

POLITECNICO DI MILANO
V Facoltà di Ingegneria
Corso di laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione



UrHealth: UN SISTEMA ADATTIVO E
MULTI-DISPOSITIVO PER LA DIAGNOSI
MEDICA

Relatore: Prof. Maristella MATERA

Correlatore: Prof. Lucia VILELA LEITE FILGUEIRAS

Tesi di Laurea di:

Lorenzo LODI RIZZINI Matr. 755752

Anno Accademico 2011-2012

Innanzitutto, vorrei esprimere la mia gratitudine ai professori Maristella Matera e Lucia Filgueiras, per l'aiuto e il sostegno fornito durante la stesura del lavoro.

Colgo l'occasione inoltre per ringraziare Guilherme Fernandes Ribeiro, Tiago Martines e Cássio Scofield che hanno contribuito attivamente allo sviluppo di questo progetto.

Ringrazio con affetto la mia famiglia che mi ha sostenuto durante gli anni di università e mi ha sempre lasciato piena libertà per poter fare quello che realmente desideravo.

Infine ringrazio i miei amici Massimo, Federico, Nicola e Marco.

Indice

Elenco delle figure	VI
Elenco delle tabelle	IX
1 Introduzione	1
1.1 Stato dei sistemi di salute	1
1.2 Sistemi mobile	2
1.3 Processo di diagnosi medica	3
1.4 Struttura della tesi	4
2 Stato dell'arte	7
2.1 Software medici esistenti	7
2.2 Tecniche usate	8
2.2.1 Case Based Reasoning	8
2.2.2 Rule based expert systems	10
2.2.3 Probabilistic expert systems	11
2.2.4 Ontologie	12
3 Requisiti	15
3.1 Caratteristiche del sistema	15
3.1.1 Adattività	15
3.1.2 Usabilità	16
3.1.3 Scalabilità	16
3.1.4 Interoperabilità	17
3.1.5 Sicurezza	17
3.2 Descrizione funzionale del sistema	17

4	Architettura	21
4.1	Architettura di alto livello	21
4.1.1	DB server	22
4.1.2	Application server	23
4.1.3	Applicazione client medico	24
4.1.4	Applicazione client paziente	24
4.2	La base di dati	24
4.2.1	Fonti di informazione	24
4.2.2	Requisiti tecnologici	25
4.3	L'algoritmo di diagnosi	26
4.3.1	Tecnica utilizzata	26
4.3.2	Rete bayesiana	28
4.3.3	Algoritmo	30
4.3.4	Apprendimento accelerato	31
4.4	L'interfaccia utente	32
4.4.1	L'interfaccia con il paziente	32
4.4.2	L'interfaccia con il medico	33
5	Implementazione	35
5.1	Sistema di apprendimento automatico	35
5.2	Interfaccia con il medico	36
5.3	Interfaccia paziente	43
6	Risultati	49
6.1	Alcune prove pratiche	49
6.2	Test di usabilità con medici	55
7	Conclusioni e Sviluppi futuri	57
7.1	Analisi dei risultati	57
7.2	Sviluppi futuri	58
7.2.1	Miglioramenti all'algoritmo	58
7.2.2	Funzionalità aggiuntive	59
7.2.3	Miglioramento dell'interfaccia mobile	60
7.2.4	Inserimento della conoscenza medica	60

A	Tecnologie adottate	61
A.1	Rete bayesiana	61
A.2	Le ontologie	61
A.3	HTML5	62
A.4	Android	62
	Bibliografia	63

Elenco delle figure

1.1	Densità dei medici nel mondo (medici per 1.000 abitanti)	1
1.2	Diffusione degli smartphone nel mondo, volumi di vendite nel 2011	3
2.1	Ciclo del Case-Based Reasoning sviluppato da Aamodt	9
3.1	Casi di uso del sistema	18
3.2	Malattie possibili in ordine di probabilità	19
4.1	Diagramma dell'architettura di comunicazione	22
4.2	Schema entità relazioni della base di dati	23
4.3	Esempio di rete bayesiana a due livelli	29
4.4	Esempio di tabella di probabilità della malattia influenza	29
5.1	Pagina per la gestione dei pazienti in cura	37
5.2	Pagina per la gestione dei sintomi del paziente	39
5.3	Pagina per la visualizzazione dei suggerimenti di diagnosi e per l'inserimento della diagnosi	40
5.4	Pagina per l'inserimento di diagnosi custom	41
5.5	Inserimento di diagnosi custom	42
5.6	Menu dell'applicazione mobile	43
5.7	Selezione di sintomi	44
5.8	Inserimento di un sintomo	46
5.9	Visualizzazione dello storico di sintomi inseriti	47
5.10	Concessione di permessi ai medici	48
6.1	Rete bayesiana dopo l'inserzione della prima diagnosi di test	50
6.2	Suggerimento di diagnosi del sistema	50

6.3	Rete bayesiana dopo l'inserzione della seconda diagnosi di test . .	51
6.4	Suggerimento di diagnosi del sistema	52
6.5	Rete bayesiana dopo l'inserzione della terza diagnosi di test . . .	53
6.6	Rete bayesiana dopo l'inserzione della quarta diagnosi di test . .	53
6.7	Suggerimento di diagnosi del sistema	54
6.8	Rete bayesiana dopo l'inserimento della quarta diagnosi di test . .	55
6.9	Esempio di interfaccia grafica per l'inserimento di sintomi	56

Elenco delle tabelle

2.1	Tecniche di reasoning a confronto	12
6.1	Prima diagnosi di test	50
6.2	Seconda diagnosi di test	51
6.3	Terza diagnosi di test	52
6.4	Quarta diagnosi di test	53
6.5	Quinta diagnosi di test	54

Capitolo 1

Introduzione

1.1 Stato dei sistemi di salute

Lo stato attuale dei sistemi di salute nel mondo prevede che ogni medico abbia in cura varie centinaia di pazienti[24]. Nei Paesi poveri o in via di sviluppo questo rapporto cresce fino ad arrivare a diverse migliaia di pazienti per medico. Questa situazione rende la sanità pubblica in questi Paesi inadeguata ad un trattamento medico soddisfacente della popolazione, sia per la mancanza, appunto, di personale medico che per il sovraffollamento delle strutture mediche esistenti.

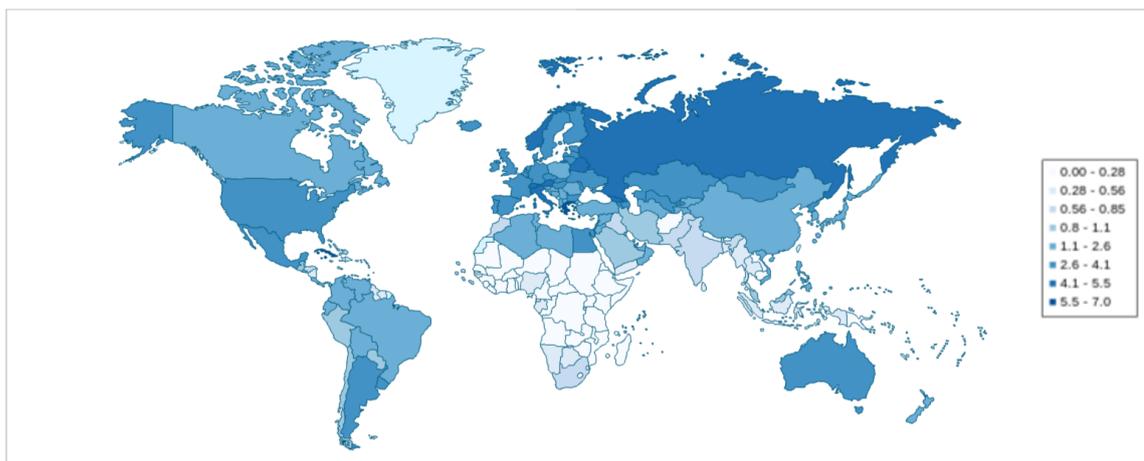


Figura 1.1: Densità dei medici nel mondo (medici per 1.000 abitanti)

Spesso è difficile porre rimedio in breve termine a questa situazione di in-

sufficienza di risorse a causa di motivazioni economiche e sociali. Per cercare di migliorare il servizio medico reso ai pazienti in questi contesti di scarsità di risorse, occorre trovare un modo di aumentare l'efficienza del personale medico esistente e delle strutture utilizzate. Un approccio a questo problema potrebbe essere l'introduzione di tecniche di telemedicina, in modo da richiedere la presenza dei pazienti presso le strutture mediche solo in casi estremi, evitando così il sovraffollamento delle stesse. Esistono vari sistemi di questo tipo, già in uno stato di sviluppo sufficientemente maturo, che attualmente vengono utilizzati con successo in alcuni contesti più sviluppati. Un altro approccio è quello di aiutare il personale medico nelle attività quotidiane in modo da migliorare la quantità e la qualità del lavoro svolto. Una delle attività in cui sono richiesti particolare attenzione e tempo da parte di personale medico qualificato è l'elaborazione di diagnosi.

1.2 Sistemi mobile

I sistemi mobile, smartphone e tablet, sono sempre più diffusi e la loro presenza si può già considerare pervasiva nella vita quotidiana di molte persone. In Europa una persona su tre utilizza quotidianamente un dispositivo mobile e questa tendenza è in crescita con previsione di raggiungere la metà della popolazione nel 2015[25]; negli Stati Uniti la diffusione è ancora maggiore.

Sono già presenti sul mercato alcune applicazioni che permettono di usare i dispositivi mobili per l'inserimento di dati medici e per la gestione di una cartella clinica digitale come ad esempio iTriage[26]. L'utilizzo di questi dispositivi per la registrazione di sintomi porta ad alcuni benefici che riguardano sia la qualità che la quantità delle osservazioni mediche registrate. Realizzando una apposita applicazione per l'inserimento di osservazioni mediche da dispositivo mobile si rende possibile l'inserimento di sintomi in qualsiasi luogo e in qualsiasi momento. Questo permette un monitoraggio dei pazienti più preciso. Tutte le volte che un paziente rileva un sintomo gli è possibile inserirlo nell'applicazione direttamente nel momento in cui questo avviene, senza dover disporre necessariamente di un PC. Questo permette di registrare tutti i sintomi che occorrono, evitando perdite dovute a dimenticanze e registrando anche l'esatto momento in cui sono avvenute. Questa maggiore completezza di informazioni può essere sfruttata sia dal perso-

1.3. Processo di diagnosi medica

nale medico che ha in cura il paziente, sia da eventuale software di supporto alla diagnosi che può utilizzare queste informazioni, rese disponibili automaticamente al sistema, per effettuare la diagnosi con maggiore precisione. L'applicazione mobile può, tra l'altro, gestire e rendere disponibile in qualsiasi momento un riassunto della situazione medica del paziente. Questo permette al paziente di mostrare tutti i dati medici più rilevanti in qualsiasi struttura medica si trovi in cura.

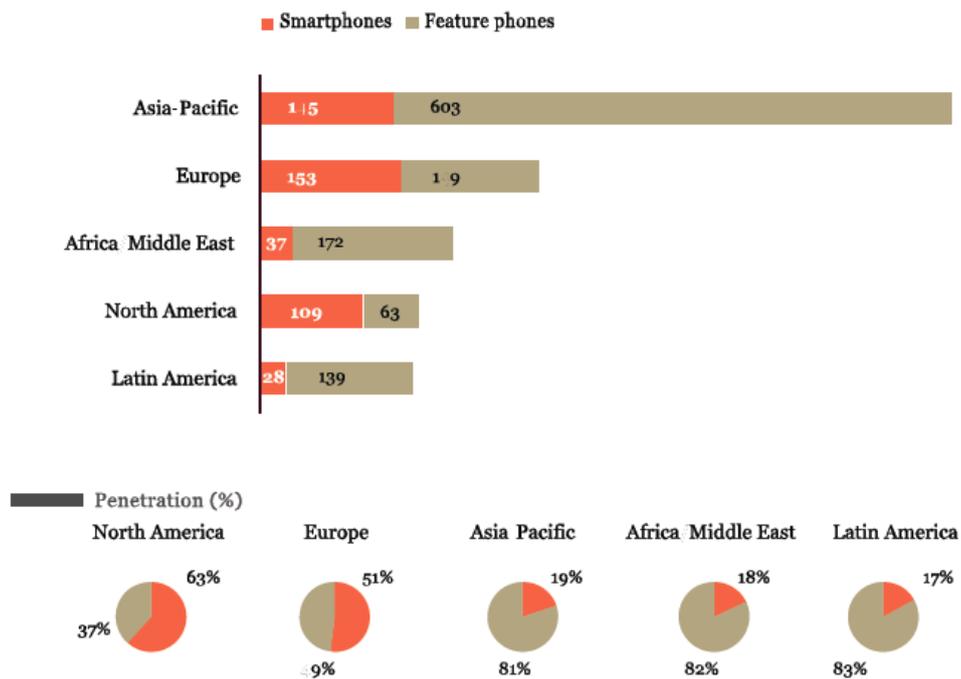


Figura 1.2: Diffusione degli smartphone nel mondo, volumi di vendite nel 2011

1.3 Processo di diagnosi medica

Il processo di diagnosi medica è una attività delicata che richiede al medico la gestione e la padronanza di grandi quantità di informazioni e conoscenza medica[1]. Se si considerano le condizioni precarie sopra descritte, nelle quali i medici sono costretti ad operare, ci si accorge che diventa umanamente impossibile riuscire a gestire tutta questa conoscenza e mantenere il livello di attenzione richiesto, in modo continuativo, per svolgere questa attività in modo efficace. Nel

processo di diagnosi, tra l'altro, è richiesta una grande esperienza ai medici che devono considerare tutte le possibili corrispondenze fra i sintomi presentati dal paziente e le potenziali malattie correlate, selezionare quali siano più probabili e decidere quali ulteriori esami eseguire. Per il personale medico che si è formato da poco tempo, spesso questa esperienza è scarsa e il tempo di apprendimento richiesto per riuscire a raggiungere un livello di esperienza sufficiente è ovviamente di diversi anni.

