MADiMAN

Multimedia Application for DIet Managment

Title	**Deliverable WP3 D-WP3-01**
Sub-title	Deliverable del WP3 riguardante il sistema di analisi del linguaggio delle ricette (NLU/IE component)
Document ID	D-WP3-01
Release #	000.07

Responsibilities

Role	Name	Company	E-mail	Date
Prepared by	Matteo Casu, Andrea Bolioli	Celi srl	casu@celi.it abolioli@celi.it	20141127
Controlled & Approved by				
Issued by		Celi srl		

MADiMAN project related info

Involved WPs	
Related Documents	
Superseded Documents	

Document Type	X	Administrative Technical
Document Status	X	Technical Report Deliverable
Document Classification	X	Restricted to MADiMAN partners Public

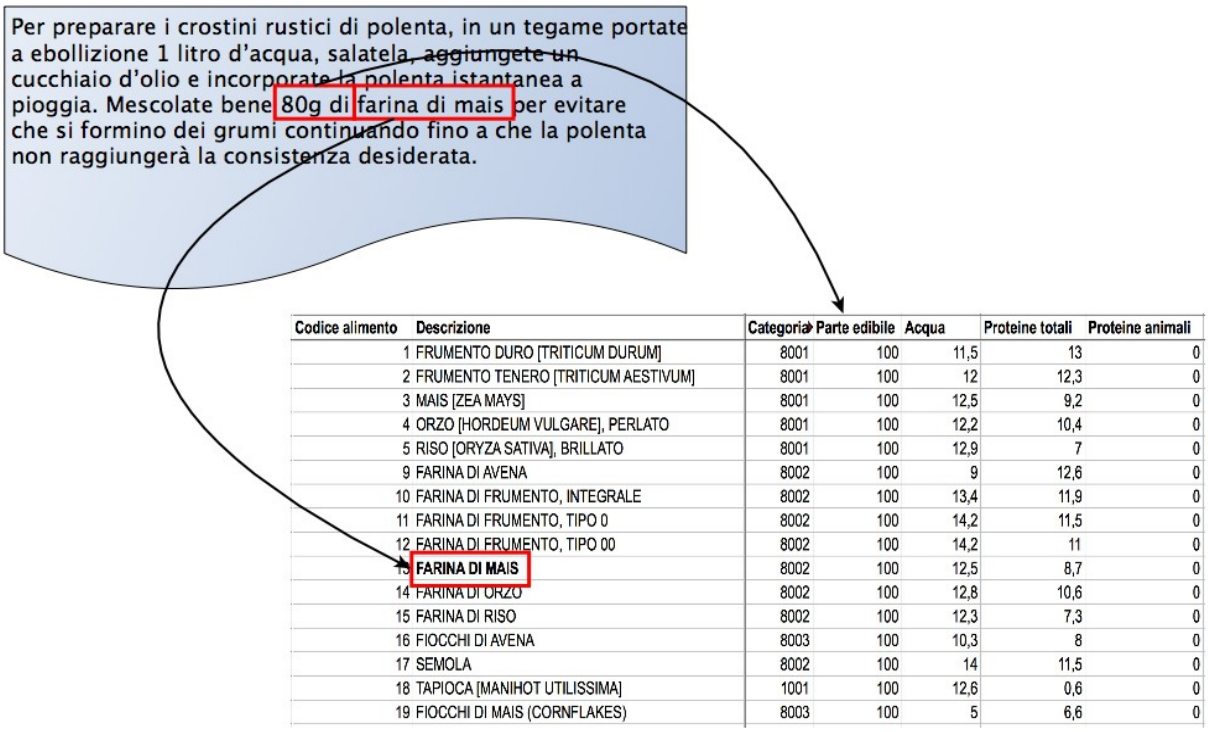
Revision history

Date	Release #	Modified Parts	Description of variations	Author
14/10/2013	000.01	Cap. 2		
31/10/2013	000.02	Cap. 1		
20/01/2014	000.03	Cap. 3 4 5 6		
02/02/2014	000.04	Cap. 7		
05/05/2014	000.05	Cap. 3 4 5 8		
10/06/2014	000.06	Cap. 6		
21/07/2014	000.07	Cap. 7		
11/09/2014	000.08	All		

Indice

1 Introduzione

Nel contesto del progetto Madimann il modulo di Natural Language Understanding (NLU)[1] e Information Extraction (IE)[2] ha la funzione di analizzare ricette, sia strutturate che scritte in linguaggio naturale, riconducendole a un formato strutturato. In particolare il modulo dovrebbe riconoscere gli alimenti, nella loro quantità specificata, la modalità di preparazione del cibo, e monitorare la presenza di certi alimenti o modalità di preparazione considerati “a rischio”. Il riconoscimento dell'alimento, con la sua quantità, ne permette il confronto con una banca dati dei principi nutritivi, per permettere il calcolo della quantità complessiva di principi nutritivi apportati da una certa ricetta (Figura 1). Nell'ambito del progetto Madiman una banca dati dei principi nutritivi è resa disponibile, e se ne parlerà in una sezione apposita.



