Università degli Studi di Modena e Reggio Emilia

Facoltà di Ingegneria di Modena

Corso di Laurea in Ingegneria Informatica

Estrazione di concetti ed analisi di documenti testuali: progetto e sviluppo dell'applicazione OKKAM-POP

Relatore:

Prof.ssa Sonia Bergamaschi

Candidato:

Michele Vitali

Correlatore:

Dott. Daniele Cordioli

Anno Accademico 2008/2009

Ai momenti felici e alle persone che sanno renderli unici.

Dedico questo lavoro alle persone che mi sono state accanto nell'ultimo difficile periodo. Ai miei genitori, cui devo un infinito sostegno, a mio fratello, esempio da una vita e agli amici con cui non si condivide solo il tempo ma qualcosa di più.

Parole chiave:

- ✓ Semantic Web
- ✓ Linked Data
- ✓ OKKAM
- ✓ Natural Language Processing
- ✓ Information extraction
- ✓ Knowledge management

SOMMARIO

INDICE DELLE FIGURE

INDICE DELLE TABELLE

INTRODUZIONE

La presente tesi di laurea specialistica descrive gli aspetti più interessanti emersi durante un progetto di ricerca svolto presso l'azienda modenese *Expert System* nel periodo compreso fra ottobre 2008 e Maggio 2009. Expert System opera nel settore del *Knowledge Management* e offre servizi avanzati per l'analisi e la gestione di dati non strutturati.

Il progetto ha reso possibile lo sviluppo di un'applicazione, denominata *OKKAM-POP*, che, sfruttando la tecnologia di analisi linguistica *COGITO*®, riesce ad estrarre in modo automatico e preciso descrizioni e caratterizzazioni delle entità *persone*, *organizzazioni* e *luoghi* presenti all'interno di un *corpus* di notizie giornalistiche. Tali informazioni sono state archiviate all'interno di un *database* corredato di una comoda e intuitiva interfaccia di navigazione che permette analisi specifiche sulle estrazioni, offrendone una visione integrata con ulteriori parametri di valutazione. L'estrazione di concetti relativi alle entità elencate precedentemente si caratterizza come aspetto propedeutico all'interno del progetto europeo *OKKAM* che rappresenta la prima forma di implementazione di un infrastruttura a supporto del *Web Semantico*, l'*ENS (Entity Name System)*. Quello appena delineato è un aspetto trasversale e interdisciplinare che pone un punto di incontro fra le tecnologie di analisi automatica del linguaggio e la realizzazione operativa del Web Semantico.

L'elaborato inizia fornendo una visione di base del Web Semantico con un confronto fra i maggiori progetti a livello mondiale, *Linked Data* e OKKAM che mirano alla sua implementazione.

In seconda battuta vengono introdotti i principali modelli di riferimento utilizzati nell'ambito del Knowledge management. Tali strumenti risultano essere fondamentali, sia a supporto del Web Semantico che delle tecnologie di analisi automatica del linguaggio.

Nel terzo capitolo viene offerta una panoramica degli aspetti principali inerenti alla applicazioni *NLP (Natural Language Processing)*, motivando il suo impiego nel Web Semantico.

Successivamente vengono descritti gli aspetti implementativi del progetto, concludendo con possibili sviluppi futuri delle attività svolte in questo ambito.

1 WEB SEMANTICO

Lo scenario attuale del *World Wide Web*, da un punto di vista estremamente generico, è quello di un enorme insieme di testi e risorse collegati tra loro. Una peculiarità essenziale di questa tecnologia, che ha contribuito a renderla la *killer application* degli ipertesti, è la sua universalità e il fatto che un link ipertestuale in modo molto intuitivo permette potenzialmente a chiunque di mettere in relazione qualsiasi tipo di risorsa pubblicata sulla Rete. I testi sono creati ad uso e consumo dei soli utenti umani, gli unici allo stato attuale in grado di comprendere i contenuti delle pagine che stanno visitando.
Gli utenti umani si orientano nel Web grazie alla loro esperienza di navigazione e alla capacità di evocazione che possono avere parole o espressioni chiave. L'esperienza è un aspetto molto importante di cui tutti ci serviamo: impariamo che determinati contenuti si possono reperire sotto determinati portali, che l'aspetto di un sito può dirci qualche cosa sul genere delle informazioni. Essa si costruisce nel tempo ma non è molto legata ad aspetti tecnici, al codice e alle applicazioni che costituiscono un sito. L'altro aspetto, quello delle parole chiave, è più legato al codice.
Queste caratteristiche non appartengono invece a nessuna applicazione, che in definitiva non è in grado, tranne qualche eccezione limitata e molto complessa, quindi non significativa, di interpretare il contenuto delle pagine.
Il termine Web Semantico è stato proposto per la prima volta nel 2001 da *Tim Berners-Lee*. Da allora il termine è stato associato all'idea di un Web nel quale agiscano agenti intelligenti, applicazioni in grado di comprendere il significato dei testi presenti sulla rete e perciò in grado di guidare l'utente direttamente verso l'informazione ricercata, oppure di sostituirsi a lui nello svolgimento di alcune operazioni. Un agente dovrebbe essere in grado di comprendere il significato dei testi presenti sulla rete, creare percorsi in base alle informazioni richieste dall'utente, guidandolo poi verso di esse e spostarsi di sito in sito collegando logicamente elementi diversi dell'informazione richiesta.
Come caratteristica ulteriore, dovrebbe verificare l'attendibilità di una informazione, tramite un sistema di ricerche incrociate o in dipendenza dal contesto.
Utilizzando questa tecnologia si può automatizzare la ricerca delle pagine, poiché all'atto della creazione del contenuto delle pagine le informazioni sono definite ed inserite secondo precise regole semantiche.