Per questi motivi si è pensato di concentrare gli sforzi nello sviluppo di una soluzione che supporti questo processo utilizzando tecnologie di intelligenza artificiale e ingegneria della conoscenza. Riuscendo a supportare il processo di diagnosi medica, ci si è posti come obiettivo non solo l'aumento della quantità di pazienti che un medico riesce a seguire in modo soddisfacente, incrementando quindi l'efficienza, ma anche il miglioramento della qualità delle diagnosi elaborate. Uno degli obiettivi del sistema descritto in questo lavoro è infatti quello di migliorare l'efficacia delle diagnosi creando un sistema di supporto a questa attività, che proponga suggerimenti utili al medico riassumendo la conoscenza raccolta attraverso l'analisi dell'operato di altri medici.

Questo lavoro è stato sviluppato presso l'*Escola Politécnica da USP* a San Paolo del Brasile durante un intercambio di doppia laurea. La parte di interfaccia del sistema è stata mostrata a personale medico locale per avere un riscontro sull'usabilità del sistema e poter apportare eventuali miglioramenti in tal senso.

1.4 Struttura della tesi

In questo paragrafo si descrive brevemente la struttura che si intende dare alla presente trattazione. Ecco un elenco e una breve descrizione di tutti i capitoli:

- **Capitolo 2** - in questo capitolo viene descritto lo stato dell'arte, cioè i software medici già esistenti e le tecniche utilizzate, che costituiscono il background della presente tesi;
- **Capitolo 3** - il capitolo terzo si occupa di descrivere nel dettaglio gli obiettivi che hanno guidato lo sviluppo di questo progetto così come le linee guida che hanno influito sull'implementazione dello stesso;

- **Capitolo 4** - in questo capitolo ci si dedica invece a descrivere il funzionamento e la struttura delle varie parti che compongono il sistema, l'algoritmo di diagnosi, la struttura di dati utilizzata e l'interfaccia con l'utente.
- **Capitolo 5** - questa sezione è dedicata invece, a descrivere l'implementazione delle parti principali che costituiscono il sistema, alle tecnologie utilizzate e alle varie scelte implementative prese;
- **Capitolo 6** - questo capitolo parla delle prove svolte e dei risultati ottenuti con un breve commento sul significato dei test svolti;
- **Capitolo 7** - si conclude con un'ultima parte dedicata alle conclusioni che si possono trarre dal lavoro svolto e sui possibili sviluppi futuri che il sistema implementato può avere.
- **Appendice A** - questa appendice raccoglie una breve rassegna delle tecnologie più importanti utilizzate nel progetto.

Capitolo 2

Stato dell'arte

2.1 Software medici esistenti

Nel settore dell'eHealth sono stati sviluppati molti sistemi medici con varie funzionalità e obiettivi. Molti sistemi si pongono come obiettivo la capacità di fornire informazioni mediche, o aiutare i medici nel monitorare più pazienti, a distanza. Tali sistemi si basano sull'utilizzo di dispositivi mobile e sulla loro integrazione con sensori e con l'infrastruttura di rete già esistente. Un esempio di sistema di questo tipo è MASPortal[11] che si pone appunto l'obiettivo di utilizzare l'infrastruttura di comunicazione grid e integrarla con sensori medici, usati direttamente a casa del paziente, per rendere pervasivo l'accesso ai servizi medici grid. Questo tipo di sistemi ha il vantaggio di permettere un miglior accesso ai servizi medici da parte di pazienti e di fornire e organizzare i dati medici di vari pazienti in modo ottimale per il personale medico che li ha in cura. Un'altra funzionalità utile è l'automatizzazione di alcune procedure mediche come il rilevamento di parametri fisiologici attraverso sensori e la gestione di emergenze. Come descritto in precedenza, questo genere di sistemi pone come obiettivo principale una miglior organizzazione nella gestione e comunicazione dei dati medici utilizzando le tecnologie informatiche e di comunicazione già esistenti.

Un'altra tipologia di sistemi riguarda il supporto alla diagnosi medica. Questi software hanno come obiettivo principale quello di fornire un sistema di ausilio al processo di diagnosi medica. Analizzando i dati a disposizione su un paziente e avendo a disposizione una base di conoscenza medica con un algoritmo di inferen-

za, questi software riescono a considerare una vasta gamma di possibili malattie correlate ai sintomi presentati e a indicare quali di queste siano più probabili. Questa tipologia di software è molto utile soprattutto considerando la vasta quantità di informazioni che un medico deve gestire per riuscire ad effettuare una diagnosi accurata[10]. La sistematicità e la completezza della base di conoscenza utilizzata da questi software sono gli elementi su cui si basa un processo di diagnosi automatizzato che prende in considerazione tutte le possibilità, diversamente da quanto un essere umano, per quanto esperto e competente, possa garantire. È bene ricordare che questi sistemi hanno sempre come obiettivo l'ausilio alla diagnosi (che, comunque, viene sempre effettuata da un medico) e mai la sua sostituzione. Sistemi di questo tipo sono ad esempio *Medical Knowledge-Based System for Diagnosis from Symptoms and Signs*[12] o Hepar II[18]. Questi sistemi hanno appunto come obiettivo l'implementazione di un sistema che realizzi un processo di diagnosi attraverso tecniche di inferenza applicate sulla conoscenza disponibile. Come è possibile leggere nell'articolo *Medical Knowledge-Based System for Diagnosis from Symptoms and Signs*[12] in letteratura esistono vari sistemi di questo tipo che si dedicano a settori specifici della medicina. Questi sistemi, dedicandosi appunto ad aree di medicina abbastanza ristrette, riescono ad offrire prestazioni con una buona accuratezza. La maggior parte di questi sistemi utilizza però tecniche e standard proprietari, rendendo difficile una possibile interoperabilità fra di loro e fra istituzioni mediche differenti, rendendo il loro impiego molto specifico e la loro adozione da parte della comunità medica difficile.

2.2 Tecniche usate

Ciascun software di supporto alla diagnosi utilizza tecniche differenti per implementare l'algoritmo di inferenza e per gestire la base di conoscenza su cui le diagnosi sono basate.

2.2.1 Case Based Reasoning

Il Cased-Based Reasoning (CBR) è una tecnica utilizzata in vari domini con sistemi basati su basi di conoscenza. Il Case Based Reasoning si basa sull'analisi di

2.2.1. Case Based Reasoning

problemi, simili a quello analizzato, già risolti e sull'adattamento della soluzione di questi problemi al problema corrente. L'idea sottostante è che problemi simili hanno soluzioni simili. Nonostante questa idea non sia sempre vera è comunque applicabile su molti domini. Il CBR si scompone quindi in due parti. La prima parte si prefigge l'obiettivo di trovare i casi simili al caso corrente. Ciò può essere ottimizzato utilizzando apposite strutture di dati e indicizzazione. La seconda parte si occupa di adattare le soluzioni dei problemi simili al problema corrente. Questo adattamento dipende molto dal dominio in cui l'algoritmo è utilizzato.

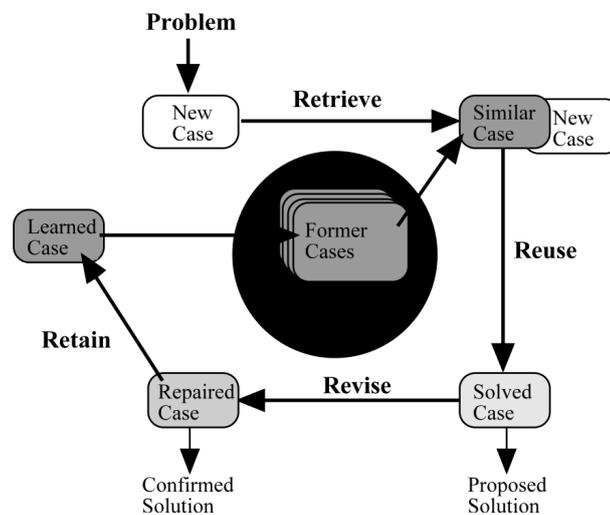


Figura 2.1: Ciclo del Case-Based Reasoning sviluppato da Aamodt

Questa tecnica presenta aspetti che renderebbero utile un suo impiego in un software di ausilio alla diagnosi. In campo medico la conoscenza non consiste solamente in regole ma è composta da nozioni e esperienza. L'esperienza deriva dai casi che si è avuto modo di studiare e i medici ricorrono a questo tipo di conoscenza quando devono effettuare delle diagnosi. Quindi gli argomenti a favore dell'utilizzo di questa tecnica nella diagnosi medica sono:

- Il reasoning effettuato con questa tecnica corrisponde a quanto avviene in realtà nell'approccio alla diagnosi utilizzato dai medici
- Incorporare nuovi casi significa aggiungere automaticamente conoscenza al sistema

- Questa tecnica permette di dividere la conoscenza dalle nozioni, permettendo di sviluppare tecniche diverse per la gestione di ciascuna
- È relativamente facile incorporare al sistema lo storico di diagnosi svolte da una struttura medica, che così entra a far parte della conoscenza del sistema

Il problema principale di questa tecnica, però, è la difficoltà di adattare le soluzioni dei casi simili al caso corrente. Specialmente nel dominio medico questa funzione è difficile da calcolare e varia molto spesso da caso a caso rendendo estremamente difficile riuscire a definire tutte le regole necessarie a prendere in considerazione tutte le differenze possibili[13]. Per questa ragione vari software che usano questa tecnica si limitano solo alla prima parte dell'algoritmo, ovvero alla ricerca di casi simili che sono proposti al medico come suggerimento lasciando allo stesso il compito di adattare la soluzione di questi casi al caso in analisi.

2.2.2 Rule based expert systems

I sistemi basati su regole hanno lo scopo di catturare la conoscenza e l'esperienza umana sotto forma di regole del tipo SE *condizione* ALLORA *azione*. Ci sono prove psicologiche che la metodologia di pensiero umana possa essere modellata in questo modo[19]. Avendo a disposizione tutte le regole che guidano gli esperti nel prendere le decisioni su un determinato dominio, si può pensare di replicare il loro metodo di risolvere i problemi in modo automatizzato. Questo tipo di approccio mette in relazione diretta i sintomi con le malattie attraverso le regole che definiscono le relazioni fra di loro. Il sistema così definito, quando interrogato, fornisce risposte digitali non considerando gli aspetti probabilistici che potrebbero intercorrere tra sintomi e malattie. Se una regola è verificata, allora automaticamente la malattia prevista dalla regola in questione è presente, in caso contrario no. Questa strutturazione rigida e le relazioni dirette che intercorrono tra sintomi e malattie permettono di effettuare facilmente il test del modello seguendo le tracce del reasoner attraverso le regole. Mentre la fase di testing è facile, la fase di affinazione del modello è molto lunga e impegnativa. Se è vero che è semplice agire direttamente sul modello, introducendo nuove regole, è anche vero che per riuscire a raggiungere un affinamento e una completezza soddisfacenti

può essere necessario introdurre una quantità di regole notevole. Dipendendo dal dominio in questione, se molto complesso come quello medico, il lavoro può essere irrealizzabile.

2.2.3 Probabilistic expert systems

I sistemi esperti di tipo probabilistico derivano da ricerche effettuate in statistica e in intelligenza artificiale. Queste ricerche si sono concentrate su concetti di rilevanza e indipendenza probabilistica e hanno portato allo sviluppo di strumenti grafici intuitivi ed efficienti, utili nella rappresentazione e modellazione della conoscenza. Uno degli strumenti di questo tipo principali sono le *Reti bayesiane*. Le reti bayesiane, che hanno fra gli obiettivi anche la capacità di catturare la conoscenza degli esperti, sono basate sui fondamenti matematici della teoria delle probabilità. A differenza dei sistemi basati su regole, i sistemi probabilistici mettono in relazione indirettamente sintomi e malattie modellando le relazioni probabilistiche tra i vari elementi del dominio. Un'altra grande differenza rispetto ai sistemi basati su regole è quindi il tipo di risultato ottenuto, che in questo caso considera l'aspetto probabilistico delle relazioni e rispecchia questo aspetto nel risultato fornito dal reasoning. Nonostante il modello sia costruito indirettamente, modellando solo le relazioni probabilistiche, se si ha una buona conoscenza del sistema è possibile ottenere comunque un buon livello di accuratezza. Mentre può essere difficile testare un sistema basato su reti bayesiane o modificarlo puntualmente, in modo diretto, specialmente se la topologia della rete è particolarmente complessa, è relativamente facile riuscire a costruire una rete partendo da dati esistenti. Utilizzando dati storici è infatti possibile, utilizzando la teoria delle probabilità, ricostruire quali sono le relazioni di dipendenza e indipendenza fra i fattori rilevanti del modello e i parametri di probabilità coinvolti. Si può ritenere quindi che avendo una quantità sufficientemente grande di dati storici del sistema che si desidera modellare e applicando i corretti strumenti di calcolo delle probabilità, si possa ottenere una rete bayesiana di partenza con un buon livello di accuratezza e sufficientemente affinata.

