POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione

Corso di Laurea Specialistica in Ingegneria Informatica

KEaKI: modello e prototipo per la generazione di riassunti e per l'inferenza di nuovi concetti ontologici

Relatore: Prof. Licia SBATTELLA
Correlatore: Ing. Roberto TEDESCO

Tesi di laurea di:
Cristina BORGNIS
Matr. 724823

Anno Accademico 2009-2010

A Claudio...

''Provate a programmare un calcolatore perché capisca l'inglese o l'italiano, e il linguaggio ordinario comincerà a sembrarvi diverso. La facilità, la trasparenza, e l'automaticità sono illusioni che nascondono un sistema di grande ricchezza e bellezza''.

Pinker

Sommario

Lo scopo di questo lavoro è quello di costruire una metodologia e un prototipo in grado di creare riassunti di testi scritti in linguaggio naturale, popolare un'ontologia ed eseguire del reasoning sui concetti contenuti nell'ontologia stessa. La base di partenza del progetto è un modello basato sui frame di verbi.
Si è sviluppato un prototipo che è in grado di definire il riassunto di un testo appartenente ad un determinato dominio.
Grazie al modello qui creato si è reso il prototipo in grado di formulare il riassunto. Il tool riesce inoltre a identificare i concetti principali del testo e a formalizzarli all'interno del dominio ontologico, sul quale successivamente vengono impiegate tecniche di inferenza per determinare concetti nuovi deducibili dal riassunto.
L'analisi statistica conclusiva mostra buoni risultati. L'utilizzo del modello di frame dei verbi consente di ottenere un riassunto conforme a quello che un redattore umano formulerebbe ed inoltre, tramite l'aumento di informazioni sul dominio, è possibile ottenere un ulteriore miglioramento delle metriche qui utilizzate.

Ringraziamenti

Eccomi qui... a scrivere anche questa pagina... la voglio scrivere proprio così, senza soffermarmi sulla punteggiatura e senza pensare troppo alla forma.... flusso di coscienza!!! FINALMENTE HO FINITOOOOO!!!! Sto uscendo anche io dal mondo universitario e, adesso, sarà tutto nuovo, tutto diverso, un altro punto di partenza, un altro capitolo della mia vita!
Ringrazio la mia relatrice, la Prof. Sbattella, per avermi concesso di sviluppare questo argomento così strano e particolare, e il mio correlatore, l'Ing. Roberto che, con tanta pazienza, mi ha aiutato nello sviluppo della tesi.
A questo punto un grazie stragrande va alla mia mamma Antonella... senza di te, i tuoi consigli, il tuo appoggio e sostegno non sarei mai arrivata fin qui! Non ce lo diciamo spesso, ma da buone scorpioncine, sappiamo tenderci la mano per aiutarci e .... tu me l'hai già tesa tante volte! GRAZIE Anto!!!... sei proprio la mia mamma preferita!!! :-D
Un grazie speciale a tutti voi... a Maura e a Enrico che riempiono sempre la casa con le loro piccole scaramucce, al nonno Sandro che mi aggiusta la mitica Ferrari testa grigia, a Marino che è il solo che riesce a placare la gran lavoratrice della famiglia, per farla riposare un po' e farci stare tutti più sereni! :-D
Ehi Gian... pensavi di sfuggire a questi ringraziamenti... eggià... devo ringraziare anche te! Sei un bel pilastro nella mia vita...e non mi riferisco alla tua altezza :-D ... come fai ancora a stare con una pazza come me?!?!?!? :-D Grazie per tante... forse troppe... cose!... in particolare per la lettura e correzione della tesi!
Un grazie particolare anche agli amici dell'università... grazie a voi per tutti i consigli che mi avete dato in questi anni! Grazie mille a tutti anche per i momenti di relax e di pausa che ci siamo concessi, per il sushi o per una semplice pizza in compagnia! Un grazie speciale a Raffaele, Mattia, Francesco,

Valeria, Francesca, Marco Cas., Tiziana, Marco Cav., Alessandro, Marco B e a Roberto C.
Grazie anche a Claudia, Angela e Valentina che nell'ultimo periodo mi hanno fatto capire quanto sia importante la vita e le vecchie amicizie che, se sono vere come le nostre, non possono spegnersi mai!!!
Grazie anche alla famiglia Pappalardo che, anche se non esiste più come appartamento, è sempre un bel ricordo dei miei anni universitari! Grazie anche alla compagnia di Invorio che, nonostante la mia assenza, è sempre lì a braccia aperte aspettando che io e Gian ci facciamo vivi! :-D
Weeeee non ci credooooooo!!! Mi sto per laureareeeeeee!!!!!!!!!!!! GRAZIE CRIIIII PER LO SBATTIMENTOOOOOOO SEI TROPPO YEAHHHH YEAAAHHHHHHHHH!!!!!!!!!!

Milano, 21 Luglio 2010

...Cristina...

Indice

Capitolo 1

Introduzione

1.1 Motivazioni

I sistemi che consentono di effettuare l'analisi o la rielaborazione di testi sono attualmente in continua evoluzione. Diversi studiosi hanno affrontato **il problema della creazione del riassunto** e l'hanno risolto con delle metodologie e soluzioni personali differenti.
Una delle motivazioni principali che ha fatto da guida alla stesura del presente elaborato di tesi è proprio la ricerca di una **nuova metodologia** per affrontare questo importante problema. A tal fine si è prodotto un modello, basato su frame di verbi, adattabile a qualsiasi dominio e lingua. Si è inoltre progettato un tool che mostra come il modello sia stato effettivamente applicato, non solo per la definizione del riassunto, ma anche per la creazione di un'ontologia. La base di conoscenza così ricavata può rappresentare la mappa concettuale del testo esaminato.
Una seconda motivazione di eguale, se non di maggiore, importanza è nata dalla volontà di ricercare nuove soluzioni che possano aiutare le persone nello svolgere riassunti, nel capire meglio le informazioni presenti nel testo e nel crearsi autonomamente una mappa mentale. Si è dunque rivolta l'attenzione a quelle categorie di **persone che hanno difficoltà nell'apprendimento** a causa di difficoltà nell'utilizzo del linguaggio naturale; ci riferiamo, nello specifico, alla dislessia e alle sue varie forme. Il progetto è quindi dedicato anche a loro, con la certezza che saranno prodotti, in futuro, altri sistemi e strumenti utili per il supporto ai loro studi o alle loro semplici letture.

1.2 Obiettivi

Gli obiettivi di questo elaborato di tesi sono molteplici. L'**obiettivo primario** consiste nel produrre un riassunto a partire da un testo appartenente ad un determinato dominio. Dopo aver ottenuto un riassunto del testo originario, il **secondo obiettivo** è la creazione di un'ontologia che possa rispecchiare il testo e ne fornisca una specie di mappa concettuale. Infine, il **terzo obiettivo** è quello di creare nuovi concetti a partire da quelli originari del riassunto; per fare quest'ultima operazione ci si è appoggiati al reasoning e alle regole di inferenza.
Per raggiungere questi obiettivi si sono utilizzati un tool di analisi linguistica TULE, un sistema a regole DROOLS e un linguaggio per la creazione e la gestione del reasoning SWRL. Il prototipo è stato scritto con il linguaggio di programmazione JAVA al quale si è unito JENA per la parte di software relativa all'interfacciamento con il dominio ontologico.
Il sistema è stato applicato al dominio di storia; si sono utilizzati in particolare testi di livello elementare. Sono stati eseguiti inoltre dei testing di validazione del modello e del prototipo che hanno permesso di determinare i punti di forza e le problematiche del sistema nel suo complesso.

1.3 Struttura del documento

Il documento è diviso in sei parti fondamentali che corrispondono alla divisione dei capitoli. Il **Capitolo 2** presenta una panoramica della teoria e della letteratura presente nel campo del trattamento automatico della lingua scritta. Nella stesura del capitolo si è voluto spiegare in modo ordinato le teorie che successivamente si sono applicate realmente nella modellizzazione e nella stesura del prototipo. In questo capitolo rientra quindi lo stato dell'arte relativo agli strumenti esistenti per l'analisi del testo scritto, la definizione di riassunto e le varie tipologie di sommario esistenti e le metodologie di estrazione di ontologie dal testo. Si è voluto chiudere il capitolo con una parentesi sugli strumenti esistenti a supporto delle persone con difficoltà di apprendimento, sottolineando come il sistema presentato in questo elaborato possa essere, in un futuro, un supporto valido per DSA.
Nel **Capitolo 3** verrà presentato il nodo centrale dell'elaborato. In questo ca-

pitolo si parlerà infatti del modello che abbiamo creato e che è alla base della progettazione del processo e del prototipo vero e proprio. Il capitolo è stato suddiviso in due parti: nella prima viene presentata la teoria linguistica che supporta l'idea progettuale, mentre nella seconda ci si concentra sul modello e le sue componenti.

Il **Capitolo 4** presenta le tecnologie impiegate per la scrittura e l'implementazione del prototipo. Il capitolo è stato suddiviso in quattro aree principali che descrivono in particolare il framework linguistico TULE, impiegato nella prima fase progettuale, il sistema basato su regole DROOLS che permette di definire le regole di livello intermedio e i sistemi per la conoscenza che sono stati fondamentali per creare l'ontologia, eseguire l'inferenza e mostrare il grafo dell'ontologia come una mappa del testo in input; l'ultima parte riguarda invece Java e Jena.

Nel **Capitolo 5** si presenta il prototipo spiegando in particolare l'architettura e il processo di esecuzione dello stesso.

Il **Capitolo 6** presenta la validazione del modello. In questo capitolo si definiscono le metriche impiegate nella valutazione, la metodologia seguita e i risultati ottenuti. Si sottolinea che è stato necessario ricercare una metodologia precisa in quanto senza non è possibile definire con precisione cosa è considerato essere un riassunto ideale di un determinato testo.

Nel **Capitolo 7** si riporteranno le conclusioni del lavoro svolto; si proporranno alcune strade che potrebbero essere seguite in futuro per migliorare il tool e per aumentarne l'utilizzabilità, anche relativamente alla sua applicazione in diversi ambiti oppure con differenti linguaggi.

Capitolo 2

Il trattamento automatico del linguaggio naturale

Dalla nascita di internet scrittori, giornalisti e persone comuni hanno trovato una nuova espressione nella scrittura e nella condivisione delle informazioni. Grazie quindi alla nascita della rete sono stati progettati e si sono diffusi sistemi automatici per la memorizzazione e la ricerca di tali informazioni. Tra questi si vuole qui concentrare l'attenzione sui software per la correzione e l'analisi di testi aventi l'obiettivo di rendere la scrittura più semplice ed efficiente.
L'**elaborazione automatica della lingua** viene definita come l'insieme delle discipline che si occupano di modelli, tecnologie, sistemi, applicazioni e software relativi all'elaborazione automatica della lingua, sia parlata che scritta. Il trattamento della lingua in modo automatico è rappresentato quindi da **Speech Processing**, ovvero elaborazione del parlato, e **Natural Language Processing**, che invece si occupa dell'analisi e dell'elaborazione del testo scritto. Tra le tante applicazioni di queste metodologie troviamo lo Speech recognition utilizzato negli smartphone per lo Speech processing, l'interazione uomo-computer tramite question answering, la traduzione automatica e definizione di riassunti per la sfera che riguarda l'elaborazione dello scritto e molte altre ancora.
In questo primo capitolo viene presentato lo stato attuale della letteratura nel campo dell'elaborazione automatica del linguaggio naturale.
Il capitolo è suddiviso in cinque aree fondamentali. La prima tratta l' **elaborazione del testo**, dove vengono esemplificate le diverse tecniche impiegate. In secondo luogo si tratterà **la conoscenza** e il dominio ontologico, che de-

finisce il completamento semantico al sistema linguistico. Successivamente si offriranno una definizione e la letteratura relative al **riassunto**. Fondamentali sono anche **le metodologie di estrazione di ontologie dal testo**. Infine si tratteranno le tematiche di utilizzo dei sistemi sopra descritti nelle **aree sociali** e di disabilità, fornendo anche informazioni relative al **mercato** e alle prospettive dell'ambito NLP.

2.1 L'elaborazione del testo scritto

L'elaborazione del testo scritto (**NLP: Natural Language Processing**) cerca di riprodurre la grande capacità umana di comprendere il linguaggio. Come detto in [20] *"Il linguaggio è una meraviglia dell'ingegneria naturale"* ed è proprio per questo che risulta essere così complesso poterlo riprodurre in modo automatico.
NLP è supportato dall'utilizzo di analizzatori semantici e sintattici, metodologie statistiche e algoritmi appositamente studiati caso per caso, modelli per rappresentare la conoscenza e tecniche di annotazione, classificazione e clustering.
NLP può essere suddiviso, in prima approssimazione, in due aree: sintesi o generazione del testo; analisi o comprensione.
Per **generazione di un testo** si intende la creazione di un testo che rispetti le leggi della lingua nella quale viene creato. Esempi applicativi possono essere la generazione delle risposte nell'interazione uomo-macchina, la traduzione tra due diverse lingue o la creazione di un riassunto di un testo.
Per **comprensione di un testo** si intende invece l'estrazione delle informazioni concettuali sulla base di regole grammaticali, sintattiche, semantiche e pragmatiche o contestuali, oppure ancora sulla base di processi statistici. Le applicazioni in questo caso sono complementari rispetto a quelle relative alla sintesi: la comprensione delle frasi pronunciate dall'uomo nell'interfaccia uomo-macchina, la comprensione del testo nella lingua d'origine per generare la traduzione o per generare un riassunto. Oltre a queste applicazioni, la **nuova prospettiva** riguarda l'analisi del testo nei sistemi di information retrieval e information extraction.

2.2 Gli strumenti per l'analisi del testo scritto

Dopo aver descritto la differenza tra sintesi ed analisi, vogliamo ora concentrare la nostra attenzione sulla comprensione del testo scritto. Si descrivono qui tutte le tecniche e gli strumenti che consentono l'analisi del testo, partendo dai più semplici relativi alle singole parole, per poi affrontare le metodologie riguardanti l'intera frase e la sua sintassi.
Relativamente ad elaborazioni che afferiscono alle singole parole verranno descritti in dettaglio l'idea di token, l'analisi morfologica e quindi la lemmatizzazione e il POS tagging. Per la sfera della sintassi si entrerà invece nel merito della differenza sostanziale tra uno shallow parser e un full parser. Tali definizioni ed esemplificazioni, tratte da [12], saranno utili per affrontare i capitoli successivi del presente elaborato di tesi.

2.2.1 Singole parole

Quando ci si trova davanti ad un testo scritto in linguaggio naturale, il primo passaggio obbligato è la *"tokenizzazione"* ovvero la divisione del testo nelle sue componenti semi-minime: le parole. Si può quindi definire il **token** come il rappresentante della parola all'interno del testo e la tokenizzazione come l'operazione base di suddivisione del testo in token. Spesso però tale suddivisione non risulta essere semplice ed immediata. L'operazione descritta può basarsi infatti sulla semplice individuazione delle word in base al rilevamento degli spazi, oppure può avvalersi di tecniche più sofisticate che risolvano problematiche quali la determinazione di parole composte.
Dopo aver determinato quali sono le singole entità su cui basare le successive analisi si può passare alla determinazione del **lemma** associato ad ogni token. Il lemma viene definito in linguistica come il rappresentante "base". di ogni possibile forma flessa da esso derivabile. Ad esempio il lemma di "possiede" è "possedere", il lemma di "giochi" è "gioco" e così via. Per determinare il lemma di una qualsivoglia parola esistono degli algoritmi e delle tecniche che derivano automaticamente la sua forma base; tale operazione viene definita, in modo del tutto generale, *lemmatizzazione*. Il processo di definizione del lemma può avvenire in due modalità molto differenti tra loro: alcuni applicativi preferiscono attuare una politica di *stemming*, mentre

altri si avvalgono di techiche più sofisticate, basate su NLP e sull'**analisi morfologica**, puntando ad ottenere risultati più precisi ed articolati. Mentre nello stemming si applica un semplice processo euristico di troncamento della parola sperando di ottenere il risultato voluto, nella lemmatizzazione, basata su NLP, si utilizzano tecniche morfologiche per derivare il corretto senso della parola, e quindi il suo lemma. Tali metodi ovviamente devono essere scelti caso per caso a seconda dell'applicativo che si intende creare, soprattutto perchè propendere per l'uno o l'altro potrebbe portare il sistema ad operare in modo molto differente, anche relativamente alle performance del software stesso.

Durante la trattazione della lemmatizzazione si è parlato di morfologia. Si definisce tale termine come la branca della linguistica che si occupa dell'analisi grammaticale e che quindi classifica i diversi token in base alla loro famiglia grammaticale: nome, verbo, aggettivo, ecc. La morfologia riesce quindi a determinare i **morfemi**, ovvero la minima unità grammaticale che si possa isolare in un token; in tale elemento è possibile distinguere tra morfema radice e morfemi derivati come suffissi, prefissi o infissi. Applicando la morfologia alla lemmatizzazione è quindi possibile la derivazione di ogni singolo morfema della parola, oltre che l'identificazione della forma base, ovvero il lemma.

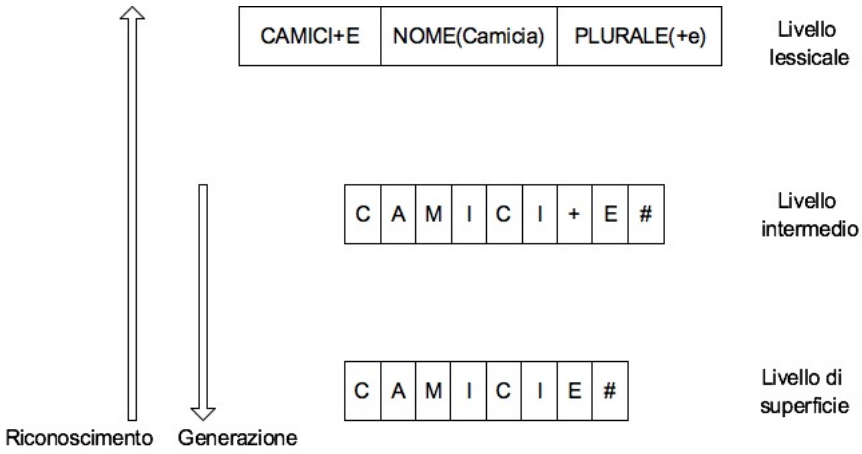

Figura 2.1: Esempio di generazione e riconoscimento.

Come esempio riportiamo la parola ‘‘camicie”: CAMICIA NOME COMUNE FEMMINILE PLURALE. Come si può notare, grazie alla lemmatizzazione basata su morfologia e grazie all'analisi morfologica stessa si è potuto

affermare che il lemma è CAMICIA, ovvero nome comune femminile. Inoltre si è determinata anche la presenza del morfema suffisso ‘‘E’’, che ha permesso di concludere che la parola è plurale.
L'analisi morfologica, che resta quindi alla base della lemmatizzazione, può essere distinta tra *generazione* e *riconoscimento* (Figura 2.1): mentre nella prima si cerca di generare forme flesse a partire dai lemmi, nella seconda si esegue l'operazione inversa procedendo nell'identificazione della forma base a partire da una delle qualsiasi flessioni della parola.
Come si può notare dall'esempio di Figura 2.1 e dalle definizione di morfologia, tali elaborazioni riguardano la singola parola e, di contro, non possono delineare con precisione il tipo di parola coinvolta nel discorso. L'analisi morfologica infatti tiene conto principalmente della grammatica della parola singola senza osservare il contesto in cui essa è immersa. Il **Part-Of-Speech Tagging** invece permette una visione più ampia associando ad ogni parola il suo POS (Part-Of-Speech) sia in base alla sua morfologia, che rispetto alla sua struttura contestuale (Figura 2.2).

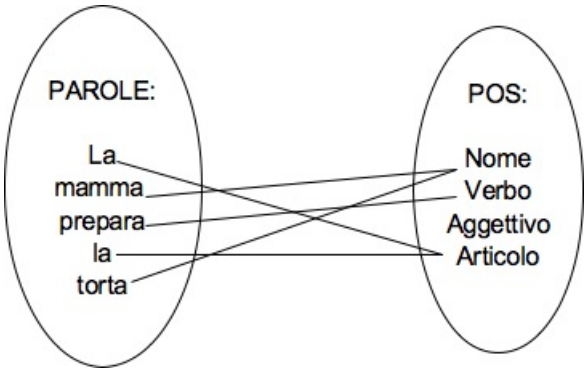

Figura 2.2: Mapping delle parole con i rispettivi POS.

2.2.2 Sintassi

Nel paragrafo precedente si sono trattate tutte le metodologie NLP per l'analisi delle singole parole. Si passa ora alla descrizione delle tecniche che consentono di determinare la struttura sintattica di intere frasi. Per determinare la sintassi delle frasi vengono applicate due tipologie di parsing: shallow parser e full parser.
Lo **shallow parser** è un parser appunto shallow, ovvero esegue un parsing

della frase dividendo semplicemente la frase in porzioni di testo aventi una loro struttura sintattica: predicato nominale, predicato verbale e i gruppi di complementi. Queste porzioni di testo vengono chiamate chunks. Di seguito si mostra un esempio di identificazione dei gruppi nominali della frase ''I saw the big dog on the hill'' in uno shallow parser:
(SENTENCE:
(NP: I)
saw
(NP: the; big; dog)
on
(NP: the; hill))

Il **full parser** è invece più complesso e, oltre a distinguere le singole parole e a descriverne la morfologia, permette di definire le *relazioni di dipendenza* tra i differenti token costituenti la frase. In particolare, alcuni tool e sistemi consentono, oltre all'estrazione delle relazioni, anche la definizione di un albero di parsing come mostrato in Figura 2.3
Di seguito si riporta un esempio di dipendenze derivato dal Parser di Stanford.

$nn(payrools-2, Factory-1)$
$nsubj(fell-3, payrools-2)$
$prep_in(fell-3, September-5)$

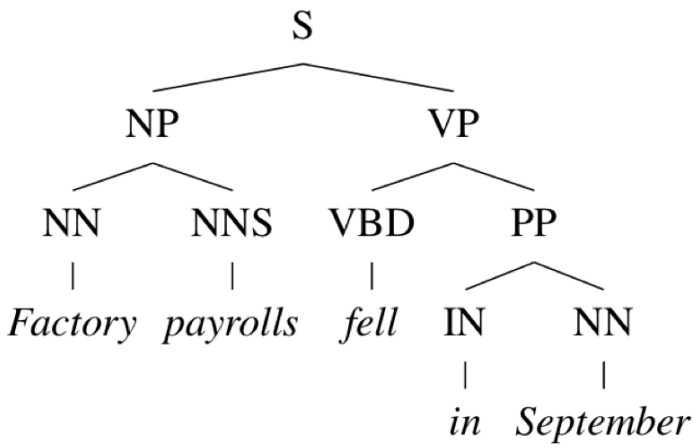

Figura 2.3: Esempio di albero dal Parser di Stanford.

In Figura 2.4 si riporta un esempio conclusivo sulle tipologie di analisi descritte, sia le tecniche riguardanti la singola parola, che le successive relative alla sintassi dell'intera frase.

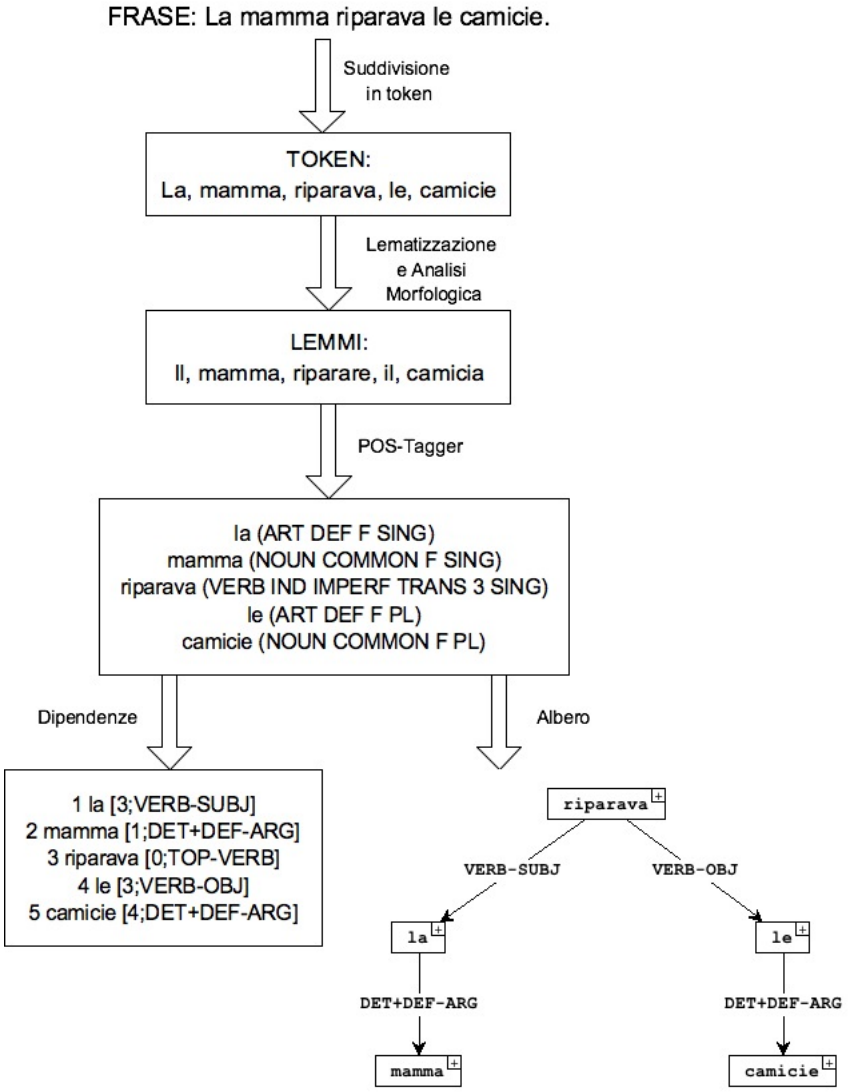

Figura 2.4: Processo completo di analisi del testo.

2.3 La semantica: La conoscenza e il dominio ontologico

Dopo l'ampia trattazione delle metodologie sintattiche sia relative alla singola parola, sia più globali e quindi dirette ad un'intera frase, si vuole qui definire l'aspetto della conoscenza che permette poi, in una fase progettuale, il giusto

connubio tra i due livelli sintattico e semantico.
La conoscenza è definita in senso del tutto generale come ''il conoscere'' ovvero il sapere d'un fenomeno, d'un fatto[1].
La conoscenza in senso più informatico viene invece descritta in un ambito di **Sistemi di Conoscenze** che vengono create, mantenute e progettate nel dominio dell'Ingegneria della Conoscenza.

