ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

TESI DI LAUREA MAGISTRALE

in

Data Mining

Information Usefulness To Support Dialogue Management in Healthcare

CANDIDATO:
Marco Lanciotti

RELATORE:
Prof. Claudio Sartori

CORRELATORI:
Prof. Célia da Costa Pereira
Prof. Catherine Escazut

Anno accademico 2020-2021

Alla mia famiglia che mi ha concesso l'opportunità di essere quel che sono oggi.

Alla mia curiosità e alla mia voglia di osare che saranno sempre fedeli compagne nella mia vita.

Contents

Introduction

In this work it will be proposed a new agent-based model to help the information exchange in the healthcare domain. More precisely, the main objective of our framework, to be used during the physician-patient interaction, is to help the physician obtain information about the patient's condition in order to reach the *correct* diagnosis as soon as possible.

However, to the best of our knowledge, no contribution has focused on how to reduce the duration of the physician-patient interaction. Indeed, the amount of time needed for interaction (to establish a diagnosis) between the patient and the physician can be considered as a determining element of the quality of a framework.
One possibility to obtain a decrease in this duration, is to help the physician ask fewer questions, for example, by asking the questions about the most useful symptoms i.e., the ones that most help the physician to make a decision (a diagnosis about a disease or the conclusion that the patient is not ill).
A contribution which can be considered as one step into that direction is the one by Teixeira *et al.* [1], who presented a goal-based framework that supports the development of intelligent conversational agents within the healthcare domain.
Indeed, they discussed the concept of *information usefulness* in a health dialogue and formalized the metrics that allow the calculation of the usefulness value of a symptom to be asked for, i.e., a symptom

the physician does not yet know whether the patient is suffering from or not.
However, Teixeira's contribution was focused on the evaluation of the metric, and they did not implement the fact that some symptoms may imply another symptom even if they have mentioned this possibility in their paper, and they did not considered the fact that a symptom may be more or less certain, i.e., they considered binary values for setting the weights of the symptoms.

The aim of this thesis is threefold: to extend Teixeira's *et al.*'s framework by making it possible to also consider and implement (i) the implications among the symptoms in the reasoning process, and (ii) non-binary values for the weights associated to the symptoms. The third aim is (iii) to propose a filtering of the dataset in order to reduce the time needed to treat the data and finally (iv) it was added a new functionality to the framework: the ability to detect further future risks of a patient already knowing his pathology.
More precisely, the specificity of the work here proposed is both (i) to create a model able to help a physician in choosing the best next action to perform, step by step during the interaction with the patient, and (ii) to reduce the duration of the inquiry process as well as its dimensionality, in order to identify the disease with as few interactions and questions as possible.

The idea of filtering the dataset has been inspired by some contributions in the literature, like the work by Krieger and colleagues [2], in which they examined the influence of gender and ethnicity, for knowledge about health and for the provision of healthcare. In their work, they underlined that the data they have used "indicate that white men and women generally have the best health and that men and women, within each ethnic group, have different patterns of disease".

They also emphasized that "the health status of men and women differs for conditions related to reproduction, but it differs for many non reproductive conditions as well". Other studies, for example, the one by Deek and colleagues [3], show that "gender, age and sociocultural factors are likely to influence health-related behavior, including screening".

The merits of our proposals have been demonstrated by the performance and effectiveness obtained after the execution of different test cases on a set of patients, each affected by some initial symptoms. For this purpose, we exploited a dataset of diseases that we have previously constructed from the Symcat dataset [4]: a generic disease is considered as a composition of related *uncertain* symptoms, each of which being then associated with a weight (a value between 0 and 1) indicating its relevance to the considered disease.
For our experiments, we have considered, on the one hand, the complete dataset obtained and, on the other hand, the dataset obtained after filtering with respect to the relevant features.The results showed that the use of the filtered data on the extended frame yielded similar results compared to the use of the full dataset, in terms of diagnostic efficiency.
In addition, a significant improvement in performance was observed with a reduction in the duration of the interaction between the physician and the patient.

The thesis is structured as follows.
Chapter 1 describes the initial framework we started with and the way we extended it.
Chapter 2 illustrates the development of the framework with particular attention to how the datasets used in the experiments were built, *Section 2.2*, and the implementation of the proposed framework

in *Section 2.3*.

Chapter 3 shows the first results obtained following a first implementation of the framework.

Chapter 4 explains the arrangements made in order to achieve further improvements in our framework.

Chapter 5 introduces the encouraging and final results obtained.

Chapter 1

Automatic Disease Diagnosis

> Breaking an old business model is always going to require leaders to follow their instinct. There will always be persuasive reasons not to take a risk. But if you only do what worked in the past, you will wake up one day and find that you've been passed by.

I decided to begin this chapter by referring to this well-known quote by Clayton Christensen[5], who emphasized that, in a business model, companies must adapt to the socio-cultural and technological change of the historical context in which they find themselves.
Cristensen, referring to a business model, tackles a current issue also in a healthcare system, which must be able to evolve according to the needs of patients and doctors and to take full advantage of technological innovations.
In the last few years, the introduction of technology in the medical field is making it possible to solve a well-known problem: the dilemma of the iron triangle of health care, introduced by William Kissick in [6]. The name derives from its triangle-shaped representation where each side reflects three critical problems: quality, access and cost.
In particular, these 3 aspects compete with each other: the improvement in one area results in a decline in at least one of the others.

Although the improvement of the healthcare system turns out to be trapped in this rigid structure, the introduction of disruptive technologies such as mobile Internet and big data analytics seem to be a solution to this issue that allows an increase in quality and accessibility and at the same time a reduction in costs.

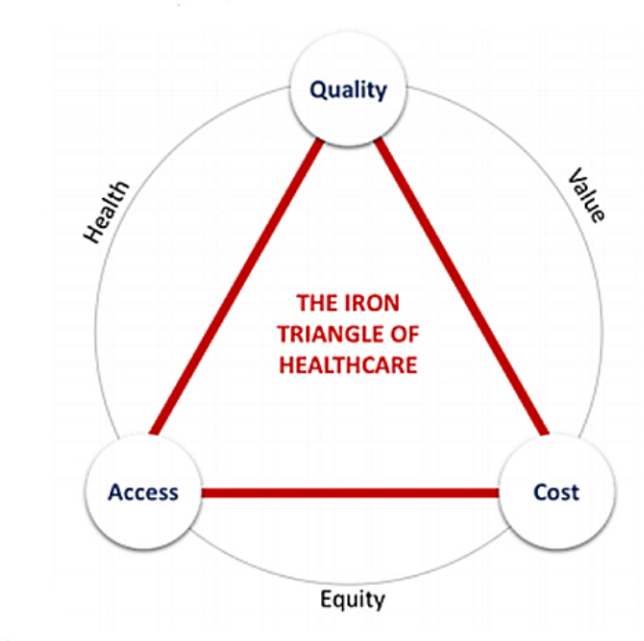

Figure 1.1: The Iron Triangle of Healthcare

Nowadays, thanks to the amount of information available online, people are able to self-diagnose their health. According to a survey in 2012 [7], 35% of U.S. adults attempted to diagnose their ailments through online services.

A self-diagnosis process usually begins with an online search for a known symptom from which a patient feels affected. However, a simple online search can lead to the deterioration of one of those aspects already discussed above, namely the quality of the information that is obtained.

In fact, if from a certain point of view, the possibility of exploiting the web allows a faster accessibility to medical data, on the other hand this leads to low quality information that can lead to irrelevant,

inaccurate or even incorrect results. To facilitate self-diagnosis of a patient while maintaining a sustainable level of quality, the concept of *symptom checking* was introduced.

1.1 State of the art

Making a *medical diagnosis* is to identify the nature of a disease by examining the symptoms of the patient. The Butterworths Medical Dictionary [8] defines the word *diagnosis* as "the art of applying scientific methods to the elucidation of the problems presented by a sick patient". Ledley *et al.* [9] proposed one of the first, and probably one of the most prominent papers formalizing the reasoning of medical diagnosis.

The framework proposed by the authors includes three fundamental components necessary to automate the decision on a medical diagnosis: (i) *medical knowledge* (a set of data linking diseases to their symptoms), (ii) the *list of symptoms* presented by the patient, and (iii) the *diagnosis* itself (a list of possible diseases, each of which is associated with its probability of occurrence).
These processes characterizing the elaboration of a diagnosis can be identified with the term symptom checking and consequently the agent capable of elaborating such prognosis is identifiable as symptom checker.

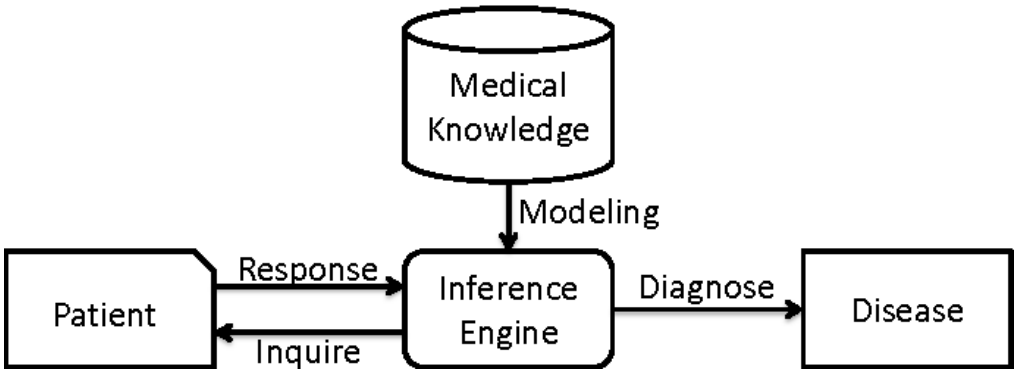

Figure 1.2: The logic components of symptom checking.

A symptom checker, described in Fig.1.2, is composed of two logical

components: *a database of medical knowledge* and *an inference engine*. The database represents the relationship between symptoms and disease, while the inference engine must take care of all the aspects that characterize the conversation with the patient. First of all it must ensure the proper execution of the process of investigation with the patient formulating the questions correctly, then, must be able to select and collect all the useful information so as to exploit them, in combination with your own database, to make a diagnosis.
At this point the symptom checker will assess the outcome of the prediction: if the confidence value is sufficiently high, the process of processing the diagnosis will end successfully by issuing the patient a list of potential diseases that the patient may have, on the other hand, if the confidence is too low, it may suggest that clinical trials be carried out to facilitate diagnosis.

Despite the contribution of Ledley *and al.* in the elaboration of a first symptom checker goes back 60 years, the authors had already since then the awareness that the use of computers could change the medical domain and in particular how can help both physicians (with the collection of clinical information and the use of old diagnoses for prediction) and medical students in learning the methods.
After the remarks of Ledley *and al.*, several scholars shifted their attention no longer to the analysis of the patients, but on how to support doctors in the work. From this current, Fieschi and Gouvernet [10] analyzing the results described in the Ledley *and al.*'s paper confirmed the goodness and the usefulness of their framework and proposed more integration of mathematical methods that could "enrich this art":

> As soon as computers were available to physicians, the question of arriving at a diagnosis using automatic methods became a major research topic.

While *Fieschi and Gouvernet* proposed the introduction of a mathematical approach to making diagnoses, Paul *and al.*, for example, went further and introduced a framework focused on the therapeutic aspects of a clinical problem. The latter was intended to suggest to doctors whether it was convenient or not to treat a patient on the basis that the patient may or may not have a certain disease [11]. The final objective of their framework was to manage the dilemma of uncertainty about the patient's illness in order to make the "correct" decision. According to the authors, in the presence of uncertainty, the administration of a treatment known to be effective for the disease under consideration will be beneficial if the disease is actually present, but may be harmful if the disease is absent. Another contribution of that work was to develop a method for calculating a therapeutic threshold such that if the probability of the disease in a given patient exceeds the threshold, it is preferable to treat the patient, and if it is below the threshold, it is preferable to withhold treatment.
The work of Ledley *and al.* has given therefore the way to a real revolution in the world of automatic diagnosis leading to the elaboration of different types of frameworks able to support patients and doctors in their investigations.
Nowadays, for example, it is quite common to find specialized symptom checkers that are able to make predictions. One of the main goals that a modern symptom checker must be able to guarantee is to achieve a high disease-prediction accuracy.

Starting from this consideration, Tang *and al.* proposed in their work [12] a new type of symptom checker that was able to achieve markedly higher disease-prediction accuracy when compared to a traditional approach.
The proposed new model consists of several small anatomical models that are responsible for different anatomical parts.

Unlike previous work in which different approaches were used for the choice of symptoms, in their work Tang *and al.* opted for the use of a *Reifoncerment Learning (RL)* [13] framework able to manage the process of investigation and diagnosis as Markovian processes.
In addition, in order to make the framework as realistic as possible, the authors decided to represent the doctors, each specialized in a predetermined medical domain, training a model for each anatomical part of the body; the combination of these models gave rise to an *ensemble model*[14].
The use of different models is also reflected in the functioning of the symptom checker: at the beginning it suggests the user to select the anatomical part of interest (e.g., selecting abdomen for abdominal pain or head for headache) so that the associated model can proceed with the analysis and then make the diagnosis. The quality of this work was then justified by the results obtained, which testify to the improvement in accuracy obtained compared to previous works. So much so that on a set of 73 diseases has reached an accuracy of 48%, which is much higher than the average of 34% reached by the classic online services presented in [7].
Despite the encouraging results shown, this proposed framework had two major limitations:

1. the lack of some relevant information about the patient useful to achieve a better prediction of the disease;

2. the incapacity of the framework to manage diseases whose symptoms could occur in different parts of patient's body.

With regard to the first limitation, the authors themselves realized that they could not guarantee users a high degree of reliability since in their project they could not refer to crucial laboratory tests such as blood tests and vital signs required to obtain more accurate pre-

dictions.
Thus, it is more realistic that a symptom checker would suggest a small number of possible diseases and then refer the patient to see relevant doctors to order lab tests and to follow up.

Tang himself *and al.* proposed in [15] to put a context in the model and turn it into a context aware model. The context of information concerned mainly three aspects of a patient: *who, when* and *where.*
The **who** aspect concerned a person's demographic information (e.g., age and gender), his heredity understood as the set of his genetic data and his clinical history; the **when** aspect, on the other hand, represented the distribution of diseases over the time of the year and finally the **where** aspect characterized by a distribution of diseases from coarse to fine location granularities(e.g., by country,city, and/or neighborhood).
In addition, in the same project, the authors also addressed the second limitation found by introducing a further improvement to improve the accuracy of diagnosis. They decided to use a *Hierarchical Reinforcement Learning (HRL)* approach, described in [16], introducing an additional latent layer , the **main agent**, which is able to coordinate the underlying models each of which specialized in a different anatomical part.
Unlike the previous work in which, depending on the symptom proposed by the user, a specific model was assigned for the entire investigation process, the introduction of the master model allows to select the specific anatomical part on which to perform the investigation process at each interaction with the framework.
This allows to solve those cases in which the disease presenting itself in different parts of the body continuously requires to interact with different models. For instance, considering the allergy, this one might show up in different areas; the framework structured in specialized

models was not able to understand the right path to follow in that case.
The application of these improvements on a simulated dataset showed a drastic improvement in terms of disease prediction accuracy by obtaining, compared to the previous work, an increase of about 10% in a set of 50 diseases and about 5% in a set of 100 diseases.