	Cased based reasoning	Rule based expert systems	Probabilistic expert systems
Tipo di modellazione	Diretta	Diretta	Indiretta
Raggiungimento completezza	Difficoltà media	Molto difficile	Difficoltà media
Manutenibilità	Facile	Facile	Difficile
Utilizzo di dati storici	Si	No	Si
Considera aspetti probabilistici	No	No	Si
Accuratezza	Buona	Media	Buona
Problemi	Difficoltà ad adattare soluzioni	Impossibile raggiungere la completezza	Difficile da mantenere direttamente

Tabella 2.1: Tecniche di reasoning a confronto

2.2.4 Ontologie

Un'ontologia è una rappresentazione formale, fatta attraverso l'utilizzo di logica del primo ordine, del modello di un dominio di interesse. Questo tipo di rappresentazione è utilizzata nel campo dell'intelligenza artificiale e della rappresentazione delle conoscenze. Le ontologie sono utili per vari scopi come ad esempio il ragionamento induttivo, usando appunto la logica del primo ordine, o per la classificazione dei concetti del dominio di interesse. La classificazione in una ontologia avviene in modo gerarchico. Esistono ontologie che, grazie alla capacità di classificare conoscenze, vengono utilizzate come glossario di base. Queste ontologie, chiamate ontologie fondazionali, sono utili per definire una base comune di conoscenza comune da utilizzare per poter comunicare senza ambiguità. Riguardo al dominio considerato dalle ontologie esistono ontologie top-level, ossia di massimo livello, e ontologie di dominio. Per quanto riguarda l'utilizzo di ontologie in software di diagnosi medica ci sono vari utilizzi. Alcuni software utilizzano la caratteristica di ragionamento deduttivo implementato con logica del primo ordine per effettuare direttamente il processo di diagnosi[14]. Altri software utilizzano le ontologie solo per classificare e gestire le nozioni utilizzate lasciando la parte di implementazione dell'algoritmo di diagnosi ad altre tecniche, per esempio quelle analizzate precedentemente. Riguardo al secondo tipo di impiego delle ontologie in campo diagnostico sono disponibili varie ontologie riguardanti il dominio biologico e medico che hanno come obiettivo la classificazione dei concetti e delle nozioni

2.2.4. Ontologie

base del dominio[2]. Queste ontologie sono normalmente mantenute da una comunità di ricercatori in campo medico e biologico che si occupa di aumentare la completezza delle stesse aggiungendo nuovi concetti in modo coerente con la loro struttura.

Capitolo 3

Requisiti

3.1 Caratteristiche del sistema

Questa sezione ha lo scopo di illustrare le caratteristiche scelte per il sistema. Queste caratteristiche sono state individuate attraverso le riunioni del team di sviluppo e con il supporto della professoressa Filgueiras per la parte di interfaccia utente.

3.1.1 Adattività

Un obiettivo del sistema sviluppato in questo progetto di tesi è l'adattabilità. L'adattabilità è la caratteristica che permette al sistema di imparare dall'esperienza accumulata fornendo diagnosi con una sempre maggiore accuratezza. Questo aspetto è importante perché permette di simulare il processo di apprendimento di vari medici e rendere disponibile questa conoscenza in modo esatto e completo ogni qual volta sia necessario fare una diagnosi. L'algoritmo di diagnosi deve quindi implementare quello che si definisce un sistema esperto. In ogni sistema esperto è prevista una fase di apprendimento iniziale inevitabile. Durante questa fase il sistema non è di alcuna utilità nel suggerire diagnosi. Per ridurre la durata di questa fase e incentivare quindi l'adozione del sistema da più strutture mediche e da più medici, il nostro sistema oltre ad imparare dall'esperienza accumulata, analizzando tutte le diagnosi effettuate ed inserite, deve poter permettere di accelerare il processo di apprendimento manualmente in modo da migliorare le sue

performance in modo rapido. Questa caratteristica è importante anche per inserire la conoscenza relativa a malattie rare o con scarsa probabilità di essere incontrate. In questo modo, infatti, non è necessario aspettare il verificarsi di una malattia per poter inserire conoscenza a riguardo nel sistema.

3.1.2 Usabilità

Uno degli aspetti principali del sistema che si è deciso di implementare è l'usabilità intesa come la facilità di utilizzo del sistema. La semplicità e la facilità di utilizzo del sistema sono obiettivi fondamentali, considerando soprattutto una probabile resistenza iniziale nell'adozione del sistema, difficoltà che può essere superata solamente attraverso l'utilizzo di una interfaccia accattivante, funzionale e semplice. Ci si accorge che questo aspetto va attentamente analizzato se si pensa a quali saranno gli utenti tipici del sistema e le condizioni in cui il sistema verrà utilizzato. Si sono identificate due categorie di utenti: i medici e i pazienti. Si è scelto di sviluppare una applicazione specifica per ciascuna categoria definendo quali requisiti specifici ogni applicazione debba avere. Per i medici deve essere sviluppata una applicazione con una interfaccia immediata e semplice che permetta di gestire i pazienti in cura, i loro sintomi, le loro diagnosi e, nel caso di medici qualificati, di inserire conoscenza nel sistema.

Nel caso dell'applicazione per pazienti, deve essere sviluppata una applicazione basica che permetta di inserire rapidamente i sintomi percepiti in modo da facilitare e incoraggiare il suo utilizzo da parte dei pazienti. Si è scelto di sviluppare una applicazione mobile per i pazienti in modo da permettere loro l'inserimento dei sintomi direttamente quando essi sono percepiti. In questo modo i pazienti, nel caso non siano in prossimità di un computer, non devono ricordarsi i sintomi ma possono utilizzare direttamente il loro dispositivo mobile per annotare questo tipo di informazione.

3.1.3 Scalabilità

Una volta che il sistema è entrato in uso, è lecito aspettarsi di avere una quantità sempre maggiore di informazioni da dover gestire. Il sistema deve poter garantire di saper sfruttare tutta l'esperienza assimilata nel modo migliore e di

poter rispondere rapidamente alle richieste di diagnosi che gli vengono inoltrate. Il sistema deve pertanto assicurare buone caratteristiche di scalabilità, ovvero le sue performance devono mantenersi buone anche aumentando la quantità di dati da gestire e le richieste effettuate. E' necessario quindi implementare un algoritmo performante e prevedere la possibilità di utilizzare più piattaforme hardware o servizi di tipo cloud flessibili. A tale scopo è indispensabile ottenere una buona portabilità del codice e l'utilizzo di tecnologie diffuse e standard.

3.1.4 Interoperabilità

Il sistema deve inoltre permettere la contribuzione di informazione da parte di più strutture mediche e di vari medici. Tale obiettivo si raggiunge tramite l'interoperabilità del sistema. Questa caratteristica può essere assicurata utilizzando standard definiti dalla comunità medica per definire gli elementi base del sistema e garantire quindi la stessa base di conoscenza condivisa con cui descrivere la conoscenza che il sistema utilizzerà.

3.1.5 Sicurezza

Un altro aspetto importante che il sistema deve considerare è la sicurezza. In particolare il sistema deve garantire l'accesso alle informazioni del paziente solamente da parte dei medici curanti. Il sistema inoltre deve permettere il salvataggio dei dati personali e medici dei pazienti in un sottosistema locale che può essere situato all'interno della struttura medica. Questo per soddisfare le normative legali riguardanti la gestione di dati personali sensibili.

3.2 Descrizione funzionale del sistema

Dopo aver definito gli obiettivi e le linee guida che devono caratterizzare lo sviluppo del sistema sono stati definiti i requisiti funzionali dello stesso.

- Il sistema deve consentire l'inserimento di uno o più sintomi scelti da una lista o cercati attraverso un apposito campo:

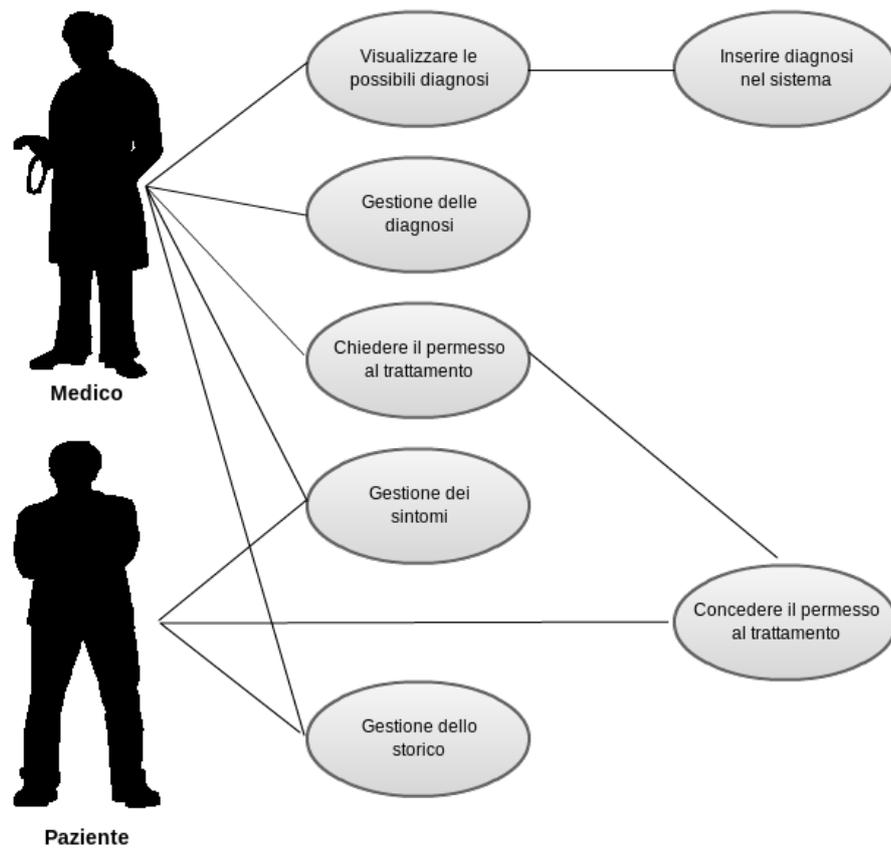


Figura 3.1: Casi di uso del sistema

- Il sistema deve mostrare al medico un elenco di malattie, compatibili con i sintomi descritti, in ordine di pertinenza indicando, anche graficamente, la probabilità di ciascuna.



Figura 3.2: Malattie possibili in ordine di probabilità

- Il sistema deve mantenere uno storico di tutte le diagnosi realizzate che deve essere visibile per il medico che le ha effettuate.
- Il medico deve poter scegliere fra le opzioni di diagnosi calcolate dal sistema o inserire manualmente una diagnosi relativa ai sintomi considerati.
- Il medico deve poter chiedere il permesso ad accedere ai dati relativi a un paziente e il paziente deve poter concederglielo.
- I dati relativi ai pazienti devono poter essere conservati in una base di dati locale in modo indipendente dai dati necessari all'algoritmo di diagnosi.
- Il sistema deve considerare le informazioni storiche presenti nel database relative alle diagnosi precedenti effettuate per migliorare i parametri dell'algoritmo di diagnosi ed ottimizzare così l'accuratezza dei suggerimenti forniti.
- Il sistema deve consentire l'inserimento, da parte di personale qualificato, di conoscenza sulle relazioni tra sintomi e malattie in modo da poter accelerare l'apprendimento del sistema.
- Il sistema deve prevedere la contribuzione alla creazione della conoscenza necessaria all'algoritmo da parte di varie istituzioni mediche.

- Il sistema deve agevolare l'inserimento di regole per poter classificare sintomi con caratteristiche quantitative in sintomi qualitativi. Ad esempio deve essere possibile inserire una regola che classifica la febbre di 39° come febbre alta.
- Il sistema deve fornire un indice di affidabilità delle diagnosi suggerite basato sulla quantità di dati presenti nello storico che supportano la diagnosi effettuata.
- Il sistema deve poter considerare la caratteristica temporale delle osservazioni mediche registrate, per fornire una diagnosi più precisa.
- Il sistema deve poter considerare la localizzazione geografica del paziente tra i fattori rilevanti per ottenere una diagnosi più precisa.

Capitolo 4

Architettura

Questo capitolo ha come obiettivo quello di definire come sono state specificate le varie parti che compongono il sistema, la scelta degli algoritmi e delle strutture di dati utilizzate così come le motivazioni che hanno determinato la definizione di queste parti.

4.1 Architettura di alto livello

L'architettura scelta è la comune architettura a tre livelli, che viene di buona norma utilizzata in situazioni analoghe alla nostra. Nell'architettura a tre livelli (detta anche *thin client*) il client non comunica direttamente con il server del database ma con un server dell'applicazione. In questo modo il client svolge solo il compito di interfaccia utente e la logica dell'applicazione viene inserita nell'*application server*. Questa soluzione è sicuramente più modulare: se si modifica la base di dati sottostante, il server dell'applicazione richiede a sua volta delle modifiche, ma l'interfaccia utente può anche restare invariata. Questa architettura inoltre presenta vantaggi dal punto di vista della sicurezza: inserendo un layer tra i client e i dati è possibile gestire policy di accesso ai dati in modo più dettagliato e flessibile garantendo un livello di sicurezza consono ai requisiti di un sistema informatico medico. La scelta dei linguaggi utilizzati, dei framework e delle piattaforme è stata guidata dalla ricerca della portabilità del codice e dall'utilizzo, quando possibile, di sistemi opensource o comunque gratuiti in modo da non dover dipendere da terzi e da non richiedere costi pesanti in licenze software

che potrebbero pesare sull'adozione del sistema da parte di istituti medici che solitamente devono convivere con budget ristretti.