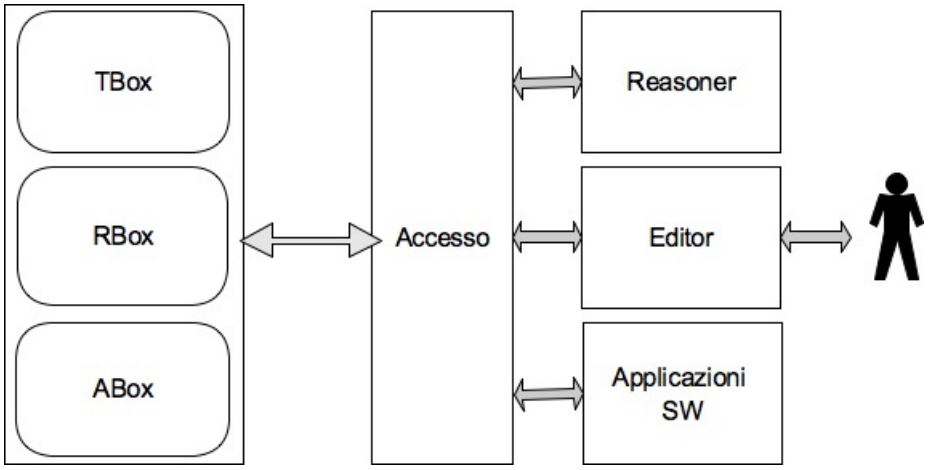

Figura 2.5: Architettura della KBS.

L'**Ingegneria della Conoscenza**, conosciuta anche come branca fondamentale dell'intelligenza artificiale, si occupa della progettazione, realizzazione, gestione delle basi di conoscenze, chiamate spesso anche **KBS (Knowledge Based System)**, che possono essere rappresentate come in Figura 2.5. Tali strutture, che in prima approssimazione sembrano essere simili a semplici basi di dati, racchiudono informazioni ben più complesse ed articolate sulle quali è possibile inferire nuove idee mediante *reasoning* e procedure automatiche di inferenza.
In particolare, in Figura 2.5 si ha un sistema basato sulle DL[2] costituito da varie componenti:

- L'**interfaccia** che consente l'accesso di reasoner, editor e applicazioni software alle conoscenze contenute nella KB;

[1]Definizione tratta dal Dizionario Sandron della lingua Italiana dell'Istituto Geografico De Agostini.
[2]Logiche descrittive

- La **base di conoscenza** che è costituita dalla TBox in cui sono definite le classi, RBox in cui sono presente le proprietà e ABox che definisce le asserzioni degli individui.

Grazie proprio alla conoscenza gli umani e gli animali riescono a prendere delle decisioni e quindi ad agire nel mondo in modo conscio e preparato. L'Ingegneria della conoscenza cerca di riportare tali conoscenze al mondo artificiale in modo tale che anche un computer, quindi in particolare un robot oppure un sistema software, possa sfruttare tali informazioni per agire e reagire al mondo che lo circonda.
E' stato precedentemente sottolineato come le basi di conoscenza siano differenti dalle basi di dati; di seguito si spiegano brevemente le motivazioni di tali diversità. Come si può intuitivamente pensare spesso l'uomo, nei suoi discorsi, sia scritti che parlati, si appoggia ad un livello astratto del discorso che contiene essenzialmente dei concetti e non delle vere e proprie parole. Se infatti con un determinato interlocutore parliamo di ‘‘casa" potremmo intendere il concetto casa nel senso di abitazione oppure di famiglia[3]. Questa informazione viene ‘‘trasportata" da chi parla a chi ascolta come un concetto che l'uomo poi formalizza e semplifica attraverso una parola. La **differenza sostanziale** quindi **tra una base di dati e una KBS** è relativa proprio all'utilizzo di **concetti** e **parole**: il database fa uso di parole mentre la KBS utilizza concetti separando quindi nettamente il termine, il concetto e gli oggetti reali:

- Il *termine* ‘‘casa", che è una parola creata dall'uomo per potersi ‘‘interfacciare" con gli altri esseri umani di lingua italiana;

- Il *concetto* casa, che è l'astrazione della parola casa alle ulteriori idee di famiglia, abitazione ecc;

- Gli *oggetti* presenti nel mondo reale appartenenti all'idea di casa, ad esempio una abitazione fisica esistente.

E' possibile capire questa tripletta tramite la grafica triangolare illustrata in Figura 2.6.
Avendo descritto quindi nel dettaglio il significato di conoscenza e la distinzione fondamentale relativa alla KBS, si vuole ora entrare nel mondo ontologico offrendo una sua definizione con degli esempi esplicativi.

[3]In questi casi l'ambiguità dei due significati viene risolta dalla pragmatica.

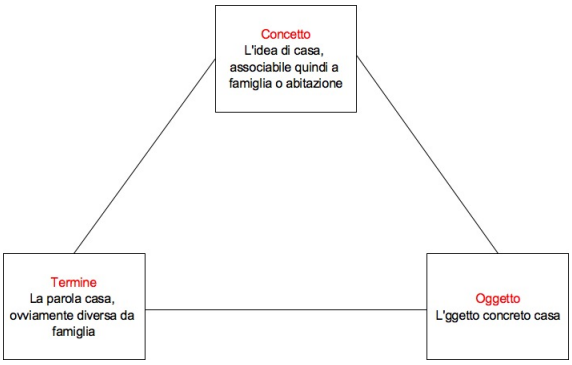

Figura 2.6: Interpretazione grafica della tripletta termine, concetto e oggetti.

2.3.1 La definizione di ontologia

L'ontologia nasce come strumento filosofico per riuscire ad inquadrare e a capire al meglio determinati comportamenti naturali. Il termine, nel tempo, assume differenti interpretazioni e viene ridefinito più volte da diversi studiosi. Nel '98, Studer ha riportato la seguente definizione *''Formal, explicit specification of a shared conceptualization"*, che può essere spiegata attraverso il disegno di figura2.7.

Come esplicitato nella definizione della KBS, le parti fondamentali che

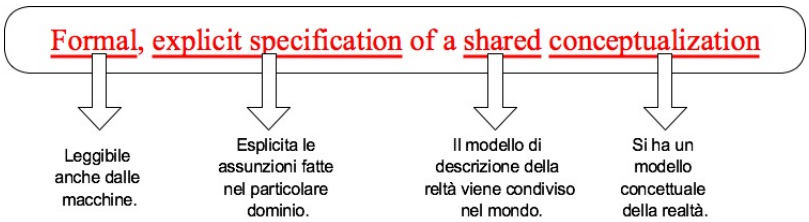

Figura 2.7: Definizione di ontologia

costituiscono l'ontologia sono le seguenti:

- le *classi*, che descrivono i concetti principali del dominio di interesse;
- le *proprietà*, che consentono di collegare fra loro le diverse classi tramite delle relazioni di dominio;
- Gli *individui*, ovvero i singoli termini che possono essere classificati come appartenenti ad una data classe.

Tali informazioni sono sempre visualizzabili attraverso un grafo che vede classi e individui come nodi, e proprietà come archi. A questo proposito si riporta un esempio (Figura 2.8):

- *Classi*: **Azienda**, **Vino**;
- *Proprietà*: **Produce**;
- *Individui*: **Fontanafredda**, **Moscato D'Asti**.

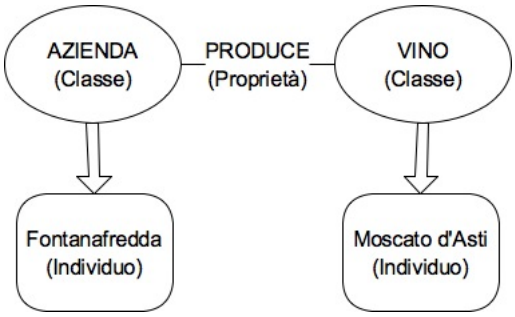

Figura 2.8: Esempio grafico dell'ontologia Azienda-Produce-Vino.

Prima di descrivere ciò che è possibile fare con le ontologie si vuole descrivere brevemente il linguaggio OWL (Web Ontology Language) che è lo standard attualmente raccomandato dal W3C. Tale linguaggio è suddiviso nelle tre complessità seguenti:

- OWL Lite: semplice da implementare ma poco espressivo;
- OWL DL: Abbastanza espressivo, utilizza una logica decidibile;
- OWL Full: molto espressivo ma non decidibile e quindi non utilizzato.

2.3.2 L'inferenza

Dopo aver stabilito che cosa è un'ontologia e come essa è composta si vuole definire ciò che è possibile fare con gli elementi e i concetti presenti nella base di conoscenza. Avendo a disposizione delle espressioni è possibile svolgere su di esse del **ragionamento automatico** che consente di passare da delle ipotesi ad una o più conclusioni, chiamate anche tesi. Viene in particolare conservata la veridicità delle premesse ottenendo una nuova espressione che ha veridicità nel contesto delle premesse fatte.
Il processo di derivazione di nuove conoscenze definito sopra viene spesso chiamato inferenza perché appunto consente di inferire, e quindi di derivare, nuove conoscenze e nuovi concetti a partire da alcune verità di partenza.
Un semplice **esempio** è il seguente:

Premesse: Bruno è papà, il papà è uomo, il papà ha almeno un figlio
Conclusioni: Bruno è uomo, Bruno ha almeno un figlio
Inferire significa quindi trarre delle conclusioni partendo da informazioni certe. Tali conclusioni sono comunque delle informazioni ricavabili dalle premesse, in quanto sono contenute in esse.

2.4 Il Riassunto

La problematica relativa alla generazione di un sommario a partire da uno o più testi si presenta come *complessa e senza una soluzione comune*. Quando si parla di generazione di riassunti in modo automatico non si arriva mai ad una metodologia unica e prevalente, ma il problema risulta essere ricco di soluzioni e proposte, che raramente garantiscono un ottimo globale e univoco.
Studiando la letteratura esistente su questo argomento ci si accorge che il panorama è definibile come una *moltitudine di soluzioni che si differenziano per metodologie e assunzioni concettuali di partenza*, e che portano a risultati spesso comparabili tra loro ad un livello principalmente qualitativo. Le soluzioni che si propongono di affrontare il problema in termini più generali, non arrivano alla creazione di un vero e proprio riassunto, oppure ne generano uno di qualità minore. I sistemi che invece si occupano di un determinato dominio ottengono risultati più precisi a svantaggio però di un utilizzo, se non ampliato, limitato sempre a quell'unico dominio. Esiste quindi un *trade-off tra delimitazione del campo di azione e del tipo di testo e precisione del riassunto generato*.
Per offrire al lettore una panoramica veloce ma allo stesso tempo esaustiva, si è scelto di descrivere le metodologie esistenti per la generazione di riassunti in base alle caratteristiche che determinano il tipo di approccio all'analisi scelto. In particolare, viene creata una suddivisione tra riassunto derivato da un **documento singolo** oppure da un **insieme di documenti**. Ci si è quindi soffermati sulla descrizione di alcune tecniche di costruzione di riassunto che, partendo dal singolo documento, applicano tecnologie basate su dominio ontologico, machine learning e caratteristiche.

2.4.1 La definizione di riassunto

Considerata la crescita esponenziale del mondo internet, la quantità di informazioni che si possono reperire sul web è sempre più in aumento. Spesso le informazioni risultano essere frammentarie e ripetitive, quindi è sempre più difficile l'identificazione di informazioni utili a soddisfare i bisogni informativi degli utenti. Il problema principale di *"Information Overloading"* può quindi essere ridotto da sistemi che possano riassumere in modo automatico uno o più documenti.
Attualmente una delle soluzioni praticate è quella di sintetizzare il contenuto informativo, con l'obiettivo di mostrare all'utente il lato concettuale del documento che si vorrebbe apprendere o conoscere; di qui la necessità di progettare basi dati in linea e sistemi automatici per consentire l'accesso alle informazioni a qualsiasi tipologia di utenza. Grazie alle moderne tecnologie è possibile interrogare basi dati, consultare prodotti automatizzati e utilizzare tutti gli strumenti necessari per poter navigare nei contenitori di dati che agevolino la ricerca dell'utente e lo orientino in un oceano di notizie che altrimenti rischierebbero di sommergerlo.
Per poter però comprendere tutte le metodologie e tecniche che consentono la generazione di un sommario, bisogna descriverne in dettaglio la definizione in modo tale da poter decidere a priori quali sono le caratteristiche che fanno di un insieme di frasi e parole un riassunto.
Un **riassunto** può essere definito come un *testo che viene prodotto da uno o più documenti diversi, che riporta le informazioni di maggiore importanza e rilevanza presenti nei documenti originali.*
Le **caratteristiche primarie** di un sommario possono essere elencate nei punti seguenti:

- Compatto: presenta le *informazioni salienti* del testo analizzato;
- Prodotto da *uno o più documenti;*
- *Lunghezza*: Mai più lungo del testo originale e spesso non più lungo di una o due pagine se il documento (i documenti) di origine è molto lungo;
- *Forme*: Potrebbe essere di tipologia differente che va dal semplice elenco di concetti fondamentali, al grafo del testo, al riassunto così come siamo abituati a pensarlo;

- Potrebbe non solo riferirsi a testo scritto ma anche *parlato o multimediale.*

L'obiettivo principale del riassunto è quello di presentare le idee principali e salienti del documento analizzato in un minor spazio, in modo tale che il lettore possa velocemente capire quello che il testo vuole mostrargli senza però doversi impiegare nella lettura dell'intero documento. Ovviamente un lettore attento e interessato può andare poi a visionare l'intero testo ma avrà comunque, grazie al sommario, individuato i passi fondamentali del testo, avendo bene in mente inoltre se gli argomenti trattati possano essere o meno di suo interesse.
Anche se, a prima vista, la creazione un sommario può non sembrare difficoltosa, tale operazione implica normalmente una serie di fasi che l'umano svolge in modo autonomo e naturale, ma che risultano essere invece complesse e laboriose se effettuate da una macchina. Il procedimento usualmente seguito da un redattore umano prevede essenzialmente la lettura dell'intero documento, la divisione del testo in parti, l'evidenziazione per ogni parte delle componenti maggiormente significative secondo il proprio istinto e bisogno, ed infine la stesura del riassunto con frasi diverse, semplici e di nuova produzione.
A tal proposito è molto esplicativo il pensiero dell'autore di [20], che afferma:

> *''Provate a programmare un calcolatore perché capisca l'inglese o l'italiano, e il linguaggio ordinario comincerà a sembrarvi diverso. La facilità, la trasparenza, e l'automaticità sono illusioni che nascondono un sistema di grande ricchezza e bellezza".*

Proprio questa idea rappresenta il percorso che si è voluto intraprendere nello svolgimento del presente elaborato di tesi.

Nella letteratura esistono svariate **classificazioni dei riassunti** in base a differenti definizioni e concetti legati al mondo della linguistica, delle modalità con cui i sommari vengono creati, del tipo di riassunto che si può ottenere oppure degli algoritmi o modelli formulati per ottenere determinati risultati.
Si può infatti distinguere, in termini di **generalità**, tra:

- *Sommari indicativi*: forniscono un'idea relativa a quello di cui il testo vuole informare senza però scendere nei dettagli specifici dei contenuti;

- *Sommari informativi*: forniscono una versione, seppur ridotta, del contenuto e dei concetti principali mantenendo quindi i dettagli informativi salienti del documento.

Si potrebbe fare una distinzione utilizzando come criterio **l'argomento trattato**:

- Riassunti *orientati all'argomento*: concentrano l'attenzione del lettore sui differenti topic di maggiore interesse;
- Riassunti *generici*: riflettono semplicemente il punto di vista dell'autore del documento stesso;
- Riassunti *basati sulla query*: si creano dei riassunti partendo dalla query formulata dall'utente.

Infine vi è la creazione di un sommario **scritto da nuovo oppure ripreso dal documento di base**:

- *Estratto*: sommario creato basandosi sul riutilizzo di alcune parti del documento come le frasi, le parole oppure interi paragrafi;
- *Abstract*: sommario creato riscrivendo e rigenerando il testo partendo da concetti espressi nel documento.

Esistono infatti differenti processi che portano alla creazione di un riassunto; gli approcci più utilizzati applicati in letteratura sono quelli di estrazione, astrazione, fusione e compressione.
Il **processo di estrazione** consente di identificare le porzioni di testo importanti e riportarle in modo identico nel riassunto, mentre il corrispondente **processo di astrazione** consente, oltre alla definizione delle porzioni più importanti del documento, di riformulare i concetti fondamentali in nuove frasi e testi.
La **fusione** invece consente la combinazione di porzioni di testo estratte, mentre la **compressione** è il processo di eliminazione dei concetti trascurabili ai fini riassuntivi.
Per ovvi motivi questi differenti metodi non vengono mai utilizzati da soli ma spesso vengono uniti e combinati per ottenere migliori prestazioni che possano far evolvere questo grande mondo della formulazione automatica del riassunto.

Per descrivere al meglio queste e altre classificazioni le abbiamo presentate nei due ambiti di produzione (a partire dal singolo documento e da più documenti) in modo da mantenere un certo ordine nella trattazione, consentendo al lettore di capire meglio le metodologie e la letteratura esistenti.

2.4.2 Il documento singolo

La creazione di riassunti pone le sue basi sulla definizione descritta principalmente da un unico testo in input. Ovviamente tale scelta dipende soprattutto dal modello che si vuole creare e dalle motivazioni che portano alla progettazione dello stesso. Se infatti si volesse concentrare l'attenzione su un unico dominio, questo approccio può essere considerato un buon metodo; al contrario bisogna capire, oltre ai concetti principali, anche il topic che si sta trattando; avere più documenti da esplorare potrebbe portare ad un incremento delle prestazioni.
Per questa e per ulteriori motivazioni spesso si tende a dire che le performance generali nei sistemi che portano alla creazione di sommari tendono ad essere migliori in approcci multi documento piuttosto che singolo; le occorrenze ripetitive dei testi in input possono essere utilizzati come indicatori di importanza in un ambiente multi-documento. Ovviamente però tale idea si rivaluta nei diversi contesti: al variare del dominio che si va a trattare e degli obiettivi che ci si prefigge, bisogna determinare, e successivamente applicare, la giusta metodologia.
I progetti eseguiti nell'ambiente a documento singolo si basano sugli approcci più svariati. Vengono difatti applicate metodologie classiche, statistiche o basate sui metodi di analisi del linguaggio naturale.
Di seguito vengono riportati degli esempi che si riferiscono principalmente a domini ontologici, features o reti neurali ampiamente applicate per l'apprendimento di una metodologia o struttura.

Il riassunto basato sul dominio ontologico

I metodi basati sui domini ontologici utilizzano appunto ontologie come entità concettuale di riferimento in modo da poter generare un sommario che sia il più dinamico e originale possibile; ciò accade perché tali metodi non si basano sulle singole parole così come le si vedono, ma piuttosto sono fondati sull'idea

di concetto, che risulta essere più ampio e generale.
Un esempio conosciuto di ontologia è **WordNet**, utilizzata in [1], nel quale ci si pone l'obiettivo di estrarre le frasi più significative del testo, semplicemente evidenziando le catene semantiche che WordNet offre relativamente ad un determinato gruppo nominale; in questo modo viene creato un riassunto senza alcuna necessità di interpretare la semantica del testo. In [1] il testo viene segmentato e vengono costruite le catene lessicali, tra le quali, vengono identificate le catene forti; sulla base di queste ultime viene redatto il riassunto. WordNet è una *rete semantica* per l'inglese sviluppata all'Università di Princeton. Questa base di dati concettuale consiste in parole connesse tra loro come nomi, verbi, aggettivi e avverbi. Queste diverse parole vengono legate tra loro formando degli insiemi di sinonimi chiamati *synset*, che rappresentano tipologie di relazioni come *iponimi e iperonimi*[4].
Le ontologie sono quindi delle entità fondamentali per poter ricostruire la conoscenza che il documento da analizzare vuole far emergere rispetto all'oceano di informazioni che viene offerto.

Il riassunto basato su caratteristiche

Nonostante nel tempo si siano ricercate delle metodologie alternative, molte delle proposte attuali riprendono comunque l'estrazione delle frasi dal documento di origine per creare il riassunto.
Molti sistemi utilizzano come dimensione le frasi, altri utilizzano interi paragrafi, oppure semplicemente singole parole ed elaborano delle caratteristiche per ogni dimensione determinata. Un esempio è [14] nel quale vengono definite caratteristiche statiche come la presenza di NE[5] nelle frasi e caratteristiche dinamiche come la similarità semantica tra le frasi e la query svolta dall'utente. Nel sistema proposto in [14] il testo originale viene sottoposto a tre fasi di elaborazione fondamentali che portano alla costruzione del riassunto: pre-processing, analisi e generazione del riassunto. La fase di pre-processing consiste in: eliminazione degli elementi superflui dal testo; suddivisione in token e identificazione delle singole frasi; riconoscimento delle NE; taggatura delle parole con il relativo POS; risoluzione di coreferenze.

[4]Gli iperonimi o sovraordinati sono vocaboli il cui significato copre tutta l'area di riferimento di altre parole dal significato più specifico, dette iponimi o sottordinati.

[5]Named Entities

Nella seconda fase vengono estratte e analizzate le diverse caratteristiche alle quali sono successivamente associati i punteggi; per svolgere questo passo è fondamentale l'utilizzo di WordNet che fornisce i sinonimi di aggettivi e avverbi contenuti nelle frasi analizzate. Nell'ultima fase il riassunto viene generato sulla base delle features con il punteggio più alto.
Nei sistemi basati su caratteristiche, le **features** vengono calcolate, analizzate, normalizzate e sommate per creare un unico valore di riferimento. Si determina inoltre un ordinamento tra le differenti dimensioni e quella che possiede il punteggio maggiore viene restituita come estratto.
Ogni tecnica che estrae frasi, paragrafi o parole calcola il punteggio basandosi su caratteristiche come la *posizione nel testo, la frequenza delle dimensioni analizzate, dimensioni chiave, o similarità semantica* tra un'ipotetica query e la dimensione stessa. E' proprio la **frequenza delle parole** che ha spesso interessato il mondo dei ricercatori nell'ambito della definizione di un sommario. L'idea su cui si basa questa particolare feature è che *più qualcosa è frequente, più esso è importante* [7]. Notiamo che, in generale, l'approccio frequenziale [19] è uno dei più diffusi rispetto all'utilizzo delle caratteristiche, ed è spesso utilizzato per la sua semplicità nelle implementazioni pratiche. Intuitivamente, possiamo però osservare che *non sempre gli elementi più frequenti in un testo risultano essere anche i più significativi, o quanto meno non sono gli unici.* Nonostante la relativa semplicità, dunque, l'approccio basato sulle frequenze si fa portatore di limiti concettuali non banali.

Dopo aver trattato le principali metodologie di creazione di un riassunto basato sul singolo testo in input si vuole anche offrire un piccolo quadro della situazione per quanto riguarda la presenza ed utilizzo di più documenti.

2.4.3 Il multi documento

Oggigiorno, con lo sviluppo del web, è importante sviluppare delle procedure per trovare in modo efficiente le informazioni che cerchiamo. I sistemi basati su testo singolo supportano la generazione di estratti, astratti, sommari basati su query, e molti altri. La definizione di riassunti di singoli documenti è efficace se si vuole capire di cosa parla un determinato testo, ma spesso è poco utile se si vuole capire quale informazione traspare da un insieme di testi; in questo caso

infatti bisognerebbe creare un sommario per ogni documento, processo che comporterebbe comunque la lettura di un certo numero di riassunti e quindi una diminuzione delle prestazioni.
Come già spiegato inizialmente, la differenza principale tra sorgenti multi o singolo documento è che il multi documento coinvolge un certo numero di testi. L'**obiettivo** in questo caso non è quello di determinare ridondanza tra i concetti presentati nei differenti documenti, ma piuttosto il fatto che il riassunto che ne scaturisce sia corretto e completo su tutti gli elementi analizzati.
I tre **problemi principali** che vanno affrontati nel caso di multi documento sono i seguenti:

- Riconoscere e copiare le *informazioni con ridondanza*;
- Identificare le *principali differenze* fra i diversi documenti;
- Assicurare una certa *coerenza* all'interno del sommario creato.

Esistono diversi sistemi basati sul web che consentono di clusterizzare documenti in base alle loro caratteristiche e contenuti; esempi sono **Google News** e **Clusty** (Figura 2.9), che consentono di creare gruppi di notizie o documenti in base alle ricerche che svolge l'utente.

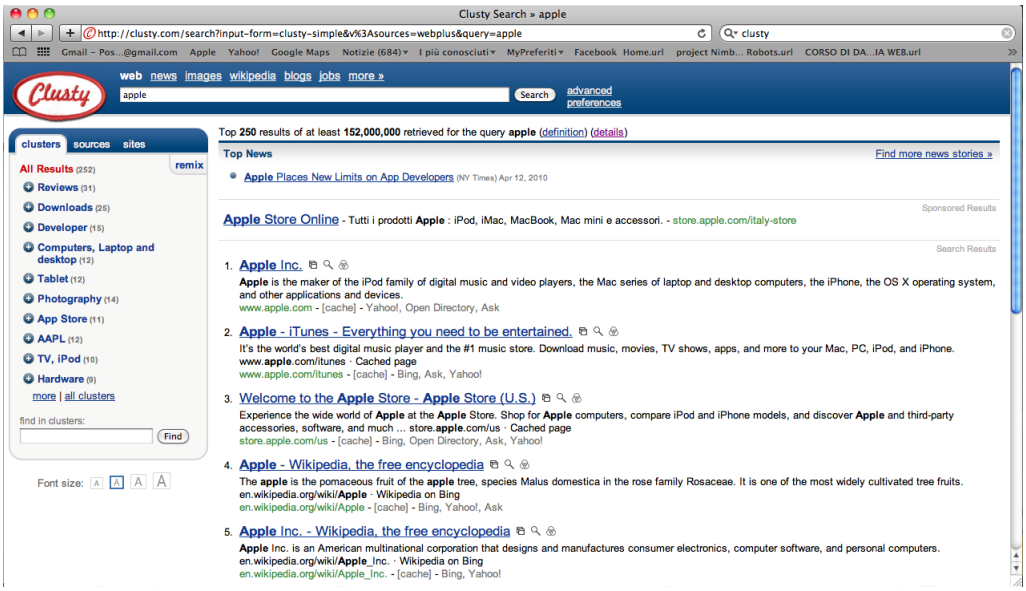

Figura 2.9: Esempio di ricerca in Clusty

2.4.4 Le evoluzioni e gli ibridi

Fino ad ora è stata presentata una quantità di metodologie particolarmente ampia e variegata. Appare allora opportuno, a questo punto, sottolineare il fatto che diverse metodologie possono essere usate contemporaneamente e parzialmente da un solo lavoro, al fine di creare una sintesi che punti essenzialmente a sfruttare gli elementi di forza di ogni tecnica specifica.
Si potrebbe infatti unire la tecnologia ontologica con quella di machine learning oppure utilizzare la frequenza insieme ai concetti ontologici o la clusterizzazione o ancora unire conoscenze linguistiche con ontologie.