The promising results obtained in the implementation of these task-oriented dialogue systems prompted other researchers to follow the guidelines drawn trying to understand where to intervene to achieve further improvements.
In this context, the work of Liao *and al.* [17] took up the idea of using an *Hierarchical Reinforcement Learning* approach in which a master agent is responsible for activating a specific model in the level below. However, they presented a set of underlying models, called 'workers', conceived differently from previous projects. If until then each underlying model specialized in a predetermined anatomical part, from now on, the models refer to a particular set of related diseases.
In general, a particular disease is related to a certain group of symptoms. That's to say, a person who suffers a disease will often carries some corresponding symptoms at the same time.
In other word, each disease has a group of corresponding symptoms and the overlap among different groups of symptoms are limited. This motivates us to classify diseases into different groups following the setting of departments in the hospital and design a hierarchical structure for symptom acquisition and disease diagnosis.
So this framework will present a master agent that will activate the specific model at the level below, each worker will be responsible for inquiring the patient about symptoms related to a certain group of diseases and finally there will be a disease classifier who will be responsible for making the final diagnosis based on the information collected

by the workers.

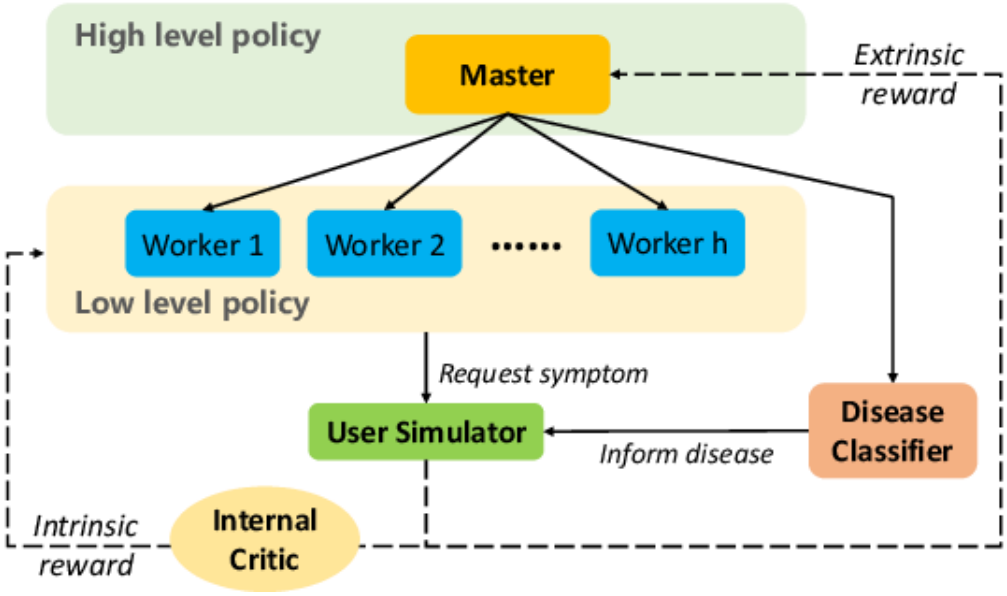

Figure 1.3: The framework of our hierarchical reinforcement learning model with two-layer policies.

1.2 The Proposed Framework: Background

1.2.1 Motivations

The works proposed above show a constant improvement over the years in the realization of increasingly more reliable and accurate symptomatic controllers. However, to the best of our knowledge, no contributions focused on how to reduce the duration of doctor-patient interaction.

Consequently, this problem leads to the second requirement for an effective symptom checker, namely good user experience. Indeed, the amount of time needed for the interaction (to establish a diagnosis) between patient and doctor can be considered a crucial element for the quality of a framework.

Let's consider, for example, the situation of a patient who wants to use a symptom checker to make a self-diagnosis of his clinical condition. When this latter interfaces with the framework, he expects to get a

response without having to come across a process of investigation too long and inquisitorial.

This leads to the definition of the design goal, namely being able to maximize the gain of information when only a limited number of symptomatic queries can be made in order to achieve high diagnostic accuracy.

One possibility to obtain a decrease in this duration and and thus improve user experience is to help the physician ask fewer questions, for example, by asking the questions about the most useful symptoms i.e., the ones that most help the physician to make a decision (a diagnosis about a disease or the conclusion that the patient is not ill).

A contribution which can be considered as one step into that direction is the one by Teixeira *et al.* [1], who presented a goal-based framework that supports the development of intelligent conversational agents within the healthcare domain. Indeed, they discussed the concept of *information usefulness* in a health dialogue and formalized the metrics that allow the calculation of the usefulness value of a symptom to be asked for, i.e., a symptom the physician does not yet know whether the patient is suffering from or not.

1.2.2 Preliminaries

Disease	Symptoms											
	Abdominal pain (s_1)	Pruritus (s_2)	Respiratory distress (s_3)	Fever (s_4)	Early awakening (s_5)	Nausea (s_6)	Incontinence (s_7)	Shortness of breath (s_8)	Pain (s_9)	Diarrhea (s_{10})	Asthenia (s_{11})	Yellow sputum (s_{12})
Hepatitis (g_1)	0.36	0.28	0.20	0.16	-	-	-	-	-	-	-	-
Colitis (g_2)	0.09	-	-	0.30	0.26	0.22	0.13	-	-	-	-	-
Tricuspid insufficiency (g_3)	0.11	-	-	0.05	-	0.16	-	0.42	0.26	-	-	-
Kidney failure (g_4)	-	-	-	0.13	-	-	-	0.35	-	0.26	0.22	0.04

Figure 1.4: Disease/Symptom Domain Knowledge

The framework suggested by Teixeira *et al.* has been proposed to aid in the management of a slot filling during a patient-physician interaction for the goal of determining the patient's disease among established disease classifications [1]. Each slot is handled as a dialogue action

that corresponds to a query that is used to get the slot's value (confirming or disregarding it).
The main goal, as in Teixeira's contribution, is to provide a technique for the dialogue manager (DM) to choose what conversation action (symptom) should be examined (questioned) next.
Let's consider a propositional language L of which a subset, L_G, is the language used to represent the *rules* associated with the goals (representing the class of diseases).

Definition 1.1. A rule $r \in R_a$ allows to classify a patient into a class g based on the information collected by an agent a. A rule can be represented as follows: $s_1 \oplus s_2 \oplus \ldots \oplus s_{n_g} \Rightarrow g$, where s_i represents the ith slot (symptom, test results, etc.) related to g and n_g is the number of slots related to g.

Definition 1.2. An information unit s (called 'slot') is a relevant information, increasing the knowledge of the belief base B_a necessary for achieving a goal.

The relevant information considered may be of different types: (i) information directly requested from the patient; (ii) information acquired through sensors; (iii) information acquired from the patient's clinical history: electronic file and data provided by the patient himself; (iv) information acquired through external services.
However, only the first class of information submitted was considered relevant for the implementation of the proposed framework.
The value of the slot is fixed initially on *unknown*; if, during the process of investigation with the patient, information is acquired about this slot its value may change to true or false depending on the input received.

Definition 1.3. An agent a is a discrete entity aiming to classify a patient with respect to a set of classes. An agent a has its own belief

base B_a and a set of goals G_a.

The *dialogue agent* a is expected to be aware of all of the physician's beliefs/knowledge, as well as the goals – a finite set of positive literals from the language of potential goals LG – and the patient's responses. For instance, in the context of diagnosis, the conversation agent's aim is to determine which of the several classes of diseases or goals (g_1, g_2, ..., g_n) corresponds to the patient's state.
The degree to which an objective is deemed to be attained, i.e., we may assume that a diagnosis for the patient has been obtained, is a progressive notion, as described in [1]. Of course, only the physician is able to decide to which extent exactly the diagnosis is established. This is why it is recommended to use threshold (τ) determined by the physician to determine whether the result is positive (i.e., the patient may have the disease) or negative (the patient may not have the disease).
Moreover, the dialogue agent a has a belief base B_a, which is divided into two subsets: B_a^m and B_a^g. B_a^m is the set of formulas from $L \setminus L_G$ that represents a's beliefs about the slot-values, e.g., the patient has *pruritus* (s_2), *fever* (s_4), *nausea* (s_6), *incontinence* (s_7), etc. B_a^m may also contain other physician-specific information, such as the co-occurrence of slots, e.g., *Abdominal Pain* implies *Pain* ($s_1 \rightarrow s_9$). This implies that, if *Abdominal Pain* is mentioned as one of the patient's symptoms during the conversation, B_a is immediately updated with both the values for the slots *Abdominal Pain* and *Pain*.
B_a^g instead, contains as many rules of the form $s_1 \oplus s_2 \oplus \ldots \oplus s_{n_g} \Rightarrow g$ (as stated in Definition 1.1), where each s_i is a positive literal of $L \setminus L_G$, representing a slot that influences the disease g, with $g \in G_a$.

Such rules express a's views about what information is required to determine a class and, as a result, achieve a certain goal. The slot-values acquired from the patient's responses to the conversation agent's queries

are examples of such pieces of information.
Assume that throughout a conversation with a patient, the dialogue agent gathers more and more information about the patient's current condition, eventually leading to the categorization of the patient into one or more classes. The information units that are still lacking can be expressed in the following way.

Definition 1.4 (Missing Information)**.** Let a be a dialogue agent with its belief base B_a and its goal set G_a. Let $g \in G_a$ be such that $B_a \not\models g$.[1] The missing information for goal g, $Missing(B_a, g)$, is defined as follows:

$$Missing(B_a, g) = \{l : l \in P(g) \text{ and } B_a \not\models l\} \tag{1.1}$$

$Missing(B_a, g)$ is the set of all the slots in the premise of g which cannot be deduced from B_a (i.e., which are not yet believed by the agent and therefore the dialogue agent should ask the patient about them).

Remark 1. In the particular case in which $B_a^m = \emptyset$, $Missing(B_a, g) = P(g)$, i.e., the missing piece of information to achieve g is $P(g)$.

Different goals may have missing information in common, we thus introduce the notion of *multiset of missing information.*

Definition 1.5 (Multiset of missing information)**.** Let a be a dialogue agent whose belief base is B_a and whose goal set is G_a. The multiset[2] of missing information to achieve the goals in G_a is:

$$Missing(B_a, G_a) = \bigcup_{k=1}^{|G_a|} Missing(B_a, g_k) \tag{1.2}$$

[1] In propositional logic, $\phi \models \psi$ means that ψ is a logical consequence of ϕ. Here, it means that we can classify the patient as having disease g from what we already know/believe ($B_a \models g$, see Def. 1.12).

[2] Reminder: a multiset is a set whose elements can have several occurrences, such as $\{p, q, p\}$.

with $\bigcup$ representing the union on multisets, $||$ representing the cardinality of a set.

Let us consider the following definitions:

Definition 1.6. The premise set $P(g)$ is the set of all the slots that help classifying goal g, i.e., $P(g) = \{s|s \in lhs(g)\}$, where "lhs" is the left hand side of a rule $r \in R_a$.

Definition 1.7. Let $G(s)$ be the set of goals related to slot s. We define $G(s) = \{g \in G_a|s \in P(g)\}$.

Definition 1.8. Let $W(g)$ be the set of all the weights related to the slots associated to goal g. We define $W(g) = \{w(s_i, g)|s_i \in P(g)\}$, where $w(s, g)$ is the association between goal g with Slot s

Definition 1.9. Let g_j be a goal and s_i be a slot. The association between g_j and s_i is represented through a real value $w_{ij} \in [0, 1]$, called **weight**. The weight represents how relevant the information contained in the slot s_i is for achieving the goal g_j.

When defining a slot s_i associated with a certain weight wij, it should be noted that since the same slot may appear in more than one rule r its weight is variable. The information of that unit is shared by more goals that consider its relevance, the "weight", differently.
In 1.9 a weight has been defined as a real value $w_{ij} \in [0, 1]$, this leads to the definition of a further project constraint: given the j_{th} goal, it holds the hypothesis

$$\sum_{i=0}^{n_{gj}} w_{ij} = 1 \tag{1.3}$$

where n_{gj} is the number of slots associated with the goal gj.

The following are the justifications for this constraint. First, we need that the agent be "confident" that the patient belongs to a certain

goal if all of the slots connected with that goal are set to true. Second, a patient's categorization with respect to a goal is accomplished by comparing an agent's confidence in the likelihood that a patient belongs to that goal to a threshold. As a result, in order to execute this comparison properly, the numerical confidence limits for all goals must be the same.

Now we need to define the overall importance of requiring an answer concerning slot s, $N_1(s)$, (which represents the extent to which s ($\neg s$) would help getting closer to a classification), with respect to all the goals $g \in G_a$ as follows:

$$N_1(s) = \left\{ \begin{array}{l} \sum_{g \in G(s)} w(s,g) + \\ \quad \sum_{s' | B_a \cup s \models s' \wedge B_a \not\models s'} w(s',g) \quad \text{if } s = true \\ \\ \sum_{g \in G(s)} w(s,g) + \\ \quad \sum_{s' | B_a \cup s' \models s \wedge B_a \not\models s} w(s',g) \quad \text{if } s = false \end{array} \right\} \tag{1.4}$$

where $w(s',g) = 0$ if $s' \notin P(g)$ and s comes from the multiset of missing information (Def. 1.5).
It can be observed that $N_1(s)$ is the gradual definition of the N_1 component proposed in [18].

We can also compute the overall weight, $N_2(s)$, that concerns the slots which are still missing after receiving the value of slot s as follows:

$$N_2(s) = (\sum_{g \in G_a} \sum_{k \in W(g)} k) - N_1(s). \tag{1.5}$$

We can notice that our definition of $N_2(s)$ is a generalization (a gradual counterpart) of the one proposed in [18].
To characterize a goal as being achieved or not, we need to know the amount of already known information about the slots related to it and

the amount of information which is still missing.

Definition 1.10. The amount of information that the agent a collected about the goal g, $\mathcal{C}(g)$ is given by the union of:

$$\mathcal{C}(g) = \sum_{(s|B_a \models s) \wedge (s \in P(g))} w(s, g) \tag{1.6}$$

representing the information slot set to true, and:

$$\mathcal{C}(\neg g) = \sum_{(s|B_a \models \neg s) \wedge (s \in P(g))} w(s, g) \tag{1.7}$$

representing the information slot set to false.

Definition 1.11. The amount of information that the agent a still has to collect for having a complete knowledge concerning goal g, $\mathcal{M}(g)$ is given by:

$$\mathcal{M}(g) = \sum_{s' \in Missing(B_a, g)} w(s', g) \tag{1.8}$$

Let τ be a threshold which allows to characterize a goal as being achieved or not, according to the expert.

Definition 1.12. A goal g is achieved, i.e., the diagnosis can be made, noted $B_a \models g$, if and only if:

- $\mathcal{C}(g) \geq \tau$: the amount of information already available is *sufficient* to make a positive diagnosis, i.e., the patient has the disease,

  Or

- $\mathcal{C}(g) + \mathcal{M}(g) < \tau$: the amount of information already available is *sufficient* to make a negative diagnosis, i.e., the patient does not have the disease.

Obviously, the value of τ has an impact on an agent's capacity to attain objectives. High values of τ decrease the likelihood of identifying a

patient as belonging to a certain goal while increasing the likelihood of the patient not belonging to any goal. For low values of τ, the opposite happens.

Remark 2. We can notice that if there are not missing slots, i.e., $\mathcal{M}(g) = 0$, we have: $\mathcal{C}(g) + \mathcal{C}(\neg g) = 1$.

This remark shows how the total amount of information is preserved when all slots are filled.

Definition 1.13. The set of goals that a slot s allows the dialogue agent to achieve is:

$$E(s) = \{g \in G_a | B_a \cup s \models g \wedge B_a \not\models g\}. \tag{1.9}$$

Another aspect for the evaluation of a goal is the possibility of associating a degree to each goal; we will define this degree *priority*

Definition 1.14. The priority of a goal g is a real value in the interval [0,1] representing the importance degree that the goal g has within the belief Base B_a.

