

POLITECNICO DI MILANO
V Facoltà di Ingegneria
Corso di Laurea Specialistica in Ingegneria Informatica
Dipartimento di di Elettronica, Informazione e Bioingegneria



**EASYFOOD: UN'APPLICAZIONE
PER LA VALUTAZIONE
SEMANTICA DELLA SIMILARITÀ
FRA RICETTE**

Relatore: **Prof. Chiara FRANCALANCI**
Correlatore: **Ing. Cinzia CAPPIELLO**

Tesi di Laurea di:

Matteo CASTELLI Matr. 783300

Domenico PANETTA Matr. 755296

Anno Accademico 2012–2013

Indice

1	Introduzione	11
2	Stato dell'arte	15
2.1	Il problema dell'obesità infantile	15
2.1.1	Cause principali dell'obesità infantile e adolescenziale .	17
2.1.2	Come viene affrontato il problema	17
2.1.3	Il ruolo della scuola	18
2.1.4	Il ruolo della famiglia	19
2.1.5	Conseguenze dell'obesità	20
2.1.6	Alcuni aspetti sociali	21
2.1.7	Fare rete per combattere l'obesità infantile	21
2.2	Ontologia	22
2.2.1	Che cos'è un'ontologia	23
2.2.2	Ontology Web Language (OWL)	23
2.2.3	Simple Protocol And RDF Query Language (SPARQL) .	25
2.3	Web Sematico	26
2.3.1	Il world wide web	26
2.3.2	Limitazioni e problematiche	27
2.3.3	Il passo successivo: Web Semantico	28
2.3.3.1	Architettura	30
2.3.3.2	eXtensible Markup Language (XML)	33
2.3.3.3	Resource Description Framework (RDF)	34
2.4	Applicazioni di dominio	39
2.4.1	Motori di ricerca semantici	39

2.4.2	Seco	41
2.4.3	Tecnologie semantiche	42
2.5	Cuciniamo	44
2.5.1	Descrizione e obiettivi	44
2.5.2	Modellazione ricetta	45
2.5.3	Grammatica e linguaggio	46
2.5.4	Implementazione	49
3	Metodologia	51
3.1	Requisiti	51
3.2	Funzionalità	53
3.2.1	Similarità semantica	56
3.2.2	Similarità per ingredienti	61
3.2.3	Similarità per nutrienti	72
3.3	Ontologia necessaria	78
4	Implementazione	81
4.1	Creazione dell'ontologia	81
4.1.1	Le classi	81
4.1.2	Le proprietà	85
4.1.3	Le istanze	87
4.2	Problematiche	88
4.2.1	Prima soluzione: reification	89
4.2.2	Seconda soluzione: molteplicità delle proprietà	90
4.2.3	Terza soluzione: creazione di un database	90
4.3	Il modello dei dati	91
4.4	Software	94
4.4.1	Architettura	95
4.4.1.1	Database	96
4.4.1.2	Semantic	98
4.4.1.3	Ontology	99
4.4.1.4	Similarity	100
4.4.1.5	Visual	101

4.4.2	Sequence Diagram	101
4.4.3	Schermate fondamentali	107
5	Risultati	111
5.1	Linguine al pesto	111
5.1.1	Similarità semantica	112
5.1.2	Similarità per ingredienti	115
5.1.3	Similarità per nutrienti	121
5.1.4	Analisi di sensitività	125
5.2	Millefoglie di melanzane	128
5.2.1	Similarità semantica	129
5.2.2	Similarità per ingredienti	131
5.2.3	Similarità per nutrienti	137
5.2.4	Analisi di sensitività	141
6	Conclusioni	145
6.1	Sviluppi futuri	147
	Bibliografia	148

Elenco delle figure

2.1	L'architettura del Web Semantico	31
2.2	Statement RDF	36
2.3	Statement RDF successivo	36
2.4	Funzionamento di un motore di ricerca semantico	40
2.5	Albero ricetta	49
3.1	Use case diagram	52
3.2	Appiattimento di una ricetta	54
4.1	La classe Alimento	82
4.2	La classe Classificazione	83
4.3	La classe Ricetta	84
4.4	Le proprietà	85
4.5	Sottoproprietà della proprietà <i>haQuantita</i>	90
4.6	Esempio della sotto ricetta pesto	94
4.7	Architettura	95
4.8	Tabella nutrienti ingrediente	97
4.9	Tabella peso ingrediente	98
4.10	File .xml generato	98
4.11	Visualizza ricetta sequence diagram	102
4.12	Crea xml sequence diagram	102
4.13	Crea ricetta wellness sequence diagram	104
4.14	Similarità semantica sequence diagram	106
4.15	Schermata principale	107
4.16	Schermata compilazione ricetta	108
4.17	Schermata similarità	109

Elenco delle tabelle

3.1	Ricetta utente e Wellness a confronto	59
3.2	Confronto fra le due ricette	68
3.3	Pesi modificati	69
3.4	Prodotto dei pesi per le quantità	69
3.5	Valori Wellness normalizzati	70
3.6	Valori wellness normalizzati	71
3.7	Valori wellness normalizzati	72
3.8	Dati per la ricetta wellness	77
3.9	Dati per la ricetta utente	78
3.10	Confronto fra le due ricette	78
5.1	ricette prese in esame	112
5.2	ricetta wellness	112
5.3	Similarità semantica giallozafferano.it	113
5.4	Similarità semantica buonissimo.org	114
5.5	Similarità semantica mangiarebene.com	115
5.6	Ricette prese in esame	116
5.7	Ricetta Wellness	116
5.8	Confronto ricette wellness, giallozafferano.it	117
5.9	Similarità quantitativa giallozafferano.it	117
5.10	Confronto ricette wellness, buonissimo.org	118
5.11	Similarità quantitativa buonissimo.org	118
5.12	Confronto ricette wellness, gustissimo.it	119
5.13	Similarità quantitativa gustissimo.it	119
5.14	Confronto ricette wellness, mangiarebene.com	120

5.15	Similarità quantitativa mangiarebene.com	120
5.16	Valori aggregati di macronutrienti per le varie ricette	121
5.17	Valori aggregati di macronutrienti per la ricetta Wellness	122
5.18	Similarità per nutrienti giallozafferano.it	123
5.19	Similarità per nutrienti buonissimo.org	124
5.20	Similarità per nutrienti mangiarebene.com	125
5.21	Versioni modificate di mangiarebene.com a confronto	126
5.22	Analisi di sensitività per mangiarebene.com	127
5.23	ricette millefoglie di melanzane prese in esame	128
5.24	ricetta millefoglie di melanzane wellness	129
5.25	Similarità semantica giallozafferano.it	129
5.26	Similarità semantica academiabarilla.it	130
5.27	Similarità semantica rossella	131
5.28	Ricette millefoglie di melanzane prese in esame	132
5.29	Ricetta millefoglie di melanzane wellness	132
5.30	Confronto ricette wellness, giallozafferano.it	133
5.31	Similarità quantitativa giallozafferano.it	133
5.32	Confronto ricetta wellness, academiabarilla.it	134
5.33	Similarità quantitativa academiabarilla.it	134
5.34	Confronto ricetta wellness, cookaround.org	135
5.35	Similarità quantitativa cookaround.org	135
5.36	Confronto ricetta wellness, rossella	136
5.37	Similarità quantitativa rossella	136
5.38	Valori aggregati di macronutrienti per le varie ricette	137
5.39	Valori aggregati di macronutrienti per la ricetta wellness	138
5.40	Similarità per nutrienti giallozafferano.it	139
5.41	Similarità per nutrienti academiabarilla.it	140
5.42	Similarità per nutrienti rossella	141
5.43	Versioni modificate di giallozafferano a confronto	142
5.44	Analisi di sensitività per giallozafferano.it	143

Capitolo 1

Introduzione

L'obiettivo fondamentale di questa tesi è quello di progettare e sviluppare un'applicazione che utilizza informazione semantica per valutare la similarità fra ricette. L'utente deve essere in grado di inserire la propria ricetta e ricevere un riscontro di similarità con una ricetta di riferimento caratterizzata da buoni principi nutrizionali. Questo lavoro è stato intrapreso nell'ambito di un progetto più ampio il cui obiettivo è di progettare e sperimentare un'applicazione per smartphone e tablet che supporti la socializzazione dei genitori in ambito pre-scolastico al fine di mitigare le difficoltà relazionali genitori-figli e rafforzare virtuosamente il mutuo rapporto reltà educativa-genitori. Tale applicazione, chiamata Easyfood, identifica la buona cucina, specialmente per i propri figli, come elemento aggregante fra genitori. In particolare, sceglie come riferimento per la buona cucina il recente paradigma della *Cucina Evolution*¹, o Cucina Welness, già personalizzato in alcuni asili nido e scuole d'infanzia secondo i requisiti specifici dell'educazione alimentare nella prima infanzia. Il progetto si propone di sperimentare una metodologia di interazione sociale che sfrutti al meglio la sinergia fra iniziative della scuola da effettuarsi nel mondo reale e l'applicazione innovativa di social networking su smartphone. Seguendo la metafora della famiglia a tavola, l'applicazione si propone di contribuire a mitigare l'emergenza educativa fornendo una best practice e un'innovativa piattaforma tecnologica riusabile in una varietà di

¹<http://www.cucinaevolution.it/>

altri contesti scolastici o educativi lombardi. Dal punto di vista funzionale due sono i principali elementi di innovazione. Il primo è che si vuole cercare di educare cambiando il meno possibile le abitudini delle famiglie, valorizzando piuttosto i loro comportamenti virtuosi. Il secondo è che non si vuole proporre l'ennesimo ricettario mascherato da app per smartphone e tablet, ma una vera e propria applicazione intelligente in grado di prendere in input una ricetta abituale di una famiglia scritta in un linguaggio naturale e:

- valutare la compatibilità della ricetta abituale con i criteri della cucina evolution
- individuare ricette simili della cucina evolution che vengono proposte come varianti più salutari della ricetta abituale
- proporre l'associazione di azioni di compensazione, previste fra le regole della cucina evolution, che permettono di riequilibrare la dieta pur mantenendo inalterate alcune ricette alle quali non si vuole rinunciare
- valutare la conformità complessiva della dieta di una famiglia coi criteri della buona alimentazione.

L'applicazione sviluppata si focalizza sui primi due punti, mentre i restanti saranno oggetto di lavori futuri. Fornirà inoltre le normali funzionalità di social networking, permettendo ai genitori di condividere ricette, di sperimentare e commentare le diverse modalità di preparazione offerte dalla cucina wellness, di proporre obiettivi educativi e di vedersi premiati con meccanismi di ranking e tecniche di "in app rewarding". L'implementazione delle funzionalità sopra elencate pone numerose sfide di ricerca in ambito informatico sui temi della data quality e della web reputation. Risulta fondamentale l'attività di ricerca e sperimentazione di algoritmi per l'identificazione di ingredienti, quantità e modalità di cottura in ricette espresse in linguaggio naturale. Inoltre è importante l'attività di ricerca per proporre e verificare empiricamente metriche di similarità fra ricette affrontando il tema di ricerca tuttora aperto della qualità dei dati non strutturati rispetto ad una base di regole che, in questo caso, sono quelle della cucina wellness.

La tesi è strutturata quindi nel seguente modo: Il secondo capitolo comincia illustrando ed esponendo il problema dell'obesità infantile con le relative conseguenze dal punto di vista sociale, per poi proseguire con la descrizione del Web Semantico che si presta come tecnologia da adottare ed impiegare in un contesto semantico. Tale capitolo si conclude con la spiegazione del software *cuciniamo* che ha permesso, grazie alla sua funzione di parser per ricette, di poter modellare e strutturare una ricetta scritta attraverso un testo libero. Il capitolo tre descrive la metodologia utilizzata per poter progettare l'applicazione: si definiscono i requisiti e le funzionalità necessarie, soffermandosi sul concetto di similarità e sull'ontologia, ritenuta come le fondamenta per sviluppare il progetto in un quadro semantico. Il quarto capitolo riguarda l'implementazione vera e propria: verrà presentata la creazione dell'ontologia con le relative problematiche, il modello dei dati e l'architettura del software. Con il quinto capitolo si mostreranno i risultati ottenuti attraverso cui, con l'ultimo capitolo, si esporranno le conclusioni e gli sviluppi futuri.

Capitolo 2

Stato dell'arte

Il capitolo si apre con un'analisi sistematica della letteratura riguardante l'obesità infantile. La diffusione di tale problema, con le annesse conseguenze a carattere sociale oltre che fisico, destano sempre maggiori preoccupazioni per genitori e specialisti del settore. Verrà presentata poi una sezione riguardo l'ontologia, rappresentazione formale che si è scelta di utilizzare per progettare l'applicazione Easyfood nel dominio di interesse appena descritto in cui mancano sistemi ed applicazioni in grado di fornire un supporto semantico all'utente. In particolare, tale rappresentazione, sarà in grado di offrire informazione e conoscenza all'utente riguardo la classificazione alimentare e la scienza nutrizionale che, oltre ad essere le linee di guida della Cucina Wellness, possono incidere notevolmente sull'educazione alimentare.

2.1 Il problema dell'obesità infantile

Il tasso di obesità nei paesi del Nord America e in Europa è vertiginosamente cresciuto negli ultimi decenni. Quasi tutti gli studi revisionati, sono concordi sul fatto che il problema dell'obesità infantile ha cominciato ad essere un'emergenza sociale a partire dagli anni '80. Nei pesi sopra citati, approssimativamente il 31% dei bambini e degli adolescenti può essere considerato obeso o comunque in sovrappeso [Salvy, 2012].

I bambini dai 6 agli 11 anni che eccedono il 95° percentile del BMI (Body Mass Index) per essere considerati in sovrappeso, è aumentata dal 4% nel 1971-1974 al 18,8% nel 2003-2004, e la percentuale degli adolescenti obesi (dai 12 ai 19 anni) è aumentata dal 4,6% nel 1966-1970 al 17,4% nel 2003-2004 [Harris, 2009]. Questi dati sono stati misurati negli USA. Dati simili sono stati trovati in Canada, UK e nel resto d'Europa. Oltre che per la salute delle popolazioni, l'obesità rappresenta un problema anche dal punto di vista economico: un altro studio USA ci conferma che i costi ospedalieri annuali per i bambini sono più che triplicati durante gli anni '80 e '90, andando dai 35 M\$ ai 127 M\$ [Desjardins, 2007].

Nonostante l'avvento dell'agricoltura 14000 anni fa abbia assicurato forniture di cibo prevalentemente stabili, le attività della vita quotidiana comportavano sostanziali spese di energia fino a 50 anni fa, quando si sono verificati dei cambiamenti radicali nel bilancio di energia acquisita/consumata dall'uomo. A supporto di questa tesi, popolazioni del Nord America che hanno preservato uno stile di vita tradizionale e privo delle tecnologie di base, hanno una scarsa probabilità di contrarre l'obesità [Han, 2010]. Come per altre condizioni, il tasso di obesità è più alto nelle minoranze etniche (studio USA) e per i bambini svantaggiati. Bambini afro-americani e ispanici hanno maggiori probabilità di contrarre l'obesità. Fra i bianchi invece il tasso di obesità è maggiore in quegli stati (ad esempio Mississippi, Louisiana e Kentucky) dove è diffusa la povertà infantile [Desjardins, 2007]. A rendere ancora più urgente la questione del sovrappeso c'è il fatto che non si tratta di un problema strettamente relegato ai paesi sviluppati; ma si riscontrano tassi preoccupanti di obesità infantile, seppur per cause differenti, anche nei paesi in via di sviluppo [Gupta, 2012, Sharma, 2011]. Si tratta di una questione che è necessario affrontare con la massima urgenza. A differenza di molte altre malattie, l'obesità rappresenta un problema anche perché quando un individuo è obeso sin da piccolo ha una probabilità altissima che lo rimanga per il resto della vita.

2.1.1 Cause principali dell'obesità infantile e adolescenziale

Per modificare i recenti trend riguardanti l'obesità, è necessario indagare quelle che sono le cause principali del guadagno di peso. La letteratura, a seconda del caso specifico, presenta un ventaglio piuttosto ampio di possibilità. Nonostante questo, anche qui quasi tutti concordano nell'individuare nell'eccessiva e sbagliata assunzione di cibo e nella scarsa attività fisica i due peggiori nemici di un corpo sano e in forma. Generalmente, l'eccessivo consumo di energie è dovuto alle cattive abitudini alimentari dei piccoli; affezionati consumatori di fast-food, merendine e bevande con altissime percentuali di zucchero. Riguardo alla scarsa attività fisica, questa è da attribuirsi ad un continuo adattamento ad uno stile di vita sempre più sedentario. Indagati questi due fattori principali, ulteriori approfondimenti relazionano l'indice di massa corporea con tutta una serie di elementi che vanno da quelli metabolici a quelli comportamentali per non dimenticare di quelli sociali, culturali, economici e genetici [Sharma, 2011]. Per fare un esempio è interessante notare come nei paesi sviluppati sono le fasce sociali più deboli a soffrire maggiormente l'obesità, mentre nei paesi in via di sviluppo, l'obesità è maggiormente diffusa nella parte di popolazione gode di una migliore condizione economica [Gupta, 2012].

2.1.2 Come viene affrontato il problema

Le soluzioni per questo tipo di problema si suddividono in farmacologiche e non farmacologiche. Naturalmente il presente lavoro si focalizza essenzialmente sulle seconde. Molti degli approcci non farmacologici sono basati su vari studi comportamentali. Il successo nel cambiamento dello stile di vita ha molto a che fare con la forza di volontà, con i comportamenti intenzionali come l'autocontrollo, la capacità di imporsi degli obiettivi e quella di sapersi ricompensare a seguito di piccoli successi ottenuti quotidianamente [Han, 2010]. Le strategie più note per combattere questa emergenza sociale sono: promozione dell'allattamento dal seno, limitare la televisione, incorag-

giare l'attività fisica, aumentare il consumo di frutta e verdura, rivedere le porzioni e limitare le bevande zuccherate [Sharma, 2011].

Da non sottovalutare invece è il ruolo della prevenzione. La prevenzione, in special modo nella popolazione più giovane, è universalmente riconosciuta come il miglior approccio per contrastare la crescente prevalenza dell'obesità. Alcuni strumenti di prevenzione sono le campagne sui mass media (promozione dello sport, sottolineare i benefici del mangiare sano, quelli del mangiar male) e politiche che vanno dall'ente locale fino ai provvedimenti nazionali. Anche i produttori alimentari, nonché i ristoratori si sono adeguati; soprattutto negli ultimi vent'anni è aumentata l'attenzione e la pressione per la pubblicazione del contenuto calorico sui vari cibi e prodotti [Han, 2010].

2.1.3 Il ruolo della scuola

Partendo dal presupposto che i bambini trascorrono buona parte del loro tempo a scuola, pare naturale coinvolgere gli istituti nella prevenzione dell'obesità. Le scuole rappresentano un ottimo luogo per la promozione di comportamenti salutari. C'è molto interesse nella prevenzione dell'eccessivo accumulo di grasso corporeo nei bambini. Le scuole vengono identificate come importanti veicoli per la promozione della salute a causa del significativo ruolo che esse svolgono nella vita del bambino. La fornitura dei pasti, la disponibilità delle attività fisiche e la possibilità di progettare l'ambiente in cui il bambino impara danno alle scuole ineguagliabili opportunità per promuovere la gestione del peso.

In molte scuole sono stati messi in atto programmi di prevenzione che hanno due caratteristiche principali: il cambiamento del contenuto calorico dei pasti e l'incoraggiamento a fare attività fisica. Una norma, fonte di dibattito negli USA, consiste nella rimozione dalla scuola di macchinette che vendono snack e bibite zuccherate. Comunque, uno studio USA ha dimostrato che solo 1,3 % delle calorie assunte nell'arco della giornata provenga dagli snack mangiati a scuola, mentre le merendine e gli snack mangiati a casa apportano ben il 69,1% delle calorie giornaliere [Han, 2010]. L'analisi di vari programmi di intervento mirati sull'attività fisica, ha inoltre mostrato

che questi programmi raramente incidono in maniera decisiva sull'indice di massa corporea dei bambini [Harris, 2009].

I punti su cui i vari programmi di prevenzione fanno forza sono essenzialmente 4:

- dieta
- attività fisica
- educazione ad uno stile di vita sano
- coinvolgimento dei genitori

Un programma può far leva su uno di questi punti o su una combinazione di vari punti [Harrison, 2010]. Al riguardo c'è anche da dire che sono molteplici i vari fattori per una buona riuscita di un intervento in ambito scolastico. Molti ricercatori hanno notato che la riuscita di questi programmi dipende anche dalla motivazione, dalla qualità e capacità di training dello staff della scuola, nonché dalla loro capacità di rispettare il programma stabilito. Generalmente programmi di questo genere devono trovare spazio in timeline già pieni di cose da fare per i bambini, e questo ne condiziona il successo oltre che la sostenibilità. Per un approfondimento su questi programmi è possibile consultare la bibliografia [Harrison, 2010, Nakamura, 2008, Carter, 2008, Harris, 2009].

2.1.4 Il ruolo della famiglia

Per la maggior parte degli individui -e ciò è tanto più vero per i bambini- la famiglia è ciò che ha maggiore influenza su di loro. E obiettivi come la perdita di peso hanno maggiore probabilità di successo in caso di intero coinvolgimento del nucleo familiare. Le famiglie rappresentano un supporto dal punto di vista della motivazione. La soluzione sembrerebbe semplice: assumere meno calorie di quelle che si spendono, ma per molte persone questo rimedio è una vera e propria sfida. A volte i fallimenti arrivano per la scarsa capacità di mantenere i cambiamenti in termini di comportamenti richiesti. A meno che non ci sia supporto. E non c'è supporto migliore della famiglia e della scuola. E' stato stabilito [Gupta, 2012] che le caratteristiche fisiche,

normative e sociali della famiglia influenzano l'adozione e il mantenimento di comportamenti che promuovono uno stile di vita sano. Le dinamiche familiari includono regole, supporto emotivo, incoraggiamento, rinforzo dagli altri membri della famiglia [Gruber, 2009].

All'interno del contesto familiare la pianificazione dei pasti, l'acquisto di cibo, la preparazione dei pasti, la consumazione del pasto, i momenti di ricreazione familiare e le attività sedentarie sono tutte opportunità di intervento. La famiglia fornisce il primo ambiente sociale in cui il bambino apprende e stabilire delle buone abitudini alimentari può essere determinante. In buona sostanza la famiglia gioca un ruolo critico nello sviluppo e nella riduzione dell'obesità. Alcune ricerche suggeriscono che determinate abitudini alimentari sviluppate nell'infanzia persistono anche nell'età adulta [Gruber, 2009]. Lo stesso vale per l'attività fisica. Essenzialmente combattere l'obesità infantile consiste nel creare all'interno della famiglia un ambiente che promuove abitudini salutari. I genitori devono imparare a parlare con i loro bambini sull'esercizio fisico e sul mangiar bene e devono incoraggiarli ad essere più attivi. Molti genitori semplicemente si rifiutano di riconoscere che i loro bambini sono obesi. Ne consegue che non fanno nulla per risolvere il problema [Gruber, 2009].

2.1.5 Conseguenze dell'obesità

L'obesità ha conseguenze che si sviluppano sotto vari aspetti: medici, sociali e psicologici [Gupta, 2012]. Il sovrappeso e l'obesità sono associati in effetti ad un gran numero di conseguenze mediche di lungo termine: generici disordini alimentari, diabete di tipo 2, ipertensione, ictus, steatosi epatica, artrite, apnea del sonno, colecisti e asma bronchiale. Alcuni studi hanno dimostrato che circa il 60% dei bambini che raggiungono il decimo anno di età in sovrappeso presentano almeno un fattore di rischio cardiovascolare, e il 25% ne ha più di 2 [Sharma, 2011]. Dal punto di vista psicologico le conseguenze sono invece: depressione, discriminazione, pregiudizio, insoddisfazione del proprio corpo, scarsa autostima, rifiuto dei pari, stigmatizzazione [Sharma, 2011, Carter, 2008].

2.1.6 Alcuni aspetti sociali

E' stato dimostrato che i bambini e gli adolescenti la cui rete sociale (familiari e compagni di scuola) che comprende individui in sovrappeso, sono molto più propensi a sottostimare il proprio peso e quindi a sviluppare una percezione inaccurata di quello che dovrebbe essere il corretto status di peso. Benché i genitori abbiano una fortissima influenza sulle credenze e sui comportamenti dei propri figli, c'è una crescente evidenza del fatto che bambini e adolescenti siano sempre più condizionati dalle abitudini alimentari e fisiche dei propri coetanei [Salvy, 2012].

Alcuni studi hanno dimostrato che per gli adulti è molto probabile contrarre l'obesità nel giro di tre decenni se i loro amici sono in sovrappeso o obesi. Un fenomeno analogo accade per i bambini. Probabilmente non si tratta tanto di imitazione del comportamento, quanto del fatto che c'è uno slittamento nella percezione dell'obesità come un problema. In buona sostanza quando un individuo vede le persone che frequenta guadagnare (o perdere) peso, la norma sociale su ciò che è socialmente accettabile o meno, tende a modificarsi. Bambini e adolescenti il cui network sociale (familiari e compagni di scuola) comprende individui sovrappeso sono molto più esposti al rischio di sottostimare il proprio peso [Salvy, 2012].

Indipendentemente dal paese di provenienza, le abitudini alimentari sono profondamente radicate nella cultura della famiglia, che rappresenta a volte la loro identità. Le famiglie devono combattere le influenze esterne che condizionano la disponibilità dei cibi preferiti e con l'introduzione di nuovi cibi e nuovi metodi di preparazione. E' stato dimostrato [Gruber, 2009] che molto spesso, i cambiamenti nella dieta riguardano soprattutto le persone che migrano in un altro paese laddove assumono la nuova cultura.

2.1.7 Fare rete per combattere l'obesità infantile

La lotta all'obesità infantile necessita di uno sforzo a livello comunitario. Molte iniziative avviate in California nel 2005 hanno dimostrato che un approccio collaborativo che coinvolge molteplici settori può notevolmente ridurre i fattori di rischio ambientale che portano all'obesità. Il lavoro in

collaborazione fra molte entità: la scuola, programmi dopo scuola, media e pubblicità . . . può portare i giovani, le famiglie ,leader delle comunità , professionisti della salute a produrre un sforzo comune a ridurre sensibilmente l'obesità. Il report del 2004 dell'istituto di medicina (IOM) **“Preventing Childhood obesity: Health in the Balance”** sostiene che la rete allargata di professionisti quali insegnanti, amministratori dell'educazione, professionisti della salute, manager e leader politici possono sostenere con successo questa causa. Il report ha inoltre suggerito un approccio a più facce che coinvolge il governo, l'industria, società di marketing e di pubbliche relazioni, nonché le scuole e le famiglie. Scuole e famiglie che poi sono gli elementi più importanti dato che poi sono loro a stare a stretto contatto con il bambino. Fondamentalmente, per una buona riuscita dei programmi di prevenzione, deve esserci una soluzione di continuità fra questi due attori: il bambino deve percepire una sorta di coerenza tra quello che fa a scuola e le abitudini alimentari che gli vengono suggerite dal contesto familiare.

2.2 Ontologia

Questo capitolo vuole concentrarsi sul ruolo fondamentale che l'ontologia assume nel contesto in analisi, ovvero nello sviluppo dell'applicazione semantica, potendo offrire un forte supporto grazie ai suoi numerosi vantaggi. Si è sottolineato in precedenza che nello scenario emergente del Web Semantico diventa importante descrivere le risorse web con metadati e poter usufruire del supporto di ontologie. Nel Web Semantico è possibile rappresentare, condividere ed esportare la conoscenza, utilizzando un linguaggio per esprimere dati e regole, per cui le macchine possono accedere ad un insieme strutturato di informazioni e a un insieme di regole di inferenza da utilizzare per il ragionamento automatico. Le tecnologie del Web Semantico consentono perciò di realizzare sistemi informativi che utilizzano ontologie per la ricerca, l'integrazione e l'analisi, migliorando nettamente la qualità dell'informazione.

2.2.1 Che cos'è un'ontologia

Il termine ontologia deriva dalla filosofia, dove viene inteso come una spiegazione sistematica dell' essere. Negli anni recenti il termine si è ampiamente diffuso nella comunità del Knowledge Engineering. Esistono diverse definizioni di ontologia, a partire da quella data in [Neches, 1991], ognuna delle quali mette in evidenza qualche aspetto. Una definizione su cui c'è ampia convergenza è quella di [Studer, 1998], che riprende e chiarisce quelle date precedentemente in [GruberT] e [Borst, 1997]:

An ontology is a formal, explicit specification of a shared conceptualisation. A 'conceptualisation' refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. 'Explicit' means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal and a constraint is that a disease cannot cause itself. 'Formal'

refers to the fact that the ontology should be machine readable, which excludes natural language. 'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

Da notare che un' ontologia include non solo i termini che sono esplicitamente definiti in essa, ma anche la conoscenza che ne può essere derivata mediante un processo di inferenza. Va posto l'accento su come le ontologie possano essere condivise e riutilizzate tra applicazioni e gruppi di persone diversi.