Il Web Semantico è quindi un nuovo modo di concepire i documenti per il World Wide Web e secondo la definizione di Tim Berners-Lee *"Il Web Semantico è un'estensione del Web corrente in cui le informazioni hanno un ben preciso significato e in cui computer e utenti lavorano in cooperazione"* [1].

Le possibilità offerte dal Web Semantico sono tante e tali che non si sono ancora approfondite le sue potenzialità. Per questo, più che di tecnologia, si parla di visione del Web Semantico.

Con l'interpretazione del contenuto dei documenti che il Web Semantico si propone di realizzare, saranno possibili ricerche molto più evolute delle attuali, basate sulla presenza nel documento di parole chiave, ed altre operazioni specializzate come la costruzione di reti di relazioni e connessioni tra documenti secondo logiche più elaborate del semplice link ipertestuale.

1.1 Lo stato dell'arte

Tim Berners-Lee, noto informatico inglese e co-inventore del World Wide Web insieme a insieme a Robert Cailliau, all'interno di una nota intitolata *What the semantic Web isn't but can represent*[2] pubblicata nel 1998, descrive la sua visione del Web Semantico nel seguente modo:

> *"Knowledge representation is a field which is currently seems to have the reputation of being initially interesting, but which did not seem to shake the world to the extent that some of its proponents hoped. It made sense but was of limited use on a small scale, but never made it to the large scale. This is exactly the state which the hypertext field was in before the Web. Each field had made certain centralist assumptions, if not in the philosophy, then in the implementations, which prevented them from spreading globally. But each field was based on fundamentally sound ideas about the representation of knowledge. The Semantic Web is what we will get if we perform the same globalization process to Knowledge Representation that the Web initially did to*

[1] Citazione tratta dall'articolo: Tim B. Lee et al., 2001, *The Semantic Web*, Scientific American; consultabile all'indirizzo *http://www.scientificamerican.com/article.cfm?id=the-semantic-web*

[2] Per la lettura completa della nota visitare il link *http://www.w3.org/DesignIssues/RDFnot.html*

Hypertext. We remove the centralized concepts of absolute truth, total knowledge, and total probability, and see what we can do with limited knowledge."

"La rappresentazione della conoscenza è un settore che è attualmente sembra avere la reputazione di essere inizialmente interessante, ma che non sembra scuotere il mondo nella misura in cui alcuni dei suoi fautori sperano. Ciò ha un senso, dato che è stato limitato l'uso su scala ridotta, senza mai applicarlo su grande scala. Questa è esattamente la situazione in cui si trovava l'ipertesto prima del Web. Ogni campo ha fatto qualche ipotesi centralista, se non nella filosofia, almeno all'interno delle implementazioni, che ha impedito loro la diffusione a livello mondiale. Ma ogni campo è fondamentalmente basato su buone idee circa la rappresentazione della conoscenza. Il Web semantico è ciò che otterremo se si applica alla rappresentazione della conoscenza lo stesso processo di globalizzazione che è stato applicato agli ipertesti attraverso il Web. Rimuoviamo il concetto centralizzato di verità assoluta, di conoscenza totale, e di totale probabilità, e vediamo cosa si può fare con una conoscenza limitata."