1.2.3 Next question selection

Teixeira *et al.* [1] proposed a function that measures the usefulness of a (not-yet-filled) slot [3]. By comparing the usefulness value of different slots, it is possible to select the best candidate for the next question for a dialogue that makes as few questions as possible to achieve a classification. The usefulness function takes into account several factors:

- **Class' priority**: a priority can be assigned for each class in domains that present critical situations class [19]. As a re-

[3] Of course, here we are interested in knowing the utility of the slots which are not a logical consequence of the belief base B_a, i.e., which cannot be deduced from previous beliefs/knowledge.

sult, whenever priority values are available, slots associated with higher priority classes have their usefulness value increased.

- **Information filled by the slot**: Some slots give more information than others, such as if they appear in several classes/diseases or if the domain knowledge reports some co-occurrence. Unlike what has been described in [1]., the method we propose here allows us to take this information into account throughout the patient-physician interaction.

- **Slot's weight**: as previously stated, not all snippets of information (slots) have the same relevance to a class depending on the domain. For example, a practitioner may say that *abdominal pain* (s_1) is more significant than *fever* (s_4) in identifying whether a patient has *Hepatitis* (g_1), therefore the latter would have a higher usefulness value for this class.
  The usefulness function also takes into account that a slot can be present in different classes with different weights. Here also, we have implemented the fact that these weights can be non binary, unlike in [1].

When we compute the usefulness value of a slot s, we need to consider what happens in both cases, i.e., if the slots s is set to 1 (the symptom occurred) or to 0 (the symptom did not occur). The resulting equation for computing the usefulness of a slot s proposed by Teixeira *et al.* is the following:

$$U(s) = \Big[\Big(|E(s)|+\frac{N_1(s)}{N_1(s)+N_2(s)}\Big)^{s=0} + \Big(|E(s)|+\frac{N_1(s)}{N_1(s)+N_2(s)}\Big)^{s=1}\Big]\frac{|O^s_{G_a|}}{|G^s_a|} \quad (1.10)$$

where:

- $|E(s)|$ is the number of goals that are satisfied thanks to the information about slot s, i.e., the number of diseases for which we can conclude if the patient belongs to them or not after checking the slot s;
- $N_1(s)$ and $N_2(s)$ are computed by taking into account the weights associated with s in all goals;
- $|O^s_{G_a}|$ is the sum of the priorities associated with the goals having the slot s in their premise set P;
- $|G^s_a|$ is the number of goals having the slot s in their premise set P.

The coefficient $\frac{|O^s_{G_a|}}{|G^s_a|}$ allows to increase the usefulness value of slots that are a premise of more important goals. The decision on when to stop exploring the domain, that is, stop acquiring information, depends on the needs of the domain and it is a role of the DM.

1.2.4 Cost

The computational cost of selecting the next question is $O(|G_a|\cdot|S|^2)$, where S is the set of symptoms. This is because, for each symptom $s \in S$, the agent has to compute $U(s)$, which in turn requires computing $N_1(s)$, whose cost is $O(|G_a|+|S|)$, and $N_2(s)$, whose cost is $O(|G_a|\cdot|S|)$.

Chapter 2

Framework Development

For the realization of the framework it has been decided to approach the problem following a classical scheme in the modeling of software products, the *Software Development Life Cycle Process*[20].
All *Software Development Life Cycle* processes are made up of a collection of discrete actions that are carried out in order to create a software product.
A software development life cycle (SDLC) describes how to design, create, maintain, and improve the efficiency of a software product. The SDLC process is a set of techniques for improving the entire quality and development process of software.
Referring to the model in Fig.2.1 that depicts the many stages of a typical software development life cycle, it was chosen to structure the development of the framework in different phases in order to simplify and organize the work in a better way.

The main phases of my project can be considered:
(i)*Planning and Analysis Phase*; (ii)*Data Preparation Phase*; (iii) *Implementation Phase*; (iv) *Improvement Phase*; (v) *Testing of the framework*

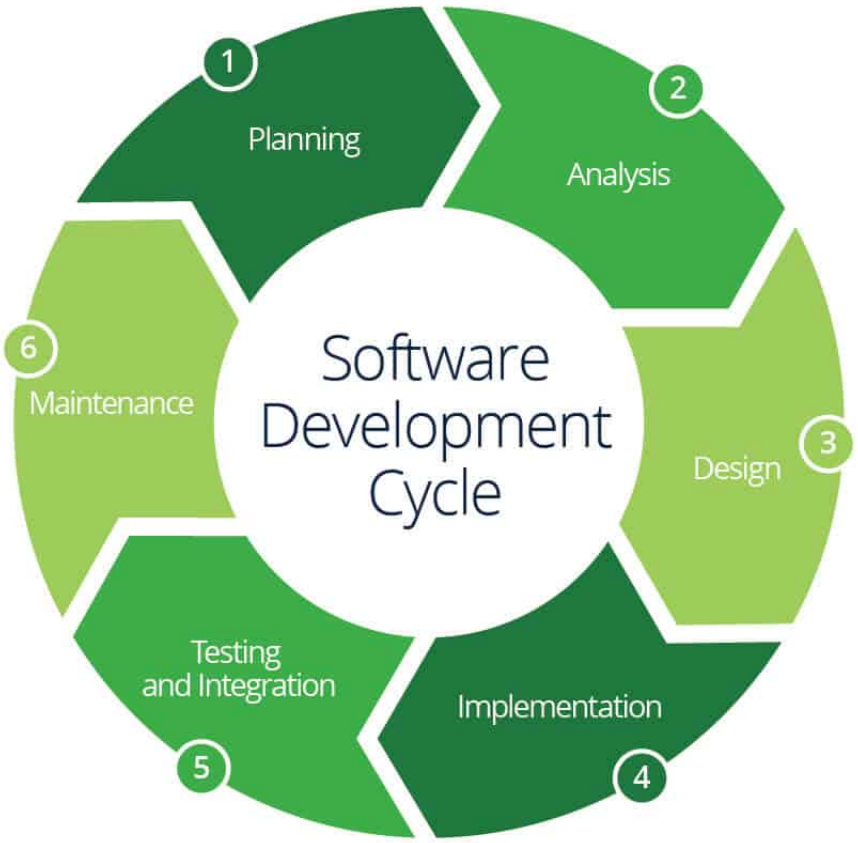

Figure 2.1: Software development life cycle

Moreover in the realization of the framework it has been decided to use not a classic approach *Waterfall*, in which the output of a previous phase is binding for the execution of a next step, but a more flexible approach like the *Agile* method in which the modeling of system requirements can vary during development.[21]
In this way the realization of the project is not seen in a sequential way in which every phase is considered 'black box' for the next phase, but the various parts of the project are elaborated in parallel and every phase is re-examined during the life of the software several times. This leds to the definition of a first implementation of the framework followed by a first phase of testing and then there was a subsequent review of the objectives with the development of new improvements in the framework and new results.

2.1 Planning and Analysis Phase

The first phase was the understanding of the project requests and therefore the long-term work planning in order to achieve the required objectives.
As mentioned, the main purpose of this framework is to create a symptom checker that is able to support the work of doctors during consultation with patients by providing the diseases with the highest probability of occurrence in the patient examined.
The novelty focuses on reducing the number of interactions with the patient and therefore making the best choice of the next question to be posed to the patient.
In order to achieve this goal, the first step was to search for a dataset similar to our objectives, analyze the useful information and model it according to the requests and project constraints

2.1.1 Dataset Description

I replicated the data pertaining to the patients, as described in [15], using Symcat's symptom-disease database [4], which contains 801 illnesses and 378 symptoms.
Each disease is defined by its symptoms, each of which is assigned a numerical value that indicates how important it is to the condition. This dataset includes a description of each illness and its symptoms, as well as the appropriate tests to use with those descriptions and the odds that a disease will arise based on age, ethnicity, and gender.

2.2 Data Preparation Phase

The analysis of the dataset was carried out during the analysis phase, assessing the relevant information and examining the superflu-

ous information. The next step is to make the data within our dataset as similar as possible and to confirm the needs of the problem; this step is called *Data-Preprocessing*
In any Machine Learning process, *Data Preprocessing* is that stage where data are processed, or encoded, to bring it to such a state that now the machine can easily analyze it. In other words, the characteristics of the data can now be easily interpreted by the algorithm.[22]

2.2.1 Splitting Dataset

Starting from the unique dataset described in 2.1.1, it was preferred to split the relevant information into smaller datasets each with a focus on a particular aspect eliminating superfluous information.
In this way the following datasets were defined at the beginning of the project:

- **weights.csv**: a dataset containing the set of diseases, each of which is described by a specific set of symptoms. Each symptom, which may appear in different diseases, has different relevance, *weight*, depending on the kind of disease it is associated with.

- **symptomsWeightAge.csv**: it is an extension of the *weights.csv* dataset in which each symptom is described by a percentage, positive or negative, value that expresses the variation in weight of the symptom depending on the age group and the disease it describes.

  Suppose you have a symptom associated with a specific disease and age range described by a positive percentage value +15% for instance; this indicates that the value of basic relevance described in *weights.csv* must be increased by 15% and vice-versa a negative value expresses a decrease of that weight.

  Therefore this dataset does not contain the weights of the symptoms already computed, but it gives us the useful information

to apply to *weights.csv* to obtain a dataset in which each disease is described from a set of symptoms with different weights depending on the age group considered.

- **symptomsRelation.csv**: a set of data in which a list of related symptoms has been defined for each symptom. In this case it was decided to consider a symptom related to another symptom only if they occur at least in a same disease.

- **genderSymtptom.csv**: in this dataset the symptoms were divided according to gender: male or female. Specifically, only those gender-specific symptoms were considered.

- **df_test.csv**: a dataset in which each disease has been associated with the most useful tests to perform to verify its diagnosis. Each test is also described by a percentage value that describes its effectiveness in correlation with that disease.

- **diseaseDataframe.csv**: definition of a set of data in which each disease is described by a set of features characterizing the probability of occurrence in patients. In particular, each of these attributes describes, through a certain percentage value, which are the categories in terms of age, gender and ethnicity more at risk for a specific disease.
  Just as the relevance of a symptom may vary depending on the age of a patient - *symptomsWeightAge.csv* -, in the same way a given disease assumes a higher risk factor on certain subjects rather than on others.

2.2.2 Normalization symptoms-weights dataset

The first problem that has been raised during the analysis of the data was their normalization; in fact, their modeling and representation were not conforming to the project constraints.

In the first dataset presented, *weights.csv*, each symptom was associated with a certain weight whose value can range from a minimum of 0 (i.e. it has no relevance to a given disease) or a fixed maximum depending on the disease considered. It has been observed that in one or more cases some symptoms for specific diseases reached values higher than 5.
However, a range of values for symptom weights in a float interval $[0, 5]$ opposes one of the project constraints described in 1.9, in which it is mentioned that the sum of all symptoms for each disease must be equal to 1.
For example, suppose d to be a known disease and
$S = \{s_1 : w_1, s_2 : w_2, s_3 : w_3, s_4 : w_4, s_5 : w_5\}$ the its list of symptoms with attached the weights $w_1...w_5$, we would like two key conditions to be met:

- The relevance of each symptom must belong to [0,1];
- The sum of all symptom weights must belong to [0,1];

To achieve this result, a normalization process was carried out on each set of symptoms for each specific disease. In order to carry out the normalization process, each set of symptoms has been considered as an array of values of which we want to satisfy the rules previously described.
In this case, after removing symptoms with relevance 0, the concept of *Manhattan norm*, called also *L1 norm*,[1] was exploited and dividing every value inside the array by the norm of the vector, we were able to re-calibrate the weights in a range [0,1] and to normalize the norm of the vector.
This process has been repeated iteratively for each disease within the dataset.

[1]It is calculated as the sum of the absolute vector values :$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$.

2.2.3 Correlation between symptoms

As described in section 1.2.1, the main purpose of our project is to achieve a reduction in the time needed for the information acquisition process and consequently a decrease in the number of interactions with the patient.
In this part we will demonstrate how to improve the performance of the framework by proposing a new method of managing the implications.
Since the concept of implications exploits the concept of correlation between symptoms, it is first necessary to create a dataset that expresses the degrees of correlation between the various symptoms.

2.2.3.1 Creation of entailments dataset

In section 2.2.1, the dataset *symptomsRelation.csv* has already been defined which expresses the correlation between symptoms. However, in the elaboration of this dataset, the focus has not been placed on the degree of correlation between the symptoms, that is, it has not been considered a numerical value expressing its relevance.
From here came the idea of elaborating a dataset, the *entailments.csv*, similar to the one already previously structured, the *weights.csv*, in which each symptom is described by a list of related symptoms, each of which is associated with a value expressing the **degree of correlation**.
The first step was to understand how to establish the degree of correlation between symptoms: starting from the initial dataset *symptomsRelation.csv*, a simple correlation between two symptoms was defined as the occurrence of both symptoms in at least one disease, now the goal is to define a strategy to define a stronger correlation between symptoms
In order to estimate when two or more symptoms are *strongly cor-*

related to each other, we have used the *Apriori algorithm* [23, 24]., which allows you to find out how the elements are associated with each other within a set of data.
Referring to this well-known algorithm in combination with the initial dataset, i followed this approach: for each symptom, described by a list of related symptoms, we calculated the frequency of occurrence in all dataset diseases with each symptom present in its correlation list. As a result, it is obtained a dataset in which each symptom is linked to a list of symptoms, each of which has a likelihood of co-occurrence with the symptom in question. In addition, i have used the concept of *relative frequency* described in [25], in order to obtain a normalized value that represented the *correlation-relevance* between symptoms—how much the symptoms are correlated to one another.
Moreover, in order to further improve the efficiency of our framework, we have introduced a new parameter: *the implication threshold*.
The implication threshold represents the minimum value of correlation-relevance for considering two symptoms as strongly correlated.
The physician has the possibility to choose the minimum threshold to establish which are the entailments.

2.2.3.2 Concept of implications

The main idea behind the concept of implications is to reduce the number of interactions with the patient by exploiting some relevant information provided by the patient.
In working out the solution, it was decided to start from the framework proposed in [1], and extended it in order to make it possible to deal with both implications between the symptoms, and with non binary weights associated to the slots.
It has been utilized Equation 1.10, which evaluates the usefulness of a symptom s prior to its acquisition throughout the inquiry process.

Thanks to this measure, it is possible to estimate which symptom provides the most useful information at a given stage of the interaction with the patient. It means that the framework is able to provide at each interaction with the patient the best symptom to ask for.
The patient at this point if affected by the symptom or not will be able to answer Yes or No to the proposed diagnosis of the framework; depending on the information obtained the framework will include the best path to follow.
Starting from this starting point, what we would like to achieve with the extension of the concept of implications is to understand whether the occurrence or non-occurrence of a symptom during an investigation process can automatically give rise to the occurrence or non-occurrence of another symptom.