2.2.2 Ontology Web Language (OWL)

Per costruire relazioni semantiche e processarle in modo automatico è necessaria una formalizzazione che le descriva in maniera completa. Questo compito è svolto dallo strato comprendente le ontologie e i relativi linguaggi. Un'ontologia permette di specificare in modo significativo i concetti e le

relazioni che caratterizzano un certo dominio di conoscenza, in particolare, secondo una delle definizioni più usate, *un'ontologia è una specificazione di una concettualizzazione [Gruber T, 1995]*. Le ontologie sono applicate per la maggior parte nel campo dell'intelligenza artificiale e nella rappresentazione e nella condivisione della conoscenza. I software possono utilizzare un'ontologia per una varietà di scopi, fra cui il ragionamento deduttivo, la classificazione, diverse tecniche di problem solving, oltre che per facilitare la comunicazione e lo scambio di informazioni fra diversi sistemi. Il linguaggio standard raccomandato dal W3C per la definizione di ontologie è l' *Ontology Web Language (OWL)*. Lo scopo di OWL è descrivere delle basi di conoscenze, effettuare delle deduzioni su di esse e integrarle con i contenuti delle pagine web in modo tale che sarà possibile effettuare delle ricerche estremamente complesse nel web evitando i problemi di omonimia e ambiguità presenti nelle normali ricerche testuali. Esistono tre versioni OWL aventi capacità espressive differenti tra loro: OWL Lite, OWL-DL ed OWL Full. Questi tre linguaggi sono differenti anche per potenza espressiva. La logica del primo ordine è infatti molto potente ma non è decidibile, ossia non è possibile costruire un algoritmo che, dati uno o più assiomi, dica in tempo finito se un'affermazione segue o non segue logicamente da tali assiomi. Ovviamente questo è sconveniente dato che si vorrebbe avere una base di conoscenza che risponda in maniera prevedibile alle richieste, e non dia informazioni in base alle deduzioni fatte fino a quel momento. Un'ontologia dunque è associabile ad una teoria logica in quanto dev'essere espressa in un linguaggio dichiarativo, dev'essere caratterizzata da un linguaggio formale e poter effettuare regole d'inferenza complete e corrette. Le prime due versioni di OWL si basano sulle cosiddette Logiche Descrittive (DL, *Description Logic*), sottoclassi e meno potenti rispetto alla logica del primo ordine, con meno potere espressivo, ma che sono decidibili. Sono pensati per essere processati automaticamente permettendo di ottenere notevoli informazioni semantiche aggiuntive a partire da una data ontologia. In particolare OWL Lite utilizza SHIF, semplice da implementare ma ha basso potere espressivo e OWL DL utilizza SHOIN, decidibile con procedure di deduzione di complessità ben conosciute. OWL Full invece usa la logica del primo ordine, espandendola con altri predicati

diventando così molto espressivo ma allo stesso tempo indecidibile. Riassumendo OWL è stato progettato per essere uno strato dell'architettura del Web semantico, estendendo e completando i concetti visti in precedenza:

- XML fornisce una sintassi per documenti strutturati ma non impone vincoli semantici sul significato dei documenti
- XML Schema è un linguaggio per limitare la struttura dei documenti XML ed estende anche XML attraverso i *datatype*
- RDF è un modello dei dati (*datamodel*) per gli oggetti (le risorse) e le relazioni fra loro, fornisce una semplice semantica e questi modelli dei dati possono essere rappresentati da una sintassi XML
- RDF Schema è un vocabolario per descrivere proprietà e classi delle risorse RDF con una semantica per la generalizzazione gerarchica di tali classi e proprietà
- OWL aggiunge più potere al vocabolario per descrivere proprietà e classi definendo: relazioni tra classi (ad esempio disgiunzione), cardinalità (“esattamente uno”), disuguaglianza e relazioni tra proprietà (ad esempio simmetria, transitiva)

2.2.3 Simple Protocol And RDF Query Language (SPARQL)

Il Simple Protocol and RDF Query Language, noto come SPARQL, è un linguaggio d'interrogazione per dati RDF sviluppato da W3C e dal 2008 rappresenta uno standard per il Web Semantico. SPARQL consente di estrarre informazioni dalle basi di conoscenza distribuite sul web, rendendo possibile la costruzione di query basate su triple patterns, congiunzioni logiche, disgiunzioni logiche e pattern opzionali. Permette quindi di identificare delle regole standard tra le strutture dati presenti in rete, in modo da rendere le realtà del Web interoperabili tra loro. SPARQL non fa altro che visitare i documenti per ricercare dei sotto-grafi che corrispondano a ciò di cui ha

bisogno l'utente che esplicita la richiesta mediante la query. Può essere detto che SPARQL agirebbe sulle risorse Web (RDF) come SQL agisce su un database: interrogando l'insieme dei dati (strutturati), sarebbe in grado di ottenere e restituire conoscenza (sottoforma di relazioni, proprietà, etc..). In breve le query SPARQL si basano sul meccanismo del *pattern matching* e in particolare su un costrutto, il *triple pattern*, che ricalca la configurazione a triple delle asserzioni RDF fornendo un modello flessibile per la ricerca di corrispondenze.

?titolo cd:autore ?autore.

In luogo del soggetto e dell'oggetto questo "triple pattern" prevede due variabili, contrassegnate con ?. Le variabili fungono in un certo senso da incognite dell'interrogazione, cd:autore funge invece da costante: le triple RDF che trovano riscontro nel modello associeranno i propri termini alle variabili corrispondenti. Per chiarire, ecco una semplice query di selezione SPARQL:

```
PREFIX cd: <http://example.org/cd/>  
SELECT ?titolo ?autore ?anno  
FROM <http://cd.com/listacd.ttl>  
WHERE {?titolo cd:autore ?autore. ?titolo cd:anno ?ann . }
```

2.3 Web Semantico

2.3.1 Il world wide web

Come sappiamo il World Wide Web è un servizio di internet che permette di navigare ed usufruire di un insieme vastissimo di contenuti (multimediali e non) collegati tra loro in modo non sequenziale attraverso legami, chiamati link, e di ulteriori servizi accessibili a tutti o ad una parte selezionata degli utenti internet. I contenuti sono organizzati nei siti web a loro volta strutturati nella pagine web le quali si presentano come composizioni di testo e/o grafica visualizzate sullo schermo del computer del browser web. Tutti i siti

web sono identificati da un indirizzo web, una sequenza di caratteri univoca chiamata URL che ne permette la rintracciabilità. Nel corso degli anni sono nati ed hanno riscosso notevole successo i cosiddetti motori di ricerca, siti web da cui è possibile ricercare contenuti nel Web in modo automatico sulla base di parole chiave inserite dall'utente, e i cosiddetti portali web, siti web da cui è possibile accedere ad ampie quantità di contenuti del Web selezionati dai redattori del portale web attraverso l'utilizzo di motori di ricerca o su segnalazione dei redattori dei siti web.

2.3.2 Limitazioni e problematiche

Le dimensioni del web sono enormi, lo possiamo immaginare con una gigantesca biblioteca di pagine HTML on-line. Se da un lato lo standard HTML con la sua semplicità ha contribuito all'affermazione del Web, dall'altro ha la grossa limitazione di occuparsi solo ed esclusivamente della formattazione dei documenti, tralasciando del tutto la struttura e il significato dei contenuti. Le informazioni contenute nel Web sono destinate ad utenti umani e la ricerca delle informazioni è molto più complicata di quanto si vorrebbe, infatti i problemi che ne conseguono sono dovuti alla maniera in cui l'informazione è memorizzata nel Web:

- Non c'è niente in un documento HTML che indichi l'argomento trattato o la fonte delle informazioni
- Un calcolatore non può comprendere il significato di un documento
- L'unico tipo di ricerca possibile è quello sintattico (per parole chiave), usando le tecniche classiche di information retrieval per documenti non strutturati. Le difficoltà sono dovute a:
 1. Polisemia e omonimia, ovvero una parola con più significati
 2. Sinonimia, più parole con lo stesso significato.

Questo pone notevoli difficoltà nel reperimento e riutilizzo delle informazioni. Per rendersi conto di questo è sufficiente eseguire una ricerca utilizzando uno

dei motori disponibili in rete e ci si accorgerà che, delle migliaia di documenti risultanti dalla query, spesso solo una piccola percentuale è d'interesse per la ricerca che s'intendeva fare. Ad esempio, per qualsiasi motore di ricerca, non esiste nessuna differenza fra il termine *Marrone* nel contesto *Luca Marrone*, calciatore italiano, e *marrone* nel contesto *il colore marrone*. Per questo motivo si è avuta la necessità di separare la presentazione dal contenuto, ovviando il problema dovuto al linguaggio HTML, attraverso l'introduzione nel 1998 dello standard XML (eXtensible Markup Language) da parte dello stesso inventore del Web, Tim Berners-Lee. Il metalinguaggio XML consente la creazione di nuovi linguaggi di marcatura e la sua caratteristica innovativa è la possibilità di aggiungere informazioni sui contenuti attraverso la definizione di opportuni tag fissati dall'utente, permettendo così di esprimere proprietà logiche e strutturali dell'informazione presentata. Con queste caratteristiche l'XML fornisce un modo comune di rappresentare i dati, cosicché i programmi software siano in grado di eseguire meglio ricerche, visualizzare e manipolare informazioni nascoste nell'oscurità contestuale. Anche l'XML, però, ha delle limitazioni in quanto diversi documenti XML possono esprimere informazioni con lo stesso significato utilizzando etichette diverse, così come informazioni con significato diverso utilizzando le stesse etichette, come si può notare in un DTD o in un XML Schema, i quali non permettono di specificare il significato delle etichette ma solo la struttura ammessa per i documenti. Risulta quindi che la sintassi XML non definisce alcun meccanismo esplicito per qualificare le relazioni tra documenti e la condivisione dell'informazione rimane ancora difficile come, d'altronde, la questione dell'integrazione d'informazione proveniente da fonti diverse.

2.3.3 Il passo successivo: Web Semantico

Lo scenario attuale del World Wide Web è quello quindi di un enorme insieme di testi collegati in qualche modo tra loro. Gli utenti umani si orientano nel Web grazie alla loro esperienza di navigazione e alla capacità di inserimento di parole o espressioni chiave. Queste caratteristiche, proprie dell'essere umano, non appartengono invece a nessuna applicazione, a nessun agente

software, di interpretare il contenuto delle risorse memorizzate nel Web. Il Web semantico è un'estensione dell'attuale World Wide Web. Fornisce un modo standard di esprimere le relazioni tra pagine web, per permettere ai calcolatori di comprendere il significato dei collegamenti ipertestuali, un modo nuovo di concepire i documenti per il World Wide Web. Si riporta la definizione di Tim Berners-Lee[Berners-Lee, 2001]: *"Il Web Semantico è un'estensione del Web corrente in cui le informazioni hanno un ben preciso significato e in cui computer e utenti lavorano in cooperazione"*. In questo ambiente è prevista la diffusione di agenti intelligenti che devono essere in grado di:

- comprendere il significato dei testi presenti sulla rete
- creare percorsi in base alle informazioni richieste dall'utente, guidandolo poi verso di esse
- spostarsi di sito in sito collegando logicamente elementi diversi dell'informazione richiesta

Il W3C¹ (World Wide Web Consortium) considera il Semantic Web come l'ideale evoluzione del Web da machine-representable a machine-understandable, e per far sì che avvenga è necessario trasformare il Web in una grande base di conoscenza. Le principali applicazioni che possono svilupparsi da questo scenario sono particolari motori di ricerca basati su concetti, informazioni multimediali personalizzate, servizi di e-business in quanto attualmente non esistono buoni metodi di ricerca per un cliente che cerchi un certo servizio completo, composto da più funzionalità (è' necessario costruire wrapper per rendere accessibili-usabili i servizi sul Web: si estraggono solo informazioni esplicite). Dunque nel Semantic Web gli agenti software "comprendono", attraverso processi di ragionamento che sfruttano le capacità razionali proprie di un agente intelligente, le descrizioni dei servizi messi a disposizione. È possibile anche sviluppare un browsing automatico per la ricerca web, un sito web costruito seguendo logiche semantiche vuol consentire ad alcuni software di

¹www.w3c.org

interpretare il contenuto di un testo imitando il meccanismo di lettura e comprensione degli esseri umani e, invece di utilizzare wrapper procedurali per alcuni servizi complessi, l'integrazione avviene in maniera dichiarativa, mediante ontologie. Occorre quindi un framework che permetta di esprimere tutto questo mare di conoscenza in modo che sia machine-understandable, ovvero che un calcolatore possa cogliere il significato del suo contenuto, capacità fin'ora, sviluppata solo dagli utenti umani. La condivisione della conoscenza necessita innanzi tutto di un linguaggio che:

- abbia una sintassi che assicuri l'interoperabilità, proprietà fondamentale nei servizi informatici di oggi giorno
- abbia una semantica condivisa
- permetta di esprimere ontologie
- permetta elaborazioni efficienti.

Inoltre è necessaria un'infrastruttura che consenta di:

- aggiungere metadati ai documenti esistenti
- referenziare entità e risorse in maniera univoca
- integrare diverse fonti di conoscenza

2.3.3.1 Architettura

L'idea del Web Semantico nasce estendendo l'idea di utilizzare schemi per descrivere domini d'informazione. Dei metadati (ovvero dati sui dati, sono informazioni comprensibili da una macchina e relativi ad una risorsa Web o qualche altra cosa, che possono essere estratti da una risorsa o possono essere trasferiti con il documento) devono mappare i dati rispetto a classi, o concetti, di questo schema di dominio. Il Web Semantico è, come l'XML, un ambiente dichiarativo, in cui si specifica il significato dei dati, e non il modo in cui si intende utilizzarli. La semantica dei dati consiste nel dare alla macchina delle informazioni utili in modo che essa possa utilizzare i dati nel

modo corretto, convertendoli eventualmente. Rispetto a un punto di vista generale il Web semantico è costituito da tre livelli principali:

1. I dati, definiti in modo strutturato tramite l'XML
2. I metadati, informazioni sui dati gestite tramite il linguaggio RDF
3. Le ontologie, usate 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.

Per chiarezza si specifica l'architettura completa, derivante dalla visione del suo creatore Tim Barners-Lee:

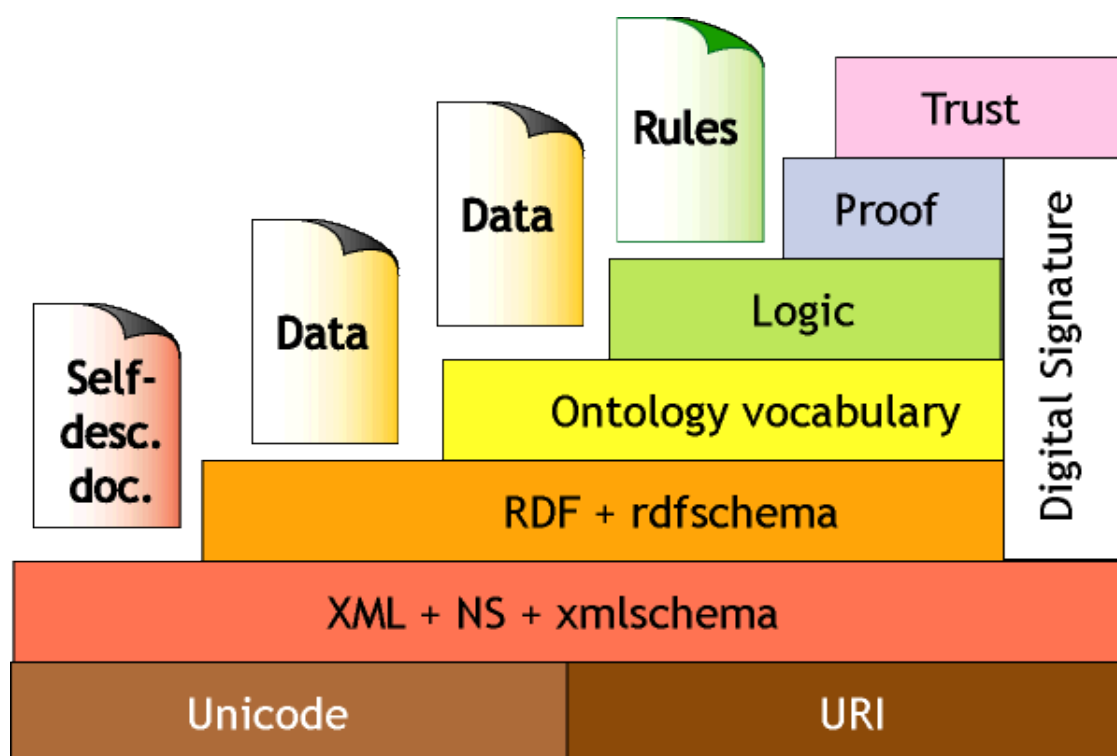


Figura 2.1: L'architettura del Web Semantico

Lo sviluppo completo di questa infrastruttura non è ancora avvenuto a causa del lungo lavoro necessario, sono state però gettate le fondamenta, a partire dal lavoro che è iniziato con l'avvento del linguaggio XML. La

sfida del semantic web è fornire un linguaggio per esprimere dati e regole per ragionare sui dati, che consenta l'esportazione sul web delle regole da qualunque sistema di rappresentazione della conoscenza. Vediamo i livelli in dettaglio del diagramma:

1. URI, Uniform Resource Identifiers, stringa necessaria per la definizione univoca di indirizzi internet ma anche di qualsiasi risorsa e documento
2. XML, gioca un ruolo fondamentale con i namespace e gli XML Schema, consente di dare ai documenti una struttura arbitraria, rendendo possibile modellare la realtà che si considera fornendo un insieme di regole sintattiche. Questo linguaggio ha il vantaggio di garantire un'alta interoperabilità dei dati, anche se soffre di difficoltà come lo scambio d'informazione tra diverse realtà, favorendo lo sviluppo del successivo linguaggio
3. RDF, Resource Description Framework, e RDF Schema usano la sintassi XML, sono usati per esprimere il significato delle risorse, per descrivere i loro tipi e per asserire eventuali proprietà tra di esse. Le asserzioni sono realizzate tramite triple che legano tra loro gli elementi di una relazione binaria. Le triple sono del tipo: soggetto (risorsa), predicato (proprietà) e oggetto (valore). Un modello RDF è rappresentabile da un grafo orientato sui cui nodi ci sono risorse o tipi primitivi e i cui archi rappresentano le proprietà.
4. livello ontologico. Una ontologia permette di descrivere proprietà e relazioni tra tipi di elementi senza però fornire informazioni su come utilizzare queste relazioni dal punto di vista computazionale
5. la firma digitale è di significativa importanza. La crittografia a chiave pubblica è la tecnica più conosciuta, potrebbe essere utilizzata per stabilire la provenienza delle ontologie oltre che dei dati
6. per ultimo il livello logico, attraverso cui le asserzioni presenti sul Web possono essere utilizzate per derivare nuova conoscenza.

2.3.3.2 eXtensible Markup Language (XML)

Il linguaggio di partenza è l'XML. Con esso diventa possibile modellare i dati dotandoli di un significato (quindi di una semantica) embrionale. Nella cerchia del Web Semantico, l'XML viene considerato il primo linguaggio che struttura e si fa portatore di informazioni sulla semantica delle risorse. L'XML può essere visto come un metalinguaggio con cui è possibile costruire altri linguaggi di markup. L'XML si allontana notevolmente dall'HTML, in quanto in quest'ultimo, infatti, i tag utilizzabili per definire gli elementi di un documento sono predefiniti e invariabili. Nel caso dell'XML, al contrario, i tag possono essere creati liberamente e sono quindi personalizzabili. Nella visione del Web Semantico l'XML sancirebbe l'ingresso della semantica nel Web. Questo linguaggio permetterebbe di aggiungere contenuto, non solo machine-readable, ma anche interpretabile in maniera differente a seconda di come verrebbe istruito il sistema che lo interpreta. Per far ciò sarebbe sufficiente dichiarare l'esistenza di blocchi semantici minimi (i tag) e definire una grammatica con delle regole sintattiche volta ad assicurare, proprio attraverso i tag, la correttezza formale nell'uso degli oggetti costruiti. Uno dei vantaggi dell'utilizzo dell'XML come (meta)linguaggio di markup universale risiede nella possibilità di accedere a informazioni provenienti da più fonti; tecnicamente infatti, un documento XML può utilizzare anche più di una grammatica. Tale caratteristica rappresenta sicuramente un grande vantaggio, ma, proprio in virtù di ciò essa porta con sé anche una serie di inconvenienti, uno tra tutti, il problema della disambiguazione del significato o polisemia. I documenti strutturati, nel momento in cui vengono confrontati o integrati, creano inevitabilmente conflitti tra termini e tag. Ciò dipende essenzialmente dal fatto di essere sviluppati in maniera indipendente rispetto alla grammatica. Il problema che si viene, dunque, a creare risiederebbe nella necessità di trovare un modo per identificare esattamente l'ambito di ciascun elemento, conciliare la presenza di elementi definiti in uno di più vocabolari, e soprattutto conciliare la presenza di elementi definiti con lo stesso nome in più vocabolari diversi. I namespace in XML si propongono di risolvere tali questioni, essi permettono di etichettare ogni tag o attributo generato in

un documento XML, mediante degli URI che identificano tali tag in maniera univoca sul Web. Il namespace è un metodo adottato all'interno dei documenti XML (e in seguito anche RDF) per evitare i conflitti semantici tra tag di identificazione. Questo sistema si pone come una possibile soluzione dei problemi di polisemia delle parole. Le componenti di XML sono: il contenuto (XML), le specifiche che riguardano gli elementi ossia la struttura (DTD, Document Type Definition o XMLSchema) e le specifiche che riguardano la visualizzazione ossia lo stile (XSL eXtensible Style Language). L'XML raggiunge interoperabilità sintattica, ovvero la capacità dei sistemi ad interpretare la sintassi e la struttura dei documenti scambiati, ma anche l'interoperabilità strutturale, parzialmente, cioè la possibilità di interpretare le strutture dei documenti di schemi logici differenti, conoscendo le regole di traduzione di tali schemi. XML, però, non consente di superare la cosiddetta interoperabilità semantica, ovvero la capacità di più sistemi di scambiarsi dati e informazioni in modo che il loro significato sia accettato e interpretato automaticamente. Per rendere quest'ultimo passo effettivo si ha la necessità di utilizzare altre tecnologie come descritto in seguito.

2.3.3.3 Resource Description Framework (RDF)

Il Resource Description Framework (RDF) è uno strumento proposto da W3C da essere usato come metodo generale per la modellazione concettuale dell'informazione che è implementata nelle risorse web, usando diverse notazioni sintattiche e formati di serializzazione. Questa tecnologia permette di aggiungere al contenuto machine-readable un insieme di asserzioni ovvero un insieme di regole per definire informazioni descrittive sui dati, più precisamente sugli elementi costitutivi di un documento web. Per ragioni pratiche la definizione formulata del W3C propone XML come metalinguaggio per la rappresentazione del modello RDF, cioè propone RDF come linguaggio basato su XML. Attraverso RDF è possibile definire qualsiasi oggetto come una risorsa RDF. Una risorsa può essere una pagina web, un documento XML, un'immagine, un libro o una persona, questo significa che qualsiasi cosa può essere descritto tramite RDF. Ogni risorsa è univocamente identificabile attraverso

un URI garantendo che i concetti non siano solo parole in un documento, ma che siano legati ad una definizione univoca reperibile in rete. La specifica di RDF è costituita da due componenti: *RDF Model and Syntax* e *RDF Schema*. Con la prima componente si descrive il modello dei dati (*Data Model*) e la sintassi XML. *RDF Schema* invece permette di definire il significato e le caratteristiche delle proprietà e delle relazioni che esistono tra le risorse descritte nel data model RDF. Il modello dei dati RDF è basato su tre concetti fondamentali:

- *Risorsa*: tutto ciò che viene descritto. Come già detto ogni risorsa è identificata da un URI
- *Proprietà*: un aspetto specifico, una caratteristica, un attributo, o una relazione utilizzata per descrivere una risorsa. Una proprietà è una coppia attributo-valore che si vuole associare ad una risorsa: l'attributo è anch'esso identificato da un URI, il valore può essere un'altra risorsa o un tipo di dato primitivo come intero, stringa etc..
- *Asserzione*: l'associazione di una proprietà ad una risorsa. Anche chiamata *statement*, l'asserzione, che è l'unità informativa elementare in RDF, è una tripla composta da un soggetto, predicato e oggetto. Essa afferma che una determinata risorsa (il soggetto) ha un certo valore (l'oggetto) per una certa proprietà (il predicato).

Un insieme di tali triple è chiamato grafo RDF. Esso può essere illustrato da un diagramma composto da nodi e archi direzionati, nel quale ogni tripla è rappresentata da un collegamento nodo-arco-nodo. La seguente figura illustra il concetto base, rappresentato la risorsa con un ellisse, la proprietà da un arco orientato, che parte dalla risorsa e punta all'oggetto, e l'oggetto da un rettangolo:

Può essere fatto un esempio in cui l'informazione che si vuole descrivere è la seguente: *Castelli Matteo è uno studente del Politecnico di Milano*. Inoltre sapendo che Matteo Castelli è proprietario della una pagina web il cui URI è `http://nome_di_un_dominio/MatteoCastelli.html` e che l'URI Homepage

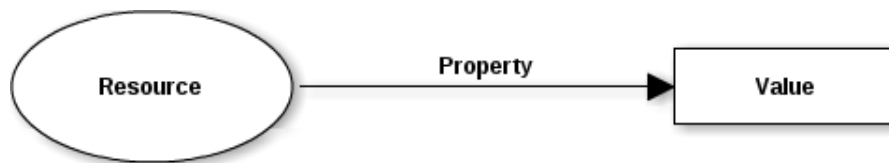


Figura 2.2: Statement RDF

del Politecnico di Milano sia <http://www.polimi.it/> lo statement che descrive l'esempio è quindi:

<http://www.polimi.it/>

Has studente

http://nome_di_un_dominio/MatteoCastelli.html.

L'oggetto di uno statement, come visto, può essere a sua volta una risorsa, consentendo in questo modo di descrivere in maniera più approfondita il valore della proprietà. Graficamente lo statement espresso avrà questa forma:

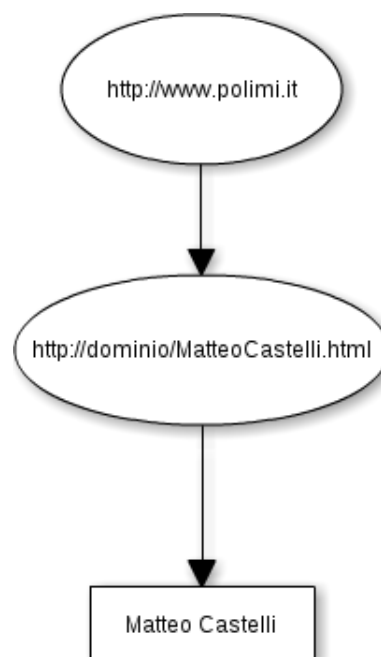


Figura 2.3: Statement RDF successivo

Il data model RDF offre un mezzo astratto e concettuale per descrivere una risorsa e per facilitare la definizione e lo scambio dei metadati, occorre