Il principio espresso all'interno di questo estratto può essere riassunto nel seguente modo. Prendendo esempio da come il World Wide Web ha mutato il destino degli ipertesti introducendo uno spazio globale, aperto, scalabile e decentralizzato che ha permesso l'integrazione di tutti i documenti web, e in generale di risorse informative, è necessario ideare e realizzare uno strumento che permetta lo stesso salto qualitativo al Web semantico. Ovvero occorre uno spazio globale per l'integrazione di risorse semantiche all'interno di unico modello che permetta di trasformare le attuali reti semantiche locali in un'unica e universale rete di conoscenza semantica, usufruibile a livello mondiale e in grado di garantire gli stessi principi del World Wide Web.
Tale obiettivo rappresenta attualmente una sfida mondiale aperta e affrontata con metodologie differenti da numerosi progetti di ricerca indipendenti e da diverse iniziative commerciali.

Dbpedia, *GeoNames*, *DBLPB*, *MusicBrainz* e *FOAF*[3] sono alcuni esempi di basi di conoscenza realizzate e pubblicate attraverso il formato semantico web *RDF/OWL*[4]. Tuttavia, considerando l'enorme mole di dati e di relazioni pubblicati sul Web, ogni social network, biblioteca digitale, catalogo commerciale, e in teoria ogni base di dati relazionale può essere trasformata in una rete semantica locale esponendo le informazioni nel formato RDF/OWL.

Apparentemente quindi esistono già le componenti principali per la realizzazione del Web semantico, ma se ciò è vero, per quale motivo l'integrazione delle reti semantiche locali non procede come ci si aspetta?

L'integrazione delle reti locali di documenti all'interno del World Wide Web fu enormemente favorito da un fattore chiave, l'introduzione e la realizzazione di un meccanismo unico e globale di indirizzamento che permette la localizzazione e il recupero delle risorse. Questo spazio di indirizzamento è implementato e reso pubblico attraverso un servizio denominato *Domain Name System*[5], il cui compito è quello di mappare ciascun *Uniform Resource Locator*[6] con la posizione fisica che assume all'interno della rete Internet. Questo sistema assicura che, ad esempio, un documento a cui è associata una URL possa essere sempre e inequivocabilmente localizzato e recuperato e che in generale ciascun link ad una risorsa possa sempre essere risolto nella relativa e appropriata locazione fisica, anche nel caso in cui la risorsa venga spostata su un'altra macchina con indirizzo IP differente.

L'integrazione di reti semantiche locali è basta su un'ampia generalizzazione di ciò che può essere indirizzato nel Web, con lo scopo di trasformare il *Web of documents* nel *Web of Resources*. Un identificativo all'interno della rete può essere associato ad oggetti

[3] Quelli elencati sono progetti, attualmente in fase di sviluppo, per la realizzazione di basi di conoscenza esposte pubblicamente e interrogabili attraverso i relativi portali oppure mediante client software. Ognuno di questi progetti ha attinenza con un particolare dominio, in particolare Dbpedia include informazioni e relazioni strutturate derivanti da Wikipedia, GeoNames tratta informazioni geografiche a livello mondiale, BDLPB mantiene informazioni bibliografiche sulla maggior parte delle pubblicazioni nel settore della Computer Science, MusicBrainz raccoglie informazioni sulle pubblicazioni musicali e infine FOAF, acronimo di Friend of a Friend, fornisce gli strumenti per creare e mantenere relazioni fra i social network e le persone in essi descritte.

[4] Sono linguaggi di markup utilizzati per esprimere esplicitamente significato, semantica e relazioni fra termini all'interno di documenti XHTML; verranno descritti con maggiore dettaglio nel paragrafo 1.2 di questo capitolo.

[5] Noto anche che l'acronimo DNS, è il servizio adibito a livello mondiale alla risoluzione dei nomi di host in indirizzi IP per il recupero delle risorse pubblicate sul Web.

[6] Noto anche con l'acronimo URL, è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa in Internet, come un documento o un'immagine.

di tipo informativo, come pagine *HTML*, documenti e immagini, ma non solo, ogni tipologia di oggetto, includendo le entità concrete come persone, località geografiche, eventi e organizzazioni devono poter usufruire di tale sistema di denominazione univoco, affinché si possano raggiungere gli obbiettivi che sono stati prefissati per il Web Semantico.

Ciò conduce inevitabilmente ad un livello più alto di integrazione, se ad esempio due reti semantiche locali indipendenti contengono un riferimento alla stessa entità, allora è fondamentale che le tali riferimenti vengano connessi tra loro e considerati come un unico riferimento sfruttando un *URI*[7] che identifichi universalmente tale entità. Tuttavia il modello appena discusso non rappresenta il caso odierno, in cui viene coniato un nuovo URI ogni volta che la stessa risorsa compare all'interno di una base di conoscenza differente.