When the symptom is in the left-hand side of an implication rule of symptoms, in case of a positive answer by the user (i.e., the user has the symptom), the framework can estimate which is the other symptom that the patient could be suffering from—the symptom in the right-hand side of the rule.
For this reason, and in order to further improve the management of the inquiry process, we have introduced an additional method that exploits the concept of implication between the symptoms.
Indeed, we have situations where the occurrence of a symptom is a necessary and sufficient condition to *imply* the occurrence of another symptom.
For example, we suppose to have a patient who informs us that he/she has a high fever. We can then consider that he/she is in a febrile state. Therefore, any disease that has a fever as one of its symptoms should be considered a possible candidate for the correct diagnosis for the patient.
We must also consider situations where the opposite occurs: the non-

occurrence of a symptom is in turn a necessary and sufficient condition to establish that another symptom certainly will not occur.
If the patient does not suffer from the requested symptom, before starting a new interaction with the patient, the framework considers the implications and checks the symptom/s to discard because in correlation with the asked symptom.
To determine when two symptoms are involved, the concept of relevance between symptoms is used. If a symptom is closely related to another symptom and is therefore present in its list of related and to a correlation value of greater relevance than the threshold imposed by the user (doctor) of the framework, the two symptoms then define implicated among themselves.
Since the concept of implication is an extremely rigid concept it is necessary to estimate a sufficiently high value of implication threshold.
In this way, the framework gains efficiency and effectiveness by reducing the inquiry process and thus optimizing the interaction with the patient.

2.2.4 Filtering Dataset

As noted above, the extension of the concept of implications has reduced the number of interactions with the patient and thus increased the performance of the framework.
However, the mere application of a system of implications did not give us satisfactory results in terms of timing, therefore, it was clear that a single reduction in the number of interactions with the patient did not necessarily imply a reduction in the time needed for the in-formation acquisition process.
From here, a remarkable problem already described by texeira et al in their work was reviewed, namely the framework's inability to handle a large amount of data. The analysis of these limits showed that

a reduction in the dimensionality of the dataset would provide some improvement in the efficiency of the framework.
The proposed solution to this problem was therefore to filter the dataset according to some peculiar characteristics of the patient: age, gender, gender and create ad hoc datasets based on these categories. In this way every time the framework interfaces with a patient with certain characteristics will know what will be the appropriate portion of data to the subject.
The idea of filtering the dataset has been inspired by some contributions in the literature, like the work by Krieger and colleagues [2], in which they examined the influence of gender and ethnicity, for knowledge about health and for the provision of health care.
In their work, they underlined that the data they have used "indicate that white men and women generally have the best health and that men and women, within each ethnic group, have different patterns of disease".
They also emphasized that "the health status of men and women differs for conditions related to reproduction, but it differs for many non reproductive conditions as well".
Other studies, for example, the one by Deek and colleagues [3], show that "gender, age and sociocultural factors are likely to influence health-related behavior, including screening".

2.2.4.1 Data Re-calibration

Filtering the initial dataset has also required recalibration of the symptom weights, which vary according to the patient characteristics (age, gender, and ethnicity).
In order to achieve this, the two dataset *weights.csv* and *symptomsWeightAge.csv* were used, from which new datasets were obtained where the relevance of the symptoms was based on a certain

age range. Therefore, for each symptom within the weights' dataset, a recalibration process has been carried out according to the percentage value of decrease or increase present in the other dataset.
Depending on the disease examined and the age range considered, any symptom describing a disease could have increased its value, in case the symptom was more relevant for that disease on a subject with a different age range, or vice-versa a decrease.
In fact, as a particular disease may occur more likely on one patient than on another depending on gender, age, ethnicity, in the same way the value of a symptom for a specific disease can change between different age groups, for example, the younger you are, the more unlikely you are to suffer from incontinence, the more the years increase, the greater the probability and therefore the relevance.
Applied this procedure to every set of symptoms characterizing the diseases it has been necessary to re-make a process of normalization, as described in 2.2.2, in order to maintain new datasets that always respected the system requirements.

In addition, once a recalibration of weights has been applied according to age, it has been chosen to combine also the information present in diseaseDataframe.csv with our new datasets. In fact, in this dataset there is information about the occurrence of a certain disease on a determined according to certain characteristics: age, gender, ethnicity. In addition to having a recalibration of weights according to the disease they are associated with and age, there is the possibility of measuring the risk that a certain disease may occur depending on the gender, age or ethnicity of the patient in question. In fact, in this dataset there is information about the occurrence of a certain disease on a specific patient based on certain characteristics: age, gender, ethnicity. In addition to having a recalibration of weights according to the disease they are associated with and age, there is the possibility

of measuring the risk that a certain disease may occur depending on the gender, age or ethnicity of the patient in question. For this reason, new datasets have been created, one for each selected age group, where each disease has been described by the set of symptoms related to re-weightscalibrated and more by the risk factor in percentage terms according to the specific characteristics of the subject under investigation.

The age ranges based on the years in which the datasets were distributed are as follows:
(i.) Age < 1; (ii) 1 ≤ Age ≤ 4; (iii) 5 ≤ Age ≤ 14;(iv) 15 ≤ Age ≤ 29; (v) 30 ≤ Age ≤ 44; (vi) 45 ≤ Age ≤ 59; (vii) 60 ≤ Age ≤ 74; (viii) Age ≤ 75.

2.2.4.2 Introduction concept of priority

Furthermore, depending on the patient, a filtered dataset provides higher relevance to particular symptoms or disease.
This is why it was introduced the *priority* property, which is utilized to filter data as well as throughout the inquiry process, as one of the essential elements for determining the *best next symptom* to ask the patient. The attribute *priority* reflects the likelihood (on a scale of 0 to 1) that a patient would develop a specific disease depending on age, gender, and ethnicity.
In addition, this value was used to calculate the *usefulness function* described in 1.2.3 and in particular for the term $|O^s_{G_a}|$, which is the sum of the disease priorities having a certain symptom s in its set of symptoms.

In order to compute this probability value, we have been inspired by the *Apriori Algorithm* proposed by Agrawal *et al.* in [23]. In a database, the Apriori algorithm is a data mining approach for identifying frequent item groups and appropriate association rules. The

found association rules will be graded on support, which indicates a rule's generality, and confidence, which represents the likelihood that the association is right for a particular instance.
These later probabilities are utilized to describe the likelihood that a certain patient, based on their age, ethnicity, and gender, would get a specific disease. The re-scaling of the sum of those probability values, using *min-max normalization* determines the *priority* for each disease.
One of the most prevalent methods of data normalizing is *min-max normalization*. The minimum value of each feature is converted to a 0, the highest value is converted to a 1, and all other values are converted to a decimal between 0 and 1.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2.1}$$

where x is an original value, x' is the normalized value.

Only the patients with a priority level higher than an established minimum *priority threshold* should be considered in the dataset. There is therefore a direct proportionality between the growth of this parameter and the reduction of the dataset. Indeed, the higher the priority threshold, the greater the reduction in the dataset, and *vice versa*.
Nevertheless, if the priority threshold is set too high, there will be a risky situation in which a large number of diseases will be missed, with the probability that the selected disease will belong to the range of previously discarded diseases.
Therefore, the priority threshold should be set to a value that results from a trade-off between the utility of discarding unnecessary diseases with a low probability of occurrence, but at the same time with a value that is not too high to discard too many diseases among which the disease under consideration could be found.
For this reason the physician is given the possibility to choose a suitable value for the *priority threshold*.

2.3 Implementation Phase

At the end of the data processing phase, we moved to the realization of the framework with particular attention to the implementation aspect.
In order to achieve this, the framework was modelled as originally conceived by Teixeira *et al.*, in which some improvements were made by introducing, for example, the concepts previously described of implications and priorities.
The key aspects on which the project was divided were :

- the creation of a virtual patient able to interact with the framework so as to simulate some tests in the future;
- the underlying structure capable of calculating at each iteration with the patient all the parameters necessary to calculate the value of usefulness of symptoms and indicate the best one for the patient considered.

2.3.1 Creation of a virtual patient

The need to create a virtual patient has been a glaring aspect since the very first steps in the realization of the project.
In fact, creating a framework of which you do not have the ability to test performance, would bring something superfluous and unnecessary for research purposes. Since the aim of the project was to improve an area in which others had already tried, the only effective way to validate the results was to be able to get a feedback.
In addition, in the medical field, it is increasingly difficult to find personal patient data on which to simulate the picture realistically, for this reason, an alternative solution has been adopted. It was decided to create a virtual patient with all the necessary characteristics such as the disease to be diagnosed, the set of symptoms associated with it

and the peculiarities of the patient: age, gender and ethnicity.
To achieve this goal, a Python *Patient* Class has been defined which is able to return an instance of the patient type characterized by the set of attributes described above.
Each time the framework is launched for its execution a different patient will be generated.

2.3.2 Framework initialization

Once created the virtual patient on which to test the functioning of our framework it is necessary to implement the framework structure. The first step to be carried out was to set the value of all those parameters customizable within the project such as the *implication threshold*, *priority threshold* and the *threshold degree* τ. Initially, it is also necessary to choose which type of data set to use during the investigation process: a **filtered** or **unfiltered** data set. The choice was trivially implemented through a keyboard command.
The chosen dataset was represented through a python data structure of dictionary type. In this dictionary, each disease is linked to a list of symptoms and each of them is described by a couple of values:

- **weight**: relevance of the symptom associated with that disease;
- **slot**: an integer value in the range [-1,1] which is useful to trace what symptoms are felt by the patient.

In addition, each disease is associated with a *priority* value indicating the relevance of the disease depending on the subject.
During the inquire process, the framework submits a symptom to the patient who can reply: (i) **No**, if the patient does not have this symptom, then the symptom slot is set to 0; (ii) **Yes**, if the patient is affected by the symptom and consequently his slot is set to 1. Eventually, the slot might remain -1 if the symptom has not yet been asked to the patient.

Furthermore, during the creation of this data structure, are also discarded all those diseases that do not respect certain constraints already mentioned above: (i) if a disease has a lower priority than the fixed priority threshold, it is not considered relevant for diagnosis; (ii) if a disease is described by a set of gender-specific symptoms, male or female, they are discarded depending on the gender of the subject.
In a situation where some specific gender-specific symptoms are discarded in a disease, it is necessary to verify that the sum of weights of the remaining symptoms is still higher than the *threshold degree* 1.12.

Subsequently, it was necessary to choose how to begin the diagnosis process with the patient. It was therefore decided to simulate the process of investigation with the patient assuming that, at the beginning of the interaction with the framework, the subject considered provided to the framework at least one ***relevant symptom*** in the development of the diagnosis. Within a set of symptoms, one is considered relevant if its weight is at least greater than a minimum value of 0.1.
In order to do this, the symptom set was represented through a python dictionary in which each symptom was structured by key: the name of the symptom and the value: the weight of the symptom. Once the dictionary was done, it was easy to filter out symptoms and preserve only those considered relevant by randomly choosing one.
Starting from the choice of the relevant symptom, the *concept of implications* described in 2.2.3.2 was immediately applied. The latter was used to identify whether there were symptoms implicated by the identified significant symptom, namely those symptoms with a higher degree of correlation than the *implication threshold*. To do this, we went to exploit the *entailments.csv* dataset defined in 2.2.3.1, and went to observe if among the related symptoms there was at least one **strongly correlated** the chosen symptom.
In addition, the *concept of implications* gives us an additional advan-

tage: the latter works as a recursive process, in fact the occurrence of a first strong correlation between two symptoms, allows consequently also cascade to verify if the new symptom implicated in its turn can be strongly correlated with another symptom and so on.
Once strongly correlated symptoms of the disease have been established, the remote possibility that a certain verifiable symptom only in subjects of a given gender could be strongly correlated with a symptom exclusive to another genus has been ruled out. In order to do this, the *symptomsGender.csv* dataset defined in 2.2.1 , was used to verify the symptoms involved.

This first phase of initialization of the framework also includes *Confidence degree* and *Missing information* parameters described in 1.2.2; for both parameters, the data structure used to represent the dataset chosen above was exploited.
The **Confidence degree C(g)**, the information knowledge about the symptoms, has been calculated from its formulation [1.10] in using the weight of the beginning symptom affected by the patient and the weights of the other symptoms related to its symptom.
Once the symptoms from which the patient is initially affected are defined, the "probability" that the disease may occur in the patient is calculated for each disease. This probability is dictated by the sum of the weights associated with the symptoms that have the fixed slot to 1, that is the symptom incurs in the patient.
The **Missing Information M(g)**, the symptoms are not yet called to the patient and on which we have no information, was calculated from its formulation[1.11] using the weight of the symptoms of which we have no information and the slots related to these are set to -1.

2.3.3 Calculate Usefulness function

After the initialization phase of the framework in which the structures to be used were defined and the first useful parameters of the project were calculated, the next step was to calculate the usefulness function **U(s)**, defined in 1.2.3, for each symptom.
This function measures the usefulness of a symptom not yet asked during the inquire process with the patient; it is useful to calculate the best symptom to propose to the patient since it is considered in line with his diagnosis.
At the end of the computation each symptom will therefore be described by a numerical value: its usefulness; the one with greater value will be the best symptom to propose to the patient in the current interaction.
The *usefulness function* is characterized by different parameters that, at each iteration, must be calculated in two specific situations:
(i) if the symptom for which the measurement is performed is needed in the patient, then the slot set to 1;
(ii) in the situation where the patient may not be affected by the symptom, then the slot will be set to 0.

2.3.3.1 Usefulness Function's parameters

The first parameter that has been calculated is the term **E(s)**, defined in 1.13, which is the number of diseases that the patient could suffer from in case he is affected by the symptom or the diseases he would not suffer from in case the symptom does not occur.

Let's begin to describe the calculation of the term **E(s=0)** in case the symptom does not occur in the diagnosis of the patient (slot = 0).
The occurrence of a disease in a patient is associated with the variation of its *Confidence degree* and *Missing information*. Each symptom

considered, as known, is described by a different weight depending on the disease in which it appears. The non-occurrence situation of a symptom causes the variation of the parameter *M(g)* in each disease in which this symptom is present; this will decrease its total value of the relevance of the symptom.
As for the parameter *C(g)*, this remains unchanged as the absence of a symptom does not cause an increase.
The analysis of these two parameters leads to the analysis of two different situations:

- M(g) - weight $< \tau$: if the parameter Missing information of the disease, decremented of the weight of the symptom, becomes smaller than the threshold degree, then i can consider such disease not harmful and increase the parameter **E(s=0)** relative to the considered symptom;
- C(g) $\geq \tau$: if the Confidence degree, while remaining unchanged, is higher than the threshold degree, then the patient is affected by the disease and must increase the parameter **E(s=0)** related to the symptom considered.

Consider now the **E(s=1)** parameter in case the symptom occurs in the patient's diagnosis (slot = 1). In this situation, the Confidence degree of each disease in which the symptom in question appears is increased by its relative weight.
Therefore, the following situation may arise:
C(g) + weight $\geq \tau$: if the value of Confidence degree increased by the weight of the symptom is greater than threshold degree, then that disease is considered harmful for the patient and the term **E(s=1)** relative to the symptom is increased.
At the end of each symptom, I will have a count of diseases satisfied, $\geq \tau$, due to the activation of each slot.

The next parameter to consider for the computation of U(s) is the term $\boldsymbol{N_1}$, defined in 1.4, that is the sum of all the weights of the symptoms strongly correlated with the considered symptom in all the diseases.

As in the previous situation, we begin the computation of the parameter $\boldsymbol{N_1}$ for $\mathbf{s = 0}$, so in the cases where the symptom may not incur in the patient.

For the calculation of this parameter we use the concept of implication which gives us a tool to say that the occurrence of a symptom can induce, in case of strong correlation, also the implicit occurrence of a further symptom. On the other hand, this concept, as already mentioned, can be further expanded also in cases where the absence of a symptom may result in the absence of another symptom.

For example, suppose you have a set of 2 strongly related symptoms: s1 $\rightarrow$[2] s2.

In general this means that if s1 is needed in the patient (slot = 1) then implicitly s2 is also incurred in the patient. Therefore, in case the symptom s2 is not needed in the patient it is also true that s1 is not needed in the patient (slot = 0); in the calculation of $\boldsymbol{N_1}$ for $\mathbf{s = 0}$ we exploit the second approach.