Codice alimento	Descrizione	Categoria	Parte edibile	Acqua	Proteine totali	Proteine animali
1	FRUMENTO DURO [TRITICUM DURUM]	8001	100	11,5	13	0
2	FRUMENTO TENERO [TRITICUM AESTIVUM]	8001	100	12	12,3	0
3	MAIS [ZEA MAYS]	8001	100	12,5	9,2	0
4	ORZO [HORDEUM VULGARE], PERLATO	8001	100	12,2	10,4	0
5	RISO [ORYZA SATIVA], BRILLATO	8001	100	12,9	7	0
9	FARINA DI AVENA	8002	100	9	12,6	0
10	FARINA DI FRUMENTO, INTEGRALE	8002	100	13,4	11,9	0
11	FARINA DI FRUMENTO, TIPO 0	8002	100	14,2	11,5	0
12	FARINA DI FRUMENTO, TIPO 00	8002	100	14,2	11	0
13	FARINA DI MAIS	8002	100	12,5	8,7	0
14	FARINA DI ORZO	8002	100	12,8	10,6	0
15	FARINA DI RISO	8002	100	12,3	7,3	0
16	FIOCCHI DI AVENA	8003	100	10,3	8	0
17	SEMOLA	8002	100	14	11,5	0
18	TAPIOCA [MANIHOT UTILISSIMA]	1001	100	12,6	0,6	0
19	FIOCCHI DI MAIS (CORNFLAKES)	8003	100	5	6,6	0

Figura 1: riconoscimento di alimenti

A valle dell'analisi occorrerà predisporre un protocollo di comunicazione tra il presente modulo e il modulo di reasoning, che effettuerà il controllo dei vincoli sul regime alimentare.

2 Analisi linguistica

Nell'ambito dell'Information Extraction, che si occupa di task che vanno dal riconoscimento di entità (Named Entity Recognition, ossia il riconoscimento nel testo di occorrenze di luoghi, persone, organizzazioni) a task più difficili, quali l'estrazione di eventi o relazioni tra entità, il task qui affrontato si situa tra quelli di difficoltà media:

1Per una introduzione alle tematiche del NLU si rimanda ad esempio alla seguente tesi di dottorato (2007) disponibile online: http://www.oocities.org/iddolev/pulc/files/suthesis-iddolev-2side.pdf .

2Si veda ad esempio J. Cowie, Y. Wilks, "Information Extraction". CiteSeerX: 10.1.1.61.6480.

occorre infatti individuare entità (in questo caso alimenti) contestualizzandole all'interno di pattern indicanti espressioni di quantità.

In Information Extraction gli approcci più comunemente utilizzati ricadono in due grandi famiglie tipologiche: quella degli approcci basati su regole (che specificano pattern da estrarre) e quelli statistici, che permettono di apprendere strategie di estrazione di entità a partire da esempi – insiemi di documenti già annotati correttamente. Mentre la seconda famiglia di approcci garantisce di coprire una varietà di casistiche non previste, ma a condizione di avere un set di documenti pre-annotati abbastanza consistente, la prima famiglia può gestire solo casi previsti da chi compila le regole, ma non necessita di *corpora* annotati, e permette un controllo più fine sul risultato dell'estrazione.

Gli approcci a regole sono quindi indicati nei casi in cui il dominio di applicazione sia abbastanza ristretto – e questo è il caso del progetto Madiman, che tratta ricette con entità che possono variare su un insieme ampio ma ben definito: alimenti e quantità.

Seguendo i principi base del *Natural Language Processing*, ci riferiremo all'unità base di analisi come a un *documento* che viene analizzato da una *pipeline di analisi*, ossia da una sequenza di moduli software che agiscono in sequenza sul documento, emettendo su di esso annotazioni di tipo *stand-off*. Le annotazioni *stand-off* (che si oppongono alle annotazioni *inline*) sono annotazioni di porzioni di testo che vengono salvate separatamente dal testo, e che si legano a questo grazie a due attributi *start* e *end* indicanti il carattere iniziale e finale a cui l'annotazione si riferisce. Le annotazioni inline sono invece annotazioni di porzioni di testo fatte direttamente sul testo, ad esempio mediante tags XML. Negli anni recenti si sono diffuse largamente le piattaforme per l'annotazione linguistica stand-off come Apache UIMA (Untructured Information Management Architecture, creata da IBM) per superare gli svantaggi delle annotazioni inline (tra i quali ad es. l'impossibilità di annotazioni parzialmente sovrapposte).

Una tipica pipeline di analisi automatica agisce dopo l'acquisizione dei documenti, ed è individuata dalla seguente sequenza di azioni[3]:

- **tokenizzazione**: consiste nello "spezzare" il testo in unità lessicali o tokens (parole, segni di punteggiatura, sigle e abbreviazioni, numeri, date, ecc.);
- **normalizzazione**: espressioni scritte in modo non standard vengono normalizzate – ad esempio ripetizioni ridondanti di punteggiatura o di vocali vengono riportate ad una forma standard (es. “mooooolto beneee!!!!!” viene annotata come “molto" "bene" "!” più delle features di enfasi); l'espressione risultante è semanticamente equivalente a quella di partenza, ma può essere trattata tra le espressioni standard del linguaggio;
- **Segmentazione in frasi**: il testo viene "segmentato" in frasi, cioè vengono annotate delle sequenze di tokens come frasi.
- **Part-of-Speech (PoS) tagging**: ogni token viene riconosciuto come parte del discorso (articolo, nome, verbo, aggettivo, avverbio, preposizione, ecc) – questo processo richiede in genere di risolvere casi ambigui (i tokens “pesca” e "porta" ad es. possono giocare il ruolo di nome o di verbo). L'insieme dei tag con cui annotare i token è chiamato tagset, ed è deciso a priori. Uno dei più comuni per l'italiano è quello del progetto TUT: http://www.di.unito.it/~tutreeb/ ;
- **Lemmatizzazione**: la lemmatizzazione (o alternativamente la sua forma semplificata, lo *stemming*) è il processo con il quale si riconduce un token con PoS tag alla sua forma base, o *lemma* (anche questo processo può richiedere disambiguazione) – ad es. i due token “pesca” (nome singolare) e “pesche” (nome plurale) vengono entrambi riconosciuti come forme flesse del lemma “pesca” (nome). Lemmatizzazione e Part-of-Speech tagging possono avvenire contestualmente, oppure si possono utilizzare due annotatori in cascata per la Lemmatizzazione e la Disambiguazione.

