been identified, one last step is required, i.e., the extraction of temporal relations. As previously mentioned, extracting temporal links is a challenging task, which strongly depends on the results of the previous extraction steps. In this research activity, it was decided to limit the extent of this task by identifying only

Event-TIMEX intra-sentence links (i.e., relations between an event and a TIMEX occurring in the same sentence). To extract these links, a few rules were manually engineered (TLINK Annotator).

Table 3.1 summarizes the methodological approaches and the UIMA Annotators developed for each pipeline step. In the next sections, each approach will be described in detail.

Table 3.1: Developed UIMA IE pipeline.

Step	Task	Methods	UIMA Annotators
1	Section Extraction	Rule-based approach	Section Annotator
2	Event Extraction	Dictionary lookup Neural network classifier SVM classifier	Event Annotator cTAKES Dictionary Lookup Annotator Supervised Event Annotator DocTimeRel Annotator
3	Attribute Extraction	Rule-based approach (ontology-driven)	Attribute Annotator
4	TIMEX Extraction	Rule-based approach	HeidelTime Annotator TimeNorm Annotator
5	TLINK Extraction	Rule-based approach	TLINK Annotator

To allow developing and evaluating some of these IE steps, it was decided to split the annotated dataset into training and test sets, made up of 60 (80%) and 15 (20%) documents, respectively. As it will be explained in the next sections, the training set was used to train supervised algorithms and to manually refine rules. The test set was used for evaluation purposes.

3.4. Event extraction

For extracting the text spans denoting relevant events, two different approaches were developed and compared: a knowledge-based approach that uses a dictionary lookup of a controlled vocabulary, which does not require annotated data, and a supervised approach based on neural networks.

3.4.1. Dictionary lookup approach

The dictionary lookup approach is a simple, yet useful solution for extracting events in an unsupervised way. In this approach, concepts included in specific dictionaries are searched for in the texts. In an effort to leverage the availability of shared terminologies, the Italian version of

UMLS and the FederFarma Italian dictionary of drugs were used [107]. The Italian UMLS Metathesaurus includes 5 knowledge sources (ICPC, LOINC, MedDRA, MeSH, MTHMST) and about 141700 distinct concepts. The FederFarma dictionary contains about 6500 drug names and 4100 active principles. To better account for all the concepts mentioned in the reports, two additional domain-specific vocabularies were manually developed, containing 38 procedures (Tests), and 30 general events of interest (Occurrences), respectively. These vocabularies are reported in Appendix 1. To expand the list of concepts to be searched for, an additional dictionary of acronyms was created, including 29 expressions that are commonly used in the CARDIO dataset.

To extract the terms included in the FederFarma dictionary and the hand-crafted lexicons, an Event Annotator was developed. This annotator receives the external dictionaries as inputs, and looks up the available terms inside the text. To allow identifying both singular and plural forms, the search is performed on TextPro normalized tokens. As an important aspect, it is possible to provide different input dictionaries to the pipeline through a simple configuration file (e.g., to process documents belonging to other clinical domains).

To identify UMLS concepts, the cTAKES Dictionary Lookup Annotator was exploited [65]. This cTAKES component is available in two versions: the original annotator, essentially looking for dictionary matches inside the text, and a *fast* annotator, which improves the search time thanks to the use of a rare word index [108]. In the developed IE pipeline, this second annotator was employed, targeting the search to the UMLS semantic types representing problems (Sign or Symptom, Injury or Poisoning, Pathologic Function, Disease or Syndrome, Mental or Behavioral Dysfunction, Neoplastic Process, Cell or Molecular Dysfunction, and Experimental Model of Disease), diagnostic procedures (Diagnostic Procedure), and treatments (Pharmacologic Substance, Antibiotic, and Therapeutic or Preventive Procedure). Although a detailed description of rare word indexing is not currently available in the cTAKES documentation, this annotation process substantially requires six steps:

1. Get *lookup windows*, i.e., strings representing the inputs to the dictionary lookup system;
2. For each lookup window, get *lookup tokens*, i.e., candidate tokens to be searched for in the dictionary;
3. For each lookup token, get matches in a dictionary index;
4. For each token match, check lookup window for a full text match*;*
5. For each full text match, create a corresponding concept;
6. Store extracted concepts as UIMA annotations;

As mentioned earlier, the Italian UMLS Metathesaurus was used as the dictionary source for the annotation process. To be used inside the cTAKES annotator, it was converted to a suitable dictionary index, required by the annotator itself. To integrate the resulting annotator within

the IE pipeline, TextPro sentences were used as lookup windows, while TextPro normalized tokens were used as lookup tokens. In this way, the identification of both singular and plural concepts was enabled.

3.4.2. Neural network classifier

As explained in Section 2.1.4.1, RNN models represent a good strategy for solving sequence labelling problems like entity recognition: they can process sequential information through cyclical connections [109]. Moreover, RNN models do not rely on manually engineered features, and are able to learn representations that are useful to describe input-output relations inside the dataset. For these reasons, in this research activity, the RNN architecture was investigated and applied to the task of event extraction. To develop the supervised RNN classifier, the annotated portion of the CARDIO dataset was exploited.

The classification model. To treat the event extraction task as a sequence labelling problem, the annotated corpus was converted to the BIO format, classifying each token as belonging to the span of one event (Beginning or Inside) or not (Outside). In the CARDIO dataset, relevant events are those belonging to one of four semantic types. To take this aspect into account, the information on semantic types was included in the token labels, using the format "BIO label – semantic type". Using this notation, the sentence "*we performed an ECG test*" would be translated to the sequence "*O O O B-TEST I-TEST*". Since the standard BIO format does not allow classifying a single token into multiple entities, for overlapping events such as "Test with Flecainide" and "Flecainide" only the longest event was kept, including its semantic type. In this way, the aim was to build a classifier that recognizes the longest, and possibly the most specific event.

Figure 3.7 shows the proposed methodological approach. The classification model takes as inputs the sentences of N tokens identified in the preprocessing phase. For each sentence, the output is a sequence of B, I, O labels including semantic types. As shown in the figure, the model includes an embedding layer and an RNN layer. The embedding layer is used to compute the features that serve as input to the RNN. As it will be explained in the following, these features are created starting from the tokens of a sentence and their POS tags.

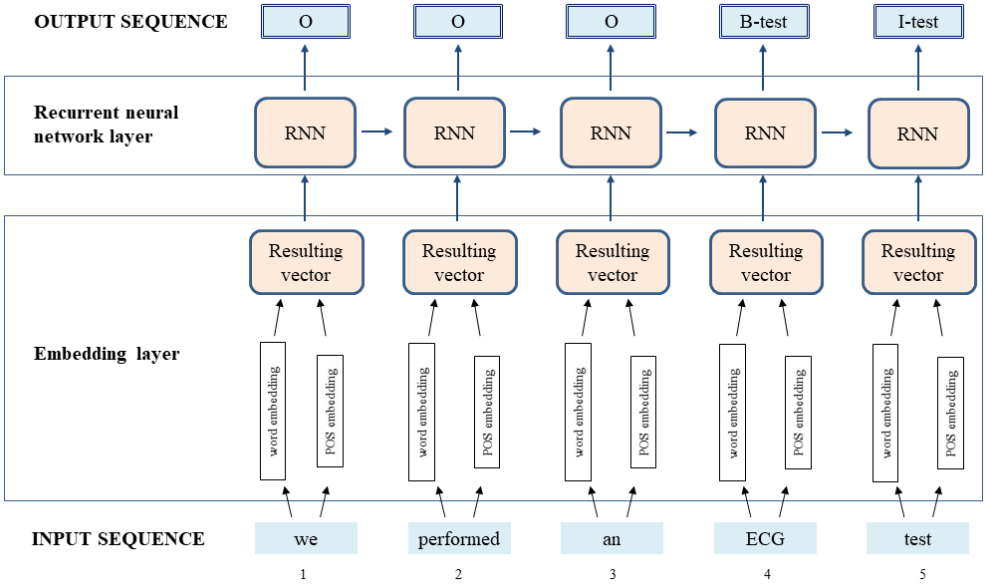

Figure 3.7: RNN model for supervised event extraction.

Embedding layer. In the developed model, both the tokens and their POS tags are converted into real-valued vectors of fixed size. The choice to include POS tags was made to account for the fact that identified BI sequences should correspond to specific POS sequences (e.g., noun-adjective, noun-preposition-noun). For token embeddings, 200-dimensional vectors were chosen, while POS embeddings were represented with 40-dimensional vectors. To define these dimensions, a simple criterion was followed: since the number of possible POS tags is smaller than the vocabulary size, it was assumed that POS tags could be effectively represented by using a smaller number of features with respect to tokens. For each input token, the token embedding and the POS embedding are concatenated, resulting in a 240-elements vector. Each of these vectors represents the n-th input to the RNN layer.

To pre-train word embeddings, a large corpus was created by merging all the available cardiology reports (32300 different words) with 3000 general domain documents gathered from the web, such as Wikipedia pages (103800 different words) [110]. As medical reports often include general domain expressions that should not be confounded with domain-specific concepts, combining these two sources was considered as useful to extend the number of "context examples" for these expressions. For pre-training the embeddings on the created corpus, the *word2vec* implementation of the CBOW algorithm was used [111]. The learnt representations were finally used to initialize the word embeddings, which were then learned in the developed model.

Recurrent neural network layer. In the developed RNN model, at each step n = 1…N, the input $\mathbf{x_n}$ is given by the 240-dimensional embedding

representing the n-th token, and the output $\mathbf{y_n}$ is given by the B, I, O label for that token. As shown in Figure 3.7, to compute each output label $\mathbf{y_n}$, the RNN considers both the current input $\mathbf{x_n}$ and information coming from the processing of previous tokens (i.e., the left context). For standard RNN architectures, the extent of previous context that can be actually accessed is quite limited, due to an effect known as the *vanishing gradient problem* [112]. The Long Short-Term Memory (LSTM) architecture is a well-known RNN model that is able to successfully store and access information over long periods of time [113]. To do so, this model maintains a state of the performed computations, and relies on specific structures, called *gates*, to learn which information should be let through and which should be "forgotten" [51]. In particular, the LSTM architecture includes three gates that operate for each new input: a "forget gate", to control the amount of information to be removed from the state, an "input gate", to decide how much of the new information will be stored in the state, and an "output gate", to select the information to be given as the output. In order to perform this control action, the gates are characterized by their own parameters, which have to be learnt during the network training phase.

Although LSTM networks provide an effective solution to the vanishing gradient problem, they also rely on a rather complex model, containing many weights to be learnt. To reduce the model's complexity, a simpler variation of the LSTM architecture has been recently proposed: the Gated Recurrent Unit (GRU) [114]. Among other modifications, GRU networks rely on only two gating components, combining the forget and the input gates into a single gate. Despite using a simpler model, the GRU architecture was shown to provide comparable results with respect to LSTM [115]. For this reason, the GRU architecture was chosen to implement the RNN layer shown in Figure 3.7. For implementing the complete network, Keras, an open source library written in Python, was exploited [116].

The developed model was trained on the annotated training set, and the resulting classifier was integrated into the IE pipeline. This integration required the implementation of a UIMA annotator (Supervised Event Annotator) that: (i) runs the RNN classifier on each TextPro sentence, and (ii) stores the output B-I sequences (i.e., the identified events) as proper UIMA annotations.

3.4.3. Properties extraction

In the IE pipeline, the dictionary lookup approach and the RNN classifier are used to extract event boundaries together with semantic types. For each identified event, four additional properties of interest are determined: the DocTimeRel, the polarity, the modality, and the experiencer. For identifying the DocTimeRel property, a suitable SVM classifier was

developed. The remaining three properties were extracted by exploiting an algorithm named ConText [106].

DocTimeRel SVM classifier. The supervised classifier for the DocTimeRel property was built by using SVMs [46]. The reason underlying this choice is that SVMs have been proven successful on a variety of classification task, including information extraction from clinical text written in Italian [86].

The DocTimeRel classifier developed in this research activity takes as input the event itself, and returns its relation to the document creation time as the output (OVERLAP, BEFORE, BEFORE/OVERLAP, AFTER). In particular, the classifier uses 8 features: the first token of the event, its POS tag, the section in which the event is found, the temporal tense of the first verb in the sentence, and four features representing the event's context (the 2 preceding and the 2 following tokens). To compute the POS tag and the verb temporal tense, the annotations produced by TextPro were used. For obtaining the event's section, the available Section annotations were exploited.

The DocTimeRel classifier was implemented through the *libsvm* library [117], training the model on the annotated training set. Based on this model, a suitable UIMA Annotator was integrated into the IE pipeline (DocTimeRel Annotator).

ConText algorithm. The ConText algorithm was developed to infer the status of a clinical concept with regard to three specific properties: Negation (affirmed, negated), Temporality (recent, historical, hypothetical), and Experiencer (patient, other) [106]. To determine these properties, the algorithm uses a few simple lexical clues occurring in the context of the concept itself. In particular, it searches for *trigger terms* preceding or following the concept. The underlying idea can be easily illustrated for the Negation property: the assumption is that a certain concept occurring in the text is affirmed by default, and a departure from the default value (i.e., a negated value) can be inferred when a trigger term denoting negation occurs "close" to the concept itself. This same idea is applied for the Temporality and the Experiencer properties.

Besides using trigger terms that change the default value of a contextual property, ConText also uses *pseudo-trigger terms*, corresponding to expressions that contain trigger terms but should not act as such. For example, the term "no increase", which includes a trigger term for a negated concept ("no"), should not prompt a change in the value of the polarity property.

The portion of text to which a trigger applies, i.e., its scope, usually includes all the concepts following the trigger term until the end of the sentence. However, specific *termination terms* can signal the end of this scope, indicating that the following concepts should not be affected by the presence of the trigger term. For example, the term "because" serves as termination term for the Negation property.

In the ConText system, all the trigger terms, the pseudo-trigger terms, and the termination terms for each property are listed in a specific lexicon. Although this resource was originally developed for the English language, most entries are available in other languages, too [118]. In this research activity, the original English lexicon was translated to the Italian language, and the algorithm was exploited as it is to identify negations, hedged conditions, and the event experiencer. For identifying generic and hypothetical events, a set of simple rules were defined by manually looking at a few reports in the annotated training set.

3.5. Attribute extraction

In the CARDIO medical reports, events are often mentioned together with a set of relevant attributes, with specific values. As explained in Section 3.1, test mentions (i.e., events with type Test) can be related to a number of different results, while drug prescriptions (i.e., events with type Treatment) are usually linked to regimen information. In Figure 3.8, an example of text containing an ECG event and four of its attributes (rhythm, heart rate, PR interval, and atrio-ventricular block) is shown.

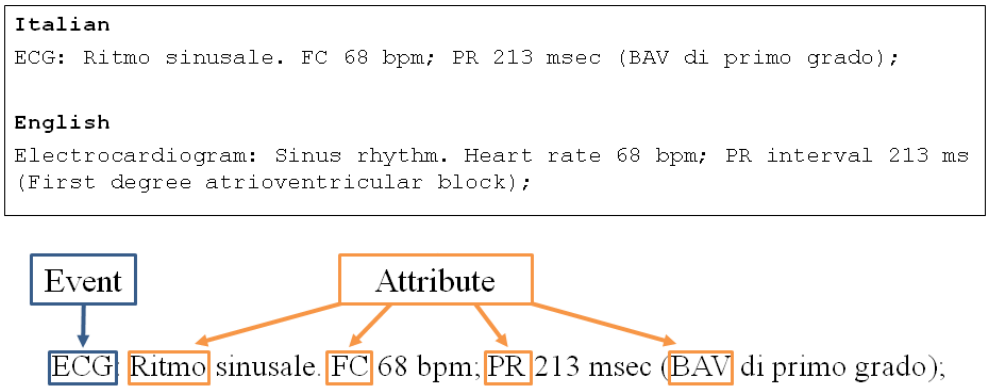

Figure 3.8: One sentence of example. One event and four attributes are highlighted.

Given that the same attribute could be in principle related to many events (e.g., the heart rate is measured in both ECGs and effort stress tests), it is not possible to rely on dictionary lookup alone to extract event-attribute relations. To capture this kind of complex information, in the developed IE pipeline relevant attributes and associated values are extracted by exploiting a domain ontology.

3.5.1. Ontology-driven annotation

The ontology developed in this research activity is structured into Event and Attribute classes, and each event is linked to the corresponding attributes through ontology relations. In addition, the same attribute can be connected to multiple events, without the need to redefine the concept. For example, the information regarding an ECG test is formalized as an Event ("ECGTest") with many Attributes, representing its results and findings (e.g., "AverageHeartRate", "Rhythm"). Some of these Attributes are shared with the Holter tests as well.

All the concepts in the developed ontology are related to a regular expression, which allows searching for concept mentions inside the text. In addition, each Attribute is characterized by a set of properties, such as the value, which can be numeric or categorical. In the first case, the attribute properties also include a unit of measurement, a minimum and a maximum value. In the second case, the value is represented by a set of possible strings. Figure 3.9 shows an example of the illustrated properties for the ECG event and two of its attributes. In particular, the following properties are highlighted:

- For all concepts, the "hasRegularExpression" property is specified, i.e., the string pattern denoting an occurrence of the concept itself;
- The *Average Heart Rate* Attribute includes the "hasUnitOfMeasurement" and "hasNumericValue" properties, representing the possible units of measurement and the range of possible values for the measured heart rate.
- The *Rhythm* Attribute includes the "hasStringValue" property, including all the possible string descriptions for the recorded rhythm (bradycardia, tachycardia, sinus rhythm).

From Figure 3.9, it is possible to notice that the only language-dependent components of the ontology are the properties "hasRegularExpression" and "hasStringValue", which are in this case specified for the Italian language.

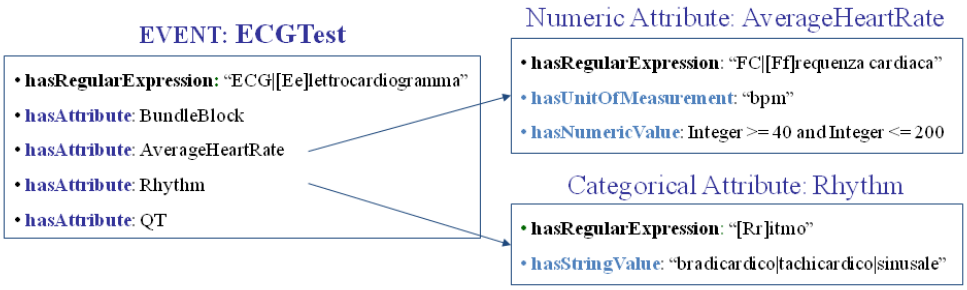

Figure 3.9: Ontology properties for one event and two of its attributes.

From a methodological point of view, the development of an ontology that could be used to support the described IE task, required two initial steps. On the one hand, it was necessary to manually review a subset of medical reports to identify candidate events with related attributes. On the other hand, external knowledge on the considered domain was needed, to verify which information was important to capture. In general, this external knowledge could either be provided by a domain expert, or there might already exist a structured database including the relevant information that can be found inside the text. In either case, this knowledge source can be considered as a gold reference, to be used to define and progressively refine the ontology. In the specific case of the CARDIO dataset, an external structured database (i.e., TRIAD) was exploited. The process followed to define all the ontology classes and their properties will be described in detail in Section 4.3.1.

The ontology was developed in Protégé [119]. To facilitate its use in the UIMA annotation process, the Protégé OWL content is automatically converted to an XML file, which includes events, attributes, and the relationships among them, without additional metadata. This file is given as input to the UIMA Attribute Annotator.

The Attribute Annotator matches each event extracted by the pipeline to the corresponding concept in the ontology, and uses relations to identify the attributes to be searched for. Then, it exploits TextPro, Section, and Event annotations to define event-specific *lookup windows* where the identified attributes should be searched for. As a final step, the annotator uses the regular expressions included in the ontology to extract all the attributes and their values.

Lookup windows definition. The lookup windows for extracting attributes and their values are mostly dependent on the semantic types of events.

Events with type Test are generally described in a detailed way, and the related attributes can be scattered across many sentences. Starting from this observation, the lookup window for an extracted test is computed by using both TextPro sentences and Section Annotations. If no sections are found in the document, the paragraph containing the event (i.e., the sequence of characters until the next newline) is used as the lookup window. If at least one section is found in the document, three situations can occur:

1. the event is contained in its matching section (e.g., an ECG test found in an ECG section); the lookup window is defined as the section itself.
2. the event is not included in any section; the lookup window is defined as one paragraph.
3. the event is included in one section not corresponding to that event (e.g., an ECG test found in the Anamnestic Fitting section); to avoid considering “section-related” information which does not

concern the considered event, the lookup window is defined as one sentence.

In case of lookup windows made up of one paragraph, an additional check is performed: if multiple tests are included in the same paragraph, the corresponding windows are truncated not to overlap with each other.

For events with type Treatment, related attributes are generally written close to the event itself. For this reason, the lookup window for a given drug is defined as the TextPro sentence containing that drug. Also in this case, an additional check is performed. If another drug is found in the same sentence, the annotator checks whether this second mention represents another name for the same drug (i.e., it is included between brackets). If so, the second drug is simply ignored, and no lookup window is created for it. In the case the second mention does correspond to a new drug, the lookup windows are truncated accordingly.

In Table 3.2, the criteria illustrated for the definition of lookup windows are summarized.

Table 3.2: Criteria for lookup window definition. For the windows marked with *, an additional check on the presence of multiple events is performed.

Event semantic type	Contextual information	Lookup window
Test	No sections available	One paragraph*
Test	Included in matching section	One section
Test	Not included in any section	One paragraph*
Test	Included in non-matching section	One sentence
Treatment	NA	One sentence*

Attribute extraction. Once the lookup window for one event is defined, the extraction of attribute names and values is performed according to the properties defined in the ontology (regular expressions and numeric/categorical values).

For events with type Test, both attributes' names (e.g., "heart rate") and values (e.g., "78 bpm") are searched for in the text. For most numeric attributes, a rigid pattern is used, composed by the attribute name, a numeric value, and the unit of measurement. As many numeric attributes share the same range of possible values, this "rigid" search is needed to avoid linking a certain value to the wrong attribute name. For categorical attributes, instead, the search is performed in two steps. First, the attribute name is identified by using the regular expression. Then, the attribute value is looked for in the whole surrounding sentence. This "relaxed" search can be performed on specific numeric attributes, too.

As a special case for the identification of attribute names, there exist a few attributes which are written with similar expressions but refer to

different concepts (e.g., basal, stress, and recovery QT interval). In this case, disambiguation is achieved by including in the ontology appropriate attribute modifiers: whenever an "ambiguous" attribute name is found, the related modifier is identified in the context surrounding the concept. For example, if the "QT" string is found in the text, the words "basal", "stress", and "recovery" (with other possible variants) are searched for in the same sentence. To select the specific attribute to be annotated, the closest modifier (in terms of tokens) is considered (e.g., BasalQT).

For events with type Treatment, attribute extraction is performed in a simpler way with respect to tests. In particular, only the attributes values (e.g., "80 mg") are searched for in the text, as the corresponding attribute names (e.g., "drug dosage") are not likely to be explicitly written. For numeric attributes, therefore, only the number and the unit of measurement are considered. In a similar way, only the possible string values of categorical attributes are looked for in the text.

3.5.2. Ontology extensions

The ontology and the Attribute Annotator were developed to process reports written in a specific language (i.e., Italian) and belonging to a specific clinical domain (i.e., molecular cardiology). On the one hand, the language-dependent components of the ontology are represented by the regular expressions referring to attribute names and values. On the other hand, knowledge on the clinical domain is needed to define the attributes to be extracted, as well as their relationships to events.

As the ontology-driven approach is the only step in the IE pipeline which heavily depends on the reports language and on the specific domain, it was decided to verify its extendibility in two different ways. First, the possibility to use the proposed approach to process texts written in English was investigated. Then, the developed methodology was adapted to a different application, involving anatomic pathology reports in the oncology domain. In this section, these two different extensions are described.

Multilingual extension. To assess the extendibility of the proposed approach to other languages, the developed pipeline was adapted to the analysis of English texts. In particular, the English versions of TextPro and ConText, and the English translation of external dictionaries were used. Also, the configuration file for the Section Annotator was conveniently translated. For adapting the ontology, the only step to be performed was the translation of regular expressions, without performing any additional changes. To this end, the "hasRegularExpression_EN" and the "hasStringValue_EN" properties were included in the already existing ontology. As regards the Attribute annotator, the definition of lookup windows remained unchanged, as it is similar for the Italian and the English languages. It is important to point out that, for other languages

characterized by longer sentences (e.g., Japanese or Chinese), it might be necessary to tune lookup windows differently.

Domain extension. In the molecular cardiology domain, identifying events and related attributes was straightforward, also thanks to the availability of the TRIAD system, containing most of the information to be extracted in a structured way. Starting from this consideration, it was considered appropriate to see whether the proposed ontology structure could be easily adapted to a different clinical domain, without the guidance of a structured repository as TRIAD. To assess this adaptability, a set of anatomic pathology reports belonging to the oncology domain was considered. A detailed description of this corpus will be given in Section 5.2.1.