2.4.5 I metodi di valutazione

La valutazione della bontà di un riassunto è considerato un *problema non semplice da risolvere* poiché, come si può facilmente immaginare, **non esiste un sommario ideale in ogni contesto e situazione**. Infatti anche ipotizzando una valutazione umana, va sottolineato che difficilmente gli esperti del settore potranno concordare pienamente sul risultato finale del riassunto. Solitamente di fronte ad un testo identico, due umani produrranno due riassunti magari simili ma non per forza identici, valutando più o meno importante un elemento piuttosto che un altro.
Per queste motivazioni sono state create e ricercate in letteratura **differenti metriche**; tra esse si citano due tipologie particolari, quali la **metrica della forma** e la **metrica del contenuto**.
La metrica della forma si focalizza principalmente sulla parte grammaticale, sulla coerenza del testo e viene normalmente misurata tramite dei punteggi che poi verranno normalizzati e sommati, al fine di offrire un valore unico rappresentante la buona forma del documento riassuntivo.
La metrica del contenuto è invece più complessa da misurare e infatti fa spesso uso di riassunti redatti da umani e si basa sulla comparazione del riassunto ottenuto in modo automatico con quello fornito dall'uomo.
Ovviamente, come succede per molte altre tipologie di sistemi, ad esempio information retrieval, è possibile far uso di metriche quali **precision, recall,**

oppure F-measure[6].

Un'interessante panoramica sulla maggioranza delle tematiche relative alla valutazione di un riassunto è stata definita in [15]. Nell'articolo si trattano diversi metodi di valutazione del riassunto suddividendoli in intriseci ed estrinseci; il primo esegue un test sul solo sommario prodotto, mentre il secondo verifica il riassunto basandosi anche sugli effetti che esso può portare alle procedure che lo utilizzano. In [15] si trattano anche metodologie per valutare la mancanza di informazione e la coerenza. Entrambe le dimensioni sono importanti per la definizione di un buon riassunto; in effetti si deve cercare un equilibrio tra la buona scrittura del testo e la sua completezza.

2.4.6 Le applicazioni

Esistono particolari ambiti in cui la produzione di un riassunto in modo automatico ha avuto maggior peso rispetto ad altre tematiche o aree. Verranno qui brevemente illustrati i topic maggiormente trattati e analizzati perché più facilmente si prestavano ad una costruzione riassuntiva, oppure semplicemente perché si aveva la grande necessità di riassumere queste informazioni per non perdersi in inutili e noiose letture.

Per primo si cita il **dominio medico**, che spesso viaggia in un binario parallelo all'Information Technology poiché è utile unire strumenti come il computer, così veloci e pratici, al bisogno informativo che necessitano specialisti medici per svolgere studi sul corpo umano o cellule tumorali in particolare. Questi due grandi mondi sono uniti anche per applicazioni quali il riassunto automatico; infatti nel dominio medico vi è una grande quantità di informazioni che spesso va ridotta alla sola rilevante per velocizzare cure mediche e quindi, estremizzando, salvare anche delle vite umane. In quest'area specifica la possibilità di ottenere sommari a basso costo temporale è fondamentale per risparmiare tempo, ottimizzare il lavoro e migliorare la produttività dei medici esperti.

Un ulteriore campo di applicazione è quello **legale**. Anche qui è fondamentale velocizzare i tempi ed ottimizzare il lavoro di risorse umane impiegate nella

[6]Precision è la percentuale di informazioni estranee presenti nel sommario. Recall è la percentuale delle informazioni omesse dal riassunto. La F-measure è un trade-off tra le prime due metriche.

ricerca del giusto percorso per un certo dibattimento o processo. Spesso documenti riassuntivi e precisi relativi alle leggi e ai loro propositi risulta essere d'obbligo per permettere agli esperti di trovare velocemente i contenuti di cui necessitano.
Con la nascita del **pervasive computing** inoltre è possibile trovare applicazioni per costruire riassunti automatici in dispositivi con piccoli schermi quali PDA o smart phone dove i sistemi vanno ottimizzati al meglio.
In maniera altrettanto importante tali sistemi trovano applicazione nell'**ambito dell'accessibilità**: per persone che hanno alcune problematiche questi sistemi risultano essere fondamentali per fornire un supporto ed un aiuto adeguato.
Concludendo si può quindi sottolineare come i sistemi di produzione di un sommario in modalità automatica siano utili soprattutto per realtà in cui vi è l'esigenza di lavorare con grandi quantità di dati non sempre ugualmente importanti e in realtà in cui il fattore di successo è il *risparmio di tempo ed energia*.

2.5 L'estrazione di ontologie dal testo

In precedenza si è offerta una trattazione del riassunto basato sul dominio ontologico. Si vuole qui invece sottolineare come la base ontologica possa essere fondamentale anche come output di un determinato processo. Le ontologie sono un grande strumento per la definizione dei concetti relativi ad un determinato dominio. Spesso infatti vengono utilizzate basi di conoscenza in diversi ambiti ma sempre in relazione a contesti ben definiti e avendo ben presente l'obiettivo da raggiungere. Nel tempo si è però capito che creare ontologie manualmente da zero ogni volta che si ha la necessità di utilizzarle in un determinato contesto è molto dispendioso. Da qui l'esigenza di implementare tecnologie e strumenti in grado di creare in modo automatico o semi-automatico delle ontologie e quindi delle interconnessioni tra concetti corrispondenti alla realtà esaminata.
Gli studiosi hanno formulato delle metodologie nuove per l'estrazione di basi ontologiche dal testo.
Un esempio significativo è il plug-in per Protègè **OntoLT** [2]. Questo approccio fornisce una metodologia e un sistema software con il quale è possibile estrarre

concetti e relazioni tra di essi in modo automatico. OntoLT definisce un certo insieme di pattern linguistici e semantici tramite annotazione; essendo però un approccio semi-automatico l'utente ha la possibilità di aggiungere nuove regole o integrare quelle esistenti.
Un **altro metodo semi-automatico** è stato sviluppato in [22]. Tale metodo consente invece l'estrazione di ontologie a partire dalla documentazione APIs e si basa sull'analisi linguistica di più basso livello come tokenizzazione e lemmatizzazione.
Esistono anche metodi di estrazione di ontologie dal testo che si basano sull'apprendimento. **Ontosophie** [3], ad esempio, è uno strumento che consente l'estrazione semi-automatica dell'ontologia utilizzando in particolar modo l'apprendimento supervisionato, grazie al quale il sistema apprende nuove regole dal testo precedentemente annotato. Anche in questo caso però è utile la presenza dell'utente, che ha la possibilità di accettare, rifiutare o modificare le istanze per il popolamento ontologico anche tramite la definizione della soglia di pruning. Ontosophie utilizza inoltre uno shallow parsing che, secondo gli autori, ha il vantaggio di essere robusto e veloce ma, spesso, risulta essere troppo poco specifico.
Esistono in letteratura anche strumenti più semplici che si basano sul **named-entity annotation** [21] o ancora sistemi specifici per un certo dominio oppure obiettivo. In [21] ci si basa sull'ipotesi che le NE presenti nel testo costituiscano una parte fondamentale per capire la semantica del testo stesso; infatti le NE vengono utilizzate per l'annotazione semantica.

Qui si intendeva semplicemente fornire una breve panoramica relativa alle tipologie di strumenti attualmente presenti e si voleva soprattutto sottolineare che vi è ancora una lunga strada da percorrere in quest'ambito per arrivare alla definizione di nuove metodologie di estrazione ontologica sempre più performanti ed estendibili ad un qualsivoglia dominio.

2.6 Il trattamento automatico della lingua come supporto alle persone con DSA

Prima di concludere questa introduzione sullo stato dell'arte, si vogliono qui descrivere le motivazioni che hanno portato alla stesura di questo elaborato di tesi. E' fondamentale osservare come lo studio, la ricerca e la definizione di nuovi sistemi nell'ambito NLP maturano grazie non solo alle implicazioni economiche che vi sono sempre all'interno di proposte nuove ed innovative, ma anche grazie alle nuove mentalità di **aiuto sociale**, del miglioramento dei servizi e infrastrutture per persone in stati particolarmente disagiati a causa di problematiche relative al linguaggio, in primo luogo *la Dislessia*[7].
Le tecnologie di trattamento automatico della lingua trovano infatti una vasta applicazione, ma soprattutto utilità, per diverse tipologie di *handicap*, non solamente per una categoria più evidente come quelle relative all'udito e alla voce oppure alla vista e al movimento, ma anche per quelle *più articolate e meno accettate* come appunto la Dislessia.
Le applicazioni di elaborazione e trattamento del linguaggio in modalità automatica possono infatti permettere una *migliore comunicazione* e una *maggiore facilità di accesso* alle informazioni e conoscenze. Tramite tali tecnologie gli utenti possono approcciarsi ai sistemi in modo più naturale e conforme alla dimensione umana.
Esistono infatti attualmente alcuni sistemi e software in grado di supportare lo studio e la lettura da parte di persone dislessiche, non vedenti o ipovedenti e disabili motori. Si citano solo alcuni esempi come le mappe mentali, i sintetizzatori vocali, controllori ortografici o audio-libri.[8] In particolare, la mappa mentale, è una forma di rappresentazione grafica del pensiero molto utile come supporto all'elaborazione delle idee e alla creatività; per tali ragioni questi sistemi sono spesso applicati in campi strutturati come quello relativo alla dislessia. Le mappe mentali presentano, infatti, alcune peculiarità che le rendono congeniali allo stile di apprendimento dei DSA; con esse le infor-

[7]La Dislessia è un Disturbo Specifico dell'Apprendimento (DSA). Con questo termine ci si riferisce ai soli disturbi delle abilità scolastiche ed in particolare a: DISLESSIA, DISORTOGRAFIA, DISGRAFIA E DISCALCULIA.

[8]Per ulteriori dettagli su software disponibili si rimanda a http://www.anastasis.it/ oppure http://www.aiditalia.org/it/software.html.

mazioni vengono presentate in una forma visiva più attraente, i concetti sono presentati senza far riferimento ad una struttura complessa grammaticale che i DSA normalmente faticano a comprendere, il testo viene presentato in uno spazio ridotto e le parti non rilevanti vengono rimosse in modo che l'apprendimento delle componenti fondamentali risulti più rapido ed efficiente da parte di persone dislessiche. Questi sistemi sono normalmente non automatici e richiedono l'intervento umano per essere utilizzati; è quindi l'umano che definisce la mappa mentale del testo, senza alcun automatismo. Uno degli obiettivi del presente elaborato consiste invece nella definizione semi-automatica delle mappe mentali che, invece, hanno bisogno di essere redatte manualmente, dal dislessico stesso.

Si vuole qui chiarire anche la motivazione che ha portato alla scelta della lingua italiana: **identità linguistica e identità culturale** sono strettamente interconnesse, nel senso che portando alla luce l'una si evidenzia anche l'altra. Spesso succede che le lingue, per le quali non vengono sviluppati sistemi di trattamento automatico, rischiano di perdere gradualmente il proprio posto nella società, insieme alle culture a cui corrispondono, causando un vero e proprio danno per la cultura di un Paese.

Proprio per tali motivazioni, culturali e sociali, il prototipo qui sviluppato formalizza una metodologia applicabile ad una qualsivoglia lingua, ma studiato qui per la lingua italiana. Sostituendo semplicemente alcuni blocchi della catena prototipale[9] è infatti possibile estendere il sistema anche ad altri linguaggi. Molti ricercatori e sviluppatori italiani svolgono oramai le proprie attività nell'ambito inglese mentre, *secondo il mio parere, bisognerebbe continuare a sviluppare anche per la propria lingua per permetterle di sopravvivere ed essere utilizzata al pari delle lingue più diffuse.*

Per offrire una panoramica a tutto tondo riguardo all'elaborazione del linguaggio naturale bisogna anche osservare che il mercato è attualmente caratterizzato da una certa immaturità; è quindi **aperto verso nuove idee e proposte**. Nella maggior parte dei casi gli attori coinvolti sono aziende medio-piccole in grado di supportare unicamente lo sviluppo delle più diffuse lingue occidentali come l'inglese.

I sistemi impiegati nel trattamento dello scritto non sempre seguono un vero

[9]Ci si riferisce al processo esemplificato nel Capitolo 5.

approccio linguistico; in molti casi i software utilizzati sono basati su tradizionali sistemi di tipo lessicale e **solo raramente si implementano soluzioni costruite su algoritmi realmente linguistici**, ovvero in grado di eseguire l'analisi morfologica, sintattica ed infine semantica.

Capitolo 3

Il modello creato

Il linguaggio umano naturale è articolato e complesso. Dalle prime scuole di infanzia ed elementari si inizia ad apprendere, prima verbalmente e poi tramite l'uso della scrittura, la grande quantità di regole che sono presenti nelle diverse lingue. Tali regole, che inizialmente possono sembrare in prevalenza grammaticali, nascondono invece ulteriori impostazioni che portano la lingua ad essere da un lato molto intuitiva e regolare per l'uomo, dall'altro lato di difficile composizione ed utilizzo. È proprio questo secondo caso che rappresenta il punto di partenza dell'elaborazione automatica del linguaggio naturale.

Con questa premessa si vuole enfatizzare come esistano regole nel linguaggio umano che nemmeno l'uomo conosce. Spesso infatti la persona umana riesce a mantener fede a tali regole, creando quindi un discorso ben formato in modo del tutto inconscio e spontaneo; l'uomo elabora nuove frasi manipolando parole, idee e concetti come se avesse ''pre-cablata'' ogni possibilità di utilizzo ed applicazione del linguaggio.

A tale scopo si vuole qui spiegare come l'esistenza dei frame di verbi, così scontati ed impliciti quando si applica il linguaggio, siano invece così particolari e di non immediata costruzione. In questo capitolo verrà fatta una breve trattazione sulle basi della **teoria linguistica** e sui suoi approcci, esemplificati in [10]. Passando quindi attraverso la storia e la letteratura di tali studi, si descriverà il **modello che abbiamo realizzato** e che è il pilastro portante del prototipo descritto nel presente elaborato di tesi.

3.1 La teoria linguistica

La **linguistica** è la *disciplina scientifica che studia il linguaggio inteso come facoltà, propria della specie umana, di usare strumenti comunicativi simbolici.* Citando [20]*''La complessità del linguaggio, dal punto di vista scientifico, è parte di quello che ci spetta di diritto dalla nascita; non è qualcosa che i genitori insegnano ai figli o qualcosa che deve essere assimilato a scuola.''.*
La teoria linguistica si occupa di elaborare e descrivere i concetti ai quali è associata tale innata capacità. Essa si può suddividere nelle seguenti **parti**:

- *fonologia*, che comprende la prosodia;
- *morfologia*;
- *sintassi*;
- *semantica*;
- *pragmatica*, le interpretazioni che le parole attivano in differenti contesti;
- *lessicologia*, che comprende l'etimologia.

Tra queste costole primeggia lo studio e l'analisi dettagliata del **ruolo del verbo**; esso è spesso stato al centro degli studi di molti linguisti, rivolti all'elaborazione di modelli che tenessero conto degli aspetti sintattici e semantici della **''proiezione argomentale''**[1]. Verrà difatti fatta una netta distinzione tra predicato ed argomenti ovvero tra ciò che viene detto riguardo all'argomento e ciò di cui si parla.

3.2 I due approcci

Si descrivono ora due approcci: **l'approccio lessicale**, per il quale risulta essere valida l'ipotesi che il numero e il tipo degli argomenti costituiscono delle grandi informazioni contenute nel verbo stesso, e **l'approccio composizionale**, per il quale invece è fondamentale la ricerca e il focus all'esterno della parola, ossia nel contesto della frase che si sta studiando.

[1]La proiezione argomentale è il meccanismo in base al quale il verbo definisce un insieme di informazioni relative alla natura, al tipo degli argomenti e alla relazione tematica che essi hanno con il verbo.

3.2.1 L'approccio lessicale

L'approccio lessicale si basa essenzialmente sull'analisi delle frasi: ne ricerca l'elemento discriminante nel predicato verbale. Nell'ambito degli approcci lessicali inoltre è possibile interpretare e considerare sia aspetti sintattici che semantici a seconda del modello applicato.

Le proprietà argomentali

La teoria lessicale nasce dagli studi [5] di concetti primitivi come **valenza** e **attante**. Nei periodi precedenti a tali studi infatti non si considerava il fatto che *alcuni complementi sono retti dal verbo, quindi obbligatori, mentre altri risultano essere facoltativi*. Nell'ambito della descrizione sintattica delle lingue classiche viene quindi introdotto il concetto di valenza, preso in prestito dal linguaggio chimico. Il verbo, infatti, è rappresentato come una sorta di atomo in grado di esercitare attrazione su un numero variabile di elementi della frase. Gli attanti vengono invece formalizzati come gli elementi, detti anche argomenti, che sono connessi all'atomo corrispondente al verbo.
Grazie a questa similitudine è infatti possibile descrivere i singoli verbi in base alle loro **proprietà valenziali**[2].
Questa teoria è fondamentale per evidenziare il fatto che frasi come le seguenti non possono essere ritenute ''ben formate'' e quindi non sono valide.

- La lampada è

- Laura torna

Sono invece corrette frasi come:

- La lampada è accesa

- Laura torna a casa

Alcune volte però può accadere che il frame di un verbo abbia un numero inferiore di argomenti rispetto a quelli che dovrebbe possedere logicamente. Bisogna quindi a questo punto dividere la valenza in sintattica e semantica. **La valenza sintattica** consiste nel numero di attanti obbligatori, ovvero quelli

[2]Le proprietà valenziali, in analogia chimica, sono il numero di argomenti retti dal verbo. Spesso tale struttura viene denominata frame del verbo

che, se non espressi, causano la non grammaticalità della frase. La **valenza semantica** invece corrisponde al numero di elementi implicati in senso più logico. Ovviamente la valenza sintattica è più facilmente descrivibile e determinabile rispetto alla semantica.
E' fondamentale inoltre sottolineare che **dato uno stesso verbo, il numero degli argomenti obbligatori può variare in relazione al significato specifico assunto nel contesto, oppure in base alla funzione pragmatica che gli argomenti supportano nel discorso**. Bisogna quindi descrivere gli attanti obbligatori sempre relativamente al contesto, alla forma sintattica della frase, al concetto che si vuole esprimere oppure al piano pragmatico del discorso.
Un ulteriore elemento da considerare consiste nella variazione del **numero e della distribuzione ed ordine degli argomenti stessi**: differenti configurazioni potrebbero, in determinati casi, offrire il significato effettivo del verbo nel particolare contesto.
Le ricerche eseguite quindi hanno evidenziato come la definizione di valenza viene data in un'ottica più sintattica che semantica, motivo per cui ci si è spostati verso un'analisi semantica relativa alla proiezione argomentale. Nasce quindi il **concetto di ruolo[3] tematico,** che viene descritto in [4] come un *modello atto a formalizzare i diversi tipi di relazioni semantiche che sussistono tra verbo e argomenti.* Se consideriamo ad esempio le due frasi seguenti:

- Il soldato uccise il capo dell'esercito opposto;
- L'esercito vinse la battaglia.

Si possono individuare, in queste due frasi, due diversi soggetti, che corrispondono a due ruoli differenti; l'uccisore e il vincitore. Tali ruoli, secondo le analisi di [4] e [6], possono essere racchiusi sotto l'unico ruolo AGENTE. L' agente è quindi il ruolo tematico che rappresenta l'idea astratta del soggetto che compie l'azione. Ovviamente anche "il capo dell'esercito opposto" e "la battaglia" sono elementi importanti nella frase e quindi nell'atto; il ruolo tematico che può essere associato a tali elementi viene definito TEMA.
Il concetto di ruolo tematico è stato applicato sia nel mondo linguistico, da

[3]Il termine ruolo, derivato appunto dalla linguistica, verrà utilizzato per definire i concetti rappresentativi degli argomenti nel prototipo.

Propp [9], in cui si utilizzano i ruoli come idea di riferimento per classificare le componenti delle fiabe, e definirne una morfologia teorica, che nel mondo più tecnico e informatico [16] nel quale, il modello teorico fornito da Propp viene studiato, ristretto e applicato, definendo un prototipo che consenta di analizzare le fiabe ed identificare i ruoli principali dei personaggi coinvolti.
Tramite il ruolo tematico si cercava di distinguere tra loro le diverse relazioni attraverso l'utilizzo di tag individuando la correlazione tra tipo di relazione verbo-argomento e la sua realizzazione sintattica, ovvero la definizione di soggetto, oggetto, complementi. Da questa analisi è emerso come **ad ogni verbo è associato un ‘‘frame" che specifica gli elementi obbligatori e quelli invece opzionali per la sintassi**.
Ad esempio il verbo ‘‘aprire" ha un frame che vuole il complemento oggetto come obbligatorio, e opzionalmente vuole un complemento di modo o di mezzo.
E' proprio in questo ambiente che si sviluppa una nuova idea; in [17] si definisce il **‘‘Theta criterion"**: in base ad esso ciascun verbo crea una ‘‘Theta grid'' che contiene informazioni relative al tipo di relazione che i diversi ruoli hanno con il verbo.
Grazie a queste ed ulteriori teorie qui non citate, si evince quindi che il verbo contiene informazioni relative agli argomenti dei quali si parla. Inoltre si è arrivati anche alla conclusione che queste informazioni possono essere rappresentate e descritte dalla teoria argomentale.
Restano però alcune domande aperte, ad esempio: quante e quali informazioni ci vengono offerte attraverso il verbo? Per rispondere a questa e ad altre domande di questo genere, molti studiosi hanno presentato i loro modelli per definire la struttura argomentale[4]. In particolare in [8] si definisce la struttura argomentale come costituita dal numero e dal tipo dei ruoli logici di una frase, suddividendoli in categorie a seconda che siano opzionali oppure obbligatori, espressi dalla sintassi oppure inglobati nella semantica e non esplicitati dalla sintassi.
Concludendo, la struttura argomentale è molto importante ed interessante per la linguistica, ma non considera aspetti più delicati che potrebbero descrivere la lingua in modo più flessibile. Sarebbe infatti utile un'analisi che tenga conto

[4]Il termine struttura argomentale nasce dall'idea che in un periodo non si ha semplicemente una lista di argomenti, ma piuttosto differenti informazioni organizzate anche gerarchicamente.

della totalità dei comportamenti sintattici del verbo nel contesto dove esso è situato.

Le proprietà aspettuali

Per rispondere ai limiti sopra esposti nasce una seconda tipologia di modelli basata sulle proprietà aspettuali invece che argomentali. In questa fase gli studiosi infatti non si basano sugli argomenti dei verbi in modo diretto ma piuttosto considerano **classi di verbi raggruppandoli per argomenti**. L'ipotesi di fondo dei modelli basati sulle proprietà aspettuali è basata sull'idea che l'aspetto sia responsabile delle restrizioni sul numero e tipo degli argomenti e il ruolo tematico.
Fondamentale per questa fase è [25], che distingue quattro classi di verbi in base alle proprietà temporali che essi hanno.

I limiti

Nell'approccio lessicale vi sono metodologie prevalentemente orientate al verbo, che cercano cioè di ricostruire argomenti e comportamenti della frase a partire dal predicato verbale.
Questa metodologia però non tiene conto di altri aspetti quali:

- Come dare informazioni delle differenze di significato nei diversi contesti;
- Come descrivere fenomeni inusuali e particolari;
- Come tener conto delle differenti tipologie di argomenti (tipo e numero) che uno stesso verbo può avere.

Atti dialogici

Il termine atti dialogici prende forma con Austin [13] e le sue teorie. In particolare Austin diceva che l'espressione formulata durante un dialogo è una forma di azione che viene in qualche modo eseguita dal parlante. Ovviamente questa idea si può facilmente ritrovare nelle frasi performative come ‘‘Vi dichiaro marito e moglie" oppure ‘‘Ti do un euro se pulisci la stanza". Infatti verbi come dichiarare o dare sono verbi performativi poiché portano ad un cambiamento nello stato del mondo.

La particolarità dello studio di Austin si basa però sul fatto che vengono identificati come atti dialogici, non solo quelli performativi, ma anche altre classi di atti come quelli seguenti:

- **Locutionary** che indicano l'espressione di una frase con un certo significato;
- **Illocutionary** cioè una frase in cui si compie l'atto di domandare, rispondere, promettere ecc.;
- **Perlocutionary** che indicano la produzione di certi effetti sul sentimento, il pensiero o le azioni del ricevente.

Spesso però il termine atti dialogici viene riferito al secondo insieme: Illocutionary act e su questo, Searle [11] ha svolto un'ulteriore analisi esplicitando altre sottoclassi degli atti dialogici:

- **Assertivi**: il parlante afferma qualcosa come vero;
- **Direttivi**: il parlante cerca di far fare qualcosa all'ascoltatore;
- **Commissivi**: il parlante si impegna a fare qualcosa in futuro;
- **Espressivi**: il parlante esprime il proprio stato psicologico relativo a una certa situazione;
- **Dichiarativi**: generano un nuovo stato del mondo per mezzo dell'espressione.

Da Searle si sono studiate le diverse possibilità di espandere gli atti dialogici e sono stati proposti diversi schemi fino ad arrivare alla definizione di DAMSL (Dialogue Act Markup in Several Layers) che codifica i diversi livelli di informazioni dialogiche.
Nel nostro modello ci siamo quindi basati sull'idea di atti dialogici ma, rispetto agli studi eseguiti da Austin e Searle, abbiamo definito un insieme di atti molto specifici e confinati al dominio considerato.

3.2.2 L'approccio composizionale

L'approccio composizionale si contrappone all'approccio lessicale poiché, invece di ricercare informazioni nel verbo e quindi all'interno della frase, definisce la ricerca all'esterno, ossia nel contesto delle singole parole.
Nell'approccio composizionale è particolare la descrizione dell'evento che viene descritto da due punti di vista: una categoria ontologica, ovvero un qualcosa di esterno al linguaggio, e una categoria linguistica, ovvero la frase vera e propria. Gli enunciati in quest'ottica non rappresentano quindi la codifica linguistica di un dato evento ma piuttosto una possibile concettualizzazione.
L'analisi dell'approccio composizionale, in particolare quella basata sull'evento, consente una maggiore flessibilità; essa infatti permette di determinare le differenti realizzazioni che un verbo offre poichè, a seconda del contesto e delle situazioni, mette in evidenza un certo tipo e numero di ruoli.