3 Per una recente introduzione pratica al NLP si veda ad es. S. Bird, E. Loper, E. Klein (2009), *Natural Language Processing with Python*. O’Reilly Media.

Lo stemming consiste invece nel troncare la parte flessionale di una forma flessa – ad es. “pesche” viene ricondotto a “pesc”. La lemmatizzazione permette di poter effettuare ricerche sul testo che ignorino la flessione;

- **Disambiguazione semantica**: due occorrenze di uno stesso lemma possono denotare due concetti diversi: un esempio è il lemma *pesca* in "Amo la pesca d'altura." e "La pesca è un frutto estivo."

  La disambiguazione a livello semantico è affrontata in NLP con il nome di Word Sense Disambiguation (WSD), ed è classicamente affrontata con varianti dell'algoritmo di Lesk[4], che si basa sulla somiglianza tra la distribuzione di lemmi presenti nell'intorno del lemma considerato e quella presente in glosse e definizioni di una risorsa linguista usata: tipicamente WordNet[5], o sue varianti multilingua[6], e più recentemente, soprattutto nell'ambito del web semantico, tesauri multilingua espressi secondo il modello SKOS[7] (un esempio in ambito "agro" è Agrovoc[8], rilasciato dalla FAO)

- **Chunking**: anche chiamato parsing superficiale, il chunking consiste nel riconoscere nel testo pattern sintattici significativi, come ad es un chunk o sintagma verbale ("ho mangiato" o "sarà cotto").

Queste sono le azioni di solito considerate preparatorie a successive analisi linguistiche, come le analisi basate su regole che prendiamo in considerazione nell'ambito del progetto.

3 I documenti (ricette)

Prendendo in considerazione le ricette come documenti da analizzare, ne considereremo tre macrotipi:

1. ricette come documenti strutturati
2. ricette come documenti contenenti testo libero (ricette espresse in linguaggio naturale)
3. casi a tipologia mista: ricette in linguaggio naturale precedute da uno schema che riporti gli ingredienti

Dal punto di vista del progetto i tre casi suggeriscono tre modi diversi di acquisire i documenti e di trattarli. In particolare:

- i documenti di tipo 1 possono essere acquisiti con dei connettori software che importino in modo diretto i dati, mappandoli in un modello dati predefinito
- i documenti di tipo 2 sono invece acquisiti e in seguito analizzati con una pipeline di analisi – i dati estratti sono riconducibili allo stesso formato usato per i documenti di tipo 1
- i casi di tipo 3 vanno trattati con attenzione: in particolare lo schema degli ingredienti va riconosciuto in modo da estrarne i dati strutturati (o semistrutturati), per non incorrere in errori grossolani come ad esempio contare più volte lo stesso ingrediente

4 http://en.wikipedia.org/wiki/Lesk_algorithm

5 http://wordnet.princeton.edu

6 http://multiwordnet.fbk.eu/

7 http://www.w3.org/2004/02/skos/

8 http://aims.fao.org/agrovoc

4 Analisi delle ricette

Affrontiamo in questa sede il caso dell'analisi del testo, ipotizzando di trovarci nella situazione di documenti di tipo 2.

Per quanto riguarda le prime fasi dell'analisi (in particolare tokenizzazione e PoS tagging) l'NLP dispone di strumenti, per le varie lingue, ormai abbastanza affidabili e standard. La normalizzazione può essere invece effettuata con regole ad hoc per il dominio di interesse. La parte di analisi più specificamente problematica è invece quella della disambiguazione e del riconoscimento di chunk sintattici.

Per farsi un'idea delle possibili difficoltà che questo particolare dominio pone all'analisi linguistica automatica si possono prendere in considerazione esempi per certi aspetti opposti, quali una ricetta ricavata da un sito web[9] e una dell'*Artusi*[10]: come è noto il linguaggio del web è perlopiù paratattico (su modello anglosassone) e presenta un lessico ragionevolmente povero in termini di complessità lessicale[11]; il linguaggio di un libro come l'Artusi (edito nel secolo XIX) è invece caratterizzato da periodi più complessi, e in genere il lessico è abbastanza ricco, senza contare l'uso frequente di termini oggi desueti, di regionalismi, e la presenza di sezioni di commento da parte dell'autore –- che, sebbene interessantissime dal punto di vista storico/antropologico, portano rumore dal punto di vista di un'analisi automatica del linguaggio.

Per il primo tipo (sito web) prendiamo un esempio tratto da *Giallo Zafferano*. Notiamo come la ricetta sia strutturata con: titolo, un'introduzione, gli ingredienti (in forma semi-strutturata), la descrizione della preparazione.

Fritto misto di verdure e formaggi

Il fritto misto di verdure è un piatto invitante e genuino preparato con diverse verdure, passate in pastella e fritte, abbinate a sfiziosi bocconcini di formaggio impanati. Può essere un gustoso antipasto, un contorno appetitoso oppure un secondo piatto sostanzioso che incontrerà anche i gusti dei palati più diffcili. A seconda della stagione potrete utilizzare diverse verdure: dal carciofo al cavolo, dalle zucchine alle melanzane. Un nuovo modo per far mangiare le verdure ai vostri bambini.