To process the new dataset, the Italian version of the TextPro system was used, and the Section Annotator was adapted by providing an external file with the new sections to be identified in the texts. Also, domain-specific lexicons were manually created for the event extraction task, in a similar way as for the molecular cardiology pipeline.

3.6. Temporal expression extraction

In clinical NLP, besides searching for relevant concepts inside the text, analyzing the extracted information from the temporal point of view is essential. To this end, one first step involves the identification of temporal expressions. Once a temporal expression is extracted, a normalization step is needed to convert it to a standard format, which serves as a formal representation along a timeline.

To extract and normalize the temporal expressions included in the CARDIO dataset, it was decided to exploit two existing unsupervised systems for temporal IE in the Italian language: HeidelTime and TimeNorm. The first tool performs both TIMEX extraction and normalization, while the second tool deals with the normalization task only.

Although both HeidelTime and TimeNorm were already available for the Italian language, they had been developed on general domain corpora, such as newspaper texts [92,93]. In this research activity, both tools were adapted to the clinical domain by manually analyzing the TIMEX entities available in the annotated training set.

3.6.1. HeidelTime and its adaptation to the clinical context

HeidelTime is a rule-based system for the extraction and the normalization of temporal expressions. For the first task, the system uses regular expressions representing the TIMEXes to be extracted. For the normalization task, knowledge resources and linguistic clues are exploited.

HeidelTime was originally developed as a contribution to the TempEval-2 challenge on temporal information extraction from general domain texts

[89]. Figure 3.10 shows the UIMA pipeline that was originally proposed for designing and using the system [32]. On the left, the workflow for system development is shown: the TempEval-2 training set (which includes annotations for tokens and sentences) was used to manually develop the rules for TIMEX extraction and normalization. In this phase, a POS tagger was run on the input tokens, thus obtaining additional information to be exploited inside the rules. To refine existing rules and create new ones, the annotations extracted by HeidelTime were compared to the gold standard TIMEX annotations (TempEval-2 Evaluator). On the right-hand side of Figure 3.10, the workflow for exploiting HeidelTime on a generic dataset is shown. The documents are first split into tokens and sentences, and a part-of-speech tagger is used to extract the token POS tags. The HeidelTime rules are then applied to these input annotations, thus allowing the extraction and the normalization of temporal expressions.

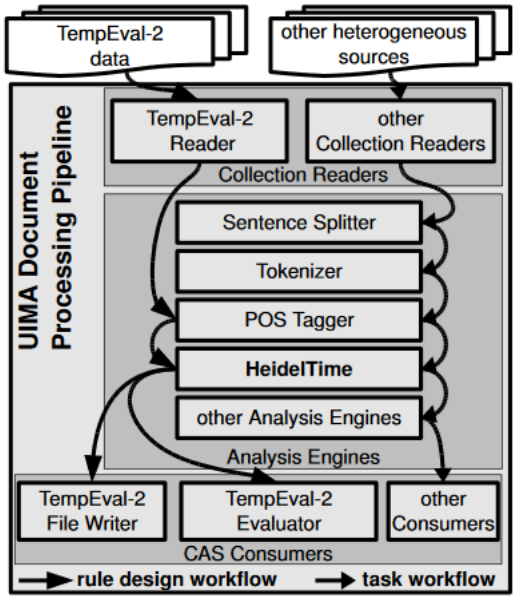

Figure 3.10: UIMA pipeline for designing and using HeidelTime [32].

Within the HeidelTime system, extraction rules are organized into four groups, which correspond to the four types of temporal expressions defined in TimeML (date, time, duration, and set). In a specific group, each rule includes (i) an extraction pattern, which identifies the expression inside the text, and (ii) a normalization function, defining how the extracted expression should be normalized. For both tasks, language-specific external resources are exploited. For example, dates in the format "DD Month YYYY" (e.g., "14 April 2009") are extracted by the following rule:

```
RULENAME="date_r0a",EXTRACTION="%reDayNumber %reMonth
%reFullYear",NORM_VALUE="group(3)-%normMonth(group(2))-
%normDay(group(1))"
```

In this case, the extraction pattern exploits three external resources containing regular expressions for day numbers (`%reDayNumber`), month strings (`%reMonth`), and years (`%reFullYear`). In the normalization function, two external resources specify how to normalize months (`%normMonth`) and days (`%normDay`). For the temporal expression "14 April 2009", the resulting normalized value is given by "2009-04-14".

For *explicit* temporal expressions, such as "14 April 2009" or "20-05-2009", the value attribute can be directly assigned by looking at the temporal expressions itself. However, *implicit* temporal expressions, such as "tomorrow" or "in October", require contextual knowledge to be correctly normalized. More specifically, these expressions can be normalized only when a reference time is available. To address this issue, HeidelTime performs the normalization task in two different steps. First, the extraction rules are applied to every sentence of a document, and extracted TIMEXes are assigned either a specific value or an underspecified format (for implicit expressions). Then, a post-processing step is executed to disambiguate these underspecified values: according to the extraction rule, either the document creation time or the previously mentioned date is used as the reference date for normalization.

The HeidelTime system includes rules and external resources in many different languages, including Italian. To process documents in the CARDIO dataset, the Italian version of HeidelTime was adapted by working on a subset of documents in the annotated training set. This adaptation process required both (i) the extension of a few general domain rules and (ii) the creation of new domain-specific ones. Also, some minor modifications were performed on the UIMA Annotator code.

In Table 3.3, a summary of the main adaptations performed on the HeidelTime system is reported.

Table 3.3: Main adaptations performed on the HeidelTime system.

Type	Modification	Examples
Extension of general domain rules	Correct normalization of dates in the format DD/MM/YY	"21/10/15"
	Added rules for dates in the format DD.MM.YYYY and DD/MM	"21.10.2015", "21/10"
	Added rules for specific sets	"every six months"
Creation of domain-specific rules	Added rules for TIMEXes including the word "die" (daily)	"2/die"
	Added rules to handle multiple times for drug intake	"at 8, 20"
	Added rules to identify the duration of effort stress tests	"7:18 min"
Creation of negative rules	Added negative rules for expressions denoting drug formats	"1/2 cp"
Improvements of the Annotator code	Check maximum distance between a TIMEX and its reference date	NA
	Normalization of approximate TIMEXes	"2-3 days"

The extension of general domain rules mostly concerned the extraction of dates. For example, TIMEXes in the format DD/MM/YY, i.e., using only two numbers for the year, were not correctly normalized for dates following the year 2000 (e.g., the date "21/10/15" was normalized to the value "1915-10-21"). To address this issue, the old rule was removed, and two specific rules were created instead. In addition, dates in the format DD.MM.YYYY and DD/MM were not recognized at all. In this case, it was sufficient to add the corresponding extraction rules: an explicit one was introduced for the former, and an implicit one for the latter. Another extension, finally, involved TIMEXes with type Set. In this case, a few patterns were added to correctly identify expressions such as "every six months".

As regards the creation of new domain-specific rules, most interventions involved TIMEXes related to drug prescriptions. For example, a few specific rules were created to extract temporal expressions including the word "die" (i.e., a daily set). For these TIMEXes, the normalization value was defined as P1D. In addition, a few rules were hand-crafted to handle multiple times for drug assumption (e.g., "at 8, 20" to indicate 8.00 am and 8.00 pm). In this case, for each specific time one TIMEX is identified, and its normalized value is given by composing an undefined date with the extracted time (e.g., "XXXX-XX-XXT08:00" and "XXXX-XX-XXT20:00"). The underlying reason is that the times for drug prescriptions should not be related to a concrete day of the year. As another domain-specific extension, new HeidelTime rules were created to capture the durations of effort stress tests. In particular, these durations are often

expressed in the format "ss:mm" (e.g., "7:18 min"), thus requiring the use of specific terms (e.g., "min") to differentiate the extraction pattern from a standard simple time (e.g., "at 18:00").

To avoid the extraction of expressions that resemble TIMEXes but should not be considered as such, a few "negative rules" were hand-crafted, too. For example, expressions denoting drug formats (e.g., "1/2 cp" stands for "one-half tablet") should not be considered as temporal expressions. To address this specific issue, a negative rule was created by removing all the temporal expressions in the form "number/number" followed by a set of specific strings, i.e., those denoting drug formats and units of measurement.

With respect to the HeidelTime Annotator code, a few changes were carried out to improve the normalization phase. Although HeidelTime deals with implicit temporal expressions by assigning them a reference date, many errors were performed by the original algorithm, which was trained on general domain texts. In particular, most normalization errors regarded TIMEXes related to drug prescriptions. To overcome this issue, whenever an implicit temporal expression has to be normalized, it has been decided to associate it to the last-mentioned date only if it falls within a maximum distance with respect to that date (in terms of characters). Otherwise, the reference to an undefined date is kept ("XXXX-XX-XX"). Another modification on the UIMA Annotator code regarded the normalization of approximate expressions such as "2-3 days", which should be normalized to a mean value like P2.5D. To this end, a function that computes the average value between two numeric inputs was implemented.

3.6.2. TimeNorm and its adaptation to the clinical domain

TimeNorm [33] is a rule-based system for temporal normalization based on a synchronous context free grammar (SCFG). Given an extracted temporal expression and a reference date, the system performs time normalization according to the rules included in the SCFG.

From a general point of view, a formal grammar consists of a set of rules, called *production rules*, which specify how to form syntactically valid strings over a given alphabet. These rules make use of two different types of symbols: *terminal symbols* (those included in the alphabet) and *non-terminal symbols*. In particular, each rule in the grammar associates a non-terminal symbol to a sequence which can be composed of terminal and/or non-terminal symbols. Therefore, the production rules define how to generate a valid string starting from a non-terminal symbol. For example, given a start symbol S, and two terminals a and b, a formal grammar could be given by the following rules:

$$S \rightarrow aa$$

$$S \rightarrow bSb$$

According to these rules, an infinite set of valid strings can be created: *aa*, *baab*, *bbaabb*, etc.

Synchronous context-free grammars are a type of formal grammar that operates simultaneously on a source language and on a target language, specifying the structure of two phrases at the same time [34]. In this case, each production rule defines how to expand a non-terminal symbol in both the source and the target language, simultaneously. In a SCFG, each rule can be written in the form:

$$X \rightarrow (R, T, A)$$

where X represents a non-terminal, R represents its expansion in the source language, T represents its expansion in the target language, and A is the alignment between the non-terminals of R and T. An example of a SCFG is represented by the following rules, which allow translating an English sentence into a Japanese sentence [34]:

$$S \rightarrow (NP_1 VP_2, NP_1 VP_2)$$
$$VP \rightarrow (V_1 NP_2, NP_2 V_1)$$
$$NP \rightarrow (i, watashi\ wa)$$
$$NP \rightarrow (the\ box, hako\ wo)$$
$$V \rightarrow (open, akemasu)$$

The subscripts on non-terminals symbols indicate the alignment between the source and the target parses. In this example, starting from a pair of linked start symbols, such as (S_{10}, S_{10}), it is possible to derive two sentences that are one the translation of the other. This is done by repeatedly applying the production rules on the two sides, simultaneously (respecting the alignments between non-terminals). The results of this process can be viewed as a pair of synchronous trees, as shown in Figure 3.11.

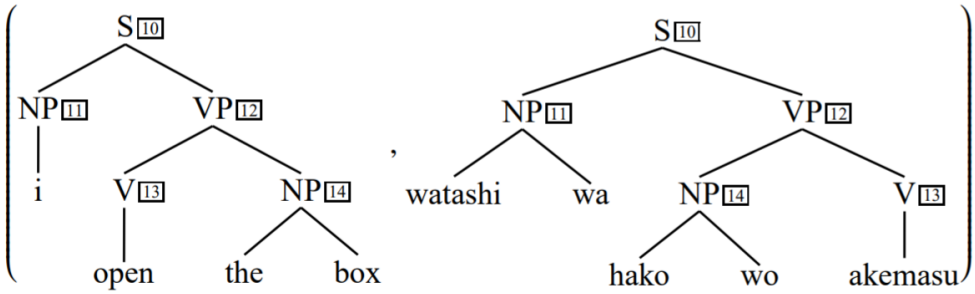

Figure 3.11: Example of an English-Japanese synchronous parse [34].

In the SCFG exploited by the TimeNorm system, the source language is the natural language, while the target language is a formal grammar of temporal operators, defining how to create temporal objects. Starting from a natural language sentence, the set of SCFG rules are applied to build the parse tree for the source side. These parses are then deterministically translated into the corresponding target side parses, using the alignments that are defined in the SCFG rules.

A concrete example of a TimeNorm SCFG rule is given by the following notation:

$$X \rightarrow (january, \text{MONTH\_OF\_YEAR } 1)$$

In this case, the non-terminal X represents a month of the year, R is a specific string month, i.e., "January" (the expansion of X in the source language), and T is a sequence representing the corresponding month number (the expansion of X in the target language).

Although TimeNorm was originally developed to process English texts in the general domain, the underlying SCFG was already available also for the Italian language [93]. To exploit TimeNorm for normalizing TIMEXes in the CARDIO dataset, the annotated training set was used to extend the Italian grammar as needed. In particular, to use TimeNorm within the developed IE pipeline, the HeidelTime system was exploited for the identification of temporal expressions and their reference time, which were then given as inputs to the TimeNorm code. The output of TimeNorm was then conveniently processed to correctly extract the type, value and *mod* properties (the optional properties *freq* and *quant* were not considered).

As for the HeidelTime system, adapting TimeNorm required both the extension of general domain rules and the creation of a few entries specific to the molecular cardiology domain. In this case, though, no major modifications were performed on the annotator's code. Table 3.4 reports the main modifications performed on the TimeNorm system.

Table 3.4: Main adaptations performed on the TimeNorm system.

Type	Modification	Examples
Extension of general domain rules	Added rules for identifying expressions like "*min*" and "*h*"	"24 h", "8 min"
	Added rules to handle dates in the format MM/YYYY	"10/2015"
	Added rules to handle approximate TIMEXes	"about 1 year"
Creation of domain-specific rules	Added rules to handle TIMEXes including the word "die" (daily)	"2/die"
	Added rules to identify the duration of efforts stress tests	"7:18 min"

As regards general domain rules, a few variants had to be added to cover those expressions that were not available in the Italian grammar. For example, neither "*min*" (short for "minutes") nor "*h*" (standing for "hours") were recognized by TimeNorm. In addition, dates in the format MM/YYYY were not considered by the normalization rules. As another problem, a few terms denoting approximate values (e.g., "about 1 year") were not available among the Italian entries. To address these issues, it was sufficient to add suitable rules to the TimeNorm grammar.

Moving to domain specific temporal expressions, a few rules had to be created, too. As for the HeidelTime system, it was necessary to consider those expressions that are characteristic of drug prescriptions. In particular, the expressions including "die" or "bid" (i.e., twice a day) were added to the grammar. Also, the duration expressions, typical for example of effort stress tests, were considered.

3.7. Timeline construction

The final aim of this research activity is to reconstruct patient clinical timelines starting from the information included in multiple textual reports. The previous sections of this chapter focused on the extraction of events (Section 3.4) and temporal expressions (Section 3.6) as two independent tasks. In this section, the problem of linking each extracted event to a corresponding reference TIMEX is addressed. In addition, an approach to summarize multiple reports referring to the same patient is described.

3.7.1. Temporal link extraction

Extracting temporal relations from a single document represents a rather complex task, as the number of possible links is given by all the possible combinations of entity pairs available in the document itself (Event-TIMEX, Event-Event, or TIMEX-TIMEX). In an effort to simplify this extraction task and still obtain useful results, it was decided to create temporal links only between Event-TIMEX pairs included in the same sentence. In addition, following the approach proposed by the THYME Annotation Guidelines, only five possible temporal relations were considered [11]:

- BEFORE: one entity happens before the other;
- BEGINS_ON: one entity begins on a certain date/time;
- ENDS_ON: one entity ends on a certain date/time;
- CONTAINS: one date/time completely contains another entity;
- OVERLAP: two entities overlap, i.e., one entity starts before the other ends;

In this work, the candidates for temporal link extraction were defined as all the Event-TIMEX pairs found in the same sentence. The subsequent classification task consisted in assigning each extracted pair to one of the following classes: BEFORE, BEGINS_ON, ENDS_ON, CONTAINS, OVERLAP, and NOLINK. The NOLINK relation was used to classify those extracted pairs that are not temporally related (e.g., a symptom which was not present on a certain date).

Given the unavailability of annotated data for temporal link identification, the TLINK Annotator was developed by manually creating a set of extraction rules. To define the features that could be used for rule development, the relevant literature in this field was reviewed [120–122]. For example, D'Souza and Ng investigated the task of temporal relation extraction and classification in the clinical domain [121]. To categorize each extracted link, the authors developed a system that combines a machine learning approach and a rule-based approach. As reported by the authors, the basic features for temporal relation classification can be divided into six categories: lexical (e.g., entity strings), grammatical (e.g., POS tags), entity properties (e.g., event polarity, TIMEX type), semantic (e.g., inclusion in semantic dictionaries), distance (e.g., entities belonging to the same sentence), and related to the document creation time (e.g., DocTimeRel class). In another paper, Mirza and Tonelli explored both temporal and causal relation extraction from general domain texts [122]. Also in this case, a combination of supervised machine learning and rule-based modules was exploited for temporal link classification. As an interesting aspect, to classify Event-TIMEX links, the authors built a set of rules exploiting the temporal sense of some prepositions (e.g., "since", "until") [123].

Based on the illustrated revised works, the following features were investigated for temporal link extraction:

- Event string: the string representing the extracted event (e.g., "Brugada Syndrome", "electrocardiogram", "Visit");
- Event DocTimeRel: the relation of the event to the document creation time (Before, After, Overlap, Before/overlap);
- Event semantic type: the semantic type of the event (Problem, Test, Treatment, Occurrence);
- Event polarity: the polarity of the event (Positive, Negative);
- Event section: the section in which the event was found (e.g., "Anamnestic fitting", "ECG test");
- TIMEX string: the string representing the extracted temporal expression (e.g., "05/07/2009", "the following day");
- TIMEX type: the type of the temporal expression (Date, Time, Duration, Set);
- TIMEX value: the normalized value of the temporal expression (e.g., "2009-07-05");

- Event-TIMEX distance: the distance between the two entities in terms of tokens;
- Temporal preposition: the presence of specific prepositions encompassing a temporal sense;
- Verb temporal tense: the temporal tense of the first verb in the sentence (e.g., past tense, future tense);
- Temporal verbs: presence of specific verbs denoting a "start" or an "end" (e.g., "begin", "terminated");

To verify whether these features could be effectively used for rule development, a set of 5 reports was randomly selected in the annotated training set. The IE pipeline was then run on these documents to (i) extract all intra-sentence Event-TIMEX pairs and (ii) compute the selected features for each extracted pair. As a final step, the correct temporal link was manually assigned to each extracted pair. Table 3.5 shows an example of an automatically extracted Event-TIMEX pair, including the computed features. In this example, the correct temporal link is CONTAINS.

Table 3.5: Event-TIMEX pair: computed features.

Element	Example EN	Example IT
Sentence	During the previous encounter in January 2010, an Holter was performed.	Nel corso del precedente controllo del gennaio 2010, è stato eseguito Holter.
Event	encounter	controllo
TIMEX	January 2010	gennaio 2010
Event DocTimeRel	Before	Before
Event semantic type	Test	Test
Event polarity	Positive	Positive
Event section	Anamnesis	Raccordo Anamnestico
TIMEX type	Date	Date
TIMEX value	2010-01-XX	2010-01-XX
Event-TIMEX distance	1	1
Temporal preposition	none	none
Verb temporal tense	past	passato
Temporal verbs	none	none

After having identified the temporal links included in the analyzed documents, a list of rules was manually created to classify new Event-TIMEX pairs. In Table 3.6, a few examples of rules are shown. For each rule, a possible sentence is given ("Example" column), including the event and the TIMEX to be associated (both between square brackets).

Table 3.6: Examples of TLINK rules.

Rule	TLINK	Example
timex type = DATE & *temporal verb type* = BEGIN ("initiate")	BEGINS_ON	**IT:** Nel [febbraio 2012], si è deciso di intraprendere terapia con [Chinidina]. **EN:** In [February 2002], it was decided to initiate [quinidine] therapy.
timex type = DATE & *temporal preposition* = "since" & *event polarity* = NEGATED	ENDS_ON	**IT:** Da [marzo 1986], nessun altro [episodio sincopale]. **EN:** Since [March 1986], no other [syncopal episode].
timex type = DATE & *timex value* = PRESENT_REF	OVERLAP	**IT:** Durante l'[attuale] controllo, abbiamo effettuato un [ECG]. **EN:** During the [current] encounter, we have performed an [ECG test].

To identify the temporal links inside the text, the TLINK Annotator performs two different tasks. First, it extracts all the possible candidate pairs by assigning to each Event the closest TIMEX belonging to the same sentence. Then, for each extracted pair, it selects the correct temporal link by applying the set of manually created rules. In the case of a NOLINK class, no temporal link is created.

3.7.2. Timeline reconstruction

As previously mentioned, extracting information from single documents represents only the first step towards clinical timelines reconstruction. To correctly summarize all the clinical information available for one patient, all the reports belonging to that patient must be processed, and the extracted data consistently aggregated. In this section, the proposed approach for summarizing the information extracted from multiple reports of the same patient is described in detail.

Methodological approach. All the different steps involved in clinical timeline reconstruction are shown in Figure 3.12:

1. The patient of interest is selected.
2. The medical reports referred to the selected patient are retrieved, and given as inputs to the NLP pipeline for single document information extraction.
3. The NLP pipeline processes the retrieved patient documents. Each report is automatically annotated with extracted events and associated attributes, as well as temporal information. In particular, the DocTimeRel of each event is identified at this point.

4. The events extracted from all patient documents are aggregated and visualized on a timeline. If one event is mentioned multiple times with reference to the same date, only one entry is created.

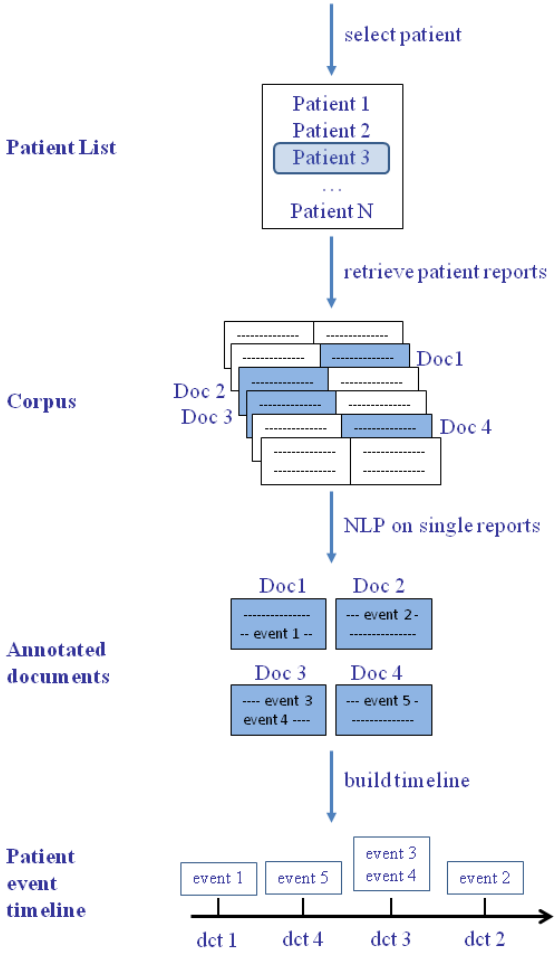

Figure 3.12: Steps for building a patient's clinical timeline.

As a first attempt, it was decided to display on the timeline only those events that are related to the document creation time by a DocTimeRel of type OVERLAP. In this basic approach, each extracted event is associated to the report creation date, and additional event-related information is included in the timeline as well. Specifically, for those events that are related to a set of attributes, extracted values are visualized too. For events that are not associated to attributes in the text, the section where the event was found, the event polarity, and the event experiencer are shown.

In the second version of the patient timeline reconstruction, events that do not "overlap" the document creation time were considered, too. According to their specific DocTimeRel value (BEFORE, BEFORE/OVERLAP, or AFTER), these events are likely to be related to

dates or times that precede or follow the report date. Hence, linking them to a specific reference TIMEX is needed to enable timeline reconstruction.

To deal with events that precede or follow the document creation date, the output of the TLINK Annotator was exploited. In particular, only the temporal relations with type "CONTAINS" were considered. There were two main reasons underlying this choice. First, this type of temporal link allows assigning to the involved event a specific point in time, which is particularly convenient for an unambiguous timeline reconstruction. For the other relation types, instead, further reasoning would be needed to precisely identify the temporal boundaries of each event. As a second motivation, the CONTAINS links were the most frequent type of temporal relations in the CARDIO dataset. Therefore, it was assumed that most of the patient timeline could be effectively reconstructed by only considering these link types.