Concludendo si può sottolineare come i due aspetti, lessicale e composizionale, sono correlati tra loro poichè quando si deve definire quali siano le proprietà semantiche e argomentali dei verbi **sono ugualmente importanti entrambi gli approcci qui presentati**. Inoltre si può evidenziare che gli **atti dialogici** sono uno dei pilastri per la costruzione del nostro modello.

3.3 La classificazione del verbo e le proprietà sintattiche

Classificare significa raggruppare in classi secondo caratteri o qualità comuni [5].
Per classificazione verbale si intende, in letteratura linguistica, l'attribuzione delle parole costituenti il verbo a differenti sottocategorie in base a differenze quali il comportamento grammaticale, sintattico o argomentale. Un esempio classico e molto dibattuto è quello della differenza tra verbi transitivi e intransitivi.
Esistono ovviamente differenti basi sulle quali è possibile eseguire una classificazione: Ad esempio i fiori, potrebbero essere classificati in base al colore, al profumo o al tipo di terreno che meglio gli si addice. Ugualmente, i verbi

[5]Definizione tratta dal Dizionario Sandron della lingua italiana dell'Istituto Geografico De Agostini.

verrebbero classificati in base al significato intrinseco che portano, in base alla formalità o informalità dello stesso, oppure in base alla quantità di significati differenti che uno stesso verbo può avere in più contesti.
Esistono quindi delle proprietà sintattiche comunemente utilizzate al fine della classificazione verbale:

- **Numero di argomenti**[6];
- **Tipo dell'argomento**, ovvero se soggetto, complemento oggetto o altro complemento;
- **Posizione degli argomenti nella frase**: è importante sia la posizione, nel senso di preverbale o postverbale, sia la posizione relativa rispetto agli altri argomenti.

Si evince quindi come la classificazione del verbo è un problema di interfaccia articolato sia sul piano semantico che sul piano lessicale e sul piano sintattico. Una buona classificazione può quindi avvenire in svariati modi. Un primo modo potrebbe essere quello di **mettere in relazione i livelli sintattico e semantico**. Un altro modo potrebbe invece essere quello di **isolare le classi di verbi in base alla sola sintassi considerando ad esempio il numero di argomenti oppure le proprietà aspettuali** (soggetto, oggetto, ecc).

3.4 Il modello

Fino ad ora sono stati analizzati **due approcci linguistici differenti**, descrivendo e trattando alcuni modelli su cui essi si basano. Dalla trattazione si evince che la sola sintassi non può rispondere, da sola, all'esigenza di ottenere informazioni importanti dai diversi frame; per avere un maggiore riscontro **è utile affiancare alla sintassi un'analisi anche semantica** e quindi composizionale. E' stata inoltre descritta la classificazione del verbo e le proprietà sintattiche che consentono di ottenerla.
Ci si vuole ora addentrare nella definizione del modello creato nel presente elaborato di tesi, specificando lo studio linguistico di base che ha poi portato alla stesura di una metodologia del tutto generale e portabile rispetto ai diversi

[6]Chiamati qui anche attanti o ruoli a seconda del contesto in cui sono stati utilizzati.

contesti e domini.
Si presentano ora il **modello** e le sue **componenti principali**.

3.4.1 L'idea

Avendo spiegato precedentemente le due diverse scuole di pensiero si vuole qui evidenziare come esse siano entrambe fondamentali per un'analisi ricca ed esaustiva del problema di classificazione del verbo.
L'idea che è alla base della formulazione del nostro modello trova ispirazione proprio dalle formulazioni teoriche sopra presentate e, in particolare, utilizza sia l'approccio lessicale che quello semantico in due livelli differenti del modello. Mentre l'approccio lessicale è utile nelle fasi iniziali di identificazione della proiezione argomentale, l'approccio composizionale entra in gioco solo successivamente nella fase ontologica semantica.
In particolare, il modello si basa sull'idea concreta e reale per la quale ogni frase contiene normalmente un predicato e un certo numero di attanti. I ruoli tematici che esprimono grammaticalmente i diversi complementi, offrono la possibilità di capire il senso, sia del predicato in sè, che dei singoli argomenti presenti nella frase. Agli argomenti è successivamente possibile associare dei veri e propri ruoli che possiedono un significato ben preciso. I diversi complementi vengono quindi mappati nei ruoli che sono poi espressi come concetti ontologici. Nel passaggio tra semplici complementi e concetti ontologici si applicano in ordine, prima l'approccio lessicale e successivamente quello composizionale.

3.4.2 Le componenti

Nella costruzione del modello ci si è quindi concentrati sulla definizione di verbi e ruoli. In particolare quindi, una volta scelto il dominio di applicazione, la prima fase è la ricerca di tutti i **predicati** e di tutti i **ruoli** che ricoprono gli aspetti principali del contesto scelto.
Nel nostro caso si è deciso di applicare la metodologia a testi di storia di tipo dinamico. Tali testi sono stati soprannominati dinamici per il semplice fatto che contengono prevalentemente azioni di guerra e di combattimento.
In questa fase iniziale si sono quindi analizzati vari testi, di tipo storico, evidenziando ed estraendo tutti i verbi e complementi che offrono informazioni

specifiche sul dominio.
Un estratto di questa analisi la si può vedere nelle Tabelle 3.1 e 3.2.

Verbi
Occupare
Conquistare
Combattere
Giungere
Sconfiggere
Vincere

Tabella 3.1: Estratto dei verbi.

Ruoli
Personaggio
Civiltà
Compagnia
Oggetto
Danno
Luogo

Tabella 3.2: Estratto dei ruoli.

3.4.3 Il mapping

La seconda fase è quella del **mapping tra i verbi e i ruoli attraverso l'applicazione dei frame dei verbi**.
In particolare si è quindi analizzato ogni singolo verbo precedentemente individuato e si sono definiti tutti i possibili complementi che quest'ultimo poteva reggere. Durante questa fase si è dovuto considerare non solo il numero e il tipo di argomenti, ma anche il loro ordine; per ogni gruppo di complementi associati ad un determinato verbo si sono quindi individuate tutte le combinazioni possibili dei complementi retti dal verbo.
Avendo a disposizione tutte le informazioni relative a verbi e complementi retti dai predicati, si può formulare il mapping con i ruoli corrispondenti ai singoli

complementi.
Da questa analisi si ottengono quindi verbo, complementi retti dal verbo e ruoli associati ai complementi. Tali informazioni forniscono i concetti che corrisponderanno alle classi dell'ontologia di base.
Un esempio di mapping è mostrato nella tabella seguente:

Mapping		
Conquistare	(SOGG) (VERB) C_OGGETTO	Protagonista (VERB) Oggetto
	(SOGG) (VERB) C_OGGETTO C_SVANTAGGIO	Protagonista (VERB) Oggetto Danno
	(SOGG) (VERB) C_OGGETTO C_MODO	Protagonista (VERB) Oggetto Modo
	(SOGG) (VERB) C_MODO C_OGGETTO	Protagonista (VERB) Modo Oggetto

3.4.4 La categorizzazione del verbo

Dopo aver analizzato diversi testi, aver identificato, in particolare, verbi e ruoli e dopo aver eseguito il mapping arrivando alla definizione delle classi ontologiche, si è definito un metodo per l'identificazione le proprietà che poi verranno aggiunte all'ontologia.
La categorizzazione del verbo in base al suo frame e quindi alle sue proprietà valenziali è la base per la determinazione delle proprietà ontologiche. Avendo definito il frame di ogni verbo si sono raggruppati tra loro i verbi aventi lo stesso frame. A ciascun gruppo si è dato un nominativo che rappresenta la classe del verbo e quindi la proprietà all'interno dell'ontologia.
Un esempio di classificazione è rappresentato nella tabella seguente:

Classificazione	**Categoria-Verbi**
Proprietà	Conquistare Occupare Invadere Sconfiggere Ottenere
Vantaggio	Scontrare Combattere
Combattimento	Attaccare Assalire Assediare Scacciare Affrontare
Moto	Sbarcare Giungere Entrare Arrivare
Vittoria	Sconfiggere Battere
Alleanza	Alleare
Violazione	Violare

3.5 L'ambiguità

A questo punto è fondamentale anche chiarire come, tramite l'applicazione del modello qui presentato, sia possibile evitare le ambiguità fra diverse terminologie.

Il modello consente infatti, attraverso l'utilizzo dei frame dei verbi, l'eliminazione delle ambiguità. Un verbo che presenta un frame di un certo tipo assumerà un certo significato, lo stesso verbo che invece possiede un frame differente assumerà un significato diverso. Ovviamente non tutti i significati diversi che uno stesso verbo possiede possono essere disambiguati in questo modo, ma la maggior parte delle volte il sistema porta risultati positivi, come si può osservare dai risultati mostrati nel Capitolo 6.

Un esempio di eliminazione dell'ambiguità lo si può osservare nella tabella

seguente:

Categoria	**Verbo**	**Esempio**	**Ruoli**
Moto	Entrare	I Galli entrarono in Roma.	Protagonista(VERB) Luogo
Obiettivo	Entrare	Pirrò entrò a far parte dell'esercito.	Protagonista(VERB) Effetto

Nell'esempio si hanno infatti due diversi significati per il verbo ‘‘Entrare”. La definizione dei frame, associabili a ciascuna delle due categorie alle quali il verbo fa riferimento, aiuta nell'eliminazione dell'ambiguità. Nel caso in esame, infatti, il verbo ‘‘Entrare” può avere significato di moto oppure di obiettivo; il frame definito dal mapping aiuta a capire se, nella frase analizzata, ‘‘Entrare”. appartiene al primo o al secondo frame e, di conseguenza, alla prima o alla seconda categoria che ne definisce quindi il suo significato.

Capitolo 4

Le tecnologie applicate

Nel presente capitolo verranno brevemente descritte le tecnologie utilizzate nella progettazione ed implementazione del prototipo. In particolare possiamo distinguere quattro aree tecnologiche di sviluppo.
Viene inizialmente presentato il **framework linguistico** composto dai sistemi TULE e TUT, che hanno permesso di eseguire la prima fase implementativa. Si passa poi alla descrizione dei **sistemi esperti basati su regole**, CLIPS e DROOLS in particolare. Importante è anche la descrizione dei **sistemi di conoscenza** ed soprattutto del dominio ontologico; si sottolinea l'utilizzo dei sistemi Jambalaya e SWRL. Infine si è voluto citare il **linguaggio di programmazione** utilizzato nella stesura dell'intero prototipo: Java.
Ogni sezione è corredata dalla spiegazione delle **motivazioni** che sono alla base di una determinata scelta tecnologica. In particolare sarà descritto il perché della scelta di TULE, DROOLS e SWRL, componenti fondamentali del progetto.

4.1 Il framework linguistico

Si inizia con la descrizione del tool linguistico **TULE (Turin University Linguistic Environment)**, che consiste in un ambiente linguistico che permette di effettuare l'analisi morfologica, la definizione dei differenti token ed ammette inoltre la presenza di un parser delle dipendenze basato su regole. Per gli scopi di questo lavoro sono molto utili sia la determinazione dei token che l'analisi morfologica, ma è fondamentale soprattutto la presenza del parser delle dipendenze.

E' proprio la presenza del parser delle dipendenze, unito alla possibilità di analizzare la lingua italiana, che ci ha fatto propendere per la scelta di TULE.

4.1.1 TULE

TULE (Turin University Linguistic Environment)[1] è un progetto sviluppato dal Gruppo NLP del Dipartimento di Informatica della Università di Torino.
TULE è un framework linguistico che, data in input una frase scritta in linguaggio naturale, ritorna l'insieme delle dipendenze, visualizzato sotto forma di albero, che descrive la struttura sintattica della frase stessa. Il sistema supporta sia l'inglese che l'italiano e utilizza un formato di rappresentazione basato sul paradigma delle dipendenze di ARS (Augmented Relational Structure).
TULE è stato creato e reso disponibile sul web come server scritto in Lisp, ma è consentita anche la possibilità di scaricare TULE in locale. Inoltre viene offerto un client con interfaccia grafica per poter eseguire direttamente l'analisi dei propri file oppure di singole frasi.
Come si può notare dall'immagine 4.1 il client invia le frasi, oppure i file di testo, al server e mostra infine i risultati, ovvero l'insieme delle dipendenze, visualizzato sotto forma di albero.

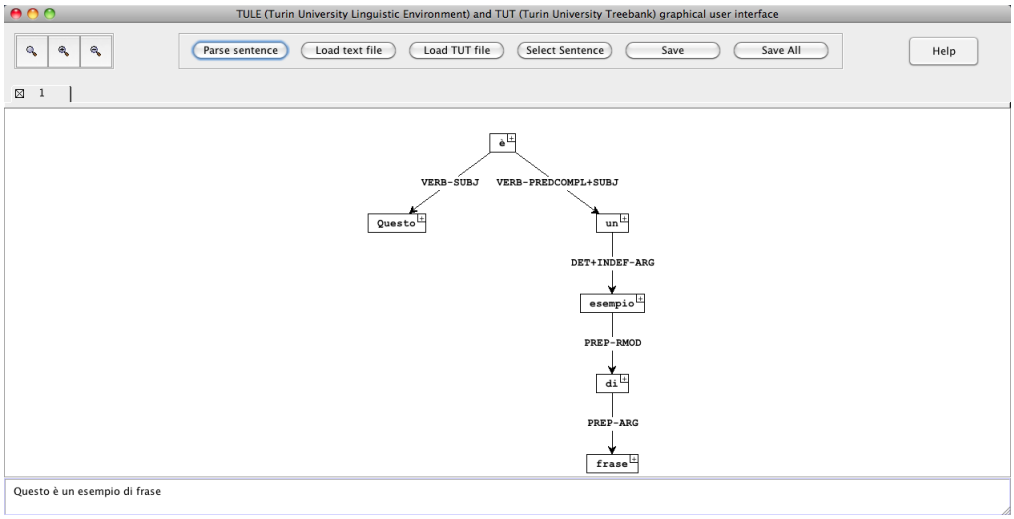

Figura 4.1: Esempio di esecuzione del client TULE

[1] http://www.tule.di.unito.it/

4.1.2 TUT

TULE è stato utilizzato in svariati progetti tra cui **TUT (Turin University Treebank)**[2], un treebanck contenente 2200 frasi italiane e 200 inglesi.
TUT è un progetto per lo sviluppo di una collezione di frasi italiane annotate morfologicamente, sintatticamente e semanticamente. TUT include infatti la definizione del formato di rappresentazione nativo che ha lo scopo principale di catturare la grande informazione offerta dalla struttura dei predicati e dei loro argomenti.
In TUT le relazioni di dipendenza sono annotate seguendo ARS (Augmented Relational Structure); quindi ogni relazione è implementata come una caratteristica che può includere valori derivati da componenti morfo-sintattici, funzionali-sintattici e sintattico-semantici.
Osserviamo ora il seguente esempio:

1 I (IL ART DEF M PL) [3;VERB-SUBJ]
2 Romani (ROMANI NOUN PROPER) [1;DET+DEF-ARG]
3 conquistarono (CONQUISTARE VERB MAIN IND REMPAST TRANS 3 PL) [0;TOP-VERB]
4 nuove (NUOVO ADJ QUALIF F PL) [5;ADJC+QUALIF-RMOD]
5 terre (TERRA NOUN COMMON F PL) [3;VERB-OBJ]
6 a (A_DANNO_DI PREP POLI LOCUTION) [3;RMOD]
7 danno (A_DANNO_DI PREP POLI LOCUTION) [6;CONTIN+LOCUT]
8 dei (A_DANNO_DI PREP POLI LOCUTION) [7;CONTIN+LOCUT]
8.1 dei (IL ART DEF M PL) [6;PREP-ARG]
9 vicini (VICINO NOUN COMMON M PL) [8.1;DET+DEF-ARG]
10 . (#. PUNCT) [3;END]

Nell'esempio troviamo la frase *"I Romani conquistarono nuove terre a danno dei vicini."* L'utilizzo di TULE nell'analisi di questa frase ci ha permesso di ricavare informazioni molto precise relative a ciascuna parola.
Ogni riga contiene tutte le informazioni relative alla singola parola, anche detta nodo dell'albero. Per ogni riga abbiamo quindi i dati seguenti[3]:

[2]http://www.di.unito.it/ tutreeb/
[3]Si rimanda all'Appendice A per ulteriori informazioni

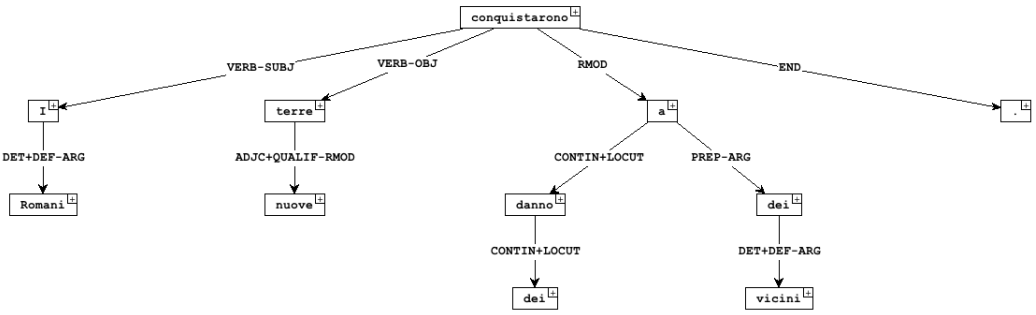

Figura 4.2: Insieme delle dipendenze, visualizzato sotto forma di albero, della frase: ‘‘I Romani conquistarono nuove terre a danno dei vicini.’’

- Il numero della riga, ovvero la posizione della parola P all'interno della frase, colorato in magenta nell'esempio;
- La parola P analizzata, in rosso nell'esempio;
- Le caratteristiche semantiche di P, in azzurro;
- Colorato in blu:
  - Il numero della parola (nodo) dalla quale P dipende;
  - La relazione che intercorre tra la parola P e il nodo precedentemente descritto;

9 vicini (VICINO NOUN COMMON M PL)[8.1;DET+DEF-ARG]

4.1.3 Le motivazioni della scelta

Prima di trattare il prossimo sistema si sottolineano le **motivazioni** che hanno portato alla scelta di TULE e TUT.
In primo luogo si cercava un parser che supportasse *l'italiano*. Secondariamente era fondamentale, per le analisi successive e per gli scopi del presente lavoro, disporre di *un analizzatore delle dipendenze* e non solamente un parser o analizzatore morfologico. Non da ultimo è stato decisivo il fatto che altri strumenti, come Italian NLP[4], non consentissero *un'analisi libera e aperta* delle frasi; infatti, mentre TULE non richiede password né registrazioni, Italian

[4]http://foxdrake.ilc.cnr.it/webtools/

NLP consente di effettuare analisi solamente previa registrazione sul web a cui bisogna fare affidamento per ogni richiesta. Tale fattore obbliga quindi a svolgere analisi online inserendo nel software l'apposita login di registrazione. Un ultimo, ma non per questo meno importante, fattore risiede nella presenza di un *server utilizzabile in locale* per TULE, che insieme alle altre qualità sopra esposte hanno portato a scegliere questo sistema.

4.2 I sistemi esperti basati su regole

La programmazione convenzionalmente utilizzata si basa prevalentamente sui linguaggi procedurali o ad oggetti che sono ottimizzati per la manipolazione di informazioni sotto forma di dati, array, liste, variabili, puntatori e cicli. Spesso però la mente umana è più facilitata nel risolvere determinati problemi usando una struttura più semplice e astratta, che all'uomo risulta essere più naturale.
I sistemi esperti forniscono allo sviluppatore una piattaforma integrata per regole, workflow, per il processing degli eventi o ancora per sistemi esperti basati su oggetti, che consentono quindi di descrivere ed eseguire un determinato compito tramite l'utilizzo di un approccio simbolico.
I compiti che vengono svolti con i sistemi esperti possono comunque essere attuati con i linguaggi di programmazione a tutti noti, ma è importante sottolineare come sia più semplice ed immediato per l'uomo concepirli ed idearli in un ambiente esperto.

Nell'elaborato è stato utile l'utilizzo dell'area riservata alla progettazione e allo sviluppo di un sistema a regole di produzione, che è una delle tecniche maggiormente impiegate per lo sviluppo di sistemi esperti. In questo paradigma le regole vengono utilizzate per rappresentare delle euristiche che specificano l'insieme delle azioni che devono essere attivate in una determinata situazione. Le regole predicano quindi sui fatti che sono delle n-ple memorizzate nella base dei fatti.
Una regola è un'entità composta dalla *parte if* e dalla *parte then*. La parte if è un insieme di pattern che specificano dati e fatti che potrebbero, se verificate, causare l'esecuzione del ramo then. Il sistema esperto fornisce un meccanismo, chiamato **motore di inferenza** (Schema di figura 4.3), che svolge un *pattern matching* delle regole e determina quali sono le regole applicabili. Il sistema

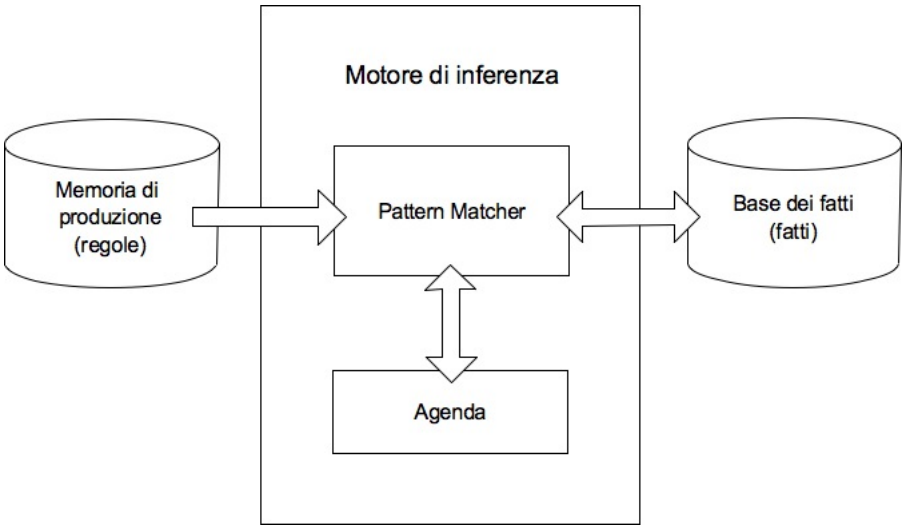

Figura 4.3: Schema di un sistema a regole di produzione ad alto livello (production rules).

quindi seleziona una regola applicabile ed esegue la porzione then della regola stessa, dopo averla eseguita seleziona la regola successiva e, se applicabile, la esegue. Il processo prosegue nello stesso modo fino a quando non vi sono più regole la cui parte if sia verificata.
Le regole vengono memorizzate nella *memoria di produzione*. I fatti che potrebbero essere applicabili vengono invece mantenuti nella *memoria di lavoro*. Spesso accade che i fatti portano all'esecuzione di più regole contemporaneamente, tali regole sono dette in conflitto. *L'agenda* permette quindi di gestire le situazioni conflittuali utilizzando delle strategie di risoluzione dei conflitti descritte in [23].
Più in dettaglio le regole, quando fanno match su uno o più fatti possono portare alle seguenti modifiche:

- Aggiungere uno o più fatti nella base dei fatti;
- Cancellare uno più fatti della base dei fatti;
- Eseguire altri comandi che non coinvolgono la base dei fatti.

Tale processo non consiste in una singola azione, ma piuttosto in una sequenza di esecuzioni: quando una regola scatta e opera sulla base dei fatti, si apportano delle modifiche che possono, a loro volta, portare all'esecuzione di un'altra regola e così via. In questo modo, si crea un ciclo, che si ferma solo

quando nessuna regola può più scattare. Se dall'esterno viene aggiunto un nuovo fatto nella base dei fatti, il tutto ricomincia. Se possono scattare più regole contemporaneamente, il sistema le esegue in sequenza basandosi su un peso (salience), che l'esperto umano può associare a ciascuna regola; in assenza di salience, l'ordine di esecuzione non è predicibile.
Riassumendo possiamo quindi affermare che un motore a regole ha le seguenti **caratteristiche**:

- Definisce *''cosa fare''* e non *''come fare''* (linguaggio dichiarativo);
- *Separa* la logica (le regole) dai dati (i fatti);
- *Focus sulla conoscenza*, a dispetto della procedura;
- Facilità ed *immediatezza nella comprensione* delle regole.

4.2.1 CLIPS

Clips[5] è un *tool di sistemi esperti* che mette a disposizione un ambiente completo per la costruzione di regole; Clips è largamente utilizzato soprattutto per creare sistemi basati su regole e basati su oggetti.
Le principali caratteristiche di Clips sono le seguenti:

- *La rappresentazione della conoscenza*: Clips fornisce un tool integrato per lo sviluppo in tre paradigmi differenti: basato su regole, orientato agli oggetti e procedurale;
- *Portabilità*: Clips è scritto in C; risulta quindi essere portabile e semplice. Può essere infatti utilizzato su differenti piattaforme come Windows XP, MacOS X, and Unix in cui è stato ampiamente testato. Inoltre è possibile modificare direttamente il codice sorgente per meglio caratterizzare il sistema per i propri scopi;
- *Integrazione ed estendibilità*: Clips può essere racchiuso in codice procedurale e può essere esteso dall'utente tramite l'utilizzo di protocolli;
- *Sviluppo interattivo*: La versione base offre un ambiente di sviluppo interattivo grafico specifico per i differenti sistemi operativi;

[5] http://clipsrules.sourceforge.net/

- *Verifica e validazione*: Clips include un certo numero di caratteristiche che supportano la verifica e la validazione dei sistemi esperti;
- *Interamente documentato*;
- *Basso costo*: software libero.

Clips è quindi risultato essere un buon sistema per l'applicazione svolta in questo elaborato.

4.2.2 DROOLS

La Business Logic integration Platform fornisce un sistema integrato per regole, worflow e processing degli eventi. **DROOLS**[6] è infatti suddiviso in *quattro sottoprogetti principali*:

- *DROOLS Guvnor* (BRMS/BPMS);
- *DROOLS Expert* (rule engine);
- *DROOLS Flow* (process/workflow);
- *DROOLS Fusion* (event processing/temporal reasoning);

DROOLS Guvnor è un repository centralizzato per le *conoscenze DROOLS* al quale si può facilmente accedere tramite il web, editor e tool appositamente preposti per gestire grandi quantità di regole, modelli, funzioni o processi. Ovviamente l'accesso a tali informazioni è controllato e può essere ristretto a determinati esperti del settore.
DROOLS Expert è il *motore per le regole* che consente di scrivere, in appositi file .drl delle regole, anche in modo semiautomatico oppure guidato. Questo sistema verrà approfondito successivamente.
DROOLS Flow fornisce la possibilità di determinare *workflow* o *processi* che descrivono in quale ordine i differenti passi devono essere eseguiti. In questo modo risulta essere più semplice la descrizione anche di processi complessi e ricchi.
DROOLS Fusion è un sistema apparentemente indipendente che si collega con gli altri moduli per *creare e gestire CEP*[7].