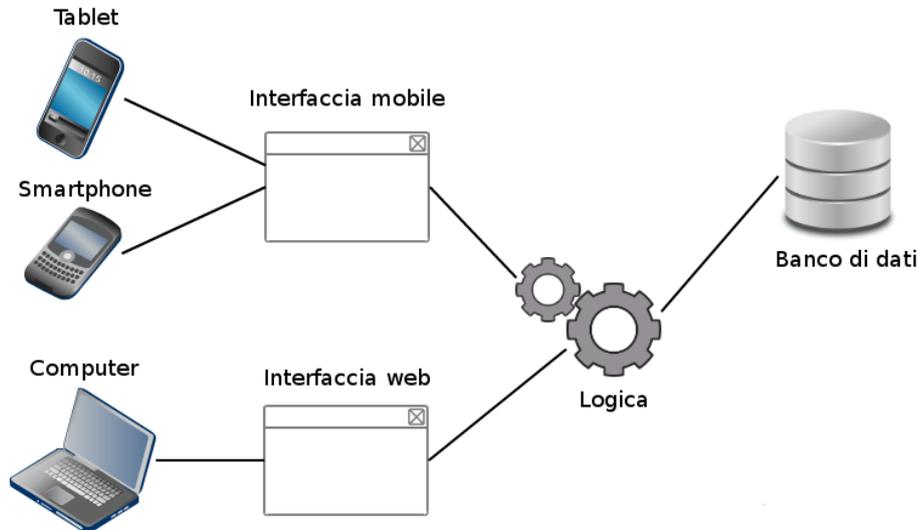


Figura 4.1: Diagramma dell'architettura di comunicazione

4.1.1 DB server

Il server che gestisce la base di dati è stato implementato usando MySQL. Questo DBMS gratuito permette di implementare un'efficiente base di dati relazionale. Nel database sono presenti le informazioni sui pazienti, sui sintomi e le malattie e le informazioni della rete bayesiana che rappresenta l'esperienza medica del sistema. Ai fini di test del sistema per praticità tutte queste informazioni sono state inserite nello stesso database. È possibile però notare, analizzando la struttura della base di dati (Figura 4.2), che le due parti fondamentali del database: informazioni sui pazienti e rete bayesiana, sono facilmente separabili in due database. Questo permetterebbe di usare due db server diversi per le due tipologie di informazioni: per esempio mantenendo i dati medici dei pazienti in un server locale, interno alla struttura medica e mettendo i dati della rete bayesiana in un servizio di cloud pubblico. Questo renderebbe possibile mantenere un alto

4.1.2. Application server

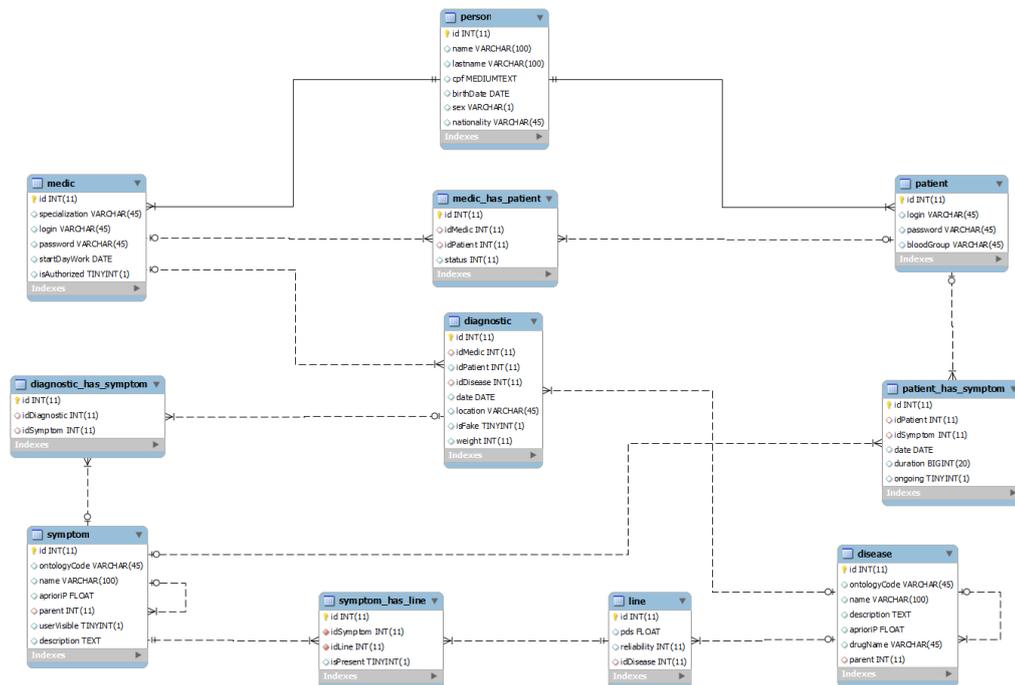


Figura 4.2: Schema entità relazioni della base di dati

livello di sicurezza sulle informazioni sensibili relative ai pazienti e un ottimo livello di performance sull'accesso ai dati (anonimi) della rete bayesiana oltre alla possibilità di condividere i dati dell'esperienza medica del sistema in modo facile permettendo l'accesso e la contribuzione da parte di varie istituzioni mediche.

4.1.2 Application server

L'applicazione è stata sviluppata in Java e per la parte logica dell'applicazione si è scelto di usare Apache Tomcat come application e web server. Questo server oltre ad essere una soluzione largamente usata nel mercato, è gratuito ed open-source e fornisce una piattaforma web per l'esecuzione di applicazioni Java. Per l'accesso ai dati da parte dell'applicazione è stato usato Hibernate. Questo ha permesso di mappare gli entity beans sulle entità del database permettendo così una più facile gestione della persistenza dei dati.

4.1.3 Applicazione client medico

Per l'applicazione client del medico si è scelto di sviluppare una web application. Questo permette una completa indipendenza dalla piattaforma in cui è eseguita e una portabilità totale. Per garantire una corretta visualizzazione sui diversi browser è stato usato HTML5 come linguaggio di programmazione. L'accesso alle informazioni è eseguito attraverso i webservice resi disponibili dall'application server.

4.1.4 Applicazione client paziente

Per quando riguarda l'applicazione client mobile del paziente si era inizialmente pensato a una web application per garantire una maggior portabilità tra vari devices. Dopo una prima analisi si è però scelto di sviluppare una applicazione nativa per poter usufruire di funzionalità più complete. L'utilizzo di una applicazione nativa permette di avere, tra l'altro, una maggior flessibilità per espansioni future come per esempio l'integrazione con sensori biomedici. Come prova di concetto è stata sviluppata una applicazione Android considerando la diffusione di questa piattaforma nel mercato mobile e la facilità di sviluppo con strumenti gratuiti che questa piattaforma offre rispetto alla concorrenza. Anche da questa applicazione l'accesso ai dati è eseguito attraverso i webservice resi disponibili dall'application server.

4.2 La base di dati

La prima parte ad essere analizzata è costituita dalle informazioni utilizzate che costituiscono le fondamenta stesse del sistema.

4.2.1 Fonti di informazione

Inizialmente si è svolta una analisi volta a classificare queste informazioni considerando le caratteristiche di origine, di destinazione e di uso in modo da poterle gestire nel modo più appropriato possibile considerando i vari requisiti.

Informazioni mediche

Le informazioni mediche, usate nel nostro sistema, possono essere considerate come le nozioni mediche riguardanti sintomi e malattie. Queste informazioni costituiscono la base di conoscenza condivisa che permette di garantire l'interoperabilità del sistema e che permette di definire in modo univoco gli elementi basilari del sistema. Queste informazioni contengono conoscenza medica che può essere modificata e inserita solo dalla comunità medica.

Informazioni sui pazienti

Questa tipologia di informazione riguarda l'anagrafica dei pazienti e lo storico di tutte le osservazioni mediche relative a ciascun paziente. Queste informazioni sono gestite dalle varie infrastrutture mediche che utilizzano il sistema. La natura di queste informazioni richiede elevati requisiti di privacy e sicurezza e la proprietà di questi dati è della singola struttura medica.

Relazioni tra osservazioni mediche e malattie

I dati riguardanti le relazioni tra osservazioni mediche e malattie rappresentano la conoscenza che il sistema ha acquisito. Queste informazioni riassumono l'equivalente dell'esperienza medica di vari specialisti, permettendo di avere la conoscenza necessaria per un processo di diagnosi. Questa conoscenza è anonima, non dipendendo da pazienti o medici specifici, deriva dall'esperienza e aumenta lungo la vita del sistema. Inoltre questa informazione deve essere accessibile da tutte le infrastrutture mediche, in modo che tutte possano contribuire alla formazione della conoscenza.

4.2.2 Requisiti tecnologici

Dopo aver svolto la precedente analisi sulla natura delle informazioni utilizzate, si sono definiti i requisiti tecnologici per gestire i vari dati. Per quanto riguarda le informazioni mediche, si è deciso di utilizzare ontologie mediche disponibili online[2]. Questo permette di utilizzare le nozioni mediche disponibili, classificate in modo univoco, in modo da garantire l'interoperabilità del

sistema.

Le informazioni sui pazienti sono fornite dalle varie infrastrutture mediche e salvate in un database. Su queste informazioni è necessario svolgere operazioni di tipo CRUD basiche e non è necessario garantire elevate prestazioni di accesso. Questo permette di salvare questi dati in un database locale, in server anche a basse prestazioni, collocato fisicamente presso l'infrastruttura medica che si occupa dei pazienti. Questo permette di proteggere i dati sensibili con le politiche di sicurezza già adottate presso la struttura medica stessa.

Per quanto riguarda le relazioni tra osservazioni mediche e malattie si è deciso di usare sempre un database relazionale. In questo caso le informazioni non hanno nessun requisito di privacy ma il sistema che le gestisce deve garantire buone performance in modo da poter eseguire l'algoritmo di diagnosi e l'aggiornamento della conoscenza in tempo reale. Il database che contiene queste informazioni può essere collocato in un servizio di cloud come ad esempio Amazon EC2[3]. Una infrastruttura di questo tipo, oltre a garantire un livello di performance flessibile e adeguato alle necessità, facilita inoltre l'accesso di varie organizzazioni mediche a queste informazioni.

4.3 L'algoritmo di diagnosi

4.3.1 Tecnica utilizzata

Nella sezione precedente si sono analizzati i vari tipi di informazione e si è visto come sia necessario gestire un tipo di informazione relativa alla conoscenza delle relazioni tra osservazioni mediche e malattie. E' questo tipo di conoscenza medica che permette di effettuare una diagnosi partendo dai sintomi presentati da un paziente. Per rendere possibile questo processo, la tecnica utilizzata per rappresentare questo tipo di conoscenza deve permettere di effettuare inferenza sui dati presenti nel modello. Dopo aver analizzato le tecniche esposte nel Capitolo 2, ci si è resi conto che le ontologie permettono di fare inferenza sopra il modello creato, a partire dal dominio medico, utilizzando logica del primo ordine. Questo tipo di inferenza permette di effettuare deduzioni deterministiche per produrre nuova conoscenza e potrebbe essere utile anche per effettuare diagnosi

medica. Questo metodo di inferenza però presenta un problema: non è adeguato alla diagnosi medica per tre motivi, in accordo con[9]:

- Pigrizia: elencare tutte le possibili combinazioni di antecedenti e conseguenti utilizzando un modello IF-THEN-ELSE per inferire possibili diagnosi è un lavoro molto pesante che non può essere portato a termine in forma completa.
- Ignoranza teorica: la medicina non ha la conoscenza medica completa sul dominio considerato per poter definire tutte le regole necessarie.
- Ignoranza pratica: anche avendo la conoscenza medica completa sul dominio e riuscendo a inserire tutte le regole necessarie a modellare questa conoscenza non è possibile avere la completa conoscenza delle condizioni del paziente perché non tutti i test necessari sono stati svolti o possono essere svolti.

Per risolvere alcuni dei problemi presenti in questo tipo di approccio si è analizzato il possibile utilizzo di Case Base Reasoning come tecnica di inferenza sui dati. Questa tecnica presenta il vantaggio di riuscire ad apprendere dai casi reali, rendendo possibile la creazione di un sistema che apprende dall'esperienza accumulata e che non ha bisogno di inserire manualmente tutte le regole di inferenza necessarie. Ma pur considerando questi benefici sono evidenti altri motivi che non rendono questa tecnica ottimale per la diagnosi medica:

- Evidenza di aneddoti: è abbastanza un solo evento di un certo tipo per dedurre una regola che ha lo stesso peso di regole supportate da maggior esperienza.
- Determinismo: le regole generate sono sempre di carattere deterministico e non considerano l'aspetto probabilistico e le incertezze tipiche di un processo di diagnosi medica.

Riassumendo, si può quindi dire che la logica di primo ordine usata nell'inferenza con reasoner su ontologie e le regole prodotte utilizzando case base reasoning non sono adatte a modellare la conoscenza medica necessaria per effettuare le diagnosi. L'inferenza svolta utilizzando queste tecniche non considera l'aspetto probabilistico e la rilevanza statistica dei vari avvenimenti che costituiscono l'esperienza del sistema.

4.3.2 Rete bayesiana

Per riuscire a modellare l'aspetto probabilistico della conoscenza medica che consente di effettuare diagnosi si è quindi deciso di utilizzare una rete bayesiana. Come descritto precedentemente, una rete bayesiana è un formalismo che permette di descrivere le relazioni di dipendenza probabilistica presenti fra vari eventi. Questa tecnica si adatta bene a modellare le interdipendenze tra osservazioni mediche e la presenza di malattie. E' possibile inoltre creare un algoritmo che permetta di effettuare diagnosi considerando le malattie con maggior probabilità e un algoritmo che mantenga sempre aggiornata la rete, modificando la topologia e i pesi probabilistici dei vari archi in tempo reale man mano che aumenta l'esperienza del sistema.

Topologia

Il primo passo fatto, una volta scelta questa tecnica, è stato quello di definire la topologia della rete. In una rete bayesiana è importante rappresentare tutte le dipendenze che intercorrono tra i vari eventi per avere la maggior accuratezza possibile. Per evitare di avere una rete troppo complessa, d'altro canto, si possono considerare alcune semplificazioni. Secondo[10] in una rete bayesiana per diagnosi medica è possibile adottare una topologia a due livelli senza perdere accuratezza. Il primo livello è rappresentato da osservazioni mediche considerate indipendenti tra di loro per semplificazione. Il secondo livello è rappresentato dalle malattie considerate anch'esse indipendenti tra di loro. Questa struttura semplifica la topologia della rete pur mantenendo una sufficiente espressività necessaria a modellare l'esperienza del sistema.