In order to calculate this parameter, we analyze for each symptom the list of related symptoms. Now, it is known that if it is any symptom it is present in the list of related symptoms of a given symptom, it will evidently be true also the opposite. However, it is not valid that, if the occurrence of a symptom automatically induces the occurrence of a subsequent symptom, then the opposite is also true. For this reason it was necessary to perform a double check also for the symptoms within the list of correlates of the chosen symptom, analyzing in turn the correlates.

[2] $\rightarrow$: this symbol explains strong correlation/implication between symptoms

In the situation where one of the related symptoms present in turn in its list of related symptoms is the selected symptom with the highest correlation value of the *implication threshold*, this means that it is strongly related to the former. This leads us to say that if the first symptom is not necessary then neither will the other. This concept can be repeated recursively using the same procedure on each symptom.
Once i get the implications for the specified symptom, i make sure I don't consider exclusive symptoms for a certain gender and for this reason i repeat a filter for gender.
The next step was to add to each other the weight of symptoms obtained through implications.
In the end, for each symptom, I will have a total weight that will indicate the relevance of the symptom considered within all diseases.

For the calculation of the parameter in case $\boldsymbol{N_1}$ for $\mathbf{s = 1}$, the computation process followed was the same.
On this occasion it was easier to understand how to proceed: referring to the example shown earlier in this we are interested in demonstrating that the occurrence of one symptom automatically induces the occurrence of another. In order to verify this it occurred if one symptom was strongly related to another(it directly implied) and so on if the next one was in turn related to another.
Once the implications were over, the procedure already demonstrated before was repeated, which led to a total weight associated with each symptom.

The following parameters for which the calculation was performed for each symptom were the terms $|O^s_{G_a}|$ and $|G^s_a|$, defined in 1.2.3, for which it was preferred to use a single structure of dictionary type for storage. For the computation of these two terms, differently from what

we have seen before, it was not necessary to subdivide them according to the occurrence or not of a symptom.
The first of the two, the parameter $|O_{G_a}s|$ is the sum of all the priorities of the diseases in which the considered symptom is present; the latter has been calculated simply by analysing the appearance of the symptom in the diseases and adding up the priorities of the diseases in which the symptom appears in the set of symptoms related to them.

The term $|G_a^s|$ is the number of the diseases in which the considered symptom is present. To do this, during the $|O_{G_a}s|$ calculation, a counter was integrated that was increased whenever the symptom appeared in the list of symptoms describing the disease. At the end of the computation therefore every symptom has been described both by the sum of priorities and by the counter of diseases in which it is present.

The last useful term to be able to perform the computation of the usefulness function is the term $\boldsymbol{N_2}$(s), described in 1.5
From 1.5 it is known that $\boldsymbol{N_2}$(s) is defined as the sum of weights of all disease symptoms minus the term $\boldsymbol{N_1}$(s) where s could be 0 or 1. However, it is not necessary to calculate this term because we know from the definition of usefulness function [1.2.3] that the term $\boldsymbol{N_2}$(s) appears only in the denominator of our function that presents itself in this way:
$N_1(s) + N_2(s)$ where $N_2(s)$ = sum of all weights - $N_1(s)$.
This means that we do not need to calculate all the term $N_2(s)$ but only the portion of the denominator useful for the calculation of the utility function, i.e. we are only interested in the sum of all weights.
The sum of all weights is the number of disease still considered from the framework because we know for 1.9 that the sum of all weights for each disease is 1, so i means that:
sum of all weights = 1 * N° of diseases actually activated.

Once the calculation of all the terms has been completed, it is possible to measure the usefulness function as in , and analyze its results: that is, to choose the next useful symptom to be placed to the patient. The patient, depending on his diagnosis, may or may not suffer from the symptom and provide this information to the framework. The framework knowing this information is able to update the useful parameters and repeat the procedure as long as at least one of the papabili diseases reaches the predetermined Threshold and is a hypothetical patient's disease.

Chapter 3

Preliminary results

The experimental work presented here has been implemented in Python with Pycharm tools and Google Colab on a machine having a configuration of Intel Core i7-8750H, 2.20 GHz, running the Windows 10 (64 bit) operating system with 16 GB of RAM.

3.1 Implementation Details

The above-mentioned technique was implemented and empirically validated as follows:

1. A patient is randomly generated from Symcat's [4] as described in Section 2.1.1. Our virtual patients are created by associating them with a set of specific features (age, sex, ethnicity) and a set of *relevant symptoms*[1] chosen at random, associated to the disease to be to diagnosed.

2. The patient-physician interviews are then simulated by our framework, taking into account the symptoms and the patient answers. The time taken by the physician to ask for the next symptom and for the patient to answer is randomly simulated with an average

[1] A relevant symptom is a symptom that is associated to the considered disease at least with weight 0.1.

iteration time of 45 seconds and a standard deviation of 30 seconds.

3.2 Parameters Considered

We take into account the following parameters:

- **Threshold degree** τ, mentioned in Section 1.2.2, defines when a goal has been reached or not , that is whether the patient suffers from this disease or not. This refers to the extent to which the given information is adequate to make a diagnosis (or, alternatively, to decide on the absence of a disease).

- **Accuracy**, refers to the framework's capacity to identify a disease. The accuracy value is in the [0, 1] range.

  - if the chosen disease belongs to the list of the first 10 possible diseases $[g_1 \dots g_{10}]$, we consider that the framework efficiency is maximal (100%), thus the accuracy value is set to 1;
  - if the chosen disease belongs to the list of the second 10 possible diseases $[g_{11} \dots g_{20}]$, we consider the framework efficiency to be 90%, thus the accuracy value is set to 0.9;
  - if the chosen disease belongs to the list of the third 10 possible diseases $[g_{21} \dots g_{30}]$, we consider the framework efficiency to be 80, thus the accuracy value is set to 0.8;
  - and so on ...

  This way of measuring accuracy is justified by the fact that the probabilities of the diseases in each block of 10 are so close to each other that it would be hard to choose one of them over the others as a "more correct" diagnosis.

- **Number of steps**: physician-patient interviews are composed of a number of question-answer based exchanges (generally the

patient answers a question about a particular symptom). Each of these exchanges is considered as a step in the procedure. In order to limit the interview duration, we decided to cap the number of steps at 20. Indeed, we think that 20 questions is enough for a physician to make a diagnosis.

- **Iteration time** is the duration of one question-answer based exchange, i.e the time taken by a physician to ask a question plus the time taken by the patient to answer.

- **Total time** is the time needed to establish a diagnosis from the input of the first symptom to the final decision about the health state of the patient made by the physician. In other words, it is the patient-physician interview duration.

  We decided to limit this total duration to 20 minutes. This limitation let us avoid the possible case where our framework is unable to lead to a solution and may turn into an infinite loop.

- **Confidence degree** $\mathcal{C}(g)$, defined in Section 1.10, corresponds to the amount of information collected so far by the agent. It determines if a goal g has been reached by considering a threshold—if the amount is above or below the fixed threshold a decision about the diagnosis can be taken. Nevertheless, due to limitations fixed for the total time and the number of steps, it may happen that the threshold degree is not reached at the end of the process: no diagnosis can be done. To overcome this problem, we keep track of all the past experiences in order to predict an issue "if we had had enough time": the final diagnosis would be similar to one decided for an ended previous process concerning a patient with quite similar symptoms.

For our experiments, we have estimated the values to set the parameters *threshold degree* (0.75), the *priority threshold* (0.15) and the *im-*

plication threshold (0.4), in order to underline the comparison between the two different types of dataset.
We would like to point out that the values of all the above-mentioned parameters can be easily set/modified (by a physician for example).

3.3 Evaluation of the results

Figures 3.1 and 3.2 present our results in terms of: mean, variance (std), first (25%) / second (50%) / third (75%) quartiles, minimum and maximum values.

	N° Steps	Accuracy	Iteration Time	Total Time	Cg
mean	10.57	0.89	1.21	13.52	0.54
std	5.74	0.23	0.06	7.23	0.22
min	1.0	0.0	1.1	1.1	0.0
25%	5.0	0.9	1.14	6.28	0.39
50%	14.0	1.0	1.23	19.12	0.52
75%	15.0	1.0	1.26	19.22	0.69
max	17.0	1.0	1.4	20.0	1.0

Figure 3.1: Statistics on the entire dataset: 150 patients
Threshold degree = 0.75; implication threshold = 0.4

We can notice that the accuracy for the filtered dataset (0.88) is very close to the accuracy obtained with the entire dataset (0.89). This means that we do not lose in terms of effectiveness using a filtered dataset.
An even better result is that the time saving is obvious when using the filtered dataset. Indeed, if we are only interested in the time needed to perform a single interaction we can notice how the filtered data set allows us to save on average more than 30 seconds per iteration.

	N° Steps	Accuracy	Iteration Time	Total Time	Cg
mean	13.65	0.88	0.46	9.53	0.57
std	7.46	0.27	0.31	7.14	0.25
min	1.0	0.0	0.1	0.1	0.0
25%	6.0	0.9	0.19	2.21	0.43
50%	18.0	1.0	0.3	9.5	0.56
75%	20.0	1.0	1.9	16.23	0.76
max	20.0	1.0	1.53	20.0	1.0

Figure 3.2: Statistics on the filtered dataset: 150 patients
Threshold degree = 0.75; implication threshold = 0.4; priority threshold = 0.15

A direct consequence is that the total time required also decreases significantly with the filtered dataset, saving on average about 4 minutes and 30 seconds over the whole patient-physician exchange process.
Another relevant fact that can be observed is that the time saving on a single interaction allows the interview to be composed of more steps on average for the filtered dataset. Indeed, by using a filtered dataset rather than the entire dataset we are able to perform on average three more interactions with a single patient. Furthermore, the possibility to reduce the duration of the interaction through the use of a filtered dataset, allowed the framework to reach in different situations 20 steps instead of a maximum of 17 when using the whole dataset. These improvements are evident even when the number of steps provided by the framework using the two different kinds of data is the maximum allowed (20).
We would like to point out that, using both datasets, we have at least one situation in which the framework obtained an accuracy of 0% with a consequent confidence level of 0. These rare situations are due to the fact that patients were "created" at random and no attention was paid to checking the right combination (disease, patient) with regard to gender. This results in situations where a female or male disease

has been associated with the other gender.
Finally, we have performed a further experiment on the filtered dataset which has consisted in increasing the value of the *priority threshold* and in increasing the value of the *implication threshold* (see Figure 3.3). As expected, the increase of the priority threshold, allows a further improvement both in terms of time saved and average number of iterations.

	N° Steps	Accuracy	Iteration Time	Total Time	Cg
mean	11.55	0.91	0.41	6.2	0.59
std	8.12	0.22	1.16	8.49	0.24
min	1.0	0.0	0.1	0.6	0.0
25%	2.0	0.9	0.13	0.42	0.43
50%	14.0	1.0	0.3	2.25	0.56
75%	20.0	1.0	0.46	7.46	0.82
max	20.0	1.0	1.3	20.0	1.0

Figure 3.3: Statistics on the filtered dataset: 150 patients
Threshold degree = 0.72; implication threshold = 0.5; priority threshold = 0.25

3.4 Further Analysis

Figure 3.4 presents the average distribution of values for both datasets (entire and filtered) with respect to the 5 attributes previously described. Let us have a particular look at the "Iteration Time" bar chart: it confirms that in most situations (more than 50% of the cases), the average iteration time, with the filtered dataset, is included in the [0.3, 0.5] interval (in minutes).
While with the entire dataset, even if the distribution of the average iteration time is less uniform, it is in any case evident how the majority of the values is included in the [1.20, 1.30] interval (in minutes). This confirms our previous considerations on interaction time.

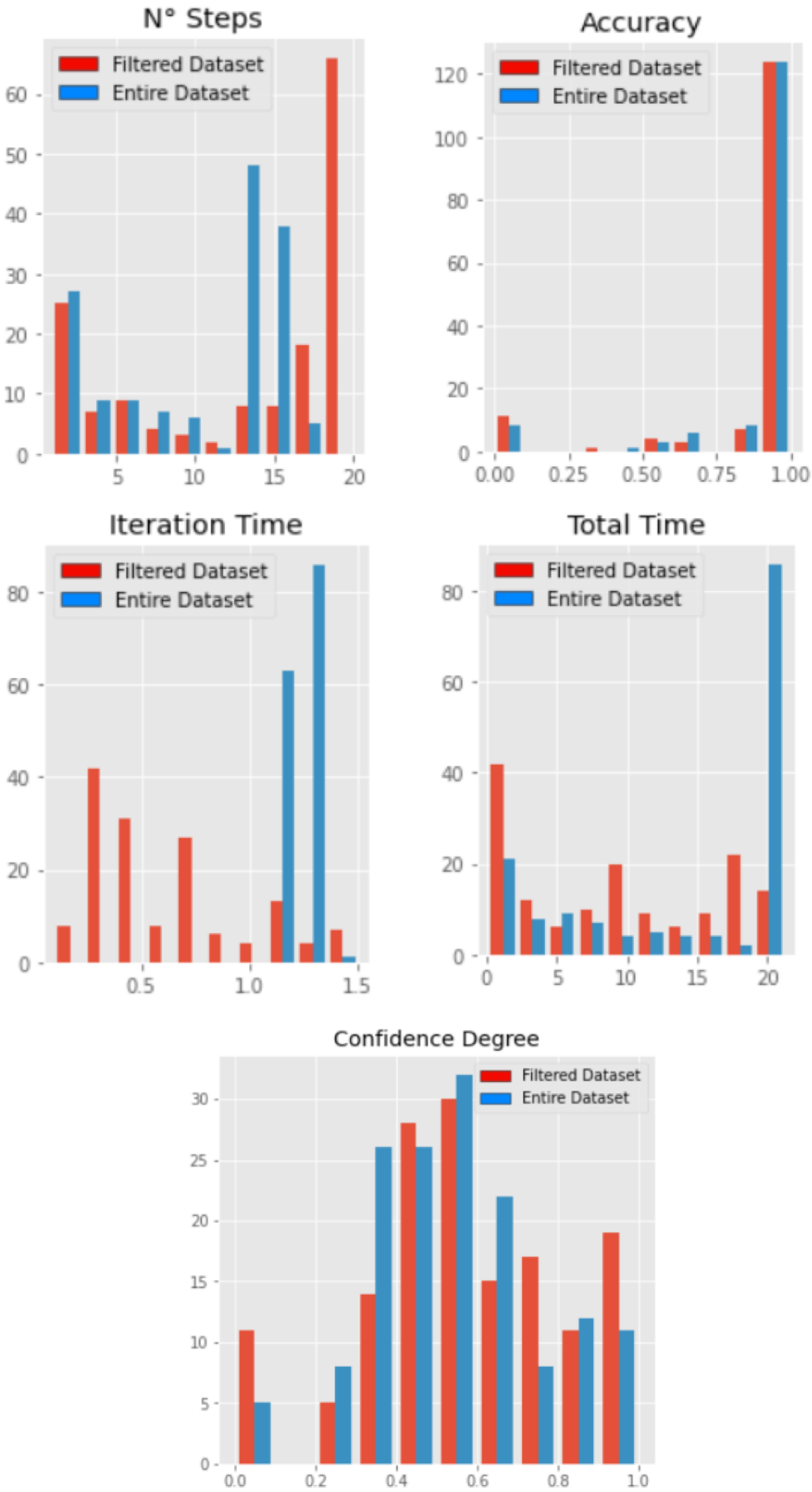

Figure 3.4: Distribution of values for both datasets: 150 patients
Threshold degree: 0.75; implication threshold = 0.4; priority threshold = 0.15

These last results obviously also impact the distribution of the values around the relevant attributes: the number of steps ("N° Steps") and the total time ("Total Time").
As described above: using a filtered dataset allows to save time for each interaction and thus to perform more interactions with the patient. Indeed, the corresponding bar chart (N° Steps) shows that in 45% of cases (68 patients over 150) the maximal number of interactions (20) is used. Therefore, the diagnosis is improved.
Nevertheless, we can observe that in terms of confidence degree and accuracy, in both graphs the distribution of values is quite similar. Indeed, as far as the degree of confidence is concerned, the majority of the values are included in the [0.4, 0.6] interval, while about 85% of the values are included in the [0.8, 1.0] range for the accuracy in 130 cases.