[6] http://www.jboss.org/drools

[7] CEP(Complex Event Processing) è un "processatore" di eventi che si occupa del processing di eventi multipli con l'obiettivo di identificarne le componenti principali.

DROOLS Expert

DROOLS Expert[8] è la parte di DROOLS che permette di creare e di gestire regole.
Il processo di Pattern Matching descritto nel paragrafo 6.2, che viene eseguito dal motore di inferenza, può essere trattato con diversi algoritmi. DROOLS implementa *l'algoritmo Rete* nella forma ReteOO, ovvero permette un utilizzo ottimizzato e avanzato di Rete per sistemi orientati agli oggetti.
Esistono inoltre due metodologie per l'esecuzione di un sistema a regole: *Forward Chaining* (data driven) e *Backward Chaining* (Goal driven) oppure gli ibridi. Per capire effettivamente quale sia il metodo migliore per una determinata applicazione è importante capire come questi funzionino: essenzialmente DROOLS è forward chaining poichè si parte dai fatti e, propagandoli, si determinano delle conclusioni. DROOLS si interfaccia molto bene anche con altri *linguaggi di programmazione*, come java, per il quale sono stati definiti plugin specifici che consentono di modificare velocemente la prospettiva (Figura 4.4) di eclipse. Aprendo la prospettiva DROOLS si possono creare dei progetti

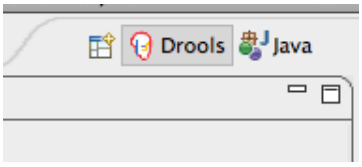

Figura 4.4: prospettiva DROOLS in eclipse

che possiedono la struttura adeguata per progettare e scrivere le diverse regole DROOLS. Tali regole saranno contenute in file .drl (Figura 4.5).

Le regole .drl vengono scritte semplicemente riportando la già discussa *sintassi if then* e vengono azionate a catena se effettivamente contengono fatti validi per la loro esecuzione.

```
rule ‘‘name’’
   attributes
   when
      <parte condizionale>
   then
      <codice da eseguire>
```

[8] http://www.jboss.org/drools/drools-expert.html

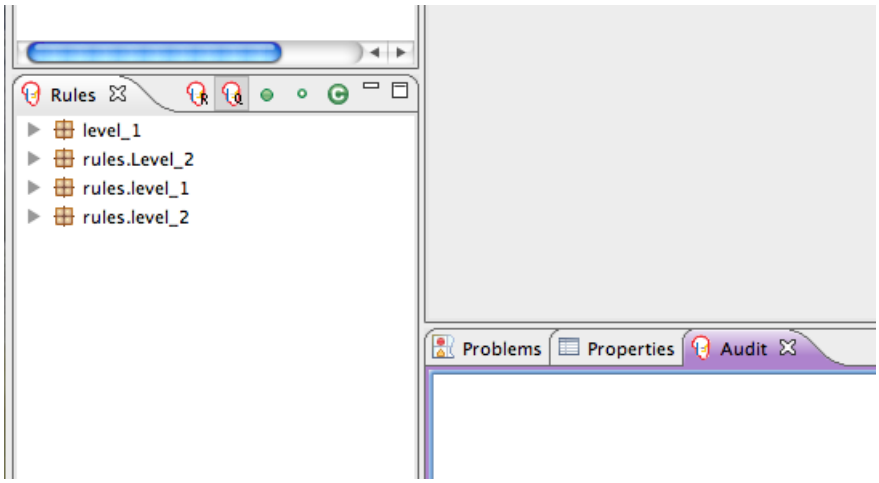

Figura 4.5: packages contenenti i file .drl di regole DROOLS

```
end
```

Un *esempio di regola* è riportato di seguito:

```
 rule ‘‘increase balance for credits’’
when
   ap :  AccountPeriod()
   acc :  Account( accountNo :  accountNo )
   CashFlow( type == CREDIT,
   accountNo == $accountNo,
   date >= ap.start  <= ap.end,
   $amount :  amount )
then
   acc.balance += $amount;
end
```

4.2.3 Le motivazioni della scelta

Avendo a disposizione dei sistemi differenti, **CLIPS e DROOLS**, che svolgono bene le stesse funzioni, si è deciso di utilizzare DROOLS semplicemente perché è di *più recente* progettazione e viene spesso aggiornato. Inoltre, come descritto, offre una *gamma di prodotti e funzionalità molto ampie*, quindi imparare ad utilizzarlo può aprire anche la strada per la sua applicazione in altri ambiti ed aree. Non è da sottovalutare anche la presenza di *ReteOO* appositamente

progettato e ottimizzato per DROOLS e per la progettazione object oriented che invece in CLIPS non è definito appositamente per oggetti.
Per queste ragioni si è deciso di utilizzare DROOLS come potente strumento di definizione e scrittura di regole per una determinata fase di implementazione del prototipo[9].

4.3 La conoscenza

Dopo aver parlato, nel Capitolo 2, dell'Ingegneria della Conoscenza e quindi della definizione di ontologia, verranno descritti ora gli strumenti basati su conoscenza utilizzati nel prototipo, soffermandosi in particolare sull'editor di ontologie **Protégé**, sulle **regole SWRL** e sul **reasoner generico** disponibile per Java.

4.3.1 Protégé

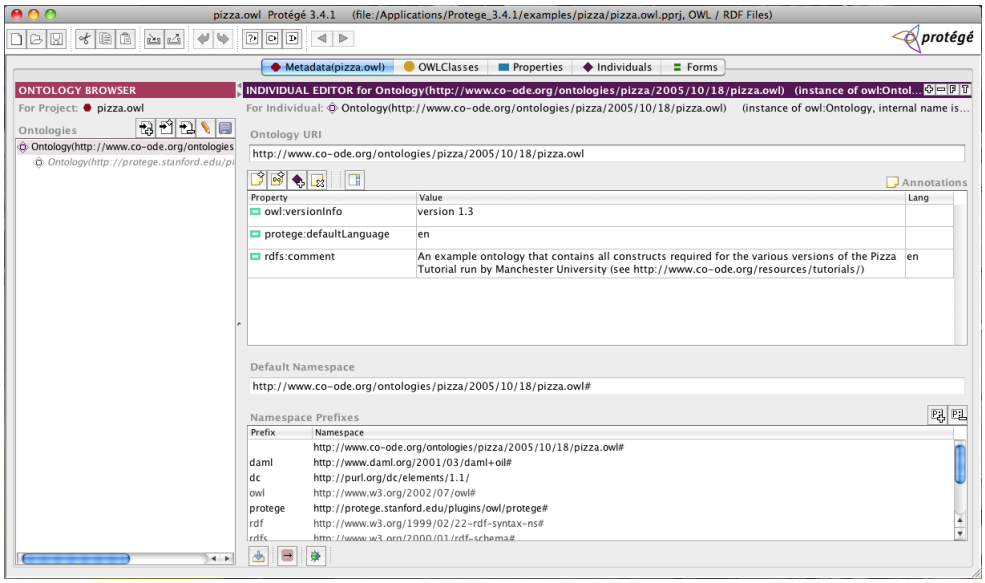

Figura 4.6: Interfaccia grafica di Protégé.

Protégé[10] è un *editor di ontologie*, il più utilizzato, libero e open source disponibile sul web nelle sue differenti versioni e per differenti sistemi operativi. La piattaforma supporta due modalità per creare e progettare

[9]Si rimanda al capitolo ... per i dettagli sull'utilizzo pratico di DROOLS nel prototipo.
[10]http://protege.stanford.edu/

ontologie: Protégé-Frames e Protégé-OWL. Le ontologie possono essere esportate in vari formati come RDF(S), OWL e XML.

Il suo utilizzo risulta essere semplice ed immediato grazie alla grande quantità di tutorial, documenti e paper e grazie alla semplice e colorata interfaccia grafica (Figura 4.6); è infatti molto semplice ed intuitivo l'inserimento, e la cancellazione, di *classi*, *proprietà* ed *individui* che vengono rappresentati tramite il tondo giallo, il quadrato blu, il rombo viola e i relativi tab. Come mostrato in Figura 4.7 è facile inserire nuove proprietà e restrizioni legate alle differenti classi tramite la GUI appositamente predisposta.

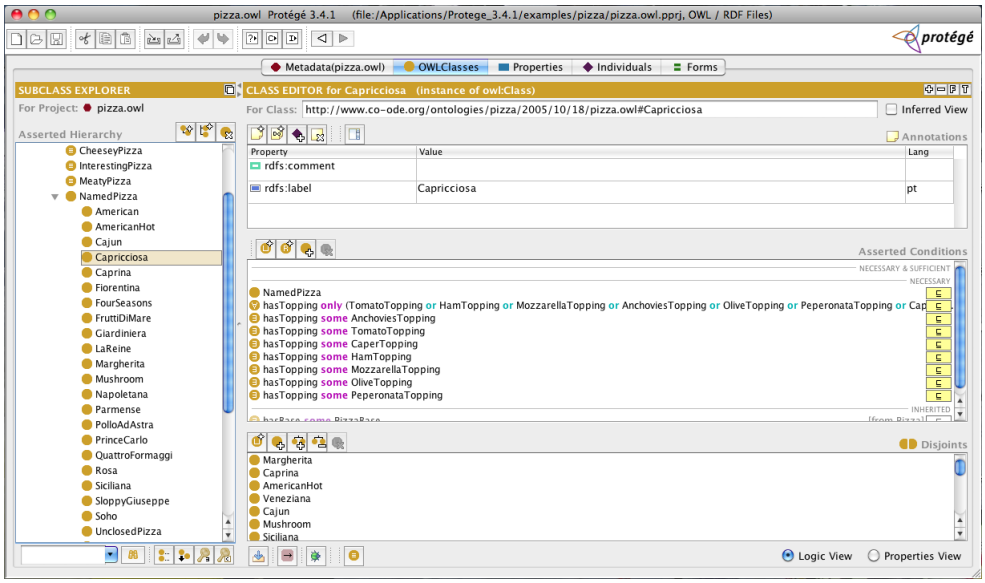

Figura 4.7: Visualizzazione della classe ''Capricciosa'' nel tab apposito.

Protégé è molto utile anche grazie ai suoi *plugin e ulteriori tab* che possono essere aggiunti al sistema. Su tutti si citano **Jambalaya**[11] (Figura 4.8), che consente di rappresentare in modo *grafico* l'insieme delle classi definite, le relazioni esistenti tra di esse e molto altro a seconda delle view scelte, e **SWRL Rule**[12] (Figura 4.9), che consente invece di definire, tramite la GUI Protégé delle *regole* con la sintassi SWRL, e di eseguire nuove regole.

[11] http://protegewiki.stanford.edu/wiki/Jambalaya
[12] http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

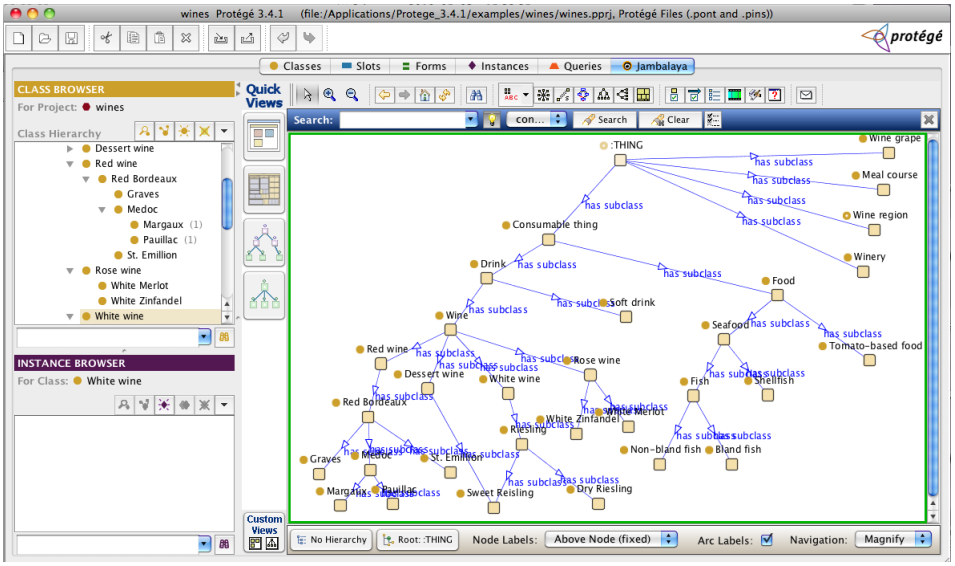

Figura 4.8: Tab Jambalaya.

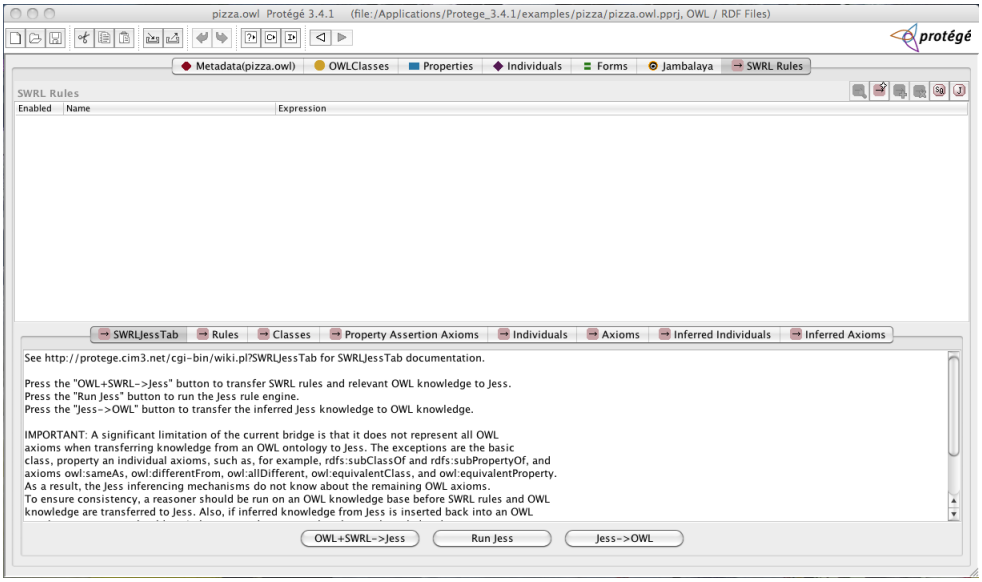

Figura 4.9: Tab SWRL Rule.

4.3.2 SWRL

SWRL (Semantic Web Rule Language) è una versione compatibile con OWL del linguaggio a regole Datalog. Una base di regole SWRL è composta da un insieme di formule condizionali, dette regole, della forma:

$$X_1 \wedge X_2 \wedge X_3 \wedge ... \wedge X_n \rightarrow Y$$

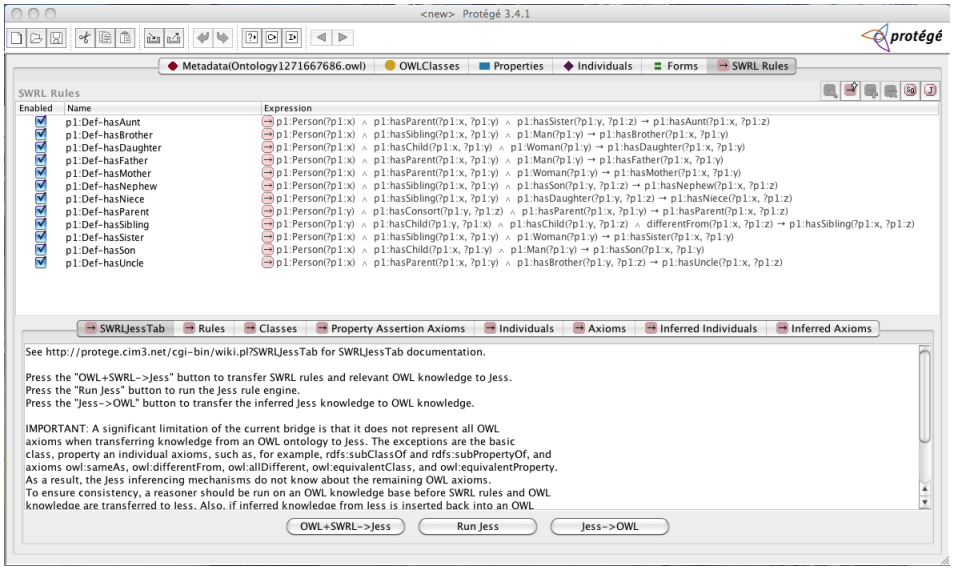

Figura 4.10: Tab SWRL Rule con diversi esempi di regole tratte dall'ontolgia della famiglia.

Dove l'antecedente è detto *corpo* della regola mentre il conseguente è chiamato *testa*.
Gli elementi che normalmente costituiscono una regola hanno una delle forme seguenti:

- A(t) dove A è una classe denominata e t è un individuo o una variabile;
- R(t,t'), dove R è una proprietà e sia t sia t' sono individui o variabili;
- P(t,t'), dove P è un attributo, t è un individuo o una variabile e t' è un dato o una variabile;
- (t=t') o $(t \neq t')$, dove sia t che t' sono individui o variabili.

Il concetto di variabile è definito come un identificatore arbitrario preceduto da "?".

Bisogna inoltre evidenziare che combinando OWL e SWRL si ottiene un linguaggio molto espressivo, così tanto espressivo da risultare indecidibile. A questo punto sembrerebbe quindi che SWRL non sia applicabile ma è possibile utilizzare le regole in *modalità ''safe"* che consente di mantenere la decidibilità. Tale modalità consiste nel legare le variabili ai soli oggetti contenuti nell'ontologia, cioè noti nell'universo descritto dalla KBS.

4.3.3 Reasoner

Si è precedentemente detto che la presenza dei concetti permette di tracciare un confine tra basi di dati e ontologie. L'esistenza della sfera concettuale non è però l'unico segnale di tale distinzione. Un ulteriore distinzione può essere fatta sulla base della presenza di un **reasoner** nel dominio ontologico, che invece non è presente nel mondo databasistico.
Un ragionatore consente di eseguire ragionamenti e inferenze in modo automatico sui fatti noti.
Essendo il mondo delle ontologie basato sulle logiche, un servizio che consente una deduzione si fonda principalmente sull'idea di dimostrare se un enunciato è conseguenza logica o meno di una KBS.
Grazie a Protégé e ai differenti plugin di Java è possibile far uso di ragionatori generali molto semplici quali **Pellet** o **Fact++**. In particolare per il prototipo si è utilizzato un ragionatore generico di Jena; dovendo scrivere anche le diverse regole SWRL risultava comoda l'applicazione tramite l'utilizzo diretto del codice.

4.3.4 Le motivazioni della scelta

Dopo aver descritto i sistemi di conoscenze utilizzati si spiegano brevemente le motivazioni che hanno portato all'utilizzo dell'ontologia, in particolare del sistema a regole SWRL.
Come verrà poi descritto nella sezione relativa al prototipo, era fondamentale, ad un certo punto della progettazione del sistema, avere un *riferimento concettuale*. Il sistema concettuale più utilizzato e più semplice da impiegare era proprio il dominio delle conoscenze e quindi ontologico.
Per quanto riguarda la scelta di SWRL vi è la volontà di *distinguere due fasi progettuali* basilari del prototipo: la parte eseguita dalle regole DROOLS dalla

parte più semantica e concettuale. Avendo oramai una certa esperienza nell'utilizzo DROOLS Expert, si sarebbe potuto continuare ad appoggiarsi sulle sue funzionalità; esistendo però un sistema come SWRL così semplice ed efficace per le inferenze e il reasoning ontologico e volendo inoltre separare le due aree progettuali del prototipo si è optato per SWRL e le sue regole.

4.4 Java e Jena

Il prototipo è stato sviluppato in **Java**, linguaggio ampiamente noto nel settore e di semplice utilizzo grazie alle sue caratteristiche e potenzialità che saranno qui descritte solo in parte e velocemente.
Java è un linguaggio semplice poiché è stato appositamente studiato per non dover eseguire lunghi periodi di training prima di poterlo effettivamente applicare. Java è noto come un linguaggio *orientato agli oggetti*, fornisce quindi tutti gli strumenti necessari alla creazione e gestione delle classi e degli oggetti più in particolare. Come altra grande potenzialità Java è *distribuito*, quindi include tutte le librerie necessarie per lavorare in rete tramite il supporto dei protocolli della pila TCP/IP. Il sistema è anche *interpretato*, quindi, se disponibile l'interprete, può essere eseguito su un qualsiasi tipo di macchina senza alcuna problematica.
Questo linguaggio possiede delle caratteristiche che lo rendono affidabile e utilizzabile anche in vasti progetti e applicativi. Sono infatti oramai presenti diversi plugin per poter avvalersi dell'utilizzo di molte funzionalità, che spesso sono invece intrinseche di altri software, anche in java, senza dover integrare più sistemi per ottenere i risultati voluti. Un esempio è rappresentato da **Jena**, un framework per il web semantico definito per Java. Jena fornisce strumenti validi per RDF, RDFS, OWL e SPARQL, propone anche un motore di inferenza basato su regole e consente quindi l'accesso, la modifica e l'inserimento di informazioni ontologiche. Queste caratteristiche unite al bisogno di lavorare sulle ontologie direttamente dal codice java ha portato alla scelta di questo framework.

Capitolo 5

Il prototipo

Le informazioni teoriche descritte nei capitoli precedenti, unite al nostro modello e alle conoscenze delle tecnologie scelte, ci consentono di porre le basi per la realizzazione del prototipo secondo gli obiettivi che sono stati presentati all'inizio del presente elaborato di tesi.
In questo capitolo si vuole quindi presentare il prototipo realizzato, descrivendo inizialmente il funzionamento tecnico del sistema e i passi che hanno portato alla rilevazione degli output. Successivamente si tratterà il sistema software che implementa tutti i passi precedentemente descritti.
Per maggiore chiarezza si suddivide il capitolo in quattro parti fondamentali che consentono di offrire una dettagliata panoramica sul sistema e sulle sue principali componenti.
Nella prima parte si riporta la **descrizione dell'architettura generale del sistema**, dando una visione tipica a *''Black-Box''* in cui si descrivono principalmente l'input al sistema, le conoscenze a priori sul dominio considerato e gli output ottenuti, come precisato dall'obiettivo. Nella seconda parte si apre la scatola nera, andando ad evidenziare **il processo interno** che viene suddiviso in tre parti fondamentali per meglio descriverne la struttura e il funzionamento. Successivamente si passa alla descrizione del **prototipo** creato, spiegando in particolare l'utilizzo dell'interfaccia e delle sue componenti. Si mostra infine un **esempio pratico** del funzionamento del prototipo.

5.1 L'architettura

L'architettura del sistema, descritta nella figura 5.1, mostra una visione Black-Box del progetto, evidenziando quindi l'input, la conoscenza a priori sul dominio e gli output ottenuti.

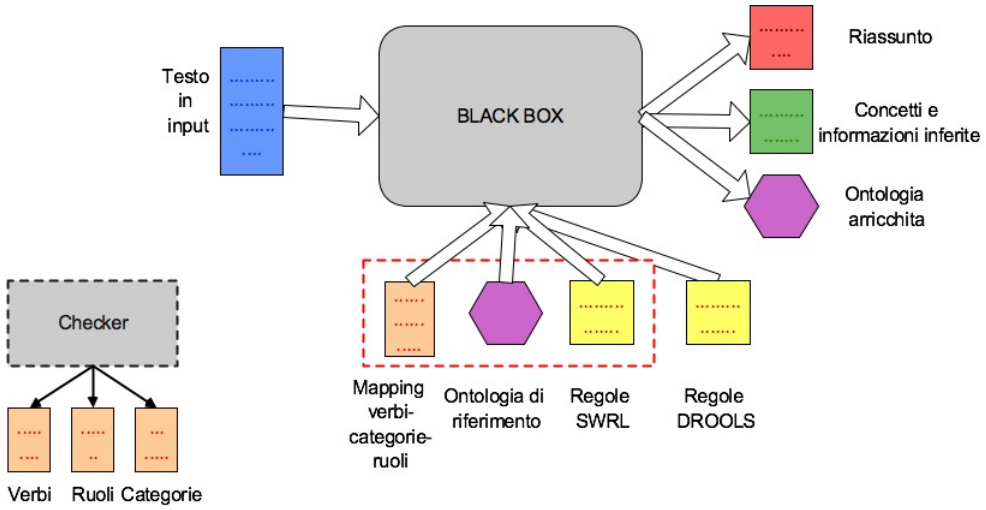

Figura 5.1: L'architettura "Black-Box" del sistema.

La figura mostra, a sinistra, l'input in **ingresso** al sistema, a destra i tre principali **output** che il prototipo produce, mentre in basso sono rappresentati, nel quadro rosso, le **componenti parametriche** del sistema. Il mapping verbi-categorie-ruoli, l'ontologia di riferimento e le regole SWRL sono infatti i parametri che, se variati, portano il modello ad assumere un **carattere del tutto generale**, poiché consentono di poter applicare il sistema ai domini più svariati.

Come verrà dettagliato in seguito, sono state utilizzate **due tipologie di regole**, quelle relative al livello più basso di elaborazione dell'output TULE e di generazione delle categorie associabili a ciascun verbo scritte con DROOLS, e quelle relative alla semantica e all'inferenza eseguite grazie a SWRL. Le due tipologie sono concettualmente molto differenti poiché, mentre la prima non dipende dal dominio analizzato, la seconda è fortemente dipendente dal contesto in quanto consente di dedurre nuove informazioni sulla base delle premesse presenti nell'ontologia arricchita. Essendo l'ontologia dipendente dal dominio considerato, la dipendenza si avrà anche rispetto alle regole. In particolare si è voluto distinguere le due tipologie di regole utilizzando anche due tecnologie ad

hoc. Dallo schema di figura 5.1 si può facilmente notare che le regole DROOLS sono state sistemate a fianco della Black-Box, ma non rientrano come input parametrico al sistema, mentre le regole SWRL sono state sistemate nel blocco rosso, proprio per accentuare le grandi differenze che vi sono tra le due tipologie.
Il prototipo prende in ingresso un testo, in formato testo Unicode (UTF-8), corrispondente al modello descritto nel Capitolo 3. Dopo aver eseguito delle elaborazioni, che descriveremo successivamente, determina i tre output fondamentali ricercati:

- Il **riassunto**;
- **Nuovi concetti** inferiti dai concetti presenti nel testo;
- **L'ontologia**, degli individui presenti nel testo, **arricchita**.

Entrando nel dettaglio, la conoscenza a priori sul dominio considerato riguarda due tipologie di informazioni:

- L'**ontologia** di riferimento;
- Il **mapping** dei verbi con le categorie e i ruoli.