Probabilità

Una volta definita la struttura della rete si sono definite le formule per calcolare le varie probabilità coinvolte. Per la probabilità a priori di ogni osservazione medica la formula è:

$$\frac{O_i}{O_t}$$

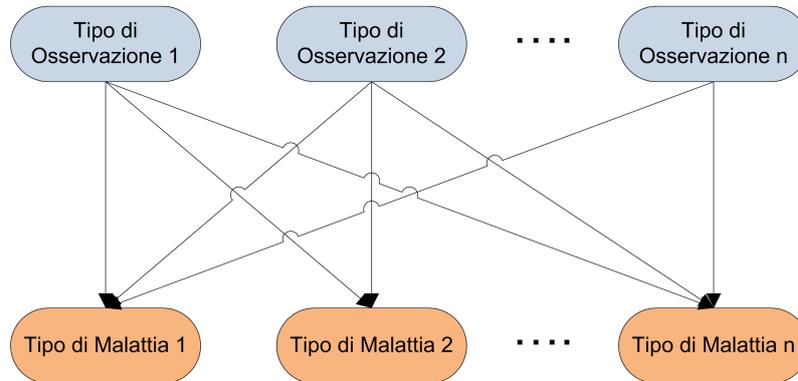


Figura 4.3: Esempio di rete bayesiana a due livelli

dove O_i è la quantità di volte che l'osservazione è stata fatta e O_t il numero di osservazioni mediche totali.

Tosse	Febbre	Probabilità Influenza
F	F	0.01
F	V	0.7
V	F	0
V	V	0.02

Figura 4.4: Esempio di tabella di probabilità della malattia influenza

Per ogni malattia è costruita una tabella di probabilità in cui ogni linea rappresenta le possibili combinazioni di osservazioni mediche in cui la malattia è stata diagnosticata. Per ogni linea è calcolata una probabilità che è definita come:

$$\frac{D_{cd}}{D_{ct}}$$

Dove D_{cd} è la quantità di volte che questa combinazione di osservazioni ha portato alla diagnosi della malattia considerata e D_{ct} è il numero di volte che questa combinazione di osservazioni è stata osservata.

La formula con cui è calcolata la probabilità di una malattia è, quella di probabilità condizionale:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

4.3.3 Algoritmo

L'algoritmo di diagnosi è costituito da due sotto algoritmi.

Algoritmo di aggiornamento della rete

Ogni volta che una diagnosi è registrata in via definitiva nel sistema si procede all'aggiornamento della rete. Per semplicità si assume che ogni diagnosi consideri solo una malattia. In caso fossero diagnosticate più malattie saranno registrate più diagnosi. I passi dell'algoritmo sono:

- Inizializzazione: sono calcolate le probabilità a priori di tutte le malattie. Questo passo è svolto durante l'inizializzazione del sistema una volta sola.
- Passo 1: sono aggiornate le probabilità a priori delle osservazioni mediche presenti nella diagnosi.
- Passo 2: è aggiornata la topologia della rete introducendo, in caso non fossero già presenti, nuovi archi di collegamento tra le osservazioni mediche della diagnosi e la malattia diagnosticata.
- Passo 3: sono ricalcolate le probabilità per tutte le linee delle tabelle di probabilità delle malattie che presentano la stessa combinazione di osservazioni mediche della diagnosi.
- Passo 4: sono ricalcolate le probabilità a priori delle malattie la cui tabella di probabilità è stata modificata.

Algoritmo di diagnosi

Per eseguire una diagnosi l'algoritmo esegue i seguenti passi:

- Passo 1: sono poste uguali a 1 le probabilità a priori di tutte le osservazioni mediche riscontrate nel paziente. Questo perché si ha la certezza di ciò che è già stato riscontrato nel paziente.

- Passo 2: sono aggiornate le probabilità delle malattie che dipendono dalle osservazioni mediche aggiornate.
- Passo 3: sono presentate in ordine di probabilità decrescente le malattie più probabili.

4.3.4 Apprendimento accelerato

Per velocizzare l'apprendimento del sistema si è pensato di rendere possibile l'inserimento di conoscenza in modo manuale. Questa modalità deve essere compatibile e facilmente integrabile con l'altro metodo di acquisizione di conoscenza del sistema attraverso l'analisi delle diagnosi mediche inserite. Quando un medico specialista vuole inserire manualmente nel sistema delle relazioni tra sintomi e malattie viene generata una diagnosi, marcata come non reale, che entrerà a far parte del sistema. È possibile dare un peso a ogni diagnosi non reale inserita in modo da incidere in modo più o meno pesante nel sistema in relazione alla conoscenza preesistente. Per fare questo è stata definita una metrica comune definita come numero di diagnosi. Ogni linea, della tabella di probabilità di una malattia, che mette in relazione la malattia con alcuni sintomi ha un peso definito come il numero di diagnosi che supporta questa relazione. Quando viene inserita una diagnosi non reale, se una linea con la relazione indicata non è presente viene creata e, successivamente, viene aumentato il peso come indicato al momento dell'inserimento. In questo modo è possibile accelerare l'apprendimento del sistema e creare relazioni tra sintomi e malattie presenti nella conoscenza medica senza dover aspettare il verificarsi effettivo della malattia.

4.4 L'interfaccia utente

Il sistema si interfaccia con due tipologie di utente che operano in modo distinto: i medici e i pazienti.

4.4.1 L'interfaccia con il paziente

Il paziente ha la possibilità di inserire, aggiornare e cancellare dati del proprio storico medico e di sintomi che può sentire in un determinato momento attraverso l'uso di un applicativo mobile.

Funzionalità

In modo più specifico le funzionalità dell'applicazione mobile per pazienti sono:

- Login: il paziente si autentica nel sistema per poter accedere ai suoi dati medici e poterli modificare.
- Gestione permessi: il paziente può scegliere quali medici sono autorizzati ad accedere ai suoi dati in modo da tutelare la propria privacy.
- Gestione storico: il paziente può visualizzare lo storico delle diagnosi precedenti e dei sintomi registrati. è possibile inoltre, registrare osservazioni mediche persistenti, che non variano da diagnosi a diagnosi come per esempio, malattie croniche o allergie.
- Gestione sintomi: il paziente può registrare sintomi man mano che avvengono scegliendoli da un elenco organizzato per categorie.

Usabilità

Per quanto riguarda l'usabilità dell'interfaccia utente si è data la priorità maggiore alla facilità di utilizzo. Per questa ragione è stata sviluppata un'interfaccia monocromatica dal design semplice in modo da non distrarre il paziente con dettagli inutili e permettere un accesso più immediato alle funzionalità dell'applicazione. Nel capitolo *Implementazione* è possibile trovare maggiori dettagli sull'implementazione dell'interfaccia con il paziente.

4.4.2 L'interfaccia con il medico

Il medico a partire dai sintomi e delle osservazioni mediche inserite dal paziente, può analizzare e modificare i dati inseriti dal paziente e visualizzare una lista di possibili diagnosi fornite dal sistema. Il medico può inoltre contribuire all'apprendimento del sistema inserendo direttamente conoscenza.

Funzionalità

In modo più specifico il sistema deve garantire le seguenti funzionalità ai medici:

- Richiesta autorizzazione: il medico può richiedere l'autorizzazione a diventare medico curante di un paziente. Una volta accettata la richiesta il medico potrà accedere ai dati del paziente.
- Mostrare storico: il medico può visualizzare lo storico di diagnosi effettuate e i sintomi registrati da ogni paziente in cura.
- Mostrare diagnosi: il sistema permette di visualizzare le possibili diagnosi di un paziente ordinandole per priorità.
- Contribuzione di conoscenza: il sistema permette di apportare conoscenza medica, riguardante le relazioni tra osservazioni mediche e malattie, direttamente da parte di personale medico qualificato. Questo permette di velocizzare il processo di apprendimento del sistema.

Usabilità

Anche sull'interfaccia con il medico si è deciso di dare la priorità maggiore alla facilità di utilizzo. Quindi, anche in questo caso come nell'interfaccia con il paziente, si è deciso di usare un design semplice e monocromatico bianco e nero. Questa scelta, oltre a non distogliere l'attenzione del medico dalle funzionalità del sistema, permette di catturare l'attenzione su eventuali informazioni importanti utilizzando un altro colore. Nel capitolo *Implementazione* è possibile trovare maggiori dettagli sull'implementazione dell'interfaccia con il medico.

Capitolo 5

Implementazione

Questo capitolo ha l'obiettivo di mostrare i dettagli implementativi delle varie parti che compongono il sistema così come le motivazioni delle varie scelte implementative.

5.1 Sistema di apprendimento automatico

Il primo passo nell'implementazione dell'algoritmo è stato scegliere quali ontologie utilizzare come base di conoscenza di sintomi e malattie. A questo scopo sono state scelte le ontologie `doid.owl` per le malattie e `symp.owl` per i sintomi[2]. Una volta ottenute le ontologie, si è provveduto a trasferire il contenuto delle stesse nel database, mantenendo le informazioni sulla gerarchia di sussunzione delle varie classi. Per fare questo si è sviluppata una libreria che facilitasse l'accesso alle informazioni contenute nell'ontologia basata su Jena[22]. Questa libreria poi può essere usata anche per aggiornare lo stato del database qualora le ontologie fossero modificate o si scegliesse di utilizzare altre ontologie. Una volta poste queste informazioni nel database si è provveduto a sviluppare l'algoritmo di diagnosi vero e proprio, anche questo scritto in Java. Come descritto nel capitolo sulla struttura, l'algoritmo di diagnosi è diviso in due parti. La prima parte si occupa di mantenere aggiornata la rete bayesiana in tempo reale ad ogni diagnosi inserita nel sistema. La seconda parte si occupa di usare le informazioni presenti nella rete bayesiana per fare stime probabilistiche di quali malattie siano più (VEROSIMILI) probabili, dati determinati sintomi.

Per la prima parte dell'algoritmo, quando il sistema è inizializzato vengono calcolate le probabilità a priori di ogni sintomo e di ogni malattia (pagina ??). Una volta che il sistema è inizializzato, ogni volta che è chiamato il webservice che registra una diagnosi, viene attivata una procedura che aggiorna solo la parte di rete interessata, considerando i sintomi relazionati alla diagnosi e la malattia diagnosticata. Quest'aggiornamento parziale permette di eseguire il codice in modo rapido anche su reti bayesiane di grandi dimensioni. Con questo codice quindi, la rete bayesiana rimane sempre aggiornata, considerando le probabilità a priori di sintomi e malattie. Usando come base questa rete, la seconda parte di algoritmo riceve in ingresso l'elenco di sintomi riscontrati in un paziente. Una volta ricevuti i sintomi, l'algoritmo crea una copia temporanea locale della porzione di rete bayesiana che comprende le malattie relazionate ai sintomi riscontrati. Le probabilità dei sintomi riscontrati vengono poste uguali a uno in questa porzione di rete e si ricalcolano le probabilità delle malattie. L'algoritmo provvede poi a ordinare le malattie in ordine di probabilità decrescente e quindi a dare come risultato le malattie più probabili. Le probabilità delle malattie ottenute dall'algoritmo su questa porzione di rete sono confrontate, per completezza, con le probabilità a priori di tutte le malattie presenti nella rete bayesiana in modo che, venga presentata nei risultati anche un'eventuale malattia non direttamente connessa ai sintomi ma con maggiori probabilità rispetto ad una relazionata. L'implementazione di questo algoritmo è stata fatta per buona parte usando query HQL[23] lasciando così all'ottimizzatore di query il compito di ottimizzare le performance sull'accesso alla grande quantità di dati presente nella parte di database relativa alla rete bayesiana.

5.2 Interfaccia con il medico

L'interfaccia del medico, implementata in HTML5 è stata sviluppata mantenendo come obiettivo principale la funzionalità e la facilità d'uso.

Nella pagina di gestione pazienti (figura 5.1) sono mostrati tutti i pazienti attualmente in cura. È possibile anche richiedere l'autorizzazione al trattamento di altri pazienti e vedere quali richieste sono ancora pendenti. Come funzionalità futura sarà possibile implementare facilmente un'allerta visuale riguardante i

UrHealth Personal Data Logout

Patients Diagnostics Help Contact

My Patients

Name	CPF	Birthday	Age	
John Doe	331.906.446-10	16/09/1943	69	Remove
Daniel Smith	038.999.625-40	14/12/1989	23	Remove

Add new patient

Clark Kent

Request permissions

Pending Requests

Name	CPF	Birthday	Age	
Luiz Fabiano	123.456.789-10	27/10/1967	45	Remove
Toad Kinopio	123.456.789-01	02/03/2008	4	Remove

Copyright © 2012 urhealth.mobi - UrHealth is not responsible for diagnostics nor treatments - contato@urhealth.mobi

Figura 5.1: Pagina per la gestione dei pazienti in cura

pazienti in cura sui quali il sistema ha diagnosticato una possibile malattia con probabilità alta.

Accedendo alla scheda dettagliata di ogni paziente (figura 5.2) è possibile gestire le osservazioni mediche relative allo stesso. Sono visualizzati i sintomi inseriti dal paziente attraverso l'applicazione mobile e altri sintomi inseriti dal medico. E' inoltre possibile vedere i sintomi relativi a diagnosi passate che potrebbero in qualche modo essere relazionati alla diagnosi corrente.

Una volta che tutti i sintomi del paziente sono stati inseriti nel sistema è possibile procedere alla pagina successiva (figura 5.3) nella quale sono proposte le possibili diagnosi suggerite dal sistema. Da questa pagina è possibile confermare una delle diagnosi suggerite o scegliere manualmente la malattia che il medico ritiene corretta.

Oltre a poter gestire i pazienti, i loro sintomi e le loro diagnosi, è permesso inserire nuova conoscenza nel sistema (Pagina 31). A tale scopo è possibile usare la pagina per l'inserimento di diagnosi custom (figura 5.4). In questa pagina si possono inserire diagnosi non relative a pazienti reali, con l'obiettivo di creare nuove relazioni tra sintomi e malattie all'interno del sistema, in modo da accelerare il suo apprendimento. Queste diagnosi custom possono derivare ad esempio da articoli scientifici medici o da altre fonti di informazione affidabili. Questa pagina non è accessibile a qualsiasi medico, ma solo agli specialisti autorizzati ad intervenire artificialmente nell'apprendimento del sistema. La procedura prevede in primo luogo l'inserimento della malattia a cui si vogliono relazionare i sintomi. Una volta inserita la malattia (figura 5.5) è possibile scegliere fino a dieci sintomi relazionati e il peso con cui questa diagnosi artificiale sarà pesata rispetto alle altre diagnosi.