Implementation details. To reconstruct the clinical timeline for one patient, the data extracted from all the reports referring to that patient are aggregated and visualized through the TimelineJS tool [124].

TimelineJS is an open-source tool that allows creating rich and interactive timelines. In this framework, a timeline is defined as a list of events, each related to a specific time. From the technical point of view, timelines are stored as JSON objects with four properties: events (the timeline itself), title (the timeline title), eras (objects that are used to label a span of time), and scale (a property that allows dealing with dates in the very distant past or future). Among these properties, the "events" object represents the actual timeline, including a list of "slide" objects, each with the following properties:

- start_date: a "date" object representing the event's starting date;
- end_date: a "date" object representing the event's end date;
- text (optional): a "text" object including a headline and a textual content;
- media (optional): a "media" object that allows including different content types, such as images and videos;

When a patient is selected, the system extracts and aggregates all patient events, saving the resulting timeline according to this TimelineJS JSON format. In particular, each "event" object is filled with an extracted event: the associated TIMEX is used to define the start_date and the end_date properties, while the event-related information (e.g., its attributes) is included in the text property.

For events that "overlap" the document creation time, this reference date is used as both the start date and the end date. For the other events, a CONTAINS temporal relation to a specific TIMEX is searched for. If such relation is found, the event is added to the timeline, using the related temporal expression to determine its date.

3.8. Proposed evaluation

In this section, the evaluation performed for each of the IE steps is described. For the event extraction and the TIMEX extraction tasks, the evaluation was conducted against the manually annotated dataset. For the attribute extraction task, the data included in the TRIAD system was considered as the gold standard.

As regards the timeline reconstruction task, considering the lack of a reference to be used for a quantitative evaluation, a preliminary manual validation was performed.

3.8.1. Event extraction evaluation

To evaluate results on event extraction, the annotated test set (gold events) described in Section 3.2.1 was used to compare different configurations. First, the dictionary lookup approach and the RNN classifier were applied separately, and then the output of the two approaches were merged.

The extraction of event text spans was evaluated using precision, recall, and F1 score. Gold events whose boundaries are correctly identified by the system, i.e., there is an exact match between the two events' offsets, represent true positives (TPs). Gold events that are not detected by the system are considered as false negatives (FNs), while events extracted by the system but not found in the gold annotations are considered as false positives (FPs). Precision (P), recall (R), and F1 score (F1) are computed as follows:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F1 = \frac{2 \cdot P \cdot R}{P + R}$$

For events marked as TPs, the performance of property extraction was calculated as well. For each property (semantic type, DocTimeRel, polarity, modality, and experiencer), the accuracy (acc) was computed as the number of correctly extracted values over the total number of extracted properties (the event TPs):

$$acc = \frac{\#\ correct\ values}{\#\ event\ TPs}$$

To analyze the effects created by a potential class imbalance, the F1 score of each property value was computed, too. As a matter of fact, this evaluation procedure highlights the effect of the misclassifications performed on minority classes.

3.8.2. Attribute extraction evaluation

The attribute extraction performance was evaluated by running a basic configuration of the IE pipeline, using the Section Annotator, the dictionary lookup Event Annotator (in this case, only the ontology events were considered), and the Attribute Annotator itself, which was the focus of the validation. Besides evaluating the event-attribute extraction system on the CARDIO dataset, two other scenarios were tested: (i) the extendibility to the English language, and (ii) the extendibility to the oncology domain.

Main evaluation on the CARDIO dataset. Given the unavailability of annotated reports, the proposed approach was evaluated against TRIAD, the hospital system that stores data on diagnoses, tests, prescriptions, and other relevant events. It is important to point out that there is not an exact alignment between information in reports and data in TRIAD: some information could have been written in the documents but not transferred to TRIAD, or the electronic data can come from sources other than the reports.

The steps of the evaluation are shown in Figure 3.13. For each event extracted by the IE pipeline (*System Event*), the matching entry in TRIAD was looked for. To match events, the date stored in TRIAD was compared to the date of the report where the event was found. For those events that could be retrieved, each of the items extracted by the pipeline (*System Items*) was matched to the corresponding data item in TRIAD. The performance was evaluated on those items extracted by the pipeline for which a TRIAD entry was found (*TRIAD Items*). For each of these items, a *correct annotation* corresponds to an exact match between the system and the TRIAD value.

The accuracy of the system was computed on single events as the ratio between correct annotations and TRIAD items.

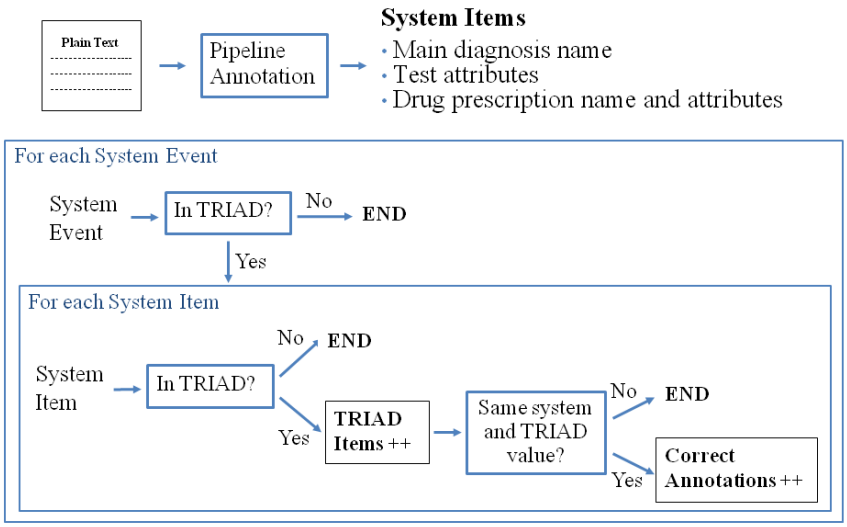

Figure 3.13: Item extraction evaluation for one report.

In Figure 3.14, an example of the performed evaluation is reported for one short report, translated to English for convenience. The figure shows the report itself (Input Text), the information extracted by the system (Extracted Information), and the matching entry in the TRIAD database (TRIAD Holter ECG matching rows). In this case, the report includes a date ("05/08/2012") and one event (an Holter ECG test) with four attributes (Rhythm, Average Heart Rate, ST Elevation, ST Elevation Type). The extracted date is used to retrieve the matching entry in TRIAD, through the Visit Date field. For each attribute related to the extracted event, the system values and the TRIAD values are compared. In this example, out of the four System Items, the number of correct annotations is three, leading to a final accuracy of 75%. In particular, due to the presence of the negated sentence "no significant elevation in the other right precordial leads", the system extracts an "ST Elevation Type" attribute with an "absent" value. However, the correct value for this attribute reported in TRIAD is "saddleback".

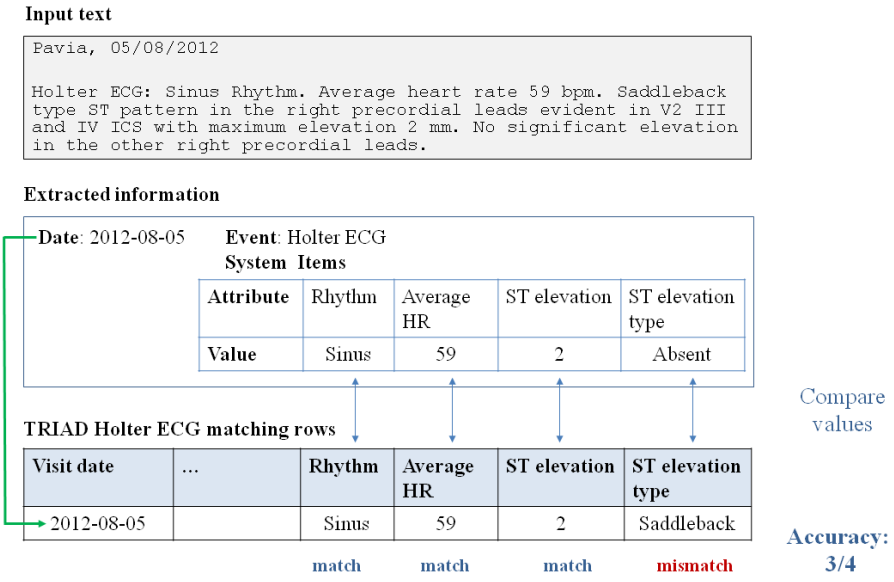

Figure 3.14: Example on accuracy computation.

3.8.3. Temporal expression extraction evaluation

The performance of temporal expression extraction was evaluated on the annotated test set. For HeidelTime, the extraction of TIMEX text spans was evaluated against the manual annotations described in Section 3.2.2 (gold TIMEXes), using precision, recall, and F1 score. The definition of TPs, FPs, and FNs are identical to those proposed for the event extraction task. Specifically, TPs are defined as gold TIMEXes whose boundaries are correctly identified by HeidelTime. FNs are represented by gold TIMEXes

that are not detected by the system. FPs, finally, are temporal expressions extracted by HeidelTime but not found in the gold annotations.

For temporal expressions marked as TPs, the accuracy of property extraction was computed as well. In particular, the value, the type, and the mod property were considered. For each property, the accuracy (acc) is given by the number of correctly extracted values over the total number of extracted TIMEXes (the TIMEX TPs):

$$acc = \frac{\#\ correct\ values}{\#\ TIMEX\ TPs}$$

As for the event extraction task, for analyzing the effects created by a potential class imbalance, the F1 score of the type property was computed, too.

3.8.4. Timeline reconstruction validation

As previously pointed out, there was no availability of a gold standard for evaluating the approach proposed for clinical timeline reconstruction. However, a preliminary evaluation was conducted to assess the system's ability to correctly aggregate data belonging to the same patient. In particular, the evaluation consisted in randomly selecting one patient, and manually reviewing his/her clinical timeline.

In a more comprehensive evaluation involving clinicians, a few criteria for success could be highlighted. First, following the approach proposed by Hirsch et al. [101], it would be important to assess whether it is possible to use the reconstructed timeline to effectively retrieve dates and relevant events. To evaluate this aspect, one possibility would be to measure the time needed to manually search for this information inside the documents, and compare it to the time required to explore the reconstructed timeline. As another evaluation criterion, it would be important to verify whether the information displayed on the timeline (e.g., event experiencer, attribute values) is sufficiently detailed to draw relevant conclusions. As a matter of fact, the approach proposed for timeline visualization uses a few heuristics to define how events are aggregated and displayed. In particular, the "day" granularity was chosen to refer each event to a specific point in time. Given that events in the CARDIO dataset are usually not related to a specific time of the day, this approximation is expected to be acceptable for most use cases. However, for certain clinical episodes, it might be important to preserve more detailed temporal information on the extracted events. To evaluate these potential issues, a specific questionnaire could be prepared to be filled in by clinicians.

Chapter 4

Results and discussion

This chapter illustrates the results obtained on the CARDIO dataset, discussing each of the pipeline steps. In particular, each section contains both the evaluation results and a discussion including a comparison to the related literature.

Section 4.1 presents the distribution of entities and their properties in the annotated subset of documents. Section 4.2 describes the results of the event extraction task, discussing both the dictionary lookup approach and the neural network classifier. Section 4.3 illustrates the performance of attribute extraction, describing the developed ontology and the obtained extraction results. Section 4.4 presents the results of temporal expression extraction and normalization, performing a comparison between the HeidelTime and the TimeNorm systems. Finally, Section 4.5 discusses the outcomes of temporal link extraction and clinical timeline reconstruction.

4.1. Statistics on the annotated corpus

To develop an annotated corpus to be used for methods development and validation, a subset of 75 documents were randomly selected from the CARDIO dataset. These documents were manually annotated with mentions of events and temporal expressions, according to the guidelines described in Section 3.2.

This section provides a summary of the performed annotations. In Table 4.1, the total number of manually annotated events and temporal expressions is reported, grouped into training set and test set. As it can be noticed from the table, the number of annotated events is about 4 times greater than the number of temporal expressions. To give a sense of the documents average size, Table 4.1 also reports the number of sentences and tokens detected in the text preprocessing phase.

Table 4.1: Statistics on the annotated corpus.

	Training set	Test set	Corpus
Documents	60	15	75
Tokens	44115	13148	57263
Sentences	3347	941	4288
Events	3159	992	4151
TIMEXes	814	288	1102

In Table 4.2, the distribution of property values for the 4151 annotated events is reported. The most frequent semantic type is the Problem type (42%), followed by Test (28%), Occurrence (17%), and Treatment (13%). As most events mentioned in the text are temporally referred to the visit date, the most common DocTimeRel value is OVERLAP (44%). About the other DocTimeRel values, 38% of events happen "before" the report date, while 17% of events occur "after". The BEFORE/OVERLAP value, finally, is specified for only 1% of events. This result reflects a precise annotation choice: the BEFORE/OVERLAP value is selected only when the text clearly indicates that an event started before the visit date and continues into and through this reference date [11]. As regards the polarity, modality and experiencer properties, it is possible to notice a strongly unbalanced distribution among classes. In particular, the majority of events are characterized by a positive polarity (88%), an actual modality (90%), and are experienced by the patient himself (92%).

Table 4.2: Distribution of Event property values in the annotated dataset.

Property	Class distribution	
Semantic type	1766 (42%)	Problem
	1155 (28%)	Test
	696 (17%)	Occurrence
	534 (13%)	Treatment
DocTimeRel	1828 (44%)	Overlap
	1574 (38%)	Before
	696 (17%)	After
	53 (1%)	Before/overlap
Polarity	3634 (88%)	Positive
	517 (12%)	Negative
Modality	3729 (90%)	Actual
	167 (4%)	Hypothetical
	139 (3%)	Hedged
	116 (3%)	Generic
Experiencer	3824 (92%)	Patient
	327 (8%)	Other

Table 4.3 shows the distribution of the type and the mod properties for the 1102 annotated temporal expressions. As shown in this table, most

TIMEXes are represented by dates (61%). Durations, sets and times account for the 19%, the 12% and the 8% of temporal expressions, respectively. As a final observation, the mod property is defined for only 6% of TIMEXes; for these expressions, the APPROX value denotes an "approximate" meaning.

Table 4.3: Distribution of TIMEX property values in the annotated dataset.

Property	Class distribution	
Type	672 (61%)	Date
	209 (19%)	Duration
	131 (12%)	Set
	90 (8%)	Time
Mod	1036 (94%)	No value
	66 (6%)	Approx

4.2. Event extraction results

This section presents the results obtained for the event extraction task, which consists in correctly identifying the text spans and the properties of the events mentioned in the texts.

4.2.1. Text span extraction

For the extraction of event spans, experiments were performed with a dictionary lookup approach and a GRU classifier. To quantify the improvements achieved by feeding POS tags to this neural network model, experiments were run also by removing these additional inputs, i.e., using word embeddings only.

To compare these GRU classifiers (with and without POS inputs) to reference supervised algorithms, both a CRF and an SVM running on embedding-related features were evaluated. Specifically, the baseline CRF classifies each token using as features its embedding, and the embeddings of the two previous tokens. For the SVM, the B, I, O labels of the previous tokens were considered, too.

Table 4.4 reports the results obtained on the test set. As the events in the annotated corpus are not necessarily included in the external dictionaries, the dictionary lookup approach did not perform well, resulting in an F1 score of 66.1% (row 1). The supervised CRF and SVM classifiers obtained better extraction results, mostly thanks to an improvement in recall (rows 2 and 3). Comparing these classifiers and the RNN models, it can be noticed that both the GRU models (rows 4 and 5) outperformed the CRF and the SVM, with an increase in both recall (up to 87.0%) and precision (up to 89.0%).

Table 4.4: Results for the extraction of event text spans (test set).

Row	Extraction method	TP	FP	FN	P	R	F1
1	Dictionary lookup	548	118	444	82.3%	55.2%	66.1%
2	CRF classifier	795	189	197	80.8%	80.1%	80.5%
3	SVM classifier	748	103	244	87.9%	75.4%	81.2%
4	GRU classifier	844	111	148	88.4%	85.1%	86.7%
5	GRU classifier with POS input	863	107	129	89.0%	87.0%	88.0%
6	Dictionary lookup + GRU classifier with POS input	895	114	97	88.7%	90.2%	89.5%

To investigate whether the combination of supervised and knowledge-based approaches could lead to an improved performance, it was decided to merge the outputs of the best performing GRU classifier and the dictionary lookup approach, and evaluate results on this new output (Table 4.4, row 6). Both methods identify events in the text, creating corresponding UIMA annotations according to the identified boundaries and semantic types. In the merging phase, the annotations found by the GRU classifier were added to the list of annotations already performed by the dictionary lookup approach. In case of two overlapping annotations with the same semantic type, only the longest one was kept. In Figure 4.1, an example of this kind of overlap is shown: as "*Sindrome del QT Lungo*" (Long QT Syndrome) and "*Difetto genetico responsabile di Sindrome del QT Lungo*" (Genetic defect responsible for Long QT Syndrome) are characterized by the same semantic type (i.e., Problem), only the longest event is annotated by the system.

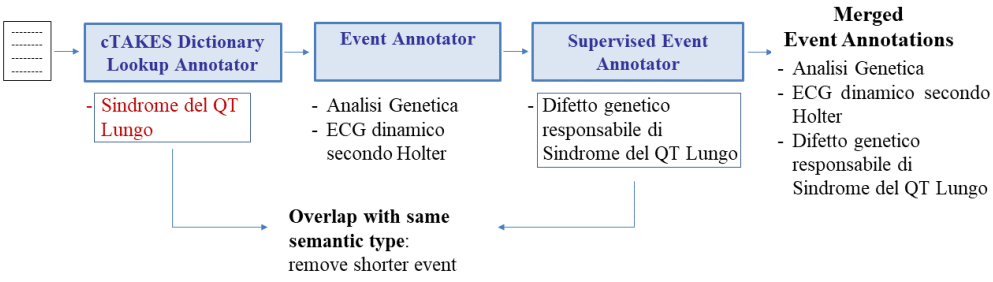

Figure 4.1: Integration of Event annotations (dictionary lookup and RNN classifier).

Comparing the results obtained in row 5 and row 6 of Table 4.4, it can be noticed that combining dictionary entries with the GRU classifier output obtained the best F1 score across all experiments (89.5%).

4.2.2. Property extraction

In the event extraction task, the dictionary lookup approach and the RNN classifier are used to extract event boundaries together with semantic types. For identifying the other event properties, the ConText rule-based algorithm (for polarity, modality, and experiencer) and an SVM (for DocTimeRel) were explored. To evaluate the performance, the best event extraction configuration obtained from the previous step was selected ("Dictionary lookup + GRU classifier with POS input"), and results were computed on the 895 events extracted with correct boundaries (TPs). For each of these events, the property values selected by the pipeline were compared to the corresponding gold values.

In Table 4.5 the accuracies of the different property extraction systems are shown ("Raw accuracy" column). For comparison purposes, the results obtained with simple majority classifiers (that assign to all properties the most frequent value in the training set) are included, too ("Raw majority accuracy" column). As expected, the developed system outperforms the majority classifier for all the properties, especially as regards the semantic type (98.5% versus 40.9%) and the DocTimeRel (83.4% versus 40.3%).

As shown in Table 4.2, the polarity, modality, and experiencer properties are characterized by a strongly unbalanced distribution among classes. To avoid overestimating the extraction performance for these properties, a weighted accuracy was computed too for both the developed pipeline step and the majority classifier ("Weighted accuracy" and "Weighted majority accuracy" columns). This measure uses different multipliers to weight the number of correct classifications for the different property values, thus increasing the impact given by less frequent values on the final performance indicator. For a property with two possible values A and B, where B is the less frequent value, the weighted accuracy (acc_W) is computed as follows (considering a weight $W > 1$):

$$acc_W = \frac{\#correctA + W \cdot \#correctB}{\#trueA + W \cdot \#trueB}$$

To compute weighted accuracies, weights were selected according to the proportion between classes. For example, as events with positive polarity are 7 times more frequent then events with negative polarity, a weight of 7 was chosen for the polarity attribute with the negative value. On the basis on this metric, it is possible to notice that the accuracies of the methods developed for the polarity, modality and experiencer properties are much greater than those obtained by the dual majority classifiers.

Table 4.5: Event property extraction accuracies (test set).

Attribute	Raw accuracy	Raw majority accuracy	Weighted accuracy	Weighted majority accuracy
Semantic Type	98.5%	40.9%	NA	NA
DocTimeRel	83.4%	40.3%	NA	NA
Polarity	96.9%	89.8%	90.2%	61.1%
Modality	93.3%	88.9%	61.5%	22.9%
Experiencer	95.3%	93.1%	72.2%	52.8%

As mentioned, the results shown in Table 4.5 were computed on those events that were correctly identified by the system, thus depending on the performance of event recognition. To verify the performance of property extraction alone and assess the variability of results, it was decided to evaluate this task on gold events, too (992). In this case, given that event semantic types are extracted together with the events themselves, extraction accuracies were computed for the remaining four properties: DocTimeRel, polarity, modality, and experiencer. As shown in Table 4.6, results were comparable to the ones computed on event TPs.

Table 4.6: Event property extraction accuracies (gold events).

Attribute	Raw accuracy	Raw majority accuracy	Weighted accuracy	Weighted majority accuracy
DocTimeRel	82.9%	43.2%	NA	NA
Polarity	97.0%	89.2%	90.8%	54.2%
Modality	93.0%	89.1%	60.6%	23.3%
Experiencer	95.5%	93.1%	72.8%	53.1%

To further analyze the effects of class imbalance, the F1 score was computed for all different property values, including those of the semantic type and the DocTimeRel properties. For properties with more than two values (e.g., the semantic type), the F1 score of a specific value (e.g., “problem”) was computed by considering that value as positive and all the others as negative.

In Table 4.7, the F1 scores for each property value are shown. As the property extraction performance was similar on both event TPs and on gold events, only the results on event TPs are reported. For all semantic type values, a great performance was achieved (F1 scores above 97%). The DocTimeRel classifier resulted in overall good results, with the only exception of the BEFORE/OVERLAP value (F1 score of 0%). With respect

to the polarity property, high F1 scores can be noticed for both the "positive" value (F1 score of 98.2%) and the "negative" value (F1 score of 85.6%). For the last two properties (modality and experiencer), slightly lower F1 scores were obtained for those values that were less represented in the dataset.

Table 4.7: F1 scores for each event property value (test set).

Property	Raw accuracy	F1 score for each value	
Semantic Type	98.5%	Problem	99.3%
		Test	98.5%
		Treatment	98.0%
		Occurrence	97.2%
DocTimeRel	83.4%	Overlap	85.3%
		Before	83.4%
		After	80.3%
		Before/Overlap	0%
Polarity	96.9%	Positive	98.2%
		Negative	85.6%
Modality	93.3%	Actual	96.7%
		Hypothetical	46.6%
		Hedged	66.7%
		Generic	77.8%
Experiencer	95.3%	Patient	97.5%
		Other	55.3%

4.2.3. Discussion

In this thesis, a supervised approach based on RNN architectures was explored to extract clinical events from medical reports written in Italian.

Event extraction results. To assess the system performance on event identification, the proposed RNN-based model was compared to an unsupervised strategy based on dictionary lookup, to a CRF classifier, and to an SVM classifier, using an independent test set. Although dictionary lookup can be an effective method to identify standardized concepts included in external dictionaries, it fails in extracting the non-standard, domain specific terms frequently used in the CARDIO dataset. As shown in Table 4.4, this approach misses a high number of relevant events, resulting in low system recall (55.2%) and an overall F1 score of 66.1%. These results are comparable to the ones obtained by Chiaramello et al., who used the MetaMap tool to extract UMLS medical concepts from Italian clinical notes [82]. In their case, the MetaMap annotation showed recall, precision and F1 score equal to 53%, 98% and 69%, respectively. As the main reason for annotation failures, the authors identified the impossibility of generating concepts variants for the Italian language. As regards the presented CRF and SVM baseline models, which rely on embedding-

related features, it is possible to notice a considerable increase in recall with respect to dictionary lookup (from 55.2% to 80.1% and 75.4%, respectively). For the SVM classifier, an increase in precision can be noticed, too (from 82.3% to 87.9%). In this case, FNs decreased from 444 to 244, and FPs decreased from 118 to 103. Compared to the SVM, the proposed RNN-based classifiers resulted in an even better performance, allowing a further improvement in recall up to 87.0%, with a decrease in FNs to 129. The RNN model using POS tags obtained the best result across all supervised classifiers, with an F1 score of 88.0%. A possible explanation for this result is that event mentions correspond to syntactically meaningful n-grams. Providing easily-obtainable syntactic information, therefore, could play a role in the correct detection of event boundaries.