L'ontologia serve appunto come base per potervi aggiungere successivamente categorie (classi) e ruoli (individui) derivati dal testo di input, in modo da poter eseguire delle inferenze con regole SWRL e il reasoner di JENA, ottenendo così nuovi concetti e legami.
Il mapping dei verbi, descritto ampiamente nel capitolo 3, è qui realizzato tramite un file in formato testuale, di cui un estratto è rappresentato in Figura 5.2, che contiene le informazioni sui frame dei verbi. Tale mapping è utilizzato per il riconoscimento, all'interno del testo, dei concetti salienti, ma soprattutto per la classificazione dei verbi in base al loro frame che successivamente viene rappresentato con una veste a categorie.
E' fondamentale sottolineare come il mapping di verbi-ruoli-categorie possa essere creato da un applicativo che risieda a monte del sistema qui presentato. Tale applicativo potrebbe, in base a studi ad hoc per i differenti domini, esaminare tutti i concetti chiave e raggrupparli classificandoli in base alle rappresentanze di ruoli e verbi.

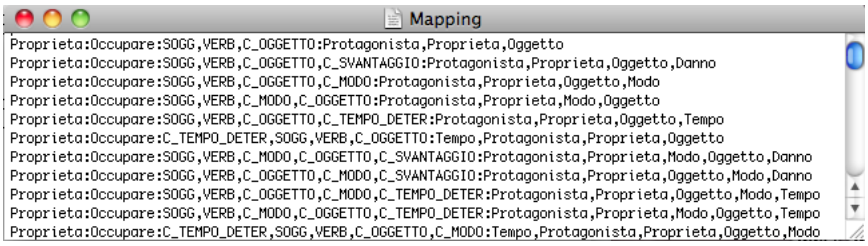

Figura 5.2: File di mapping Verbi-Categorie-Ruoli.

Nell'architettura troviamo inoltre il **blocco checker** che consente di effettuare la verifica delle definizioni di verbi, ruoli e categorie che sono determinate inizialmente in relazione al dominio da analizzare.
All'inizio della progettazione si è determinato un elenco preciso dei termini incontrati per il dominio considerato; tali informazioni sono state successivamente suddivise in verbi, ruoli e categorie. Dopo aver raccolto le differenti terminologie, si è eseguita una mappatura di tali parole all'interno del file descritto in figura 5.2. Tale file è stato successivamente utilizzato come sonda per scoprire la presenza di tali frame nel testo.
La funzionalità fondamentale del checker è quindi quella di verificare che tutte le parole e concetti presenti nel file di mapping siano state definite per il dominio considerato. Se così non fosse si ha un problema di incompatibilità e vanno riviste quindi, da parte di un linguista esperto, le mappature che potrebbero contenere inesattezze o mancanze.

5.2 II processo

Nel paragrafo precedente ci si è concentrati sulla descrizione della parte esterna al sistema. Si sono infatti analizzati gli elementi che ‘‘dialogano" con il sistema, sia per quanto riguarda i suoi ingressi sia per le informazioni che si ottengono in uscita.
Avendo descritto quello che circonda il sistema, i suoi ingressi e i suoi output, si vuole ora aprire la scatola nera per poter spiegare come, dal testo in input, si riesca ad arrivare ai tre output sopra esposti. Per descrivere al meglio il processo di estrazione dei concetti e ampliamento degli stessi si fa riferimento alla Figura 5.3 in cui sono evidenziati i diversi blocchi della catena di descrizione dell'intero processo.
Il processo descritto dalla figura va letto dal basso verso l'alto. Si è deciso di

rappresentare tale flusso nel verso di crescita verticale, proprio per dare l'idea di una suddivisione a livelli del progetto stesso. Tali layer si accostano partendo da informazioni di più basso livello, dal linguaggio naturale alla sintassi, fino ai livelli superiori che riportano idee e concetti tipiche del livello più semantico. Si passa quindi da un livello puramente sintattico ad uno di livello maggiore, passando per il layer di regole che consentono di creare questo importante legame.
Nel diagramma si è voluta inserire una ''legenda" a colori che consente di identificare, anche in strati differenti, i blocchi, e quindi le operazioni, che concorrono al raggiungimento di un obiettivo parziale.

Come si può notare infatti si è scelto di utilizzare uno stesso colore, anche se con gradazioni differenti, per evidenziare i **macro-blocchi** seguenti:

- Grigio: blocchi generici;
- Blu e azzurro: Blocco TULE-TUT;
- Rosa e rosa chiaro: Regole di definizione dei complementi inter-frasi;
- Giallo: Blocco di estrazione di informazioni di sintagma;
- Rosso e rosso chiaro: Regole di estrazione dei complementi intra-frase;
- Verde e verde chiaro: Blocchi semantici;
- Viola e violetto: Ontologie.

Si descrive ora il processo, considerando i macro blocchi che lo compongono e le differenti interazioni che vi sono tra essi.
Il sistema inizia la sua elaborazione partendo da un **testo** appartenente al dominio di interesse e avente una struttura come quella precisata nel Capitolo 3. Dopo aver memorizzato tutte le componenti del file di input si passa ad un primo macro processo in cui si eseguono le operazioni di suddivisione in token, lemmatizzazione e analisi morfologica, POS-Tagger e analisi delle dipendenze. Tutte queste tecniche, ampiamente descritte all'interno del secondo capitolo, vengono applicate da **TULE**.
Dopo aver ottenuto tutte le informazioni morfologiche e delle dipendenze necessarie alle elaborazioni successive, si passa alla seconda fase, nella quale,

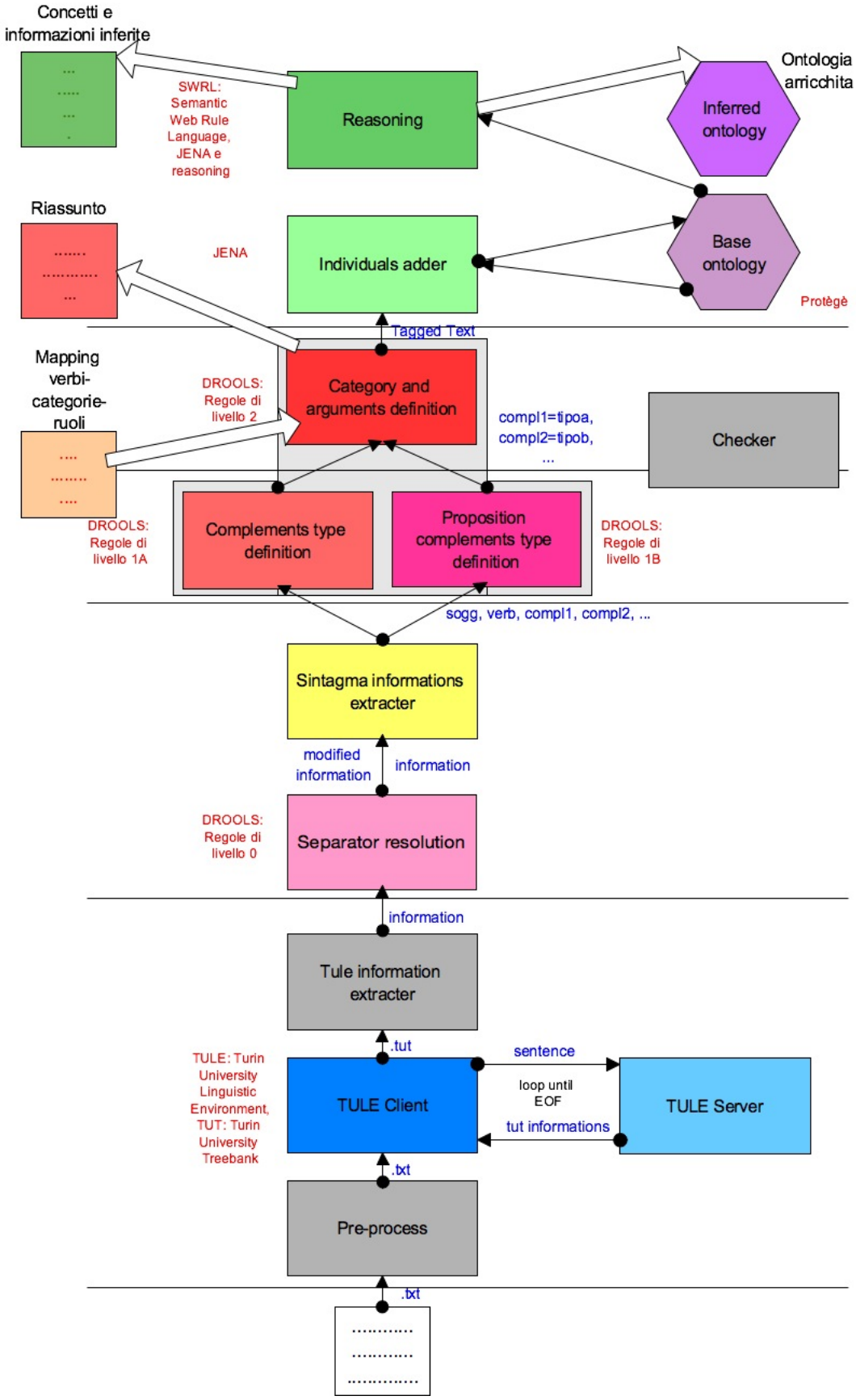

Figura 5.3: Schema generale del sistema.

tramite **regole DROOLS**, si procede all'identificazione dei particolari frame definiti nel file di mapping verbi-ruoli-categorie. Tale operazione, descritta in dettaglio nei paragrafi successivi, viene eseguita tramite diversi livelli di regole che infine portano alla definizione delle categorie associate a ciascun verbo e ai ruoli associabili ad ogni attore del testo analizzato. Dopo questa operazione si ottiene quindi un testo ''taggato" in cui i tag associati alle singole parole del testo sono la categoria del verbo e i ruoli dei diversi attori; si ha come output del processo di regole una particolare taggatura del testo basata sui frame.
I tag particolari relativi ai ruoli precedentemente trovati sono quindi indispensabili, insieme alle categorie dei verbi, per la fase successiva; in essa si aggiungono, all'interno dell'ontologia di base, le parole associate ai ruoli come individui che apparterranno alle classi definite dai ruoli stessi e le categorie dei verbi che rappresentano quindi le proprietà dell'ambiente ontologico.
Una volta individuati quindi ruoli e categoria del verbo di ogni singola frase presente nel testo, si può immaginare la situazione di Figura 5.4 in cui si ha quindi una corrispondenza stretta tra classi dell'ontologia e ruoli del testo, tra proprietà dell'ontologia e categoria del verbo ed infine tra gli individui dell'ontologia e i termini associati ai ruoli. E' proprio grazie a questa corrispondenza che si esegue l'arricchimento ontologico.
L'esempio mostra come i ruoli ATTORE e OGGETTO corrispondano alle classi dell'ontologia ATTORE e OGGETTO, mentre AGIRE viene identificato come la proprietà che lega le due classi; LA MAMMA e LA TORTA sono dei nuovi individui appartenenti alle due classi descritte sopra.

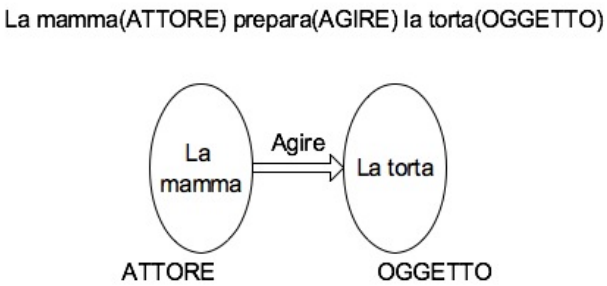

Figura 5.4: Esempio di corrispondenze tra testo ''taggato'' e ontologia.

L'ultima fase fondamentale, eseguita grazie alle regole SWRL, è la componente di **reasoning**, nella quale si eseguono delle inferenze sui concetti precedentemente inseriti nell'ontologia, cercando quindi di trovarne di nuovi. L'obiettivo

è quindi l'arricchimento ontologico e la generazione di nuove frasi importanti per il contesto considerato.
Si sono evidenziati, in particolare, tramite le frecce bianche, i tre output fondamentali del processo e il file di mapping; così facendo si è voluto sottolineare in che fase essi vengono effettivamente prodotti e utilizzati all'interno del sistema complessivo.
Dopo aver eseguito una trattazione macroscopica sul processo si descrivono in dettaglio, nei prossimi tre paragrafi, le tre componenti qui solo citate e raccontate in modo globale. Si andrà quindi nel merito dei macro-processi TULE, regole ed infine della parte più semantica relativa all'ontologia.

5.2.1 TULE

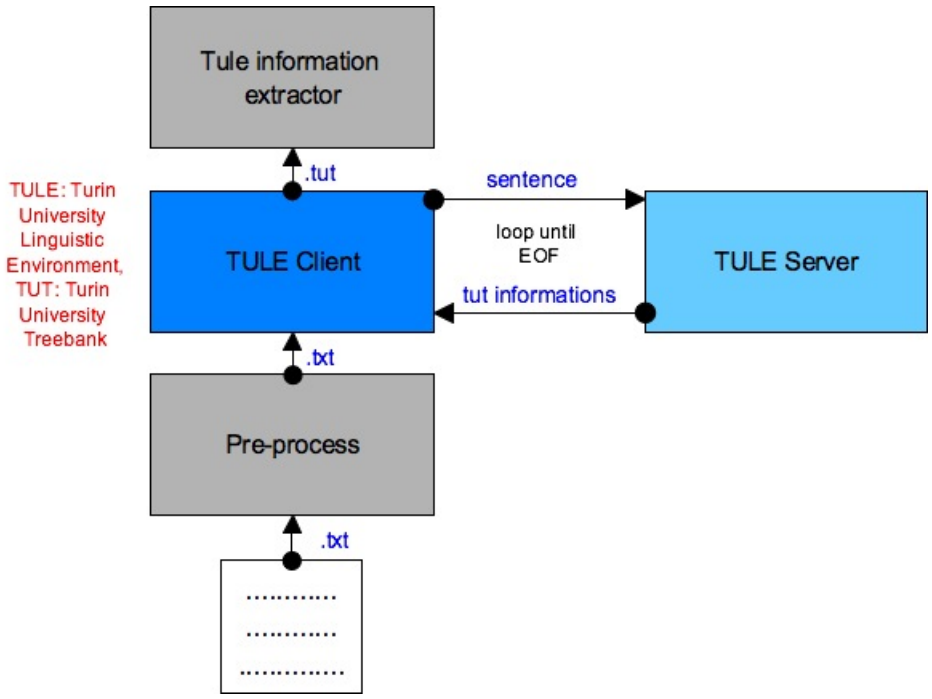

Figura 5.5: Schema della parte TULE.

Il **primo macro-processo** che si incontra, a partire dal testo, è quello relativo a TULE che si articola in quattro blocchi fondamentali.
Dopo aver scelto il testo che si vuole analizzare, si svolge su di esso una prima elaborazione: **Pre-process**. Tale blocco consente di eliminare eventuali problematiche che il testo possiede e che, date come input a TULE, possono riportare delle anomalie. Un esempio classico di queste modifiche sono gli accenti;

le lettere accentate devono essere tradotte in semplici parole con apostrofi in modo tale che TULE le comprenda e le analizzi. Dal blocco pre-process si ha ancora un output di tipo file di testo txt e, come spiegato nell'introduzione del processo generale, tale blocco è stato rappresentato in grigio semplicemente perché è un blocco generale, che non fa uso di nessun sistema esterno complesso.

I successivi blocchi; **TULE Client** e **TULE Server**, sono i componenti di base per la parte sintattica di elaborazione delle parole e delle frasi spiegata nel Capitolo 2 ed esemplificata in Figura 2.4. Riguardo al sistema TULE si è svolta un'ampia trattazione nella parte riguardante le tecnologie usate, ma si vuole qui spiegare come questo strumento è stato utilizzato nel processo del sistema. TULE, analizzando una frase alla volta, crea il file .tut che possiede tutte le informazioni morfologiche, i lemmi, i POS e le dipendenze relative alla totalità del testo. Il file .tut viene quindi passato al livello successivo che andrà ad estrapolare dal file tutte le informazioni essenziali per le elaborazioni successive.

L'ultimo blocco della catena TUT è infatti il **Tule information extractor** che, come evidenziato sopra, consente l'estrazione di ogni informazione utile e la sua memorizzazione in appositi oggetti java (Figura 5.6). Anche tale blocco, come il Pre-process è del tutto generico, per questo è mostrato in grigio.

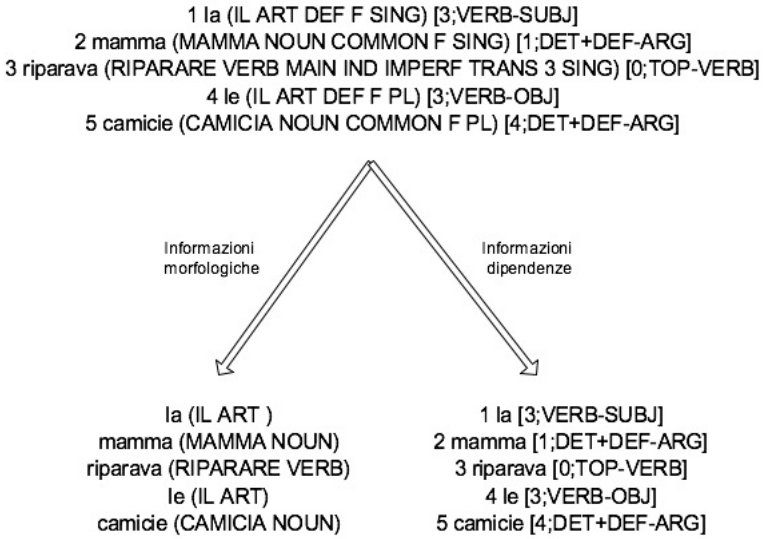

Figura 5.6: Schema di derivazione delle informazione eseguito dal blocco Tule information extractor.

5.2.2 DROOLS

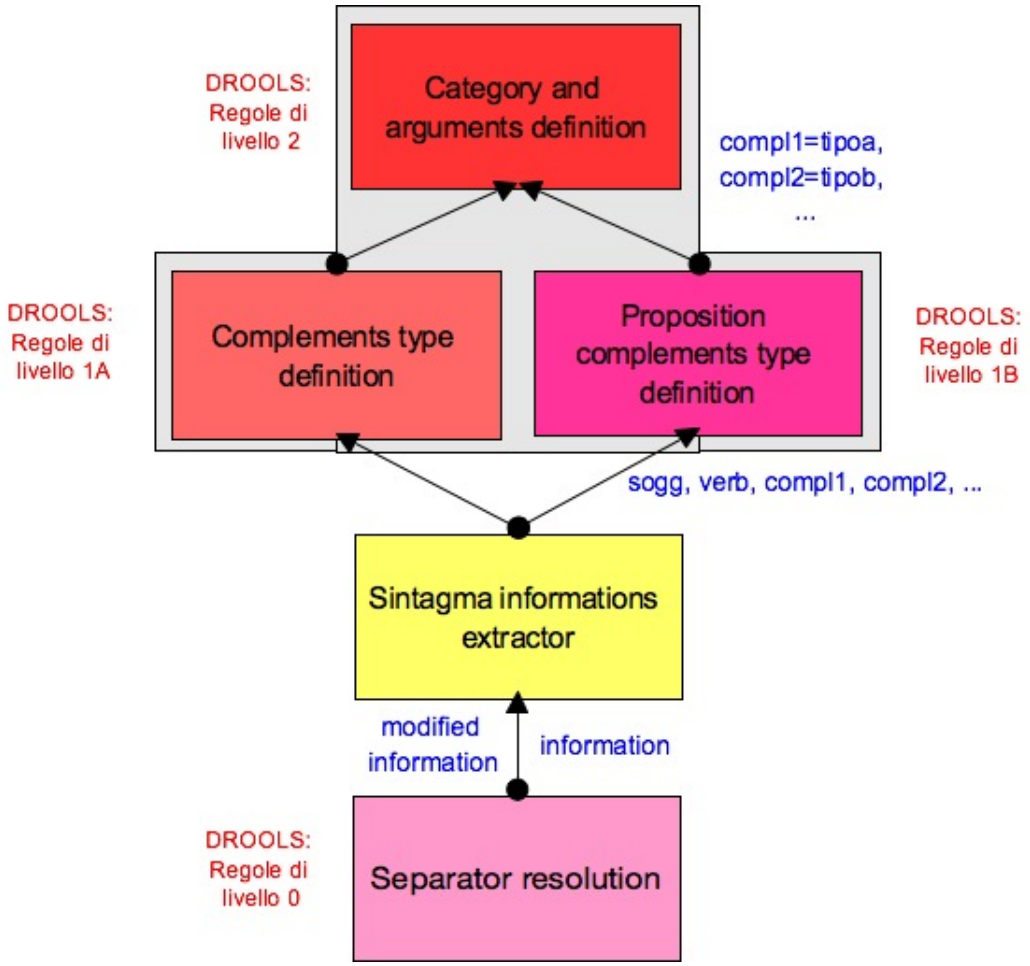

Figura 5.7: Schema della parte regole.

Si descrive ora nel dettaglio il **secondo macro-processo** che corrisponde all'insieme di regole, appartenenti a diversi livelli di progettazione, che consentono, come output finale, di derivare le differenti categorie e i diversi ruoli presenti nel testo esaminato.
Come è possibile osservare dallo schema di Figura 5.7 si possono distinguere **tre componenti fondamentali** del processo:

- Regole per la definizione di complementi inter-frasi (rosa e rosa chiaro);
- Regole per l'identificazione dei complementi intra-frase (rosso e rosso chiaro);
- Blocco di estrazione di informazioni di sintagma (giallo).

Esiste però un'ulteriore classificazione dei componenti di questo macro-processo in base ai livelli di regole, e quindi in base al flusso di esecuzione delle stesse.
Il primo blocco, **Separator resolution**, permette di separare le frasi che contengono un elemento appunto di separazione, riportando il soggetto principale nelle diverse frasi ottenute. Se si hanno quindi delle frasi complesse legate fra loro, ad esempio, dalla congiunzione ''e'', tali frasi saranno separate ottenendone due nuove nelle quali saranno presenti tutti gli elementi del frame canonico come soggetto, verbo e complementi.
Il secondo passo, **Sintagma informations extractor**, consente di estrapolare, da ogni frase, le informazioni sintagmatiche, ovvero permette la separazione delle componenti nominali, verbali e di complemento, considerandole come entità globali. Dopo aver svolto questo passo non si lavorerà quindi con le singole parole ma piuttosto con le parti nominali verbali e i gruppi di complementi.
Nella terza fase si hanno due entità che vengono svolte parallelamente: **Complements type definition** e **Proposition complements type definition**. Entrambi i blocchi contengono delle regole che consentono, data una componente complemento, di estrarre la tipologia del complemento analizzato. La differenza fondamentale tra le due risiede nel fatto che, mentre il blocco di sinistra estrae complementi canonici come ad esempio complementi di modo, tempo, luogo ecc., l'altro estrae dei complementi non comuni che si basano sulla presenza di parole di legame con altre frasi, come ad esempio la presenza di perché, poiché, sebbene, ecc.
Bisogna, a questo punto, evidenziare anche la differenza fondamentale che si ha tra i due blocchi rosa chiaro e rosa: mentre il primo lavora sulla separazione delle frasi, il secondo considera la frase subordinata come un complemento aggiuntivo della frase principale.
Si riporta in Figura 5.8 un semplice esempio chiarificatore sulle operazioni svolte dai blocchi corrispondenti ai livelli zero e uno del processo, esplicitando in particolare dove ciascun blocco, e quindi ciascuna tipologia di regola, ha potere di azione.
Dopo aver esemplificato in dettaglio i livelli di regole zero ed uno, si passa ora alla descrizione dell'ultimo blocco: **Category and arguments definition**. Gli ingressi a questo blocco sono semplicemente la lista, per ogni frase, delle parti nominali, verbali e dei diversi complementi con il tipo del complemento

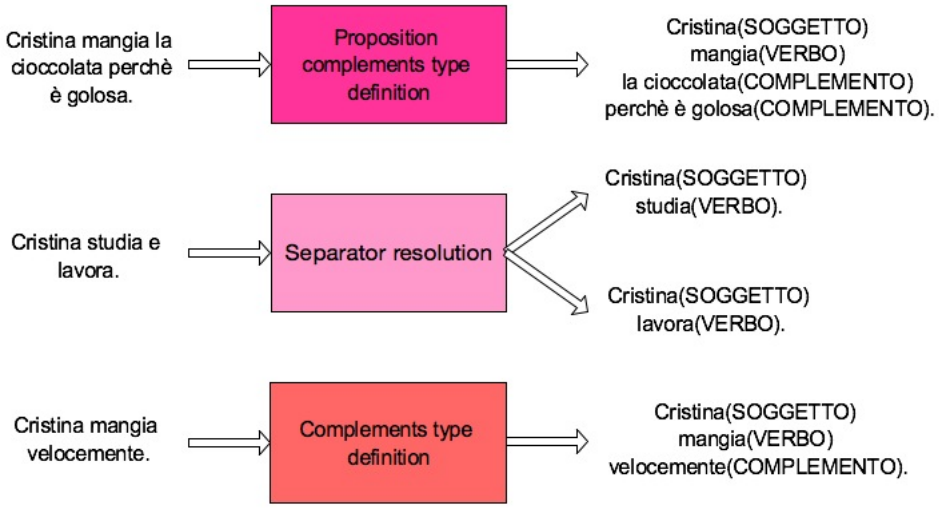

Figura 5.8: Esemplificazione di input e output dei principali blocchi dell'area DROOLS.

stesso associato. A questo punto quindi, le regole di livello 2, consentono, a partire dagli input, e grazie al file di mapping, di determinare innanzitutto la categoria del verbo e successivamente anche i ruoli delle diverse componenti della frase.
Alla fine di questo maxi-processo abbiamo quindi in mano le informazioni concettuali precise di ogni singola frase significativa del testo, riguardo il dominio considerato; l'output del blocco è quindi il riassunto. Ovviamente tutte le frasi che non hanno un corrispettivo nel file di mapping non vengono analizzate e vengono quindi scartate poiché considerate non rappresentative del dominio esaminato; le altre invece costituiranno il riassunto. La selezione delle frasi che formeranno il riassunto è eseguita a valle delle regole di livello 2 e non viene svolta tramite delle regole dirette ma, si basa essenzialmente sul file di mapping categorie-verbi-ruoli.