però una sintassi per la serializzazione di RDF, permettendo così di salvare i modelli RDF in forma persistente, trasferirli da un'applicazione all'altra e consentire il loro recupero dal Web. La forma canonica di serializzazione prevede l'utilizzo di una sintassi XML. L'esempio di statement precedente può essere espresso, tramite la sintassi XML, nella maniera seguente:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:a="http://nome_di_un_dominio/schema_studente/">
  <rdf:Description about="http://www.polimi.it">
    <a:studente
      rdf:resource="http://nome_di_un_dominio/MatteoCastelli.html"/>
  </rdf:Description>
  <rdf:Description
    about="http://nome_di_un_dominio/MatteoCastelli.html">
    <a:name>Matteo Castelli</a:name>
  </rdf:Description> </rdf:RDF>
```

Quando si utilizza XML, la prima riga di un file RDF è `<?xml version="1.0"?>`. Tale dichiarazione indica che il contenuto del file è scritto in un linguaggio realizzato in XML. L'attributo `xmlns:rdf` del tag `<rdf:RDF>` è la definizione di un namespace XML. Ogniqualvolta nel documento compare un concetto con prefisso `rdf:`, il sistema utilizzerà la definizione corrispondente a quel nome, contenuta all'URI specificato nella definizione del namespace. Oltre al namespace `rdf:` si possono specificare le URI alle quali sono definiti i vari concetti che intendiamo utilizzare, un namespace per ogni occorrenza di `xmlns`. In RDF però non esistono livelli di astrazione: risorse e relazioni sono organizzate in un grafo piatto. In altri termini non è possibile definire tipi (o classi) di risorse con loro proprietà specifiche. E' stata aggiunta di conseguenza una seconda componente, *RDF Schema* (RDFS), che ricorda un sistema di tipi dei linguaggi di programmazione object-oriented (come ad esempio il linguaggio Java). Una risorsa può quindi essere definita come una istanza di una classe (o di più classi) e le classi possono essere organizzate in modo gerarchico. *RDF Schema* utilizza il modello RDF stesso per definire un sistema di tipi RDF, fornendo un insieme di risorse e proprietà predefinite

che possono essere utilizzate per definire un'insieme di tassonomie. L'insieme delle risorse e delle proprietà di base è chiamato vocabolario dell' RDF Schema. I costrutti di RDFS sono le classi RDFS e le proprietà costruite sopra il vocabolario RDF.

Classi

- *rdfs:Resource* è la classe di ogni cosa. Ogni risorsa RDF è istanza di questa classe
- *rdfs:Class* definisce una risorsa come una classe. Corrisponde al concetto di classe nella programmazione object-oriented
- *rdfs:Literal* sottoclasse di *rdfs:Resource* rappresenta un letterale, ovvero una stringa di testo
- *rdfs:Property* rappresenta le proprietà. Essa è sottoclasse di *rdfs:Resource*
- *rdfs:subClassOf* specifica la relazione di ereditarietà fra classi

Proprietà Fin'ora abbiamo visto la definizione di classi in termini di altre classi. A differenza di quanto accade in molti linguaggi, proprietà e classi hanno definizioni separate e, in particolare, le proprietà sono definite in termini sia delle classi che costituiscono il loro range che delle classi a cui tali proprietà si applicano. Inoltre le proprietà sono istanze della classe *rdfs:Property*.

- *rdfs:domain* dichiara la classe del soggetto in una tripla il cui secondo componente è il predicato. Permette di imporre vincoli di dominio
- *rdfs:range* dichiara la classe o il tipo di dato dell'oggetto in una tripla il cui secondo componente è il predicato. Permette di imporre vincoli di intervallo
- *rdfs:subPropertyOf* specializza una proprietà, indicando che è una sotto proprietà di altre

Con RDF, tuttavia, il limitato potere espressivo e la semantica ancora non abbastanza definita non rende possibile effettuare vere e proprie inferenze sulla conoscenza. Sarebbe auspicabile arricchire questo linguaggio o definirne altri che forniscono qualche capacità in più. A questo scopo il W3C ha sviluppato il linguaggio OWL (Ontologi Web Language).

2.4 Applicazioni di dominio

Si è visto come il Web Semantico, attraverso una definita architettura, serva a strutturare semanticamente la conoscenza del Web. Questo processo è utile per facilitare il compito dei motori di ricerca semantici. Per l'utente comune la ricerca semantica è un passo significativo in avanti: certo non è perfetta, poiché le sfumature del linguaggio sono infinite, ma consente comunque di semplificare la gestione della conoscenza non strutturata. Questo tipo di conoscenza non è immediatamente e facilmente accessibile dai motori di ricerca attuali: è infatti difficile cercare rapidamente all'interno l'informazione desiderata, o meglio avere accesso con semplicità ai concetti chiave. La ricerca semantica può colmare questa lacuna e potrebbe riguardare svariati temi, finora solo parzialmente soddisfatti dagli attuali strumenti informatici, come ad esempio la gestione della conoscenza aziendale, ovvero la capacità non solo di trattare le diverse anagrafiche (di prodotto, clienti, fornitori, dipendenti) e di classificare i documenti tecnici o amministrativi (analisi di mercato, specifiche tecniche, norme, procedure), ma di arrivare anche a gestire i contenuti di questi documenti permettendo, ad esempio, il reperimento delle informazioni in funzione delle specifiche esigenze del richiedente, integrando quanto reso disponibile da fonti diverse.

2.4.1 Motori di ricerca semantici

I domini di applicazione per il Web Semantico dunque sono ampi e vasti, quello che c'è in comune e che si vuole presentare è l'infrastruttura del funzionamento tra il Web Semantico e i motori di ricerca: per realizzare un motore di ricerca semantico occorre mappare il Web su delle ontologie, ovvero

documenti scritti in OWL, linguaggio precedentemente descritto, e sviluppare un ragionatore automatico che usufruisca di tali ontologie per poter esaminare il Web grazie ad un'interazione semantica. Questo processo può essere schematizzato tramite la seguente figura:

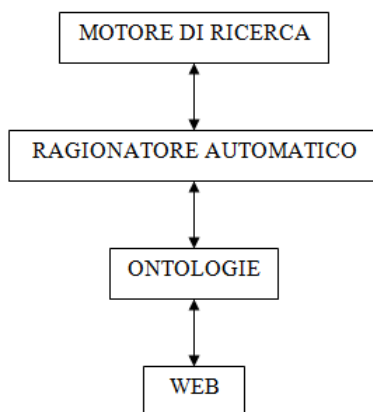


Figura 2.4: Funzionamento di un motore di ricerca semantico

In particolare un motore di ricerca semantico è in grado di estrapolare e interpretare il significato logico di una frase tenendo anche conto del contesto in cui la parola è collocata. Esso deve permettere che la ricerca semantica si avvicini al meccanismo di apprendimento umano: l'utente sviluppa una mappa che gli consente di estrarre il significato contenuto nell'informazione e così cercherà di fare un motore di ricerca semantico, analizzando l'informazione in modo simile all'essere umano, interpretando il significato logico contenuto nella frasi del testo e cercando di capire il significato dal contesto in questione. Per immaginare al meglio questo funzionamento è utile riportare un frammento di un'intervista² a Frank van Harmelen³ in cui risponde alla domanda su come si possa immaginare il Web semantico e la sua interazione con i motori di ricerca semantici:

“Abbiamo sviluppato diversi linguaggi che possono essere elaborati dai computer e che possono descrivere al computer qual è il contenuto di una particolare pagina Web. Per esempio, voi potreste affermare in uno di questi

²<http://lgxserve.ciseca.uniba.it/lei/ai/networks/03-2/vanharmelen-it.pdf>

³Vrije Universiteit Amsterdam, Faculty of Sciences, Department of Artificial Intelligence

linguaggi che vi è qualcosa che viene chiamato la ‘Vrije Univesiteit di Amsterdam’, che esiste qualcuno chiamato ‘Frank van Harmelen’, e che tra questi due oggetti sussiste una relazione specifica, e cioè ‘è un dipendente della’. Potreste anche definire il concetto di ‘edificio’, il fatto che un edificio specifico ‘è parte della’ Vrije Universiteit, e che Frank van Harmelen ‘lavora in’ quell’edificio. Naturalmente tutte queste relazioni devono essere definite: le relazioni tra me e l’edificio (‘lavora in’) è fundamentalmente diversa dalla relazione tra l’università e quell’edificio (‘parte di’). Quando descrivete tutte queste cose in uno dei linguaggi che abbiamo costruito, il computer capirà che cosa state cercando semplicemente perché gliel’avrete spiegato prima. In questo modo il computer potrebbe già darvi un supporto molto più efficace nella vostra ricerca di una specifica informazione.”

2.4.2 Seco

I motori di ricerca semantici, come si è visto, assumono un ruolo fondamentale nella gestione della conoscenza semantica. Un progetto importante, la cui rilevanza giustifica la citazione, per cui al professore Stefano Ceri⁴ è stato assegnato un IDEAS Advanced Grant, si chiama *Search Computing (SeCo)*, finanziato dall’ European Research Council (ERC). Search Computing è stato definita⁵ come *la nuova scienza multidisciplinare che fornirà le astrazioni, i metodi e gli strumenti necessari per rispondere a domande simili alle seguenti:*

- *Who are the strongest European competitors on software ideas?*
- *Who is the best doctor to cure insomnia in a nearby hospital?*
- *Where can I attend an interesting conference in my field close to a sunny beach?*

Service Computing evolve verso *Search Computing*, un nuovo paradigma dove la classificazione è il fattore dominante. Mentre i sistemi tradizionali di ricerca

⁴Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria

⁵<http://www.search-computing.it/>

rispondono a query di uno specifico dominio, Search Computing permette di rispondere a domande per mezzo di una moltitudine di servizi di ricerca che cooperano dinamicamente. In sintesi Search Computing è uno sforzo multidisciplinare che richiede l'aggiunta di solidi principi software in collaborazione ad altre scienze come la matematica, la ricerca operativa, la psicologia, la sociologia, la rappresentazione della conoscenza e le scienze economiche e giuridiche.

2.4.3 Tecnologie semantiche

Le tecnologie semantiche nell'IT costituiscono l'insieme delle risorse, delle metodologie e degli strumenti necessari per interpretare le rappresentazioni di quello che esiste nel mondo fisico e sociale. Il Web come le grandi organizzazioni pubbliche e private, sempre più affollato da documenti, testi, immagini, video e altre forme di immagazzinamento di conoscenza ed esperienza, necessita delle tecnologie semantiche per favorire l'accesso e la condivisione del patrimonio informativo a tutti gli utenti, privati, aziende o amministrazioni pubbliche. Queste tecnologie spostano il fulcro della condivisione dati, da sistemi basati sull'informazione a sistemi basati sulla conoscenza. Si ricorda che i due concetti sono profondamente diversi: l'informazione è un insieme di dati che ha un senso, un significato; la conoscenza è un'informazione che è stata interpretata da qualcuno o, in questa nuova accezione, da qualcosa, ossia da una macchina che ha capacità di ragionamento. Ad oggi le tecnologie semantiche sono già una realtà diffusa tra le industrie private, che utilizzano questi sistemi per strutturare dati e informazioni non strutturate. I motivi principali per cui sono sviluppate tecnologie semantiche sono diversi e vanno dal supporto in decisioni complesse, al risparmio in tempi e costi ed aumento di efficienza, dall'aumento del livello di qualità del servizio al cliente, all'aumento e al miglioramento dell'efficacia. La semantica, nel contesto dei sistemi informativi, consiste dunque in una dettagliata descrizione di dati e di modelli in grado di migliorare significativamente l'automazione dei processi, l'integrazione delle informazioni e le performance complessive. Per quanto riguarda le Pubbliche Amministrazioni, il fulcro delle ricerche è sulla seman-

tica applicata all'e-Government. L'efficacia dell'e-Government è, infatti, oggi impattata da una serie di problemi di integrazione tra i sistemi e le informazioni delle Pubbliche Amministrazioni attraverso una vasta eterogeneità di processi, organizzazioni, tecnologie, corpus normativi e lingue. Tali differenze rappresentano una sfida per raggiungere l'interoperabilità e l'integrazione. Le tecnologie semantiche rappresentano potenzialmente un formidabile strumento per risolvere le differenze sopra esposte, creando un paradigma di e-Government fondato sul patrimonio informativo a disposizione delle Pubbliche Amministrazioni ed orientato al soddisfacimento delle esigenze dei cittadini. Le Amministrazioni rappresentano i maggiori produttori di informazioni e documenti, molto di più delle aziende, le quali sono, tipicamente, caratterizzate da processi più snelli e meno formali. Questa peculiarità ne spiega ulteriormente come l'egovernment sia un importante campo di applicazione per le tecnologie semantiche. Ma quali sono le difficoltà di adozione per le organizzazioni pubbliche? In primis i processi e le informazioni non sono ancora sufficientemente modellizzati, a causa della complessità e delle dimensioni delle Amministrazioni. Il potenziale del Web Semantico è quindi enorme ma è anche vero che nel recente periodo si è sviluppato un certo scetticismo sul tema, in quanto le aspettative sono molto elevate ed i risultati finora ottenuti ridotti. Si parla di Web Semantico da diversi anni e di significative ed estese implementazioni non vi sono molti esempi. Forse il principale ostacolo all'esplosione del web semantico è di natura procedurale ed organizzativa piuttosto che tecnologica. C'è bisogno di ontologie comunemente accettate, formalmente definite e diffuse presso le diverse realtà. Il che non è un obiettivo particolarmente facile in virtù delle disomogeneità tra Amministrazioni discusse precedentemente. Si potrebbe pensare che la definizione di ontologie sia facile nelle diverse realtà pubbliche, non essendo, rispetto alle aziende, in competizione tra loro, anzi debbono collaborare e scambiare informazioni nel modo più integrato possibile. Allo stesso tempo però le dimensioni, la diversificazione delle linee di comando, la lunghezza del processo decisionale possono diventare un ostacolo insormontabile. Il recepimento di raccomandazioni e standard è tipicamente un processo difficile per le Pubbliche Amministrazioni (si pensi ad esempio alla dematerializzazione

del cartaceo ed alla conservazione sostitutiva che, sebbene producano evidenti ed innegabili vantaggi economici, non sono ancora diffusi come dovrebbero). Pertanto si dovrebbero studiare meccanismi che prevedano delle facilitazioni e degli aiuti economici alle Amministrazioni virtuose. Anche le tecnologie del web semantico non sono del tutto consolidate, pertanto bisognerà aspettare le evoluzioni dei prossimi anni per capire se l'applicazione semantica in ambito pubblico sarà un successo piuttosto che un'occasione mancata.

2.5 Cuciniamo

Per poter iniziare a sviluppare il progetto Easyfood è stato fin da subito necessario dover implementare un software che consentisse l'inserimento di una ricetta in testo libero e la rappresentasse internamente secondo la sua struttura, individuando e selezionando gli ingredienti e le relative quantità e misure. In rete⁶ è stata trovata la presentazione di un software per una facile e versatile gestione di un ricettario, progetto per l'esame di Linguaggi e Modelli Computazionali dell'Università di Bologna, il cui autore, Pierpaolo Tommasi, è stato contattato con la richiesta di poter avere il codice sorgente, non caricato on-line sul sito web trovato. La forte disponibilità dell'autore ha così permesso uno sviluppo iniziale più rapido del progetto primario Easyfood.

2.5.1 Descrizione e obiettivi

Il software Cuciniamo può essere eseguito su vari sistemi operativi (Windows, Linux e Mac) e supporta l'inserimento di ricette scritte in precedenza su file anche con editor di testo esterni. Lo scopo principale del software risulta poter ordinare una ricetta, catalogarla e renderla disponibile in varie visualizzazioni, per aiutare l'utente nella sua preparazione. Per la realizzazione l'autore ha creato un linguaggio abbastanza semplice ed intuitivo per la scrittura di ricette che:

- Non vincoli l'utente a ricordarsi costrutti particolari

⁶<http://enricodenti.disi.unibo.it/Ling/2010-2011/BestProjects/Tommasi/index.html>

- Utilizzi i soli caratteri stampati sulla tastiera per la grammatica
- Ponga vincoli leggeri sull'ordine con cui possano essere passate le informazioni e sia case insensitive

Il compilatore realizzato per tale linguaggio prende quindi in input una stringa di caratteri (inserita direttamente nella finestra del programma o caricata da un file di testo), effettua l'analisi sintattica e semantica producendo, nel caso vadano a buon fine le operazioni descritte, una rappresentazione adeguata. Sono inoltre previste delle segnalazioni all'utente nel caso si verificano degli errori.

2.5.2 Modellazione ricetta

Il primo passo è stato quello di riuscire a rappresentare e modellare una ricetta dal punto di vista del programmatore. In particolare una ricetta ha:

- Un nome
- Informazioni aggiuntive predefinite (come la categoria d'appartenenza, la difficoltà, il tempo di svolgimento..)
- Ricette, o meglio, sottoricette opzionali
- Degli ingredienti e una descrizione su come procedere alla preparazione

Nel dettaglio l'autore ha scelto di rappresentare:

- Il nome della ricetta
- La categoria d'appartenenza
- Il tempo di preparazione
- Le difficoltà
- L'autore
- Gli strumenti utilizzati

- Le festività d'appartenenza
- Un promemoria
- Ingredienti e quantità
- Il procedimento necessario
- Link alle immagine d'aiuto

Un esempio di ricetta che può essere compilata dal software può essere il seguente:

LINGUINE AL PESTO WELLNESS

Salva: linguinepestowellness

Autore: CucinaWellness

Categoria: Primo Tempo: 20 minuti

Prepara: pesto wellness

Salva: pestowellness

Tempo: 2 minuti

Strumenti: frullatore

Ingredienti: 40g di basilico, 15g di pinoli, 10g di parmigiano, 10g di aglio, 20g di olio extra vergine, 1 pizzico di sale, 4 cubetti di ghiaccio

Preparazione: mettere nel frullatore il basilico, i pinoli, l'aglio, il parmigiano e l'olio con i cubetti di ghiaccio. Frullare per qualche secondo

—

Ingredienti: 280g di linguine, 160g di fagiolini, 200g di patate

Preparazione: Preparare una dadolata di patate, tagliare i fagiolini a tocchetti e metterli a cuocere insieme alla pasta. Scolare la pasta al dente con le patate e i fagiolini e condire in una terrina fuori dal fuoco con il pesto. Decorare con qualche foglia di basilico.

2.5.3 Grammatica e linguaggio

La grammatica è strutturata in quattro sezioni:

- Nome della ricetta

- Informazioni opzionali (inserite in ordine qualsiasi)
- Eventuali richiami ad altre ricette indispensabili
- Ingredienti e preparazione

In una forma più corretta si riporta la scrittura estesa della grammatica in questione:

$$\begin{aligned} \text{Ricetta} ::= & \text{NomeEsteso} \langle \text{SEPARATORE} \rangle (\text{Opzioni} \langle \text{SEPARATORE} \rangle)^* \\ & (\text{Sottoricetta} \langle \text{SEPARATORE} \rangle)^* \\ & \langle \text{_INGREDIENTI} \rangle \text{ListaIngredienti} \langle \text{SEPARATORE} \rangle \\ & \langle \text{_PREPARAZIONE} \rangle \text{Preparazione} (\langle \text{FINENOTA} \rangle \mid \langle \text{EOF} \rangle) \\ & (\langle \text{FINE} \rangle \mid \langle \text{EOF} \rangle) \end{aligned}$$

1. Il nome della ricetta (prima sezione):

$$\text{NomeEsteso} ::= (\langle \text{STRING} \rangle)^+$$

2. Le informazioni aggiuntive (seconda sezione):

$$\begin{aligned} \text{Opzioni} ::= & (\langle \text{_SALVA} \rangle \text{NomeSalvataggio} \\ & \mid \langle \text{_AUTORE} \rangle \text{Autore} \mid \langle \text{_CATEGORIA} \rangle \text{TipoCategoria} \\ & \mid \langle \text{_DIFFICOLTA} \rangle \text{ValoreDifcolta} \mid \langle \text{_TEMPO} \rangle \text{Tempo} \\ & \mid \langle \text{_PRIVATA} \rangle \text{Privata} \mid \langle \text{_STRUMENTI} \rangle \text{ListaStrumenti} \\ & \mid \langle \text{_FESTIVITA} \rangle \text{Festivita} \mid \langle \text{_PROMEMORIA} \rangle \text{Promemoria}) \end{aligned}$$

3. L'inserimento di sottoricette (terza sezione) è possibile in due modi, o richiamando lo scopo oppure richiamando un esterno (o uno stream da database, in una futura implementazione) che verrà immediatamente analizzato e inserito.

4. Ingredienti e preparazione (quarta e ultima sezione):

$$\text{ListaIngredienti} ::= \text{Ingrediente} (\text{"}, " \text{Ingrediente})^*$$

$$\begin{aligned} \text{Ingrediente} ::= & \text{QuantitaIngrediente} \text{TipoIngrediente} \mid \text{TipoIngrediente} (\\ & \text{QuantitaIngrediente})? \end{aligned}$$

La grammatica, secondo la classificazione di Chomsky, è di tipo 2 (context-free), cioè:

$$A \rightarrow \alpha$$

con $\alpha \in (VT \cup VN)^*$, $A \in VN$

Il linguaggio generato è di tipo 2 in quanto è presente self-embedding, ovvero presenza di simboli non terminali autoinclusivi del tipo:

$$A \rightarrow \beta A \delta$$

dove $A \in VN$, $\beta, \delta \in V^+$

Inoltre il linguaggio è di tipo LL(1) perché nessuna produzione genera stringa vuota e gli insiemi degli starter symbols corrispondenti alla parte destra delle produzioni alternative di uno stesso metasimbolo sono tra loro disgiunti. Si ha quindi un PDA deterministico con analisi top-down. Per quanto riguarda i controlli semantici la grammatica è molto restrittiva sui caratteri utilizzabili, vengono infatti eseguiti controlli semantici sui tipi resi inutili. Sgravando l'utente dal ricordare un ordine prestabilito con cui immettere le informazioni aggiuntive non è più possibile (attraverso la sola grammatica) impedire che un'informazione venga inserita più volte. Semanticamente si controlla quindi che un'opzione venga inserita al più una volta (generando un warning in caso contrario). Infine nel caso in cui un'opzione sfora dai valori consentiti (e.g. difficoltà: 11) viene generato un warning e l'opzione ignorata. Ogni ricetta viene salvata sottoforma di albero, si esplicano in seguito le regole con cui viene generato:

- Ogni ricetta, e quindi anche una sottoricetta, è un nodo dell'albero
- Ogni informazione aggiuntiva inserita è un'opzione in più fornita al nodo
- Ogni ingrediente è una foglia della ricetta a cui appartiene

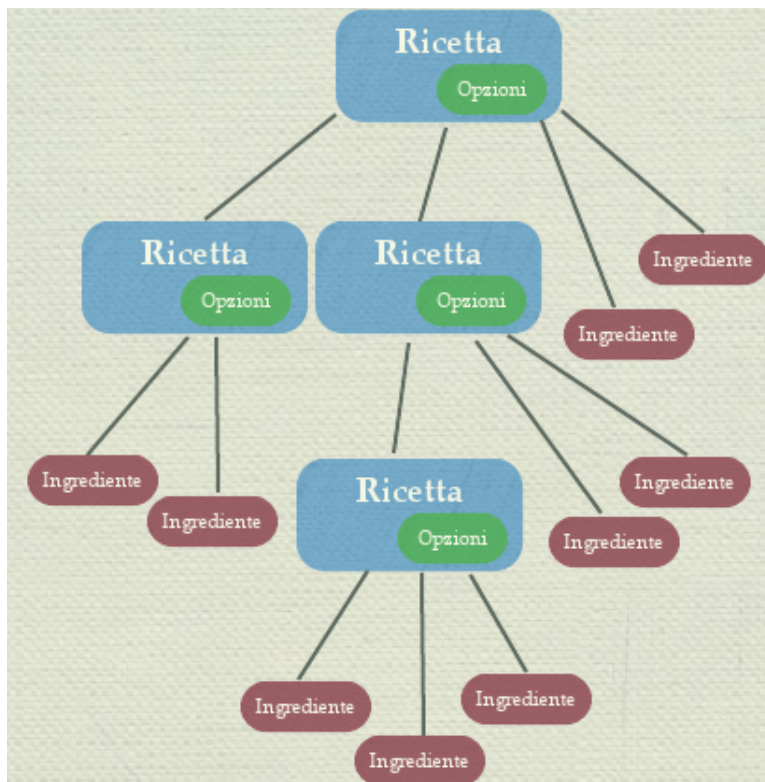


Figura 2.5: Albero ricetta

2.5.4 Implementazione

L'autore ha scelto di suddividere il progetto in sei package (di cui tre auto-generati) per un facile mantenimento:(inserire immagini package). Per quanto riguarda i dettagli implementativi è rilevante riportare la classe **MyRicetta**, perno dell'architettura creata inizialmente da **MyDepthVisitor** (estensione di **DepthVisitor** creata da JTB). Le sue peculiarità sono:

- Derivazione da **TreeModel** per presentarsi come albero
- Derivazione da **MyGraphModel** per essere disegnata sottoforma di semplice grafo (ogni nodo fornisce informazioni su profondità massima sottostante e ampiezza, ovvero numero pesato dei figli)
- Caching dei risultati che coinvolgono chiamate ricorsive

Inoltre al suo interno la classe **MyRicetta** contiene:

- Un'istanza di **MyOpzioni** per tenere traccia delle informazioni aggiuntive inserite (che la sgrava dal compito di lavoro su queste)
- Un vettore (**ArrayList** dimensionato in maniera esatta) di **MyIngrediente**.

Il software sviluppato permette principalmente di calcolare la totalità degli ingredienti necessari con le relative quantità e disegnare un grafico che accompagni l'utente nella reale preparazione. Come si è spiegato sono presenti anche ulteriori informazioni aggiuntive e permettono di

- Salvare automaticamente la ricetta su disco
- Catalogare la ricetta in base alla categoria e/o festività d'appartenenza
- Fare una stima del tempo necessario per la preparazione
- Trovare la difficoltà di preparazione effettiva
- Memorizzare un vasto numero di informazioni utili.

Capitolo 3

Metodologia

Da quanto emerso nel capitolo precedente le tecnologie utilizzate per la realizzazione del progetto sono nuove e ancora poco esplorate, in effetti ci si aspetta che nell'immediato futuro ci siano ulteriori sviluppi rispetto a quanto descritto nello stato dell'arte. Nel panorama delle applicazioni che si occupano di alimentazione, non sono stati trovati software che utilizzano informazione semantica per realizzare una comparazione fra ricette. Tantissimi sono i siti internet che forniscono dei ricettari statici, e ancor di più le applicazioni che lo fanno sui social media. Rimane il fatto che si tratta di archivi di ricette che non vanno oltre la possibilità di commentare e condividere tali ricette. Non esiste di fatto alcun software in grado di realizzare una similarità fra ricette e tanto meno che lo faccia prendendo come punto di riferimento un'ontologia.

Il software realizzato vuole andare a colmare questo gap tecnologico e nel presente capitolo ci si propone di spiegare come tali tecnologie sono state realizzate e quali sono le funzionalità implementate nel software.

3.1 Requisiti

L'utente che utilizza il software ha a disposizione diverse funzionalità che lo aiutano nella preparazione di una ricetta. In Figura 3.1 è riportato lo Use Case Diagram dedicato appunto alla descrizione delle funzioni offerte dal

sistema. L'attore coinvolto è un'utente qualsiasi che inserisce una ricetta e può:

- visualizzare, tramite il grafico creato in seguito alla compilazione, la struttura della ricetta inserita
- visualizzare gli ingredienti necessari alla preparazione della ricetta
- avere un termine di paragone, attraverso la similarità, tra la sua ricetta e quella wellness contenuta nell'ontologia. In particolare si potranno scegliere tre tipi di similarità:

1. Similarità per nutrienti
2. Similarità semantica
3. similarità per ingredienti

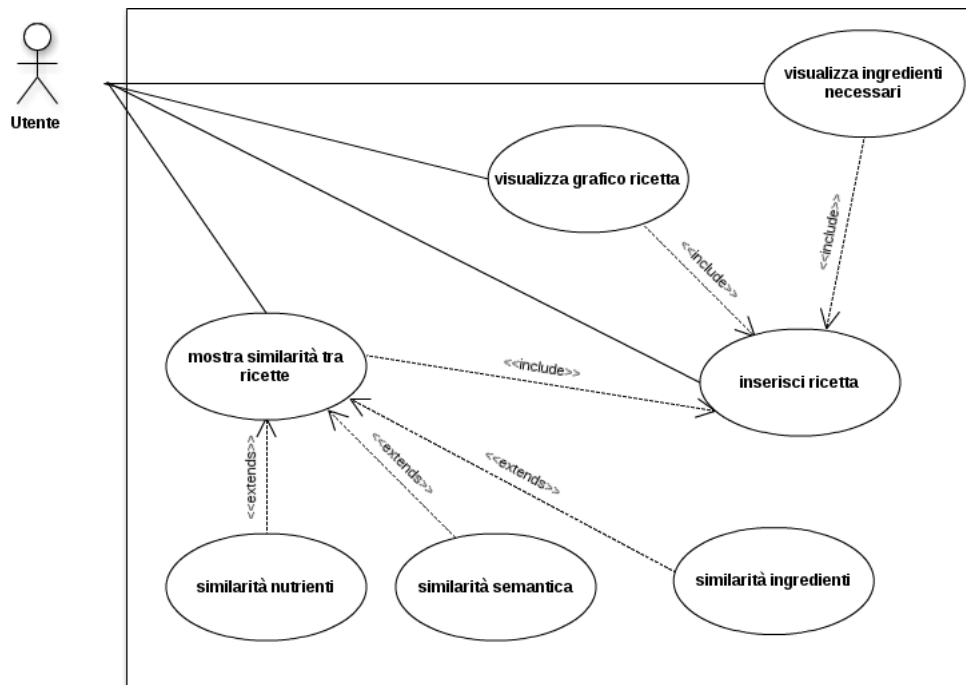


Figura 3.1: Use case diagram

3.2 Funzionalità

Riassumendo il funzionamento del software, l'utente inserisce nell'applicazione la sua ricetta e, questa, dopo essere stata elaborata, viene trasformata in un documento xml. Partendo da questo documento, viene popolato un oggetto di tipo **Ricetta** che verrà poi utilizzato per il calcolo della similarità. Questo calcolo infatti viene effettuato tra la suddetta ricetta e l'equivalente ricetta Wellness che è stata trovata all'interno dell'ontologia. Risolta la ricerca, anche in questo caso viene caricato un oggetto di tipo **Ricetta**. In altre parole, in input alla similarità, ci sono due oggetti di tipo **Ricetta**: quello relativo all'utente e quello relativo alla cucina Wellness. Definire quanto due ricette siano simili è un'operazione relativamente complessa. In effetti è necessario stabilire in base a cosa queste due ricette debbano essere simili tra loro. Per rispondere a questo interrogativo, come detto poc'anzi, sono state individuate 3 tipologie fondamentali di similarità:

- Similarità per nutrienti
- Similarità semantica
- Similarità per ingredienti

Nei prossimi paragrafi saranno approfonditi gli algoritmi di funzionamento di ognuno di essi, per ora si vuole fornire una spiegazione generale del loro utilizzo e del loro significato. La più semplice fra queste, la similarità semantica, dice quanto i due insiemi di ingredienti (Wellness e Utente) sono simili fra loro. In casi patologici questo indice però può dare un'informazione poco indicativa. Ad esempio potremmo avere due ricette che effettivamente hanno gli stessi ingredienti, ma le cui quantità (a parità di numero di persone per cui è pensata la ricetta) sono molto diverse fra loro. E' evidente che serve un'informazione di maggior dettaglio e a questa esigenza risponde la similarità quantitativa. Naturalmente anche questa cerca gli ingredienti che corrispondono fra le due ricette, la sua peculiarità consiste nell'andare a prendere in considerazione le quantità e, in particolare, le differenze di quantità, degli ingredienti presenti in entrambe le ricette. La terza ed ultima tipologia di similarità vuole rispondere all'esigenza di conoscere quanto

due ricette sono simili fra loro in base ai macronutrienti e all'apporto calorico.

Prima di procedere con la descrizione di dettaglio dei vari algoritmi è bene sottolineare alcune operazioni preliminari. La prima di queste operazioni consiste nell'appiattare l'albero della ricetta Wellness. Come detto, tale ricetta può essere composta da ingredienti che in realtà sono sottoricette, e da ciò deriva proprio l'utilizzo di un albero per rappresentare tali informazioni. Quello che si genera quindi è un albero a due livelli: il livello 0 composto dalla radice e il livello 1 composto da tutti gli ingredienti (anche di eventuali sotto ricette) della ricetta.

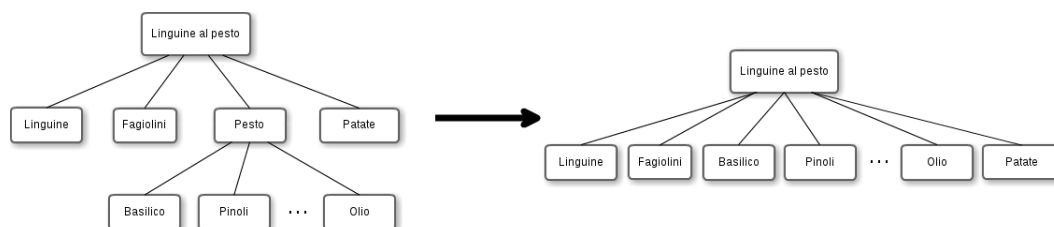


Figura 3.2: Appiattamento di una ricetta

L'algoritmo di appiattamento di una ricetta viene utilizzato esclusivamente per la ricetta Wellness. In realtà nulla vieta che tale algoritmo possa essere applicato anche alla ricetta inserita dall'utente. Questo però non viene fatto perchè, salvo alcuni piccoli accorgimenti, all'utente viene chiesto di inserire un testo libero. Generalmente questo testo sarà il cut&paste di un testo trovato in rete e si ritiene improbabile che tale testo sia ben formattato al punto da specificare eventuali sottoricette. In altre parole si ritiene che l'albero della ricetta inserita dall'utente sia già composto da due soli livelli.

Lo step successivo, e questo interessa sia la ricetta Wellness che la ricetta inserita dall'utente, è quello della formattazione degli ingredienti. Come descritto in precedenza, le informazioni fondamentali di un ingrediente sono: il nome, la quantità e la misura. Se il nome è un'informazione che non può mancare, la quantità e la misura hanno vari modi per essere espressi. La

condizione ideale prevederebbe che al nome dell'ingrediente venisse affiancata una quantità espressa in grammi. Alcuni esempi potrebbero essere:

- 80g di noci
- 120g di parmigiano
- 250g di fusilli

Purtroppo non sempre si verifica tale condizione, alcuni casi che hanno bisogno della formattazione sono i seguenti:

- 1 limone
- 2 cucchiaini di parmigiano
- un mazzetto di rucola
- una tazza di latte

Lo scopo della formattazione è prendere la dicitura di quantità e misura di un ingrediente e tentare di stimarne la quantità in grammi. Nello specifico, la parte di software che assolve a questo compito può contare su un database che, riconosciuto l'ingrediente, fornisce la quantità per unità di misura. Ad esempio per l'ingrediente "parmigiano", la quantità è "2" e la misura è "cucchiaini". Il software cerca "parmigiano" nel suo database e se la misura è espressa in "cucchiaini" trova che mediamente un cucchiaino di parmigiano equivale a 20 grammi. Essendo la quantità "2" è possibile sostenere che 2 cucchiaini di parmigiano equivalgono all'incirca a 40 grammi di parmigiano.

Nello specifico avviene tale formattazione: "2 cucchiaini di parmigiano" diventano "40g di parmigiano".

Il terzo ed ultimo passaggio consiste nell'assegnare un peso agli ingredienti della ricetta Wellness. L'utilità di tale operazione sarà approfondita nella similarità quantitativa. Per ora si vuole solamente spiegare come viene effettuata. Dato un albero, che a questo punto ha un solo livello di n ingredienti, il peso di ognuno di essi corrisponde a $\frac{1}{n}$. Conclusa questa semplice operazione è possibile procedere con i calcoli di similarità vera e propria.

3.2.1 Similarità semantica

Volendo schematizzarne il funzionamento, l'algoritmo di similarità semantica si compone di tre fasi principali. C'è una prima fase in cui si aumenta la similarità per quegli ingredienti che si trovano in entrambe le ricette, una seconda in cui vi è un ulteriore aumento per quelle coppie di ingredienti (utente-Wellness) che sono diversi, ma fanno parte della stessa categoria alimentare, e una terza parte in cui si abbassa la similarità per quegli ingredienti per cui non vi è alcuna corrispondenza.

Si procede nella spiegazione dettagliata dell'algoritmo presentando, passo per passo, dei frammenti di pseudo-codice. Alla conclusione di tale descrizione è possibile trovare l'algoritmo per intero.

Procediamo nella spiegazione osservando il primo frammento di codice:

Algoritmo 3.1 Similarità semantica prima parte

```
sim_semantica = 0;

//Per ogni ingrediente della ricetta utente
for( Ingrediente i : ricUtente.listaIngredienti() ){

    //Controllo che sia presente nella ricetta Wellness
    Ingrediente c = cercaInIngredientiWellness(i);

    //Se l'ingrediente utente esiste anche nella Wellness
    if(c != null){
        sim_semantica = sim_semantica + c.getPeso();
    }
}
```

Ovviamente si parte dall'inizializzazione della variabile di similarità. Quindi si vanno a scorrere tutti gli ingredienti della ricetta utente e per ognuno di essi, attraverso la funzione *cercaInIngredientiWellness(i)*, si vuole verificare che tale ingrediente sia o meno presente anche nella ricetta Wellness. Tale funzione restituisce l'oggetto **Ingrediente** trovato. Nel caso non venga trovato alcun ingrediente corrispondente, la funzione ritorna **null**. Se la ricerca è stata condotta con successo, la similarità sarà aumentata del peso dell'ingrediente Wellness.

A questo punto si procede con la seconda fase:

Algoritmo 3.2 Similarità semantica seconda parte