To investigate whether it was possible to improve the performance by exploiting the availability of external dictionaries, the developed classifier was integrated into a dictionary-based NLP pipeline for clinical information extraction. Although the considered dictionaries only partially include the relevant terms found in reports, merging the outputs of the two proposed approaches allowed a slight improvement in results (F1 score up to 89.5%), with an increase in recall, and a slight drop in precision. After an error analysis, it was noticed that, thanks to the dictionary lookup approach, the integrated system was able to find very specific and infrequent medical concepts, such as "*Hashimoto's Thyroiditis*" and the "*Tiklid*" drug, that could not be extracted by the RNN classifier (neither these terms nor similar ones were ever seen in the training set). On the other hand, the slight increase in false positives was due to the extraction of dictionary entries that overlapped only partially with the gold text spans.

Property extraction results. Besides extracting event spans and semantic types, ad-hoc methods were developed to extract other event properties, such as the polarity and the modality. Correctly extracting these details is a crucial task, as events that are mentioned with a negative or a hypothetical meaning should not be represented as part of a patient's history.

Overall, the ConText system achieved good extraction accuracies, indicating that this kind of rule-based approaches represents an effective method to extract details on the polarity, modality, and experiencer of event mentions. As shown in Table 4.5, the extraction of these properties resulted in weighted accuracies that were considerably greater than those achieved by the majority classifiers. As an interesting observation, slightly lower F1 scores were obtained for those values that were less represented in the dataset (Table 4.7). For example, the classification of the experiencer property resulted in an F1 score of 97.5% for the "patient" value, and 55.3% for the "other" value. As a matter of fact, this second value is found in only 8% of all annotated events.

As regards the SVM-based extraction of the DocTimeRel property, a good overall performance was obtained as well, with F1 scores above 80% for the OVERLAP, BEFORE, and AFTER values. The only exception was represented by the BEFORE/OVERLAP value, for which the computed F1

score was 0% (i.e., no true positives). In this case, the limited number of examples available in the training set did not allow a correct classification on the test set.

Comparison to related work. Although the challenges related to clinical information extraction depend on the task complexity (e.g., corpus heterogeneity and size, event definition) it is possible to compare the obtained results to previous work on clinical texts in Italian. The system proposed by Esuli et al., based on a two-stage CRF method, resulted in a F1 score of 85.9% for a single-annotator experiment [84]. Comparing the approach proposed in this work to the paper by Esuli et al., two main differences can be found in the information extraction task definition. First, in this paper the authors aim to extract longer segments compared to the ones in the CARDIO dataset (their average segment length is 17.33 words), which might explain the lower results they obtained. On the other hand, their approach is evaluated at the token level (each token counts as a TP, FP, TN, or FN for a given tag), which credits the system also for partial success. As regards the system presented by Attardi et al., which relies on a statistical sequence labeller, the authors obtained higher F1 scores (e.g., 98.26% for the annotation of body-parts and treatments) with respect to the ones shown in Table 4.4 [86]. In this case, though, the reference annotations were mostly generated automatically, which is likely to have biased the creation of the corpus used for evaluation. As another difference related to the task definition, the authors developed different classifiers to account for different semantic types (e.g., body-parts and treatments, and other mentions were classified with two different models). Targeting the classification model to each semantic type may have increased the performance by reducing the variability of the features. Finally, in the work by Gerevini et al., the authors propose a supervised method for automatic report classification, which represents a different task with respect to event extraction [87]. While their work is focused on labelling each whole report according to five levels of classification (exam type, test result, lesion neoplastic nature, lesion site, and lesion type), one of the aims of this research activity is to identify events and their properties inside the text.

LSTM models have been investigated for a long time in the area of general domain NER, especially for English texts [54,55]. As regards other languages, Lample et al. used neural network architectures based on LSTMs to identify named entities in Spanish, German, and Dutch [55]. More specifically, the authors proposed two models: (i) a bidirectional LSTM with a sequential conditional random layer above it, and (ii) a model that builds and labels chunks using an algorithm inspired by transition-based parsing (with states represented by stack LSTMs). The proposed models obtained a state of the art performance for all the analyzed languages. Specifically, while the LSTM-CRF model achieved an F1 score of 90.94% on the English dataset, using the same approach on the Spanish dataset (the most similar language to Italian) led to an F1 score of 85.75%, which is in line with the results shown in Table 4.4. As another example,

Athavale et al. used bidirectional LSTM architectures to perform NER in Hindi, reaching a F1 score of 77.48% on the test set [125]. These results, though obtained on general domain NER tasks, show the great potential for extension of LSTM approaches to other domains and languages. With respect to the approach proposed in this dissertation, the application of LSTM models to Italian clinical text has shown a good performance, which is in line with the recent advances found in the literature.

Compared to the English language, Italian presents some specific challenges, being more morphologically complex and allowing a flexible word order, for example in the ordering of nouns and adjectives. As far as it is known, only one work has applied recurrent architectures to extract named entities from Italian news [57]. With respect to general domain text, clinical notes are characterized by an additional layer of complexity, as sentences do not always respect syntactic rules, and domain-specific jargon (abbreviations, acronyms) is commonly used. Despite the overall complexity of the considered task, the RNN-based approach that was proposed in this research activity performs well. In particular, the use of LSTM architectures allowed improving the system recall with respect to other approaches. Limiting false negatives is a challenge, as relevant events are usually mentioned in the reports by using terms that are often institution-specific (or even expert-specific), and are thus not likely to be found in standardized terminologies. For this reason, it is particularly important to develop an effective solution for the extraction of non-standard events, too. As an interesting observation, the performed experiments indicate that integrating a well-performing supervised approach with a dictionary lookup strategy can represent a good choice to further improve the extraction performance.

Limitations. The supervised methods proposed for event extraction suffer from some limitations.

First, although a considerable effort was put into the corpus annotation, the size of the annotated dataset is small in comparison to other works (e.g., [40]). Also, although the annotation guidelines were developed through a shared effort, the dataset used in the performed experiments was annotated by a single person. Based on this work, the task seems to be learnable with high accuracy even with small datasets. However, since the scope was limited, questions remain about how general this conclusion is. In the future, two additional annotators will be involved, and more documents will be annotated with the information of interest. At the moment, a second annotator is working on the same subset of documents considered in this research activity, following the developed annotation guidelines. Analyzing the inter-annotator agreement will allow assessing the reliability of the annotations and the complexity of the problem.

Another limitation related to the annotation process concerns the definition of the event tags: in this research activity, it was decided to consider clinical events mostly expressed through noun phrases, without taking into account verbs representing actions (e.g., “the patient

complained...”). In future work, it will be interesting to extend annotations to verbs denoting events, too.

Finally, with respect to the developed RNN classifier, the proposed LSTM architecture was not combined to other models, such as CRFs. Although this choice was made to reach a trade-off between the complexity of the classifier and the still limited dimension of the annotated corpus, it would be interesting to explore additional configurations.

4.3. Attribute extraction results

This section presents the evaluation conducted for the attribute extraction task, which involves: (i) the development of the domain-specific ontology, and (ii) the use of this ontology to identify the names and the values of each event attribute.

4.3.1. CARDIO ontology development

To define the events and the attributes to be extracted from the CARDIO dataset, a domain ontology was manually developed (*CARDIO ontology*). The ontology was designed and refined on a development set including 4429 reports. The main steps that were followed are shown in Figure 4.2.

1. As a first step, a set of 20 reports were randomly selected from the development set and analyzed to identify the information of interest (“set for ontology design”). To define Event and Attribute classes in the ontology, both the information written in the text and the data stored in the Maugeri hospital’s system (TRIAD) were considered. For example, it was noticed that most reports include sections that describe specific diagnostic tests, with related results. As these results are also reported in TRIAD, they were considered as relevant items for the extraction. Starting from this observation, the specific attributes to be extracted were discussed with clinicians, who provided the clinical knowledge needed to deal with ambiguous terms and acronyms. After this manual analysis, each of the identified tests was modelled as an ontology Event (e.g., “ECG Test”), and all the related results were captured in suitable ontology Attributes (e.g., “Average Heart Rate”). To define which attribute types should be considered (numeric or categorical), both domain knowledge and the TRIAD structure were exploited. As a result of these analyses, a first version of the Attribute Annotator was created (*system version 1*).
2. To evaluate the extraction performance, *system version 1* was ran on the whole development set. The ontology and the annotator were then iteratively improved according to the results of an error analysis (Figure 4.2, step 2). In this phase, two major

improvements were performed on the ontology: many regular expressions were enhanced, and specific modifiers were created for a few ambiguous attributes. As regards the Attribute Annotator, the identification of attribute names was considerably improved by changing the definition of event-specific lookup windows. Thanks to the performed changes, a *system version 2* was obtained.

3. For the final evaluation, *system version 2* was run on an independent test set (1003 reports).

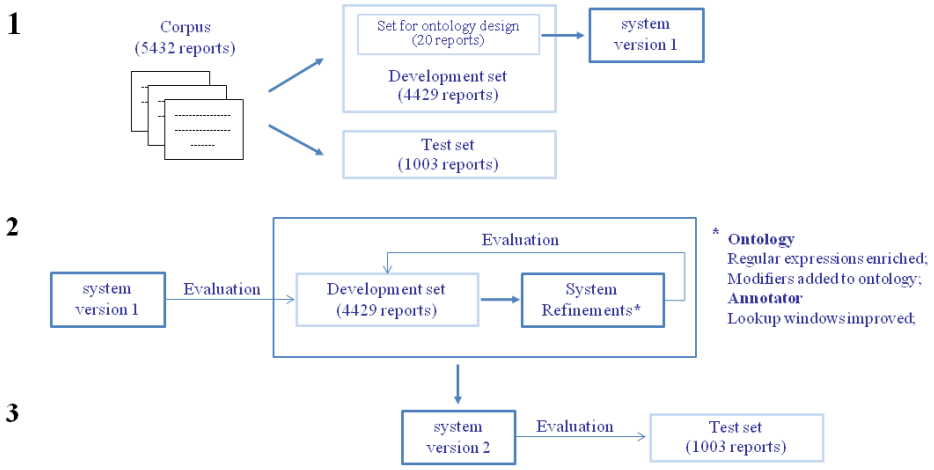

Figure 4.2: CARDIO Ontology development and refinement.

The developed ontology contains 11 events and 61 attributes: 44 attributes are numeric, the others are categorical. Figure 4.3 shows the class structure as it was defined in the Protégé framework. Both events and attributes are grouped into three main classes: Problem, Test, and Treatment (events with type Occurrence were not included as they are not currently related to any attribute).

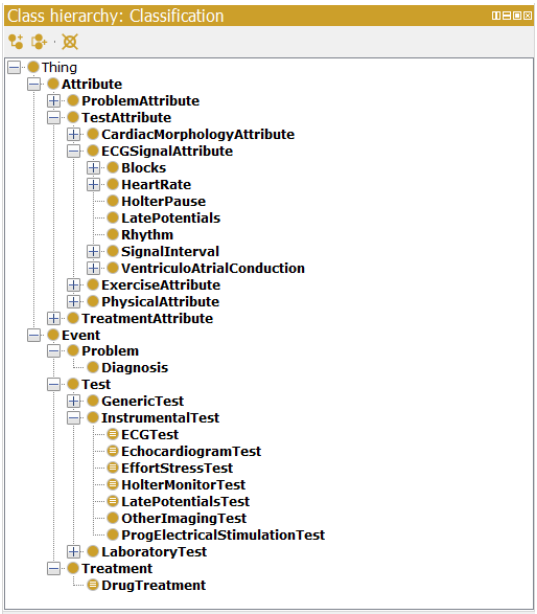

Figure 4.3: Domain ontology for the CARDIO dataset.

In the developed ontology, the main diagnosis of the patient represents one event (*Diagnosis* class) identified by its name, and currently has no attributes. Drug prescriptions represent another event (*DrugTreatment* class), identified by a name, with three attributes: dose, frequency and format. The other events are diagnostic procedures (e.g., *ECGTest* class, *EchocardiogramTest* class), each with several attributes. To provide an example, Figure 4.4 shows how the ECG test class is related to several attributes through the "hasAttribute" relation.

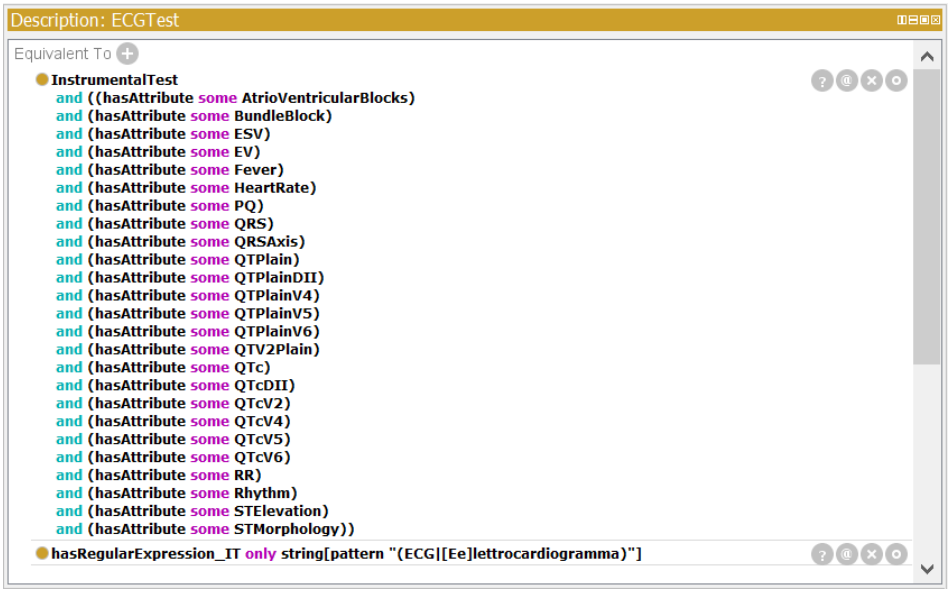

Figure 4.4: ECGTest class description.

4.3.2. Validation against TRIAD

As explained in Section 3.8.2, the performance of attribute extraction was evaluated against TRIAD, the hospital system recording data about patient diagnoses, tests, and treatments. In particular, the evaluation was conducted on the five events most frequently stored in TRIAD: main diagnosis, prescribed drugs, and three diagnostic tests (ECG, Holter ECG, and Effort stress test). For prescribed drugs, two evaluations were performed: (i) an evaluation focused on only drug names, and (ii) an evaluation considering drug names with associated dosages. Given that drug format and prescription frequency are not included in TRIAD, these two attributes were not evaluated.

In the process of developing the ontology and the Attribute Annotator, two system versions were created: *system version 1* and *system version 2* (Figure 4.2). In the evaluation phase, *system version 1* was run on the development set (Table 4.8). The final *system version 2* was run both on the development and the test sets. Results on the test set are shown in Table 4.8; it is important to mention that a similar performance was obtained on the development set (data not shown). In the table, column "a" represents the number of items extracted by the system; column "b" represents the number of extracted items for which an entry was detected in TRIAD; column "c" corresponds to the number of correct annotations; column "d" represents the *accuracy*, computed as c/b. Cells marked with * are related to the results for drug names and dosages.

Table 4.8: Evaluation of *system version 1* on the development set, and of *system version 2* on the test set. SV: System Version.

SV	Set	Event Name	System Items (a)	TRIAD Items (b)	Correct Annot. (c)	Accuracy (d)
1	**Dev (4429 docs)**	Main Diagnosis	4202	4077	3607	**88.5%**
		ECG	26669	22546	21352	**94.7%**
		Holter ECG	26767	21538	19058	**88.5%**
		Effort Stress Test	9683	3978	2367	**59.5%**
		Prescribed Drug	8720 (8270*)	2436 (4584*)	2186 (2860*)	**89.7% (62.4%*)**
2	**Test (1003 docs)**	Main Diagnosis	927	913	845	**92.6%**
		ECG	7452	5070	4885	**96.4%**
		Holter ECG	7173	5127	4757	**92.8%**
		Effort Stress Test	2543	1118	1064	**95.2%**
		Prescribed Drug	1999 (1999*)	538 (930*)	435 (672*)	**80.9% (72.3%*)**

In the evaluation of *system version 1*, an accuracy of 88.5% was obtained for diagnoses. The error analysis showed that most errors were due to problems in matching diagnosis names in reports with the corresponding entries names in TRIAD. As regards tests, the system achieved the best accuracy for ECGs (94.7%), followed by Holter tests (88.5%), which are described with more complex sentences. On the other hand, a poor performance was obtained for Effort Stress tests (59.5%). In this case, the main issue was the misclassification of attributes that are written in the same way (e.g., "QT" for QT interval), but are related to different test phases (e.g., baseline, stress, recovery QT length). Regarding drug prescriptions, drug names identification led to good results (89.7%). However, extracting drug dosages was not trivial (62.4%) because, while the TRIAD database contains only daily doses, reports often include sentences with both unit dosages and frequencies of assumption.

In the evaluation of *system version 2* on the test set, higher accuracies were achieved for almost all events. For diagnoses, the knowledge provided by physicians was exploited to refine the mapping of the terms used in TRIAD to those used in the reports. The most significant improvement concerned effort stress tests, with an increase in accuracy from 59.5% to 95.2%. The only decrease in performance was given by drug names (from 89.7% to 80.9%). Many of the non-matching drug names in this case were due to the insertion of erroneous data in TRIAD: two very similar drugs ("Metoprolol" and "Metoprolol Retard") were frequently stored with the same name.

4.3.3. Discussion

In this research activity, besides extracting clinical events, the proposed IE approach allows capturing attributes of interest and linking them to the events they are related to. To define events and related attributes, a domain-specific ontology was developed.

Ontologies for information extraction. Formalizing information through ontologies brings several advantages. First, thanks to the flexible Event-Attribute structure, it is possible to relate the same attribute to different events, reusing shared concepts multiple times. Second, the inclusion of regular expressions in the ontology makes the proposed approach easily language extensible. To analyze reports in another language, it would be in principle sufficient to translate regular expressions. This multilingual extension will be further discussed in Section 5.1. Finally, although ontologies are built for a specific domain, they allow easy updates and extensions to account for new information. The proposed approach, for example, could be applied to the analysis of reports coming from other clinical domains. To adapt the system, only the ontology (and possibly the external dictionaries) should be updated. A detailed discussion on this

aspect will be provided in Section 5.2, which presents an extension of the developed ontology to process anatomic pathology reports for breast cancer patients.

The possibility to adapt an IE system to a new language or domain is an important feature, as it allows reusing the same framework to process different kinds of documents. As a main advantage of the approach proposed in this thesis, it is possible to tune the underlying ontology without the availability of annotated data. On the other hand, efforts have to be put into manually defining the entities to be extracted, as well as their regular expressions. Conversely, using a machine learning approach, adapting the extraction system would only require re-training the underlying classification model. However, this step could not be performed without the availability of a large and reliably-annotated corpus. As a matter of fact, manually developing such a corpus would probably require greater efforts with respect to those needed for ontology adaptation.

Comparison to related work. Although a few works have used ontologies for clinical NLP, it is possible to highlight a few differences between the approach used in this research activity and previous efforts on this area. Spasić et al. developed an ontology-driven system, KneeTex, that extracts information from English knee MRI reports [29]. KneeTex is focused on the extraction of findings and anatomical regions, with possible classifiers defined in the ontology. Although the resulting system performs very well for the considered task, the developed ontology is strongly domain-specific. In the CARDIO ontology, instead, the definition of events and attributes is more general, thus facilitating the extension to other clinical domains. Mykowiecka et al. proposed an ontology-driven system to analyze mammography reports in Polish [30]. The proposed system works well on the analyzed clinical domain. However, a considerable manual effort was put into complex rules engineering. As a main difference, the CARDIO ontology is automatically translated into an XML file, which is then given as an input to the NLP pipeline. Moreover, the approach used to extract and relate information has a smaller dependency on syntax. Toepfer et al. used an ontology to extract objects (with attribute and values) from German transthoracic echocardiography reports [31]. The developed IE system performs well. The proposed ontology structure is similar to the CARDIO one, especially as regards the definition of attributes and values. As one main difference, the IE task proposed in this research activity does not consider objects, but events characterized by a semantic type. Moreover, events are related to attributes that can be either numeric or categorical, rather than only textual variants. As a final consideration, while Toepfer et al. defined the ontology in a semi-automatic way, the CARDIO ontology was developed by manually analyzing reports. In the future, it would be interesting to explore the possibility of automatically developing the ontology from free-text [126,127]. To this end, concepts automatically extracted from UMLS could be exploited.

As a distinctive feature of this research activity, it was possible to exploit a structured database, i.e., TRIAD, to guide the definition of the IE task and evaluate the extraction performance. The availability of structured data that is correlated to the text content is an interesting situation. With respect to the methodology proposed in this thesis, for example, it could be possible to automatically define the IE ontology starting from TRIAD, thus reducing the manual effort needed to extract relevant information from text.

Limitations. The approach proposed for the extraction of event attributes has some limitations. First, given the unavailability of annotated data, it was not possible to evaluate the performance on a gold standard dataset. Given the large size of the CARDIO dataset, manually annotating all the documents would be hard and time consuming. To overcome this issue, system annotations were validated against the data included in TRIAD. However, only the item values that are both extracted from reports and found in TRIAD could be considered, thus focusing on the accuracy of extracted items. To evaluate false negatives, data that are available in TRIAD, but not extracted from reports, should be considered. Since additional sources could have been used to fill in the database, some attributes that are available in TRIAD may not be present inside the documents; for this reason, computing FNs (and therefore the system's recall) is not possible. To evaluate false positives, on the other hand, data extracted from reports, but not available in TRIAD, should be considered. Given that data entered in TRIAD are not guaranteed to be complete, it is hard to evaluate false positives as well. As a final remark, even when the same items are present in both reports and TRIAD, there could be human errors in data entry.

Another limitation of the proposed approach concerns the structure of the considered reports. In the Italian clinical setting, the structure of clinical free text reports is in general defined by the specific center that issues the report to the patient. For this reason, completely unstructured documents are as frequent as more structured texts organized in sections and paragraphs. In the majority of the documents considered in this work the content is organized in a clear way, and it is possible to identify specific sections, some of which actually correspond to clinical events (e.g., "ECG test", "Holter ECG test"). This feature may have affected the results that were obtained, which may not extend equally well to other clinical corpora.

As a final limitation, regular expression matching was used to look for attribute names and values in the text. However, the possible variants of concepts were not considered (except those already included in the ontology). Also, misspelled forms were not dealt with. To further improve the developed system, these aspects will be considered in a future version.

4.4. Temporal expression extraction results

This section presents the results obtained for the TIMEX extraction task, which consists in correctly identifying the text spans and the properties of the temporal expressions mentioned in the texts.

4.4.1. Text span extraction

For extracting the text spans denoting temporal expressions, the Italian version of HeidelTime was used. In particular, the system was adapted to the molecular cardiology domain by manually looking at a subset of reports included in the annotated training set.

Table 4.9 reports the results obtained by running the original system on the training set, and the updated system on both the training and the test sets. As shown in the first two rows of the table, tuning the system's resources allowed increasing the F1 score from 59.2% to 93.8% on the training set. On the other hand, using the updated system on the test set resulted in an F1 score of 95.1%.

Table 4.9: Results for the extraction of TIMEX text spans.

System	Set	TP	FP	FN	F1
HeidelTime original	Training	425	196	389	**59.2%**
HeidelTime updated	Training	760	47	54	**93.8%**
HeidelTime updated	Test	273	13	15	**95.1%**

4.4.2. Property extraction

To identify the properties of each TIMEX (type, value, and mod), both HeidelTime and TimeNorm were adapted to the clinical context by analyzing a subset of reports in the annotated training set. As inputs to TimeNorm, the TIMEXes extracted by the "updated" version of HeidelTime were considered.

Table 4.10 shows the accuracies obtained for the property extraction task. In particular, the results marked in bold represent the final performance of the two updated systems on the test set.

Starting with HeidelTime, running the original system on the training set obtained initial high accuracies (above 91%) for the three considered properties. Tuning the system's rules on the training set allowed greatly improving the value extraction accuracy (from 91.5% to 97.8%). However, this same accuracy was slightly lower on the test set (93.8%). With respect

to the other properties, the highest improvement in accuracy was obtained for the type property (accuracy of 99.3% on the test set).

Moving to the TimeNorm performance, running the original system on the training set obtained worse results with respect to those of HeidelTime (accuracies between 57% and 71%). Thanks to the system's adaptation, it was possible to greatly improve the performance over the training set, with the highest improvement for the value property (from 56.7% to 94.5%). Using this updated system on the test set resulted in good results, too (accuracies around 90% for the three properties).

Table 4.10: Results for the extraction of TIMEX type, value, and mod properties.

System	Set	TP	Property	Correct	Accuracy
HeidelTime original	Training	425	type	404	95.1%
			value	389	91.5%
			mod	423	99.5%
HeidelTime updated	Training	760	type	717	94.3%
			value	743	97.8%
			mod	757	99.6%
HeidelTime updated	**Test**	**273**	**type**	**271**	**99.3%**
			value	**256**	**93.8%**
			mod	**271**	**99.3%**
TimeNorm original	Training	760	type	521	68.6%
			value	431	56.7%
			mod	540	71.1%
TimeNorm updated	Training	760	type	717	94.3%
			value	718	94.5%
			mod	686	90.3%
TimeNorm updated	**Test**	**273**	**type**	**247**	**90.5%**
			value	**243**	**89.0%**
			mod	**243**	**89.0%**

To take into account the class imbalance problem for the type property, Table 4.11 reports the F1 scores for each type value on the test set. From this table, it is possible to notice that overall high F1 scores were obtained for most TIMEX types (F1 scores between 89% and 99%). As a matter of fact, the only low result was obtained by TimeNorm on TIMEXes with type Time (F1 score of 58.6%).