Ingredienti:

Cavolfiore 1

Broccoli 1

Carciofi 4

Patate 3

Zucchine 3

9Ad esempio il sito http://www.giallozafferano.it .

10P. Artusi, La Scienza in cucina e l'Arte di mangiar bene, Landi 1891.

11La complessità lessicale su un testo è definibile in modo quantitativo come il rapporto tra il numero di lemmi e il numero di token nel testo.

Sale q.b.

Pepe q.b.

Ricotta fresca 250 gr

Caciotta romana 50 gr

Formaggio fresco

provolone 50 gr

Noce moscata q.b.

Farina 70 gr

Olio di semi q.b. per friggere

Limoni mezzo

per la pastella:

Farina 370 gr

Acqua 550 gr

Sale 1 pizzico

per la panatura:

Pangrattato q.b.

Farina q.b.

Uova 3 tuorli

Sale q.b.

Pepe q.b.

Preparazione

Per preparare il fritto misto di verdure e formaggi iniziate predisponendo le verdure: pulite e tagliate a bocconcini il cavolfiore e il broccolo (1), cuocete in acqua bollente i broccoli per 3 minuti (2) e il cavolfiore per 5 minuti, devono rimanere croccanti. Tagliate le patate a fettine sottili (3) e cuocetele per 1 minuto in acqua bollente salata. Pulite ora i carciofi seguendo le indicazioni che trovate qui[12], tagliateli in quarti e poneteli in una ciotola con acqua e limone per evitare che anneriscano (4). Lavate le zucchine e ricavatene delle fettine sottili (5). A questo punto preparate i formaggi: in una capiente ciotola ponete la ricotta, le fettine di caciotta tagliate finemente e 2 cucchiai di provolone grattugiato, farina, sale, pepe, noce moscata (6) e mescolate tutto (7) per formare delle polpette di circa 15 gr ciascuna (8). Infine tagliate la restante parte di provolone a fettine (9). Predisponete il

12In questo punto del testo è incluso, nell'originale, un link a una pagina di istruzioni.

necessario per la panatura: in una ciotola sbattete i tuorli con il sale, e ponete su due ampi vassoi la farina e il pangrattato.

Iniziate con la panatura dei formaggi: passateli nella farina (10), poi nell'uovo (11) e infine nel pangrattato (12) e metteteli da parte.

Proseguite nello stesso modo con metà dei carciofi (13). Ora preparate la pastella: in una ciotola versate l'acqua e aggiungete a poco a poco la farina mescolando con una frusta in modo che non si formino grumi (14), lasciate riposare la pastella in frigorifero per circa mezz'ora. Tutto è pronto per la frittura: ponete sul fuoco un'ampia padella con abbondante olio di semi, non appena l'olio sarà caldo iniziate a friggere per prime le patatine (15) e poi lasciatele asciugare su carta assorbente.

Immergete nella pastella i bocconcini di cavolfiore (16), di broccolo, le zucchine e i restanti carciofi, friggeteli (17) e lasciateli asciugare. Terminate la frittura con i formaggi e i carciofi impanati. Abbiate cura di asciugare tutte le verdure e i formaggi su un foglio di carta assorbente per eliminare l'olio in eccesso (18). Servite subito la frittura di verdure e formaggi calda e croccante!

Osserviamo ora più da vicino una ricetta tratta dall'*Artusi*. Abbiamo scelto una ricetta che presenta un elenco di ingredienti iniziale (non tutte lo hanno), non troppo lunga, senza i commenti dell'autore che nel testo dell'Artusi sono comunque abbastanza frequenti. Notiamo le fantasiose espressioni per indicare quantità guidate dal buon senso, come "odore di noce moscata".

Tortellini all'italiana (agnellotti)

Braciuole di maiale nella lombata, circa grammi 300

Un cervello di agnello o mezzo di bestia più grossa

Midollo di bue, grammi 50

Parmigiano grattato, grammi 50 Rossi d'uovo n. 3 e, al bisogno, aggiungete una chiara

Odore di noce moscata

Disossate e digrassate le braciuole di maiale, e poi tiratele a cottura in una cazzaruola con burro, sale e una presina di pepe. In mancanza del maiale può servire il magro del petto di tacchino nella proporzione di grammi 200, cotto nella stessa maniera. Pestate o tritate finissima la carne con la lunetta; poi unite alla medesima il cervello lessato e spellato, il midollo crudo e tutti gli altri ingredienti, mescolandoli bene insieme. Quindi i tortellini si chiudono in una sfoglia come i cappelletti e si ripiegano nella stessa guisa, se non che questi si fanno assai più piccoli. Ecco, per norma, il loro disco.

Per quanto riguarda il **riconoscimento di pattern** riguardanti le quantità di alimenti possiamo individuare due problematiche particolarmente interessanti per la riuscita del task che ci siamo posti:

- normalizzazione delle unità di misura
- riconoscimento di pattern esprimenti quantità ed eliminazione delle ridondanze

La prima problematica riguarda il fatto che in diverse ricette (nonché nella stessa ricetta) possono essere usate diverse unità di misura --- tra queste, oltre a quelle di derivazione scientifica, abbiamo anche quantità la cui interpretazione è lasciata dall'autore alla comprensione del contesto, o al senso comune: si pensi quindi, tra gli esempi, a diverse unità (grammi, kilogrammi...), diversi modi di esprimere la stessa unità (*50g*, *50 grammi, mezzo chilo*…), espressioni contenenti quantità indeterminate (“un pizzico di”, “quanto basta”, “Q.B.”, “un pugno di”), accanto a espressioni di quantità indeterminate ma per le quali si può facilmente assumere un valore di default: ad esempio si potrà associare “un bicchiere di” a una certa quantità in centilitri. Caso analogo si ha con espressioni cardinali di quantità, per ingredienti interi (che non siano tipi naturali): ad esempio “un uovo” o “due trote”, senza indicarne il peso, sono espressioni analoghe a "un bicchiere".