```
//Per ogni coppia di ingredienti per cui non è stata
//trovata l'eguaglianza nello step precedente
for ( Ingredienti utente "u" non usati ){
    for( Ingredienti wellness "w" non usati ){

        if ( hannoPadreUguale(u,w) ){
            sim_semantica = sim_semantica + (0.5*getPesoMedio())
        }

    }
}
```

Nei cicli *for* annidati si vanno a scorrere due liste di ingredienti. La prima lista consiste nell'insieme di ingredienti utente non trovati nella Wellness e la seconda nell'insieme di ingredienti Wellness che invece non sono nella ricetta utente. Per ogni coppia di questi ingredienti si controlla all'interno dell'ontologia se appartengono alla stessa classe. Tale controllo viene effettuato dalla funzione *hannoPadreUguale(u, w)* che restituisce in output un valore booleano. Se il risultato dà esito positivo la similarità viene incrementata della metà del peso medio degli ingredienti Wellness.

Infine la terza fase:

Algoritmo 3.3 Similarità semantica terza parte

```
//Penalizzo per ingredienti per cui
//non vi è alcuna corrispondenza
for ( Ingredienti utente "u" nessuna corrispondenza ){
    sim_semantica = sim_semantica - (0.25*getPesoMedio())
}

for ( Ingredienti wellness "w" nessuna corrispondenza ){
    sim_semantica = sim_semantica - (0.25*getPesoMedio())
}

return sim_semantica;
```

Ancora si lavora, questa volta separatamente, su due liste di ingredienti. Nella prima ci sono gli ingredienti utente che non hanno nulla a che fare con un qualsiasi ingrediente Wellness, nella seconda ci sono gli ingredienti

Wellness che non hanno nulla a che fare con nessuno degli ingredienti utente. Per ognuno di questi ingredienti viene detratto dalla similarità il 25% del peso medio.

Qui di seguito è disponibile lo pseudo-codice dell'intero algoritmo:

Algoritmo 3.4 Similarità semantica

```
float similaritaSemantica(Ricetta ricWellness, Ricetta ricUtente){
sim_semantica = 0;

//Per ogni ingrediente della ricetta utente
for( Ingrediente i : ricUtente.listaIngredienti() ){

    //Controllo che sia presente nella ricetta Wellness
    Ingrediente c = cercaInIngredientiWellness(i);

    //Se l'ingrediente utente esiste anche nella Wellness
    if(c != null){
        sim_semantica = sim_semantica + c.getPeso();
    }
}

//Per ogni coppia di ingredienti per cui non è stata
//trovata l'eguaglianza nello step precedente
for ( Ingredienti utente "u" non usati ){
    for( Ingredienti wellness "w" non usati ){

        if ( hannoPadreUguale(u,w) ){
            sim_semantica = sim_semantica + (0.5*getPesoMedio())
        }

    }
}

//Penalizzo per ingredienti per cui
//non vi è alcuna corrispondenza
for ( Ingredienti utente "u" nessuna corrispondenza ){
    sim_semantica = sim_semantica - (0.25*getPesoMedio())
}

for ( Ingredienti wellness "w" nessuna corrispondenza ){
    sim_semantica = sim_semantica - (0.25*getPesoMedio())
}

return sim_semantica;
}
```

Per una migliore comprensione dell'algoritmo si procede la spiegazione

con un esempio pratico. Supponiamo di avere una ricetta Wellness composta dai seguenti ingredienti e i relativi pesi indicati con α :

wellness		utente
Ingrediente	Peso	Ingrediente
melanzane	$\frac{1}{9}$	melanzane
pasta fillo	$\frac{1}{9}$	lasagne
pomodori	$\frac{1}{9}$	pomodori
pepe rimbas	$\frac{1}{9}$	pepe
aglio	$\frac{1}{9}$	aglio
olio	$\frac{1}{9}$	olio
sale	$\frac{1}{9}$	sale
basilico	$\frac{1}{9}$	origano
limone	$\frac{1}{9}$	
		scamorza
		farina
		rucola
		peperoncino
		grana

Tabella 3.1: Ricetta utente e Wellness a confronto

Osservando le due ricette si può notare che hanno i seguenti ingredienti in comune:

- Melanzane
- Pomodori
- Aglio
- Sale
- Pepe
- Olio

All'interno di una variabile viene salvata la sommatoria dei pesi: pertanto

$$sim_semantica = \frac{6}{9}$$

. Questo calcolo conclude la prima parte dell'algoritmo.

A questo punto consideriamo l'insieme degli ingredienti della ricetta Wellness che non sono presenti nella ricetta utente:

- Pasta fillo
- Limone
- Basilico

e l'insieme degli ingredienti utente che non sono presenti nella ricetta Wellness:

- Lasagne
- Scamorza
- Origano
- Farina
- Rucola
- Peperoncino
- Grana

Quello che si vuole fare nella seconda fase è confrontare gli ingredienti dei due insiemi e aumentare la similarità nel caso, all'interno dell'ontologia, questi abbiano stesso padre (in grassetto nella Tabella 3.1). Quindi si va a considerare positivamente nel calcolo della similarità il fatto che pur non utilizzando gli stessi ingredienti, comunque vengono usati ingredienti appartenenti alla stessa categoria alimentare. Naturalmente, a differenza di prima, viene assegnata solo una parte del peso: nel nostro caso si è scelto il 50%.

Nel caso in questione Pasta fillo e Lasagne sono due tipi di ingredienti appartenenti alla stessa categoria "Pasta", così come Basilico e Origano appartengono alla categoria "Erbe". Trovate queste corrispondenze viene incrementato il valore della similarità:

$$sim_semantica = sim_semantica + 2 * (\frac{1}{9} * 0,5)$$

. Alla fine del secondo step la similarità sarà uguale a $\frac{7}{9}$. Confrontando i due insiemi di ingredienti, è possibile notare che anche l'ingrediente Rucola appartiene alla categoria "Erbe". Naturalmente in questo caso non viene incrementata la variabile perchè l'ingrediente Basilico della ricetta Wellness è già stato utilizzato come corrispondente dell'Origano: in altre parole, gli ingredienti per cui viene trovata una corrispondenza vengono utilizzati una sola volta.

La terza ed ultima parte dell'algoritmo assegna una penalizzazione per ogni ingrediente delle due ricette per i quali non vi è alcuna corrispondenza. Per la ricetta Wellness è rimasto un solo ingrediente:

- Limone

mentre per la ricetta utente sono rimasti:

- scamorza
- farina
- rucola
- peperoncino
- grana

Si è scelto di assegnare una penalizzazione del 25% del peso. Pertanto

$$sim_semantica = sim_semantica - 6 * (\frac{1}{9} * 0,25)$$

Al termine dell'algoritmo il risultato di similarità semantica fra le due ricette sarà del 61,11% .

3.2.2 Similarità per ingredienti

La similarità per ingredienti cerca di fornire un'informazione relativa all'uguaglianza delle due ricette tenendo conto delle quantità dei vari ingredienti. A livello generale l'algoritmo si comporta in modo analogo alla similarità

semantica: c'è una prima fase in cui si considerano gli ingredienti uguali, una seconda in cui si cercano gli ingredienti con padre uguale e una terza in cui si abbassa la similarità a causa degli ingredienti per cui non vi è alcuna corrispondenza.

Come nel caso precedente, si procede nella spiegazione dettagliata dell'algoritmo.

Il seguente è il primo frammento di codice:

Algoritmo 3.5 Similarità per ingredienti prima parte

```
sim_ingr = 0;
diff = 0;

//Per ogni ingrediente della ricetta wellness
for( Ingrediente i : ricWellness.listaIngredienti() ){
    //Vado a ricalcolare il peso
    if( i.getNome() in principali) i.setPesoPrincipale();
    else i.setPesoNonPrincipale();
    //Calcolo il prodotto del peso per la quantità
    i.setProdottoQuPe();
}
```

La prima operazione consiste nell'inizializzare alcune variabili che saranno utilizzate nelle fasi successive.

All'interno di questo tipo di similarità viene data all'utente la possibilità di inserire fino ad un massimo di 3 ingredienti principali. Come visto nelle operazioni preliminari, a tutti gli ingredienti viene assegnato lo stesso peso ma, nel caso alcuni di questi vengano indicati come principali, si applica una modifica al fine di aumentarne l'importanza nel calcolo della similarità. Si è scelto questo approccio perchè la quantità non è sufficiente per indicare l'importanza di un ingrediente all'interno della ricetta. Se prendessimo ad esempio le linguine al pesto così come indicate dalla cucina wellness, il basilico, che è da considerarsi fondamentale, pesa solamente 40 dei 768 grammi totali. Entrando nel dettaglio dell'algoritmo, vengono presi tutti gli ingredienti della ricetta wellness e, per ognuno di questi, si controlla che sia o meno presente all'interno della lista di ingredienti indicati come principali. Nel caso tale controllo dia esito positivo, il peso dell'ingrediente viene raddoppiato. Agli ingredienti che invece non sono stati indicati come principali viene assegnato

un nuovo peso che chiaramente sarà inferiore al precedente e tale per cui la somma di tutti i pesi dia comunque 1. Effettuato il calcolo del peso, questo viene moltiplicato alla quantità.

Si procede con la normalizzazione dei valori appena calcolati:

Algoritmo 3.6 Similarità per ingredienti seconda parte

```
//Per ogni ingrediente della ricetta wellness
for( Ingrediente i : ricWellness.listaIngredienti() ){
    //Normalizzo il prodotto del peso per la quantità
    valore = i.getProdottoQuPe()/sommaProdottoQuPeW();
    i.setProdottoQuPeNorm(valore);
}
```

La funzione *sommaProdottoQuPeW()* implementa la sommatoria dei prodotti dei pesi per le quantità:

$$\sum_i q_{w_i} * \alpha$$

Prendendo, attraverso la *i.getProdottoQuPe()*, il prodotto del peso per la quantità del singolo ingrediente, e dividendolo per la sommatoria appena indicata, si ottiene il valore normalizzato:

$$q_w_alpha_norm_i = \frac{\alpha_i * q_{w_i}}{\sum_i \alpha_i * q_{w_i}}$$

A questo punto comincia il paragone fra le due ricette:

Algoritmo 3.7 Similarità per ingredienti terza parte

```
//Per ogni ingrediente della ricetta utente
for( Ingrediente u : ricUtente.listaIngredienti() ){

    //Controllo che sia presente nella ricetta Wellness
    Ingrediente w = cercaInIngredientiWellness(i);

    //Se l'ingrediente utente esiste anche nella Wellness
    if(w != null){
        u.setPeso( w.getPeso() );
        u.setProdottoQuPe();
        valore = u.getProdottoQuPe()/sommaProdottoQuPeW();
        u.setProdottoQuPeNorm(valore);
        valore = ASS(w.getProdottoQuPeNorm()-u.getProdottoQuPeNorm());
        diff = diff + valore;
    }
    else{
        u.setPeso( getPesoMedio() );
        u.setProdottoQuPe();
        valore = u.getProdottoQuPe()/sommaProdottoQuPeW();
        u.setProdottoQuPeNorm(valore);
    }
}

//Assegno ad una variabile la media delle diff
mae = diff / getNumUguali();
```

Per ognuno degli ingredienti della ricetta utente, si va a controllare con la funzione *cercaInIngredientiWellness(i)*, che tale ingrediente sia presente anche nella ricetta wellness. La funzione restituisce *null* se la ricerca dà esito negativo, restituisce l'oggetto **Ingrediente** trovato in caso di esito positivo. Successivamente viene calcolato il peso e il prodotto del peso per la quantità dell'ingrediente utente. Il peso è quello della ricetta wellness se esiste un ingrediente wellness corrispondente, altrimenti viene preso il peso medio degli ingredienti Wellness. Effettuata la normalizzazione, in caso di ingrediente trovato, viene aggiunta ad una variabile *diff* la differenza in valore assoluto fra il valore normalizzato appena calcolato dell'ingrediente utente e il valore normalizzato dell'ingrediente wellness. Concluso il ciclo *for*, in una variabile viene salvato il valore medio di queste differenze. In altre parole, la variabile *MAE* rappresenta, per gli ingredienti uguali fra le due ricette, quanto questi si discostano a livello di quantità. In realtà il conto viene fatto sui valori

normalizzati del peso per la quantità, ma essendo ingredienti uguali, viene utilizzato lo stesso peso. Se per esempio avessimo due ingredienti uguali con quantità uguali, il relativo valore di *diff* sarebbe 0 e quindi quel tale ingrediente non contribuirebbe affatto allo scostamento.

Lo step successivo consiste nel migliorare la similarità in funzione delle coppie di ingredienti che hanno padre uguale:

Algoritmo 3.8 Similarità per ingredienti quarta parte

```
//Per ogni coppia di ingredienti per cui non è stata
//trovata l'eguaglianza nello step precedente
for ( Ingredienti utente "u" non usati ){
    for( Ingredienti wellness "w" non usati ){

        if ( hannoPadreUguale(u,w) ){
            u.setPeso( w.getPeso() );
            u.setProdottoQuPe();
            valore = u.getProdottoQuPe()/sommaProdottoQuPeW();
            u.setProdottoQuPeNorm(valore);
            valore = ASS(w.getProdottoQuPeNorm()-u.getProdottoQuPeNorm());
            mae = mae - valore;
        }
    }
}
```

Nell'Algoritmo 3.7 sono stati cercati gli ingredienti uguali: quando il criterio di uguaglianza veniva soddisfatto, la coppia di ingredienti Wellness - utente, veniva ricordata settando a *true* un apposito campo del modello dei dati. Essendo questo campo inizializzato a *false*, è ora possibile dedurre gli ingredienti per cui non vale l'uguaglianza. Confrontando tutte le possibili coppie di ingredienti delle due ricette per cui il suddetto campo vale *false*, si cercano quelli che all'interno dell'ontologia appartengono alla stessa classe. Per tali coppie, si prende l'ingrediente utente, si setta il peso del relativo ingrediente Wellness, lo si normalizza, si calcola il valore assoluto della differenza con il valore normalizzato Wellness e lo si sottrae alla variabile *MAE*. Tale variabile contiene in sostanza quanto mediamente gli ingredienti uguali si scostano fra loro. Con questi calcoli si cerca di abbassare tale valore in virtù del fatto che alcune coppie di ingredienti precedentemente considerati diversi non sono proprio diversi ma hanno un certo grado di somiglianza.

L'ultima fase è quella in cui si va a penalizzare per gli ingredienti per i quali non vi è alcuna relazione:

Algoritmo 3.9 Similarità per ingredienti quinta parte

```
//Penalizzo per ingredienti per cui
//non vi è alcuna corrispondenza
for ( Ingredienti utente "u" nessuna corrispondenza ){
    mae = mae + u.getProdottoQuPeNorm();
}
for ( Ingredienti wellness "w" nessuna corrispondenza ){
    mae = mae + w.getProdottoQuPeNorm();
}

sim_ingr = 1 - mae;
return sim_ingr;
```

Così come accaduto nell'Algoritmo 3.7, anche nell'Algoritmo 3.8 sono stati segnati con una variabile booleana quegli ingredienti che sono già stati utilizzati nel computo dell'algoritmo di similarità. I rimanenti, per i quali l'apposito campo è rimasto settato a *false*, sono automaticamente quelli per cui non vi è alcuna corrispondenza. Per tali ingredienti viene sommato al *MAE* il valore normalizzato del peso per il prodotto. In precedenza si prendeva il valore assoluto della differenza tra ingrediente Wellness e ingrediente utente, ma mancando in questo caso uno dei due è come se si facesse la differenza con 0.

Listato dell'intero algoritmo:

Algoritmo 3.10 Similarità per ingredienti

```
float similaritaIngredienti(Ricetta ricWellness, Ricetta ricUtente, Ingrediente principali){
    sim_ingr = 0;
    diff = 0;

    //Per ogni ingrediente della ricetta wellness
    for( Ingrediente i : ricWellness.listaIngredienti() ){
        //Vado a ricalcolare il peso
        if( i.getNome() in principali) i.setPesoPrincipale();
        else i.setPesoNonPrincipale();
        //Calcolo il prodotto del peso per la quantità
        i.setProdottoQuPe();
    }

    //Per ogni ingrediente della ricetta wellness
    for( Ingrediente i : ricWellness.listaIngredienti() ){
        //Normalizzo il prodotto del peso per la quantità
        valore = i.getProdottoQuPe()/sommaProdottoQuPeW();
        i.setProdottoQuPeNorm(valore);
    }

    //Per ogni ingrediente della ricetta utente
    for( Ingrediente u : ricUtente.listaIngredienti() ){
        //Controllo che sia presente nella ricetta Wellness
        Ingrediente w = cercaInIngredientiWellness(i);
        //Se l'ingrediente utente esiste anche nella Wellness
        if(w != null){
            u.setPeso( w.getPeso() );
            u.setProdottoQuPe();
            valore = u.getProdottoQuPe()/sommaProdottoQuPeW();
            u.setProdottoQuPeNorm(valore);
            valore = ASS(w.getProdottoQuPeNorm()-u.getProdottoQuPeNorm());
            diff = diff + valore;
        }
        else{
            u.setPeso( getPesoMedio() );
            u.setProdottoQuPe();
            valore = u.getProdottoQuPe()/sommaProdottoQuPeW();
            u.setProdottoQuPeNorm(valore);
        }
    }

    //Assegno ad una variabile la media delle diff
    mae = diff / getNumUguali();

    //Per ogni coppia di ingredienti per cui non è stata
    //trovata l'eguaglianza nello step precedente
    for ( Ingredienti utente "u" non usati ){
        for( Ingredienti wellness "w" non usati ){

            if ( hannoPadreUguale(u,w) ){
                u.setPeso( w.getPeso() );
                u.setProdottoQuPe();
                valore = u.getProdottoQuPe()/sommaProdottoQuPeW();
                u.setProdottoQuPeNorm(valore);
                valore = ASS(w.getProdottoQuPeNorm()-u.getProdottoQuPeNorm());
                mae = mae - valore;
            }
        }
    }

    //Penalizzo per ingredienti per cui
    //non vi è alcuna corrispondenza
    for ( Ingredienti utente "u" nessuna corrispondenza ){
        mae = mae + u.getProdottoQuPeNorm(); 67
    }
    for ( Ingredienti wellness "w" nessuna corrispondenza ){
        mae = mae + w.getProdottoQuPeNorm();
    }

    sim_ingr = 1 - mae;
    return sim_ingr;
}
```

Procediamo nella spiegazione con un esempio:

wellness			utente	
Ingrediente	q_w	α	Ingrediente	q_u
linguine	280g	0,1	linguine	600g
basilico	40g	0,1	basilico	50g
patate	200g	0,1	patate	320g
parmigiano	10g	0,1	parmigiano	90g
aglio	15g	0,1	aglio	30g
olio	20g	0,1	olio	3g
sale	3g	0,1	sale	3g
fagiolini	160g	0,1	fagiolini	30g
pinoli	15g	0,1	pinoli	12g
ghiaccio	25g	0,1		
			noci	3g
			pecorino	30g

Tabella 3.2: Confronto fra le due ricette

Supponiamo che l'utente indichi come ingredienti principali "Basilico" e "Pinoli". Essendo la ricetta composta da 10 ingredienti, questi partono da un peso iniziale di 0,1. La modifica è molto semplice, infatti il loro peso viene moltiplicato per un certo fattore.

$$peso_nuovo = peso_vecchio * 2$$

A questo punto il peso dei due ingredienti citati diventa 0,2, mentre, quello di tutti gli altri, affinché la sommatoria dei pesi dia sempre 1, diventa 0,075.

Ecco un riepilogo dei nuovi pesi:

wellness			utente	
Ingrediente	q_w	α	Ingrediente	q_u
linguine	280g	0,075	linguine	600g
basilico	40g	0,2	basilico	50g
patate	200g	0,075	patate	320g
parmigiano	10g	0,075	parmigiano	90g
aglio	15g	0,075	aglio	30g
olio	20g	0,075	olio	3g
sale	3g	0,075	sale	3g
fagiolini	160g	0,075	fagiolini	30g
pinoli	15g	0,2	pinoli	12g
ghiaccio	25g	0,075		
			noci	3g
			pecorino	30g

Tabella 3.3: Pesi modificati

Lo step successivo prevede di moltiplicare il peso per la quantità:

wellness				utente	
Ingrediente	q_w	α	$q_w * \alpha$	Ingrediente	q_u
linguine	280g	0,075	21	linguine	600g
basilico	40g	0,2	8	basilico	50g
patate	200g	0,075	15	patate	320g
parmigiano	10g	0,075	0,75	parmigiano	90g
aglio	15g	0,075	1,125	aglio	30g
olio	20g	0,075	1,5	olio	3g
sale	3g	0,075	0,225	sale	3g
fagiolini	160g	0,075	12	fagiolini	30g
pinoli	15g	0,2	3	pinoli	12g
ghiaccio	25g	0,075	1,875		
				noci	3g
				pecorino	30g

Tabella 3.4: Prodotto dei pesi per le quantità

quindi, se ne calcola la sommatoria:

$$\sum_i \alpha_i * q_{w_i} = 64,475$$

Quello che interessa è capire il peso relativo che ha un ingrediente sull'intera ricetta; pertanto si effettua una normalizzazione nel seguente modo:

$$q_w_ \alpha_ norm_i = \frac{\alpha_i * q_{w_i}}{\sum_i \alpha_i * q_{w_i}}$$

wellness					utente	
Ingrediente	q_w	α	$q_w * \alpha$	$q_w_ \alpha_ norm$	Ingrediente	q_u
linguine	280g	0,075	21	0,326	linguine	600g
basilico	40g	0,2	8	0,124	basilico	50g
patate	200g	0,075	15	0,233	patate	320g
parmigiano	10g	0,075	0,75	0,012	parmigiano	90g
aglio	15g	0,075	1,125	0,017	aglio	30g
olio	20g	0,075	1,5	0,023	olio	3g
sale	3g	0,075	0,225	0,003	sale	3g
fagiolini	160g	0,075	12	0,186	fagiolini	30g
pinoli	15g	0,2	3	0,047	pinoli	12g
ghiaccio	25g	0,075	1,875	0,029		
					noci	3g
					pecorino	30g

Tabella 3.5: Valori Wellness normalizzati

A questo punto parte il paragone tra le due ricette e per gli ingredienti uguali viene calcolato il valore normalizzato:

$$q_u_ \alpha_ norm_i = \frac{\alpha_i * q_{u_i}}{\sum_i \alpha_i * q_{w_i}}$$

Considerando ancora gli ingredienti uguali viene calcolata la differenza in valore assoluto dei valori normalizzati, quindi viene fatta la sommatoria delle differenze: 0,83.

L'errore medio assoluto si calcola dividendo questa sommatoria per il numero n di ingredienti trovati:

$$MAE = \frac{\sum_i diff_i}{n}$$

wellness					utente				diff	
Ingrediente	q_w	α	$q_w * \alpha$	$q_w_ \alpha_ norm$	Ingrediente	q_u	$q_u * \alpha$	$q_u_ \alpha_ norm$		
linguine	280g	0,075	21	0,326	linguine	600g	45	0,698	0,372	
basilico	40g	0,2	8	0,124	basilico	50g	10	0,155	0,031	
patate	200g	0,075	15	0,233	patate	320g	24	0,372	0,140	
parmigiano	10g	0,075	0,75	0,012	parmigiano	90g	6,75	0,105	0,093	
aglio	15g	0,075	1,125	0,017	aglio	30g	2,25	0,035	0,017	
olio	20g	0,075	1,5	0,023	olio	3g	3	0,003	0,020	
sale	3g	0,075	0,225	0,003	sale	3g	0,225	0,003	0	
fagiolini	160g	0,075	12	0,186	fagiolini	30g	2,25	0,035	0,151	
pinoli	15g	0,2	3	0,047	pinoli	12g	2,4	0,037	0,009	
ghiaccio	25g	0,075	1,875	0,029						
					noci	3g	0,3	0,005		
					pecorino	30g	3	0,047		
									$\sum_i diff_i$	0,834
									MAE	0,093

Tabella 3.6: Valori wellness normalizzati

L'algoritmo prevede di modificare il MAE in base agli ingredienti che hanno padre uguale, ma in questa coppia di ricette non ci sono ingredienti che soddisfano tale criterio, pertanto si passa alla fase successiva. Il MAE calcolato in Tabella 3.6 viene ulteriormente incrementato del valore normalizzato degli ingredienti rimanenti:

Ingrediente	wellness				Ingrediente	utente			diff
	q_w	α	$q_w * \alpha$	$q_w_ \alpha_ norm$		q_u	$q_u * \alpha$	$q_u_ \alpha_ norm$	
linguine	280g	0,075	21	0,326	linguine	600g	45	0,698	
basilico	40g	0,2	8	0,124	basilico	50g	10	0,155	
patate	200g	0,075	15	0,233	patate	320g	24	0,372	
parmigiano	10g	0,075	0,75	0,012	parmigiano	90g	6,75	0,105	
aglio	15g	0,075	1,125	0,017	aglio	30g	2,25	0,035	
olio	20g	0,075	1,5	0,023	olio	3g	3	0,003	
sale	3g	0,075	0,225	0,003	sale	3g	0,225	0,003	
fagiolini	160g	0,075	12	0,186	fagiolini	30g	2,25	0,035	
pinoli	15g	0,2	3	0,047	pinoli	12g	2,4	0,037	
ghiaccio	25g	0,075	1,875	0,029					0,029
					noci	3g	0,3	0,005	0,005
					pecorino	30g	3	0,047	0,047
								$\sum_i diff_i$	0,080
								<i>mae*</i>	0,173

Tabella 3.7: Valori wellness normalizzati

In conclusione, per questa coppie di ricette, la similarità è:

$$1 - MAE = 82,71\%$$

3.2.3 Similarità per nutrienti

La similarità per elementi nutritivi è l'ultima tipologia di similarità ed ha il compito di fornire all'utente un'indicazione di quanto due ricette siano simili in base ai macronutrienti che le caratterizzano. Come al solito abbiamo una ricetta inserita dall'utente e una equivalente ricetta Wellness ottenuta dall'ontologia. Ancor prima di procedere con la spiegazione dell'algoritmo è bene precisare che in questo caso l'ontologia non solo fornisce le quantità dei vari ingredienti Wellness, ma è anche in grado di dare le quantità di lipidi, proteine e carboidrati per 100g di prodotto. Ne consegue che il primo step è proprio quello di calcolare le quantità assolute dei tre macronutrienti. Si comincia con la ricetta wellness:

Algoritmo 3.11 Similarità per nutrienti prima parte

```
sim_nut = 0;

//Per ogni ingrediente della ricetta wellness
for( Ingrediente w : ricWellness.listaIngredienti() ){
    //Calcolo dei valori dei tre macronutrienti
    w.setProteine();
    w.setLipidi();
    w.setCarboidrati;
}
```

Ai fini del calcolo della similarità interessano i valori aggregati dei macronutrienti:

Algoritmo 3.12 Similarità per nutrienti seconda parte

```
//Calcolo i valori di macronutrienti dell'intera ricetta
carboidratiTotW=0;
proteineTotW=0;
lipidiTotW=0;
for( Ingrediente w : ricWellness.listaIngredienti() ){
    carboidratiTotW=carboidratiTotW+w.getCarboidrati();
    lipidiTotW=lipidiTotW+w.getLipidi();
    proteineTotW = proteineTotW+w.getProteine();
}
```

Il passaggio successivo è quello di trasformare queste informazioni in termini di Kcalorie. Dal sito dell'INRAN, l'Istituto Nazionale di Ricerca per gli Alimenti e la Nutrizione, si ricava che:

- 1 grammo di proteine equivale a 3,1 Kcal
- 1 grammo di carboidrati equivale a 3,8 Kcal
- 1 grammo di lipidi equivale a 9,3 Kcal

Lo scopo di tale conversione è quello di capire la quantità totale di Kcalorie della ricetta, quindi calcolare in che percentuale ogni macronutriente contribuisce a tale valore. Ad esempio per le proteine:

$$percentProteineW = \frac{proteineTotW Kcal}{KcalTotW}$$

dove $KcalTotW$ rappresenta il totale delle calorie per la ricetta wellness, e $proteineTotW Kcal$ rappresenta le calorie dovute alle proteine totali presenti nella ricetta.

Algoritmo 3.13 Similarità per nutrienti terza parte

```
//Calcolo le Kcal dell'intera ricetta
carboidratiTotWKcal=carboidratiTotW*3.8;
proteineTotWKcal=proteineTotW*3.1;
lipidiTotWKcal=lipidiTotW*9.3;
KcalTotW=carboidratiTotWKcal+proteineTotWKcal+lipidiTotWKcal;

//Calcolo in che percentuale i tre macronutrienti concorrono nel
//costituire le calorie totali della ricetta
percentProteineW=proteineTotWKcal/KcalTotW;
percentCarboidratiW=carboidratiTotWKcal/KcalTotW;
percentLipidiW=lipidiTotWKcal/KcalTotW;
```

Alla conclusione dell'Algoritmo 3.13 il software ha calcolato le tre percentuali precedentemente descritte per la ricetta Wellness.

E' necessario calcolare gli stessi valori per la ricetta utente. L'unica differenza è data dal fatto che i tre valori percentuali vengono ancora calcolati sul totale delle calorie della ricetta Wellness. Tale compito viene assolto dall'Algoritmo 3.14:

Algoritmo 3.14 Similarità per nutrienti quarta parte

```
//A questo punto faccio gli stessi conti sulla ricetta utente
for( Ingrediente u : ricUtente.listaIngredienti() ){
    u.setProteine();
    u.setLipidi();
    u.setCarboidrati();
}
//Calcolo i valori di macronutrienti dell'intera ricetta
carboidratiTotU=0;
proteineTotU=0;
lipidiTotU=0;
for( Ingrediente u : ricUtente.listaIngredienti() ){
    carboidratiTotU=carboidratiTotU+u.getCarboidrati();
    lipidiTotU=lipidiTotU+u.getLipidi();
    proteineTotU = proteineTotU+u.getProteine();
}

//Calcolo le Kcal dell'intera ricetta
carboidratiTotUKcal=carboidratiTotU*3.8;
proteineTotUKcal=proteineTotU*3.1;
lipidiTotUKcal=lipidiTotU*9.3;

KcalTotU=carboidratiTotUKcal+proteineTotUKcal+lipidiTotUKcal;
percentProteineU=proteineTotUKcal/KcalTotW;
percentCarboidratiU=carboidratiTotUKcal/KcalTotW;
percentLipidiU=lipidiTotUKcal/KcalTotW;
```

A questo punto ci sono i tre valori percentuali per ognuna delle due ricette. Viene fatto un paragone tra le percentuali dello stesso macronutriente e in

una variabile viene salvato il valore assoluto della differenza. Ad esempio per le proteine:

$$assProteine = ASS(percentProteineW - percentProteineU)$$

dove *percentProteineU* e *percentProteineW* rappresentano la percentuale di calorie dovute alle proteine, rispettivamente della ricetta utente e della ricetta Wellness, calcolate entrambe sul totale delle calorie della Wellness.

Viene fatta la media fra questi tre valori:

$$mediaAss = \frac{assCarboidrati + assProteine + assLipidi}{3}$$

E viene restituito come valore di similarità:

$$1 - mediaAss$$

Algoritmo 3.15 Similarità per nutrienti quinta parte