UrHealth

Personal Data
Logout

Patients
Diagnostics
Help
Contact

John Doe

Personal Data
Symptoms
Diagnostics

Active Symptoms

Symptom	Period	
Cyclic fever	19/01/2013 to 22/01/2013	Remove
Progressive weakness	10/01/2013 to 13/01/2013	Remove

Add New Symptom

or

Previous Symptoms

Symptom	Period	
Breathing problems	09/12/2012 to 10/12/2012	Remove
Breathing problems	09/12/2012 to 10/12/2012	Remove
Loss of appetite	09/12/2012 to 10/12/2012	Remove
Tiredness	09/12/2012 to 10/12/2012	Remove
Continuous fever	09/12/2012 to 10/12/2012	Remove
Wheezing	09/12/2012 to 10/12/2012	Remove
Chronic cough	09/12/2012 to 10/12/2012	Remove
Breathing problems	09/12/2012 to 10/12/2012	Remove
Chronic cough	09/12/2012 to 10/12/2012	Remove
Chronic cough	09/12/2012 to 10/12/2012	Remove
Breathing problems	09/12/2012 to 10/12/2012	Remove

Figura 5.2: Pagina per la gestione dei sintomi del paziente


UrHealth

Personal Data
Logout

Patients	Diagnostics	Help	Contact
John Doe			
Personal Data	Symptoms	Diagnostics	

Active Symptoms

Symptom	Period
Cyclic fever	19/01/2013 to 22/01/2013
Progressive weakness	10/01/2013 to 13/01/2013

Suggested Diagnostics

Disease	Probability	
Bronchitis	<input type="text" value="7%"/>	Confirm
Intrinsic asthma	<input type="text" value="4%"/>	Confirm
Allergic asthma	<input type="text" value="1%"/>	Confirm
Cough variant asthma	<input type="text" value="1%"/>	Confirm

Add other diagnostic

or

Search:

Previous Diagnostics

Disease	When	Symptoms
Intrinsic asthma	10/12/2012	Chronic cough Wheezing Breathing problems
Bronchitis	10/12/2012	Chronic cough Breathing problems Loss of appetite Tiredness Continuous fever

Figura 5.3: Pagina per la visualizzazione dei suggerimenti di diagnosi e per l'inserimento della diagnosi

The screenshot shows the UrHealth web interface. At the top left is a black cross icon followed by the text "UrHealth". To the right are two buttons: "Personal Data" and "Logout". Below this is a navigation bar with four tabs: "Patients", "Diagnostics", "Help", and "Contact". The "Diagnostics" tab is active. The main content area is titled "Custom Diagnostics" and contains a table with the following data:

Disease	When	Symptoms	
Allergic asthma	10/12/2012	Wheezing Chronic cough Breathing problems	Remove

Below the table is the section "Add New Custom Diagnostic". It includes a "Choose a Disease" label and a dropdown menu with "Choose one...". Below this is the word "or" and a "Search:" label followed by a text input field. The next section is "Batch procedure", which contains the text: "Use this link to run the batch procedure to update the probabilities. It will take several minutes! Please make sure you know what you are doing!" and a blue "Start!" link. Below that is the "Batch Status" section, which shows the text: "Last information 01/02/2013 01:36:01: Batch executed successfully". At the bottom of the page is a copyright notice: "Copyright © 2012 urhealth.mobi - UrHealth is not responsible for diagnostics nor treatments - contato@urhealth.mobi".

Figura 5.4: Pagina per l'inserimento di diagnosi custom



UrHealth

Patients
Diagnosics
Help
Contact

Custom Diagnostics

Disease	When	Symptoms	
Allergic asthma	10/12/2012	Wheezing Chronic cough Breathing problems	Remove

Add New Custom Diagnostic

Choose a Symptom (up to 9)

or

Search:

Disease: Swine influenza

Symptom 1: Cyclic fever

Symptom 2: Chronic cough

Weight:

Batch procedure

Use this link to run the batch procedure to update the probabilities.
It will take several minutes! Please make sure you know what you are doing!

[Start!](#)

Batch Status

Last information 01/02/2013 01:36:01: Batch executed successfully

Copyright © 2012 urhealth.mobi - UrHealth is not responsible for diagnostics nor treatments - contato@urhealth.mobi

Figura 5.5: Inserimento di diagnosi custom

5.3 Interfaccia paziente

L'interfaccia paziente, implementata come una applicazione Android, è stata strutturata in modo semplice, per garantire le funzionalità base per poter testare il sistema completo. Per semplificare e velocizzare l'implementazione sono state tralasciate alcune funzionalità che potranno essere inserite facilmente in futuro, come per esempio il salvataggio offline dei sintomi o la visualizzazione dello storico delle diagnosi effettuate sul paziente.

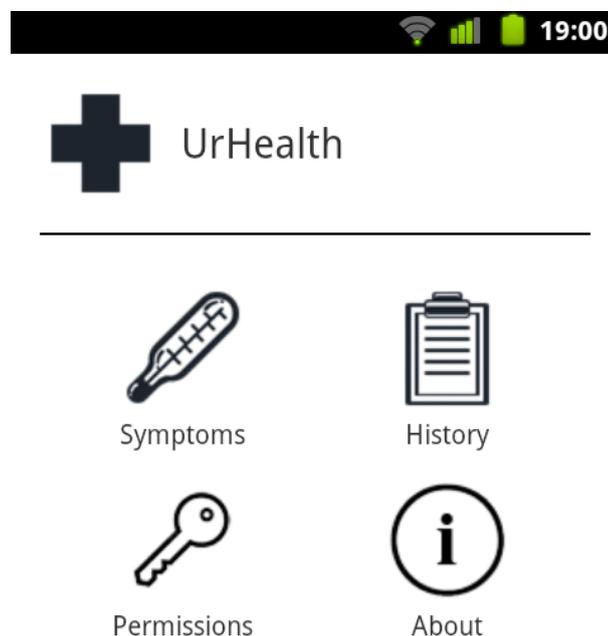


Figura 5.6: Menu dell'applicazione mobile

Dopo aver fatto il login nell'applicazione è possibile vedere la schermata di menu (Figure 5.6). Da questa view è possibile accedere alle varie funzionalità rese disponibili al paziente.

La funzionalità principale è quella dell'inserimento dei sintomi (Figura 5.7). Da questa view è possibile cercare il sintomo che si desidera inserire. La selezione può essere fatta navigando l'albero di classificazione dei sintomi (mappato sulla struttura dell'ontologia) o attraverso l'apposito campo di ricerca. Vale la pena ricordare che nel sistema è presente un campo per marcare i sintomi visibili al

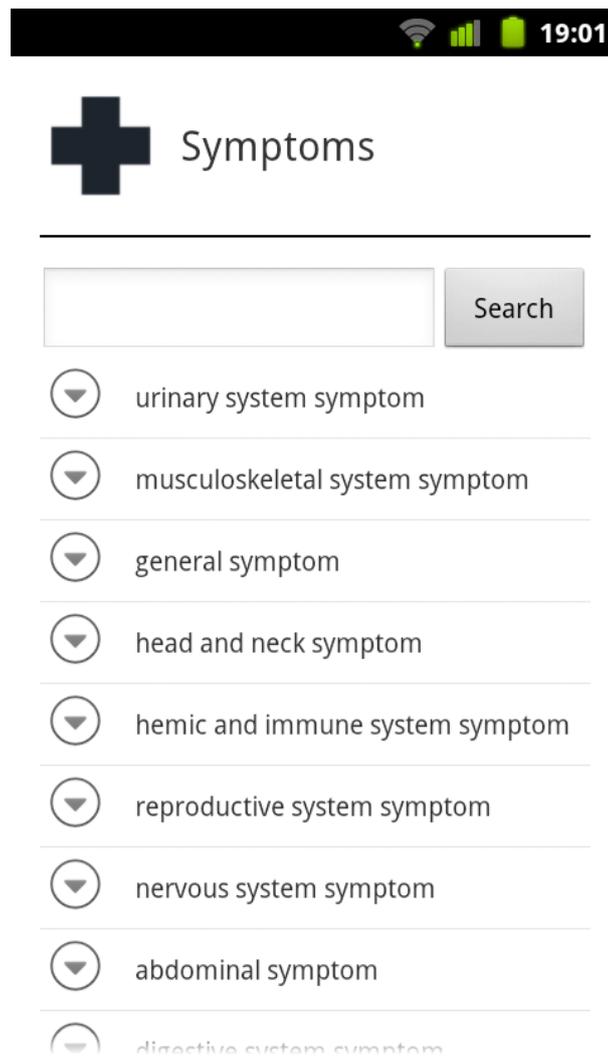


Figura 5.7: Selezione di sintomi

paziente e quelli non visibili. In questo modo dall'applicazione mobile sono visualizzabili e selezionabili solo i sintomi visibili al paziente. Questa proprietà di visibilità è definita da personale medico autorizzato.

Una volta selezionato il sintomo che si desidera inserire si accede alla view dalla quale è possibile selezionare la durata del sintomo (figura 5.8). Per semplicità in questo prototipo di applicazione è possibile scegliere solamente il giorno di inizio e fine ma il sistema sottostante accetta dati più dettagliati ed è quindi possibile in futuro, modificando solo questa applicazione, registrare con maggior dettaglio la durata dei sintomi registrati.

Oltre all'inserimento di sintomi, l'applicazione permette di visualizzare lo storico dei sintomi precedentemente registrati (figura 5.9). Questa schermata ad esempio è utile per il paziente per verificare se un determinato sintomo è già stato inserito o meno.

Infine è possibile, attraverso una apposita view (figura 5.10) verificare quali medici hanno accesso ai dati del paziente e autorizzare le richieste di autorizzazione presentate da altri medici.

19:23

+ Symptoms

thirst

Thirst is a general symptom characterized by a sensation of dryness in the mouth and throat associated with a desire for liquids.

Start

+	+	+
Jan	01	2013
-	-	-

End

+	+	+
Jan	02	2013
-	-	-

Submit

Figura 5.8: Inserimento di un sintomo



Figura 5.9: Visualizzazione dello storico di sintomi inseriti

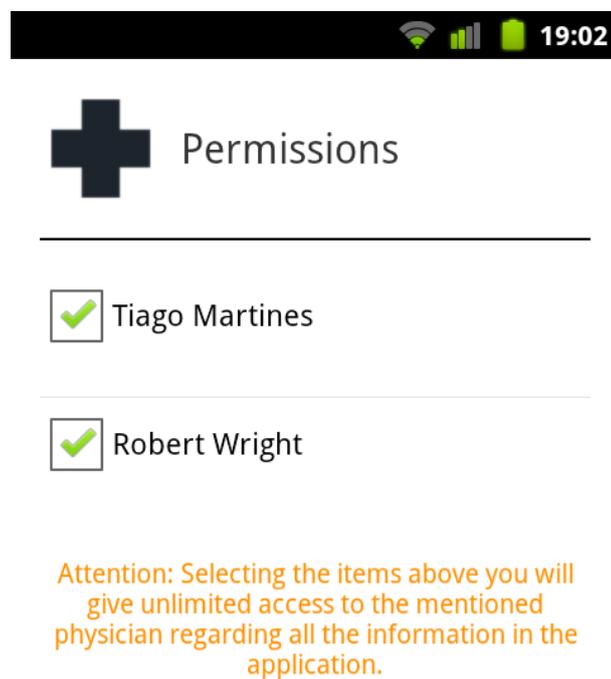


Figura 5.10: Concessione di permessi ai medici

Capitolo 6

Risultati

In questo capitolo saranno mostrati i risultati ottenuti dal sistema sia dal punto di vista delle performance dell'algoritmo che sull'usabilità delle interfacce.

6.1 Alcune prove pratiche

Occorre precisare, prima di descrivere le prove effettuate, che testare il sistema in modo completo con una mole di dati rilevante richiederebbe un tempo decisamente grande, non compatibile con la tipologia di lavoro sviluppata. Per questo motivo non si sono effettuati test che prevederebbero il confronto dei suggerimenti del sistema con diagnosi mediche reali come abitualmente fatto su progetti ben più grandi, in quanto avrebbero richiesto l'inserimento di una grandissima quantità di dati comparabile a quanto il sistema apprenderebbe in alcuni anni di utilizzo. Per verificare il funzionamento del sistema si è provveduto a inserire una serie di diagnosi di test, a monitorare la formazione della rete bayesiana corrispondente e a verificare che i dati relativi ai suggerimenti di diagnosi fossero compatibili con la rete bayesiana presente nel sistema al momento della diagnosi. Per verificare in modo immediato la topologia della rete bayesiana presente nel sistema è stato sviluppato un apposito strumento che permette di visualizzarla in una interfaccia web.

Utilizzando l'interfaccia web del medico sono stati inserite varie diagnosi, utilizzando lo strumento descritto precedentemente si è verificato che la rete evolvesse correttamente e che le diagnosi fornite fossero coerenti con quanto previsto dal-

l'algoritmo. I risultati ottenuti sono stati conformi a quanto aspettato validando quindi l'implementazione dell'algoritmo. Tra i vari test è stato effettuato il seguente.

Nel sistema viene inserita la diagnosi della tabella 6.1.