5.2.3 L'ontologia

L'output del macro-blocco di regole arriva come nuovo ingresso dell'ultima parte del processo, la parte dedicata all'**arricchimento ontologico**.
Il primo sistema che si incontra è l'**Individuals adder** che, prese in ingresso le componenti categoria e ruoli delle singole frasi, va ad aumentare l'informazione ontologica contenuta nell'ontologia di base. In questa prima fase vengono quindi aggiunti individui all'ontologia che, grazie alla presenza delle diverse

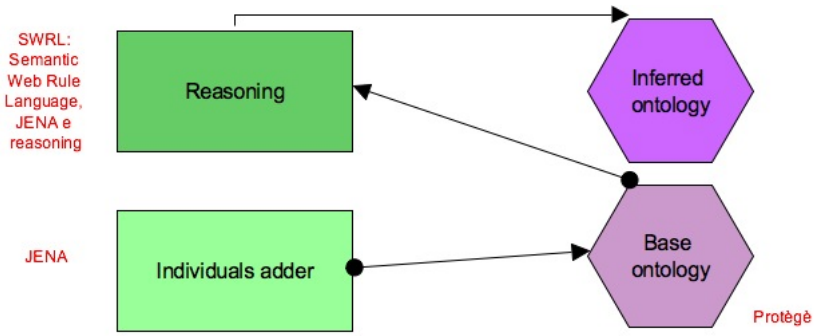

Figura 5.9: Schema della parte ontologica.

classi e proprietà, vengono legati fra loro rispettando le relazioni di dominio esistenti per il settore specifico analizzato.
Dopo aver eseguito l'operazione di arricchimento di individui tramite JENA e i suoi metodi, si passa alla fase **Reasoning**. In tale fase sono coinvolti, non solo l'ontologia precedentemente arricchita, ma anche il reasoner generico di JENA e le regole SWRL, necessarie alla produzione di nuove informazioni inferibili da quelle già presenti nell'ontologia arricchita. Dopo aver svolto quest'ultimo passo, tramite la serializzazione del nuovo modello ontologico inferito, si ottiene una nuova ontologia che esprime al meglio il testo considerato poiché, oltre a contenere i concetti salienti del dominio, contiene anche i concetti derivabili da essi. I nuovi concetti inferiti ampliano così le conoscenze di base che si deriverebbero dall'analisi critica eseguita da un lettore umano.

A questo punto è facile capire le motivazioni che sono alla base della decisione di utilizzare due tipologie di regole, SWRL e DROOLS. Nello schema di Figura 5.10 si mostrano gli output dei due macro-blocchi in modo tale da rendere più esplicito il loro utilizzo e la loro applicazione all'interno del sistema.

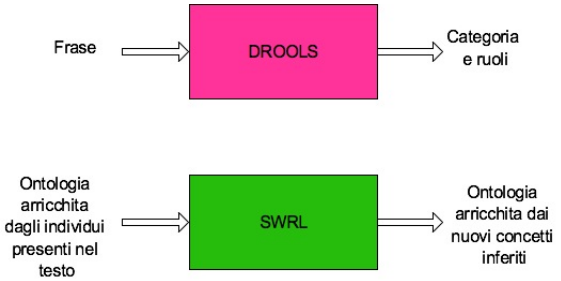

Figura 5.10: Schema di comparazione tra DROOLS e SWRL.

5.3 Applicazione di un esempio al processo

Si vuole qui descrivere il processo tramite un esempio più applicativo. L'esempio è stato appositamente definito su un testo molto corto per semplificare la grafica e la spiegazione stessa.
In basso troviamo il testo da analizzare. Come output del primo blocco troviamo l'analisi svolta dal sistema TULE e, quindi, il file .tut.
Dopo aver eseguito i primi blocchi della fase DROOLS si ottiene la suddivisione della singola frase in parti nominali e verbali e la descrizione di ognuna si esse in base al fatto che si tratti di soggetto, verbo oppure complemento. In questa fase si determina anche il tipo di complemento, come si può notare dall'esempio stesso.
Con l'ultimo blocco di DROOLS entra in gioco il file del mapping che consente di stabilire quali sono le frasi da considerare, in base al verbo e, soprattutto, consente di ‘‘taggare" il testo descrivendo i singoli elementi. La definizione degli elementi e la loro mappatura vengono eseguiti in base ai ruoli e alla categoria presenti nella riga di mapping del verbo e del frame definito dalla frase considerata.
In questo modo, all'uscita del macro-blocco DROOLS si ottiene il riassunto e gli individui da inserire nell'ontologia di base.
Dopo aver inserito gli individui all'interno dell'ontologia è possibile eseguire le regole di inferenza SWRL e derivare nuovi concetti, presenti a sinistra nel diagramma, e l'ontologia arricchita, presente a destra.
In questo modo si sono ottenuti i tre output fondamentali: riassunto, concetti inferiti e ontologia arricchita.

5.4 Il prototipo

Dopo aver spiegato nel dettaglio l'architettura impiegata nella realizzazione del prototipo, si vuole qui entrare nel dettaglio dell'implementazione.
Per la realizzazione del prototipo si è deciso di definire un'applicazione implementata con java. Si è pensato di impiegare tale linguaggio poiché esso risulta essere oramai ampiamente utilizzato e poiché consente inoltre il facile interfacciamento con altre realtà, come il dominio ontologico oppure i sistemi a regole, che sono fondamentali per la definizione del modello e successivamente

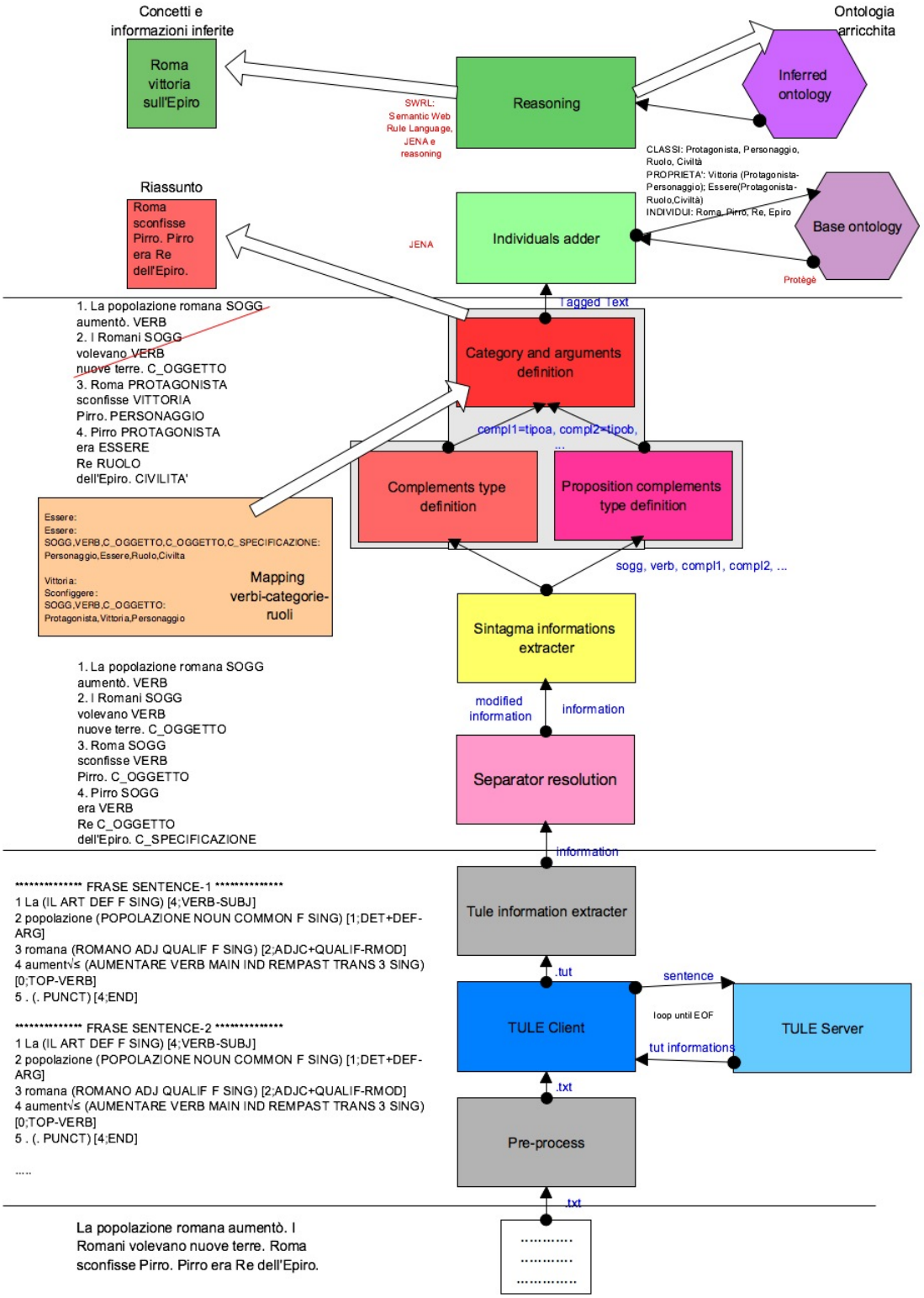

Figura 5.11: Esempio per chiarire la descrizione del processo.

Figura 5.12: I comandi che possono essere impartiti dall'utente.

del prototipo stesso. Java infatti consente l'utilizzo di plugin e librerie specialistiche di un determinato ambito. In questo caso è stato utile l'utilizzo di Jena per interfacciarsi con l'ontologia, Drools per utilizzare i sistemi a regole ed infine anche TULE per le prime fasi di elaborazione del testo.
Come si può osservare dalla Figura 5.13, l'interfaccia è molto **semplice** ed **intuitiva**. Si è fatta questa scelta semplicemente perché il sistema potrebbe, in futuro, essere impiegato nell'ambito dislessia e disabilità; un aspetto visuale di immediata visibilità facilita la comprensione da parte sia di un utente generico, sia diversamente abile.
Tramite l'utilizzo del primo pulsante **Apri** è possibile selezionare il testo che si vuole analizzare. Tale comando attiverà la chiamata a TULE che eseguirà direttamente l'analisi sintattica del testo. A questo punto, il primo macro-processo TULE mostrato in precedenza sarà stato eseguito.
Utilizzando invece il secondo pulsante **Testo** verrà mostrato il testo aperto dall'utente sul video. Tale informazione verrà presentata nella prima area.
Dopo aver svolto l'operazione di caricamento del testo si può quindi procedere alla definizione del **riassunto** vero e proprio. Il tool esegue, a questo punto, l'analisi del macro-processo DROOLS e riporta quindi in output all'utente, all'interno della seconda area, il testo riassunto. Tale sommario riporta solamente le porzioni di frasi che sono state riconosciute come aventi un verbo appartenente alle categorie precedentemente definite dal mapping.
Il processo che definisce il riassunto non è quindi una semplice analisi frequenziale e non si ricorre nemmeno ad una funzione di importanza, ma piuttosto a considerazioni legate alla struttura del testo, al dominio e al mapping che consente infatti di estrapolare, non solo i verbi significativi ma anche i ruoli che si accompagnano ad essi. Questa idea è fondamentale per poter eseguire l'arricchimento ontologico e l'inferenza di nuovi concetti.
Il passo successivo è l'inferenza. L'utente può derivare i nuovi concetti premendo sull'ultimo pulsante **Inferenza**; il risultato, e quindi i nuovi concetti inferiti, sarà visualizzato nell'ultima area di testo. In questa fase viene esegui-

to il macro-blocco di arricchimento ontologico e SWRL che, una volta definite le regole di reasoning, permettono insieme di derivare l'ontologia dal testo e inferire conoscenza.

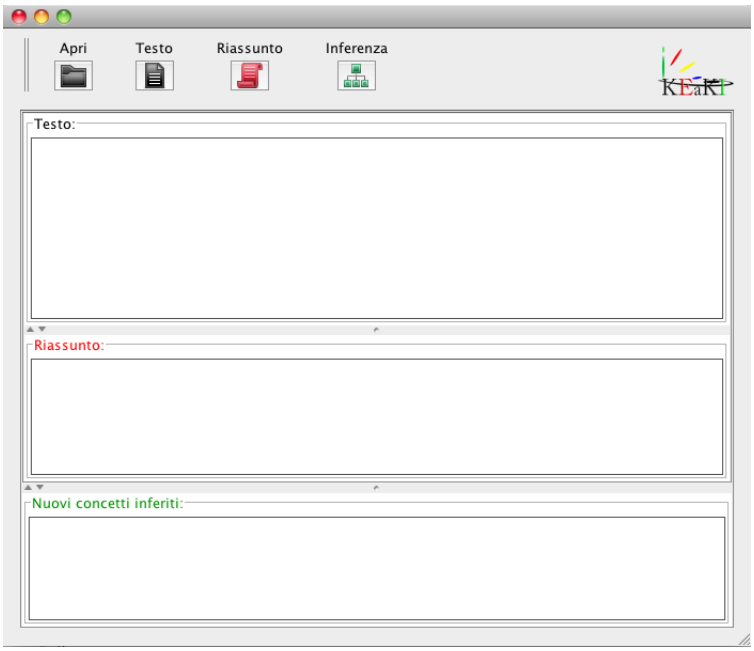

Figura 5.13: Interfaccia grafica del prototipo.

5.5 Un esempio

Si vuole ora presentare un esempio applicativo. Per questo esempio si è scelto un testo di storia tratto da un libro di quinta elementare.
Come si può notare dalla Figura 5.14 l'utente ha inserito il testo, ha eseguito l'analisi che ha portato alla definizione del riassunto e ha, infine, eseguito le regole di inferenza dei nuovi concetti, a partire dal riassunto stesso.

Dopo aver ottenuto il riassunto e i nuovi concetti inferiti si può osservare anche come, l'ontologia di partenza, anche se inizialmente vuota, si è popolata di concetti ontologici. In particolare, come è mostrato in Figura 5.16, si è ottenuta un'ontologia che ha per classi i ruoli precedentemente individuati e per proprietà le categorie dei verbi presenti nel riassunto generato dal prototipo.

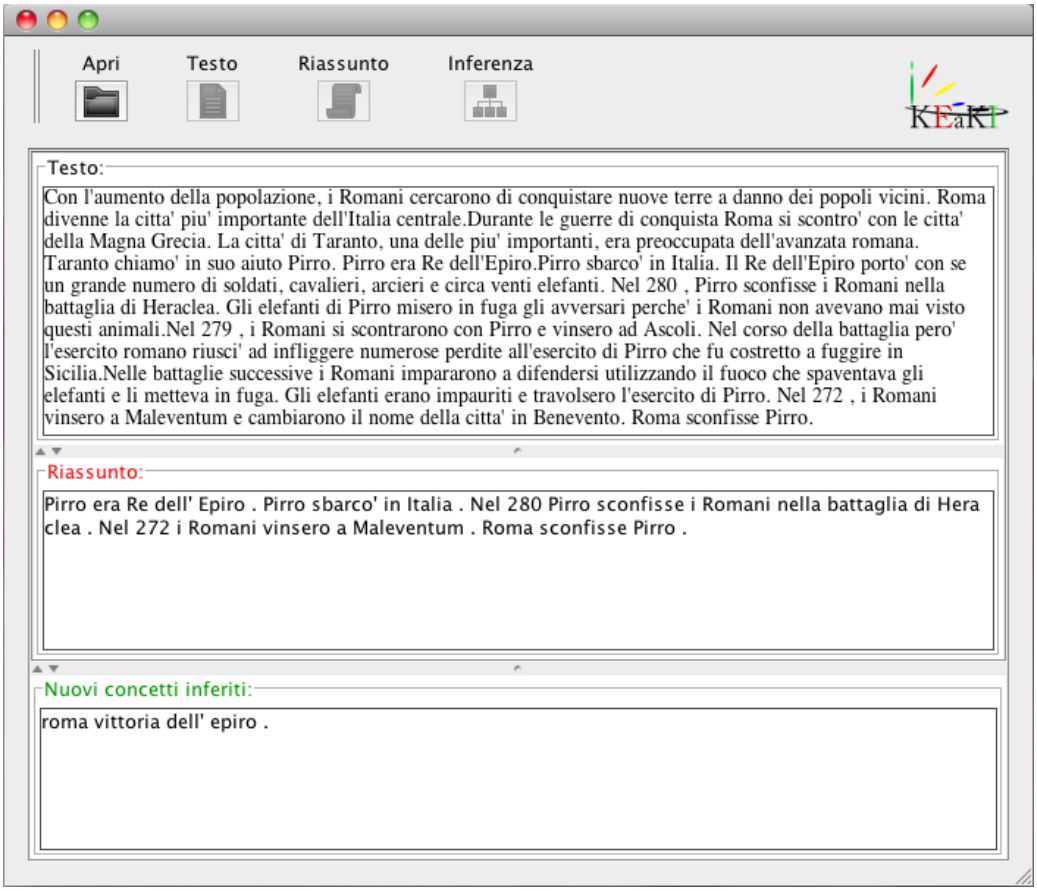

Figura 5.14: Esempio applicativo.

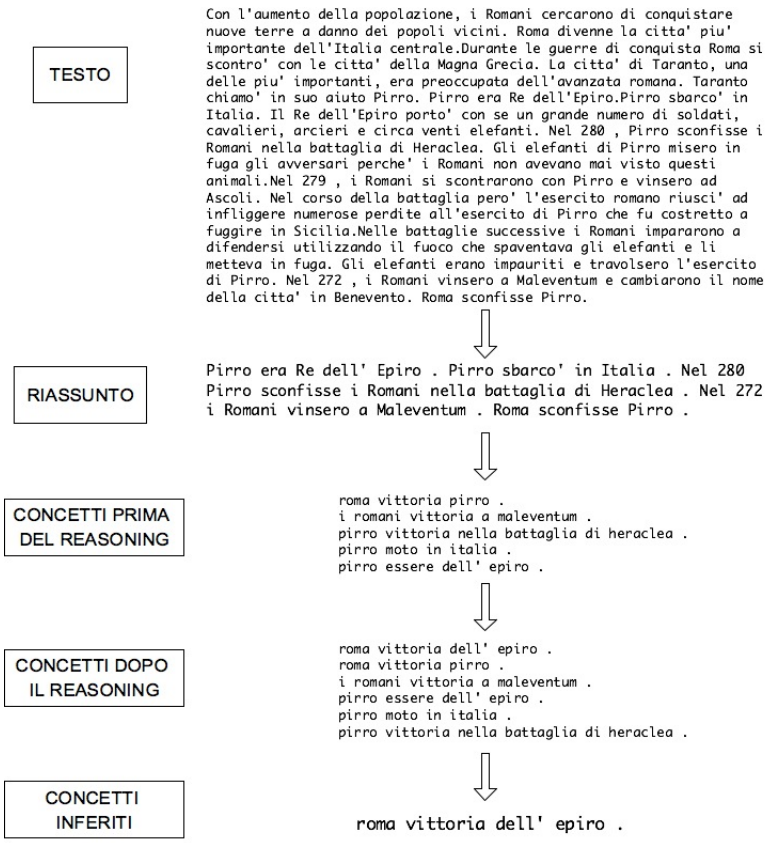

Figura 5.15: Esempio di creazione del riassunto e di derivazione di concetti tramite reasoning.

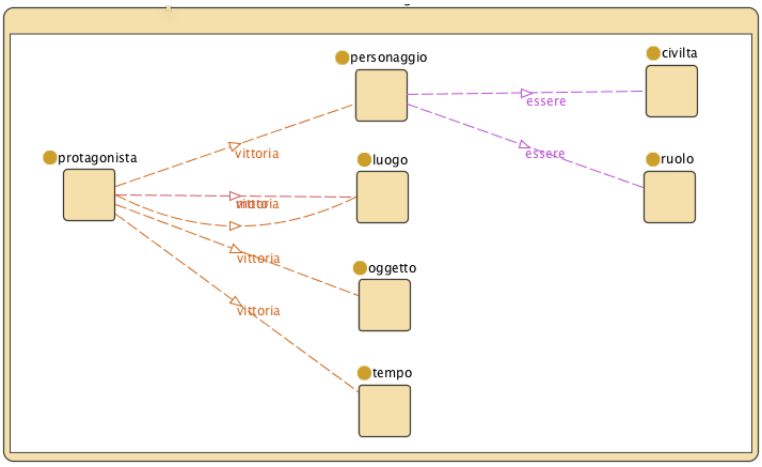

Figura 5.16: Grafico dell'ontologia con classi e proprietà derivate dal testo.

Capitolo 6

Validazione del modello

Dopo aver affrontato la descrizione del modello e del processo, che costituiscono la base del prototipo vero e proprio, ci si concentra ora sulla valutazione del sistema. Si tratterà quindi, in questa fase, la validazione del modello, rispetto alla creazione del riassunto.
Nel presente capitolo si affronterà la tematica della validazione, descrivendo inizialmente **le metriche** utilizzate, che nell'insieme offrono un'ampia visione del sistema, dei suoi punti di forza e delle sue criticità. Saranno descritte in particolare le quattro metriche principali adottate. Si tratteranno le classificazioni statistiche Precision e Recall impiegate in larga misura nell'information retrieval. Si descriverà anche la metrica di accuratezza ed infine si applicherà la metrica che descrive la percentuale di restrizione del testo riassunto rispetto al testo in input al sistema.
In secondo luogo si andrà a descrivere il **percorso metodologico** che ha portato alla validazione del nostro sistema. Si esemplificheranno quindi le decisioni prese in merito al metodo di valutazione impiegato; tale metodologia verrà spiegata nel dettaglio in modo da sottolineare come si è affrontato e risolto il problema di determinazione del riassunto ideale per il modello costruito.
Si mostreranno infine **i risultati** ottenuti, che saranno descritti e commentati per evidenziare le cause delle problematiche presenti nel modello o nel prototipo e le aree di miglioramento. Verranno infatti definite **le problematiche** in base a differenti livelli, creando dei gruppi di errori dovuti principalmente agli strumenti utilizzati, al nostro modello, oppure ancora al processo di estrazione di informazioni e alle regole impiegate per la definizione delle categorie.

6.1 Le metriche

Si sono applicate e calcolate diverse metriche utilizzando i risultati ottenuti dalle simulazioni eseguite su un certo insieme di testi di storia. Questo processo è fondamentale al fine di meglio comprendere come lavora il prototipo e che tipo di risultati si possono ottenere. Dopo l'esecuzione della validazione è infatti possibile ragionare sull'efficienza e sull'utilità del tool progettato.
Le due principali metriche impiegate sono **Precision** e **Recall**:

$$Precision = \frac{|\{concetti\ attinenti\} \cap \{concetti\ recuperati\}|}{|\{concetti\ recuperati\}|} \tag{6.1}$$

$$Recall = \frac{|\{concetti\ attinenti\} \cap \{concetti\ recuperati\}|}{|\{concetti\ attinenti\}|} \tag{6.2}$$

Le definizioni delle equazioni 6.1 e 6.2 possono anche essere espresse come segue:

$$Precision = \frac{veroPositivo}{veroPositivo + falsoPositivo} \tag{6.3}$$

$$Recall = \frac{veroPositivo}{veroPositivo + falsoNegativo} \tag{6.4}$$

Tali valori consentono di determinare se ogni risultato recuperato da una ricerca è attinente, per quanto concerne la precision, e il numero dei concetti attinenti che sono stati recuperati dalla ricerca, tramite la recall.
Un'altra metrica fondamentale è l'**Accuratezza**, che è stata qui impiegata per riuscire a percepire il grado di corrispondenza del dato teorico, derivabile da una serie di valori misurati, con il dato reale o di riferimento, ovvero la differenza tra valor medio campionario e valore vero o di riferimento:

$$Accuratezza = \frac{veroPos + veroNeg}{veroPos + veroNeg + falsoPos + falsoNeg} \tag{6.5}$$

Bisogna sottolineare, a questo punto, la differenza fondamentale che c'è tra precision e accuratezza. Mentre la precision indica la varianza rispetto al valore medio campionario, l'accuratezza evidenzia invece quanto i valori sono vicini all'obiettivo.
Infine si è ritenuto utile andare ad eseguire anche un calcolo percentuale sulla **restrizione del testo**. Si è voluto quindi determinare quanto il testo riassunto dal sistema è più ristretto rispetto al testo originario definendo quindi la metrica di percentuale di riduzione del testo:

$$Riduzione = \frac{concettiEstratti \cdot 100}{concettiTotaliPresentiNelTesto} \tag{6.6}$$

6.2 Metodologia

Come mostrato nel Capitolo 2, esistono, in letteratura, diversi metodi per la valutazione dei sistemi per la creazione automatica di riassunti. In particolare, la definizione di cosa sia effettivamente il riassunto ideale è un problema frequente e ancora aperto ad idee ed innovazioni. Può infatti risultare semplice ed immediata l'esecuzione del riassunto da parte di una singola persona, che possa essere poi considerato come base per la valutazione. Ci si è accorti però che un solo utente non bastava ma erano necessari diversi utenti che definissero il proprio riassunto, in modo tale da poter avere delle soluzioni e delle risposte sempre migliori e generali. Tale tipologia di sistema però aveva alcuni problemi: l'effettivo confronto dei diversi riassunti e la definizione di unità valida di base nel testo analizzato. Nel tempo quindi questo metodo è stato modificato e si sono delineate oggi delle classificazioni di metodi più articolate e complesse. Oggi viene fatta una netta distinzione tra valutazione **intrinseca** ed **estrinseca** [15]. La prima cerca di valutare il riassunto in base principalmente alla sua coerenza e ricchezza di informazioni. La seconda cerca di individuare quale sia l'impatto dell'utilizzo di tale riassunto in altri sistemi.
I metodi intriseci si suddividono a loro volta in metodi di valutazione basati sulla comparazione con dei riassunti ideali e metodi basati sulla comparazione con il testo originario.
I metodi **basati sul testo di input** sono generalmente basati su una semplice analisi della frasi o delle parole presenti nel testo originario che si possono ritrovare poi nel riassunto. Tali metodi si suddividono in semantici e sintattici, proprio sulla base dell'utilizzo di frasi o di semplici parole. Un esempio di metodo basato sulla comparazione del riassunto con il testo originale è [24], nel quale si applica una metodologia di tipo semantico.
Per quanto riguarda invece le **metodologie che si basano sul confronto con un riassunto modello** si suddividono i due testi, quello riassuntivo e quello originale, in unità semplici minime e si calcola il fattore di coincidenza tra le due componenti.
Per la validazione si è deciso di utilizzare il secondo metodo descritto, andando quindi ad eseguire un paragone tra un riassunto ideale e il riassunto che invece viene definito dal prototipo. Dopo aver analizzato la letteratura in merito alla definizione di una metodologia per determinare il riassunto ideale, si è deciso di

applicare il sistema a **"piramide"** [18], che consiste nel definire una piramide che rappresenta appunto il riassunto ottimo per il caso in esame.
Per applicare il metodo a piramide bisogna innanzitutto definire cosa si intende come informazione minima per la valutazione del riassunto. La *Summary Content Unit*, che è stata impiegata per questa valutazione, consiste in una unità concettuale definita da una frase semplice singola. Se si hanno quindi frasi subordinate o collegate tra loro con congiunzioni di un qualsiasi tipo, si considerano le spezzate di queste frasi, aventi un singolo significato intrinseco. Se consideriamo ad esempio la frase "La storia va studiata e non va semplicemente letta" si possono individuare due particolari SCU:

- La storia va studiata;
- La storia non va semplicemente letta.