```
//Mi calcolo il valore assoluto della differenza
assCarboidrati=ASS(percentCarboidratiW - percentCarboidratiU);
assProteine=ASS(percentProteineW - percentProteineU);
assLipidi=ASS(percentLipidiW - percentLipidiU);

sim_nut=1-((assCarboidrati+assProteine+assLipidi)/3);
return sim_nut;
```

Qui di seguito è disponibile l'algoritmo per intero:

Algoritmo 3.16 Similarità per nutrienti

```
float similaritaNutrienti(Ricetta ricWellness, Ricetta ricUtente){
    sim_nut = 0;

    //Per ogni ingrediente della ricetta wellness
    for( Ingrediente w : ricWellness.listaIngredienti() ){
        //Calcolo dei valori dei tre macronutrienti
        w.setProteine();
        w.setLipidi();
        w.setCarboidrati;
    }

    //Calcolo i valori di macronutrienti dell'intera ricetta
    carboidratiTotW=0;
    proteineTotW=0;
    lipidiTotW=0;
    for( Ingrediente w : ricWellness.listaIngredienti() ){
        carboidratiTotW=carboidratiTotW+w.getCarboidrati();
        lipidiTotW=lipidiTotW+w.getLipidi();
        proteineTotW = proteineTotW+w.getProteine();
    }

    //Calcolo le Kcal dell'intera ricetta
    carboidratiTotWKcal=carboidratiTotW*3.8;
    proteineTotWKcal=proteineTotW*3.1;
    lipidiTotWKcal=lipidiTotW*9.3;
    KcalTotW=carboidratiTotWKcal+proteineTotWKcal+lipidiTotWKcal;

    //Calcolo in che percentuale i tre macronutrienti concorrono nel
    //costituire le calorie totali della ricetta
    percentProteineW=proteineTotWKcal/KcalTotW;
    percentCarboidratiW=carboidratiTotWKcal/KcalTotW;
    percentLipidiW=lipidiTotWKcal/KcalTotW;

    //A questo punto faccio gli stessi conti sulla ricetta utente
    for( Ingrediente u : ricUtente.listaIngredienti() ){
        u.setProteine();
        u.setLipidi();
        u.setCarboidrati;
    }
    //Calcolo i valori di macronutrienti dell'intera ricetta
    carboidratiTotU=0;
    proteineTotU=0;
    lipidiTotU=0;
    for( Ingrediente u : ricUtente.listaIngredienti() ){
        carboidratiTotU=carboidratiTotU+u.getCarboidrati();
        lipidiTotU=lipidiTotU+u.getLipidi();
        proteineTotU = proteineTotU+u.getProteine();
    }
    //Calcolo le Kcal dell'intera ricetta
    carboidratiTotUKcal=carboidratiTotU*3.8;
    proteineTotUKcal=proteineTotU*3.1;
    lipidiTotUKcal=lipidiTotU*9.3;
    KcalTotU=carboidratiTotUKcal+proteineTotUKcal+lipidiTotUKcal;
    percentProteineU=proteineTotUKcal/KcalTotU;
    percentCarboidratiU=carboidratiTotUKcal/KcalTotU;
    percentLipidiU=lipidiTotUKcal/KcalTotU;

    //Mi calcolo il valore assoluto della differenza
    assCarboidrati=ASS(percentCarboidratiW - percentCarboidratiU);
    assProteine=ASS(percentProteineW - percentProteineU);
    assLipidi=ASS(percentLipidiW - percentLipidiU);

    sim_nut=1-((assCarboidrati+assProteine+assLipidi)/3);
    return sim_nut;
}
```

Si procede nella spiegazione con due ricette d'esempio. La Tabella 3.8 e la Tabella 3.9 presentano le due liste di ingredienti con le relative quantità di Proteine Lipidi e Carboidrati. Il passo successivo è quello di calcolare la sommatoria di ognuno dei macronutrienti, quindi, le calorie che tale valore comporta. In conclusione è possibile ricavare le percentuali di calorie dei nutrienti sul totale delle calorie.

wellness				
Ingrediente	q_w	Proteine	Lipidi	Carboidrati
linguine	280g	33,6g	11,2g	235,2g
basilico	40g	1,2g	0g	2g
patate	200g	4g	2g	34g
parmigiano	10g	3,3g	2,8g	0g
aglio	15g	0g	0g	1,2g
olio	20g	0g	20g	0g
sale	3g	0g	0g	0g
fagiolini	160g	3,2g	0g	3,2g
pinoli	15g	4,65g	7,5g	0,6g
ghiaccio	25g	0g	0g	0g
Tot. Macronutriente		49,95g	43,5g	276,2g
Kcalorie		154,845	404,55	1049,56
% sul Tot Kcal		0,0962394846	0,2514364914	0,652324024

Tabella 3.8: Dati per la ricetta wellness

Ingrediente	utente			
	q_w	Proteine	Lipidi	Carboidrati
linguine	350g	42g	14g	294g
basilico	50g	1,5g	0g	2,5g
patate	160g	3,2g	1,6g	27,2g
parmigiano	90g	29,7g	25,2g	0g
aglio	30g	0g	0g	2,4g
olio	3g	0g	3g	0g
sale	3g	0g	0g	0g
fagiolini	200g	4g	0g	4g
pinoli	15g	4,65g	7,5g	0,6g
pecorino	30g	7,5g	0g	9,6g
Tot. Macronutriente		92,55g	51,3g	340,3g
Kcalorie		286,905	477,09	1293,14
% sul Tot Kcal		0,1783176037	0,2965216554	0,803714212

Tabella 3.9: Dati per la ricetta utente

Nella Tabella 3.10 vengono messe a confronto le percentuali delle due ricette. Si calcola la media dei valori assoluti delle differenze, quindi si giunge al valore di similarità che è pari al 90,71%.

	Proteine	Lipidi	Carboidrati
% sul Tot Kcal Wellness	0,0962394846	0,2514364914	0,652324024
% sul Tot Kcal Utente	0,1783176037	0,2965216554	0,803714212
Assoluto della differenza	0,082078119	0,045085164	0,151390188
Media		0,092851157	
Similarità		0,907148843	

Tabella 3.10: Confronto fra le due ricette

3.3 Ontologia necessaria

Sottolineando ciò che è stato già detto, il Web assomiglia ad un enorme continente di cui non esiste la mappa. Le uniche possibilità che abbiamo di recuperare risorse utili sono basate su ricerche sintattiche, che affidano la loro efficacia alla corrispondenza tra parole chiave. Con le ontologie invece, è possibile realizzare mappe semantiche del Web interpretabili sia dall'uomo

che dalla macchina. L'approccio è quello bottom-up: non un'enorme mappa imposta dall'alto ma tante piccole mappe interconnesse, realizzate da singoli utenti, a cui interessa formalizzare un dominio o, più semplicemente, aggiungere semantica al proprio sito Web. Un'ontologia quindi definisce un vocabolario utile per condividere informazioni su una tematica specifica e contiene al suo interno definizioni di concetti comprensibili dalle macchine e relazioni tra essi. Ma perchè scegliere di utilizzare un'ontologia per sviluppare un'applicazione semantica per la *Cucina Wellness*? Proviamo a spiegarne alcune:

- Ontologie esistenti in cui fossero raggruppati, classificati e suddivisi gli alimenti per definizione e caratteristiche non sono state trovate. E' difficile anche reperire nel Web una base di conoscenza coerente e soprattutto completa che descriva e riporta, secondo una precisa classificazione alimentare, tutti i possibili alimenti di cui si nutre l'essere umano e tutte le informazioni nutrizionali corrispondenti. Per la costruzione dell'ontologia Wellness sono stati usati i dati, riguardanti le proprietà nutrizionali degli alimenti, forniti dall'Istituto Nazionale di Ricerca per gli Alimenti e la Nutrizione¹ nella sezione banche dati. Per quanto riguarda invece la classificazione sono state seguite le linee guida trovate on-line o libri di testo. [Costantini, 2006]
- Per come è stata pensata, l'ontologia *Cucina Wellness*, non avrebbe contenuto solamente dati e informazioni nutrizionali degli alimenti, ma sarebbe stata anche una base di conoscenza che fornisse ricette appartenenti a questo tipo di cucina salutare per cui si vuole sviluppare l'applicazione. Uno dei suoi scopi è quindi quello di poter unificare queste duplici funzionalità, non limitandosi ad essere un semplice ricettario.
- Permettere il riutilizzo della conoscenza acquisita: supponiamo di voler sviluppare in seguito anche un'ontologia sui vini (già presente nel Web) inserendo in essa qualità come il colore, il sapore, l'anno di im-

¹<http://www.inran.it/>

bottigliamento e via dicendo. Si può creare quindi un'altra ontologia che permette di associare un determinato pasto o ricetta al vino ottimale.

- In un'implementazione e miglioramento futuri l'ontologia può essere sfruttata al meglio per dare suggerimenti di similarità fra ricette e fra ingredienti, aggiungendo magari proprietà di sostituibilità fra ingredienti o informazioni riguardanti la cottura di questi che cambiano le proprietà nutrizionali. Dei ragionatori automatici poi potranno sfruttare queste caratteristiche descritte nell'ontologia stessa per derivare le informazioni necessarie.

Capitolo 4

Implementazione

4.1 Creazione dell'ontologia

Per la creazione dell'ontologia è stato usato il software gratuito *TopBraid Composer Free Editions*, sviluppato da TopQuadrant¹, ottimo ambiente di sviluppo per applicazioni semantiche. Come prima cosa si è creato un nuovo file *CucinaWellness.owl* che è l'ontologia che si vuole definire.

4.1.1 Le classi

L'ontologia è stata creata con lo scopo di fornire una solida base a delle implementazioni future, cercando di comprendere tutti gli aspetti che riguardano i concetti di cucina ed educazione alimentare. La prima classe, come in ogni documento .rdf o .owl, è la classe predefinita di default owl:Thing e ogni classe che verrà creata sarà figlia di questa. Si è pensato di definire diverse classi che rappresentino le entità coinvolte nell'ambito alimentare a partire dall'alimento, qualsiasi sostanza assimilata da un organismo vivente per la propria nutrizione. La classe **Alimento** è la radice di una struttura gerarchica compresa nella tre sottoclassi di **Alimento** che sono **Animale**, **Vegetale** e **Minerale**. Infatti tutti i tipi di alimento si possono includere in una delle tre classi appena definite. Di seguito una figura che rappresenta la gerarchia appena espressa:

¹<http://www.topquadrant.com/index.html>

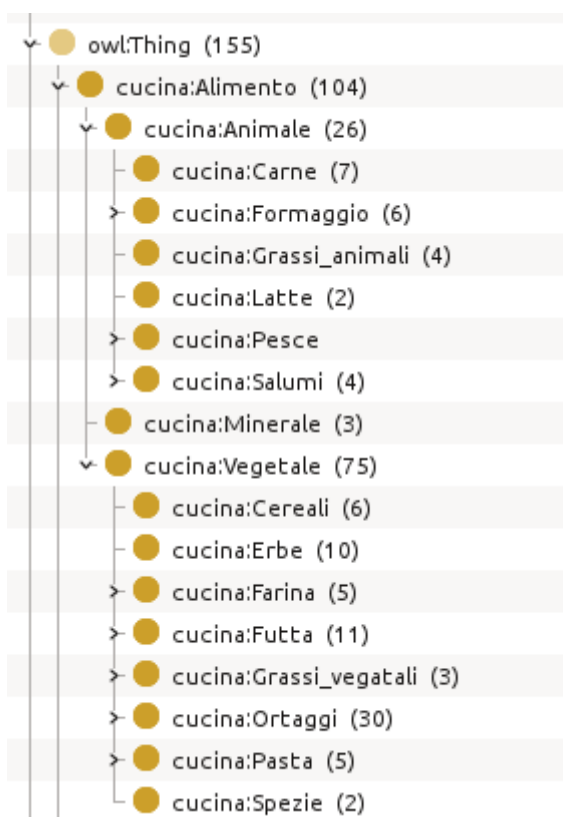


Figura 4.1: La classe Alimento

Un esempio di codice in RDF/XML per la classe Vegetale, sottoclasse di Alimento è il seguente:

```

<rdf:Description
  rdf:about="http://example.org/CucinaWellness#Vegetale">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Vegetale
  </rdfs:label>
  <rdfs:subClassOf
    rdf:resource="http://example.org/CucinaWellness#Alimento"/>
</rdf:Description>

```

Dopo aver definito i possibili tipi di alimento, si è proceduto alla classificazione dell'alimento. Tra le numerose possibili classificazioni degli alimenti, si è seguita la suddivisione in sette gruppi proposta dall' Istituto Nazionale della Nutrizione. Il significato di tale suddivisione consiste nel fatto che ogni

gruppo fornisce uno specifico apporto di sostanze nutritive per cui, se almeno un alimento di ciascuno dei sette gruppi è presente nell'alimentazione quotidiana, si è certi che vengano ingeriti i nutrienti essenziali. E' stata quindi creata una classe **Classificazione** composta da sette classi, rappresentanti i sette gruppi.

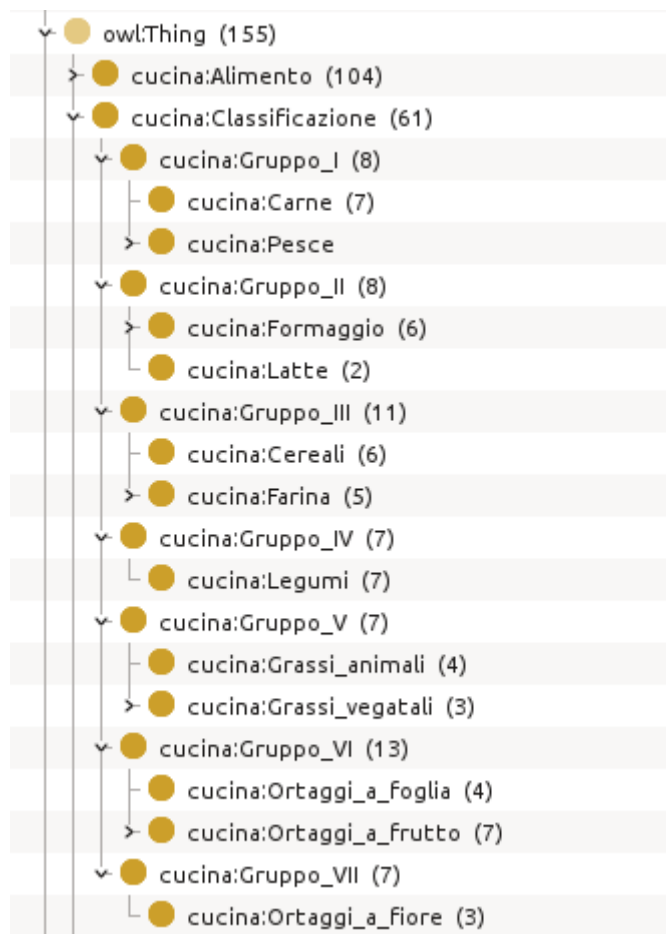


Figura 4.2: La classe Classificazione

La terza classe fondamentale di questa ontologia è la classe **Ricetta** definita dalla canonica suddivisione in **Primo**, **Secondo**, **Sugo**, **Contorno**, **Frutta** e **Dolce**. Si riporta quindi la figura che rappresenta quest'ultima classe:



Figura 4.3: La classe Ricetta

4.1.2 Le proprietà

Le proprietà aggiunte rispetto a quelle standard sono di tipo *owl:ObjectProperty* e *owl:DatatypeProperty*. Le prime servono per legare le classi, le seconde per legare le classi a tipi di dato predefiniti come *xsd:string*, *xsd:int* e *xsd:float*. Si riporta di seguito l'immagine con tutte le proprietà create:

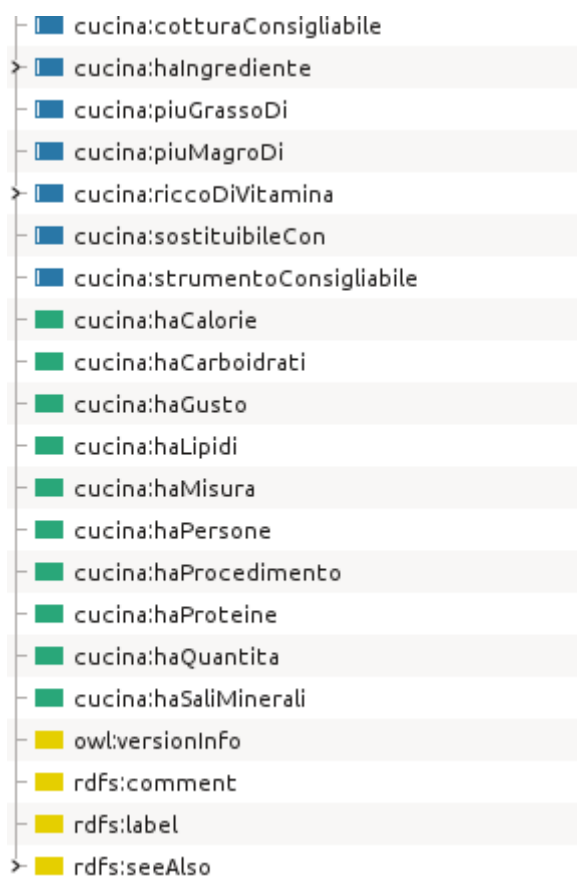


Figura 4.4: Le proprietà

E' necessario sottolineare che le proprietà utilizzate si limitano ad essere quelle base, che descrivo una ricetta e gli ingredienti che la costituiscono, come:

- *haIngrediente*:
domain: Ricetta
range: Alimento or Ricetta

- *haCalorie*:
domain: Alimento or Ricetta
range: float
- *haLipidi*
domain: Alimento or Ricetta
range: float
- *haMisura*
domain: Alimento
range: string
- *haPersone*
domain: Ricetta
range: int
- *haProcedimento*
domain: Ricetta
range: int
- *haProteine*
domain: Ricetta or Alimento
range: float
- *haQuantita*
domain: Alimento
range: String

Si riporta anche un esempio di codice in cui si mostra la subproprietà *haSugo* e la sua relazione con la proprietà *haIngrediente*, avendo sfruttato la possibilità di creare una *subproperty*:

```
<rdf:Description
  rdf:about="http://example.org/CucinaWellness#haSugo">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range
    rdf:resource="http://example.org/CucinaWellness#Sugo"/>
```

```

<rdfs:domain
  rdf:resource="http://example.org/CucinaWellness#Ricetta"/>
<rdfs:subPropertyOf
  rdf:resource="http://example.org/CucinaWellness#haIngrediente"/>
</rdfs:Description>

```

4.1.3 Le istanze

Sono poi state create le istanze, ovvero gli individui di una classe. Quando si crea un'istanza si possono completare le proprietà definite per quella classe, aggiungendo altre istanze, se si tratta di una *owl:ObjectProperty*, o inserendo gli opportuni valori, se siamo nel caso di una *owl:DatatypeProperty*. La quasi totalità delle istanze presenti nell'ontologia appartengono alla classe **Alimento** e, come descritto in precedenza, sono possono essere alimenti di origine animale, vegetale e minerale, con le successivi diverse classificazioni. Prendiamo come esempio il *Basilico* descritto dal pezzo di codice seguente:

```

<rdfs:Description
  rdf:about="http://example.org/CucinaWellness#Basilico">
  <cucina:haLipidi
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.8
  </cucina:haLipidi>
  <cucina:haVitaminaLiposolubile
    rdf:resource="http://example.org/CucinaWellness#Vitamina_K"/>
  <cucina:haMisura
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    g
  </cucina:haMisura>
  <cucina:haVitaminaLiposolubile
    rdf:resource="http://example.org/CucinaWellness#Vitamina_A"/>
  <cucina:haCarboidrati
    rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
    5.1
  </cucina:haCarboidrati>
  <rdfs:label
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Basilico
  </rdfs:label>

```

```

<cucina:haCalorie
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
  39
</cucina:haCalorie>
<cucina:haProteine
  rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
  3.1
</cucina:haProteine>
<cucina:haQuantita
  rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
  40
</cucina:haQuantita>
<rdf:type
  rdf:resource="http://example.org/CucinaWellness#Erbe"/>
<cucina:haVitaminaIdrosolubile
  rdf:resource="http://example.org/CucinaWellness#Vitamina_C"/>
</rdf:Description>

```

Come si può notare l'istanza *Basilico* possiede le proprietà principali *haLipidi*, *haMisura*, *haQuantita*, *haCarboidrati*, *haCalorie*, *haProteine* che descrivono la composizione dell'alimento in questione. Nella successiva sezione si andrà poi a definire uno dei problemi sorti attraverso l'uso del linguaggio *owl*, ovvero che se ad un ingrediente, ad esempio il *Basilico*, si imposta un determinato valore (e.g. *10 g*) per la proprietà *haQuantita*, questo valore sarà riportato per tutte le possibili ricette che hanno come ingrediente il *Basilico*, non potendo così andare a definire possibili valori multipli per la stessa istanza. Per chiarezza espositiva si riporta la descrizione grafica, attraverso il software Protegè, che permette, a differenza della versione gratuita TopBraid Composer, di visualizzare graficamente l'ontologia, della ricetta *Linguine al pesto*, istanza della classe **Ricetta**.

4.2 Problematiche

Come annunciato in precedenza, attraverso l'uso del linguaggio *owl* si sono riscontrate delle difficoltà nella modellazione della quantità, proprietà di un ingrediente, che risulta essere dipendente dalla ricetta a cui appartiene il

determinato ingrediente. Infatti, prendendo come esempio il caso dell'istanza *Basilico*, ingrediente contenuto sia nella ricetta *Linguine al pesto* che in *Millefoglie di melanzane*, le rispettive quantità cambiano, nel particolare la prima ricetta necessita di 40 grammi e la seconda di 10 grammi. Questi duplici valori di proprietà non possono essere modellizzati attraverso il potere espressivo del linguaggio *owl* rendendo inappropriato l'uso di quest'ultimo per la rappresentazione dei valori delle proprietà come quella presa in analisi.

4.2.1 Prima soluzione: reification

Non potendo perciò descrivere quanto detto, si è pensato come aggirare questo problema partendo dal concetto di *reification*: si modella la relazione-proprietà a cui si vuole associare un peso, nel nostro caso la proprietà *haQuantità*, come una classe per la quale il peso è una proprietà (*dataType property*). In generale per rappresentare una relazione pesata tra istanze della classe *A* e della classe *B*, si dovrebbe creare una terza classe *RelazionePesata* che contiene tutte le possibili istanze che descrivono i possibili valori assunti. Si devono definire poi le seguenti proprietà che legano le classi descritte:

- *fromRelazionePesata*: classe *A* → classe *RelazionePesata*
- *toRelazionePesata*: classe *RelazionePesata* → classe *B*
- *haPeso*: classe *RelazionePesata* → *xsd:float*

Al contrario di quanto si pensasse, questa soluzione non funziona, infatti in questo modo si aggiunge valore alla relazione, attribuendole un peso, che non dipende però dalla classe da cui si parte, ovvero si perde così l'informazione di dipendenza che l'ingrediente ha nei confronti della ricetta a cui appartiene. Cambiando istanza della classe *A*, nel nostro caso le ricette, le istanze delle classi *RelazionePesata* e Classe *B* rimangono le stesse, perdendo il potere espressivo necessario.

4.2.2 Seconda soluzione: molteplicità delle proprietà

Una seconda soluzione a cui si è pensato è quella di definire una proprietà, come è stato fin'ora, *haQuantita* e tante sottoproprietà *haQuantitaNomeRicetta* quante sono le ricette a cui appartiene quel determinato ingrediente, potendo così esprimere la molteplicità dei valori delle quantità in analisi. La seguente figura mostra il concetto appena descritto:

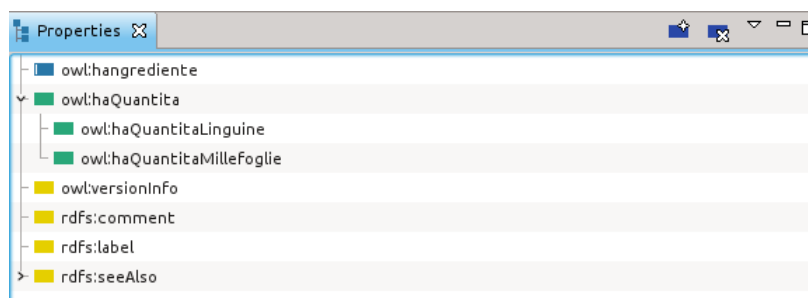


Figura 4.5: Sottoproprietà della proprietà *haQuantita*

Questa soluzione riesce a gestire la dipendenza dell'ingrediente nei confronti della ricetta a cui appartiene, è stata però scartata perchè non sembra scalabile in quanto all'aumentare delle ricette in cui è presente lo stesso ingrediente, aumentano le proprietà necessarie per descrivere le diverse quantità dello stesso, appesantendo inutilmente l'ontologia. Nel caso in analisi sono necessarie ben due proprietà distinte per descrivere i valori della quantità del Basilico, contenuto nella ricetta Linguine al pesto e MilleFoglie di melanzane. Se si dovesse usare questo sistema per tutti gli ingredienti di tutte le ricette che potrebbe contenere l'ontologia, quest'ultima sarebbe troppo pesante e non scalabile.

4.2.3 Terza soluzione: creazione di un database

Con l'ultima soluzione esposta, oltre che appesantire l'ontologia e a renderla non scalabile, si sarebbe messo in ombra il suo scopo principale, ovvero quello di descrivere le relazioni e le proprietà semantiche costruite tra gli ingredienti. Si è scelto così di creare un database interno all'applicazione in cui si sono salvati i possibili valori diversi di quantità di un ingrediente,

rispetto alle ricette a cui appartiene. Sono state definite quindi due tabelle, una riportante la quantità e la misura di un ingrediente in base alla ricetta, e l'altra riportante le informazioni nutrizionali di ogni ingrediente, nel nostro caso la quantità su 100 grammi di proteine, lipidi, carboidrati e calorie, prese dall'*INRAN, Istituto Nazionale di Ricerca per gli Alimenti e la Nutrizione*. Attraverso questa soluzione si è distinto la funzionalità semantica dell'ontologia, riportante i diversi tipi di alimento, i gruppi di classificazione degli alimenti e la classificazione di una ricetta, da quella semplice del database di riportare i possibili valori multipli delle quantità di un ingrediente.

4.3 Il modello dei dati

Il modello dei dati utilizzato per la rappresentazione delle ricette è fondamentalmente costituito da due classi: la classe **Ingrediente** e la classe **Ricetta**. La classe Ingrediente ha una serie di attributi che ne specificano le caratteristiche:

- nome : nome dell'ingrediente. Esempio Aglio
- quantità : quantità dell'ingrediente. Esempio 10
- misura : unità di misura della quantità. Esempio g

Altri esempi possono essere:

patate 2 (non viene specificata la misura)

latte 2 bicchieri

zucchero 4 cucchiari

noci 120 g

- peso : rappresenta il livello di importanza dell'ingrediente all'interno della ricetta. All'inizio tutti gli ingredienti hanno stesso peso, ma, all'interno del software, viene data all'utente la possibilità di specificare un certo numero di ingredienti principali. La specifica di tali ingredienti comporta un aumento del peso all'interno della ricetta e questo,

a seconda del fatto che tale ingrediente sia presente o meno all'interno della ricetta Wellness, andrà a incidere in maniera sostanziale nel calcolo della similarità.

- *proteine100* : percentuale di proteine su 100 g di ingrediente
- *carboidrati100* : percentuale di carboidrati su 100 g di ingrediente
- *lipidi100* : percentuale di lipidi su 100 g di ingrediente
- *calorie100* : Kcal apportate da 100 g di ingrediente
- *proteineAss* : grammi di proteine contenute nella quantità indicata di ingrediente.

$$proteineAss = \frac{proteine100}{100} * quantita$$

- *carboidratiAss* : grammi di carboidrati contenuti nella quantità indicata di ingrediente.

$$carboidratiAss = \frac{carboidrati100}{100} * quantita$$

- *lipidiAss* : grammi di lipidi contenuti nella quantità indicata di ingrediente.

$$lipidiAss = \frac{lipidi100}{100} * quantita$$

- *calorieAss* : Kcal apportate dalla quantità indicata di ingrediente.

$$calorieAss = \frac{calorie100}{100} * quantita$$

I rimanenti attributi non rappresentano delle caratteristiche specifiche di un ingrediente, ma servono per memorizzare dei dati utili ai vari algoritmi di similarità.

- *qu_pe* : prodotto della quantità per il peso

$$qu_pe = quantita * peso$$

- qu_pe_norm : prodotto della quantità per il peso normalizzato

$$qu_pe_norm = \frac{qu_pe}{\sum_i qu_pe_i}$$

- cal_pe : prodotto delle calorie per il peso

$$cal_pe = calorie * peso$$

- cal_pe_norm : prodotto delle calorie per il peso normalizzato

$$cal_pe_norm = \frac{cal_pe}{\sum_i cal_pe_i}$$

La classe **Ricetta** invece è composta dai seguenti attributi:

- $nome$: nome della ricetta
- $numPersone$: numero di persone per cui è pensata la ricetta
- $preparazione$: procedimento di preparazione della ricetta
- $proteineTot$: proteine apportate dall'intera ricetta. Si calcola come la sommatoria delle proteine dei singoli ingredienti componenti la ricetta

$$proteineTot = \sum_i proteineAss_i$$

- $carboidratiTot$: carboidrati apportati dell'intera ricetta

$$carboidratiTot = \sum_i carboidratiAss_i$$

- $lipidiTot$: lipidi apportati dall'intera ricetta

$$lipidiTot = \sum_i lipidiAss_i$$

- `calorieTot` : Kcal apportate dall'intera ricetta

$$calorieTot = \sum_i calorieAss_i$$

- `root` : l'insieme degli ingredienti di una ricetta viene rappresentata con una struttura dati ad albero e l'attributo in questione ne rappresenta la radice. L'esigenza di utilizzare tale struttura deriva dal fatto che un certo ingrediente di una ricetta può essere costituito da una serie di sottoingredienti. Quindi gli elementi (ingredienti) collegati alla radice sono di tipo **Ingrediente**, e tale oggetto (nel caso della sottoricetta), può avere come figli una serie di oggetti **Ingrediente**.

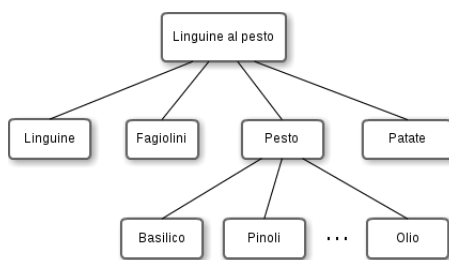


Figura 4.6: Esempio della sotto ricetta pesto

4.4 Software

In questo capitolo sarà mostrato il funzionamento del software, comprendendo l'architettura generale con la descrizione dei moduli principali, i sequence diagram e le schermate fondamentali. Questi riprendono anche il software *cuciniamo*, usato come parser di un testo semi strutturato inserito dall'utente, che, come ampiamente spiegato nel capitolo corrispondente, permette il lavoro successivo di similarità fra la ricetta inserita e quella presente nell'ontologia wellness. Viene spiegato in breve al lettore il meccanismo dell'intero software:

- L'utente inserisce una ricetta che può essere un file *.txt* o delle stringhe facendo un semplice *cut and paste* da uno dei numerosi siti di cucina on-line
- Cliccando sul tasto compila il software costruirà l'albero della ricetta (creando un oggetto MyRicetta) mostrando, opzionalmente, la sua struttura e gli ingredienti necessari. Crea inoltre un file xml, con il nome della ricetta come nome, salvandolo in una cartella a parte
- L'utente ora può richiedere che venga effettuata la similarità: il software cerca nell'ontologia se è presente una ricetta wellness con lo stesso nome e, in caso affermativo, costruisce due oggetti **Ricetta**, uno per la ricetta inserita dall'utente e l'altro per quella trovata nell'ontologia.
- Il modulo che si occupa di calcolare la similarità, attraverso gli opportuni metodi, restituisce all'utente un valore percentuale.

4.4.1 Architettura

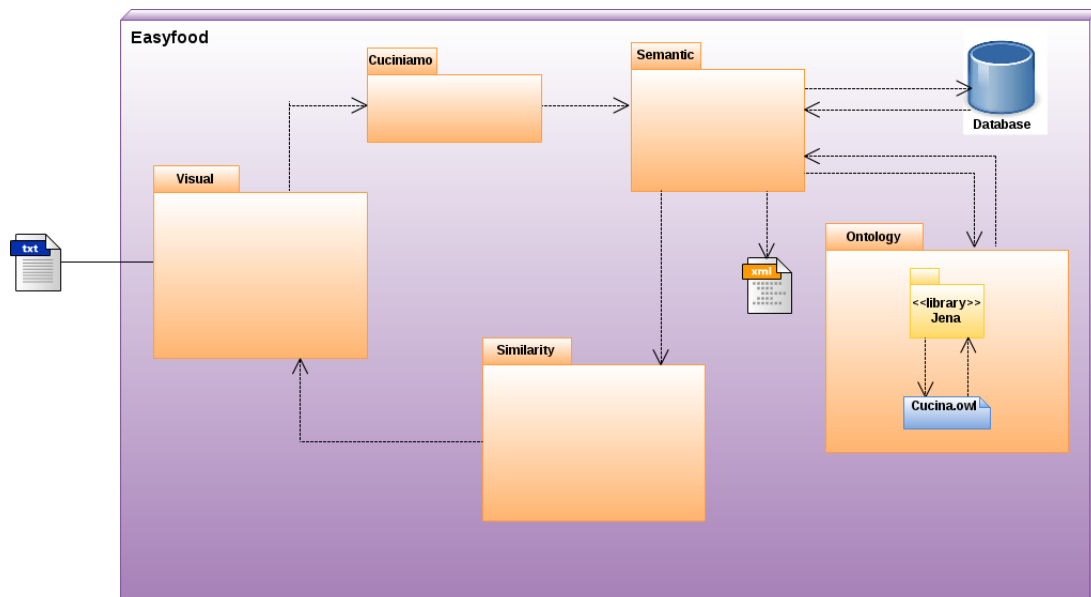


Figura 4.7: Architettura

La figura 4.6 mette in evidenza i moduli principali dell'architettura. Il modulo *Cuciniamo* si occupa della compilazione della ricetta inserita dall'utente, passata al modulo *Semantic* che, attraverso l'interazione con il *database* e l'*ontologia*, crea due istanze Ricetta corrispondenti alla ricetta inserita dall'utente e a quella wellness trovata nell'*ontologia*. Il *database* contiene le informazioni nutrizionali degli alimenti (utilizzati come ingredienti sia nella ricetta dell'utente che in quella wellness) e le quantità necessarie per la preparazione della ricetta wellness. Con il supporto della libreria *Jena* l'*ontologia* fornisce le informazioni strutturali e semantiche della ricetta wellness. Dopo aver generato il file .xml della ricetta inserita dall'utente, il modulo *Similarity* si occupa del calcolo riguardante la similarità che verrà mostrato a video grazie all'interazione con modulo Visual, responsabile della grafica dell'intero software.

4.4.1.1 Database

Come database si è scelto di utilizzare *Apache Derby* per semplicità e comodità d'uso. *Apache Derby* è un open source relational database management system (RDBMS) basato su tecnologia Java e SQL infatti, scritto e implementato interamente nel linguaggio di programmazione Java, fornisce all'utente un *database engine* che può essere strettamente incorporato in ogni sistema basato su Java. Dopo aver installato il componente necessario si è passati alla creazione vera e propria del database: si è creato uno *schema CUCINAWELLNESS* all'interno del database *MyDB* definendo due tabelle: *CUCINAWELLNESS.NUTRIENTIINGREDIENTE* e *CUCINAWELLNESS.PESOINGREDIENTE*. La prima tabella riguarda le informazioni nutrizionali degli alimenti che sono utilizzati come ingredienti nelle ricette: per ogni alimento sono riportati i valori (in formato DOUBLE) in grammi delle quantità di carboidrati, lipidi, proteine e calorie su 100 grammi di alimento. La seguente figura raffigura la tabella appena descritta.

NOME [VARCHAR(30)]	CARBOIDRATI [DOUBLE]	LIPIDI [DOUBLE]	PROTEINE [DOUBLE]	CALORIE [DOUBLE]
Fagiolini	2.0	0.0	2.0	18.0
patate	17.0	1.0	2.0	85.0
linguine	84.0	4.0	12.0	353.0
pesto genovese	20.0	10.0	20.0	300.0
aglio	8.0	0.0	0.0	90.0
basilico	5.0	0.0	3.0	39.0
ghiaccio	0.0	0.0	0.0	0.0
olio extra vergine di oli	0.0	100.0	0.0	899.0
parmigiano	0.0	28.0	33.0	387.0
pinoli	4.0	50.0	31.0	595.0
sale	0.0	0.0	0.0	0.0
linguine al pesto	278.0	44.0	50.0	1624.0
limone	1.0	0.0	0.0	6.0
melanzane	1.0	0.0	1.0	18.0
pasta fillo	79.0	1.0	10.0	353.0
pepe rimbas	63.0	3.0	10.0	0.0
pomodori	3.0	0.0	1.0	19.0
millefoglie di melanzane	116.0	19.0	20.0	683.0
salsa di pomodoro	3.0	0.0	1.0	18.0
olio	0.0	100.0	0.0	899.0
pecorino	32.0	0.0	25.0	1640.0
noci	5.0	57.0	10.0	582.0
lasagne	79.0	1.0	10.0	353.0
scamorza	1.0	25.0	25.0	1398.0
origano	68.0	4.0	9.0	265.0
farina	77.0	0.0	11.0	345.0
rucola	3.0	0.0	2.0	25.0

Figura 4.8: Tabella nutrienti ingrediente

La seconda tabella invece, che è risultata necessaria per la mancata capacità di modellazione attraverso il linguaggio *owl*, rappresenta, per ogni ingrediente contenuto in una specifica ricetta, i valori relativi di quantità e misura. Come si può notare dalla figura sottostante l'ingrediente *basilico*, contenuto sia nella ricetta “linguine al pesto wellness” che nella ricetta “millefoglie di melanzane”, ha come valori di quantità rispettivamente *40g* e *10g*.

NOME [VARCHAR(30)]	RICETTA [VARCHAR(30)]	QUANTITA [VARCHAR(15)]	MISURA [VARCHAR(10)]
patate	linguine al pesto	200	g
fagiolini	linguine al pesto	160	g
linguine	linguine al pesto	280	g
aglio	pesto genovese	1	spicchio
basilico	pesto genovese	40	g
ghiaccio	pesto genovese	25	g
olio extra vergine di oli	pesto genovese	20	g
parmigiano	pesto genovese	10	g
pinoli	pesto genovese	15	g
sale	pesto genovese	qb	NULL
aglio	millefoglie di melanzane	1	spicchio
basilico	millefoglie di melanzane	10	g
limone	millefoglie di melanzane	50	g
melanzane	millefoglie di melanzane	500	g
olio extra vergine di oli	millefoglie di melanzane	10	g
pasta fillo	millefoglie di melanzane	120	g
pepe rimbas	millefoglie di melanzane	qb	
pomodori	millefoglie di melanzane	120	g
sale	millefoglie di melanzane	qb	NULL

Figura 4.9: Tabella peso ingrediente

4.4.1.2 Semantic

Il modulo Semantic, preso come input l'oggetto MyRicetta creato precedentemente attraverso l'uso del parser fornito con l'applicazione *Cuciniamo* crea il documento *nomeRicetta.xml* corrispondente come si può vedere dal listato di codice xml seguente:

```
<Ricetta xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" nome="Linguine al pesto">
  <Opzioni>
    <Salva>linguinealpestoBuonissimo</Salva>
    <Tempo>0</Tempo>
  </Opzioni>
  <ListaIngredienti>
    <Ingrediente misura="g" quantita="300" tipo="linguine"/>
    <Ingrediente misura="manciata" quantita="1" tipo="basilico"/>
    <Ingrediente misura="spicchio" quantita="1" tipo="aglio"/>
    <Ingrediente misura="g" quantita="20" tipo="olio"/>
    <Ingrediente misura="g" quantita="10" tipo="parmigiano"/>
    <Ingrediente misura="g" quantita="80" tipo="patate"/>
    <Ingrediente misura="g" quantita="150" tipo="fagiolini"/>
    <Ingrediente tipo="sale"/>
  </ListaIngredienti>
  <Procedimento>Sbucciate la patata e dopo averla lavata, tagliatela a piccoli dadini. Eliminate le estremità dei fagiolini compresi i filacci e lavate con cura anch'essi. A questo punto mettete in una pentola con abbondante acqua salata le patate e aggiungete anche i fagiolini. Fate cuocere il tutto, e quasi al termine della cottura aggiungete la pasta che quindi cuocerà direttamente con le verdure. Quando la pasta sarà cotta scolatela, conservando un po' dell' acqua di cottura e mettete in una ciotola. Nel frattempo, lavate con cura il basilico ed asciugatelo. Dopo di che mettelo in un frullatore ed aggiungete il parmigiano, l' aglio e l'olio e tritate il tutto molto finemente. Quando il pesto sarà ben amalgamato aggiungetevi un po' d'acqua di cottura della pasta ed usate la salsa che ne ottenete per condire la pasta con le verdure.</Procedimento>
</Ricetta>
```

Figura 4.10: File .xml generato

In seguito alla creazione del file *.xml* viene definita un'istanza della classe Ricetta corrispondente alla ricetta inserita dall'utente: questa, come visto nel modello dei dati, estende la classe *NTree* e rappresenta una struttura

ad albero in cui ogni nodo contiene, oltre che il puntatore al nodo padre e ai figli, anche un elemento *Ingrediente*. Per la creazione completa di ogni istanza *Ingrediente* il modulo si occupa di interrogare il database per prelevare le informazioni nutrizionali corrispondenti. Infine, dopo aver fatto una ricerca nell'ontologia per verificare la presenza di una ricetta con lo stesso nome inserito dall'utente, si crea in automatico un'altra istanza *Ricetta* che avrà come elementi gli ingredienti contenuti nella ricetta wellness trovata nell'ontologia. Per quest'ultima fase è comunque necessaria anche l'interazione con il database sia per prelevare le informazioni nutrizionali degli ingredienti wellness che per prelevare le quantità necessarie per la preparazione.

4.4.1.3 Ontology

Questo modulo si riserva il compito di gestire la connessione con l'ontologia *CucinaWellness.owl* spiegata nei dettagli nella sezione corrispondente. Per far ciò è stato utilizzato un open source Java framework, chiamato *Apache Jena*, che mette a disposizione del programmatore una serie di metodi Java per la manipolazione dei dati contenuti in un'ontologia. Inoltre sono state definite delle query di default, standard, che sono utilizzate appunto per interrogare l'ontologia fornita come supporto semantico: la seguente query, scritta nel linguaggio SPARQL, verifica se due diversi ingredienti, passati come argomento, appartengono alla stessa classe definita nella classificazione degli alimenti.