3.4.1 Presence of outliers

In 3.3, where the results obtained were analysed in detail, it was found that in most circumstances, the accuracy values obtained are distributed in the range [0.8,1.0] demonstrating a high level of accuracy and disease forecasting.
During the previous analysis, we observed that in some rare circumstances, some values did not reach high scores, resulting in a decrease in the average accuracy.
We have decided to analyze the reasons why these out-of-context results *outliers*[2] were obtained. To this aim, we represent these values by creating a plot of outliers.

[2]an **outlier** is a data point that differs significantly from other observations.

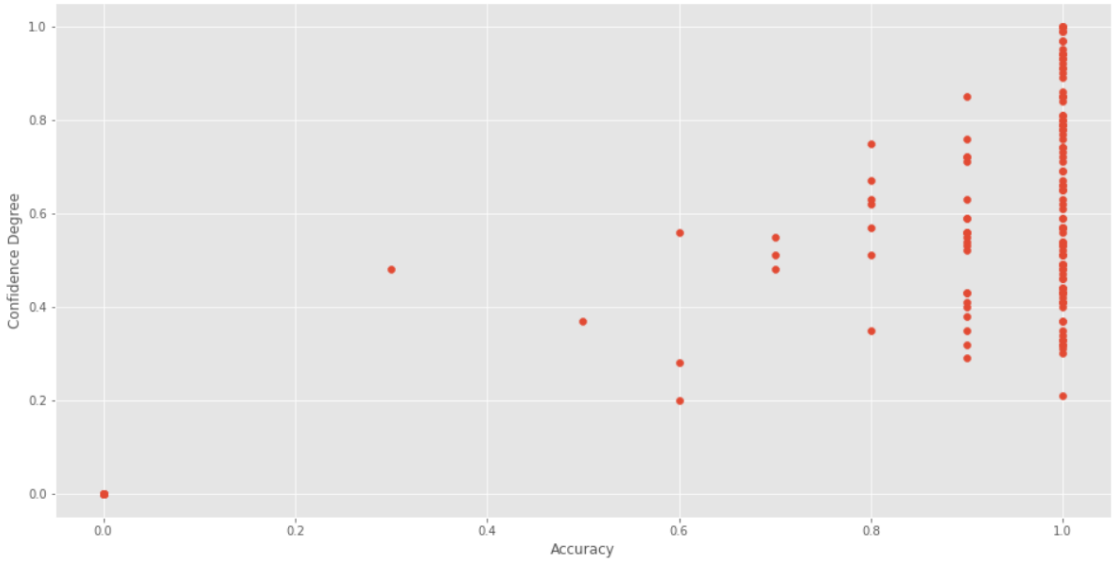

Figure 3.5: Outliers on Accuracy attribute filtered dataset;150 patients.
Threshold degree: 0.75; implication threshold = 0.4; priority threshold = 0.15

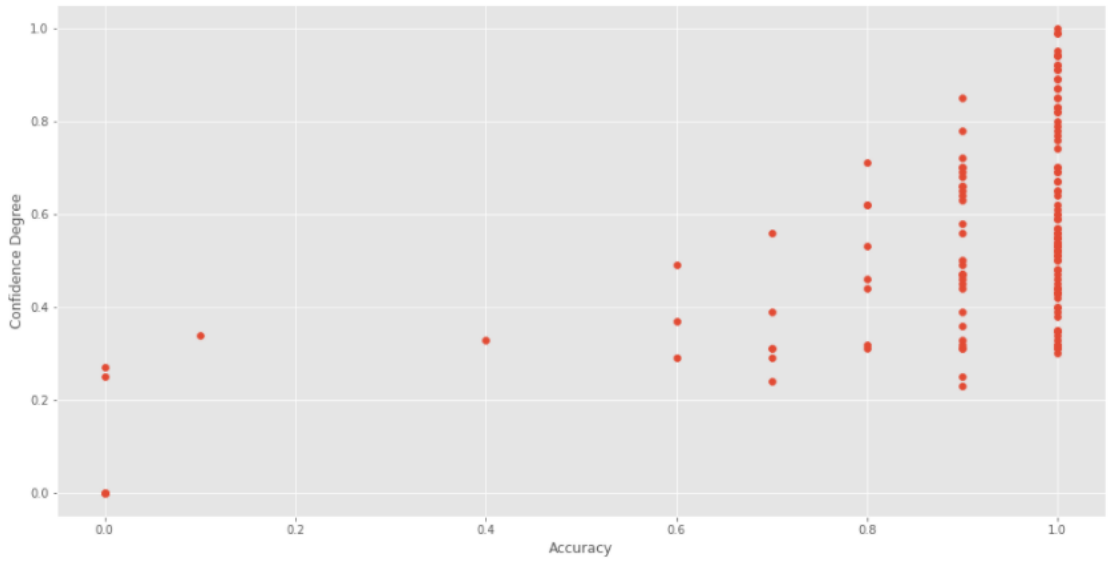

Figure 3.6: Outliers on Accuracy Attribute entire dataset:150 patients.
Threshold degree: 0.75; implication threshold = 0.4;

As shown in the graphs 3.5 and 3.6 on both datasets there are some cases where the accuracy value is close to zero; This means that the confidence degree too is is extremely near to 0.

As mentioned in Section 3.3, this is due to the fact that the patients were created randomly, resulting in situations where a typically female or male disease is associated with the wrong gender. This results in

an inability of the framework to correctly identify the correct disease-related symptom to ask about.
On the other hand, this erroneous result confirms the effectiveness of our framework that in those situations it is unable to recognize the disease associated with the patient since it is unreal that a patient of a given gender could have a disease of the opposite sex.
After understanding the reasons for the occurrence of outliers, the next step was to re-calibrate the results excluding those values. After understanding the reasons justifying the presence of the outliers, the next step has been to re-calibrate the results by excluding those values. We exploited the parameters obtained previously and presented in Section 3.3 about the first, second, third quartiles. Indeed, using these parameters, we were able to estimate a threshold for the accuracy, below which a result obtained could be considered as an outlier. The obtained graphs show and confirm that such a threshold value turns out to be 0.8 on both data sets as the histograms showed previously suggested.
Once we set the minimum precision value at 0.8, we decided to remove all values in our results below this threshold, in order to estimate what the results would be without considering outliers. The results showed that for both of datasets, there were about 20 outliers out of 150 in the test set, a percentage lower than 15%. In addition, these new results show an improvement, in terms of accuracy, of about 3/4%, reaching an accuracy of about 92.5%.

Chapter 4

Framework improvements

4.1 Concept of combinations of implications

In the 2.2.3.2 section, it was already described the *concept of implications*: a technique useful to reduce the process of inquisition with the patient. In fact, using this method, we are able to evaluate in advance whether a patient affected or not affected by a specific symptom may be affected or not affected by another symptom.
In this part of the work the aim is to further extend this concept. Previously, it was explained that this system was used only in the implications between two individual symptoms, so a symptom could imply or be implicated only by another symptom. Now we would like to introduce the possibility of considering a chain of implications by exploiting reciprocal combinations of symptoms.
During the inquisition process with the patient, we discover a set of symptoms that affect or do not affect the subject considered and, in any case, we can obtain some useful information starting from these sets of symptoms. The main idea is to establish at each step of the framework (each iteration with the patient) which are the symptoms that occur and vice versa which are those that don't occur. In this approach, it is possible exploiting the concept of implications and its improvements about the combinations. At the end of any iteration, the

framework will compute before the single implication between symptoms (one to one) and after we will create a combination of 2, 3 or 4 symptoms which together could imply one or more symptoms.
Starting from this idea for the single implication, we have decided to extend it and to apply it to the combination of implications. In this situation, we will have more symptoms on the left side that could imply another symptom (right side).
It refers to a combination of two or more symptoms in two particular situations: (i) when all the symptoms in question, if necessary in the patient, can be considered at the same time to imply the occurrence of another symptom by the concept of implication; (ii) how much all the symptoms in question, not needing in the patient, can lead to the absence of an additional symptom.
Thus, the combination of two or more symptoms is described by the concept of logical AND. Suppose we consider for example two symptoms s_1 and s_2 and a third symptom s_3 implied by the combination of these two, if we wanted to use the nomenclature previously used should express this concept in this way: $s_1 \wedge s_2 \rightarrow s_3$
On this occasion, the dynamic above explained, become stronger and more rigid. In fact, when we have a combination between symptoms, two conditions simultaneously have to be verified:

1. the symptom on the right side of our implication, that is the symptom of which we would like to know the behaviour, must be present in the correlation list of all the symptoms in and on the left side;

2. The sum of co-occurrence probabilities for the symptom considered within the symptom correlation lists on the right side must be greater than a given parameter defined specifically for the combination of symptoms. Previously, a parameter for single implication had been defined, defined as *implication threshold*, now

we will introduce a new parameter: ***implication combination threshold***.

This approach should be used to reduce the dimensionality of the dataset step by step and, obviously, the number of interactions with the patient.
The main idea should be to diagnose the patient's disease in a number of steps less than the previous work. Nevertheless, this approach that exploits the combination of implications become more useful when the set of symptoms considered is quite thick since the number of different combinations increase in according to the dimension of the symptoms' set.
On the other hand, a symptoms' set too large could damage the performance of the framework. In fact, the negative aspect of the implications combination can be explained through the binomial coefficient $\binom{n}{k}$ where n = length of the symptoms' set and k = the number of elements within the combinations. Therefore, the dimensionality of the list could represent a bottleneck in term of the performance for the framework, for this reason in this case too it is necessary to establish a trade-off between the dimensionality of the list and the number of iteration desired.

4.2 Concept of filtered entailments

We have already described the realization of a dataset in 2.2.3.1 that kept tracks for each symptom of a list of correlated symptoms obtained exploiting the Apriori algorithm that has permitted us to compute the probability of co-occurrence of one symptom with another.
Starting from this dataset and from the idea of the filtering dataset, depending on the peculiar characteristic of a patient (age, sex, eth-

nicity), we introduced a new dataset which was a fusion of these two aspects. In fact, when for the first time, we have realized this dataset we considered the relationship between symptoms in a general way without considering that the relevance (weight) of a symptom changes according to specific aspects of the subject as the correlation between symptoms depends on it.
For this reason, we decided to describe the correlation between symptoms exploiting different datasets, each one linked to a specific range age of a patient. Furthermore, considering the high *implication combination threshold* fixed, we decided to reduce the dimensionality of this dataset excluding all those correlations described by a co-occurrence probability less than a certain tolerance.
At the end of this process, we created more datasets according to age, gender, ethnicity so as to leaner the datasets and increase the performance of the framework.

4.3 Introduction of a filter stronger on the gender

Previously, we have generated virtual patients in a randomized way useful to perform our test. Each one is described by age, gender, ethnicity and by a disease casually chosen. Nevertheless, during the generation of our patient, we didn't 'focus our attention on the association between disease and patient considering everything in a randomised way.
This situation lead the framework to create associations between subject and disease without considering the possibility that a certain kind of disease could not occur in a specific patient. For example, we had some diseases which occurred more frequently or, in some cases, only on a specific gender like all range of diseases related to the reproductive apparatus of the individual.
Therefore, the creation of our virtual patient is quite relevant to avoid

creating unreal associations because in that case, the framework will not be able to follow the right pattern to recognize the disease. In order to resolve this situation was introduced a filter above all in terms of gender that fixes a minimum threshold to establish toward which kind of subject the disease could be linked: the ***gender threshold***. Thanks to this approach we are able to avoid those outliers situation described in 3.4.1 , caused by coupled non-real disease patients.

4.4 Performance improvement

The efficiency of our framework can not be evaluated only by its capacity to recognize the disease of a patient with higher accuracy, but it must also be capable to manage the entire process in a reasonable time. In fact, this framework is an instrument useful to give a first estimation of the diagnosis to the physician and to the patient too, but in order to do it has to be enough fast to manage the inquisition process.

Therefore, we introduced different software techniques to reduce the interaction time with the patient and consequently the entire time for the process. We noticed that the bottleneck in terms of the time of this project was the computation of usefulness function which gives us the symptom with a higher likelihood to occur in the patient. Calculating the usefulness function required, in particular, the individual calculation of all the parameters described in 1.2.3.

The computation of each one of these parameters requires a fixed time and calculate those parameters in a sequential way implies a waste of time quite relevant. In order to avoid this situation, we decided to introduce the notion of threads.

"In computer science, a thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system".[26].

Following this approach, it was decided to instantiate a thread for each parameter so as to allow its calculation parallel to the others and not in a sequential manner. In this way, the time-cost function is no longer a sum of all the times used to calculate each parameter, but it becomes equal to the time of the most expensive parameter to calculate. In our function, we observed that the highest time is given by the computation of N_1 because requires a recursive calculation of the symptom implications.

4.5 Recalibration of priority

The concept of priority has been used to provide a relevant value to the disease. This parameter is used during the computation of the usefulness function, in particular way for the term $|O^s_{G_a}|$, furthermore, for filtering the dataset. In fact, as the gender filter, the **priority threshold** parameter it is useful to discard all sets of diseases not relevant for the subject considered.
This parameter, as we have already seen, it is a combination of three diseases' attributes: age, ethnicity and gender. Nevertheless, in the previous paper, we used this parameter using a normalization approach different which foresaw the normalization only after the sum of the single attributes.
In this new recalibration, we have preferred before to normalize the single value of the attribute and then normalize the sum of the individual attributes. In this way, the values obtained are more balanced and realistic.

4.6 Additional Risks for a patient

The realized framework is able to recognize, with a certain grade of accuracy, the disease of a patient starting from a specified set of

symptoms which affect the subject. The next step in our work was to develop a framework that was also able to estimate the additional risks for a patient with a known pre-defined disease. Therefore, our idea is to provide the doctor with a tool that is able to diagnose diseases, but which, starting from an already known condition, suggests what further risks, expressed in terms of diseases, might occurr in a subject during its clinical journey. To realize this new feature of the framework, the concept of implications already presented previously has been exploited again.

Each disease as we know is described within our dataset by a series of symptoms that make that pathology unique and peculiar. Each patient suffering from a known syndrome will also be described by a subset of symptoms characterizing the disease. These symptoms combined with each other or individually by the concept of implication can verify the occurrence of new symptoms that will become the new input symptoms of our framework. The diagnosis made using these symptoms will provide us with a list of all the most plausible risks in percentage terms associated with the subject described.

Our attention was therefore concentrated in deriving what could be the future symptoms that a patient would have encountered and consequently the pathology that could arise. Through this approach we can provide the doctor with an estimate of what could be the additional diseases a patient may encounter.

We are aware of the fact that to obtain more reliable results it is necessary to have a deeper knowledge of the subject, however with this solution we want to underline how the concept of implications is versatile, intuitive and able to provide us, in a simple and linear way, with estimates, which, despite approximate, can provide the doctor useful information about the patient's clinical future.