Table 4.11: F1 scores for each TIMEX type value (test set).

System	Raw accuracy	F1 score for each value	
HeidelTime updated	99.3%	Date Duration Set Time	99.1% 88.7% 97.0% 95.0%
TimeNorm updated	90.5%	Date Duration Set Time	93.9% 89.8% 97.0% 58.6%

4.4.3. Discussion

In this thesis, two existing rule-based systems for the extraction of temporal expressions were applied to medical reports written in Italian.

TIMEX extraction results. The extraction of TIMEX text spans was performed by adapting the HeidelTime system to the molecular cardiology domain. As shown in Table 4.9, this adaptation allowed a great improvement of the F1 score over the training set (from 59.2% to 93.8%). As an interesting observation, the same updated system obtained a higher F1 score on the test set (95.1%), which is probably due to the small variability of the temporal expressions included in these reports.

As regards the extraction of TIMEX properties, it was decided to compare the performance of two systems, HeidelTime and TimeNorm, which rely on different rule-based approaches. From Table 4.10 it is possible to notice that HeidelTime performed better than TimeNorm for the extraction of all TIMEX properties. Starting from the value property, the adaptation of HeidelTime resulted in an increase in accuracy from 91.5% to 97.8% on the training set, with a final accuracy of 93.8% on the test set. Conversely, the adaptation of TimeNorm allowed raising the accuracy from 56.7% to 94.5% on the training set, with a final accuracy of 89.0% on the test set. Moving to the type property, while applying HeidelTime on the test set allowed obtaining an accuracy of 99.3%, using TimeNorm on the same documents resulted in an accuracy of 90.5%. As shown in Table 4.11, this lower result was mostly due to the difficulty in correctly identifying the temporal expressions with type Time (F1 score of 58.6%). With respect to the mod property, finally, HeidelTime achieved an accuracy of 99.3% on the test set, whereas TimeNorm obtained an accuracy of 89.0%.

As an important remark, the lower results obtained by TimeNorm could be probably explained by considering that this system does not access the TIMEX context. As a matter of fact, to normalize specific temporal expressions, knowledge about their surrounding context would be needed. For example, when dealing with times for drug prescriptions (e.g., “at 8, 20”), accessing the preceding tokens would be crucial to obtain a correct value normalization.

Comparison to related work Despite the unavailability of papers on temporal IE from clinical narratives in the Italian language, it is interesting to compare the results obtained in this research activity to other works using the HeidelTime or the TimeNorm systems.

In the EVENTI task for temporal IE from newspaper texts [90], both systems were used and adapted to analyze the Italian language [92,93]. Manfredi et al. worked on tuning HeidelTime resources to process the documents included in the EVENTI dataset [92]. As stated by the authors, most efforts were put into extending the existing Italian resources by carefully applying the guidelines provided by the task organizers. In particular, while modifying existing patterns to improve normalization was regarded as a rather simple task, a considerable work was needed to improve the performance in the extraction phase. By running the updated system on the test set, the authors obtained F1 scores of 82.1%, 70.9%, and 79.2% for the extraction of text spans, TIMEX values, and TIMEX types, respectively. These results are lower than the ones obtained in this research activity, which might be explained by the greater complexity of the EVENTI tasks definition. As another interesting work, Mirza and Minard adapted the TimeNorm system for normalizing the temporal expressions included in the EVENTI dataset [93]. The main adaptations concerned the translation and the modification of the existing English grammar and a few modifications on the TimeNorm code. Also, a preprocessing step was performed to deal with temporal expressions formed by only one or two digits (either a unit or the name of a month are added, based on the TIMEX context). By performing experiments on the test set, the accuracy in determining the TIMEX value was 66.5%, while the accuracy in determining the TIMEX type was 80.0%. Also in this case, the obtained performance is lower than the one presented in this research activity, which is probably due to the difference in complexity between the two considered tasks.

As an interesting work exploring HeidelTime on non-English clinical narratives, Hamon and Grabal proposed the tuning of this system for identifying temporal expressions in clinical texts written in English and in French [128]. For the English use case, the authors exploited the corpus used in the 2012 i2b2 Challenge [41]. The French corpus, instead, was developed for the purpose of the specific study. In this corpus, 182 documents were used as the training set, and 120 documents were used as the test set. As stated by the authors, the most important adaptations involved the enrichment and the encoding of linguistic expressions specific to the clinical domain, such as “b.i.d.” (i.e., twice a day). This finding is similar to the one that was reached by adapting the HeidelTime system on the CARDIO dataset. As regards the performance on text span extraction, the evaluation conducted on the English dataset resulted in an F1 score of 85%, while the experiments performed on the French dataset obtained an F1 score of 94.3%. As a matter of fact, these results are in line with the F1 extraction scores shown in Table 4.9.

Limitations. The approach proposed for temporal IE presents one main limitation, that is the small size of the annotated dataset. In particular, the good results that were obtained might be influenced by the limited variability of the temporal expressions found in the texts, i.e. the annotated TIMEXes are frequently expressed in very similar ways. Despite this limitation, the conducted experiments show that it is possible to port rule-based systems for temporal processing from the general to the clinical domain. As a future improvement, an extension of the annotated corpus will be performed.

4.5. Reconstructed patient timelines

This section presents the results obtained for the timeline reconstruction task, which requires to consistently aggregate all the events referring to one patient on a single temporal line.

The CARDIO dataset includes 5432 documents belonging to 1786 different patients. To highlight the importance of aggregating different reports referring to the same individual, Table 4.12 reports the distribution of the number of documents that are available for each patient. For 649 patients (36.3%), the corpus contains only one report. For all the other patients, at least two reports could be retrieved.

Table 4.12: number of reports per patient (CARDIO dataset).

# patient reports	# patients	% patients
1	649	36.3 %
2	333	18.6 %
3	244	13.7 %
4	185	10.4 %
5	153	8.6 %
[6-10]	187	10.5 %
[11-15]	27	1.5 %
[16-29]	8	0.4 %

4.5.1. Patient timeline validation

To assess whether the complete system was able to correctly aggregate data belonging to the same patient, a preliminary evaluation was performed by

randomly selecting one patient, and manually reviewing his/her clinical timeline. In particular, two versions of the timeline were reviewed. In the first version, only the events that "overlap" the document creation time were considered (*simple timeline*). In the second version, events that happen "before" or "after" the document creation time were included, too (*complete timeline*). In this case, the temporal links computed through a rule-based approach were exploited (Section 3.7.1).

For the selected patient, the corpus contains 10 documents. In the simple timeline, 83 distinct events with an OVERLAP relation to the document creation time were identified. All these events are associated to the corresponding visit date, and events mentioned multiple times are consistently aggregated. In the complete timeline, 123 distinct events were extracted and visualized. Of these, 40 events were associated to a specific date thanks to the computed temporal links.

Figure 4.5 shows one portion of the reconstructed timeline, together with the additional information that is visualized when one event is selected. For example, for the ECG test performed in May 2011 (Figure 4.5-a), the system displays nine extracted attributes (heart rate, PQ length, etc.), with their associated values. For the clinical problem "sudden death" (*morte improvvisa*, Figure 4.5-b), the displayed information regards the section ("Family history"), the polarity ("affirmed") and the experiencer ("family") of the event.

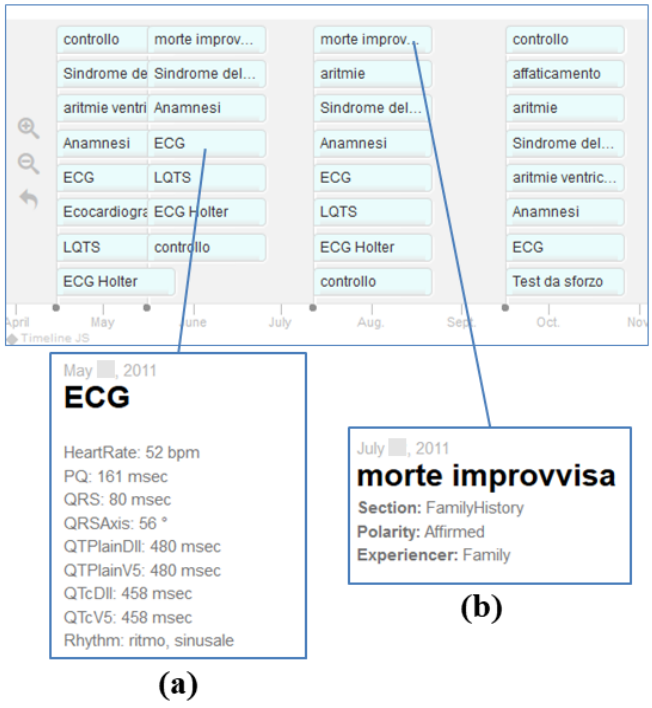

Figure 4.5: Portion of patient simple timeline with event details.

To allow comparing the simple and the complete versions of the timeline, Figure 4.6 and Figure 4.7 show these two different reconstructions for the same patient. As it can be noticed from these figures, the complete timeline includes a higher number of events, i.e., including those that were found through the temporal links. In particular, each of these "additional" events is also related to the document containing the event mention itself. For example, Figure 4.7 shows how the highlighted Echocardiogram event ("*Ecocardiogramma*") was found in a specific report ("Visit ID: 2201").

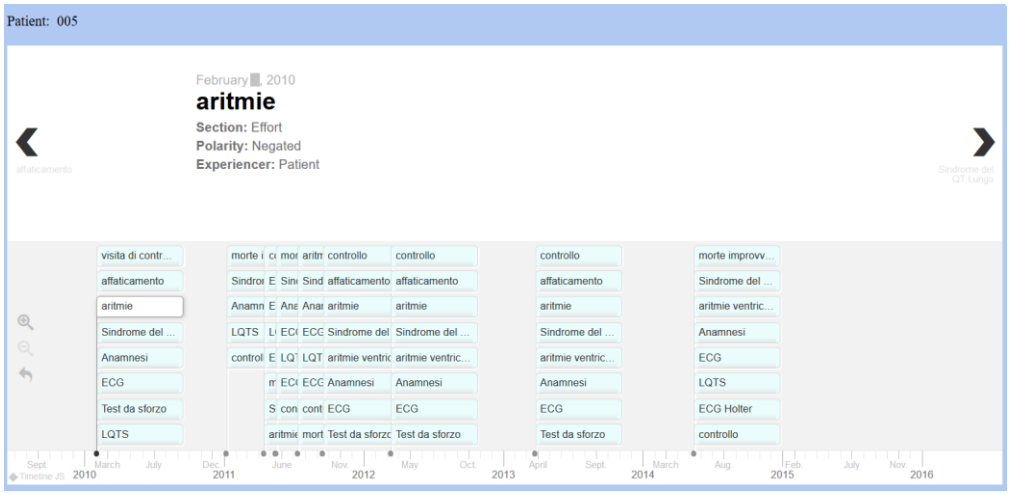

Figure 4.6: Patient simple timeline.

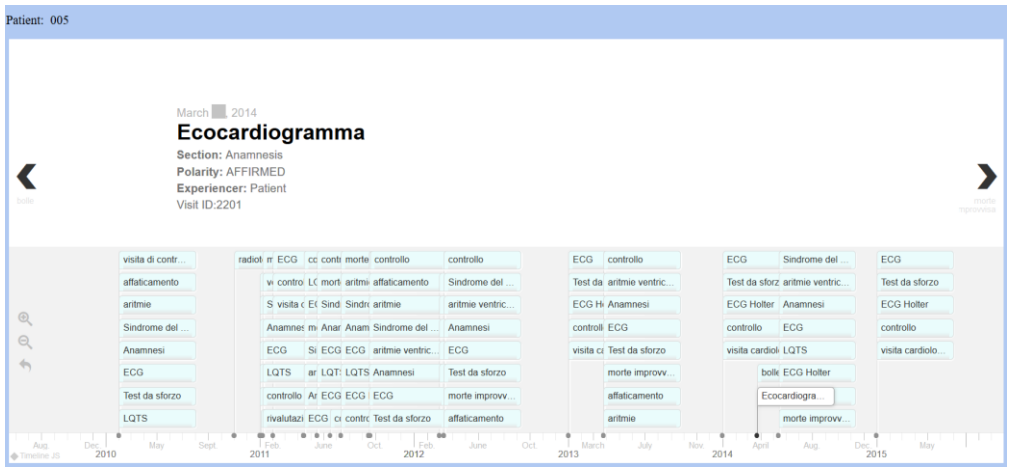

Figure 4.7: Patient complete timeline.

As explained in Section 3.7.2, the reconstructed timelines were visualized thought the TimelineJS tool. As an interesting visualization feature, TimelineJS allows both adjusting the temporal granularity and switching from one event to the other in a dynamic way. By exploiting

these functionalities, it is easy to visualize and compare events regardless of their temporal distance.

4.5.2. Discussion

The system presented in this research activity extracts and summarizes individual information included in longitudinal medical reports. Extracted information is aggregated and displayed on a clinical timeline that can be used in an interactive way.

Many methods have been proposed in the literature to automatically perform patient record summarization [96]. Although a few summarizers have been developed to deal with textual reports, most of them are intended to process documents written in English. This research activity is focused on the design and the implementation of summarization methods for the Italian language.

Although the proposed system is still at an early stage, the qualitative evaluation that was performed indicates that it is able to present information effectively and timely. For example, looking at the timeline portion reported in Figure 4.5-a, it is immediately clear that the selected patient performed an ECG test with certain results on a specific date. By improving both the NLP pipeline and the visualization strategy, it would be possible to enrich the timeline as needed.

Comparison to previous work. Besides the considered language, the proposed approach differs from previous works in other ways. As regards the way to display the information, Liu and Friedman use an XML tree structure to visualize summarized records [99], while Bashyam et al. present information as a problem list that can be used to populate a timeline grid [100]. In this research activity, it was decided to display aggregated data by building an interactive clinical timeline. This is similar to the approach proposed by Hirsch et al. [101]. However, while HARVEST is a problem-oriented system (since it renders patient data through a timeline and a problem cloud), this dissertation deals with events with different semantic types, including treatments and procedures. Moreover, additional information on the extracted events (e.g., attributes of interest) is integrated into the timeline.

Use in the clinical practice. The developed system has the potential to serve as a useful cognitive support for physicians for multiple reasons. First, it allows reducing the time needed to retrieve single patient documents and manually search for information in free-text. Second, it displays information belonging to the same patient on a single temporal line, facilitating the process of reviewing and making sense of multiple data points. Besides its usage for patient-level summarization, the system could be exploited to compare the clinical histories of different patients. This

analysis would be useful to support decision making in case of patients with similar records.

As regards the data sources to be summarized, this research activity is focused on unstructured medical reports. The application of NLP techniques to the clinical text is an important step to gather all the valuable information that is available for one patient. Nevertheless, relevant clinical data are often available also in other formats. As previously pointed out, in the Molecular Cardiology Unit of the ICS Maugeri hospital, patient data are stored in a structured database called TRIAD. In the future, it would be interesting to integrate this data source, too. As another major extension, the system could be enriched with non-textual data, such as diagnostic images and signals. From the technical point of view, these types of content could be easily integrated into the timeline, allowing physicians to visualize patient information from different sources at the same time.

Limitations. Although the proposed approach allows visualizing information effectively and timely, some limitations can be highlighted. As a first drawback, the duration of events is currently not considered, i.e., each event is related to a single specific date. This assumption does not accurately represent those conditions or prescriptions that start during one visit and last for a certain period of time. To overcome this limitation, further work on temporal relation extraction will be performed. In particular, for events that can be related to an interval of time rather than a date (e.g., temporal links with class "BEGINS_ON" or "ENDS_ON"), suitable temporal spans will be displayed.

Another limitation of this work concerns the lack of a comprehensive evaluation. As noted by Pivovarov et al., previous research on clinical summarization lacks standard evaluation metrics [96]. In the analyzed literature, indeed, different evaluation approaches were used. Liu and Friedman conducted an experiment to determine whether their system functioned properly and promptly: they generated the default views for three patients with more than 10 discharge summaries [99]. Bashyam et al. evaluated the different modules of the proposed NLP system, and obtained positive feedback from the users who tried the whole system on a small set of reports [100]. Hirsch et al., finally, assessed clinical usability with physician participants, using a timed, task-based chart review and questionnaire [101]. As a general consideration, evaluating a summarizer ideally requires either comparing automatically-created summaries to gold standard ones, or assessing its usefulness for the completion of a specific task. In the first case, a considerable human effort should be put into gold standard creation. In the second case, implementation into clinical care would be needed. Nevertheless, the extraction performance on single reports was evaluated (i.e., event, attribute, and TIMEX extraction), obtaining good results. As regards the whole summarizer, it was decided to perform a manual review of one patient clinical timeline. The resulting qualitative considerations could be considered as a preliminary evaluation for the complete system.

Chapter 5

Extensions and integrations

This chapter presents and discusses a few additional experiments conducted during this research activity. Section 5.1 describes a multilingual extension of the developed IE pipeline, considering English texts provided by the Molecular Cardiology Laboratories of the ICS Maugeri hospital. Section 5.2 discusses the extension of the event-attribute extraction task to a different domain, highlighting the main changes to be addressed. In particular, medical reports belonging to patients with breast cancer were considered. Finally, Section 5.3 illustrates two applications of the developed IE techniques in a real setting: the first one involves the cardiology domain, while the second one concerns oncology reports.

5.1. Multilingual extension

The IE pipeline presented in this dissertation was developed for processing medical reports written in Italian. However, most of the described modules rely on tools and resources that are applicable to other languages, too. For example, the TextPro tool, the UMLS dictionary, and the HeidelTime tool are available also for the English language. In addition, the RNN-based event annotator could be easily trained on English texts, provided the availability of annotated data.

As anticipated in Section 3.5.2, the attribute extraction module is the only step in the IE pipeline which heavily depends on the reports language, due to the use of regular expressions that are specific to the considered corpus. To assess the multilingual extendibility of the proposed approach, experiments were run on an English corpus belonging to the molecular cardiology domain. In this section, the results of this multilingual extension are described.

5.1.1. English cardiology dataset

To test the multilingual extension of the proposed approach, a suitable English corpus had to be used. Although the Molecular Cardiology Unit outpatient service is mostly delivered to Italian patients, it was possible to find 37 reports written in English (prepared for foreign patients), which had a similar structure with respect to the Italian ones. These reports were used for the translation of the ontology and the evaluation of the attribute extraction module. In particular, 10 documents were used as a guide to translate regular expressions, and the remaining 27 reports were used as the test set.

5.1.2. Validation against TRIAD

After adapting the pipeline to the analysis of English text, the resulting system (*system version 2-EN*) was run on the English test set (27 reports). As for the Italian pipeline, the evaluation was conducted against the TRIAD system.

Table 5.1 reports the results obtained for this multilingual extension. As it can be noticed from column "d" (Accuracy), the English pipeline obtained good accuracies, in particular for the extraction of diagnoses names (94.7%), ECG test attributes (94.9%), and drug names (85%). However, for events that are described with long sentences, a slightly lower performance can be highlighted with respect to the Italian counterpart. In particular, the English pipeline obtained accuracies of 86.2% for the Holter ECG test, and of 87.2% for the Efforts Stress test. The corresponding accuracies for the Italian pipeline were 92.8% and 95.2%, respectively (Table 4.8). This difference in results was probably caused by a few translations that were not straightforward, mainly due to syntactic differences among languages (e.g., word order).

Table 5.1: Multilingual extendibility results. SV: system version.

SV	Set	Event Name	System Items (a)	TRIAD Items (b)	Correct Annot. (c)	Accuracy (d)
2-EN	**EN test (27 docs)**	Main Diagnosis	27	19	18	**94.7%**
		ECG	183	78	74	**94.9%**
		Holter ECG	115	65	56	**86.2%**
		Effort Stress Test	110	39	34	**87.2%**
		Prescribed Drug	91 (91*)	20 (31*)	17 (23*)	**85% (74.2%*)**

5.1.3. Discussion

The ontology proposed in this research activity was developed for clinical text written in Italian. To evaluate the feasibility of a multilingual extension based on translation, the IE pipeline was adapted to process a set of clinical reports in the English language.

Multilingual extension. Although the considered corpus was small, this preliminary evaluation showed encouraging results. Despite this, it must be pointed out that the syntactic structure of sentences can be different across languages: the translation of concepts could be not straightforward, especially for those that are expressed by several words. To mitigate this issue, one option could be restricting attribute names to span only one word (e.g., "Regurgitation" instead of "Mitral regurgitation"), and including all other identifying words (e.g., "Mitral") in a suitable modifier, to be searched for in a lookup window. However, this solution would introduce a layer of complexity into the definition of the ontology. As another interesting observation, all the considered reports were written in English by Italian physicians, which might have introduced a translation bias. In other words, these documents could be closer to Italian reports than those written by native English speakers. In future work, it will be interesting to re-evaluate the system performance on a corpus of documents issued by an English institution. Finally, since the multilingual extendibility assessment was focused on the ontology, the different coverage of dictionaries across languages was not considered. Analyzing this aspect would be instead relevant in a more comprehensive assessment including the whole NLP pipeline.

Limitations. A main limitation of the described multilingual extension concerns the size of the considered corpus. To ultimately assess the feasibility of an approach based on ontology translation for multilingual extendibility, it would be necessary to consider a larger corpus, with more variability. Unfortunately, it was not possible to retrieve many documents that were originally written in English, and it was decided not to automatically translate reports to avoid introducing translation errors. Still, the evaluation that was conducted can be considered as a preliminary assessment, with promising results.

5.2. Extension to different domains

The IE pipeline presented in this dissertation was developed for processing medical reports in the field of molecular cardiology. However, most of the described modules could be used to process documents belonging to other domains, too. First of all, the UMLS Lookup Annotator searches for entries included in a wide terminology, containing terms from different medical specialties. As regards the other annotators, adaptations to other domains

are feasible as well. For example, the Event Annotator can receive as inputs different medical dictionaries, provided as external resources through a configuration file. The RNN-based classifier could be easily retrained on other texts, subject to the availability of reliable annotations. With respect the temporal IE modules, tuning the underlying resources on a specific domain should be straightforward, as it has been done for the molecular cardiology field.

Due to the use of an ontology, the attribute extraction module is the pipeline step with the highest dependency on the considered domain. To assess the extendibility of the proposed approach to other domains, experiments were run on a corpus of medical reports belonging to the oncology field. In this section, the results of this adaptation are described.

5.2.1. Oncology dataset

The extendibility of the developed pipeline was assessed on a corpus of medical reports provided by the Hospital Papa Giovanni XXIII in Bergamo, Italy. This corpus, which will be referred to as *ONCO dataset,* consists of 221 anatomic pathology reports belonging to patients with breast cancer.

Each report in the ONCO dataset is generated through an electronic form, which includes a set of predefined sections to be filled in by physicians. The most relevant sections are the following:

- Clinical information (*Notizie cliniche*), including references to previously performed tests.
- Sent specimen (*Materiale inviato*), describing one or more analyzed specimens, such as a breast quadrant or a nipple.
- Specimen description (*Testo macro*), containing details on the analyzed specimens (e.g., their size).
- Diagnosis (*Testo diagnosi*), illustrating the reached diagnostic conclusions (e.g., “invasive ductal carcinoma” or “no metastasis found”). In this section, the histopathological stage of the tumor and possible prognostic factors are usually reported.

In Figure 5.1, an example of a complete anatomic pathology report is depicted. In this case, four sections are highlighted: “clinical information”, including the references to three previous examinations, “sent specimen”, which lists five different specimens, “specimen description”, containing a few details about the analyzed specimens, and “diagnosis”, which reports the diagnostic conclusions for each of the analyzed specimens.

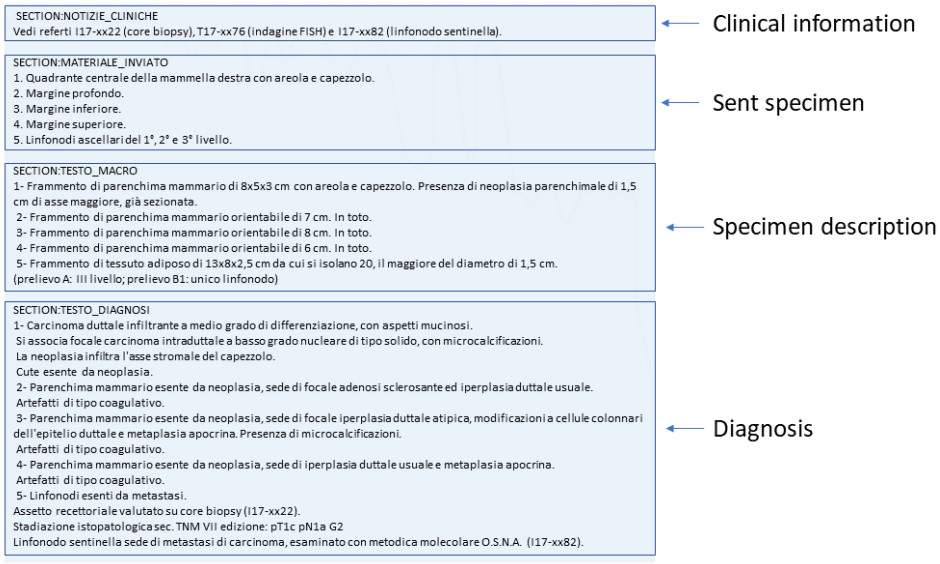

Figure 5.1: Example of a medical report in the ONCO dataset. The 4 sections composing the report are highlighted.