```
prefix rdfs: <" + RDFS.getURI() + ">
prefix rdf: <" + RDF.getURI() + ">
ASK
WHERE {
  ?x rdf:type ?padre .
  ?y rdf:type ?padre .
  ?x rdfs:label ?nomeX .
  ?y rdfs:label ?nomeY .
  FILTER regex(?nomeX, "\"" + getIngr1() + "\", \"i\") .
  FILTER regex(?nomeY, "\"" + getIngr2() + "\", \"i\")}
```

4.4.1.4 Similarity

Come si evince dal nome, il modulo Similarity è quello che permette di effettuare la similarità fra due ricette. Esso è composto da tre parti ognuna delle quali si occupa di calcolare rispettivamente:

- la similarità semantica
- la similarità per ingredienti
- la similarità per fattori nutritivi

Per ognuno di essi è stata implementata una classe Java che dati in ingresso due oggetti di tipo **Ricetta**, provvede al calcolo della similarità. Questi tre moduli sono coadiuvati nel loro compito da un quarto componente che si occupa di implementare delle operazioni preliminari. Entrando nel dettaglio, si può dire che gli oggetti di tipo **Ricetta** utilizzati dai tre algoritmi non sono quelli forniti direttamente dal modulo Semantic (vedi Figura 4.6), ma vengono preliminarmente sottoposti ad alcune formattazioni. Affinchè gli algoritmi possano adempiere al proprio compito sono necessarie tre operazioni:

- appiattimento della struttura ad albero degli ingredienti della ricetta
- formattazione della dicitura di quantità e misura degli ingredienti (vedi Sezione 3.2)
- calcolo dei pesi degli ingredienti con eventuale modifica in caso di indicazione di ingredienti principali

All'interno del modulo Similarity è stata infine implementata l'analisi di sensitività. Per scelta di progetto questa è stata effettuata esclusivamente sulla similarità semantica. Un'ulteriore classe infatti provvede a richiamare tale similarità cambiando volta per volta il fattore di premiazione per gli ingredienti che hanno padre uguale e il fattore di penalizzazione per gli ingredienti che non hanno alcuna corrispondenza all'interno dell'altra ricetta.

4.4.1.5 Visual

Il modulo Visual è stato ampliato e aggiornato, rispetto al software iniziale *Cuciniamo*, con l'utilizzo dell'ambiente di sviluppo *NetBeans* e del framework *Swing* per Java orientato allo sviluppo delle interfacce grafiche. Questo modulo quindi, che gestisce la grafica dell'applicazione, interagisce con l'utente permettendo l'inserimento di una ricetta tramite testo libero o caricamento da file e tutte le scelte di opzione riguardo al tipo di similarità a cui si è interessati. Quest'ultima parte si preoccupa di mostrare a video gli ingredienti della ricetta inserita dall'utente dando la possibilità di scegliere eventuali ingredienti principali che influenzerebbero il calcolo della similarità gestito appunto dal modulo corrispondente.

4.4.2 Sequence Diagram

I seguenti Sequence Diagram descrivono i relativi scenari che potrebbero verificarsi in seguito all'avvio e all'interazione dell'utente con il software. Per la visualizzazione della ricetta inserita si utilizzano il parser, che compila, il syntax tree, che costruisce l'albero, e il visual che permette, tramite le *JSwing Java*, di mostrare a video il risultato. Questo scenario sarà ripetuto nei prossimi in quanto sono necessari per le successive azioni. La seguente figura mostra il Sequence Diagram corrispondente.

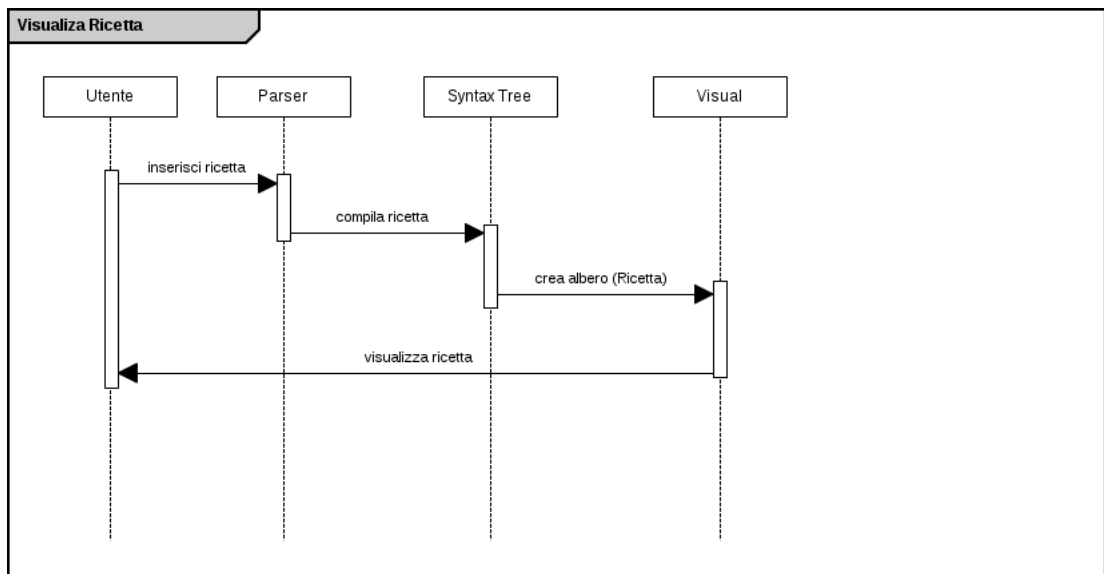


Figura 4.11: Visualizza ricetta sequence diagram

Per la creazione del file .xml vi è un componente dedicato. Esso salva il file generato in una cartella all'interno del software.

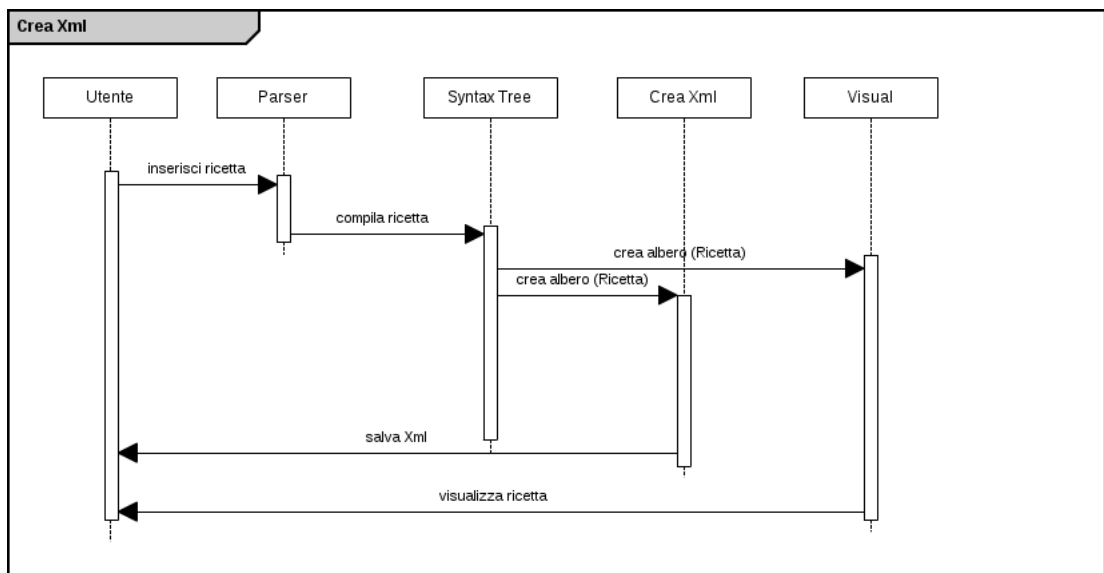


Figura 4.12: Crea xml sequence diagram

In seguito alla creazione del file .xml vengono create due istanze ricetta, corrispondenti a quella inserita dall'utente e a quella presente nel database.

Il seguente Sequence Diagram mostra la generazione della ricetta wellness: il componente Ricetta Builder, con il nome della ricetta inserito dall'utente, interrogando il database e l'ontologia restituisce la ricetta corrispondente.

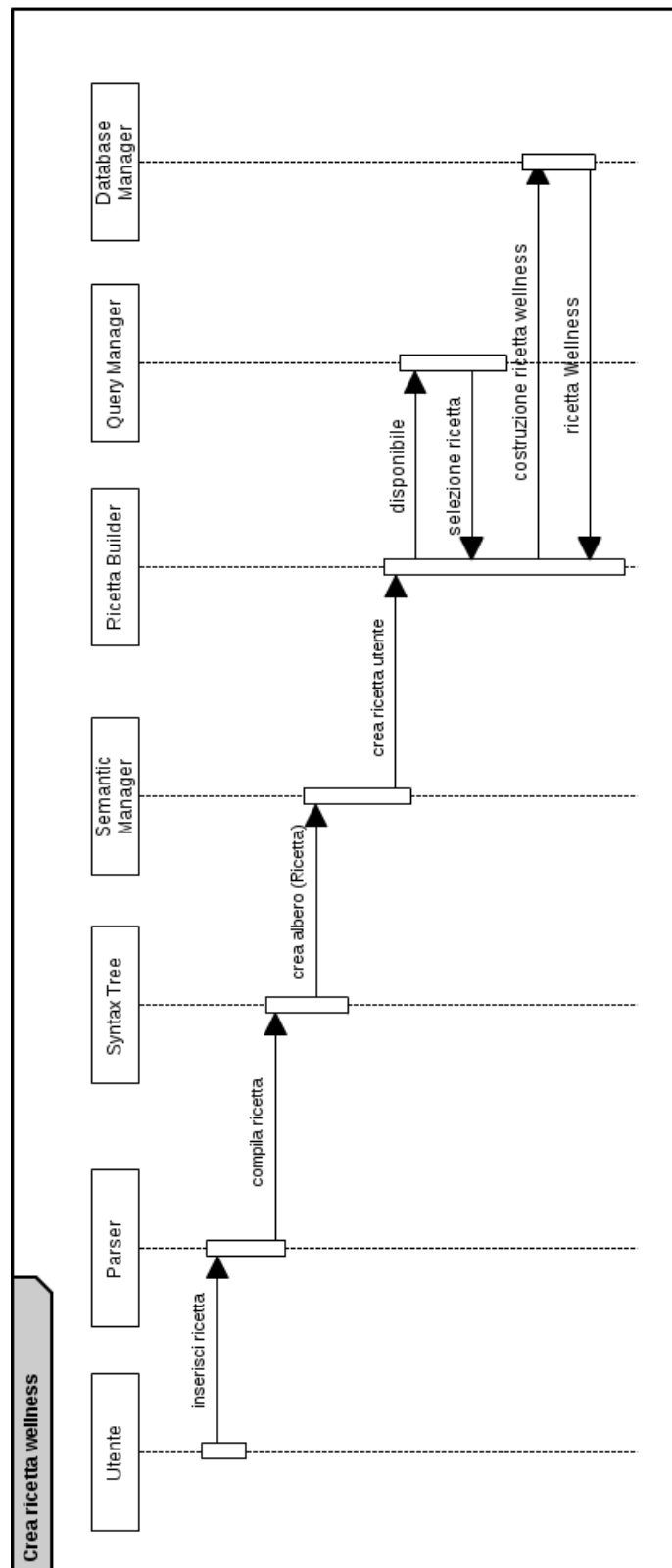
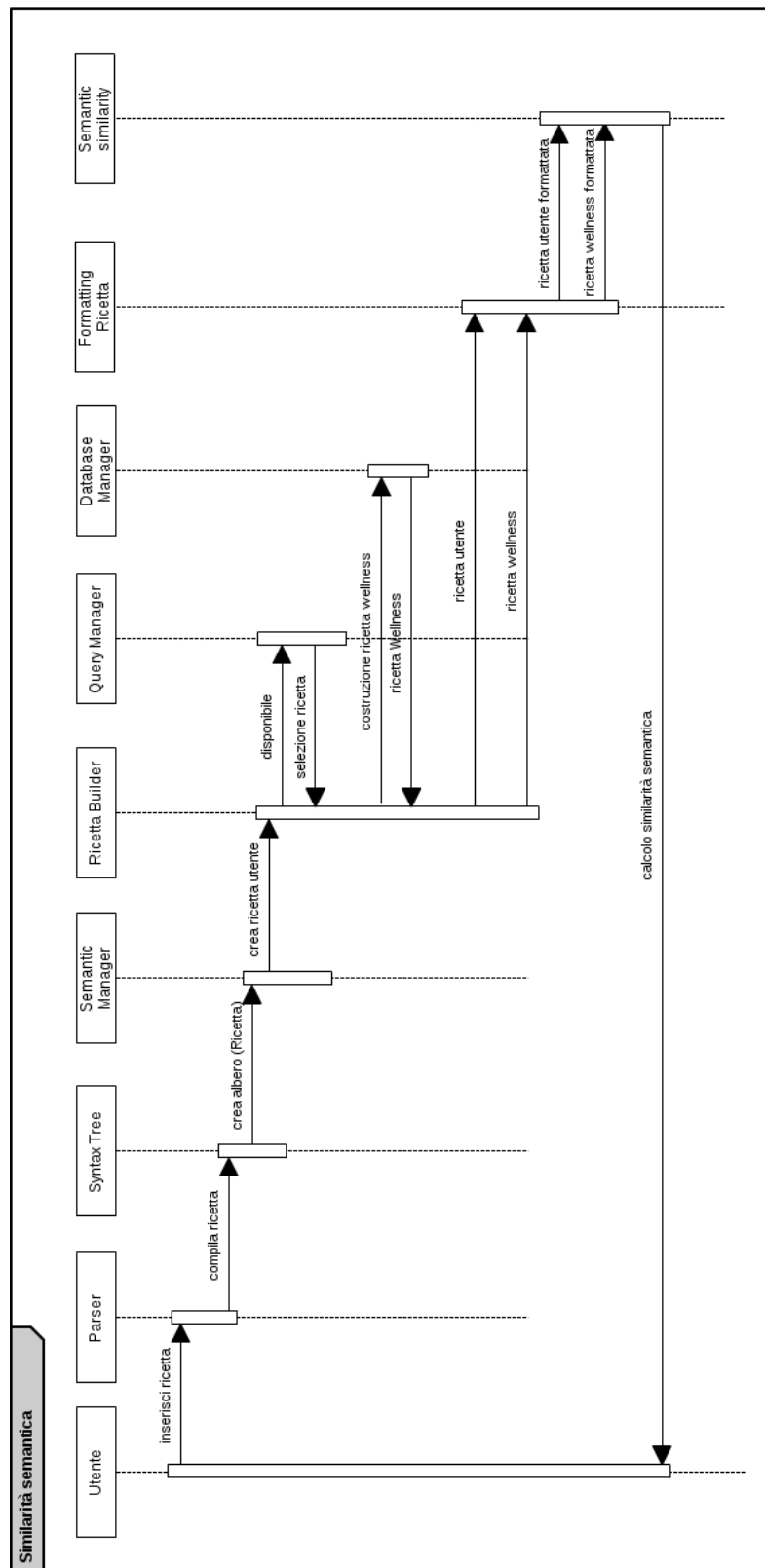


Figura 4.13: Crea ricetta wellness sequence diagram

In uno scenario in cui l'utente è interessato a sapere quanto è simile, secondo il metodo della similarità semantica, la sua ricetta alla ricetta wellness si aggiungono i componenti *Formatting Ricetta* e *Semantic Similarity*. Il primo si occupa di tradurre in modo standard le quantità e le misure relative agli ingredienti delle ricette, in particolare si avrà una corrispondenza tra la quantità inserita (“un”, “uno”, “1”,...) e una quantità intera, e tra la misura inserita (“spicchio”, “mazzo”, “cucchiaino”,..) e una misura corrispondente in grammi. Il secondo infine, ricevendo le due ricette generate, calcola la similarità desiderata, in questo caso la similarità semantica.



106
 Figura 4.14: Similarità semantica sequence diagram

4.4.3 Schermate fondamentali

Dopo descritto il funzionamento del software, la sua architettura, l'interazione fra i suoi componenti e i possibili scenari che si possono verificare, è utile mostrare, attraverso una simulazione di uno scenario, il software vero e proprio con i suoi componenti grafici, riportando le finestre utili a capire il software nel suo complesso. Se si avvia l'applicazione Easyfood la schermata principale che verrà mostrata è riportata nella seguente figura:

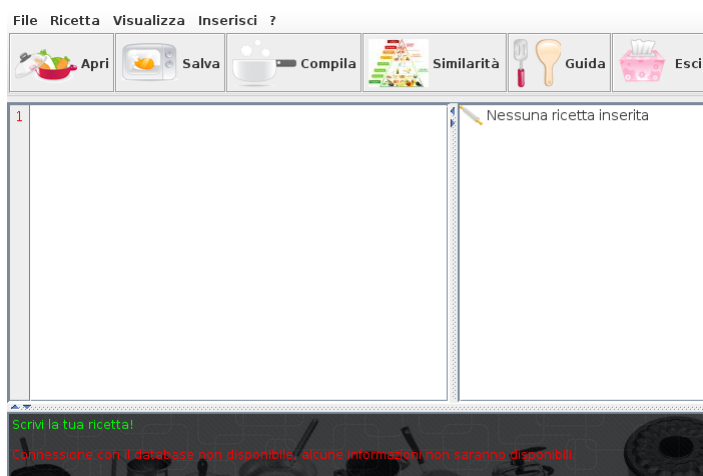


Figura 4.15: Schermata principale

Come si può notare vi sono due pannelli principali, uno riservato all'inserimento della ricetta, tramite un file di testo selezionato dal proprio computer cliccando sul bottone apri oppure scrivendo direttamente del testo semi strutturato, l'altro invece utile per la visualizzazione degli ingredienti con le relative quantità. La figura 4.15, raffigurante la schermata in seguito alla compilazione della ricetta, avvenuta cliccando il tasto compila, mette in risalto il metodo di inserimento di una ricetta: nel primo pannello infatti è necessario inserire il titolo della ricetta, una sezione riguardante gli ingredienti e un'ultima sezione che riporta il testo per la preparazione. Il secondo pannello, come già annunciato, riporta tramite una struttura ad albero, in questo caso si tratta di un albero a due livelli, la radice e le foglie come ingredienti, la ricetta inserita con i suoi ingredienti e le relative quantità e misure.

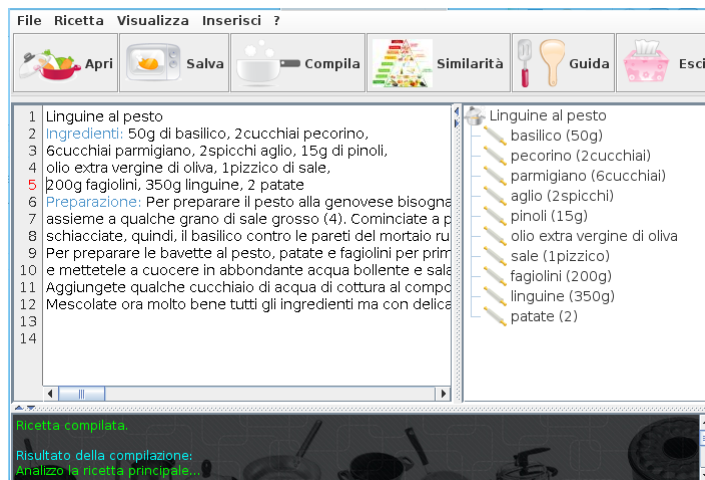


Figura 4.16: Schermata compilazione ricetta

In seguito alla compilazione della ricetta, premendo il bottone Similarità, si accede alla schermata relativa alla similarità semantica dell'applicazione. Questa schermata, come presenta la figura 4.16, è suddivisa in tre sezioni, o pannelli: il primo, se vi è la necessità dell'utente, dedicato alla selezione degli ingredienti principali che, in questo caso, risulta essere il pecorino. La selezione degli ingredienti principali si ricorda che è soggettiva, ovvero dipende dai gusti e dalle preferenze dell'utente, ed andrà a modificare il peso relativo all'ingrediente, influenzando il calcolo della similarità. Il secondo pannello è dedicato alla selezione della similarità in base al tipo che si preferisce: similarità semantica, per ingredienti e per nutrienti. Il terzo ed ultimo pannello riporta invece i risultati suddivisi per tipo di similarità.

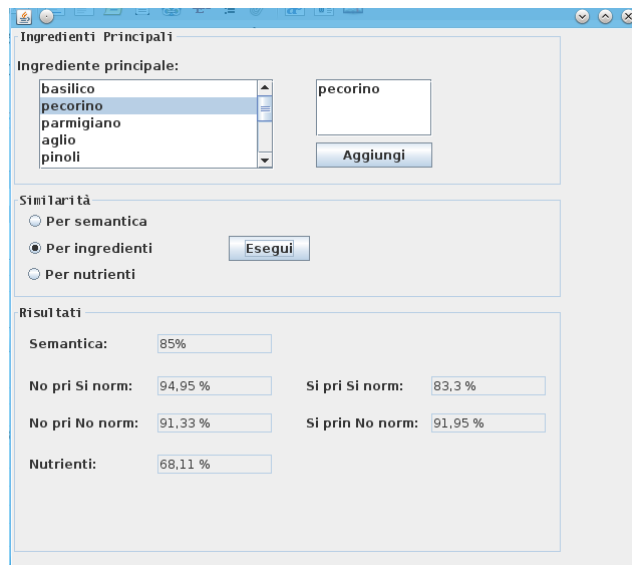


Figura 4.17: Schermata similarità

Capitolo 5

Risultati

In questo capitolo si vogliono presentare e commentare i risultati prodotti dal software realizzato. Verranno prese varie versioni delle seguenti ricette: linguine al pesto e millefoglie di melanzane, e verranno mostrati i valori di similarità forniti in output. Si discuterà della validità di tali valori e di quanto questi si discostino al variare degli ingredienti principali indicati dall'utente. In conclusione, per ognuna delle due ricette, verrà presentata l'analisi di sensitività sui risultati forniti dalla similarità semantica.

5.1 Linguine al pesto

L'obiettivo è quello di simulare l'immissione della ricetta da parte dell'utente e verificare la consistenza delle indicazioni che gli vengono restituite. Si suppone che questi fornisca in input il cut&paste di una ricetta trovata in una pagina web; pertanto sono state cercate in internet varie versioni della stessa ricetta. Per quello che riguarda le linguine al pesto vengono fornite 3 versioni trovate in altrettanti siti di ricette. In Tabella 5.1 vengono indicati gli ingredienti di tali ricette.

giallozafferano.it	buonissimo.org	mangiarebene.com
Ingrediente	Ingrediente	Ingrediente
linguine	linguine	linguine
basilico	basilico	basilico
patate	patate	patate
parmigiano	parmigiano	parmigiano
aglio	aglio	aglio
olio	olio	olio
sale	sale	sale
fagiolini	fagiolini	fagiolini
pinoli		pinoli
pecorino		pecorino
		noci

Tabella 5.1: ricette prese in esame

Qui di seguito invece c'è la ricetta wellness che viene presa come punto di riferimento:

wellness
Ingrediente
linguine
basilico
patate
parmigiano
aglio
olio
sale
fagiolini
pinoli
ghiaccio

Tabella 5.2: ricetta wellness

5.1.1 Similarità semantica

Per ognuna delle versioni si presenta il risultato di similarità semantica. Per favorire il confronto, e individuare quali fattori determinano gli scostamenti di risultato della similarità, viene, volta per volta, affiancata alla ricetta inserita dall'utente, la stessa ricetta con qualche modifica in modo tale da

mettere in evidenza il funzionamento dell’algoritmo in funzione dell’informazione semantica. In particolar modo si andranno a verificare quei casi in cui un ingrediente appartiene alla stessa classe di un altro ingrediente.

Partiamo dunque dalla ricetta trovata su giallozafferano.it riportando anche le varie modifiche effettate (nelle versioni modificate, vengono indicati in grassetto quegli ingredienti che, all’interno dell’ontologia, hanno padre uguale rispetti a quelli che stanno sulla stessa riga):

giallozafferano.it	giallozafferano 2	giallozafferano 3	wellness
Ingrediente	Ingrediente	Ingrediente	Ingrediente
linguine	spaghetti	spaghetti	linguine
basilico	basilico	basilico	basilico
patate	patate	patate	patate
parmigiano	parmigiano	pecorino	parmigiano
aglio	aglio	aglio	aglio
olio	olio	olio	olio
sale	sale	sale	sale
fagiolini	fagiolini	ceci	fagiolini
pinoli	pinoli	pinoli	pinoli
			ghiaccio
pecorino	pecorino	pecorino	
Similarità semantica	85%	80%	72,5%

Tabella 5.3: Similarità semantica giallozafferano.it

Il valore di similarità è abbastanza alto per la prima versione di giallozafferano.it. Tale risultato è giustificato dal fatto che ben 9 ingredienti coincidono. Si ha quindi una similarità all’85% a causa di pecorino e ghiaccio, ingredienti non corrispondenti. Le altre versioni risultano essere ricette “peggiorate”, nel senso che vengono alterati gli ingredienti in modo tale che appartengono alla stessa classe di quelli corrispondenti, e quindi non otterranno un peso di similarità massimo ma leggermente penalizzato.

buonissimo.org	buonissimo 2	buonissimo 3	wellness
Ingrediente	Ingrediente	Ingrediente	Ingrediente
linguine	spaghetti	spaghetti	linguine
basilico	basilico	basilico	basilico
patate	patate	patate	patate
parmigiano	parmigiano	pecorino	parmigiano
aglio	aglio	aglio	aglio
olio	olio	olio	olio
sale	sale	sale	sale
fagiolini	fagiolini	fagiolini	fagiolini
			pinoli
			ghiaccio
Similarità semantica	75%	70%	65%

Tabella 5.4: Similarità semantica buonissimo.org

Nel caso di buonissimo.org la similarità scende notevolmente rispetto al caso precedente. Una similarità del 75% è giustificata dal fatto di avere solo 8 ingredienti corrispondenti fra loro e altri due per i quali il software non trova alcuna corrispondenza nella ricetta inserita dall'utente. Come nel caso precedente vengono utilizzate come esempio altre ricette modificate in modo peggiorativo, tale per cui si possa vedere la variazione del valore di similarità, dipendente dall'informazione semantica di appartenenza alla stessa classe di alimento definita nell'ontologia.

	mangiarebene.com	mangiarebene 2	mangiarebene 3	wellness
	Ingrediente	Ingrediente	Ingrediente	Ingrediente
	linguine	linguine	spaghetti	linguine
	basilico	basilico	basilico	basilico
	patate	patate	patate	patate
	parmigiano	pecorino	pecorino	parmigiano
	aglio	aglio	aglio	aglio
	olio	olio	olio	olio
	sale	sale	sale	sale
	fagiolini	fagiolini	fagiolini	fagiolini
	pinoli	pinoli	pinoli	pinoli
				ghiaccio
	noci	noci	noci	
Similarità semantica	85%	80%	75%	

Tabella 5.5: Similarità semantica mangiarebene.com

Infine, per quanto riguarda l'ultima ricetta presa in analisi, mangiarebene.com, a causa dei 2 ingredienti diversi si ottiene una similarità semantica dell'85%. Modificando la ricetta ancora per sottolineare il funzionamento della proprietà semantica, si può notare come si abbassano i valori delle corrispondenti similarità, andando a penalizzare quegli ingredienti che non coincidono ma hanno proprietà nutrizionali simili, appartenendo alla stessa classe.

5.1.2 Similarità per ingredienti

A differenza di quanto accade nella similarità semantica, in questo contesto vengono prese in considerazione le quantità degli ingredienti, pertanto le ricette sono:

giallozafferano.it		buonissimo.org		gustissimo.it		mangiarebene.com	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
linguine	350g	linguine	300g	linguine	400g	linguine	600g
basilico	50g	basilico	5g	basilico	20g	basilico	50g
patate	160g	patate	80g	patate	80g	patate	320g
parmigiano	90g	parmigiano	10g	parmigiano	25g	parmigiano	90g
aglio	30g	aglio	15g	aglio	30g	aglio	30g
olio	3g	olio	20g	olio	3g	olio	3g
sale	3g	sale	3g	sale	3g	sale	3g
fagiolini	200g	fagiolini	150g			fagiolini	30g
pinoli	15g			pinoli	12g	pinoli	12g
pecorino	30g			pecorino	25g	pecorino	30g
						noci	3g

Tabella 5.6: Ricette prese in esame

La ricetta Wellness invece è in Tabella 5.7:

wellness	
Ingrediente	Quantità
linguine	280g
basilico	40g
patate	200g
parmigiano	10g
aglio	15g
olio	20g
sale	3g
fagiolini	160g
pinoli	15g
ghiaccio	25g

Tabella 5.7: Ricetta Wellness

Così come nel paragrafo precedente, si procede con la presentazione dei risultati mostrando volta per volta le coppie di ricette in esame.

Partiamo con giallozafferano.it :

giallozafferano.it		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
linguine	350g	linguine	280g
basilico	50g	basilico	40g
patate	160g	patate	200g
parmigiano	90g	parmigiano	10g
aglio	30g	aglio	15g
olio	3g	olio	20g
sale	3g	sale	3g
fagiolini	200g	fagiolini	160g
pinoli	15g	pinoli	15g
		ghiaccio	25g
pecorino	30g		

Tabella 5.8: Confronto ricette wellness, giallozafferano.it

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		93,87%
si		88,46%
no	basilico	90,85%
si	basilico	90,54%
no	basilico,pinoli	91,58%
si	basilico,pinoli	91,3%
no	linguine,basilico,olio	90,31%
si	linguine,basilico,olio	90,82%

Tabella 5.9: Similarità quantitativa giallozafferano.it

I valori di similarità senza alcuna indicazione di ingredienti principali sono il più alto e il più basso. In ogni caso, pur considerando alcuni ingredienti come principali, o il fatto di normalizzazione o meno i prodotti delle quantità per i pesi, è possibile dire che i valori subiscono scostamenti minimi. Se per esempio consideriamo come unico ingrediente principale il basilico, la similarità si aggira attorno al 90%. Se invece prendiamo il caso dove oltre al basilico consideriamo i pinoli vediamo i valori crescere attorno al 91%. La crescita è giustificata dal fatto che l'utente ha inserito come ingredienti principali, due ingredienti presenti in entrambe le ricette. Allo stesso tempo tale crescita è così piccola perchè pur avendo addirittura la stessa quantità,

per quanto sia stato aumentato il peso dei pinoli si tratta solamente di un ingrediente di 15g.

buonissimo.org		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
linguine	300g	linguine	280g
basilico	5g	basilico	40g
patate	80g	patate	200g
parmigiano	10g	parmigiano	10g
aglio	15g	aglio	15g
olio	20g	olio	20g
sale	3g	sale	3g
fagiolini	150g	fagiolini	160g
		pinoli	15g
		ghiaccio	25g

Tabella 5.10: Confronto ricette wellness, buonissimo.org

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		92%
si		84,95%
no	basilico	89,34%
si	basilico	91,99%
no	basilico,linguine	89,09%
si	basilico,linguine	92,98%
no	linguine,parmigiano,fagiolini	92,3%
si	linguine,parmigiano,fagiolini	95,07%

Tabella 5.11: Similarità quantitativa buonissimo.org

Anche in questo caso i valori subiscono variazioni particolarmente basse. Concentrandoci sui valori con tre ingredienti principali, possiamo notare che i risultati di similarità vanno dal 92,3% al 95,07%. In effetti, i tre ingredienti presi in esame, non solo sono presenti in entrambe le ricette, ma andando a guardare le quantità, è possibile notare che hanno differenze riscalate di quantità. Se nel caso del parmigiano le quantità sono addirittura uguali, nel caso delle linguine e dei fagiolini si registrano rispettivamente 20g e 10g di differenza.

gustissimo.it		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
linguine	400g	linguine	280g
basilico	20g	basilico	40g
patate	80g	patate	200g
parmigiano	25g	parmigiano	10g
aglio	30g	aglio	15g
olio	3g	olio	20g
sale	3g	sale	3g
pinoli	12g	pinoli	15g
		fagiolini	160g
		ghiaccio	25g
pecorino	25g		

Tabella 5.12: Confronto ricette wellness, gustissimo.it

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		85,49%
si		72,7%
no	pinoli	77,16
si	pinoli	69,58
no	linguine,patate	76,76%
si	linguine,patate	64,56%
no	parmigiano,basilico,pinoli	81,08%
si	parmigiano,basilico,pinoli	74,65%

Tabella 5.13: Similarità quantitativa gustissimo.it

Al solito, la versione proposta da gustissimo.it presenta i valori più bassi di similarità. Oltre a questo, a differenza dei casi già presentati, si può osservare una maggiore variabilità dei risultati: si va da un minimo del 64,56% ad un massimo dell' 85,49%.

mangiarebene.com		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
linguine	600g	linguine	280g
basilico	50g	basilico	40g
patate	320g	patate	200g
parmigiano	90g	parmigiano	10g
aglio	30g	aglio	15g
olio	3g	olio	20g
sale	3g	sale	3g
fagiolini	30g	fagiolini	160g
pinoli	12g	pinoli	15g
		ghiaccio	25g
noci	3g		
pecorino	30g		

Tabella 5.14: Confronto ricette wellness, mangiarebene.com

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		91,13%
si		83,32%
no	linguine	83,64%
si	linguine	81,75%
no	linguine,parmigiano	83,36%
si	linguine,parmigiano	80,75%
no	basilico,aglio,pinoli	88,9%
si	basilico,aglio,pinoli	86,31%

Tabella 5.15: Similarità quantitativa mangiarebene.com

Infine la versione proposta da mangiarebene.com. In questo caso la similarità ritorna ad essere particolarmente alta e ad oscillare tra l'80% e il 90%. Volendo osservare la differenza fra i risultati ottenuti con due e con tre ingredienti principali, si registra in entrambi i casi un aumento di poco superiore al 5%. Gli ingredienti presi nei due casi in esame sono completamente diversi e l'aumento è chiaramente giustificato dalle quantità degli ingredienti coinvolti. Se nel caso con due ingredienti principali ci sono le linguine che hanno quantità doppia, nel caso con tre ingredienti principali, si riscontrano quantità più basse e un minore scarto fra la versione utente e quella wellness.

5.1.3 Similarità per nutrienti

La similarità per nutrienti si occupa di fornire un'indicazione di quanto due ricette sono simili dal punto di vista nutrizionale. Nell'algoritmo vengono presi in considerazione i tre principali macronutrienti: carboidrati, lipidi, proteine e quanto ognuno di questi contribuisca al computo delle Kcalorie totali della ricetta. Qui di seguito le ricette utilizzate:

giallozafferano.it		buonissimo.org		mangiarebene.com	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
linguine	350g	linguine	300g	linguine	600g
basilico	50g	basilico	5g	basilico	50g
patate	160g	patate	80g	patate	320g
parmigiano	90g	parmigiano	10g	parmigiano	90g
aglio	30g	aglio	15g	aglio	30g
olio	3g	olio	20g	olio	3g
sale	3g	sale	3g	sale	3g
fagiolini	200g	fagiolini	150g	fagiolini	30g
pinoli	15g			pinoli	12g
pecorino	30g			pecorino	30g
				noci	3g
Carboidrati	340,3g		270,05g		574,13g
Lipidi	51,3g		35,6g		63,11g
Proteine	92,55g		44,05g		121,72g
Calorie	2057,13Kcal		1493,82Kcal		3145,95Kcal

Tabella 5.16: Valori aggregati di macronutrienti per le varie ricette

Mentre in Tabella 5.17 è presente la ricetta Wellness:

wellness	
Ingrediente	Quantità
linguine	280g
basilico	40g
patate	200g
parmigiano	10g
aglio	15g
olio	20g
sale	3g
fagiolini	160g
pinoli	15g
ghiaccio	25g
Carboidrati	276,2,3g
Lipidi	43,5g
Proteine	49,95g
Calorie	1608,95Kcal

Tabella 5.17: Valori aggregati di macronutrienti per la ricetta Wellness

Per ognuna delle tre ricette in Tabella 5.16 si presentano alcune versioni appositamente modificate al fine di osservare il comportamento dei risultati prodotti dall'algorithm.

Cominciamo con la ricetta di giallozafferano.it :

giallozafferano.it 1		giallozafferano.it 2		giallozafferano.it 3		wellness	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
linguine	350g	linguine	300g	linguine	450g	linguine	280g
basilico	50g	basilico	50g	basilico	50g	basilico	40g
patate	160g	patate	160g	patate	110g	patate	200g
parmigiano	90g	parmigiano	50g	parmigiano	180g	parmigiano	10g
aglio	30g	aglio	30g	aglio	30g	aglio	15g
olio	3g	olio	3g	olio	3g	olio	20g
sale	3g	sale	3g	sale	3g	sale	3g
fagiolini	200g	fagiolini	160g	fagiolini	200g	fagiolini	160g
pinoli	15g	pinoli	15g	pinoli	15g	pinoli	15g
						ghiaccio	25g
pecorino	30g	pecorino	30g				
Carboidrati	340,3g		297,5g		415,8g		276,2g
Lipidi	51,3g		38,1g		80,1g		43,5g
Proteine	92,55g		72,55g		133,25g		49,95g
Calorie	2057,13Kcal		1709,73Kcal		2737,12Kcal		1608,95Kcal
Similarità per nutrienti	90,71%		95,83%		76,63%		

Tabella 5.18: Similarità per nutrienti giallozafferano.it

La versione originale proposta da giallozafferano.it ha una similarità del 90,71%. Pur trattandosi di un valore abbastanza elevato per quanto sono diverse le due ricette dal punto di vista nutrizionale, l'algoritmo fornisce un'indicazione coerente al variare dei valori della ricetta. Nella seconda prova ad esempio sono stati cambiate le quantità di alcuni ingredienti (in grassetto) in modo tale da rendere la ricetta più vicina alla versione Wellness ed effettivamente con una similarità del 95,83% l'algoritmo riesce a cogliere tali variazioni. La terza ed ultima prova invece propone un caso peggiorativo.

In Tabella 5.19 l'analisi di similarità per la ricetta di buonissimo.org:

buonissimo.org 1		buonissimo.org 2		buonissimo.org 3		wellness	
Ingrediente	Quant.	Ingrediente	Quant.	Ingrediente	Quantità	Ingrediente	Quant.
linguine	300g	linguine	280g	linguine	350g	linguine	280g
basilico	5g	basilico	30g	basilico	5g	basilico	40g
patate	80g	patate	80g	patate	80g	patate	200g
parmig.	10g	parmig.	10g	parmig.	80g	parmigiano	10g
aglio	15g	aglio	15g	aglio	15g	aglio	15g
olio	20g	olio	20g	olio	20g	olio	20g
sale	3g	sale	3g	sale	3g	sale	3g
fagiolini	150g	fagiolini	150g	fagiolini	150g	fagiolini	160g
						pinoli	15g
						ghiaccio	25g
Carboidrati	270,05g		254,5g		312,05g		276,2,3g
Lipidi	35,6g		34,8		57,2g		43,5g
Proteine	44,05g		42,39g		73,15g		49,95g
Calorie	1493,82Kcal		1422,18Kcal		1944,51Kcal		1608,95Kca
Similarità per nutrienti	97,61%		96,13%		93,05%		

Tabella 5.19: Similarità per nutrienti buonissimo.org

La versione originale della ricetta ha un valore di similarità molto alto e in effetti dal punto di vista calorico la differenza è veramente bassa. Nel fare queste valutazioni è necessario tenere conto del fatto che sono state prese ricette per 4 persone; quindi se sul totale della ricetta, come accade in questo caso, vi è uno scarto di circa 100Kcal, a livello di singola persona questo si traduce in una differenza di 25Kcal. Delle altre due versioni proposte, si vuole sottolineare il comportamento della seconda. Pur cambiando le quantità di alcuni ingredienti in modo da renderle più simili alla ricetta Wellness, il valore di similarità diventa più basso. Apparentemente questo comportamento sembra inspiegabile. In realtà, andando ad osservare i valori dei macronutrienti, l'abbassamento della similarità trova una giustificazione: il fatto di aver diminuito la quantità di linguine, che nella versione 1, a livello di carboidrati totali dava un valore molto simile al valore wellness, in questo caso ha comportato un abbassamento dei carboidrati al punto da aumentare la differenza tra le due ricette.

Infine la Tabella 5.20 riporta l'analisi per mangiarebene.com:

mangiarebene.com 1		mangiarebene.com 2		mangiarebene.com 3		wellness	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
linguine	600g	linguine	500g	linguine	400g	linguine	280g
basilico	50g	basilico	50g	basilico	50g	basilico	40g
patate	320g	patate	320g	patate	220g	patate	200g
parmigiano	90g	parmigiano	45g	parmigiano	90g	parmigiano	10g
aglio	30g	aglio	30g	aglio	30g	aglio	15g
olio	3g	olio	3g	olio	3g	olio	20g
sale	3g	sale	3g	sale	3g	sale	3g
fagiolini	30g	fagiolini	30g	fagiolini	30g	fagiolini	160g
pinoli	12g	pinoli	12g	pinoli	12g	pinoli	15g
						ghiaccio	25g
noci	3g	noci	3g	noci	3g		
pecorino	30g	pecorino	30g	pecorino	30g		
Carboidrati	574,13g		490,13g		389,13g		276,2,3g
Lipidi	63,11g		46,51g		54,11g		43,5g
Proteine	121,72g		94,87g		95,72g		49,95g
Calorie	3145,95Kcal		2589,13Kcal		2278,65Kcal		1608,95Kcal
Similarità per nutrienti	68,11%		79,69%		86,13%		

Tabella 5.20: Similarità per nutrienti mangiarebene.com

La versione originale di mangiarebene.com propone di utilizzare 600g di linguine: essendo più del doppio della quantità della ricetta Wellness, nel computo della similarità, è chiaramente questo il fattore determinante. Nelle successive prove infatti, oltre al parmigiano e alle patate, è stata modificato al ribasso la quantità di tale ingrediente e come ci si aspettava la similarità è aumentata.

5.1.4 Analisi di sensitività

L'analisi di sensitività è stata svolta facendo unicamente riferimento alla similarità semantica. Nel contesto delle linguine al pesto si è scelto di svolgerla utilizzando la ricetta proposta da mangiarebene.com. I valori di similarità semantica dipendono da due fattori fondamentali: la percentuale del peso che viene aggiunta quando ci sono ingredienti diversi ma con padri uguali e la percentuale del peso che viene sottratta in caso di ingredienti completa-

mente diversi. Nell'esecuzione della similarità semantica, questi due valori corrispondono rispettivamente al 50% e al 25%.

Quello che si vuole osservare attraverso questa analisi, è la variazione del valore di similarità in funzione della variazione di questi due fattori. Nella Tabella 5.23 è possibile osservare la solita ricetta wellness e le 4 versioni della ricetta utente utilizzate. Le varie versioni sono state modificate in modo da avere cambiamenti minimi. Gli ingredienti in grassetto sono quelli che il software riconosce come diversi, ma appartenenti alla stessa categoria di quelli che si trovano sulla stessa riga.

mangiarebene.com 1	mangiarebene.com 2	mangiarebene.com 3	mangiarebene.com 4	wellness
Ingrediente	Ingrediente	Ingrediente	Ingrediente	Ingrediente
linguine	linguine	linguine	linguine	linguine
basilico	origano	origano	origano	basilico
patate	patate	patate	patate	patate
parmigiano	parmigiano	pecorino	pecorino	parmigiano
aglio	aglio	aglio	aglio	aglio
olio	olio	olio	olio	olio
sale	sale	sale	sale	sale
pinoli	pinoli	pinoli	pinoli	pinoli
fagiolini	fagiolini	fagiolini		fagiolini ghiaccio
pecorino	pecorino			
noci	noci	noci	noci	
		pepe	pepe	

Tabella 5.21: Versioni modificate di mangiarebene.com a confronto

I dati riportati in Tabella 5.24 mostrano i valori di similarità semantica a fronte di un'escursione massima del 20%, con passo 5%, dei due fattori:

padri uguali	no corrisp	Similarità1	Similarità2	Similarità3	Similarità4
0,40	0,15	85,5%	79,5%	73,5%	62%
0,40	0,20	84%	78%	72%	60%
0,40	0,25	82,5%	76,5%	70,5%	58%
0,40	0,30	81%	75%	69%	56%
0,40	0,35	79,5%	73,5%	67,5%	54%
0,45	0,15	85,5%	80%	74,5%	63%
0,45	0,20	84%	78,5%	73%	61%
0,45	0,25	82,5%	77%	71,5%	59%
0,45	0,30	81%	75,5%	70%	57%
0,45	0,35	79,5%	74%	68,5%	55%
0,50	0,15	85,5%	80,5%	75,5%	64%
0,50	0,20	84%	79%	74%	62%
0,50	0,25	82,5%	77,5%	72,5%	60%
0,50	0,30	81%	76%	71%	58%
0,50	0,35	79,5%	74,5%	69,5%	56%
0,55	0,15	85,5%	81%	76,5%	65%
0,55	0,20	84%	79,5%	75%	63%
0,55	0,25	82,5%	78%	73,5%	61%
0,55	0,30	81%	76,5%	72%	59%
0,55	0,35	79,5%	75%	70,5%	57%
0,60	0,15	85,5%	81,5%	77,5%	66%
0,60	0,20	84%	80%	76%	64%
0,60	0,25	82,5%	78,5%	74,5%	62%
0,60	0,30	81%	77%	73%	60%
0,60	0,35	79,5%	75,5%	71,5%	58%
MIN		79,5%	73,5%	67,5%	54%
MAX		85,5%	81,5%	77,5%	66%

Tabella 5.22: Analisi di sensitività per mangiarebene.com

Come era prevedibile, man mano che la ricetta utente cambia sempre più rispetto alla versione wellness, i valori di similarità tendono ad abbassarsi. Non solo, l'algoritmo risulta poco sensitivo al variare dei due fattori solamente se la ricetta è abbastanza simile alla ricetta wellness; nel caso 1 infatti l'escursione che si può registrare tra il valore massimo e il valore minimo, non va oltre il 6%. Quando invece la ricetta comincia a differire in maniera sostanziale si osserva che l'algoritmo diventa particolarmente sensibile. Mentre nel caso migliore c'è un'escursione massima del 6%, in quello peggiore, il

caso 4, vi è una differenza del 12%.

5.2 Millefoglie di melanzane

L'altra ricetta su cui si è testato il software realizzato è, come già preannunciato, la millefoglie di melanzane. Anche per questa ricetta sono state cercate ed utilizzate varie versioni presenti on-line, come riportato dalla Tabella 5.23.

giallozafferano.it	academiabarilla.it	cookaround.com	rossella
Ingrediente	Ingrediente	Ingrediente	Ingrediente
lasagne	melanzane	melanzane	melanzane
melanzane	basilico	mozzarella	pasta sfoglia
scamorza	salsa di pomodoro	pomodori	mozzarella
pomodori	scamorza	basilico	pomodori
origano	olio extra vergine di oliva	farina	grana
farina	parmigiano	olio extra vergine di oliva	olio
aglio	sale	pangrattato	sale
rucola		olio	basilico
peperoncino		uovo	
sale			
pepe			
olio extra vergine di oliva			
grana			

Tabella 5.23: ricette millefoglie di melanzane prese in esame

Si riporta ora invece la ricetta wellness nella tabella seguente:

wellness
Ingrediente
melanzane
pasta fillo
pomodori
pepe rimbas
aglio
limone
sale
olio extra vergine di oliva
basilico

Tabella 5.24: ricetta millefoglie di melanzane wellness

5.2.1 Similarità semantica

Per ogni versione di ricetta, come nel caso delle linguine al pesto, si presenta il risultato di similarità semantica.

In Tabella 5.25 è riportata la prima ricetta, millefoglie di melanzane di giallozafferano.it, con le sue versione modificate:

	giallozafferano.it 1	giallozafferano 2	giallozafferano 3	wellness
	Ingrediente	Ingrediente	Ingrediente	Ingrediente
	lasagne	pasta fillo	lasagne	pasta fillo
	melanzane	melanzane	melanzane	melanzane
	pomodori	pomodori	pomodori	pomodori
	sale	sale	sale	sale
	pepe	pepe	pepe	pepe rimbas
	aglio	aglio	aglio	aglio
	olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva
	basilico	basilico	origano	basilico
	farina	farina	farina	
	peperoncino	peperoncino	peperoncino	
	scamorza	scamorza		
	origano	origano		
	grana	grana		
				limone
Similarità semantica	61,11%	72,22%	69,4%	

Tabella 5.25: Similarità semantica giallozafferano.it