Occorrerà quindi, al fine di permettere il calcolo dei principi nutritivi nella ricetta, ricondurre le varie quantità riconosciute a unità di misura standard, utilizzabili dagli altri componenti del sistema.

La seconda problematica da affrontare è la varietà con la quale la quantità si lega, nel linguaggio, all'alimento –- un problema che in parte si sovrappone al precedente, ma se ne discosta ampiamente nel caso del riconoscimento di coreferenze: si pensi infatti al caso in cui un ingrediente (inteso come un alimento legato a una quantità, ad esempio “un uovo”) venga introdotto a inizio ricetta, e poi richiamato più avanti: “si prenda ora l'uovo ...”.

Per quanto riguarda invece il **riconoscimento di alimenti** un problema è quello – a cui si è già accennato -- dei termini desueti, regionalismi, nomi scientifici, e più in generale delle varianti che si discostano dal termine più in uso nell'italiano standard. In questo caso torneranno utili liste di sinonimi, magari strutturate in tesauri o ontologie. Esempi usati in NLP e IR sono le basi dati lessicali generali come WordNet, tesauri di dominio come Agrovoc, ontologie e basi di conoscenza (come DBpedia). Da queste risorse è possibile estrarre i sinonimi che, nella pipeline di analisi, aiutino l'individuazione degli alimenti. Non è da escludere l'introduzione di un vocabolario ad hoc per il progetto, che tenga conto di termini desueti o colloquiali probabilmente non presenti nelle risorse citate. Da considerare inoltre il fatto che la banca dati dei principi nutritivi contiene già alcune sinonimie per gli alimenti listati, come ad esempio i nomi scientifici.

Nell'ambito del presente progetto si è provveduto a creare dei gazzetteer/lessici con informazioni estratte da varie fonti tra cui dbPedia italiana[13]. Diamo qui un esempio di query SPARQL, da effettuare sul dataset rilasciato da DBpedia stessa[14], per ottenere un elenco di cibi (gli elementi della categoria "portate di cucina" in DBpedia), eventualmente da controllare e "ripulire" a mano:

```
select distinct ?label

where {

?x rdfs:label ?label.

?x dcterms:subject ?subject.

?subject rdfs:label ?subjectLabel.

?subject skos:broader* <http://it.dbpedia.org/resource/Categoria:Portate_di_cucina>.

}
```

13 http://it.dbpedia.org

14 http://it.dbpedia.org/dati/

5 Banche dati degli alimenti e dei principi nutritivi

Sono oggi disponibili, anche liberamente scaricabili via web, diverse banche dati che listano alimenti assieme alle quantità dei principali principi nutritivi presenti in una certa “parte edibile” degli alimenti stessi. Non tutte sono strutturate nello stesso modo, così come non tutte presentano lo stesso grado di granularità nel descrivere gli alimenti o i loro principi nutritivi. In alcuni casi inoltre tali banche dati presentano utili sinonimie (si veda la sezione sull' “analisi linguistica”) non sempre però strutturate in modo direttamente utilizzabile da un software. Vediamo in questa sezione alcuni esempi di strutture di tali banche dati, compresa la banca dati usata per le analisi preliminari del progetto, fornita nell'ambito del progetto stesso.

Tra le dimensioni interessanti (ai fini dei nostri task) nella valutazione di una banca dati degli alimenti poniamo:

- numero di alimenti o loro varianti/presentazione, tenendo conto che in alcuni casi un alimento costituirà più entrate della banca dati, essendo disponibile in varianti dalle caratteristiche nutritive diverse –- si pensi al caso dei pomodori secchi o pelati in scatola;
- granularità degli alimenti stessi --- tenendo anche conto del punto precedente, si consideri che c'è una certa arbitrarietà nel selezionare cosa costituisca una singola entrata della base dati: ad esempio, “pizza con pomodoro e mozzarella” può comparire come singolo alimento in una base dati. In ogni caso, il modulo di estrazione dovrebbe privilegiare i “longest matches”, e quindi assumendo di avere tra gli elementi possibili “olive snocciolate” estrarre l'intera espressione “olive snocciolate” e non “olive”
- insieme di principi nutritivi --- anche in questo caso, la granularità non è la stessa per tutte le basi dati: in alcuni casi la quantità di grassi saturi sarà indicata come aggregato, altre volte le quantità di grassi in un alimento saranno specificate per tipo di grasso (vegetale, animale, etc.)

Si tenga presente che il task richiede di estrarre, accanto agli alimenti nelle rispettive quantità, anche le **modalità di preparazione**: come è prevedibile uno stesso alimento conterrà principi nutritivi diversi (oltre ad essere considerato o meno a rischio) a seconda della modalità di preparazione (ad esempio se bollito o fritto). È pur vero che questa informazione, in alcune basi dati, potrebbe essere codificata all'interno del tipo di alimento: troveremo infatti “manzo (bollito)” e “manzo (fritto)” tra le entrate della base dati. Tale casistica è simile al caso del tipo di confezionamento. Nel contesto del presente progetto "rilassiamo" il problema del riconoscimento delle modalità di preparazione dei vari alimenti al riconoscimento delle modalità di preparazione sul piatto nel suo insieme. Le modalità di preparazione saranno quindi modellizzate come una collezione di modalità.