It is important to point out that specimen mentions can be found in both the "Sent specimen" and the "Specimen description" sections. As a matter of fact, physicians could use the first section to provide a very general description, and exploit the second section to characterize all the analyzed elements in detail. Moreover, whenever multiple items are mentioned in the "Sent specimen" section, each of them is assigned a different number (*specimen number*). These numbers are then used in the other sections to keep track of the specific item that is being referred to. In Figure 5.1, an example of this particular structure can be found.

As another important remark, each report might include different diagnoses, each related to one specific specimen. For example, in the report shown in Figure 5.1, an invasive ductal carcinoma was found in the first analyzed specimen (a breast quadrant, with the areola and the nipple), while the other specimens did not show signs of neoplasia (for three margins) or metastasis (for the axillary lymph nodes).

5.2.2. Information extraction task

As a first step for adapting the developed pipeline to the oncology domain, it was necessary to formalize the IE problem, identifying the information to be extracted from the texts. To this end, a set of 20 reports was randomly selected to be manually reviewed and discussed with physicians ("set for ontology design"). Moreover, to facilitate the identification of relevant concepts (including their variants), the n-grams that are most frequent in the considered dataset were computed.

As the result of these first analyses, the following relevant entities were identified:

- **Specimen**. Given that anatomic pathology reports describe pathologic findings on one or more specimens (e.g., core biopsies, organ portions), these entities were considered as one of the main targets for the IE task. To identify the most frequent specimen types, the Italian guidelines on the use of cells and tissues for diagnostic investigations were exploited [129]. In particular, according to these guidelines, all specimens can be grouped in two categories: biopsies or surgical resections.
- **Diagnosis**. Considering that each document contains relevant diagnostic conclusions (even in a negated form), these diagnoses represented another important information to be extracted. To specify the most relevant diagnoses that can be found in the texts, the medical knowledge provided by physicians was fundamental.
- **Histopathological stage**. In the case a breast cancer is found, the related report often includes its histopathological stage, which follows a standardized format called "TNM staging system" [130]. According to this format, specific characters are used to identify the tumor size (T), the lymph node involvement (N), and whether the cancer has metastasized (M). As the histopathological stage summarizes the main findings obtained by analyzing the specimen, extracting this entity was essential.
- **Prognostic factor**. Prognostic factors are patients' characteristics that are used to estimate the chance of recovery from a disease or the chance of a relapse [131]. Reports in the ONCO dataset frequently include an assessment of a few prognostic factors, such as the expression of estrogen receptors and progesterone receptors in a breast cancer. As discussed with physicians, it was important to capture the results of these assessments, too.

Ontology structure. In this research activity, to extract information from medical reports, it was proposed to use a domain ontology structured into event and attribute classes. To reuse this structure to analyze reports in the ONCO dataset (thus creating an *ONCO ontology*), the four identified entities were modeled as ontology events (specimen, diagnosis, histopathological stage, prognostic factor), and for each of these events a few attributes of interest were identified. For example, analyzed specimens can be related to a specific size (e.g., "8x5x3 cm"), while assessed prognostic factors can be linked to a test result (positive or negative). Moreover, both specimens and diagnoses can be related to the specimen number.

Event-Event relations. In the automatic processing of anatomic pathology reports, it is important to keep the relation between each extracted diagnosis and the specimen it refers to. In the ONCO ontology, diagnoses

and specimens were both represented as Events, each associated to specific attributes. To allow creating diagnosis-specimen links, therefore, an extension of the event-attribute approach was required. In particular, relations between pair of events were taken into account, too.

To allow linking each diagnosis to its related specimen, a new UIMA annotator was developed and integrated into the IE pipeline. For each diagnosis extracted from the text, the annotator looks for the related specimen in two different ways. First, a specimen mention is looked for in the same sentence containing the diagnosis. Second, the fact that each diagnosis and specimen can be related to a specific number is exploited. In particular, for those diagnoses that are linked to one number, this number is used to retrieve the associated specimens.

5.2.3. ONCO ontology development

To develop and refine the ONCO ontology and the annotation process, the same approach proposed for the CARDIO application was used (Section 4.3.1). The first version of the system (*system version 1*) was built on the "set for ontology design", considering both the information written in reports (with the aid of automatically computed n-grams) and the available domain knowledge. Then, the ontology and the annotator were refined in an iterative way, evaluating the system's performance on the same "design" set.

For the ONCO application, there was no availability of a structured database to be used as the gold standard. However, the system's output was iteratively validated through several discussions with the domain expert. Some of the modifications that were performed in this phase can be summarized as follows:

- For each found lump (e.g., a nodule), its distance from the resection margin was extracted. To disambiguate between a distance and a specimen size (both expressed as a number followed by "cm" or "mm"), a suitable modifier was used.
- For the computed distances, quantity modifiers such as "more than" or "less than" were identified, too.
- As ONCO reports frequently mention tests that were performed in the past, these references were extracted. In particular, each test was linked to the specific reference ID used in the text.

As the result of this refinement process, the final version of the system (*system version 2*) was obtained. This system version was evaluated on an independent test set (34 documents).

The ONCO ontology contains 44 events and 16 attributes: 3 attributes are numeric, the others are categorical.

The left-hand side of Figure 5.2 shows the class structure as it was defined in the Protégé framework. Both events and attributes are arranged into four main classes: Specimen, Diagnosis, Histopathology Stage, and Prognostic Factors. Specimens are grouped into biopsies and surgical resections, such as organs (e.g., left breast) and organ portions (e.g., left nipple). As an interesting characteristic, all specimens are related to a specific organ through an "hasAttribute" relation. Moreover, a few specimens can be linked to an organ portion or a nodule, too. Therefore, there are a few ontology classes (i.e., Localized Organ, Organ Portion, and Nodule) that can actually represent both events and attributes.

The right-hand side of Figure 5.2 shows one specimen, namely a Core Biopsy, that can be related to three "higher-level" specimens. As it can be noticed from its attribute list, this item can be related to (i) an organ (*LocalizedOrgan*), such as the left breast, (ii) an organ portion (*OrganPortion*), such as a breast quadrant, or (iii) a nodule (*Nodule*).

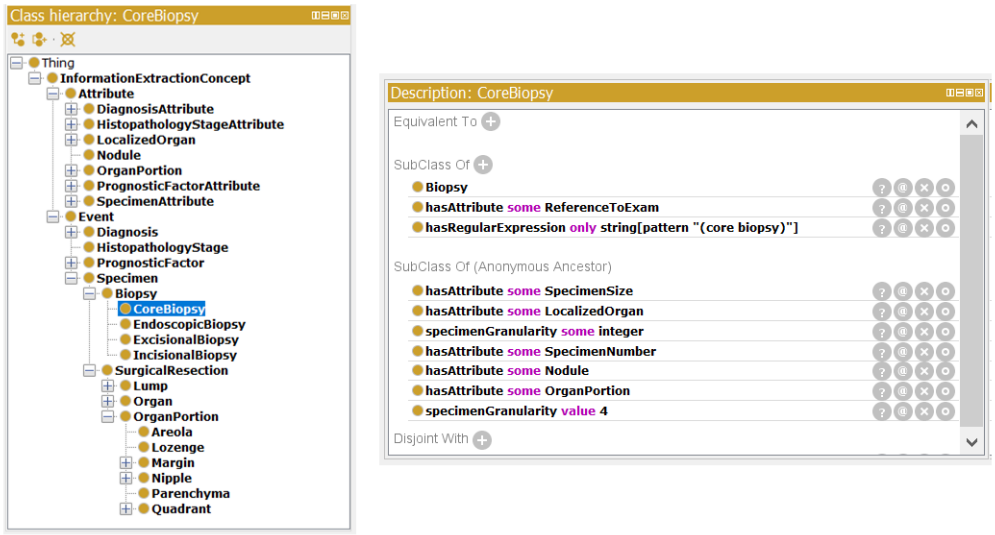

Figure 5.2: Domain ontology for the ONCO dataset. On the left side, the whole class structure is depicted. On the right side, the description of the Core Biopsy class is shown.

Specimen granularity property. In the ONCO reports, the "Sent specimen" section lists all the specimens that were analyzed. It often happens that one single line of this section mentions more than one specimen at different levels of detail (e.g. the left breast and the specification of the quadrant). To allow managing this feature, the ontology includes the "specimen granularity" property, which specifies the most specific item to be considered. For assigning a granularity value to each specimen in the ontology (according to its level of detail), the indications provided by physicians were followed. Going from the lowest to the highest level of detail, four different classes were identified: organs

(granularity of 1), organ portions (granularity of 2), nodules (granularity of 3), and biopsies (granularity of 4).

For those lines in the "Sent specimen" section which include multiple specimen mentions, the one with the highest granularity (i.e., the highest level of detail) is searched for. For example, the following line describes one specimen, more precisely a core biopsy (granularity of 4):

> Italian: *core biopsy del quadrante supero-esterno della mammella sinistra.*
> English: core biopsy of the upper outer quadrant of the left breast.

However, the same line includes two other specimen mentions, namely an upper outer quadrant (granularity of 2) and a left breast (granularity of 1). These mentions actually specify the organ portion and the localized organ to which the core biopsy is related. In this case, only the core biopsy (i.e., the specimen with the highest granularity) is considered by the following IE steps (i.e., attribute extraction and diagnosis-specimen relation extraction).

5.2.4. Validation with expert

To evaluate the final performance of the system, *system version 2* was run on a test set made up of 34 documents.

To give a sense of the task complexity, the number of items that were automatically extracted from each report was computed (*system items*). These items include: extracted events and their attributes (with associated values), specimen-number links, diagnosis-specimen links, and attribute quantity modifiers (e.g., "less than").

In the test set, a total of 476 system items were identified, which corresponds to an average of 14 extracted items per report. Figure 5.3 shows the number of system items in the 34 documents. In 22 reports, less than 10 items per document were identified, indicating that these documents are characterized by a simple content. In the most complex report, 68 system items were found.

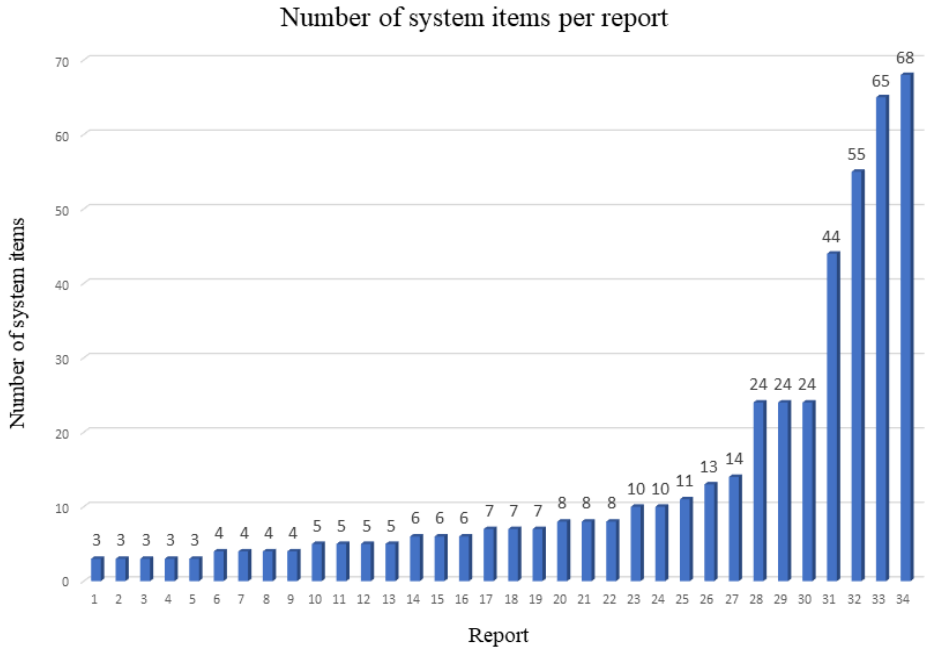

Figure 5.3: number of system items per report (ONCO test set).

To enable the evaluation of the IE system, the information extracted from each report was written on a suitable output file, including both the original report and the system items. This output was manually reviewed by a domain expert, who was trained to identify three types of errors:

- *Additional items*, i.e., relevant information that was not previously considered as an item to be extracted (and was therefore not included in the ONCO ontology).
- *False negatives (FN)*, i.e., information that should have been extracted but was not found in the system's output.
- *False positives (FP)*, i.e., errors found in the system's output, such as incorrect specimen-number associations or items linked to the wrong event.

In Table 5.2, the total number of additional items, false negatives, and false positives are shown ("raw count" column). These three groups were further analyzed by removing duplicates or similar entries ("distinct count" column); for example, although the string "c-erbB-2" was marked as an additional item in several reports, this item was counted only once in the "distinct count" computation.

As it can be noticed from the table, most errors were due to additional items that are currently not searched for (38 distinct items). As regards false negatives and false positives, which are instead a more direct measure of the performance of the IE system itself, the raw counts were 15 and 26, respectively.

Table 5.2: Evaluation results: error types (ONCO test set).

Items	Raw count	Distinct count
Additional items	57	38
FN	15	11
FP	26	21

For false negatives and false positives, an error analysis was performed, dividing both groups into relevant categories. For false negatives, three different categories were identified. For false positives, five main categories were highlighted. Table 5.3 reports the number of errors for each identified category.

As regards false negatives, 9 out of 15 errors were due to a missing variant among the ontology regular expressions. The other 6 errors were performed in the attribute annotation phase (e.g., the system was not able to identify an event's attribute).

With respect to false positives, most issues (9 out of 26) were given by the creation of wrong specimen-number links. Six errors were caused by a missing additional item, e.g., an attribute was linked to the wrong event because the real related event had not been extracted. In 4 cases, a specimen was not correctly recognized as another specimen's attribute, e.g., a quadrant was not identified as the organ portion of an extracted biopsy. In 4 cases, the same item was extracted twice, e.g., both "*carcinoma lobulare*" and its substring "*carcinoma*" were extracted as different events. Finally, three errors were due to the definition of an inadequate lookup window to search for an event's attributes, e.g., a specimen size was linked to the wrong specimen.

Table 5.3: Error analysis for false negatives and false positives (ONCO test set).

Items	Categories	# errors
FN	• considerable variation with respect to the regular expression in the ontology	6
	• small variation with respect to the regular expression in the ontology	3
	• error performed by attribute annotator	6
FP	• specimen-number link error	9
	• error due to missing additional item	6
	• specimen not recognized as an attribute	4
	• same information extracted twice	4
	• error due to inadequate lookup window	3

Starting from the identified error types, it was possible to compute the recall (R) and the precision (P) of the IE system. To this end, true positives

(TP) were defined as those system items that were regarded as completely correct by the domain expert. To compute true positives, it was sufficient to subtract false positives from the total number of system items:

$$TP = \#\ system\ items - FP$$

Given the numbers of true positives, false negatives, and false positives, the following formulas were applied:

$$R = \frac{TP}{TP + FN} \qquad P = \frac{TP}{TP + FP}$$

The so computed values were 96.8%, for recall, and 94.5%, for precision.

5.2.5. Discussion

In this research activity, an ontology-driven approach was proposed to extract events and attributes from medical reports in the molecular cardiology field. For assessing the possibility to extend this approach to another clinical domain, a corpus of anatomic pathology reports was considered.

Ontology adaptation. Despite the evident differences between the cardiology and the oncology reports, the proposed ontology structure was reused without major modifications. In particular, it was possible to exploit the event-attribute framework to model the relevant entities to be extracted.

As an interesting observation, the resulting ONCO ontology presents a different distribution of events and attributes with respect to the CARDIO one. While cardiology reports include more attributes than events (11 events and 61 attributes), the oncology dataset shows the opposite trend (44 events and 16 attributes). As a matter of fact, anatomic pathology reports are characterized by many different events, especially referred to specimens. To take into account this aspect, the specimen-related portion of the ontology was carefully developed, including advices coming from specific guidelines on this field [129]. Despite this variety, many of these events often share the same few attributes, which explains the relatively low number of Attribute classes that were created.

As an important result of the described adaptation, processing the ONCO dataset required to address a new IE task, which is the extraction of Event-Event relations. In particular, it was necessary to link each extracted diagnosis to the corresponding specimen. To this end, a suitable UIMA Annotator was developed, relying on two simple criteria. First, the intra-sentence occurrence of a diagnosis-specimen pair was considered. Second, the relation of each event to a specimen number was exploited. Although these two approaches do not allow reconstructing relations that are reported

in a complex way, they perform well when the relation to be extracted is clearly stated within the text. According to the positive feedback given by the domain expert and the evaluation results, the obtained output could be effectively used to help retrieve useful information for both diagnostic purposes and biomedical research.

One interesting observation regards the use of n-grams to facilitate the creation of the ontology. In this research activity, the n-grams that most frequently occur in the ONCO dataset were automatically computed and manually analyzed. Looking at these n-grams was particularly useful for extending the regular expressions included in the ontology. In future work, it would be interesting to automatically create the ontology starting from the extracted n-grams.

The developed IE system was manually evaluated by a domain expert, with promising results. In particular, most relevant items were extracted in the correct way, leading to a recall of 96.8% and a precision of 94.5%. Moreover, the conducted evaluation allowed identifying several items that were regarded as relevant by the domain expert, but were not previously considered (38 additional items). In a future version of the system, these items will be added to the existing ontology.

Limitations. The IE approach proposed for the ONCO dataset has a few limitations. First, the conducted evaluation highlighted a few issues concerning the annotation process (Table 5.3). The most frequent error type involved the extraction of specimen-number links. In particular, specimen sizes were often mistaken for specimen numbers, leading to the construction of an incorrect link. To address this issue, future work will focus on how to disambiguate these different items.

Another main limitation regards the small size of the corpus considered for the evaluation (34 documents). In future work, it will be interesting to extend the evaluation procedure to all the available anatomic pathology reports (i.e., those that were not included in the "set for ontology design"). As a matter of fact, the domain expert is currently validating more documents, which will allow gathering further suggestions on how to improve the system.

5.3. Exploitation in real settings

This research activity presents an NLP system that is able to retrieve the events included in unstructured medical reports, including their contextual properties (e.g., negations) and the available temporal details. Such a system could be exploited in the clinical practice for different purposes, above all to enable the access to the valuable information locked in free text. With respect to biomedical studies, populating research repositories with clinical data extracted from free text can considerably contribute to the reuse of collected data.

In this section, the problem of exploiting the developed IE pipeline in a real setting is discussed. In particular, two applications involving the use of Informatics for Integrating Biology and the Bedside (i2b2) software are presented [132].

The main goal of i2b2 is to provide a secure presentation of patient information (e.g., electronic health records and other patient data) to facilitate the reuse of clinical data for research purposes. Within this mission, the i2b2 software was designed to support two main use cases [133]. The first is to browse through all the available data to find sets of patients that would be of interest for further research. The second is to use the data provided by the medical record to analyze the phenotype of the identified patients, in support of subsequent studies. To empower these two use cases, in the i2b2 infrastructure data are collected into suitable repositories, which can be easily queried by researchers.

5.3.1. CARDIO i2b2

The ICS Maugeri hospital in Pavia is currently exploiting the i2b2 software to support research activities in different fields. Among these, several efforts have been made in molecular cardiology research.

The CARDIO-i2b2 project is an initiative to customize the i2b2 tool with the aim to support translational research in cardiology [134]. CARDIO-i2b2 integrates clinical and research data coming from multiple sources and allows the users to jointly query them. In particular, it gathers data from the Molecular Cardiology Laboratories databases and merges them with the clinical information from the TRIAD system. The collected data are then stored in the i2b2 data warehouse, where facts are hierarchically structured as ontologies. In 2012, a total of 591 patients, 13987 visits, 367 concepts and 262512 observations had been exported from TRIAD and inserted into the i2b2 data warehouse.

As part of this research activity, the information extracted through the cardiology IE pipeline is currently being integrated into the CARDIO-i2b2 data warehouse, together with the available structured data. To this end, the data extracted from each report are converted to a structured format that is suitable for i2b2 integration.

5.3.2. i2b2 Bergamo application

In the hospital Papa Giovanni XXIII in Bergamo, there are ongoing efforts to implement an i2b2 platform for research in the oncology field. The objective is to create a data warehouse that integrates all the information available for cancer patients in a predefined format. Currently, there are about 24.400 patients that are treated in the oncology unit. Out of these, about 23.500 have already been included in the i2b2 system.

For all hospital patients, a lot of relevant information is available in a variety of different forms, which does not facilitate data access and analysis. For example, about 19.000 unstructured histology reports related to breast cancer are currently available.

In this research activity, the proposed IE pipeline was adapted to the oncology domain, thus allowing the processing of the anatomic pathology reports produced by the Papa Giovanni XXIII hospital. In particular, the developed IE system was targeted to the field of breast cancer. As future work, the information extracted through this oncology pipeline will be integrated into the i2b2 data warehouse. To this end, the ontology used in the IE approach is currently being converted to a suitable i2b2 ontology, which will facilitate the data integration phase.

Figure 5.4 summarizes the proposed methodological approach. First, the ONCO ontology was developed starting from the available anatomic pathology reports. This ontology is currently exploited by an IE pipeline that converts the input texts into a structured output, to be stored in a i2b2 data warehouse. To be able to store and then query the extracted information, a suitable i2b2 taxonomy is being developed starting from the ONCO ontology.

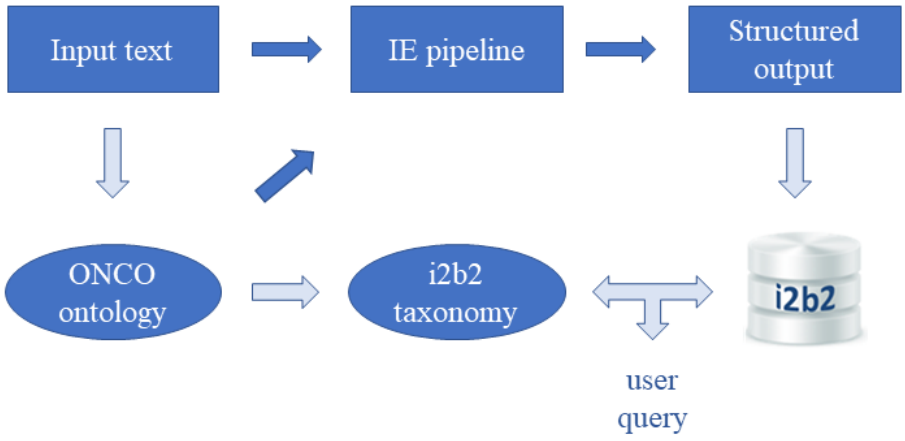

Figure 5.4: Ontology-driven IE and i2b2 ontology curation.

Chapter 6

Conclusions

This chapter presents the main conclusions of this research activity. Section 6.1 provides a summary of the proposed methodologies, with a focus on the obtained results. Section 6.2 highlights the main novelties and the contributions of the overall research. Finally, Section 6.3 outlines a few possible future directions for the conducted work.

6.1. Work summary and main results

This work addressed the problem of extracting information from medical reports in the Italian language, with the final aim of reconstructing patients' clinical timelines. Specifically, a system that extracts and summarizes individual information included in longitudinal medical reports was developed. To mine information from single free-text reports, an IE pipeline was implemented, made of different annotators. The pipeline processes each document by extracting clinical events and their attributes, as well as temporal expressions. Besides processing single documents, the developed system aggregates the data extracted from multiple reports referred to the same patient, reconstructing and visualizing events on a temporal line.

The methodologies proposed in this research activity were developed on a large corpus of documents belonging to the molecular cardiology domain. To enable the evaluation of the IE steps as well as the development of supervised classification methods, a subset of this corpus was manually annotated with mentions of relevant events, temporal expressions, and their properties (e.g., the event's polarity). To perform this annotation task, a set of specific guidelines was created.

The event extraction task was addressed by developing a supervised approach based on recurrent neural networks. To train and validate the RNN-based classifier, the annotated dataset was exploited. Developing classifiers that do not rely on complex features (that would require the use of specific NLP tools) is an important step to analyze documents across

different domains and languages. As clinical texts are often available only in the institution's local language, the availability of IE techniques that do not require language-specific resources is particularly important. The RNN-based approach developed in this work performed well for event extraction in the Italian language, indicating that the investigated RNN models (GRU neural networks) can be a good choice for this kind of tasks.