I valori di similarità semantica di questa ricetta, in riferimento alla prima versione, sembrano veritieri in quanto l'utente inserisce non pochi ingredienti che non sono presenti nella ricetta di riferimento Wellness. La seconda versione mantiene gli stessi ingredienti, andando a modificare quelli che erano considerati come “fratelli”, appartenendo alla stessa classe; migliorata la ricetta si può notare come il valore di similarità semantica cresce, come cresce, ripetuto alla prima versione, il valore della terza ed ultima versione: si eliminano gli ingredienti non contenuti nella ricetta Wellness per alzare la similarità, ma si peggiora inserendo ingredienti della stessa classe anziché identici.

In Tabella 5.26 è riportata la seconda ricetta, millefoglie di melanzane di academiabarilla.it, con le relative modifiche:

academiabarilla.it 1	academiabarilla 2	academiabarilla 3	wellness
Ingrediente	Ingrediente	Ingrediente	Ingrediente
melanzane	melanzane	melanzane	melanzane
basilico	basilico	origano	basilico
sale	sale	sale	sale
olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva
	pasta fillo	pasta fillo	pasta fillo
salsa di pomodoro	salsa di pomodoro		
scamorza			
parmigiano			
		pomodori	pomodori
		pepe	pepe rimbas
			aglio
			limone
Similarità semantica	22,22%	41,67%	66,67%

Tabella 5.26: Similarità semantica academiabarilla.it

Il valore di similarità semantica della versione originale della ricetta è bassissimo: solo quattro ingredienti coincidono con quella Wellness di riferimento e, sia da una parte che dall'altra non vi si verificano riscontri positivi. Con le altre due versioni si dimostra che è possibile migliorare la ricetta, cambiando alcuni ingredienti: la seconda versione è più simile, per similarità semantica, in quanto non si considerano ingredienti come scamorza e parmi-

giano e si aggiunge la pasta fillo. La terza ed ultima versione è migliorata e vi è l'aggiunta di un ingrediente, il pepe, considerato come un alimento appartenente alla stessa classe dell'ingrediente a cui corrisponde nella ricetta Wellness: la similarità cresce.

In Tabella 5.27 è riportata la terza ricetta, millefoglie di melanzane rossella (altra versione di giallozafferano.it):

rossella 1	rossella 2	rossella 3	wellness
Ingrediente	Ingrediente	Ingrediente	Ingrediente
melanzane	melanzane	melanzane	melanzane
pasta sfoglia	pasta sfoglia	pasta sfoglia	pasta fillo
pomodori	pomodori	pomodori	pomodori
basilico	basilico	basilico	basilico
olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva
sale	sale	sale	sale
mozzarella	mozzarella		
grana	grana	grana	
	limone	limone	limone
	pepe	pepe	pepe rimbass
			aglio
Similarità semantica	47,22%	75%	77,78%

Tabella 5.27: Similarità semantica rossella

Anche quest'ultimo caso mette in luce il funzionamento dell'algoritmo: partendo da una ricetta simile all'incirca per la metà degli ingredienti si cerca di migliorarla, verificando un aumento del valore di similarità semantica, andando ad eliminare e/o aggiungere ingredienti, ripettando le regole semantiche imposte.

5.2.2 Similarità per ingredienti

Si riportano in Tabella 5.28 le ricette complete, con gli ingredienti e le relative quantità per poter calcolare la similarità quantitativa.

giallozafferano.it		academiabarilla.it		cookaround.org		rossella	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
lasagne	200g	melanzane	2	melanzane	150g	melanzane	150g
melanzane	500g	basilico	1 mazzetto	mozzarella	1	pasta sfoglia	1 rotolo
scamorza	300g	salsa di pomodoro	200g	pomodori	2	mozzarella	220g
pomodori	500g	scamorza	150g	basilico	2 foglie	pomodori	1
origano	qb	olio evo	qb	farina	80g	grana	3 cucchiari
farina	qb	parmigiano	qb	olio evo	8 cucchiari	olio	qb
aglio	qb	sale	qb	pangrattato	3g	sale	qb
rucola	qb			olio	qb	basilico	qb
peperoncino	qb			uovo	1		
sale	qb						
pepe	qb						
olio evo	qb						

Tabella 5.28: Ricette millefoglie di melanzane prese in esame

In Tabella 5.29 viene riportata la ricetta millefoglie di melanzane wellness:

wellness	
Ingrediente	Quantità
melanzane	500g
pasta fillo	120g
pomodori	120g
pepe rimbas	qb
aglio	1 spicchio
limone	50g
sale	qb
olio extra vergine di oliva	10g
basilico	10g

Tabella 5.29: Ricetta millefoglie di melanzane wellness

Cominciamo con i risultati di similarità per giallozafferano.it, qui di seguito le due ricette a confronto:

giallozafferano.it		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
lasagne	200g	pasta fillo	120g
melanzane	500g	melanzane	500g
pomodori	500g	pomodori	120g
sale	qb	sale	qb
pepe	qb	pepe rimbas	qb
aglio	qb	aglio	1 spicchio
rucola	qb	basilico	10g
olio extra vergine di oliva	qb	olio extra vergine di oliva	10g
farina	qb		
peperoncino	qb		
scamorza	300g		
origano	qb		
grana	qb		
		limone	50g

Tabella 5.30: Confronto ricette wellness, giallozafferano.it

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		61,66%
si		57,44%
no	melanzane	73,98%
si	melanzane	62,34%
no	melanzane,pomodori	40,35%
si	melanzane,pomodori	70,59%
no	melanzane,pomodori,aglio	40,36%
si	melanzane,pomodori,aglio	69,25%

Tabella 5.31: Similarità quantitativa giallozafferano.it

Le due ricette presentate sono tutto sommato abbastanza diverse. I risultati in Tabella 5.31 mostrano una variazione della similarità che oscilla tra il 40% e il 70%. La ricetta utente riporta molti ingredienti che non sono presenti nella wellness; fra questi la scamorza la cui quantità, a differenza degli altri è più che consistente. Questo certamente incide nei calcoli di similarità ed in particolar modo laddove non è stata effettuata la normalizzazione. Se infatti andassimo ad osservare solamente i valori con normalizzazione i risultati sono evidentemente più stabili.

Confronto e risultati di similarità per academiabarilla.it:

academiabarilla.it		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
melanzane	2	melanzane	500g
basilico	1 mazzetto	basilico	10g
sale	qb	sale	qb
olio extra vergine di oliva	qb	olio extra vergine di oliva	10g
parmigiano	qb		
salsa di pomodoro	200g		
scamorza	150g		
		pasta fillo	120g
		pomodori	120g
		aglio	1 spicchio
		pepe rimbas	qb
		limone	50g

Tabella 5.32: Confronto ricetta wellness,academiabarilla.it

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		21,96%
si		13,39%
no	melanzane	7,32%
si	melanzane	43,77%
no	melanzane,olio extra vergine di oliva	8,05%
si	melanzane,olio extra vergine di oliva	45,74%
no	melanzane,olio extra vergine di oliva,sale	4,71%
si	melanzane,olio extra vergine di oliva,sale	48,04%

Tabella 5.33: Similarità quantitativa academiabarilla.it

Anche in questo caso le ricette sono molto diverse fra loro. A differenza del caso precedente gli ingredienti per i quali non vi è alcuna corrispondenza hanno mediamente quantità molto alte. Questo significa che in fase di penalizzazione la similarità viene abbassata di molto e a conferma di quanto detto per la Tabella 5.31 a risentirne sono soprattutto gli algoritmi all'interno dei quali non è stata implementata la normalizzazione dei pesi per le quantità.

Confronto e risultati di similarità per cookaround.org:

cookaround.org		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
melanzane	150g	melanzane	500g
olio extra vergine di oliva	8 cucchiari	olio extra vergine di oliva	10g
pomodori	2	pomodori	120g
basilico	2 foglie	basilico	10g
farina	80g		
mozzarella	1		
pangrattato	3g		
olio	qb		
uovo	1		
		pasta fillo	120g
		sale	qb
		limone	50g
		pepe rimbas	qb
		aglio	1 spicchio

Tabella 5.34: Confronto ricetta wellness, cookaround.org

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		42,16%
si		35,81%
no	melanzane	19,78%
si	melanzane	55,26%
no	melanzane,pomodori	12,74%
si	melanzane,pomodori	58,86%
no	melanzane,pomodori,aglio	5,83%
si	melanzane,pomodori,aglio	59,34%

Tabella 5.35: Similarità quantitativa cookaround.org

Ennesimo caso di ricette molto diverse per numero di ingredienti e per quantità. Se però andiamo a considerare le similarità per le quali vi è normalizzazione, i risultati, soprattutto con indicazione di ingredienti principali, sono decisamente stabili. Nel primo caso, in cui vengono indicate le melanzane, la similarità si attesta attorno al 55%; quando invece vengono indicati più ingredienti principali, la similarità cresce in funzione del peso dell'ingrediente aggiunto. Ad esempio nel caso dell'aglio, che pesa relativamente poco, vi è un aumento minimo: dal 58,86% al 59,34%.

Confronto e risultati di similarità per rossella (altra versione di giallozafferano.it):

rossella		wellness	
Ingrediente	Quantità	Ingrediente	Quantità
melanzane	150g	melanzane	500g
pasta sfoglia	1 rotolo	pasta fillo	120g
pomodori	1	pomodori	120g
basilico	qb	basilico	10g
olio extra vergine di oliva	qb	olio extra vergine di oliva	10g
sale	qb	sale	qb
mozzarella	220g		
grana	3 cucchiari		
		limone	50g
		pepe rimbas	qb
		aglio	1 spicchio

Tabella 5.36: Confronto ricetta wellness, rossella

Normalizzazione	Ingredienti principali	Similarità quantitativa
no		95,8%
si		95,33%
no	melanzane	68,26%
si	melanzane	87,78%
no	melanzane,pomodori	59,44%
si	melanzane,pomodori	82,99%
no	melanzane,pomodori,pasta sfoglia	64,53%
si	melanzane,pomodori,pasta sfoglia	77,32%

Tabella 5.37: Similarità quantitativa rossella

L'ultimo caso presenta invece una situazione differente. Le due ricette hanno pochi ingredienti diversi e salvo che per la mozzarella e il limone le quantità sono decisamente basse. Questo porta a valori di similarità relativamente alti, e in situazioni come questa anche gli algoritmi senza normalizzazione forniscono dati tutto sommato consistenti.

5.2.3 Similarità per nutrienti

Procediamo con l'analisi di similarità per gli elementi nutritivi. In Tabella 5.41 le ricette utilizzate:

giallozafferano.it		academiabarilla.it		rossella	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
lasagne	200g	melanzane	2	melanzane	150g
melanzane	500g	basilico	1 mazzetto	pasta sfoglia	1 rotolo
scamorza	300g	salsa di pomodoro	200g	mozzarella	220g
pomodori	500g	scamorza	150g	pomodori	1
origano	qb	olio extra vergine di oliva	qb	grana	3 cucchiari
farina	qb	parmigiano	qb	olio	qb
aglio	qb	sale	qb	sale	qb
rucola	qb			basilico	qb
peperoncino	qb				
sale	qb				
pepe	qb				
Carboidrati	340,3g		270,05g		231.68g
Lipidi	51,3g		35,6g		247.4g
Proteine	92,55g		44,05g		91.05g
Calorie	2057,13Kcal		1493,82Kcal		3463.459Kcal

Tabella 5.38: Valori aggregati di macronutrienti per le varie ricette

Mentre in Tabella 5.42 la ricetta Wellness:

wellness	
Ingrediente	Quantità
melanzane	500g
pasta fillo	120g
pomodori	120g
pepe rimbas	qb
aglio	1 spicchio
limone	50g
sale	qb
olio extra vergine di oliva	10g
basilico	10g
Carboidrati	107,49g
Lipidi	11,29g
Proteine	18,8g
Calorie	571,73Kcal

Tabella 5.39: Valori aggregati di macronutrienti per la ricetta wellness

Come fatto per le linguine al pesto vengono confrontate tre versioni per ogni ricetta; la prima è giallozafferano.it:

giallozafferano.it 1		giallozafferano.it 2		giallozafferano.it 3		wellness	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
lasagne	200g	lasagne	100g	lasagne	200g	pasta fillo	120g
melanzane	500g	melanzane	500g	melanzane	500g	melanzane	500g
pomodori	500g	pomodori	100g	pomodori	120g	pomodori	120g
sale	qb	sale	qb	sale	qb	sale	qb
pepe	qb	pepe	qb	pepe	qb	pepe rimbas	qb
aglio	qb	aglio	qb	aglio	qb	aglio	1 spicchio
rucola	qb	rucola	qb	rucola	qb	basilico	10g
farina	qb	farina	qb	farina	qb		
peperoncino	qb	peperoncino	qb	peperoncino	qb		
scamorza	300g	scamorza	300g	scamorza	90g		
origano	qb	origano	qb	origano	qb		
						olio extra vergine di oliva	10g
						limone	50g
Carboidrati	187,66g		96,66g		174,16g		107,49g
Lipidi	81,04g		80,05g		28,55g		11,29g
Proteine	106,97g		92,98g		50,68g		18,8g
Calorie	1798Kcal		1400,01Kcal		1084,43Kcal		571,73Kca
Similarità per nutrienti	28,48%		46,91%		70,11%		

Tabella 5.40: Similarità per nutrienti giallozafferano.it

Il fatto di avere una ricetta (prima versione) che ha circa il triplo delle calorie rispetto alla Wellness giustifica una similarità relativamente bassa: 28,48%. Nella seconda versione si è provato ad abbassare le quantità di alcuni ingredienti in modo da renderle più simili alla Wellness e in effetti l'algoritmo ha risposto con un aumento di circa il 20% della similarità. La maggior parte degli ingredienti utente non presenti nella Wellness hanno quantità decisamente bassa, l'unica eccezione è rappresentata dalla scamorza. Si è cercato infatti di andare ad abbassare notevolmente la sua quantità in modo tale da vedere la risposta dell'algoritmo. Un risultato del 70,11% può essere ritenuto ragionevole considerando il drastico abbassamento di lipidi e proteine.

In Tabella 5.44 i risultati per academiabarilla.it:

academiabarilla.it 1		academiabarilla.it 2		academiabarilla.it 3		wellness	
Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
melanzane	2	melanzane	3	melanzane	2	melanzane	500g
basilico	1 mazzetto	basilico	1 mazzetto	basilico	1 mazzetto	basilico	10g
sale	qb	sale	qb	sale	qb	sale	qb
olio evo	qb	olio evo	qb	olio evo	qb	olio evo	10g
parmigiano	qb	parmigiano	qb	parmigiano	qb		
salsa pom.	200g	salsa pom.	80g	salsa pom.	200g		
scamorza	150g	scamorza	150g	scamorza	150g		
				pasta fillo	120g	pasta fillo	120g
						pomodori	120g
						aglio	1 spicchio
						pepe rimbas	qb
						limone	50g
Carboidrati	10,75g		8,65g		105,55g		107,49g
Lipidi	41,34g		41,34g		42,54g		11,29g
Proteine	43,64g		43,94		55,64g		18,8g
Calorie	560Kcal		553Kcal		969,2Kcal		571,73Kcal
Similarità per nutrienti	57,78%		57,26%		75,97%		

Tabella 5.41: Similarità per nutrienti academiabarilla.it

Pur essendo un primo, la versione originale di academiabarilla.it non riporta alcun tipo di pasta fra gli ingredienti. Il numero dei carboidrati totali ne risente parecchio e la similarità non va oltre il 60%. Si è fatto un test cercando di aumentare le melanzane e diminuire la salsa da pomodoro, ma sia macronutrienti che calorie sono rimaste tutto sommato invariate. Nel terzo ed ultimo test si è proposta una versione con pasta, questo ha fatto sì che i carboidrati salissero notevolmente e per quanto le calorie siano aumentate notevolmente rispetto alla versione wellness, la similarità è salita a circa il 76%.

In conclusione rossella (altra versione di giallozafferano.it):

	rossella 1		rossella 2		rossella 3		wellness	
	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità	Ingrediente	Quantità
	melanzane	150g	melanzane	200g	melanzane	150g	melanzane	500g
	pasta sfoglia	100g	pasta sfoglia	100g	pasta sfoglia	120g	pasta fillo	120g
	pomodori	1	pomodori	1	pomodori	1	pomodori	120g
	basilico	qb	basilico	qb	basilico	qb	basilico	10g
	olio evo	qb	olio evo	qb	olio evo	qb	olio evo	10g
	sale	qb	sale	qb	sale	qb	sale	qb
	mozzarella	220g	mozzarella	120g	-			
	grana	3 cucchiaini	grana	3 cucchiaini	grana	3 cucchiaini		
							limone	50g
							pepe rimbas	qb
							aglio	1 spicchio
Carboidrati		47,68g		48,18g		56,88g		107,49g
Lipidi		95,4g		76,4g		61,2g		11,29g
Proteine		63,05g		45,55g		24,85g		18,8g
Calorie		1263,86Kcal		1034,81Kcal		862,34Kcal		571,73Kcal
Similarità per nutrienti		33,15%		46,72%		60,63%		

Tabella 5.42: Similarità per nutrienti rossella

La similarità della prima versione è relativamente bassa. Il fatto di avere la mozzarella fra gli ingredienti incide notevolmente sui lipidi e quindi sulle calorie totali. Nelle altre due versioni si è cercato rispettivamente di diminuirne il peso ed eliminarla completamente. Come conseguenza si è ottenuto un abbassamento delle calorie e indirettamente un miglioramento della similarità.

5.2.4 Analisi di sensitività

L'analisi di sensitività è stata svolta, come per il caso della ricetta delle linguine al pesto, facendo unicamente riferimento alla similarità semantica. In questo contesto si è utilizzando la ricetta proposta da giallozafferano.it. Si ricorda che i valori di similarità semantica dipendono da due fattori fondamentali: la percentuale del peso che viene aggiunta quando ci sono ingredienti diversi ma con padri uguali e la percentuale del peso che viene sottratta in caso di ingredienti completamente diversi. Nell'esecuzione della similarità semantica, questi due valori corrispondono rispettivamente al 50% e al 25%.

In Tabella 5.43 è possibile osservare la ricetta Wellness millefoglie di melanzane ed altre tre versioni della ricetta utente modificate appositamente per variare le corrispondenze con gli ingredienti, sfruttando così, secondo diverse possibilità, l'informazione semantica esistente.

giallozafferano.it	giallozafferano.it 2	giallozafferano.it 3	giallozafferano.it 4	wellness
Ingrediente	Ingrediente	Ingrediente	Ingrediente	Ingrediente
lasagne	lasagne	pasta fillo	pasta fillo	pasta fillo
melanzane	melanzane	melanzane	melanzane	melanzane
pomodori	pomodori	pomodori	pomodori	pomodori
sale	sale	sale	sale	sale
olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva	olio extra vergine di oliva
basilico	rosmarino	basilico	rosmarino	basilico
pepe	pepe	pepe	pepe	pepe rimbas
aglio	cipolla	aglio	cipolla	aglio
				limone
farina	farina	farina	farina	
peperoncino	peperoncino	peperoncino	peperoncino	
scamorza	scamorza	scamorza	scamorza	
grana	grana	grana	grana	

Tabella 5.43: Versioni modificate di giallozafferano a confronto

I dati riportati in Tabella 5.44 mostrano i valori di similarità semantica a fronte di un'escursione massima del 20%, con passo 5%, dei due fattori:

padri uguali	no corrisp	Similarità1	Similarità2	Similarità3	Similarità4
0,40	0,15	72,22%	58,89%	78,89%	65,56%
0,40	0,20	68,89%	55,56%	75,56%	62,22%
0,40	0,25	65,56%	52,22%	72,22%	58,89%
0,40	0,30	62,22%	48,89%	68,89%	55,56%
0,40	0,35	58,89%	45,56%	65,56%	52,22%
0,45	0,15	72,78%	60,56%	78,89%	66,67%
0,45	0,20	69,44%	57,22%	75,56%	63,33%
0,45	0,25	66,11%	53,89%	72,22%	60%
0,45	0,30	62,78%	50,56%	68,89%	56,67%
0,45	0,35	59,44%	47,22%	65,56%	53,33%
0,50	0,15	73,33%	62,22%	78,89%	67,78%
0,50	0,20	70%	58,89%	75,56%	64,44%
0,50	0,25	66,67%	55,56%	72,22%	61,11%
0,50	0,30	63,33%	52,22%	68,89%	57,78%
0,50	0,35	60%	48,89%	65,56%	54,44%
0,55	0,15	73,89%	63,89%	78,89%	68,89%
0,55	0,20	75,56%	60,56%	75,56%	65,56%
0,55	0,25	67,22%	57,22%	72,22%	62,22%
0,55	0,30	63,89%	53,89%	68,89%	58,89%
0,55	0,35	60,56%	50,56%	65,56%	55,56%
0,60	0,15	74,44%	65,56%	78,89%	70%
0,60	0,20	71,11%	62,22%	75,56%	66,67%
0,60	0,25	67,78%	58,89%	72,22%	63,33%
0,60	0,30	64,44%	55,56%	68,89%	60%
0,60	0,35	61,11%	52,22%	65,56%	56,67%
MIN		58,89%	45,56%	65,56%	52,22%
MAX		75,56%	65,56%	78,89%	70%

Tabella 5.44: Analisi di sensitività per giallozafferano.it

In queste ricette, più che per l'analisi di sensitività riguardante le lingue al pesto, si ha una variazione media dei valori di similarità semantica in base alle variazioni dei pesi utilizzati. Per la prima versione abbiamo un'escursione del 16,67%, tale ricetta è stata modificata in modo tale che alcuni ingredienti (basilico e aglio) fossero sostituiti con ingredienti appartenenti alla stessa classe (il rosmarino, appartenente alla classe Erbe, e la cipolla, appartenente alla classe ortaggi a bulbo) e l'analisi di sensitività segnala una variazione dei valori del 20%. La terza versione, rispetto alla prima, ha al posto delle

lasagne la pasta fillo, come consiglia la ricetta Wellness. I valori risultanti dall'analisi mostrano una variazione del 13,33% rispetto a una variazione del 17,78% dell'ultima ricetta presa in considerazione in cui si cambiano gli ingredienti sempre rispettando la proprietà di appartenenza alla stessa classe. Attraverso questa analisi di sensitività si può verificare che più i valori di similarità semantica si abbassano, ovvero la ricetta dell'utente è poco simile a quella consigliata dalla Cucina Wellness, più le variazioni dei risultati dell'analisi effettuata subiscono scostamenti maggiori, entro i limiti però del 20%. Dunque l'algoritmo non si comporta in modo eccellente ma, venendo usato per valutare ricette simili, per noi può essere considerato ragionevole, andando bene per i nostri obiettivi.

Capitolo 6

Conclusioni

L'obiettivo di questa tesi, come ampiamente discusso e mostrato, è stato quello di progettare un'applicazione che valutasse, attraverso l'uso della semantica, la similarità fra ricette di cucina cercando di non essere un semplice ricettario ma una vera applicazione intelligente che supporti i genitori per una corretta educazione alimentare in ambito pre-scolastico. Per raggiungere questo proposito sono stati seguiti vari obiettivi secondari: definire ed implementare un'ontologia che fosse usata come supporto semantico dall'applicazione, modellizzare le regole principali della Cucina Wellness per poter offrire una sana e corretta dieta nutrizionale in virtù di una educazione alimentare che, come introdotto nella sezione apposita sull'obesità infantile, sta subendo un'evoluzione con conseguenti effetti negativi sulla crescita infantile. L'ausilio dell'ontologia ha permesso di plasmare una rappresentazione formale della classificazione degli alimenti in base ai principi nutritivi e alle loro molteplici funzioni permettendo così di generare e processare nuova conoscenza che risulta essere utile e favorevole per proporre delle linee guida per una corretta educazione alimentare. Risulta essere inoltre un ottimo framework per poter aggiungere altra informazione semantica che non riguarda solamente il confronto degli ingredienti ma anche che dia la possibilità di imparare ad usare le giuste materie prime, a dosare gli ingredienti e, soprattutto, a cucinare tali ingredienti per ottimizzare piatti ricchi di molecole bioattive, vitamine, antiossidanti e sali minerali importanti per il nostro be-

nessere. È invece risultato poco efficiente purtroppo l'uso del linguaggio OWL per poter dichiarare valori diversi, nello specifico le relative quantità degli ingredienti, di alcune proprietà, essendo queste dipendenti dalle risorse che le utilizzano, vincolando la scelta di adoperare un database incorporato che favorisce però la scalabilità del sistema, una delle proprietà fondamentali da rispettare durante la progettazione di un sistema informatico.

Osservando i risultati presentati nel quinto capitolo è possibile trarre delle conclusioni sugli algoritmi implementati. Per quello che riguarda la similarità semantica si può dire che in generale l'algoritmo tiene fede a quelle che sono le differenze fra le ricette sulle quali opera. Per variazioni minime della ricetta, l'algoritmo presenta variazioni minime di similarità. Un'ulteriore prova della bontà dell'algoritmo è fornita dall'analisi di sensitività. In questo caso sono state effettuate numerose prove al variare dei fattori di penalizzazione per ingredienti diversi e premiazione per ingredienti appartenenti alla stessa classe nell'ontologia. In generale, per ricette molto simili fra loro, l'algoritmo è poco sensitivo alla variazione di tali fattori. Invece, per ricette che sono molto diverse fra loro, l'analisi di sensitività ha mostrato che la similarità semantica può essere molto variabile. Pertanto un miglioramento dell'algoritmo in tal senso sarebbe auspicabile. Riguardo alla similarità per quantità degli ingredienti, sono stati presentati numerosi casi. Quelli che hanno sempre soddisfatto le aspettative sono quelli in cui è stata fatta la normalizzazione e in particolare quelli in cui sono stati indicati gli ingredienti principali. Il fatto di indicare tali ingredienti, permette all'algoritmo di calibrare meglio i pesi e quindi ottenere valori di similarità più consistenti con quelle che sono poi le reali differenze tra le ricette. Riguardo ai risultati per i quali non vi è stata normalizzazione, se le ricette sono molto diverse, e gli ingredienti diversi hanno anche quantità consistenti, i risultati proposti sono poco significativi. Infine la similarità per nutrienti. L'algoritmo restituisce valori che effettivamente rispecchiano le differenze a livello nutritivo delle ricette. Volendo trovare un punto di debolezza è possibile dire che in termini assoluti fornisce risultati abbastanza alti, ma paragonando più ricette poco differenti fra loro, l'algoritmo si comporta relativamente bene; per piccole variazioni di quantità, fornisce leggere differenze di similarità.

6.1 Sviluppi futuri

Da quanto emerso dalla ricerca sull'obesità infantile svolta nel secondo capitolo, la socializzazione fra i genitori può essere vista come una leva fondamentale per far fronte all'emergenza alimentare. Permettere ai genitori di conoscersi, confrontarsi, scambiarsi ricette e consigli, nonché fare tutto questo con uno spirito ludico e di collaborazione, può incidere non poco sul miglioramento delle abitudini alimentari dei propri figli. Gli smartphone e più in generale i device mobili rappresentano lo strumento migliore per andare incontro all'esigenza di fare rete: realizzare un'applicazione per tali dispositivi sembra la conseguenza naturale di quanto svolto fino ad ora. L'applicazione deve avere come target di utilizzatori i genitori, e come scopo quello di permettergli di creare reti sociali al fine di mitigare gli eventuali problemi alimentari dei bambini. In questa fase di analisi è necessario capire quali sono le funzionalità di cui gli utenti target potrebbero aver bisogno.

Qui di seguito viene proposto un elenco:

- Un elemento di base è rappresentato dalla chat. L'applicazione deve essere in grado di mettere in connessione tra loro i genitori ed eventualmente fare in modo che riescano e interloquire con l'asilo frequentato dal bambino.
- Una seconda funzionalità potrebbe essere quella di dare la possibilità agli utenti di organizzare eventi in casa propria. Sarà possibile fare gli inviti, proporre una ricetta da realizzare durante l'evento ("Biscotti al cacao con poche calorie", "hamburger di verdure",...), dare la possibilità di accettare/rifutare l'invito.... In questi eventi si immagina che genitori e bambini si riuniscano in casa di chi ha proposto l'evento e si cerchi di coinvolgere attivamente i bambini nella realizzazione della ricetta. Quindi potrebbe essere necessario dare la possibilità a chi propone l'evento di chiedere agli invitati di portare alcuni strumenti affinché tutti i bambini possano lavorare.
- Ogni utente potrebbe condividere la propria dieta: ovvero quello che mangia il proprio bambino al di fuori dell'asilo e creare una sorta di clas-

sifica dei migliori utenti. Si potrebbero fare varie classifiche in base a determinate cifre di merito. Alcuni esempi: la classifica dei tre migliori utenti che nell'ultimo mese hanno cucinato le ricette più aderenti alla cucina evolution, la classifica di coloro che per più giorni hanno cucinato ricette della suddetta cucina o ancora la classifica di coloro le cui ricette hanno il più basso apporto di calorie/zuccheri/grassi. Si assume che una sana competizione potrebbe incidere sulla motivazioni dei genitori nel portare i propri figli ad attenersi al regime alimentare della cucina evolution.

- Un utente potrebbe voler far conoscere agli altri una ricetta che ha provato e che vuole condividere perché la ritiene ottima in base a un certo fattore : costo / semplicità di preparazione / tempo di preparazione Dal proprio profilo deve poter scrivere la ricetta (o fare copy&paste da dove l'ha trovata) e selezionare gli utenti con cui vuole condividerla. Anche qui si potrebbe assegnare un punteggio ai profili che fanno più proposte, le cui ricette sono considerate le migliori in termini di calorie o le più gustose.
- Si potrebbe creare un ricettario a cui vi partecipano tutti gli utenti. La base del ricettario deve essere composta dalle ricette proprie della cucina evolution e, in aggiunta a queste, le ricette proposte dagli utenti. Su queste ricette l'utente può esprimere due forme di valutazione : attraverso una RatingBar a stelle o attraverso dei commenti testuali.
- Lo scopo primario dell'applicazione è quello di favorire l'utilizzo della cucina evolution nella dieta dei bambini. Il problema in effetti è quello dell'obesità infantile, ma un altro modo di combatterlo è legato all'attività fisica. Si potrebbero inserire delle funzionalità che permettono la condivisione di determinati risultati sportivi. Ad esempio il fatto di aver fatto una corsa o una qualsiasi attività agonistica. Probabilmente potrebbe aver maggior presa sui piccoli l'organizzazione di partite di calcetto, basket, pallavolo e fare in modo che l'app sia in grado di favorire l'organizzazione di tali eventi.

Bibliografia

S. Salvy, K. Haye, J. Bowker, R Hermans, Influence of peers and friend on children's and adolescents' eating and activity behaviors. *Physiology & Behavior* 106 (2012) 369–378

Kevin C. Harris MD, Lisa K. Kuramoto MSc, Michael Schulzer MD PhD, Jennifer E. Retallack MD, Effect of school-based physical activity interventions on body mass index in children: a meta-analysis, *CMAJ* MARCH 31, 2009 180(7).

Elise Desjardins and Anne L. Schwartz Collaborating To Combat Childhood Obesity, *Health Affairs*, 26, no.2 (2007):567-571

Joan C Han, Debbie A Lawlor, Sue Y S Kimm, Childhood obesity, May 6, 2010, *The Lancet Journal*, DOI:10.1016/ S0140- 6736(10)60171-7

Nidhi Gupta, Kashish Goel, Priyali Shah, and Anoop Misra ,Childhood Obesity in Developing Countries: Epidemiology, Determinants, and Prevention, *Endocrine Reviews*, February 2012, 33(1):48 –70

Manoj Sharma, Dietary Education in School-Based Childhood Obesity Prevention Programs, *Health Promotion and Education, Advances in Nutrition review* (2011),University of Cincinnati Faculty, Walden University, Cincinnati, OH 45221-0068

Flo Harrison, Andrew P. Jones, A framework for understanding school based physical environmental influences on childhood obesity, *Health and Place Journal* (2010),School of Environmental Sciences, University of East Anglia, Norwich, NR4 7JT, United Kingdom

Teiji Nakamura, The integration of school nutrition program into health promotion and prevention of lifestyle-related diseases in Japan, *Clinical Nutrition Journal* (2008), Kanagawa University of Human Services, Japan

Frances A. Carter, and Cynthia M. Bulik, *Childhood Obesity Prevention Programs: How Do They Affect Eating Pathology and Other Psychological Measures?* , The American Psychosomatic Society (2008)

Gruber KJ, Haldeman LA. Using the family to combat childhood and adult obesity. *Prev Chronic Dis* 2009;6(3):A106. http://www.cdc.gov/pcd/issues/2009/jul/08_0191.htm.

Gruber KJ, Haldeman LA. Using the family to combat childhood and adult obesity. *Prev Chronic Dis* 2009;6(3):A106. http://www.cdc.gov/pcd/issues/2009/jul/08_0191.htm. T. Berners-Lee, J. Hendler, O. Lassila, *The Semantic Web*, Scientific America, 2001

Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In *International Journal Human-Computer Studies*, Vol. 43, pp. 907-928.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991) in *Enabling Technology for Knowledge Sharing*. *AI Magazine*, 12(3) pp. 16-36

Rudi Studer and V. Richard Benjamins and Dieter Fensel in *Knowledge Engineering: Principles and Methods*. *Data Knowl. Eng.* 25(1-2): 161-197 (1998)

Borst WN. *Construction of Engineering Ontologies*. *Centre for Telematica and Information Technology* , University of Twente. Enschede, The Netherlands, 1997.

Aldo Mariani Costantini, Carlo Cannella, Gianni Tomassi. *Alimentazione e nutrizione umana*. Il pensiero scientifico Editore. 2006