Un'interessante estensione del concetto di rischio potrebbe derivare dal monitoraggio degli alimenti considerati a rischio per allergici, la cui presenza nei menu è peraltro normata da una direttiva europea del 2011, recepita in Italia dal 13 dicembre 2014. Gli alimenti rientranti sono i seguenti: cereali contenenti glutine, crostacei, uova, pesce, arachidi, soia, latte, lattosio, frutta a guscio (mandorle, nocciole, noci, pistacchi), sedano, senape, semi di sesamo, anidride solforosa e solfiti, lupini, molluschi.

Avendo specificato le dimensioni di interesse, vediamo come esempio di base dati quella fornita dal partner medico del progetto –- tenendo conto che nel contesto del progetto rivolgeremo la preferenza a una base dati che abbia una copertura ampia e accurata degli alimenti tipici della zona geografica a cui la versione italiana dell'applicazione si rivolge: alimenti che comunque non coincidono necessariamente con quelli della dieta mediterranea, essendosi ormai ampiamente diffuse abitudini alimentari tipiche dei paesi ad esempio anglosassoni.

La base dati in questione contiene circa 790 alimenti, con principi nutritivi rispondenti alle seguenti categorie: proteine (animali e vegetali), lipidi (animali e vegetali, saturi totali, monoinsaturi totali, polinsaturi totali), acido oleico, acido linoleico, colesterolo, glucidi, amido, fibra alimentare, alcool, energia (in kcal e in kj), ferro, calcio, sodio, potassio, fosforo, zinco, tiamina, riboflavina, niacina, vitamine (C, B6, E, D, acido folico, retinolo).

È utile citare già in questa sezione che per i fini del progetto sarà utile considerare di avere a disposizione anche una base di conoscenza con i pesi specifici per le conversioni in grammi delle quantità di liquidi (normalmente espressa in decilitri o millilitri -- o, in casi specifici, in altre unità di misura: si veda la Sezione 6).

Tra le basi di conoscenza recanti informazioni fisiche citiamo alcune risorse online:

- convertitore pesi/volumi per ingredienti: http://www.onlineconversion.com/weight_volume_cooking.htm
- il National Nutrient Database del Dipartimento dell'Agricoltura statunitense: http://ndb.nal.usda.gov

La base dati sarà condivisa e accessibile ai vari moduli software.

6 Prove di analisi

Per il presente studio si sono selezionate alcune ricette da diverse fonti web che potessero rappresentare tipologie paradigmatiche di ricette da analizzare, e che contenessero una certa varietà delle problematiche descritte in precedenza. Ecco l'elenco di alcune di esse:

- http://www.cucchiaio.it/ricette/ricetta-caponata-classica
- http://www.cucchiaio.it/ricette/ricetta-hamburger-chianina-terre-ditalia-pane-integrale-germogli-chips
- http://www.cucchiaio.it/ricette/ricetta-filetto-salmone-norvegese-fresco-griglia-peperoncini-pure-avocado-lime
- http://www.cucchiaio.it/ricette/ricetta-tiramisu
- http://it.wikibooks.org/wiki/Libro_di_cucina/Ricette/Caponata
- http://it.wikibooks.org/wiki/Libro_di_cucina/Ricette/Hamburger
- http://it.wikibooks.org/wiki/Libro_di_cucina/Ricette/Carpaccio_di_salmone_al_profumo_di_aneto
- http://it.wikibooks.org/wiki/Libro_di_cucina/Ricette/Tiramisù

Gli esempi riportati costituiscono quattro coppie di ricette; ogni coppia è costituita da due ricette analoghe, una presa da *Il Cucchiaio d'Argento*[15] e una da Wikibooks[16].

Le ricette in questione rappresentano esempi di documenti parzialmente strutturati, in cui la sezione di preparazione è facilmente separabile da quella degli ingredienti; tuttavia, la preparazione è espressa come testo libero.

Le ricette sono state analizzate con una pipeline linguistica di base di CELI (pipeline di annotatori UIMA descritta in precedenza), al fine di far emergere le problematiche più comuni tra quelle previste, o eventuali problematiche non previste. Le quantità sono state estratte con un sistema a regole (Jboss Drools[17], cfr. Sezione 8), usando semplici regole basate su pattern e su un gazzetteer di ingredienti estratto dalla banca dati degli ingredienti.

Riportiamo un esempio di regola condizione-azione Drools, che nel sistema di annotazione considerato va ad agire sulle annotazioni Uima:

```
rule "rule1"
when
  >$number:Lemma(lexCat=="NUMBER", $numberPosBegin:posBegin, $numberPosEnd:posEnd)
  >$grammiExpr:Lemma(normalForm in ("g", "gr"), posBegin==($numberPosEnd+1), $grammiPosBegin:posBegin)
  >$di:Lemma(normalForm in ("di"), posBegin==($grammiPosBegin+1), $diPosBegin:posBegin)
  >$ingrediente:NE(type == "ingredienti", posBegin==($diPosBegin+1), $nomeIngrediente:ingrediente)
then
  >Quantity newQuantity = new Quantity();
  >newQuantity.setType("ingredienti");
  >newQuantity.setName($nomeIngrediente);
  >newQuantity.setValue($number.getLemma());
  >newQuantity.setBegin($ingrediente.getBegin());
```

15 http://www.cucchiaio.it/

16 http://it.wikibooks.org

17 http://www.drools.org

```
        >newQuantity.setEnd($ingrediente.getEnd());
        >newQuantity.setPosBegin($ingrediente.getPosBegin());
        >newQuantity.setPosEnd($ingrediente.getPosEnd());
        Store newOpinion
    end
```

La regola intercetta le porzioni di testo che esprimono il seguente pattern:

<numero> ("g"|"gr") "di" <ingrediente>

ossia, un numero, seguito da "gr di" (o espressioni equivalenti), seguito da un ingrediente. La regola annota la porzione di testo con un'annotazione di tipo Quantity, che porta con sé l'ingrediente trovato e la quantità. Numeri e ingredienti sono estratti a monte dell'applicazione della regole come Named Entities usando altri componenti software, che sfruttano risorse (espressioni regolari, lessici, gazzetteers) e tecniche quali lemmatizzazione e disambiguazione; per gli ingredienti il gazzetteer usato è quello del database degli ingredienti citato in Sezione 5.