Besides extracting event mentions, this work focused on identifying their contextual properties, too. To extract the polarity, the modality, and the experiencer of each event, a rule-based approach was used. To identify the relation of each event to the document creation time, an SVM was created by exploiting the annotated dataset. For all properties, the proposed extraction methods obtained good results, indicating that they could be effectively reused for different IE tasks.

To extract the attributes that might be related to each event, an ontology-driven, rule-based approach was proposed. To identify and formalize relevant concepts, both domain knowledge and the information written in reports were exploited. The resulting IE system was partially validated against a structured clinical database named TRIAD. The results of this validation indicate that the proposed approach performs well, thus suggesting its eligibility to analyze languages such as Italian, where shared corpora and resources may not be easily accessible. Moreover, thanks to the use of regular expressions, the developed ontology can be easily enriched and translated. As regards multilingual extension, the performed preliminary assessment indicates that this approach could be effectively extended to other languages. With respect to the processing of documents belonging to different domains, the adaptation conducted on the oncology dataset shows that the proposed ontology structure could be adapted as needed.

For the identification and the normalization of temporal expressions, two existing rule-based systems, i.e., HeidelTime and TimeNorm, were adapted to the analysis of clinical documents. The performance of the proposed approaches was evaluated on the annotated dataset, leading to promising results. Therefore, as a first step for temporal IE on clinical corpora, tuning the investigated systems represents a good strategy, as it only requires the extension of external resources.

As a last step, the developed system summarizes the information extracted from longitudinal medical reports referred to the same patient. To this end, each event extracted from the reports is linked to a reference time. For all events, the relation to the document creation time is computed. If one event "overlaps" the document creation time, a temporal link to this reference time is created. Conversely, if one event is found to happen before or after the document creation date, a temporal link to another time is extracted, relying on a rule-based approach. The preliminary evaluation conducted on the reconstructed timelines indicates that the system could be successfully used to aggregate and present information in an effective way.

6.2. Main novelties and contributions

In the field of clinical NLP, several systems have been found in the literature, especially for the English language. Many systems include machine learning modules, developed through supervised techniques. To both develop such modules and enable their evaluation, the availability of annotated corpora is essential. For languages other than English, however, shared clinical corpora are not easily available. As a main consequence, research on IE from clinical narratives in non-English languages is still limited, especially as regards the analysis of temporal details. This research activity presents a system that extracts clinical events and temporal expressions from medical reports written in Italian. In the conducted work, various contributions and novelties can be highlighted.

A first novelty involves the annotation of resources in the Italian language, which is the language of interest for this research activity. In particular, a set of guidelines for temporally annotating a clinical corpus written in Italian was presented. As far as it is known, these are the first guidelines for the Italian language that include directions on both clinical and temporal event annotation, and were developed based on state-of-the-art guidelines in these two fields [11,91].

A second contribution of this work concerns the methodology proposed for the event extraction task. In this research activity, a supervised framework not relying on elaborately engineered features was proposed. In particular, the developed classifier is based on recurrent neural networks. As far as it is known, no other work has focused on exploiting these types of networks to perform event recognition on clinical narratives written in Italian. As mentioned, it was possible to find only one paper using RNN models, but on the general domain [57]. Another novelty related to the event extraction task regards the extraction of event properties, such as the polarity and the DocTimeRel. As far as it is known, this is the first time this problem is addressed for clinical narratives written in Italian.

Another contribution of this research activity concerns the event-attribute extraction task, which was addresses without the use of supervised machine learning techniques. The proposed ontology-driven approach, embedded in a well-performing NLP methodology, allows the analysis of the Italian language without the need for annotated data. Besides extracting clinical events, this approach allows capturing attributes of interest and linking them to the events they are related to. This particular task is not currently addressed by many of the systems available for clinical IE.

As one last contribution to be highlighted, this research activity deals with the problem of processing multiple medical reports referred to the same patient, with the aim of reconstructing an interactive clinical timeline. As far as it is known, no other work has addressed this problem for medical reports written in Italian.

Overall, the proposed IE approach has several potential implications to clinical practice. First of all, relevant information is extracted by using specific relations that are defined through a domain knowledge formalization. This feature allows converting textual content into a structured format that verifies an event-attribute logic, and can be easily queried. The possibility to timely access this clinical data is of paramount importance to facilitate patient review and support clinical decision. In addition, thanks to the proposed summarization approach, it is possible to visualize information belonging to the same patient on a single temporal line, which facilitates the process of analyzing multiple patient data.

As pointed out by Botsis et al., NLP methods can play an important role in reducing the health data that is unavailable, inaccessible or incomputable [8]. This is particularly important for biomedical research, too, especially as regards the reuse of collected data. In this research activity, there are ongoing efforts to integrate the data extracted through the developed IE pipeline into two i2b2 data warehouses. Integrating the information extracted from text into research repositories can considerably contribute to the reuse of collected data.

6.3. Future directions

For each of the conducted activities, some main future directions can be outlined.

Starting from the annotation task, a corpus of 75 documents was annotated with mentions of events and temporal expressions. The Event annotations were exploited for the development of a supervised classifier, while the TIMEX annotations were used for adequately tuning two temporal IE systems. As a first direction for future work, an extension of the annotated corpus will be performed, above all to evaluate the system's generalization capability. Moreover, two additional annotators will be involved, thus allowing the computation of an inter-annotator agreement. On the basis of this measure, it will be possible to assess the complexity of the extraction problems and better evaluate the performance of the developed IE systems.

Moving to the approach proposed for attribute extraction, the domain ontologies used in the CARDIO and in the ONCO contexts were manually developed. In future work, the possibility of automatically developing the ontology from free-text will be explored [126,127]. To this end, the most frequent n-grams extracted from the reports could be exploited. In addition, the concepts already present in UMLS could be reused. As another direction for future work, it will be interesting to consider the variants and the misspelled forms of the regular expressions included in the ontology, thus improving the search of concepts inside the text.

With respect to the extraction of events and attributes, two future applications of the developed pipeline can be delineated. As a first practical application, the cardiology pipeline could be used to automatically populate

TRIAD, thus saving a lot of manual work. In this case, using an automatic extraction system would be also useful to check the manually entered items to improve data quality. As another future work involving research on NLP, the annotations currently produced by the IE system could be considered as a pseudo-gold standard corpus, thus enabling the exploration of supervised methods for the event-attribute extraction task.

In this research activity, the problem of summarizing multiple reports referred to the same patient was addressed by reconstructing a clinical timeline (including all the events extracted for that patient). Also in this case, two future directions can be highlighted. First, it will be interesting to integrate additional data sources in the reconstructed timeline, beyond the information retrieved from free text. For example, possible structured data could be integrated, too. Another direction for future work regards the evaluation of the timeline from the point of view of the complete system. To assess its added value in a real clinical setting, an extrinsic evaluation involving physicians will be performed.

It is important to point out that, to integrate the proposed system in the clinical practice, a few practical issues should be taken into account. From the technical point of view, processing a large number of documents requires the availability of a pipeline that runs sufficiently fast. To this end, optimizing each of the extraction tasks is fundamental. Another aspect to be carefully considered regards the usability of the system. First, clinicians would need time to learn how to exploit it to retrieve relevant patient information (i.e., events and dates). Moreover, even if the system is easy to understand, they could still prefer to look for the needed information in a manual way. To moderate the effect of these potential obstacles, future work will concern system optimization and usability testing.

Appendix 1

Domain-specific vocabularies

In this research activity, to account for all the domain-specific concepts mentioned in the CARDIO dataset, two vocabularies were manually developed, containing 38 concepts with type Test, and 30 concepts with type Occurrence, respectively. The first vocabulary is shown in Table A1.1, while the second is reported in Table A1.2. Concepts are reported in the language used in the considered reports, i.e. Italian. For each concept, the corresponding UMLS concept unique identifier (CUI) and semantic type are shown, too.

Table A1.1: Concepts with type Test.

Concept	UMLS CUI	UMLS semantic type
Holter	C0013801	Diagnostic Procedure
ECG Holter	C0013801	Diagnostic Procedure
Holter ECG	C0013801	Diagnostic Procedure
ECG dinamico secondo Holter	C0013801	Diagnostic Procedure
controllo Holter	C0013801	Diagnostic Procedure
ECG	C1623258	Diagnostic Procedure
controllo ECG	C1623258	Diagnostic Procedure
esame ECOcardiografico	C0013516	Diagnostic Procedure
controllo ECOcardiografico	C0013516	Diagnostic Procedure
registrazione ECGgrafica	C1623258	Diagnostic Procedure
Studio elettrofisiologico	C0430467	Diagnostic Procedure
Potenziali tardivi	C0199591	Diagnostic Procedure
SAECG	C0199591	Diagnostic Procedure
Test ergometrico	C0015260	Diagnostic Procedure
Test da sforzo	C0015260	Diagnostic Procedure
Test da sforzo al treadmill	C0087110	Diagnostic Procedure
Test ergometrico al treadmill	C0087110	Diagnostic Procedure
EST	C0015260	Diagnostic Procedure
Test al treadmill	C0087110	Diagnostic Procedure
Walking test	C0430506	Diagnostic Procedure
RMN cuore	C3888835	Diagnostic Procedure
RM cuore	C3888835	Diagnostic Procedure
cineRM	C3888835	Diagnostic Procedure

Ecocolordoppler	C2022193	Diagnostic Procedure
Ecocolordopplergrafia	C2022193	Diagnostic Procedure
Ecocardiogramma	C0013516	Diagnostic Procedure
Esame Ecocardiografico	C0013516	Diagnostic Procedure
Test farmacologico	C1096540	Diagnostic Procedure
Test con Flecainide	C1096540	Diagnostic Procedure
Test alla Flecainide	C1096540	Diagnostic Procedure
Diagnostica molecolare	C1513388	Laboratory Procedure
Diagnosi molecolare	C1513388	Laboratory Procedure
analisi genetica	C0796344	Laboratory Procedure
analisi genetiche	C0796344	Laboratory Procedure
indagine genetica	C0796344	Laboratory Procedure
Esame obiettivo	C0031810	Diagnostic Procedure
Ematochimici	C0018941	Laboratory Procedure
Esami ematochimici	C0018941	Laboratory Procedure

Table A1.2: Concepts with type Occurrence.

Concept	UMLS CUI	UMLS semantic type
visita cardiologica	C1512346	Health Care Activity
visita cardiologica-genetica	C1512346	Health Care Activity
visita cardiologica	C1512346	Health Care Activity
visita	C1512346	Health Care Activity
valutazione cardiologica	C1512346	Health Care Activity
consulenza cardiologica	C0010210	Health Care Activity
consulenza cardiologica-genetica	C0017382	Therapeutic or Preventive Procedure
consulenza genetica	C0017382	Therapeutic or Preventive Procedure
visita di controllo	C0422303	Health Care Activity
controllo	C0422303	Health Care Activity
dimissione	C0030685	Health Care Activity
ricovero	C0184666	Health Care Activity
follow-up	C1522577	Health Care Activity
dimette	C0030685	Health Care Activity
dimettiamo	C0030685	Health Care Activity
ricovera	C0184666	Health Care Activity
ricoveriamo	C0184666	Health Care Activity
dimessa	C0030685	Health Care Activity
dimesse	C0030685	Health Care Activity
dimessi	C0030685	Health Care Activity
dimesso	C0030685	Health Care Activity
ricoverata	C0184666	Health Care Activity
ricoverate	C0184666	Health Care Activity
ricoverati	C0184666	Health Care Activity
ricoverato	C0184666	Health Care Activity
rivalutazione	C0422303	Health Care Activity
gravidanza	C0032961	Organism Function
parto	C1148523	Organism Function
partorito	C1148523	Organism Function
travaglio	C0022864	Organism Function

Appendix 2

Temporal Annotation Guidelines for Italian clinical text

The aim of this research activity is to extract clinical and temporal information from clinical narratives in the Italian language. These guidelines, which were developed on a set of medical reports belonging to the cardiology domain, show how to annotate the relevant events and the temporal expressions that can be found in medical reports. To develop the guidelines presented in this work, the THYME annotation guidelines [1] and the It-TimeML annotation guidelines [2] were used as a guide.

1. EVENTS

An EVENT is defined as "anything that is relevant to the patient's clinical timeline", including both clinical events and general events:

- Clinical events (tests, treatments or problems) are represented through noun phrases

  Examples for the Italian language: "*Sindrome di Brugada*" (Brugada Syndrome), "*Episodi sincopali*" (Syncopal episodes), "*Elettrocardiogramma*" (Electrocardiogram)

- General events are represented through noun phrases or verbs

  Examples for the Italian language: "*ricovero*" (admission), "*parto*" (childbirth), "*dimesso*" (discharged)

Clinical Events. To identify clinical events, the following concepts were considered:

- concepts included in UMLS (or synonyms) belonging to selected semantic types (problems, tests, treatments)
- concepts included in ICD9cm-diagnosis (or synonyms)

- drugs and active principles included in FederFarma (an Italian dictionary of drugs)
- diagnostic procedures and general events that are relevant in the cardiology domain

NOTE: In UMLS, relevant events are those belonging to one of the following semantic types:

- Pharmacologic Substance TREATMENT
- Antibiotic TREATMENT
- Therapeutic or Preventive Procedure TREATMENT
- Diagnostic Procedure TEST
- Sign or Symptom PROBLEM
- Injury or Poisoning PROBLEM
- Pathologic Function PROBLEM
- Mental or Behavioral Dysfunction PROBLEM
- Neoplastic Process PROBLEM
- Cell or Molecular Dysfunction PROBLEM
- Experimental Model of Disease PROBLEM
- Disease or Syndrome PROBLEM

General Events. OCCURRENCES are events that play a role in the patient's clinical history but are not included in the other three groups (e.g., "hospital admission", "discharge", etc.).

To identify OCCURRENCES, an external dictionary of concepts was manually developed, considering both the information included in the reports and the concepts annotated as Occurrences in the 2012 i2b2 Challenge ("Evaluating temporal relations in clinical text") [3].

Examples for the Italian language: "*visita cardiologica*" (cardiac examination), "*visita medica*" (medical examination), "*dimissione*" (discharge), "*ricovero*" (admission), "*follow-up*" (follow-up), "*dimettiamo*" (we discharge), "*attività sportiva*" (physical activity)

1.1 Event annotation procedure

- Results and findings of tests (e.g., "Rhythm" and "Heart Rate" for ECG tests) are not annotated as EVENTS. The only exception to not capturing the test result is when the result of a test is a

diagnosis (e.g., "prolonged QT interval"). In other words, a result is not a separate EVENT from the test which indicated it, but a diagnosis or disease shown by a test is.

- A very generic word such as "*test, esame, patologia, terapia, episodio, evento*" (test, exam, pathology, episode, event) should not be annotated when the context (e.g., the set of surrounding words) does not specify its meaning.
    - Examples of events to be annotated: "*terapia farmacologica*" (pharmacological therapy), "*terapia antiaritmica*" (antiarrhythmic therapy), "*episodio sincopale*" (syncopal episode), "*evento aritmico*" (arrhythmic event). All these strings represent specific events (their meaning is conveyed by the adjectives).
    - Exception: In a sentence like "*Dopo il primo ciclo di penicillina non è stata più effettuata la terapia*" (after the first round of penicillin, the therapy was no longer administered) the word "terapia" (therapy) should be annotated because its meaning is specified within the sentence. Moreover, this annotation is needed to capture the fact that the therapy has stopped.
- Words like "*mutazioni*" (mutations) and "*difetto*" (defect) should be annotated only when the context includes the related gene/disease.

1.2 Event properties

For each EVENT annotation, five properties of interest must be specified:

1) ***DocTimeRel***. This property represents the temporal relation between the EVENT and the document creation date:

    - BEFORE: this value is used for events that ended before the patient was seen (and thus, before the document was written).

    - AFTER: this value is used for events that are scheduled or planned to begin after the document creation time.

    - OVERLAP: this value is used for events or states which are true at the time that the patient was seen (and thus, when the document was written).

- BEFORE-OVERLAP (uncommon): this value is used for events that started BEFORE the document creation time and continue into and through this reference time (OVERLAP).

  Note: It is important to make sure that any time the BEFORE-OVERLAP relation is used, the documents explicitly states that the EVENT started before the document creation time and continues through this reference time.

2) ***Semantic Type***. This property represents the semantic type of the event. As previously explained, it takes one of four possible values: PROBLEM, TREATMENT, TEST or OCCURRENCE.

   Note: The annotator must be careful with words that could have different semantic types according to the context. For example, the word "*controllo*" (check) could be part of an OCCURRENCE or a TEST. An example for the first type is given by the sentence "*sottoponiamo il paziente ad un controllo*" (we examine the patient). Two examples for the second type are given by "*controllo della pressione arteriosa*" (blood pressure test) and "*controllo Holter*" (Holter test).

3) ***Polarity***. This property is used to explicitly indicate whether an event has occurred (POSITIVE) or not (NEGATIVE). As an important remark, a NEGATIVE value means "did not happen" or "not true", rather than "negative" in the medical testing sense.

4) ***Contextual modality***. This property provides information about the event's modality:

   - ACTUAL (default): it is used for events that have already happened or are scheduled (without hedging) to happen. This value is used most of the time, and is the default option.

     Note: test that have already been scheduled should have an ACTUAL modality.

   - HEDGED: events are marked as hedged when they are mentioned with any sort of hedging. This hedging can be lexical ("seems", "likely", "suspicious", "possible", "consistent with"), or phrasal ("I suspect that...", "It would seem likely that").

     Examples for the Italian language: events that are referred to as "*sospetto*", "*suggestivo di*", "*probabile*".

- HYPOTHETICAL: this value is useful for annotating hypothetical events (e.g., diagnoses, theories). Hypothetical events will often follow "if" statements ("If X happens, then we'll perform Y") or other sorts of conditionals ("Depending on the patient's response, we might treat with B or with C").

  Examples for the Italian language:
  - "*Si raccomanda di evitare stati di elevato stress psico-fisico*."
  - "*Utile l'assunzione di integratori di potassio in caso di pratica di attività sportiva, sudorazione eccessiva, vomito o diarrea*".
  - "*è stato spiegato al paziente che sarebbe utile seguire Test farmacologico*"
  - "*si potrebbe considerare la terapia…*"
  - "*al fine di garantire protezione da eventuali eventi aritmici…*"

- GENERIC: this value is used for events which may be mentioned in a note, but are only mentioned in a general sense, and **should not appear on the patient's clinical timeline**.

  Example for the Italian language: "*Secondo i più recenti studi, nei pazienti con Sindrome di Brugada è indicato…*"

  Note: if an EVENT is GENERIC, the DocTimeRel will always be OVERLAP.

5) ***Experiencer***: this property identifies the subject to which an event refers. There are two possible values for the Experiencer property: PATIENT or OTHER.

   Note: for most GENERIC events, the Experiencer will be OTHER.

2. TIMEXes

A TIMEX is a reference to time. Examples might be phrases like "today", "tomorrow", "24 hours ago", "at this time" and "early March". In addition, specific dates are annotated as TIMEX objects as well.

- Noun phrases ("this weekend", "tomorrow", "yesterday", "Tuesday", "Last May", "May 16th", "6/9/1985").

  Examples for the Italian language: *lunedì* (Monday), *il mese scorso* (last month).

- Adjective phrase ("two-hour-long", "half-hour" as in "a half-hour trip", "preoperative", "post-partum").

  Examples for the Italian language: *annuale* (annual), *mensile* (monthly), *quotidiano* (daily).

- Adverbial phrase ("lately", "recently", "shortly", "hourly", "intraoperatively").

  Examples for the Italian language: *oggi* (today), *ieri* (yesterday), *finora* (so far).

- Time/Date patterns, such as “31-12-2006”, “14.30”, “24/08”.

2.1 TIMEX annotation procedure

- Any preposition which precede (or in some cases, follow) a temporal expression must be left unmarked, even when it seems to provide additional context for interpreting the TIMEX (e.g., “During”, “From”, “After”, etc.).

  Exceptions for the Italian language:

  - “*circa*” (about), “*intorno a*” (around), “*verso*” (around). These propositions must be included into the extent of the tag because they have a role in the normalization of the TIMEX;
  - “*per ora*” (for now), “*dopo domani*” (the day after tomorrow), “*di recente*” (recently). Given that these whole expressions are considered as single units, the prepositions must be annotated, too;

- All pre- and post-modifiers of a temporal expression must be included into the tag.

  Example for the Italian language: “*durante lo scorso mese*” (during the last month).

- The word “*dopo*” (after) must be included into the tag span only when it has the function of adjective, otherwise it must be excluded.

  Examples for the Italian language: “*tre giorni dopo*”, but “*dopo tre giorni*”.

2.2 TIMEX properties

For each TIMEX annotation, five properties of interest can be specified:

1) ***Type***: this property represents the type of the temporal expression.

- DATE: These TIMEXes can be calendar dates (“January 4”) and other verbal expressions which can be mapped to calendar dates either concretely (“last week”, “this month”, “next Friday”, or “this time”), or in a fuzzier sense (“lately”, “the past”);

- TIME: this value is used for specific time points within a day, for instance, “3:00PM” or “23:45”;

  Note: In other words, temporal expressions which give minute-by-minute or hour-by-hour detail are marked as TIME. Day-by-day (or larger) details are marked with DATE.

- DURATION: this value is used for temporal expressions denoting a span of time, rather than a point ("24 hours" or "all of February").

  Note: Two dates can be used to construct a duration. However, since each TIMEX represents a single point in time (rather than a duration), both will still be labeled as DATE.

  Example: “From May 1st$_{DATE}$ to the 3rd$_{DATE}$, she will refrain from eating solid food”.

- SET: this value is used to describe reoccurring temporal expressions (e.g., "three times weekly", "monthly", or “1/day”).

2) ***Value***: this property represents the normalized value of the temporal expression. Its annotation is strictly dependent upon the type assigned to the temporal expression. In the following, a few examples for each TIMEX type are provided (in the Italian language).

DATE
Format: YYYY-MM-[WW]-DD

<TIMEX type="DATE" value="2008-12-02">venerdì due dicembre 2008</TIMEX>

<TIMEX type="DATE" value="2008-W49">questa settimana</TIMEX>

If some information cannot be recovered from the context, then the missing information must be signaled using the placeholders X.

Ad <TIMEX type="DATE" value="XXXX-08-XX">agosto</TIMEX>

Note: The word "*oggi*" (today) is given a different value according to the surrounding context.

Example 1: "nowadays, there are many satellite channels in television"

<TIMEX type="DATE" value="PRESENT_REF">oggi</TIMEX>ci sono moltissimi canali satellitari in TV

Example 2: "Today the new government takes office"

<TIMEX type="DATE" value="2008-12-02">oggi</TIMEX>si insedia il nuovo Parlamento

Special cases (from the It-TimeML annotation guidelines [2])

Temporal Expression	value	Annotation sample
al momento, per il momento, in questi giorni, tuttora, per ora, nel presente, a oggi, adesso, attuale	PRESENT_REF	value="PRESENT_REF"
recentemente, in passato, tempo fa, poco fa, ex, passato	PAST_REF	value="PAST_REF"
al/in futuro, il domani (generic reference) *futuro*	FUTURE_REF	value="FUTURE_REF"
autunno, autunnale	FA	value="XXXX-FA"
primavera, primaverile	SP	value="XXXX-SP"
estate, estivo	SU	value="XXXX-SU"
inverno, invernale	WI	value="XXXX-WI"
fine settimana, week-end	WE	value="XXXX-XX-WE"
semestre	H	value="XXXX-H1"
trimestre	Q	value="XXXX-Q1"
quadrimestre	Qu	value="XXXX-Qu1"
bimestre	B	value="XXXX-B1"

TIME
Formats: THH:MM:SS, THH:MM or THH

<TIMEX type="TIME" value="T16:00"> le 16.00 </TIMEX>

<TIMEX type="DATE" value="2008-11-27"> Ieri </TIMEX> alle <TIMEX type="TIME" value="2008-11-27T16:00"> 16.00 </TIMEX>

Special cases (from the It-TimeML annotation guidelines [2])

Temporal Expression	value	Annotation sample
mattina	MO	value="XXXX-XX-XXTMO"
mezzogiorno, mezzodì	MI	value="XXXX-XX-XXTMI"
pomeriggio	AF	value="XXXX-XX-XXTAF"
sera, serata	EV	value="XXXX-XX-XXTEV"
notte, nottata	NI	value="XXXX-XX-XXTNI"
giorno (day time or working hours)	DT	value="XXXX-XX-XXTDT"

DURATION
Format: the value begins with a 'P'

<TIMEX type="DURATION" value="P4M">4 mesi</TIMEX>

Per <TIMEX type="DURATION" value="PT45M">45 minuti</TIMEX>

<TIMEX type="DURATION" value="PXY">alcuni anni fa</TIMEX>

<TIMEX type="DURATION" value="P0.5D">mezza giornata</TIMEX>

Note: special tokens are used to represent durations referring to periods of the day (MO, MI, AF, EV, NI, DT), weekends (WE), seasons (SP, SU, FA, WI), quarters (Q), year halves (H), and fiscal years (FY).