Chapter 5

Final Results

5.1 Case Study - Steps, Diseases, Symptoms trend

The idea is to create different graphs through which it is possible to make an analysis of the trend of three different parameters compared to each other: the number of steps, the number of diseases and the number of missing symptoms.
First of all, we consider the first type of graph: *the comparison between the number of steps and diseases* described in Fig. 5.1
At each step, the number of diseases could remain the same or it could decrease. The variance of the number of diseases depends on the answer of the patient to the symptom proposed by the framework. As we know, we could have two different situations: (i) if the patient answers 'Yes', it means that the patient is affected by the chosen symptom and the framework has to update the value of the Confidence degree for each disease; (ii) if the answer patient is 'No', well the considered symptom doesn't occur in the patient pathology and the value of Confidence degree remain unchanged.
Nevertheless, in this situation, the framework has to check through the concept of implications, if there are other symptoms that indirectly can be considered useless for the patient's pathology.

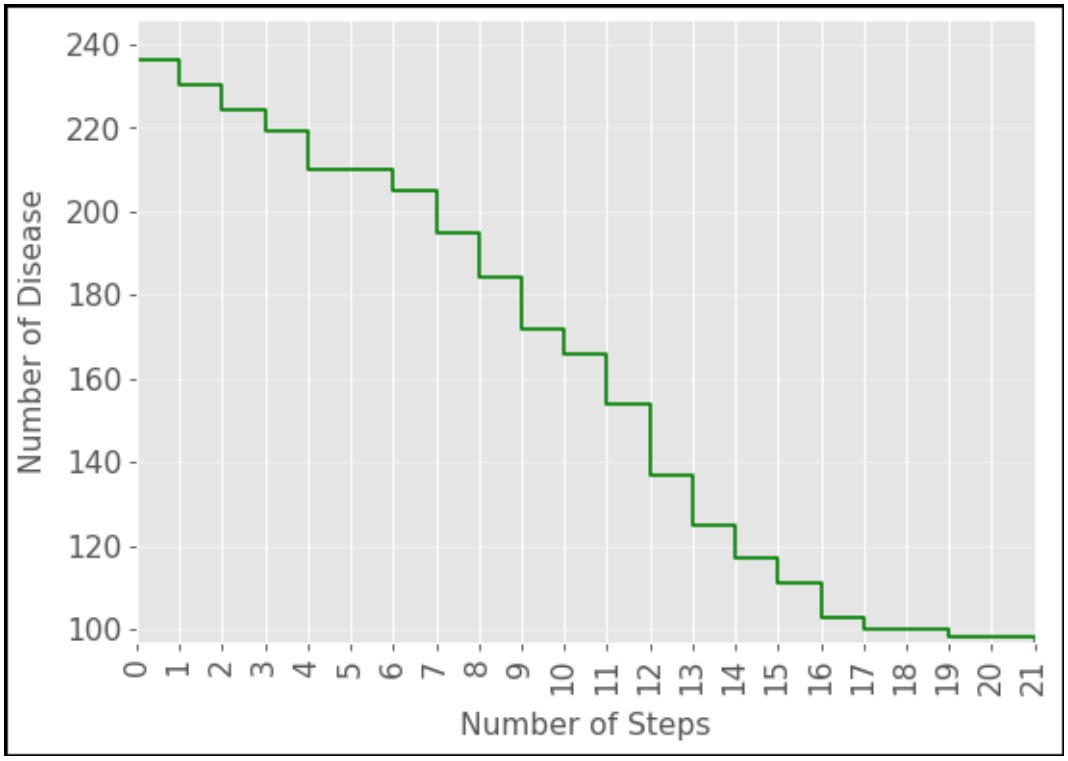

Figure 5.1: Comparison between number of diseases and the number of steps threshold degree = 0.75; single implication threshold = 0.4; implication combination threshold = 0.75; priority threshold = 0.4, gender threshold = 0.45

After this procedure, the framework has to check the number of diseases still considerable; so it verifies if the sum of Confidence degree (Cg) and Missing information(Mg) for each disease is still greater than the threshold. For each disease, where this sum becomes less than the threshold we know that is not verifiable as illness for the patient. Between two consecutive steps, the graph can decrease as a ladder or could remain constant. When the graphs remain constant the framework has chosen the right symptom for the patient, otherwise, if the answer is No, it will be there a reduction of diseases.

In order to examine this comparison, Fig. 5.1, an inquisition process was chosen with a patient aged 29-45 who was found to be suffering from 'Polycystic kidney disease'. Furthermore, as mentioned in [27], the occurrence of symptoms is more frequent in a range of age between the 30 and 50 years old; it shows the the goodness of our filter about

the priority that in this situation was able to select the right disease for the age range of the patient. Otherwise, about the gender,' males and females are equally affected' as described in [28], so the chosen of the framework about the gender is not so relevant. The framework was able to classify the disease of the patient with maximum accuracy and with a Confidence degree of 0.67.
The framework in this situation starts from about 240 diseases that it is about 1/3 of the entire dataset, about 801 diseases, this it is could possible thanks to the high value fixed for the filters. The graph shows the fast reduction of the diseases in the case in which the patient wasn't affected by a specific symptom and by its strongly correlated symptoms. Instead, we could observe between steps fourth and six and, above all, at the end of the inquisition process the number of diseases remains quite constant. It means that the framework, thanks to the deep reduction of the dataset, was able to understand the right symptoms that could have afflicted the patient. Considering the same situation, the markable reduction of the diseases between some steps is strictly linked to the job of implications between symptoms. In fact, if we have more symptoms implied each other, it should permit to the framework to discard a larger set of diseases.

Analyzing the concept of implications, it was decided to analyze the actual benefits by making a *comparison between the missing symptoms*, that is, the symptoms still to be proposed to the patient, *and the number of steps* described in Fig. 5.2. In this way it has been tried to understand the effectiveness of the mechanism of the implications: every time that between a step and the other the number of symptoms decreases of more than one, then the system of implications has intervened in efficient way.

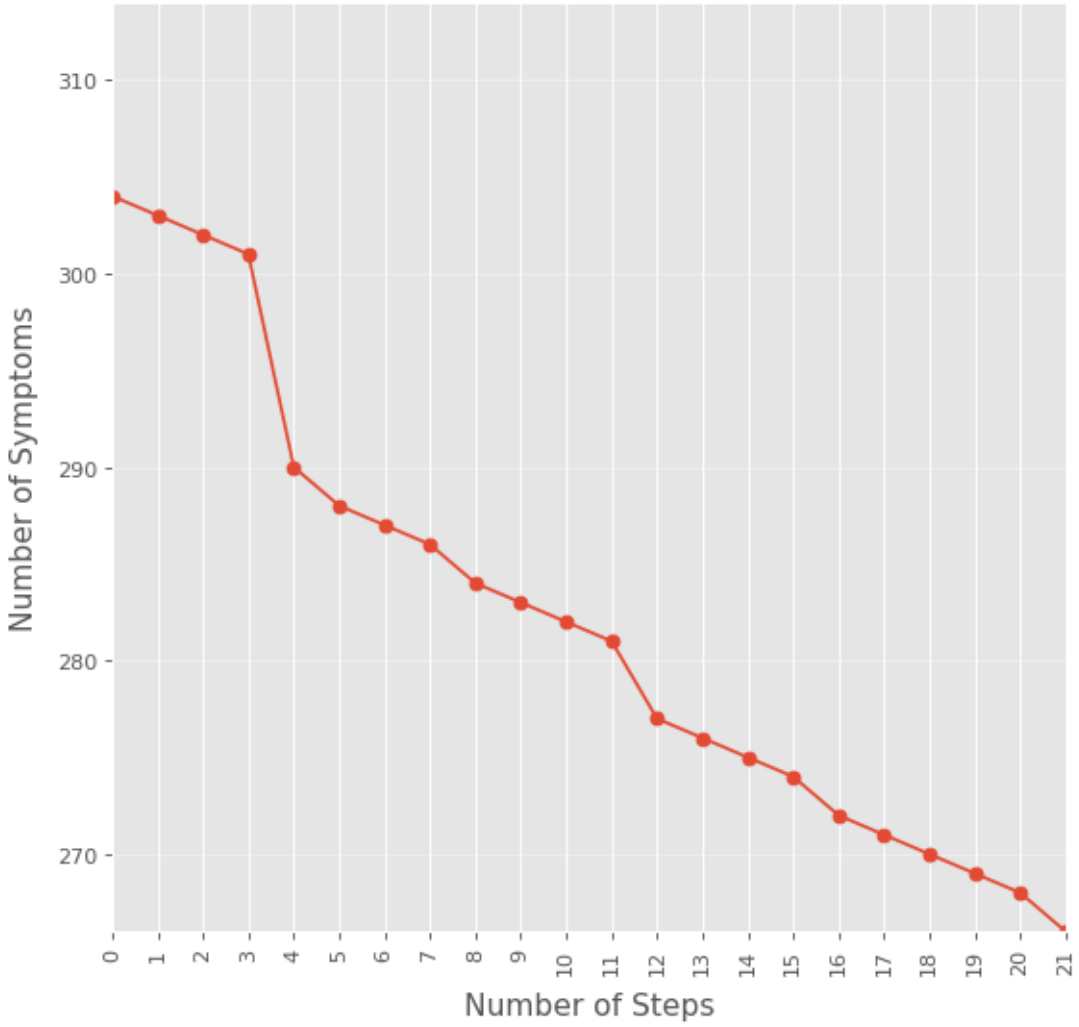

Figure 5.2: Comparison between number of missing symptoms and steps threshold degree = 0.75; single implication threshold = 0.4; implication combination threshold = 0.75; priority threshold = 0.4, gender threshold = 0.45

In this situation we can observe like between the third and fourth step there is a sharp decrement of the number of missing symptoms and in a parallel way in the graph of the disease shows the reduction of the number of disease in the same intervals. This suggests us that in this interval the answer of the patient was false and the framework, exploiting the concept of implications, it was able to reduce the number of symptom considered and consequently also the disease which they don't satisfy more the relation already described in 1.12.

The last situation that we want to examine is shown in Fig.7: the comparison between the increment of the number of steps and the trend of the iteration time. We know that the iteration time explains the time useful to finish a single interaction with the patient. Nevertheless, the introduction of the combination of implications has introduced a greater expenditure in terms of time as the number of steps performed by the framework increases. In particular, depends on the answer of the patient, the lists containing the symptoms 'Yes'(the patient is affected by it) or 'No (the patient isn't affected by it) grow constantly.

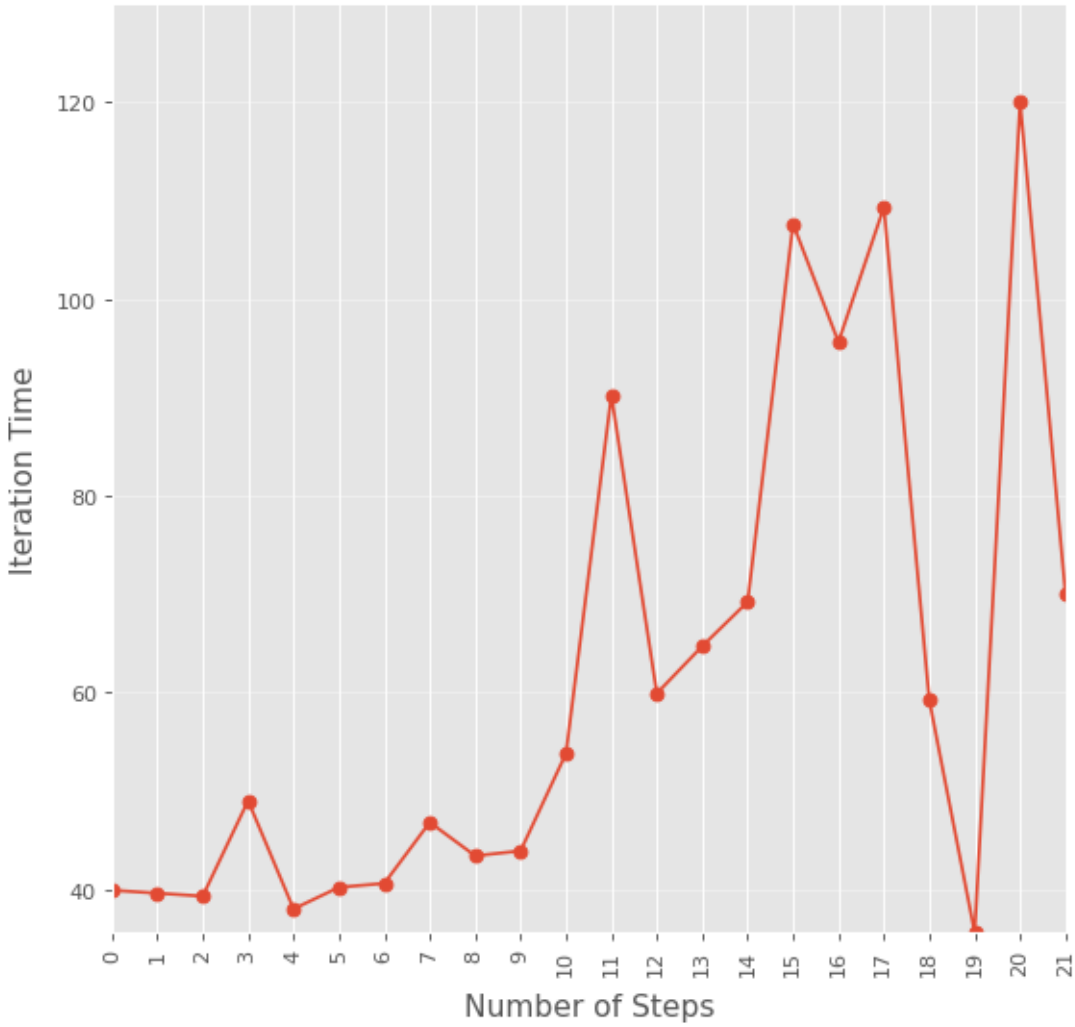

Figure 5.3: Comparison between the n°of steps and the trend of iteration time threshold degree = 0.75; single implication threshold = 0.4; implication combination threshold = 0.75 priority threshold = 0.4, gender threshold = 0.45

For example, we can observe that in the range between the eighteen and the nineteen the iteration time decrease radically, infact if we observe the figure 5 in this range the number of disease remain constant, this means that the framework asks to the patient a 'Yes' symptom and the number of these one is less than the number of 'No' symptoms, and this implies a minor number of implications. On the other hand, when we observe the next step(the nineteenth-twentieth), there is an increment of the time caused by the combinations between the 'No' symptoms. This fact is evident because in that range of steps there is another decrement of diseases shown in Figure 5.

5.2 Case study - Additional Risks

Another extensions of our framework is the capability of this one to recognize, starting from a already known disease about the patient, the other risks that could occur in the clinical history of the patient. In order to demonstrate the efficiency of our job we decided to examine a more than a randomical situation obtained during our tests. Starting from a set of patients already affected by a disease, through the procedure described in 4.6, the framework is able to estimate a group of diseases with a high probability of occurring in the patient over the years. The table shows the predictions (the best four additional risk-disease) of our framework on three different subject, each one described by different features and by a different disease.
Starting from the first situation, we have a male patient, hispanic and with an age belongs to a range of 30-44 years old. This one is affected by the *'Hyponatremia'*, described in [29], and the framework has estimated the *'Hypokalemia'* as the disease most likely to occur in the described subject. In the article [30] it was explained that in a lot of situations the symptoms of *'Hyponatremia'* they can also lead to association with *'Hypokalemia'*.