	Sintomi	Malattia	Peso
Diagnosi 1	Wheezing Chronic cough Breathing problem	Intrinsic Asthma	1

Tabella 6.1: Prima diagnosi di test

La rete generata in seguito a questa diagnosi, partendo da una rete completamente vuota, è rappresentata nella figura 6.1

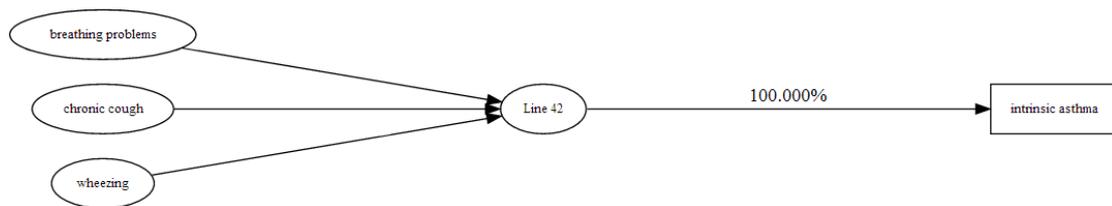


Figura 6.1: Rete bayesiana dopo l'inserzione della prima diagnosi di test

Con questa rete viene richiesta una diagnosi per i sintomi della tabella 6.2 e il suggerimento di diagnosi è mostrato nella figura 6.2. È possibile notare come la probabilità del suggerimento sia del 33% perché manca il sintomo Wheezing che ha una probabilità a priori del 33% visto che finora sono stati registrati solo i 3 sintomi della prima diagnosi.

Disease	Probability	
Intrinsic asthma	<input type="text" value="33%"/>	Confirm

Figura 6.2: Suggerimento di diagnosi del sistema

Nonostante il suggerimento del sistema, viene inserita come malattia *Bronchitis*, e la rete che si ottiene è quindi quella di figura 6.3.

6.1. Alcune prove pratiche

	Sintomi	Malattia	Peso
Diagnosi 2	Chronic cough Breathing problems Loss of appetite Tiredness Continuos fever	Bronchite	1

Tabella 6.2: Seconda diagnosi di test

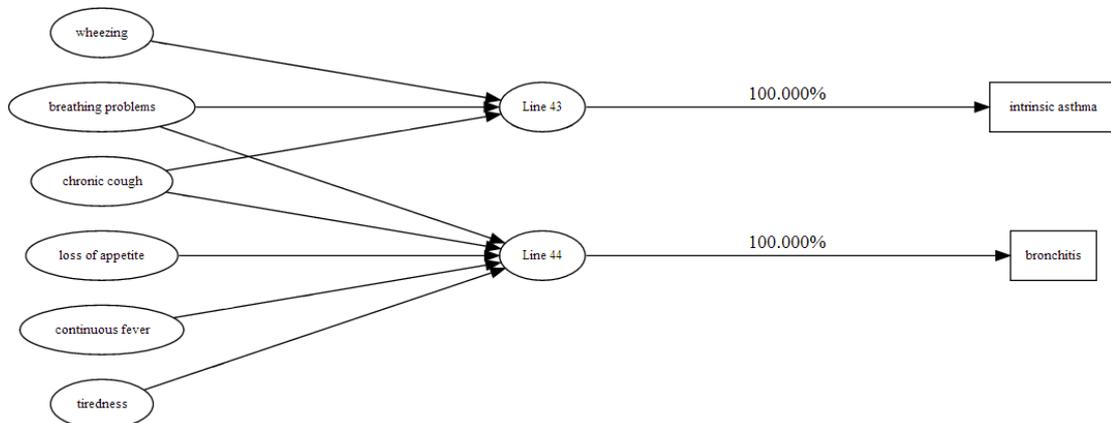


Figura 6.3: Rete bayesiana dopo l'inserzione della seconda diagnosi di test

Continuando con il test viene richiesta la diagnosi per i sintomi presenti nella tabella 6.3 e il suggerimento del sistema è mostrato nella figura 6.4.

Disease	Probability	
Intrinsic asthma	 100%	Confirm
Bronchitis	 0%	Confirm

Figura 6.4: Suggerimento di diagnosi del sistema

	Sintomi	Malattia	Peso
Diagnosi 3	Wheezing Chronic cough Breathing problem	Cough variant asthma	1

Tabella 6.3: Terza diagnosi di test

è possibile verificare che in questo caso i sintomi combaciano perfettamente con quelli della prima diagnosi inserita e quindi il sistema suggerisce la stessa malattia diagnosticata precedentemente con probabilità del 100% e *Bronchitis* con probabilità quasi nulla dato che sono presenti solo 2 sintomi dei 5 riscontrati nella diagnosi precedente. In questo caso viene diagnosticata una terza malattia: *Cough variant asthma*, e la rete ottenuta è rappresentata in figura 6.5. È possibile verificare come alla stessa combinazione di sintomi siano adesso associate due malattie diverse in modo equiprobabile visto che entrambi casi sono supportati da una sola diagnosi.

Il test continua inserendo una diagnosi fittizia. Questa tecnica, descritta nel Capitolo 5, permette di inserire nuova conoscenza manualmente attribuendo un peso corrispondente al numero di diagnosi che supportano il fatto inserito. In questo caso viene inserita la diagnosi della tabella 6.4 e la rete ottenuta è quella di figura 6.6.

È possibile verificare come l'inserimento di questa diagnosi si sia ripercossa sul sistema ridistribuendo le probabilità della prima combinazione di sintomi ed è possibile notare come la *Allergic asthma* supportato dall'ultima diagnosi con peso 2 abbia il doppio di probabilità delle altre due malattie.

6.1. Alcune prove pratiche

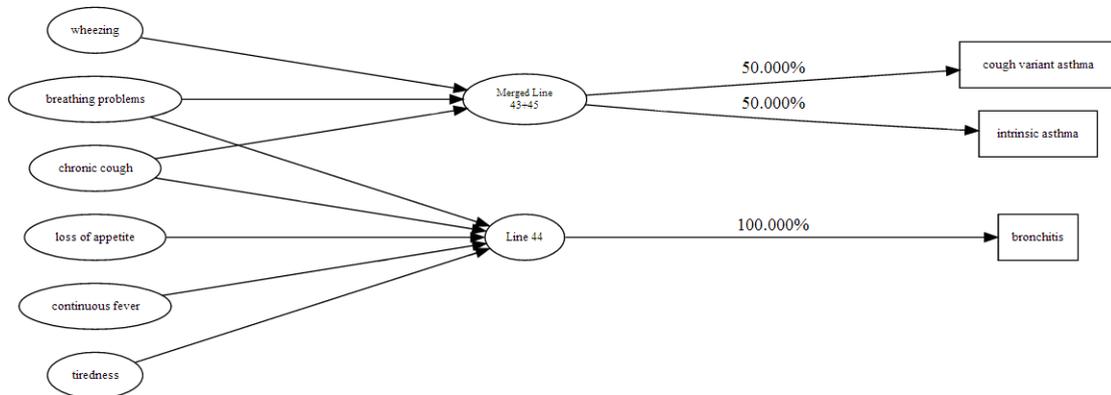


Figura 6.5: Rete bayesiana dopo l'inserizione della terza diagnosi di test

	Sintomi	Malattia	Peso
Diagnosi 4	Wheezing Chronic cough Breathing problem	Allergic asthma	2

Tabella 6.4: Quarta diagnosi di test

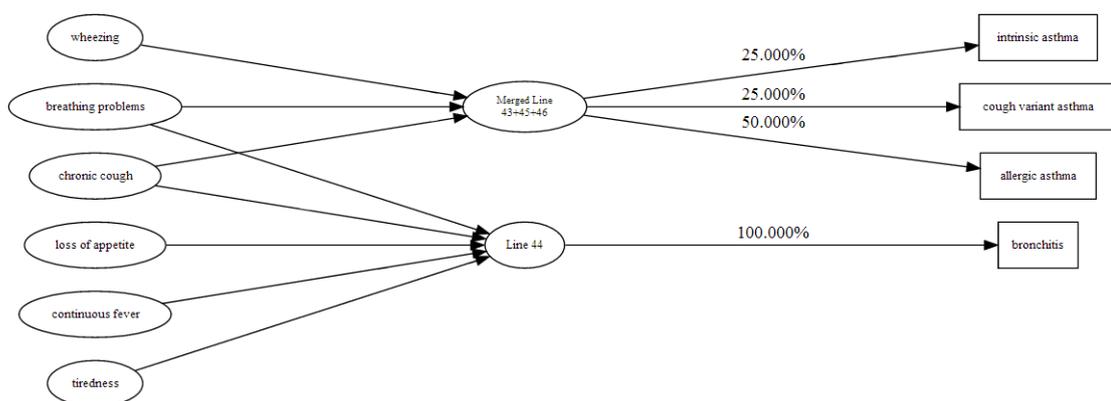


Figura 6.6: Rete bayesiana dopo l'inserizione della quarta diagnosi di test

Il test continua chiedendo una diagnosi per i sintomi della tabella 6.5. I risultati ottenuti sono visibili in figura 6.7.

Disease	Probability	
Allergic asthma	 11%	Confirm
Intrinsic asthma	 5%	Confirm
Cough variant asthma	 5%	Confirm
Bronchitis	 0%	Confirm

Figura 6.7: Suggerimento di diagnosi del sistema

	Sintomi	Malattia	Peso
Diagnosi 5	Chronic cough Breathing problem	Intrinsic asthma	1

Tabella 6.5: Quinta diagnosi di test

è possibile vedere come la redistribuzione delle probabilità dell'inserimento precedente influenzi direttamente questa diagnosi in cui *Allergic asthma* presenta il doppio di probabilità delle altre due malattie come ci si aspetta dalla rete bayesiana analizzata precedentemente. In questo caso viene confermato *Intrinsic asthma* e la rete ottenuta dopo questa diagnosi è rappresentata in figura 6.8.

Seguendo l'evolversi del sistema lungo test di questo tipo si è verificato il corretto funzionamento delle varie parti del sistema. È possibile verificare come il sistema apprenda da ogni diagnosi inserita e come utilizzi questa conoscenza nelle successive diagnosi elaborate.

Un altro test è stato effettuato per verificare i tempi di esecuzione dell'algoritmo. Ad ogni diagnosi inserita nel sistema la rete si deve aggiornare e il sistema deve riuscire a fornire suggerimenti per la diagnosi in tempi brevi. Ai fini di test il sistema è stato caricato su una utenza gratuita del servizio EC2 di Amazon[3]. I tempi di esecuzione sono stati assolutamente rapidi, l'aggiornamento della rete è avvenuto quasi istantaneamente e anche i suggerimenti di diagnosi sono forniti in frazioni di secondo. Questo era aspettato anche considerando la quantità ridotta

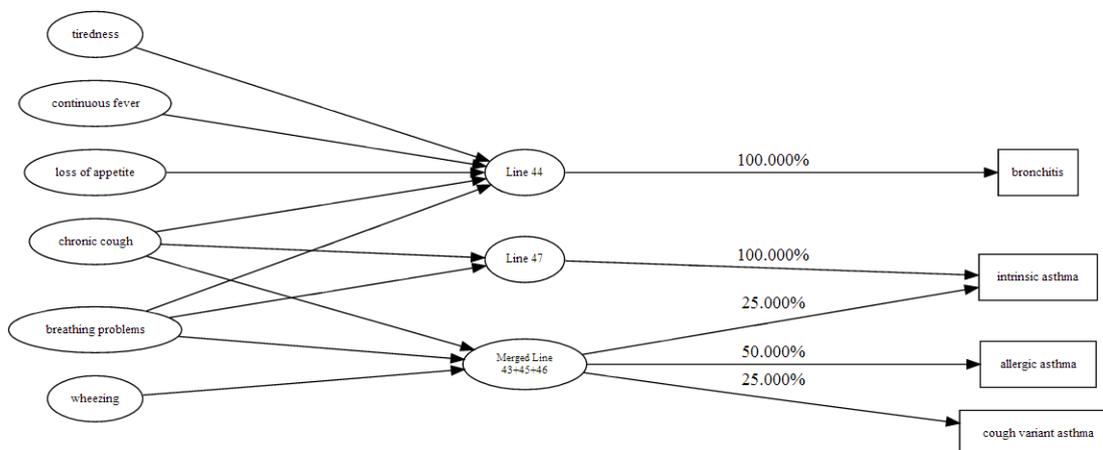


Figura 6.8: Rete bayesiana dopo l’inserimento della quarta diagnosi di test

di dati presenti nel sistema. È importante però ricordare che l’algoritmo è ottimizzato in modo tale da considerare solo la porzione di rete che interessa i sintomi analizzati e che quindi non ci si aspetta tempi di esecuzione lunghi anche quando il sistema avrà a disposizione una quantità di dati maggiore da elaborare.

6.2 Test di usabilità con medici

Per quanto riguarda l’interfaccia web disponibile per i medici e l’applicazione mobile sono stati effettuati test con alcuni medici. I medici hanno apprezzato la semplicità e l’immediatezza dell’interfaccia che hanno ritenuto funzionale e adeguata allo scopo. Sono anche stati fatti presente alcuni suggerimenti riguardanti l’interfaccia dell’applicazione Android per i pazienti. In particolare è stato suggerito di aggiungere un’interfaccia per l’inserimento dei sintomi di tipo grafico che permetta di scegliere i sintomi indicando, su un manichino, l’area del corpo umano interessata. Un’altra osservazione ha riguardato la gestione dello storico dei sintomi del paziente sempre nell’applicazione Android che deve essere migliorata anch’essa per fornire una maggior funzionalità.