Example: three nights

<TIMEX type="DURATION" value="PT3NI">3 notti</TIMEX>

SET

To fully annotate sets, the TIMEX tag can include either the **quant** or **freq** properties, if not both.
The quant and freq properties (optional) are fulfilled only in presence of specific expressions realizing them, they cannot be inferred.

Example 1: "once a week"

<TIMEX type="SET" value="P1W" freq="1X">una volta a settimana</TIMEX>

Example 2: "every three days"

<TIMEX type="SET" value="P3D" quant="ogni">ogni tre giorni</TIMEX>

Example 3: “three days per week”

<TIMEX type="SET" value="P1W" freq="3D">3 giorni a settimana</TIMEX>

Example 4: “every October”

<TIMEX type="SET" value="XXXX-10" quant="ogni">ogni ottobre</TIMEX>

Note: In the last example, the value looks like a point and not a duration: in this way it is possible to mark the calendar information (e.g., every October) present in the temporal expression. The general rule, useful to understand when to use a DATE-like annotation instead of a DURATION-like format, is that if there is no specified calendar date (for example, October or Monday), then the value for the SET will be like that of a DURATION.

3) ***Quant***: see Value property for type SET

4) ***Freq***: see Value property for type SET

5) ***Mod***: this property represents the modality of the temporal expression. In this research activity, either the default value or the “APPROX” value can be used. In particular, the “APPROX” value is used for approximate TIMEXes, such as “*all’incirca tra un anno*” (in about one year).

3. DOCTIME

The DOCTIME represents the date in which the report was written. This date must be annotated by using the TIMEX notation.

References for guidelines development

[1] Styler IV WF, Bethard S, Finan S, Palmer M, Pradhan S, de Groen PC, et al. Temporal annotation in the clinical domain. Trans Assoc Comput Linguist 2014;2:143–154.

[2] Caselli T, Lenzi VB, Sprugnoli R, Pianta E, Prodanof I. Annotating events, temporal expressions and relations in Italian: the It-TimeML experience for the Ita-TimeBank. Proc. 5th Linguist. Annot. Workshop, Association for Computational Linguistics; 2011, p. 143–151.

[3] Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. J Am Med Inform Assoc JAMIA 2013;20:806–13.

References

1. Friedman C, Hripcsak G. Natural language processing and its future in medicine. Acad Med J Assoc Am Med Coll. 1999 Aug;74(8):890–5.

2. Hirschberg J, Manning CD. Advances in natural language processing. Science. 2015 Jul 17;349(6245):261–6.

3. Piskorski J, Yangarber R. Information Extraction: Past, Present and Future. In: Poibeau T, Saggion H, Piskorski J, Yangarber R, editors. Multi-source, Multilingual Information Extraction and Summarization. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 23–49.

4. Sun W, Rumshisky A, Uzuner O. Temporal reasoning over clinical text: the state of the art. J Am Med Inform Assoc. 2013 Sep 1;20(5):814–9.

5. Feblowitz JC, Wright A, Singh H, Samal L, Sittig DF. Summarization of clinical information: A conceptual model. J Biomed Inform. 2011 Aug;44(4):688–99.

6. Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inf. 2008;128–44.

7. Velupillai S, Mowery D, South BR, Kvist M, Dalianis H. Recent Advances in Clinical Natural Language Processing in Support of Semantic Analysis. Yearb Med Inform. 2015 Aug 13;10(1):183–93.

8. Botsis T, Hartvigsen G, Chen F, Weng C. Secondary Use of EHR: Data Quality Issues and Informatics Opportunities. Summit Transl Bioinforma. 2010 Mar 1;2010:1–5.

9. Pustejovsky J, Castano JM, Ingria R, Sauri R, Gaizauskas RJ, Setzer A, et al. TimeML: Robust specification of event and temporal expressions in text. New Dir Quest Answering. 2003;3:28–34.

10. Sun W, Rumshisky A, Uzuner O. Annotating Temporal Information in Clinical Narratives. J Biomed Inform. 2013 Dec;46(0).

11. Styler IV WF, Bethard S, Finan S, Palmer M, Pradhan S, de Groen PC, et al. Temporal annotation in the clinical domain. Trans Assoc Comput Linguist. 2014;2:143–154.

12. Jurafsky D, Martin JH. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR; 2000.

13. Beesley KR, Karttunen L. Finite State Morphology. CSLI Publications; 2003. 536 p.

14. Zanchetta E, Baroni M. Morph-it! A free corpus-based morphological resource for the Italian language. In: Proceedings of Corpus Linguistics 2005. 2005.

15. Tjong Kim Sang EF, Buchholz S. Introduction to the CoNLL-2000 shared task: Chunking. In: Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning. Association for Computational Linguistics; 2000. p. 127–132.

16. Pakhomov S, Buntrock J, Duffy P. High Throughput Modularized NLP System for Clinical Text. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions. Stroudsburg, PA, USA; 2005. p. 25–28.

17. Xu H, Stenner SP, Doan S, Johnson KB, Waitman LR, Denny JC. MedEx: a medication information extraction system for clinical narratives. J Am Med Inform Assoc JAMIA. 2010 Feb;17(1):19–24.

18. Liu S, Ma W, Moore R, Ganesan V, Nelson S. RxNorm: prescription for electronic drug information exchange. IT Prof. 2005 Sep;7(5):17–23.

19. Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med. 1993 Aug;32(4):281–91.

20. ICD - ICD-10-CM - International Classification of Diseases, Tenth Revision, Clinical Modification [Internet]. 2017 [cited 2017 Jul 12]. Available from: https://www.cdc.gov/nchs/icd/icd10cm.htm

21. SNOMED International [Internet]. [cited 2017 Jul 12]. Available from: http://www.snomed.org/snomed-ct

22. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc JAMIA. 2010 Jun;17(3):229–36.

23. Friedman C. A broad-coverage natural language processing system. Proc AMIA Symp. 2000;270–4.

24. Zhou L, Plasek JM, Mahoney LM, Karipineni N, Chang F, Yan X, et al. Using Medical Text Extraction, Reasoning and Mapping System (MTERMS) to Process Medication Information in Outpatient Clinical Notes. AMIA Annu Symp Proc. 2011;2011:1639–48.

25. Carrell DS, Halgrim S, Tran D-T, Buist DSM, Chubak J, Chapman WW, et al. Using Natural Language Processing to Improve Efficiency of Manual Chart Abstraction in Research: The Case of Breast Cancer Recurrence. Am J Epidemiol. 2014 Mar 15;179(6):749–58.

26. Patterson OV, Freiberg MS, Skanderson M, J Fodeh S, Brandt CA, DuVall SL. Unlocking echocardiogram measurements for heart disease research through natural language processing. BMC Cardiovasc Disord. 2017 Jun 12;17(1):151.

27. Wimalasuriya DC, Dou D. Ontology-based information extraction: An introduction and a survey of current approaches. J Inf Sci. 2010 Jun 1;36(3):306–23.

28. Spasic I, Ananiadou S, McNaught J, Kumar A. Text mining and ontologies in biomedicine: making sense of raw text. Brief Bioinform. 2005 Sep;6(3):239–51.

29. Spasić I, Zhao B, Jones CB, Button K. KneeTex: an ontology–driven system for information extraction from MRI reports. J Biomed Semant. 2015;6:34.

30. Mykowiecka A, Marciniak M, Kupść A. Rule-based information extraction from patients' clinical data. J Biomed Inform. 2009 Oct;42(5):923–36.

31. Toepfer M, Corovic H, Fette G, Klügl P, Störk S, Puppe F. Fine-grained information extraction from German transthoracic echocardiography reports. BMC Med Inform Decis Mak. 2015 Nov 12;15.

32. Strötgen J, Gertz M. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In: Proceedings of the 5th International Workshop on Semantic Evaluation. 2010. p. 321–324.

33. Bethard S. A Synchronous Context Free Grammar for Time Normalization. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013. p. 821–826.

34. Chiang D. An introduction to synchronous CFGs [Internet]. 2006 [cited 2017 Oct 11]. Available from: http://www3.nd.edu/~dchiang/papers/synchtut.pdf

35. Chapman WW, Nadkarni PM, Hirschman L, D'Avolio LW, Savova GK, Uzuner O. Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. J Am Med Inform Assoc JAMIA. 2011 Oct;18(5):540–3.

36. Albright D, Lanfranchi A, Fredriksen A, Styler WF, Warner C, Hwang JD, et al. Towards comprehensive syntactic and semantic annotations of

the clinical narrative. J Am Med Inform Assoc JAMIA. 2013 Oct;20(5):922–30.

37. Elhadad N, Chapman WW, O' Gorman T, Palmer M, Savova GK. The ShARe Schema for the Syntactic and Semantic Annotation of Clinical Texts. Under Review.

38. Uzuner Ö, South BR, Shen S, DuVall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc JAMIA. 2011;18(5):552–6.

39. Pradhan S, Elhadad N, South BR, Martinez D, Christensen L, Vogel A, et al. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. J Am Med Inform Assoc JAMIA. 2015 Jan;22(1):143–54.

40. Elhadad N, Pradhan S, Lipsky Gorman S, Chapman WW, Manandhar S, Savova GK. SemEval-2015 task 14: Analysis of clinical text. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 2015. p. 303–10.

41. Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. J Am Med Inform Assoc JAMIA. 2013 Sep;20(5):806–13.

42. Bethard S, Derczynski L, Pustejovsky J, Verhagen M. Clinical TempEval. ArXiv14034928 Cs. 2014 Mar 19;

43. Bethard S, Derczynski L, Savova G, Pustejovsky J, Verhagen M. SemEval-2015 Task 6: Clinical TempEval. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 2015. p. 806–814.

44. Bethard S, Savova G, Chen W-T, Derczynski L, Pustejovsky J, Verhagen M. Semeval-2016 task 12: Clinical tempeval. Proc SemEval 2016. 2016;1052–1062.

45. Sutton C, McCallum A. An Introduction to Conditional Random Fields. Found Trends® Mach Learn. 2012;4(4):267–373.

46. Gunn SR, others. Support vector machines for classification and regression. ISIS Tech Rep. 1998;14:85–86.

47. Lee H-J, Xu H, Wang J, Zhang Y, Moon S, Xu J, et al. UTHealth at SemEval-2016 Task 12: an End-to-End System for Temporal Information Extraction from Clinical Notes. In: Proceedings of SemEval-2016. 2016. p. 1292–1297.

48. Tourille J, Ferret O, Tannier X, Névéol A. Temporal information extraction from clinical text. In: Proceedings of the 15th Conference of the

European Chapter of the Association for Computational Linguistics. 2017. p. 739–45.

49. Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics. 2010. p. 384–394.

50. Mikolov T, Karafiát M, Burget L, Cernockỳ J, Khudanpur S. Recurrent neural network based language model. In: Proceedings of INTERSPEECH 2010. 2010. p. 1045–8.

51. Goldberg Y. A Primer on Neural Network Models for Natural Language Processing. arXiv:151000726. 2015 Oct 2;

52. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res. 2011;12(Aug):2493–2537.

53. Mesnil G, He X, Deng L, Bengio Y. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Proceedings of INTERSPEECH 2013. 2013.

54. Hammerton J. Named entity recognition with long short-term memory. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003. 2003. p. 172–175.

55. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. arXiv:160301360. 2016;

56. Li P, Huang H. Clinical Information Extraction via Convolutional Neural Network. arXiv:160309381. 2016 Mar 30;

57. Bonadiman D, Severyn A, Moschitti A. Deep Neural Networks for Named Entity Recognition in Italian. In: Proceedings of CLiC-IT 2015. 2015.

58. Santos CN dos, Guimarães V. Boosting Named Entity Recognition with Neural Character Embeddings. ArXiv150505008 Cs. 2015 May 19;

59. Graves A. Supervised Sequence Labelling with Recurrent Neural Networks. In: Studies in Computational Intelligence. Springer; 2012.

60. Bottou L. Stochastic Gradient Descent Tricks. Neural Networks: Tricks of the Trade. Springer. 2012;421–36.

61. Sahlgren M. The distributional hypothesis. Ital J Disabil Stud. 2008;20:33–53.

62. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:13013781. 2013;

63. Mikolov T, Yih W, Zweig G. Linguistic Regularities in Continuous Space Word Representations. In: Proceedings of NAACL-HLT 2013. 2013. p. 746–751.

64. Erhan D, Bengio Y, Courville A, Manzagol P-A, Vincent P, Bengio S. Why does unsupervised pre-training help deep learning? J Mach Learn Res. 2010;11(Feb):625–660.

65. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc JAMIA. 2010;17(5):507–13.

66. Wang C, Akella R. A Hybrid Approach to Extracting Disorder Mentions from Clinical Notes. AMIA Summits Transl Sci Proc. 2015 Mar 25;2015:183–7.

67. Kovačević A, Dehghan A, Filannino M, Keane JA, Nenadic G. Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives. J Am Med Inform Assoc. 2013 Sep;20(5):859–66.

68. Cunningham H, Maynard D, Bontcheva K, Tablan V. GATE: An Architecture for Development of Robust HLT Applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA; 2002. p. 168–175.

69. Maynard D, Roberts I, Greenwood MA, Rout D, Bontcheva K. A framework for real-time semantic social media analysis. Web Semant Sci Serv Agents World Wide Web. 2017 May 12;

70. Cunningham H, Tablan V, Roberts A, Bontcheva K. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. PLOS Comput Biol. 2013 Feb 7;9(2):e1002854.

71. Ferrucci D, Lally A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Nat Lang Eng. 2004;10(3–4):327–348.

72. BioNLP UIMA Component Repository [Internet]. [cited 2017 Jul 26]. Available from: http://bionlp-uima.sourceforge.net/

73. Manning CD, Surdeanu M, Bauer J, Finkel JR, Bethard S, McClosky D. The stanford corenlp natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014. p. 55–60.

74. Apache OpenNLP [Internet]. [cited 2017 Jul 26]. Available from: https://opennlp.apache.org/

75. Pianta E, Girardi C, Zanoli R. The TextPro Tool Suite. In: Proceedings of the 6th edition of the Language Resources and Evaluation Conference. 2008.

76. Evaluation of NLP and Speech Tools for Italian | EVALITA [Internet]. [cited 2017 Jul 26]. Available from: http://www.evalita.it/

77. Schadow G, McDonald CJ. Extracting Structured Information from Free Text Pathology Reports. AMIA Annu Symp Proc. 2003;2003:584–8.

78. Meystre S, Haug PJ. Evaluation of medical problem extraction from electronic clinical documents using MetaMap Transfer (MMTx). Stud Health Technol Inform. 2005;116:823–828.

79. cTAKES 4.0 Component Use Guide - Apache cTAKES - Apache Software Foundation [Internet]. [cited 2017 Jul 28]. Available from: https://cwiki.apache.org/confluence/display/CTAKES/cTAKES+4.0+Component+Use+Guide#cTAKES4.0ComponentUseGuide-ComponentOverview

80. Tang B, Wu Y, Jiang M, Denny JC, Xu H. Recognizing and Encoding Discorder Concepts in Clinical Text using Machine Learning and Vector Space Model. In: Proceedings of CLEF 2013. 2013.

81. Velupillai S, Mowery DL, Abdelrahman S, Christensen L, Chapman WW. Blulab: Temporal information extraction for the 2015 clinical tempeval challenge. In: The 9th International Workshop on Semantic Evaluation (SemEval 2015). 2015. p. 815–819.

82. Chiaramello E, Pinciroli F, Bonalumi A, Caroli A, Tognola G. Use of "off-the-shelf" information extraction algorithms in clinical informatics: A feasibility study of MetaMap annotation of Italian medical notes. J Biomed Inform. 2016 Oct;63:22–32.

83. Alicante A, Corazza A, Isgrò F, Silvestri S. Unsupervised entity and relation extraction from clinical records in Italian. Comput Biol Med. 2016 May 1;72:263–75.

84. Esuli A, Marcheggiani D, Sebastiani F. An enhanced CRFs-based system for information extraction from radiology reports. J Biomed Inform. 2013 Jun;46(3):425–35.

85. Attardi G, Cozza V, Sartiano D. Adapting Linguistic Tools for the Analysis of Italian Medical Records. In: The First Italian Conference on Computational Linguistics CLiC-it 2014. 2014. p. 17.

86. Attardi G, Cozza V, Sartiano D. Annotation and extraction of relations from Italian medical records. In: Proceedings of the 6th Italian Information Retrieval Workshop. 2015.

87. Gerevini AE, Lavelli A, Maffi A, Maroldi R, Minard A-L, Serina I, et al. Automatic Classification of Radiological Reports for Clinical Care. In: Proceedings of AIME 2017, 16th Conference on Artificial Intelligence in Medicine. Springer, Cham; 2017. p. 149–59.

88. Bartalesi Lenzi V, Sprugnoli R. Evalita 2007: Description and Results of the TERN Task. In: Proceedings of the First International Workshop EVALITA 2007.

89. Verhagen M, Sauri R, Caselli T, Pustejovsky J. SemEval-2010 task 13: TempEval-2. In: Proceedings of the 5th international workshop on semantic evaluation. Association for Computational Linguistics; 2010. p. 57–62.

90. Caselli T, Sprugnoli R, Speranza M, Monachini M. EVENTI EValuation of Events aNd Temporal Information. In: Proceedings of the Fourth International Workshop EVALITA 2014.

91. Caselli T, Lenzi VB, Sprugnoli R, Pianta E, Prodanof I. Annotating events, temporal expressions and relations in Italian: the It-TimeML experience for the Ita-TimeBank. In: Proceedings of the 5th Linguistic Annotation Workshop. Association for Computational Linguistics; 2011. p. 143–151.

92. Manfredi G, Strötgen J, Zell J, Gertz M. HeidelTime at EVENTI: Tuning Italian Resources and Addressing TimeML's Empty Tags. Proc CLiC-It 2014. 2014 Dec;39–43.

93. Mirza P, Minard A-L. FBK-HLT-time: A complete Italian Temporal Processing System for EVENTI-Evalita 2014. In: Proceedings of the Fourth International Workshop EVALITA 2014.

94. RIS: Ricerca e innovazione nella sanità [Internet]. 2014 [cited 2017 Aug 17]. Available from: http://progetto-ris.isti.cnr.it/

95. Powsner SM, Tufte ER. Graphical summary of patient status. The Lancet. 1994 Aug 6;344(8919):386–9.

96. Pivovarov R, Elhadad N. Automated methods for the summarization of electronic health records. J Am Med Inform Assoc JAMIA. 2015 Sep;22(5):938–47.

97. Shahar Y, Goren-Bar D, Boaz D, Tahan G. Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. Artif Intell Med. 2006 Oct;38(2):115–35.

98. Were MC, Shen C, Bwana M, Emenyonu N, Musinguzi N, Nkuyahaga F, et al. Creation and evaluation of EMR-based paper clinical summaries to support HIV-care in Uganda, Africa. Int J Med Inf. 2010 Feb;79(2):90–6.

99. Liu H, Friedman C. CliniViewer: a tool for viewing electronic medical records based on natural language processing and XML. Stud Health Technol Inform. 2004;107(Pt 1):639–43.

100. Bashyam V, Hsu W, Watt E, Bui AAT, Kangarloo H, Taira RK. Problem-centric Organization and Visualization of Patient Imaging and Clinical Data. RadioGraphics. 2009 Mar 1;29(2):331–43.

101. Hirsch JS, Tanenbaum JS, Lipsky Gorman S, Liu C, Schmitz E, Hashorva D, et al. HARVEST, a longitudinal patient record summarizer. J Am Med Inform Assoc JAMIA. 2015 Mar;22(2):263–74.

102. INHERITED ARRHYTHMIAS DATABASE [Internet]. [cited 2017 Aug 18]. Available from: http://triad.fsm.it/cardmoc/

103. Verhagen M. Temporal closure in an annotation environment. In: Language Resources and Evaluation, Number 39. 2005. p. 211–241.

104. Chen W-T, Styler W. Anafora: A Web-based General Purpose Annotation Tool. In: Proceedings of the NAACL HLT 2013. 2013. p. 14–19.

105. Unified Medical Language System (UMLS) [Internet]. [cited 2017 Jan 7]. Available from: https://www.nlm.nih.gov/research/umls/

106. Harkema H, Dowling JN, Thornblade T, Chapman WW. Context: An Algorithm for Determining Negation, Experiencer, and Temporal Status from Clinical Reports. J Biomed Inform. 2009 Oct;42(5):839–51.

107. FederFarma [Internet]. [cited 2017 Jan 7]. Available from: https://www.federfarma.it/

108. cTAKES 3.2 - Fast Dictionary Lookup - Apache cTAKES - Apache Software Foundation [Internet]. [cited 2017 Aug 27]. Available from: https://cwiki.apache.org/confluence/display/CTAKES/cTAKES+3.2+-+Fast+Dictionary+Lookup

109. Goodfellow I, Bengio Y, Courville A. Chapter 10: Sequence Modeling: Recurrent and Recursive Nets. In: Deep Learning. p. 321–65.

110. Lyding V, Stemle E, Borghetti C, Brunello M, Castagnoli S, Dell'Orletta F, et al. The PAISA corpus of italian web texts. In: Proceedings of the WaC-9 Workshop. 2014. p. 36–43.

111. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed Representations of Words and Phrases and their Compositionality. ArXiv13104546 Cs Stat. 2013 Oct 16;

112. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw. 1994;5(2):157–166.

113. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997 Nov;9(8):1735–80.

114. Cho K, van Merrienboer B, Bahdanau D, Bengio Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv:14091259. 2014 Sep 3;

115. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv14123555 Cs. 2014 Dec 11;

116. Keras: The Python Deep Learning library [Internet]. [cited 2017 Aug 29]. Available from: https://keras.io/

117. Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol TIST. 2011;2(3):27.

118. Chapman WW, Hillert D, Velupillai S, Kvist M, Skeppstedt M, Chapman BE, et al. Extending the NegEx lexicon for multiple languages. Stud Health Technol Inform. 2013;192:677–81.

119. Protégé [Internet]. [cited 2017 Jan 7]. Available from: http://protege.stanford.edu/

120. D'Souza J, Ng V. Classifying Temporal Relations with Rich Linguistic Knowledge. In: HLT-NAACL. 2013. p. 918–927.

121. D'Souza J, Ng V. Knowledge-rich temporal relation identification and classification in clinical notes. Database J Biol Databases Curation. 2014 Nov 19;2014.

122. Mirza P, Tonelli S. CATENA: CAusal and TEmporal relation extraction from NAtural language texts. In: The 26th International Conference on Computational Linguistics. 2016. p. 64–75.

123. Litkowski K, Hargraves O. Coverage and inheritance in the preposition project. In: Proceedings of the third ACL-SIGSEM workshop on prepositions. Association for Computational Linguistics; 2006. p. 37–44.

124. Knight Lab. Timeline JS [Internet]. [cited 2017 Mar 23]. Available from: https://timeline.knightlab.com/

125. Athavale V, Bharadwaj S, Pamecha M, Prabhu A, Shrivastava M. Towards Deep Learning in Hindi NER: An approach to tackle the Labelled Data Scarcity. In: Proc of the 13th Intl Conference on Natural Language Process.

126. Liu K, Hogan WR, Crowley RS. Natural Language Processing methods and systems for biomedical ontology learning. J Biomed Inform. 2011 Feb;44(1):163–79.

127. Hoxha J, Jiang G, Weng C. Automated learning of domain taxonomies from text using background knowledge. J Biomed Inform. 2016 Oct;63:295–306.

128. Hamon T, Grabar N. Tuning HeidelTime for identifying time expressions in clinical texts in English and French. EACL 2014. 2014;101–105.

129. Ministero della Salute. Linee guida tracciabilità, raccolta, trasporto, conservazione e archiviazione di cellule e tessuti per indagini diagnostiche di anatomia patologica [Internet]. 2015 [cited 2017 Oct 6]. Available from: http://www.salute.gov.it/portale/documentazione/p6_2_2_1.jsp?lingua=italiano&id=2369

130. Stages of Breast Cancer | Breastcancer.org [Internet]. [cited 2017 Oct 6]. Available from: http://www.breastcancer.org/symptoms/diagnosis/staging

131. NCI Dictionaries [Internet]. National Cancer Institute. [cited 2017 Oct 6]. Available from: https://www.cancer.gov/publications/dictionaries

132. Kohane IS, Churchill SE, Murphy SN. A translational engine at the national scale: informatics for integrating biology and the bedside. J Am Med Inform Assoc JAMIA. 2012 Apr;19(2):181–5.

133. Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, Churchill S, et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc JAMIA. 2010 Apr;17(2):124–30.

134. Segagni D, Tibollo V, Dagliati A, Napolitano C, G Priori S, Bellazzi R. CARDIO-i2b2: integrating arrhythmogenic disease data in i2b2. Stud Health Technol Inform. 2012;180:1126–8.