La sperimentazione ha evidenziato la pervasività di una problematica citata in Sezione 4: le quantificazione dei liquidi (normalmente espressa in decilitri), che obbliga a considerare una base di conoscenza con i pesi specifici per le conversioni in grammi delle quantità.

Analogo problema (anche questo previsto in Sezione 4) riguarda le espressioni vaghe (o contestuali) di quantità, tra cui *un cucchiaio di mandorle*, *un bicchiere d'acqua* etc. In realtà molti di questi casi ricadono in assunzioni che possono essere fatte a monte per le quantità di ingredienti: ad esempio, un cucchiaino contiene mediamente 5 grammi di ingrediente, un cucchiaio 15 grammi (e hai fini del tipo di dieta in questione si possono trascurare le differenze tra gli ingredienti per tali quantità), mentre un bicchiere di liquido equivale mediamente a 200 millilitri (per risalire ai grammi occorre considerare il liquido, come già discusso).

Va aggiunto che nel mondo anglosassone le quantità per bicchieri, tazze, e altri contenitori/misurini sono standardizzate (una descrizione abbastanza dettagliata è data dalla pagina internazionale Wikipedia delle misure per la cucina[18]), per cui i casi d'uso in cui le ricette seguano queste modalità di espressione sono facilitati.

7 Protocollo di interscambio

In linea di principio l'output del modulo di Information Extraction potrebbe essere costituito dalle occorrenze nel testo dei pattern identificati o direttamente dal conteggio dei principi nutritivi. Per motivi di modularità e separazione dei task assumiamo che l'API fornita dal modulo di Information Extraction fornisca solo gli alimenti riconosciuti nel testo con le loro quantità. Il calcolo dei nutrienti tramite l'uso della base dati è banale e viene delegato al modulo che si occupa di calcolare le quantità di nutrienti (assumiamo per semplicità che sia l'orchestratore).

Per quanto riguarda le **unità di misura**, l'ipotesi prevede di assumere un'unica unità di misura (ad esempio i grammi) e quindi di non esprimerla esplicitamente nell'output. Tuttavia ciò comporta una complicazione: per diversi alimenti l'unità di misura standard è costituita dai decilitri (si pensi a liquidi o salse di condimento, come l'aceto o il ketchup). In questi casi la massa in grammi è ricavabile dal peso specifico dell'alimento – assumendo di avere tale informazione nella banca dati usata (cfr. Sezione 5). Un problema diverso, ma di analoga soluzione (cfr. Sezione 6), si ha con le espressioni vaghe di quantità, come “3 cucchiai di” o “un bicchiere di”.

Il modulo di analisi linguistica calcolerà/ricondurrà a forma normale (in pesi) internamente le quantità degli ingredienti, restituendo un output indicante, per ogni alimento, la quantità espressa in grammi (o alternativamente il dato rilevato, con unità di misura e quantità in valore assoluto, permettendo poi di approntare, a valle dell'estrazione delle quantità, una fase di **normalizzazione dati** in cui le quantità sono ricondotte a masse tramite la conoscenza del peso specifico di ogni alimento – assumiamo però in questa descrizione la prima ipotesi). L'output intermedio, fornito dal modulo NLP, sarà quindi esprimibile in formato JSON:

```
{
   "id_ricetta":"ricetta1",
```

18 http://en.wikipedia.org/wiki/Cooking_weights_and_measures

```
    "ingredients":{
       "manzo":70,
       "patate_lesse":150
    },
    "modalità":[
       "fritto"
    ],
    "criticità":[
       "fritto"
    ]
}
```

L'output finale del processo sarà costituito da un insieme di principi nutritivi con il conteggio di quanto essi sono presenti nella ricetta. Nell'esempio vengono riportate una lista di modalità di preparazione riscontrate e una lista di criticità che in questo caso attesta la presenza di fritto. Si può infatti assumere che, oltre alle modalità di preparazione dei singoli alimenti, sia importante monitorare la presenza di certi **alimenti o modalità di preparazione considerati "a rischio"**, di cui si vuole tenere traccia a livello di ricetta (e quindi di singolo pasto). In generale si potrebbero rappresentare le criticità rappresentandone la presenza o assenza tramite variabili booleane, ma la strategia adottata appare preferibile in quanto più compatta.

Il contributo del modulo di Information Extraction e NLU sarà quindi di permettere al sistema di arrivare ad avere i nutrienti di una ricetta, facilmente serializzabili in un file XML o JSON, come da esempio seguente:

```
{
   "id_ricetta":"ricetta1",
   "lipidi":30,
   "proteine":20,
   "modalità":[
      "fritto"
   ],
   "criticità":[
      "fritto"
   ]
}
```

8 Tecnologie

Tra le tecnologie usate da CELI nella sua esperienza di analisi automatica di testo non strutturato (anche di stream in real-time) abbiamo citato, per quanto riguarda l'architettura di annotazione linguistica, Apache UIMA[19]. UIMA permette di impostare pipeline di analisi in cui ogni documento viene sovra-annotato da moduli software che agiscono in successione (gli annotatori software), ognuno sfruttando l'analisi dei suoi "collaboratori" precedenti.