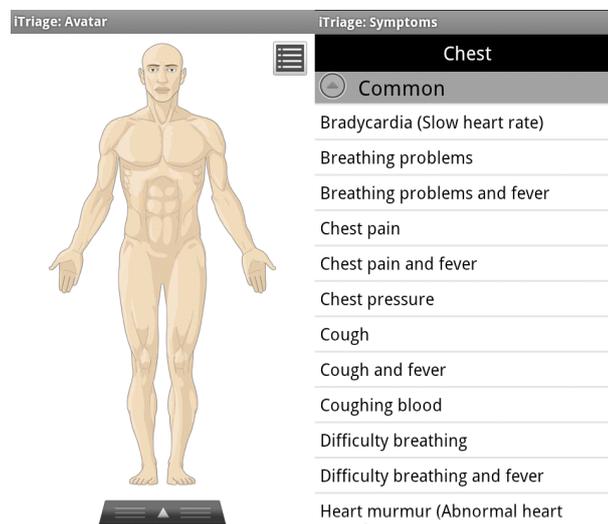


Figura 6.9: Esempio di interfaccia grafica per l'inserimento di sintomi

Capitolo 7

Conclusioni e Sviluppi futuri

In questa parte della tesi verranno trattate alcune conclusioni sul lavoro svolto e ci si concentrerà sui possibili miglioramenti e sviluppi futuri che possono essere apportati al sistema.

7.1 Analisi dei risultati

I risultati ottenuti sono stati soddisfacenti. Come descritto nel capitolo precedente, il sistema è stato testato per le funzionalità dei vari componenti con esito positivo. Chiaramente non è stato possibile testare il sistema comparandolo a diagnosi mediche reali, sia per la difficoltà di accesso a queste informazioni, sia per la grande mole di lavoro richiesta per arrivare a un sistema con un livello di esperienza sufficiente.

Il sistema è stato sviluppato utilizzando tecniche e tecnologie adeguate. Le reti bayesiane hanno permesso di modellare in modo efficace le relazioni tra osservazioni mediche e malattie costituendo quindi la parte di esperienza del sistema. Le ontologie hanno permesso di costruire una base di conoscenza condivisa, con cui è possibile avere in comune fra varie strutture mediche le informazioni raccolte e costruire uno standard di comunicazione che permetta loro di contribuire apportando conoscenza. Questo permetterà, una volta adottato il sistema, di accelerare il suo apprendimento con il contributo di molti medici. Anche l'utilizzo di interfaccia mobile per i pazienti consente di beneficiare di una presenza più pervasiva nella vita degli stessi, dando l'opportunità di raccogliere maggiori informazioni e

in modo più dettagliato e completo.

Questo sistema non può ancora essere considerato completo. Il suo obiettivo è quello di costruire un punto di partenza per un sistema sempre più completo che possa beneficiare dei vantaggi descritti precedentemente e che possa essere adottato da strutture mediche. Per raggiungere questo obiettivo è necessario aggiungere alcune funzionalità e migliorare alcuni aspetti come descritto nella prossima sezione.

7.2 Sviluppi futuri

Questa sezione si occupa di descrivere quali sono i miglioramenti che possono essere apportati al sistema ed i suoi sviluppi futuri.

7.2.1 Miglioramenti all'algoritmo

L'algoritmo può essere migliorato sotto due aspetti:

- Negazione certa di osservazioni mediche
- Considerazione di altri aspetti determinanti

Il sistema attualmente permette di inserire osservazioni mediche che sono state rilevate, considerando tutto quello che non è stato inserito come possibile, utilizzando la probabilità a priori di ciascuna osservazione medica considerabile. Un miglioramento, da questo punto di vista, è lasciare la possibilità al medico di escludere definitivamente le osservazioni mediche riducendo a zero la loro probabilità ai fini dell'elaborazione della diagnosi, migliorando così l'accuratezza dell'algoritmo. Il secondo punto riguarda gli aspetti considerati come determinanti ai fini dell'elaborazione della diagnosi. Attualmente il sistema considera come possibili elementi relazionati alle malattie le osservazioni mediche, principalmente sintomi, presenti nell'ontologia *symp.owl*[2]. Per come è strutturato il sistema, è possibile definire altri elementi che possono influenzare la diagnosi, come per esempio, le caratteristiche temporali dei sintomi e la localizzazione geografica del paziente. Ad esempio sarebbe possibile sfruttare le informazioni fornite dall'applicazione mobile del paziente per raccogliere le informazioni temporali e geografiche e utilizzare dal punto

di vista dell'algoritmo una sottorete bayesiana localizzata geograficamente, creata a partire dalle diagnosi effettuate nell'area geografica di interesse.

7.2.2 Funzionalità aggiuntive

Alcune funzionalità che possono migliorare l'utilità del sistema sono le seguenti:

- webservice pubblico
- accreditamento del personale medico qualificato per l'inserimento di conoscenza

Il sistema, così come è strutturato oggi, permette di fare consultazioni e avere suggerimenti di diagnosi solo utilizzando l'interfaccia web dedicata al personale medico. Il sistema avrebbe una fruibilità maggiore se rendesse disponibile un webservice che, analizzando i sintomi forniti, fornisse un suggerimento di diagnosi. Questo servizio è facilmente implementabile, una volta stabilita come base di conoscenza condivisa l'ontologia utilizzata per definire i sintomi. L'aggiunta di questa funzionalità permetterebbe l'accesso al servizio di diagnosi da parte di terzi che non utilizzano il nostro sistema, o che pretendono mantenere il loro preesistente, ma che vogliono comunque usufruire della conoscenza che abbiamo a disposizione. Questo servizio potrebbe inoltre essere incorporato in mashup e essere utilizzato per l'implementazione di webapplication con fini diversi.

Attualmente il sistema permette l'apporto di conoscenza medica manualmente attraverso l'inserimento di diagnosi finte a peso variabile. Una funzionalità che migliorerebbe questo aspetto è l'introduzione di un sistema di accreditamento automatico dei medici. Nei dati riguardanti il medico è possibile definire se il medico è autorizzato o meno all'inserimento manuale di conoscenza. Sarebbe utile aggiungere la possibilità di definire l'area di specializzazione del medico, in modo da attribuire un peso compatibile con le conoscenze dello stesso sulle modifiche introdotte, o da inibire in modo selettivo le aree in cui il medico può apportare conoscenza.

7.2.3 Miglioramento dell'interfaccia mobile

Come suggerito dal personale medico che ha testato l'applicazione mobile (Capitolo 6), è necessario migliorare alcune funzionalità dell'interfaccia con il paziente. In particolare è necessario modificare la gestione dello storico dei sintomi, migliorandone la fruibilità, permettendo non solo la visualizzazione delle osservazioni mediche inserite ma anche l'accesso alle malattie diagnosticate e ai referti medici passati. È necessario inoltre migliorare l'inserimento dei sintomi da parte del paziente, permettendo, oltre alla ricerca testuale degli stessi, la selezione attraverso l'utilizzo di un avatar (Figura 6.9) come in applicazioni già presenti sul mercato che richiedono questa funzionalità.

7.2.4 Inserimento della conoscenza medica

Il sistema, infine, può migliorare la sua efficacia solo disponendo della conoscenza medica necessaria. È quindi necessario spingere sull'adozione del sistema da parte di più istituzioni mediche possibili, in modo da garantire un apporto consistente e rapido di esperienza al sistema e rendere quindi più appetibile il suo utilizzo una volta raggiunto un livello di accuratezza sufficiente.

Appendice A

Tecnologie adottate

In questa appendice verranno descritte più dettagliatamente alcune tecnologie utilizzate nel nostro sistema.

A.1 Rete bayesiana

Una rete bayesiana è un modello grafico stocastico che rappresenta diverse variabili probabilistiche e le loro dipendenze condizionali attraverso l'uso di un grafo aciclico diretto. Nel grafo i nodi rappresentano le variabili casuali che possono essere quantità osservabili o ipotesi. Gli archi, invece, rappresentano le relazioni di dipendenza probabilistica; i nodi non connessi sono condizionalmente indipendenti fra di loro. Ad ogni nodo è associata una funzione di probabilità che prende in ingresso i possibili valori delle variabili padre (nodi che hanno archi che puntano al nodo considerato) e restituisce la probabilità della variabile rappresentata dal nodo. Esistono vari algoritmi che effettuano inferenza e apprendimento a partire da reti bayesiane.

A.2 Le ontologie

In informatica, un'ontologia è una rappresentazione formale, condivisa ed esplicita di una concettualizzazione di un dominio di interesse. Più nel dettaglio, si tratta di una teoria assiomatica del primo ordine esprimibile in una logica descrittiva. Il termine ontologia (formale) è entrato in uso nel campo dell'intelligenza

artificiale e della rappresentazione della conoscenza, per descrivere il modo in cui diversi schemi vengono combinati in una struttura dati contenente tutte le entità rilevanti e le loro relazioni in un dominio. I programmi informatici possono poi usare l'ontologia per una varietà di scopi, tra cui il ragionamento induttivo, la classificazione, e svariate tecniche per la risoluzione di problemi. In questo progetto le ontologie sono utilizzate al fine di classificare le osservazioni mediche e le malattie e di costituire quindi una base condivisa con cui descrivere la conoscenza. Questo tipo di ontologie è chiamata fondazionale e costituisce come un glossario di base del dominio di interesse organizzato in modo gerarchico attraverso relazioni di sussunzione.

A.3 HTML5

L'HTML5 è un linguaggio di markup per la strutturazione di pagine web. Rispetto all'HTML4 questa nuova versione ha migliorato il disaccoppiamento tra struttura di una pagina web, definita dal markup, e lo stile definito dalle direttive di stile. Questo nuovo standard permette una migliore portabilità delle pagine web che possono così essere renderizzate in modo identico su browser e piattaforme differenti.

A.4 Android

Android è un sistema operativo per sistemi mobile caratterizzato da una struttura open source e dall'essere basato su kernel Linux. Queste caratteristiche facilitano lo sviluppatore, che può così sviluppare nuove applicazioni usando strumenti gratuiti e piattaforme aperte. Android inoltre è il sistema operativo mobile con la quota di mercato maggiore[27] e la gamma di dispositivi più ampia. Sviluppare per Android permette quindi di raggiungere il più grande pubblico possibile.

Bibliografia

- [1] Raghupathi W. *Health Care Information Systems*. Communications of the ACM, New York v.40, p. 81-82, ago. 1997.
- [2] The Open Biological and Biomedical Ontologies. <http://www.obofoundry.org>. 2012
- [3] Amazon Elastic Cloud. <http://aws.amazon.com/ec2/>. 2012
- [4] Estatísticas de Celulares no Brasil. <http://www.teleco.com.br/ncel.asp> Data ultimo acesso: 12/04/2012
- [5] Quem quer comprar tablets no Brasil?. <http://www.navegg.com/infografico-quem-quer-comprar-tablets-no-brasil/> Data ultimo acesso: 12/04/2012
- [6] Mobilidade e Oportunidades de Mudança na Saúde. <http://informationweek.itweb.com.br/8128/saude-movel-apresenta-oportunidades-de-mudanca-de-vida-diz-estudo/> Data ultimo acesso: 12/04/2012
- [7] An introduction to SOA. <http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html> Data ultimo acesso: 26/05/2012

-
- [8] Exertier F. *J2EE Deployment: The JonAS Case Study*. Première Conférence sur le Déploiement et la (Re) Configuration de Logiciels, 2004, Grenoble, France
- [9] Russel S., Norvig P. *Artificial intelligence: A modern approach*. Englewood Cliffs: Prentice Hall, 1995
- [10] Bucci G., Sandrucci V., Viacrio E. *Ontologies and Bayesian Networks in Medical Diagnosis* In: The 2011 44th Hawaii International Conference on System Sciences, 2004
- [11] V. Koufi, F. Malamateniou, G. Vassilacopoulos *A system for the provision of medical diagnostic and treatment advice in home care environment* In: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, 2008
- [12] Watcharachai Wiriyasuttiwong, Walita Narkbuakaew *Medical Knowledge-Based System for Diagnosis from Symptoms and Signs* In: INTERNATIONAL JOURNAL OF APPLIED BIOMEDICAL ENGINEERING VOL.2 (54-59), NO.1 2009
- [13] Rainer Schmidt, Stefania Montani, Riccardo Bellazzi, Luigi Portinale, Lothar Gierl *Cased-Based Reasoning for medical knowledge-based systems* In: International Journal of Medical Informatics 64 (2001) 355–367
- [14] Rodríguez A. *MEDBOLI: Medical Diagnosis Based on Ontologies and Logical Inference* In: The 2009 International Conference on eHealth, Telemedicine, and Social Medicine, 2009, Washington, DC, USA

- [15] Fenz S., Tjoa A., Hudec M. *Ontology-based generation of bayesian networks* In: International Conference on Complex, Intelligent and Software Intensive Systems. 2009.
- [16] Agnieszka Oni Ko, Peter Lucas, and Marek J. Druzdzal *Comparison of Rule-Based and Bayesian Network Approaches in Medical Diagnostic Systems* In: AIME '01 Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine (283-292)
- [17] Druzdzal M. J., Gaag L. C. *Building probabilistic networks: where do the numbers come from?* Pittsburgh: IEEE, 2000.
- [18] Wasyluk H., Onisko A., Druzdzal MJ. *Support of Diagnosis of liver disorders based on a casual bayesian network model* In: Medical Science Monitor, 7(Suppl. 1):327-332, May 2001
- [19] A. Newell, H.A. Simon. *Human Problem Solving* Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [20] A. Aamodt *Cased-Based reasoning: foundation issues* AICOM 7(1994) 39-59
- [21] Randolph A. Miller *Medical Diagnostic Decision Support Systems Past, Present, and Future* J Am Med Informatics Assoc. 1994.
- [22] Jena <http://jena.apache.org/> Data ultimo accesso: 26/05/2012
- [23] Hibernate <http://www.hibernate.org/> Data ultimo accesso: 26/05/2012

- [24] Mappa della distribuzione del mondo del numero di medici ogni mille abitanti *<http://www.indexmundi.com>* Data ultimo accesso: 07/03/2013
- [25] Tasso di penetrazione smartphone nel mercato europeo *<http://www.key4biz.it>* Data ultimo accesso: 11/03/2013
- [26] Applicazione per la gestione di dati medici *<https://www.itriagehealth.com/>* Data ultimo accesso: 11/03/2013
- [27] Quota di mercato sistemi operativi mobile *<http://www.idc.com>*
Data ultimo accesso: 11/03/2013