Dopo aver fatto questa precisazione si può descrivere il modello a piramide. Tale modello si basa sull'idea di far eseguire il riassunto da parte di più lettori. Dopo aver raccolto diversi riassunti si esegue un'analisi sulle SCU presenti nel testo di partenza e nei vari riassunti; si elencano tutte le SCU presenti nel testo originale e si attribuisce un punteggio a ciascuna SCU, calcolato sommando il numero di volte in cui compare tale SCU nel riassunto svolto dal singolo individuo. Se ad esempio SCU1 compare in 5 riassunti, eseguiti da 5 persone differenti, esso avrà un punteggio pari a 5, se SCU2 è presente in 3 riassunti avrà un punteggio pari a 3 e così via.
Dopo aver calcolato il punteggio di ogni SCU si riempie una piramide con le SCU corrispondenti; al vertice della piramide verranno inserite le SCU con maggiore punteggio, mentre alla base quelle con punteggio pari ad uno. Un esempio è mostrato in Figura 6.1.

Dopo aver costruito la piramide e averla inizializzata con i valori delle differenti SCU è possibile eseguire su di essa diverse tipologie di calcoli, applicando metriche differenti che utilizzano somme pesate oppure semplicemente le definizioni di Precision e Recall classiche.
In particolare si è deciso di applicare le definizioni di Precision e Recall mostrate in [18]:

$$Recall = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=1}^{n} T_i} \tag{6.7}$$

$$Precision = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=0}^{n} D_i} \tag{6.8}$$

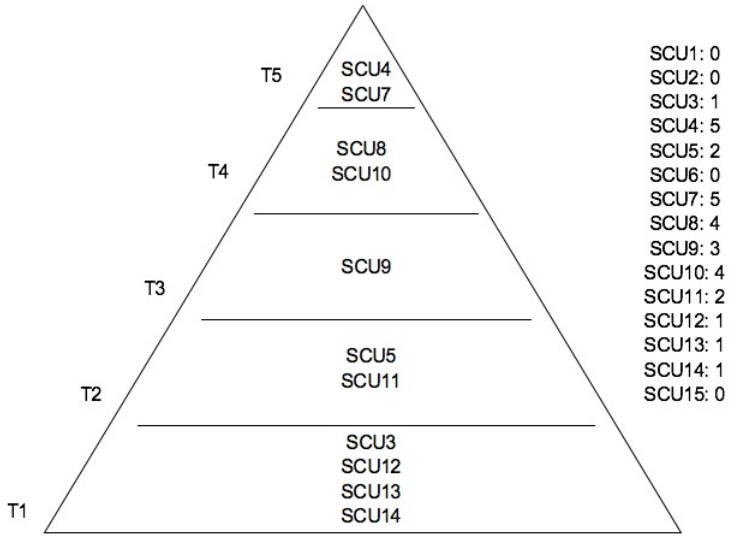

Figura 6.1: Esempio di piramide.

Dove D_i sono il numero di SCU nel riassunto calcolato dal prototipo che compare nel livello T_i della piramide. D_0 è invece il numero delle SCU del riassunto prodotto dal sistema che non compaiono nella piramide, ovvero sono al livello zero della stessa.

Per il testing si sono utilizzati dieci **testi di storia** di tipologia dinamica tratti da libri di testo delle scuole elementari. Il numero di parole medie dei testi è 228, il numero medio di SCU nei diversi testi è 29; quindi mediamente ciascuna SCU è formata da 8 parole. I testi contengono prevalentemente descrizioni di battaglie nelle quali i verbi sono principalmente di azione, corrispondono all'idea di atti dialogici illocutionary e sono anche in numero limitato. Il sistema può quindi essere facilmente applicato a quelle tipologie di testi che hanno un numero di verbi significativi limitato ma che hanno anche dei verbi particolari e di utilizzo non comune o canonico. Se infatti si volesse eseguire l'analisi di testi comuni e del tutto generici si avrebbero dei risultati decisamente poco precisi poiché il riassunto svolto dal prototipo conterrebbe la quasi totalità delle SCU presenti nel testo di origine.

Per quanto riguarda invece le persone umane che hanno redatto i riassunti, si sono scelte **5 persone**, tra le quali 4 adulte e un bambino dell'età di dieci anni. Tra le 4 persone adulte ve ne è una con **certificazione di dislessia** e anche per quanto riguarda il bambino si è scelto un certificato dislessico. Questa scelta si è basata semplicemente sul fatto che il sistema può essere, in futuro, ampliato ed utilizzato da queste tipologie di persone, quindi si è ritenuto importante

iniziare già ad eseguire dei test anche con utenze differenti.

6.3 I risultati

Dopo aver svolto i diversi testing, aver collezionato i diversi risultati ed aver eseguito i calcoli delle metriche descritte sopra si sono ottenuti i risultati mostrati in Figura 6.2.

DOCUMENTO	Recall	Precision	Accuracy	%Restrizione
1	0,47	0,9	0,39	40
2	0,53	1	0,54	30,3
3	0,5	1	0,49	30,3
4	0,61	0,89	0,44	50
5	0,55	1	0,65	24
6	0,63	0,92	0,49	48,15
7	0,56	1	0,49	40
8	0,56	0,82	0,49	42,31
9	0,67	1	0,54	43,9
10	0,64	0,9	0,48	50
MEDIE	Recall	Precision	Accuracy	% Restrizione
	0,572	0,943	0,5	39,896

Figura 6.2: Risultati ottenuti dall'analisi di 10 testi.

In particolare i grafici 6.3 e 6.4 mostrano i risultati seguenti:

- La **Precision è, in media, molto elevata**, tende infatti ad uno. Tale valore esprime la bontà del sistema nel determinare i concetti importanti. Infatti ogni risultato recuperato dal prototipo è attinente al dominio considerato;
- La **Recall è invece mediamente sufficiente**. Questo indica che il sistema non recupera tutti i concetti che gli utenti invece ritenevano essere importanti;
- L'**Accuracy ha valori medi di 0,5**. Tale valore significa semplicemente che si hanno delle problematiche sistematiche che portano il sistema a non avere una grande accuratezza.

Si vogliono ora ricercare **le motivazioni** dei valori ottenuti dall'osservazione di tali risultati. Non avendo potuto, per questioni di tempo e per la limitatezza nell'analisi linguistica di base, descrivere una mappatura più vasta di verbi, si

è scelto di derivare un riassunto che fosse il più fedele possibile al dominio considerato, perdendo magari alcuni concetti che sono secondari. Tra essere più specifici possibili o estrapolare tutti i concetti importanti e semi-importanti si è scelta la prima strada. Si è deciso quindi di calcolare un riassunto più sintetico possibile tralasciando alcuni dettagli che invece gli utenti avrebbero potuto selezionare.
Proprio a causa di tale scelta si ha un valore di precision molto elevata a discapito di un valore comunque buono di recall. Si può infatti notare che il testo viene comunque mediamente ristretto del 40 percento e, considerando che i testi sono di livello elementare, è una buona percentuale di riduzione.

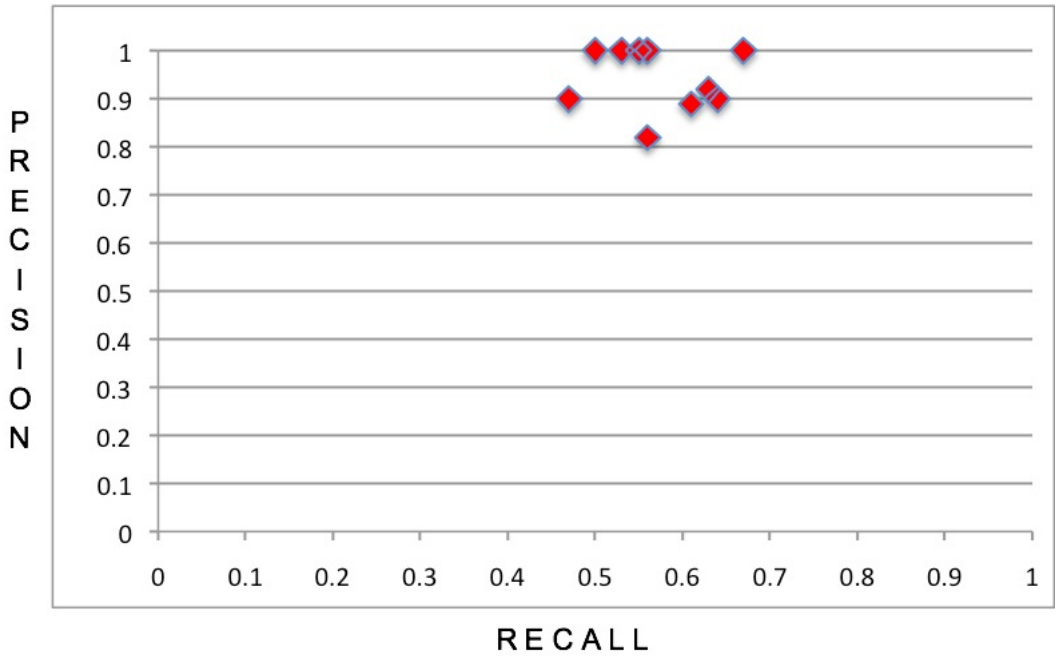

Figura 6.3: Grafico di Precision, in ordinata, e Recall, in ascissa.

Tali valori evidenziano comunque delle **problematiche** del modello o del sistema. Possiamo infatti raggruppare le problematiche in tre tipologie differenti:

- Problematiche dovute agli strumenti;
- Problematiche dovute al modello;
- Problematiche dovute al processo.

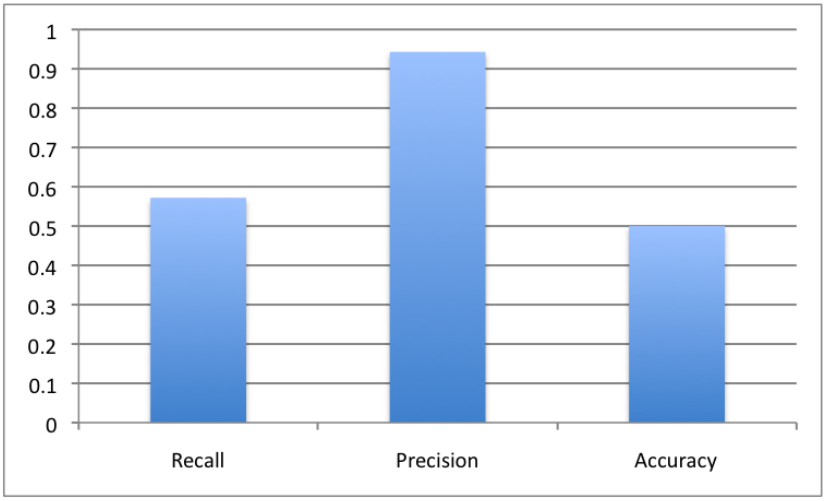

Figura 6.4: Grafico di Precision, Recall e Accuracy.

Rispetto agli strumenti si sono notati degli errori nell'output di **TULE**. Alcuni termini o accostamenti di parole non vengono esattamente riconosciuti quindi, già in partenza, ci si trova a dover utilizzare dei dati e delle informazioni non reali.

Rispetto al modello invece si hanno delle limitazioni dovute al fatto che il modello realizzato per il dominio in esame non è completo. Solo un esperto linguista potrebbe essere esaustivo in questo processo. Non avendo avuto la possibilità di svolgere un'**analisi linguistica** a tutto tondo non si è potuto delineare un modello e un mapping del tutto esaustivo.

Vi sono anche problematiche relative al processo che non prevede ad esempio la risoluzione di **coreferenza**[1]. Tale analisi, essendo ancora in fase di studio anche nella letteratura, non è stata qui presa in considerazione a causa della sua elevata complessità, che non avrebbe consentito, per motivi di tempo, di sviluppare il sistema nel suo complesso fino al livello più alto di inferenza.

[1]La coreferenza è un fenomeno in cui due diverse componenti del testo fanno riferimento alla stessa entità. L'esempio più comune è l'anafora pronominale in cui un pronome si riferisce ad un termine precedentemente espresso (per esempio, "La tua felpa è grigia, la mia è rosa." dove il secondo "La" indica la felpa).

Capitolo 7

Conclusioni e sviluppi futuri

7.1 Conclusioni

Lo scopo di questo lavoro di tesi è stato quello di definire un nuovo modello che permettesse l'estrazione dei concetti principali da un testo scritto in linguaggio naturale, l'arricchimento di un'ontologia di tali informazioni, l'inferenza di nuovi elementi mediante reasoning e, infine, la presentazione di una sorta di mappa mentale del testo originario.
Oltre alla metodologia, basata sull'utilizzo della linguistica e dei frame di verbi, si è progettato e implementato un prototipo che consente di eseguire i passi sopra elencati.
Per ottenere tali obiettivi si sono ricercate inizialmente informazioni sullo stato dell'arte relativo all'elaborazione del linguaggio naturale e a tutte le diverse metodologie applicate nel tempo per ottenere un riassunto; successivamente si sono cercate le tecniche che fanno uso di ontologie per ottenere riassunti ed eseguire del reasoning sulle informazioni che si hanno a disposizione.
Anche relativamente al lato più modellistico del problema, si sono svolte delle ricerche sui frame di verbi, per capire quali sono gli elementi da considerare quando ci si avvicina a questo mondo puramente linguistico.
Oltre al lato teorico, linguistico e modellistico, si è svolta anche una ricerca nell'ambito tecnologico, con lo scopo di trovare i sistemi più adatti per raggiungere gli obiettivi che ci si era prefissati. Per raggiungere dei buoni risultati sono state fatte delle ricerche sugli strumenti di NLP presenti in letteratura, sugli strumenti a regole e sulle tecnologie e linguaggi per la parte di reasoning

e inferenza. Grazie a queste ricerche si sono scoperti TULE, DROOLS, JENA e SWRL che sono stati alla base dell'implementazione del prototipo.
Infine sono stati fondamentali i testing e le prove svolte per la validazione del modello e del prototipo. Durante questa fase sono stati impiegati metodologie di valutazione e metriche che hanno permesso di ottenere risultati incoraggianti relativamente ad un insieme base di testi. I risultati ottenuti hanno infatti evidenziato i punti di forza e di debolezza del sistema, tenendo sempre presente che la metodologia proposta è di carattere del tutto generale, sia per quanto riguarda la lingua che il tipo di testo, grazie alla presenza della parametrizzazione del sistema stesso.
Si vogliono ora sottolineare i miglioramenti che la metodologia e il prototipo definiti nel presente elaborato possono offrire rispetto agli strumenti attualmente presenti in letteratura, presentati nel Capitolo 2. La caratteristica principale di KEaKI riguarda il significativo utilizzo di NLP e delle sue teorie; molti sistemi sviluppati in precedenza non fanno uso di tali metodologie, ma si basano su altri strumenti come la statistica. L'impiego di tecniche di NLP consentono di ottenere dei risultati più precisi e dettagliati proprio perché basati su un approccio linguistico. Il sistema qui presentato risulta essere di grande rilevanza letteraria poiché affronta, all'interno della stessa struttura e processo, sia la definizione del riassunto sia l'estrazione di ontologie dal testo; in altri studi questi risultati si dovrebbero derivare da due strumenti differenti che li producono separatamente. Un'ultima, ma non meno importante, miglioria, riguarda le decisioni prese in sede di implementazione; si è sviluppato un sistema modulare che consente di effettuare modifiche sia a livello di lingua utilizzata sia nel layer di dominio analizzato: ciò può essere fatto semplicemente modificando il livello TULE per la lingua, il livello SWRL e il file di mapping per quanto riguarda invece il dominio.

7.2 Sviluppi futuri

Il sistema nasce con questo elaborato di tesi e, quindi, è ancora un germoglio che fornisce un'idea, una metodologia e un prototipo di base, che però andrebbero ancora ampliati. Sono infatti interessanti diversi sviluppi che, in un futuro, potrebbero avere grande utilità per tutto il progetto.
Una prima possibilità è **l'ampliamento delle componenti parametriche**

al sistema. Si tratta quindi di aumentare le regole relative a DROOLS e il mapping verbi-ruoli-categorie, in modo da svolgere ulteriori valutazioni dei risultati del sistema e, in particolare, degli effetti di tale modifica sulle metriche precision e recall. Questo sviluppo porterebbe il sistema ad aumentare i concetti che estrae dal testo e, quindi, diminuirebbe la metrica di percentuale di riduzione del testo che, in testi grandi, potrebbe comunque mantenersi ad un buon livello. Studiare quindi le variazioni che si avrebbero utilizzando maggiori informazioni parametriche potrebbe essere uno spunto interessante.
Un secondo sviluppo è relativo alla fase di testing. Sarebbe infatti utile applicare il sistema a **testi più lunghi**, per capire come il prototipo reagisce e come effettivamente potrebbero cambiare i valori delle metriche statistiche qui applicate. Sempre relativamente alla validazione sarebbe utile svolgere un'**analisi di altri metodi di selezione e identificazione del riassunto ideale**. Nel presente elaborato è stato applicato il metodo a piramide ma, potrebbe essere ricercato e utilizzato un altro metodo. Dopo aver applicato diverse metodologie si potrebbero valutare i risultati ottenuti da queste per compararle tra loro e trovarne una di riferimento per studi futuri.
Un ulteriore sviluppo futuro potrebbe riguardare l'ultima fase del processo. Nel prototipo qui realizzato si è determinata una sorta di mappa mentale facendo uso dell'ontologia e della sua realizzazione grafica. Sarebbe utile definire una **nostra mappa concettuale** che non faccia uso della grafica ontologica. Per ottenere questo risultato basterebbe inserire un nuovo layer di livello più alto rispetto a quelli esistenti, in modo tale da ottenere il grafo dei concetti principali relativi al testo analizzato.
Infine, avendo qui applicato il sistema al solo dominio di storia, sarebbe interessante provare a definire le componenti parametriche per utilizzare il sistema in **altri contesti**. Sarebbe quindi interessante applicare il nostro modello e utilizzare il prototipo qui descritto per altre realtà.

Appendice A

Tabelle del modello

Mapping			
Categorie	**Verbi**	**Complementi**	**Ruoli**
Proprietà	Conquistare Occupare Invadere Ottenere Sconfiggere	(SOGG) (VERB) C_OGGETTO	Protagonista (VERB) Oggetto
		(SOGG) (VERB) C_OGGETTO C_SVANTAGGIO	Protagonista (VERB) Oggetto Danno
		(SOGG) (VERB) C_OGGETTO C_MODO	Protagonista (VERB) Oggetto Modo
		(SOGG) (VERB) C_MODO C_OGGETTO	Protagonista (VERB) Modo Oggetto
		(SOGG) (VERB) C_OGGETTO C_TEMPO_DETER	Protagonista (VERB) Oggetto Tempo
		C_TEMPO_DETER (SOGG)	Tempo Protagonista

		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_SVANTAGGIO	Danno
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_SVANTAGGIO	Danno
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		C_SVANTAGGIO	Danno
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MODO	Modo
		C_SVANTAGGIO	Danno
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MODO	Modo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)

		C_OGGETTO	oggetto
		C_MODO	Modo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MODO	Modo
		C_SVANTAGGIO	Danno
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		C_SVANTAGGIO	Danno
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MODO	Modo
		C_SVANTAGGIO	Danno
		C_TEMPO_DETER	Tempo
		(SOGG)	protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		C_SVANTAGGIO	Danno
	Alleare	(SOGG)	Protagonista
		(VERB)	(VERB)
Alleanza		C_LUOGO	Luogo

		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_LUOGO	Luogo
	Violare	(SOGG)	Protagonista
		(VERB)	(VERB)
Violazione		C_OGGETTO	Oggetto
		C_COMPAGNIA	Compagnia
	Scontrare	(SOGG)	Protagonista
	Combattere	(VERB)	(VERB)
Vantaggio		C_SVANTAGGIO	Danno
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_SVANTAGGIO	Danno
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_LUOGO	Luogo
		(SOGG)	(SOGG)
		(VERB)	(VERB)
		C_SVANTAGGIO	Danno
		C_LUOGO	Luogo
		(SOGG)	Protagonista

		(VERB)	(VERB)
		C_MODO	Modo
		C_SVANTAGGIO	Danno
		C_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_LUOGO	Luogo
		C_SVANTAGGIO	Danno
Combattimento	Attacare	(SOGG)	Protagonista
	Assalire	(VERB)	(VERB)
	Assediare	C_OGGETTO	Oggetto
	Scacciare	(SOGG)	Protagonista
	Affrontare	(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MEZZO	Mezzo
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)

		C_OGGETTO	Oggetto
		C_MEZZO	Mezzo
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MEZZO	Mezzo
		C_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MEZZO	Mezzo
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MEZZO	Mezzo
		C_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista

		(VERB)	(VERB)
		C_MEZZO	Mezzo
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MEZZO	Mezzo
		C_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MEZZO	Mezzo
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
Moto	Sbarcare	(SOGG)	Protagonista
	Giungere	(VERB)	(VERB)
	Entrare	C_MOTO_A_LUOGO	Luogo
	Arrivare	(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_MOTO_DA_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		C_MOTO_A_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo

		C_FINE	Effetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_FINE	Effetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		C_MEZZO	Mezzo
		C_FINE	Effetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_MEZZO	Mezzo
		C_FINE	Effetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_MOTO_DA_LUOGO	Luogo
		C_FINE	Fine
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		C_MOTO_A_LUOGO	Luogo
		C_FINE	Effetto
		(SOGG)	Protagonista
		(VERB)	(VERB)

		C_MOTO_A_LUOGO	Luogo
		C_MOTO_DA_LUOGO	Luogo
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		C_MOTO_A_LUOGO	Luogo
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_MOTO_DA_LUOGO	Luogo
		C_FINE	Effetto
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_DA_LUOGO	Luogo
		C_MOTO_A_LUOGO	Luogo
		C_FINE	Effetto
		C_MEZZO	Mezzo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MOTO_A_LUOGO	Luogo
Vittoria	Sconfiggere Battere	(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto

		C_STATO_IN_LUOGO	Luogo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_LUOGO	Luogo
Distruzione	Distruggere	(SOGG)	Protagonista
	Saccheggiare	(VERB)	(VERB)
	Travolgere	C_OGGETTO	Oggetto
	Scacciare	(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MODO	Modo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_COMPAGNIA	Compagnia

		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_MODO	Modo
		C_COMPAGNIA	Compagnia
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_MODO	Modo
		C_OGGETTO	Oggetto
		C_COMPAGNIA	Compagnia
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
		C_TEMPO_DETER	Tempo
		C_TEMPO_DETER	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_OGGETTO	Oggetto
Obiettivo	Giungere Entrare Arrivare	(SOGG)	Protagonista
		(VERB)	(VERB)
		C_FINE	Effetto
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_FINE	Effetto
		C_TEMPO_DETERM	Tempo
		C_TEMPO_DETERM	Tempo
		(SOGG)	Protagonista
		(VERB)	(VERB)
		C_FINE	Effetto

Bibliografia

[1] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. *In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), Madrid*, 1997.

[2] Paul Buitelaar, Daniel Olejnik, and Michael Sintek. Ontolt: A protégé plug-in for ontology extraction from text. In *Second International Semantic Web Conference*, 2003.

[3] David Celjuska and Maria Vergas-Vera. Ontosophie: a semi-automatic system for ontology population from text.

[4] Fillmore C.J. The case for case. *Universal in Linguistics Theory, New York*, pages 1–88, 1968.

[5] Proverbio G. and Tirocini Cerrina A. *Elementi di sintassi strutturale.* Rosenberg and Sellier, 2002.

[6] J. S. Gruber. *Studies in lexical relations.* PhD thesis, MIT, 1965.

[7] E. Hovy and C. Lin. Advances in automated text summarization.

[8] Pustejovsky J. The generative lexicon. *Cambridge Mass, The MIT Press*, 1995.

[9] Propp Vladimir Ja. *Morfologia della fiaba.* Piccola Biblioteca Einaudi, 2000.

[10] Elisabetta Jezek. *Classi di Verbi tra Semantica e Sintassi.* Pubblicazioni della Facoltà di Lettere e Filosofia dell'Università di Pavia, Edizioni ETS, 2003.

[11] Searle J.R. A taxonomy of illocutionary acts. *Gunderson, K. Language, Mind and Knowledge, Minnesota Studies in the Philosophy os Science.*, Vol. VII:pp. 344–369, 1975.

[12] Daniel Jurafsky and James H. Martin. *Speech and language processing.* II edition, Prentice Hall, 2009.

[13] Austin J. L. How to do things with words. *Harvard University Press, Cambridge, MA.*, 1962.

[14] Bawakid A. Oussalah M. A semantic summarization system: University of birmingham at tac. *Proceedings of the First Text Analysis Conference*, 2008.

[15] I. Mani. Summarization evaluation: An overview. *In Proceeding of NAACL 2001, Pittsburgh, Pennsylvania, Usa.*, 2001.

[16] Despontin Marco. Analisi e rielaborazione di fiabe: Sviluppo di una metodologia e di un prototipo. Master's thesis, Politecnico di Milano, 2007.

[17] Chomsky N. Lectures on government and binding. *Dordrecht Foris*, 1981.

[18] Ani Nenkova, Rebecca Passonneau, and Kathleen Mckeown. The pyramid method: incorporating human content selection variation in summarization evaluation. *ACM Transactions on Computational Logic*, Vol. V,(No. N):Pages 1–23, February 2007.

[19] J Larocca Neto, AD Santos, CAA Kaestner, and AA Freitas. Generating text summaries through the relative importance of topics. *In Proc. Int. Joint Conf. IBERAMIA-2000 and SBIA-2000*, volume 1952 of Lecture Notes in Artificial Intelligence:pages 301–309, November 2000.

[20] Steven Pinker. *L'istinto del linguaggio: come la mente crea il linguaggio.* Oscar Saggi Mondadori, 1997.

[21] Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, Damyan Ognyanoff, and Miroslav Goranov. Towards semantic web information extraction. In *2nd International Semantic Web Conference*, 2003.

[22] Marta Sabou. Extracting ontologies from software documentation: a semi-automatic method and its evaluation.

[23] Niels Peter Strandberg. Rule-based expert systems – a practical example. *Artificial Intelligence and Intelligent Systems*, Novembre 2005.

[24] T.A. Van Dijk. Recalling and summarizing complex discourse. *in Text Processing*, pages pp. 49–93, 1979.

[25] Vendler Z. Verbs and times. *Linguistics in philosophy*, 97-121, Ithaca, Cornell University Press.