In the second situation, we have a male subject, white affected by *'Spermatocele'*. Furthermore, this diagnosis on a patient so old is confirmed by the article [31] that defines this disease 'usually identified in middleaged men, and its incidence increases with age'. As for the predictions concerning the further risks arising from the *'Spermatocele'*, we have to make two different considerations:

- The article confirms that the " *Varicoceles* occur secondary to the compression of the *Spermatocele*"; this means that the prevision of our framework about the *varicocele* is a good prediction.

- On the other hand we have as a consequence of the *spermatocele*, the occurrence of epididymitis that goes in opposite to the article that cites : 'spermatoceles may be found after vasectomy, inguinal herniorrhaphy, or epididymitis'.

These considerations for this result give us a a new point of reflection suggest us that the framework is capable to find the diseases more strictly correlated with the starting disease without be aware to identify always the actual subsequent illness. The last prediction deals with a Female patient, with an ethnicity not defined and with an age belons to a range 30-44 years old. This one is affected by *'Spherocytosis'*, described in [32] which mention also the related disorder linked to this disease. In according to this article,'Spherocytosis may arise as the result of immune disorders, toxic chemicals and drugs, alcohol abuese, antiviral agents (eg, ribavirin), physical damage, and infections'. This consideration confirms what has already been said previously, namely that our framework is not always able to ascertain the consequences following the patient's illness. In fact, sometimes he establishes what are the reasons and previous illnesses that could give life to his current situation. However, these results show us the usefulness of this framework which is able to provide a suggestion to

the doctor on the possible diseases from which the patient is affected or from which the patient could be affected in the future.

Patients' List				
Age	Gender	Ethnicity	Disease	Add. Risks
30-44Y	Male	Hispanic	Hyponatremia	Hypokalemia 0.9, Labyrinthitis 0.85, Hypovolemia 0.8, Peritonitis 0.85
45-59Y	Male	White	Spermatocele	Epididymitis 0.29, Varicocele 0.27, Abdominal hernia 0.26, Injury to the hip 0.25
30-44Y	Female	Other	Spherocytosis	Idiopathic excessive menstruation 0.9, Alcohol abuse 0.8, Preeclampsia 0.68, Poisoning due to antidepressants 0.54

Chapter 6

Conclusion and future works

In this work was presented an extension of an existing framework by Teixeira *et al.* [1], who proposed a goal-based framework that supports the development of intelligent agent dialogue within the health care domain. The extensions proposed here concern three main points: (i) to make it possible to also consider the *implications among the symptoms in the reasoning process*, (ii) to consider non binary weights associated to the symptoms, (iii) to propose a dataset filtering method in order to reduce the time needed to process the data, (iv) to make possible the analysis of future risks starting from subjects with an already known diagnostic status. The results obtained using the Symcat dataset are promising and confirm the validity of our proposal.

As a future work, Machine Learning (ML) techniques, which are techniques based primarily on statistics, can be used to build the model of medical behavior. In this way, we will have a hybrid framework based on both statistical techniques, which will allow us to extract some models from historical data, and symbolic-based techniques, which allows us to fully integrate the knowledge of experts, see for example [33]. Note that in literature, contributions on the use of Artificial Intelligence (AI) techniques are based on statistics [34] or symbolic [33].

Just to name a few statistics-based methods, Dahiwade *et al.*, for example, proposed in [35] a data mining approach to design a heart disease prediction system based on the symptoms of the patient. The approach by Kao *et al.* [36] aimed at integrating some contextual information in a reinforcement learning scheme, and the results show an improvement on accuracy of symptom checking over non-contextual-based approaches. The recent work by Alsuliman *et al.* [37] aims at providing an easy-to-follow overview to physicians, about the recent use of AI and ML techniques in several medical fields, including haematology, neurology, cardiology, oncology, radiology, ophthalmology, cell biology and cell therapy.

To sum up, one essentially agrees with Pauker *et al.* [11] who argued that any system actually able to provide a medical competence should use a judicious combination of categorical and probabilistic reasoning. As pointed out by Fieschi *et al.* in [10], categorical or logical reasoning is necessary to set the boundaries of the decision-making context, for example the links between diagnosis and symptoms, while probabilistic reasoning is necessary to address uncertainty, for example to address situations where there is uncertainty or ignorance about a symptom.Therefore, this is supposed to be the direction to follow in the future.

Bibliography

[1] Milene Santos Teixeira, Célia da Costa Pereira, and Mauro Dragoni. Information usefulness as a strategy for action selection in health dialogues. In *WI/IAT*, pages 323–330. IEEE, 2020.

[2] N. Krieger and E. Fee. Man-made medicine and women's health: The biopolitics of sex/gender and race/ethnicity. *International Journal of Health Services*, 24:265 – 283, 1994.

[3] Amanda Deeks, Catherine Lombard, and Helena Teede. The effects of gender and age on health related behaviors. *BMC Public Health*, 9(213):265 – 283, 2009.

[4] Symcat: Symptom-based, computer assisted triage. `http://www.symcat.com`, 2017.

[5] Clayton M. Christensen. *The Innovator's Dilemma: The Revolutionary Book that Will Change the Way You Do Business*. Harper Paperbacks, 01-2003.

[6] Ronald Andersen. Medicine's Dilemmas: Infinite Needs Versus Finite Resources. *JAMA*, 272(23):1870–1871, 12 1994. ISSN 0098-7484. doi: 10.1001/jama.1994.03520230080047. URL `https://doi.org/10.1001/jama.1994.03520230080047`.

[7] Hannah Semigran, Jeffrey Linder, Courtney Gidengil, and Ateev Mehrotra. Evaluation of symptom checkers for self diagnosis and

triage: Audit study. *BMJ (Clinical research ed.)*, 351:h3480, 07 2015. doi: 10.1136/bmj.h3480.

[8] A.S. MacNalty and M. Critchley. *Butterworths Medical Dictionary*. Butterworths, 1978.

[9] Robert S. Ledley and Lee B. Lusted. Reasoning foundations of medical diagnosis. *Science*, 130(3366):9–21, 1959.

[10] M. Fieschi and J. Gouvernet. Reasoning foundations of medical diagnosis revisited. *Yearbook of Medical Informatics*, pages 78–82, 1999.

[11] Stephen G. Pauker and Jerome P. Kassirer. Therapeutic decision making: A cost-benefit analysis. *N Engl J Med*, 293:229 – 234, 1975.

[12] Kai-Fu Tang. Inquire and diagnose : Neural symptom checking ensemble using deep reinforcement learning. 2016.

[13] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *CoRR*, abs/1811.12560, 2018. URL `http://arxiv.org/abs/1811.12560`.

[14] Richard Maclin and David W. Opitz. Popular ensemble methods: An empirical study. *CoRR*, abs/1106.0257, 2011. URL `http://arxiv.org/abs/1106.0257`.

[15] Hao-Cheng Kao, Kai-Fu Tang, and Edward Chang. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, Apr. 2018. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11902`.

[16] Andrew Levy, Robert Platt, and Kate Saenko. Hierarchical reinforcement learning with hindsight, 2019.

[17] Kangenbei Liao, Qianlong Liu, Zhongyu Wei, Baolin Peng, Qin Chen, Weijian Sun, and Xuanjing Huang. Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning, 2020.

[18] Laurence Cholvy and Célia da Costa Pereira. Usefulness of information for goal achievement. In *PRIMA*, volume 11873 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 2019.

[19] Milene Santos Teixeira, Célia da Costa Pereira, and Mauro Dragoni. A goal-based framework for supporting medical assistance: The case of chronic diseases. In *PRIMA*, volume 12568 of *Lecture Notes in Computer Science*, pages 281–298. Springer, 2020.

[20] Iqbal Sarker, Md Faruque, Ujjal Hossen, and Atikur Rahman. A survey of software development process models in software engineering. *International Journal of Software Engineering and its Applications*, 9:55–70, 11 2015. doi: 10.14257/ijseia.2015.9.11.05.

[21] Wilfred Van Casteren. The waterfall model and the agile methodologies : A comparison by project characteristics - short. 02 2017. doi: 10.13140/RG.2.2.10021.50403.

[22] Wesam Bhaya. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12:4102–4107, 09 2017. doi: 10.3923/jeasci.2017.4102.4107.

[23] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499. Morgan Kaufmann, 1994.

[24] Jiao Yabing. Research of an improved apriori algorithm in data mining association rules. *International Journal of Computer and Communication Engineering*, pages 25–27, 01 2013. doi: 10.7763/IJCCE.2013.V2.128.

[25] Manikandan S. Frequency distribution. *Journal of pharmacology & pharmacotherapeutics*, 2:54–6, 01 2011. doi: 10.4103/0976-500X.77120.

[26] David R. Butenhof. *Programming with POSIX Threads*. Addison-Wesley Longman Publishing Co., Inc., USA, 1997. ISBN 0201633922.

[27] Christian R. Halvorson, Matthew S Bremmer, and Stephen C. Jacobs. Polycystic kidney disease: inheritance, pathophysiology, prognosis, and treatment. *International Journal of Nephrology and Renovascular Disease*, 3:69 – 83, 2010.

[28] Carsten Bergmann, Jan Senderek, Marc Küpper, Frank Schneider, Christian Dornia, Ellen Windelen, Thomas Eggermann, Sabine Rudnik-Schöneborn, Jutta Kirfel, Laszlo Furu, Luiz Onuchic, Sandro Rossetti, Peter Harris, Stefan Somlo, Lisa Guay-Woodford, Gregory Germino, Markus Moser, Reinhard Büttner, and Klaus Zerres. Pkhd1 mutations in autosomal recessive polycystic kidney disease (arpkd). *Human mutation*, 23:453–63, 05 2004. doi: 10.1002/humu.20029.

[29] Ewout J. Hoorn and Robert Zietse. Diagnosis and treatment of hyponatremia: Compilation of the guidelines. *Journal of the American Society of Nephrology : JASN*, 28 5:1340–1349, 2017.

[30] Horacio J. Adrogué and Nicolaos E. Madias. The challenge of hyponatremia. *Journal of the American Society of Nephrology*, 23(7):1140–1148, 2012. ISSN 1046-6673. doi: 10.1681/ASN.

2012020128. URL `https://jasn.asnjournals.org/content/23/7/1140`.

[31] L. Erdem and BÜlent Akduman. Intratesticular varicocele concomitant with spermatocele. *Journal of Diagnostic Medical Sonography*, 20:440–443, 11 2004. doi: 10.1177/8756479304270562.

[32] Silverio Perrotta, Prof Gallagher, and Narla Mohandas. Hereditary spherocytosis. *Lancet*, 372:1411–26, 11 2008. doi: 10.1016/S0140-6736(08)61588-3.

[33] Mark A. Musen, Yuval Shahar, and Edward H. Shortliffe. *Clinical Decision-Support Systems*, pages 698–736. 2006.

[34] Adnan Qayyum, Junaid Qadir, Muhammad Bilal, and Ala I. Al-Fuqaha. Secure and robust machine learning for healthcare: A survey. *CoRR*, abs/2001.08103, 2020.

[35] Dhiraj Dahiwade, Gajanan Patle, and Ektaa Meshram. Designing disease prediction model using machine learning approach. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1211–1215, 2019.

[36] Hao-Cheng Kao, Kai-Fu Tang, and Edward Y. Chang. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In *AAAI*, pages 2305–2313. AAAI Press, 2018.

[37] Tamim Alsuliman, Dania Humaidan, and Layth Sliman. Machine learning and artificial intelligence in the service of medicine: Necessity or potentiality? *Current Research in Translational Medicine*, 68(4):245–251, 2020.

[38] Robert S. Ledley and Lee B. Lusted. Reasoning foundations of medical diagnosis. *Science*, 130(3366):9–21, 1959. ISSN 0036-8075. doi: 10.1126/science.130.3366.9. URL `https://science.sciencemag.org/content/130/3366/9`.

[39] L. Cholvy and C. D. C. Pereira. Usefulness of information for goal achievement. In *PRIMA*, 2019.

Ringraziamenti

Per aspera sic itur ad astra

Virgilio, Aen., IX 641

Nella mitologia greca si narrava che solo gli eroi capaci di compiere imprese eroiche in vita sarebbero stati destinati all'Olimpo; testimonianza del fatto che solo chi è in grado di affrontare le avversità è destinato a raggiungere la gloria.

Attraverso le asperità sino alle stelle, semplici parole risuonate spesso nella mia testa in questi anni e che mi hanno spronato a non arrendermi di fronte alle difficoltà, consapevole che alla fine di una strada tortuosa ci sarà sempre una cima da cui poter ammirare dall'alto tutto ciò che si è attraversato.

Oggi ufficialmente si conclude quello che anni fa definì *un percorso di vita, prima di un percorso di studi*, senza però rendermi conto che ero giunto solo a metà del mio tragitto.

Tuttavia, un viaggio così lungo sarebbe stato quasi impossibile da portare a termine senza il supporto e l'affetto delle persone che mi sono state accanto, di quelle persone che anche in caso di sbandata ti possano aiutare a tenere ben salde le redini indicandoti la direzione da seguire.

Durante questo percorso ho incontrato molte persone che hanno arricchito la mia vita, alcune appagate si sono fermate a metà strada, altre semplicemente hanno preso strade differenti, altre ancora invece, come una folata di vento in piena estate, sono state una boccata di aria fresca e niente più.

In questa sezione vorrei ringraziare chi mi ha concesso l'opportunità di percorrere questo viaggio, spesso non condividendo le mie tappe, ma sempre sostenenomi fiducioso delle mie scelte.

Grazie a chi mi ha dimostrato che la fine di un viaggio è solo l'inizio di un altro, che il cambiamento non è fallimento, bensì solo un nuovo punto di partenza.

Grazie a chi mi ha mostrato il coraggio di perseverare nelle proprie scelte, di rincorrere le proprie passioni senza piegarsi ai 'diktat' di una società che troppo spesso ci impone scelte obbligate.

Grazie a chi mi ha insegnato che la tenacia e la determinazione sono più forti di tutto, anche dei mali che la vita, a volte ingiustamente, ti pone davanti.

Grazie a coloro che mi hanno accompagnato nelle prime tappe di questo viaggio, parti integrante della mia crescita umana prima e professionale dopo, con cui ho condiviso gioie e dolori, risate e litigi, insegnandomi che l'unione fa la forza, che chiedere aiuto non è segno di debolezza, ma di intelligenza.

Grazie a chi è rimasto anche quando non sarebbe stato giusto farlo, a chi mi è stato accanto semplicemente amandomi, accarezzandomi l'anima e addolcendomi il cuore.

Grazie a coloro con cui ho condiviso questi ultimi due anni, che mi hanno accolto nella loro vita come parte integrante, mostrandomi che l'essere fratelli non è necessariamente vincolato ad un legame di sangue. A loro che nel momento del bisogno ci sono sempre stati: supportandomi, sostenendomi, accogliendomi.

Grazie a chi c'è sempre stato, ancora nella mia vita e spalla su cui appoggiarsi nei momenti di difficoltà.

Vorrei ringraziare tutti i miei amici: da quelli che porto nel mio cuore dall'infanzia a quelli che ho incontrato nel mio cammino e che hanno deciso di camminare al mio fianco e di portarmi fino a qui. A tutti voi, così diversi ma così importanti, ognuno per ragioni uniche e speciali, voglio esprimere la mia più assoluta gratitudine.

Grazie alla città di Bologna, che mi accolse da estraneo, e in cui oggi, invece, mi sento a casa; grazie all'Università e ai Professori tutti che mi hanno dato l'opportunità di mettermi sempre alla prova e di vivere esperienze che porterò nel cuore.