All'interno dei singoli annotatori UIMA possono essere utilizzate tecnologie diverse (basate su regole o su machine learnig). Nel caso di sistemi a regole che permettano di riconoscere pattern sintattici e/o agire sulle

19http://uima.apache.org .

annotazioni precedenti, un motore particolarmente versatile (che utilizziamo in diversi annotatori) è Jboss Drools[20].

Per quanto riguarda le librerie atte ad accedere e manipolare dataset RDF (come quelli della Linked Open Data Cloud[21] – utili ad acquisire dati anche linguistici, come sinonimie per i termini occorrenti nei documenti) citiamo i due standard *de facto* open source disponibili: Apache Jena[22] e Sesame[23].

9 Architettura del modulo

Per dare una visione di insieme del contributo NLU/IE al resto del sistema (soprattutto al modulo di ragionamento automatico) rimandiamo alla Figura 2: l'Orchestrator comunica via API REST con il modulo NLU/IE, che funziona come una funzione da ricette a alimenti e quantità (secondo il data model espresso in JSON in sezione 7). L'Orchestrator sottomette quindi la ricetta al modulo di NLU/IE, che risponde con un JSON. Dagli alimenti e quantità trovati nella ricetta, l'Orchestrator può calcolare le quantità di nutrienti grazie al database a disposizione del sistema. A questo punto l'Orchestrator userà il modulo di ragionamento per capire se la ricetta soddisfa i vincoli per l'utente corrente (si rimanda ai rispettivi deliverable per i dettagli sul funzionamento dei moduli citati e dei loro formati di interscambio).

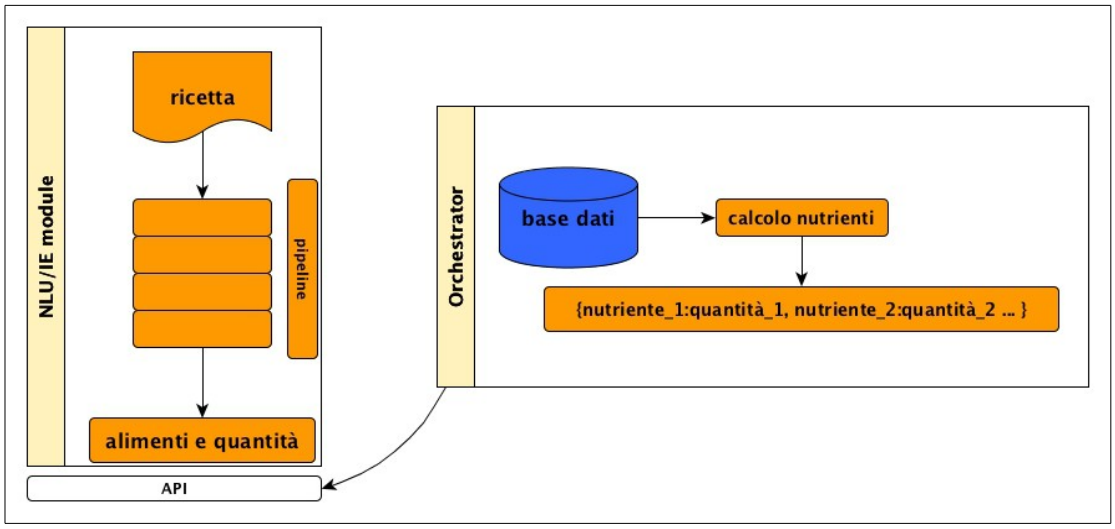


Figura 2

10 Stime per la realizzazione

Per la stima per la realizzazione del modulo descritto facciamo l'assunzione che le ricette siano costituite da testo non strutturato o semi-strutturato (comunque digitalizzato) e che siano costituite da un insieme

20 https://www.jboss.org/drools/ .

21 http://linkeddata.org .

22 https://jena.apache.org .

23 http://openrdf.callimachus.net .

predeterminato. La stima cambierebbe nel caso in cui le ricette debbano essere recuperate in modalità diverse, ad es.:

- dal web
- da basi dati non determinate a priori
- da archivi non digitali, da digitalizzare

Assumiamo anche che la lingua delle ricette sia l'italiano. È possibile affrontare la realizzazione del modulo per altre lingue con le modalità descritte, utilizzando specifiche risorse e specifici moduli di analisi linguistica.

La realizzazione del modulo includerebbe le seguenti macro-attività:

1. l'analisi e la raccolta di esempi di dati di input (le ricette) e delle risorse (database, gazzetteers, etc.) - il presente studio assolve in parte questo punto
2. sviluppo regole di analisi
3. predisposizione/configurazione della pipeline di analisi e sviluppo dei servizi web per l'analisi "live" (ossia, a ogni richiesta via API) delle ricette
4. installazione/rilascio in ambiente di test e di produzione
5. erogazione e manutenzione annua

Una stima ragionevole in giorni/persona per i punti elencati potrebbe essere così ripartita:

1. 10 giorni da parte di un analista
2. 10 giorni da parte di un linguista computazionale per: sviluppo delle regole di analisi, scelta dei moduli software e risorse linguistiche
3. 5 giorni di uno sviluppatore per la predisposizione della pipeline di analisi e dei servizi web
4. 3 giorni di un sistemista per l'installazione del modulo

Questa stima non comprende il costo di manutenzione annuo della piattaforma di erogazione, non essendo possibile definirne completamente, allo stato attuale, la topologia delle componenti né i volumi di utilizzo.