A tale proposito esistono essenzialmente due punti di vista che introducono approcci differenti per la soluzione del medesimo problema relativo all'integrazione di reti semantiche contenenti istanze o descrizioni della stessa entità o risorsa. Il punto di vista *a priori* si fonda sull'idea che la molteplicità di URIs per la stessa entità non è di per se un aspetto negativo e che la soluzione appropriata è la creazione di un ulteriore livello di astrazione che permetta la correlazione di tutti gli URIs che fanno riferimento ad essa.

Tale approccio è quello che stato utilizzato per il progetto denominato Linked Data[8], il cui scopo è quello di fornire una metodologia di esposizione, condivisione e connessione dei dati sul Web attraverso URL dereferenziaibli. Il termine dereferenziabile descrive la possibilità di ottenere la rappresentazione della descrizione di un'entità attraverso il suo URI.

Il punto di vista *a posteriori*, invece, è basato sull'idea che la proliferazione di molteplici URIs per la stessa entità debba essere limitato fin dall'inizio e che un'adeguata soluzione debba supportare nel modo più diffuso possibile l'utilizzo di URIs univoci e globali.

---

[7] Unified Resource Identificator, è una stringa che identifica univocamente una risorsa generica che può essere un indirizzo Web, un documento, un'immagine, un file, un servizio, un indirizzo di posta elettronica, ecc. Si differenzia dall'URL per il fatto che un URI rappresenta un concetto più generalizzato, in particolare un URL è uno specifico tipo di URI.

[8] Per una descrizione dettagliata del progetto Linked Data consultare il paragrafo 1.3 di questo capitolo.

Questa seconda prospettiva è quella perseguita progetto europeo OKKAM[9], il cui principale intento è la realizzazione di una nuova infrastruttura a livello mondiale a supporto del Web Semantico.

1.2 Strumenti a disposizione del Web Semantico

All'interno di questa introduzione al concetto di Web Semantico non si andrà nel dettaglio delle tecnologie e delle metodologie impiegate per renderlo operativo, tuttavia di seguito viene fornita una breve descrizione degli strumenti con cui attualmente è possibile integrare informazioni semantiche all'interno dei contenuti Web.

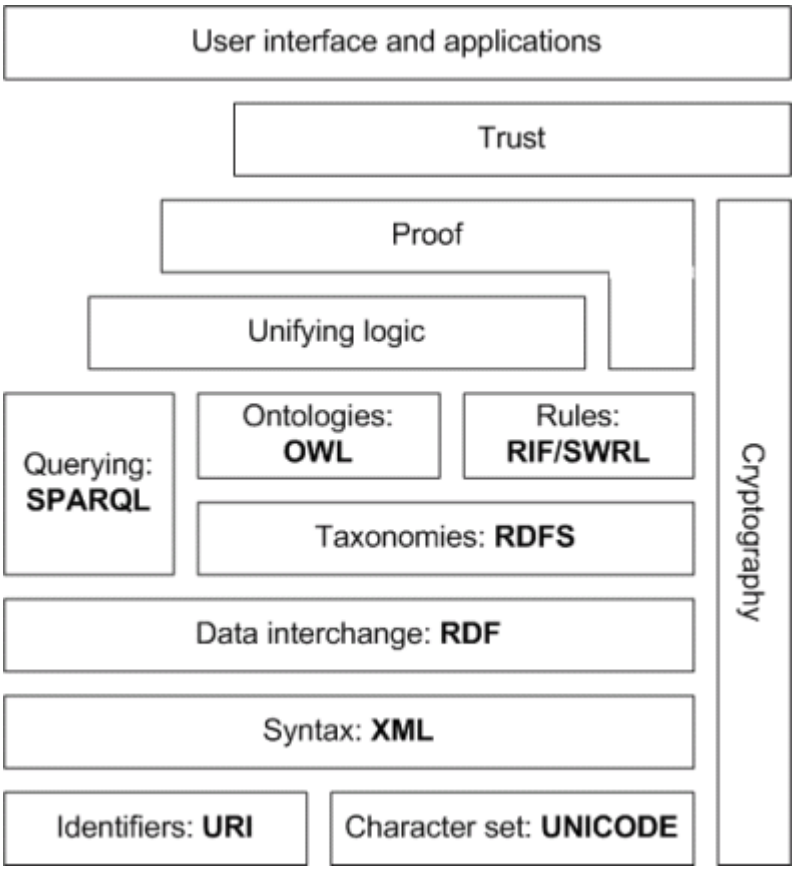

Figura 1.1 - Stack del Web Semantico.

[9] Per una descrizione dettagliata del progetto OKKAM consultare il paragrafo 1.4 di questo capitolo.

In Figura 1.2.1 sono rappresentati i livelli logici che insieme costituiscono il concetto più ampio di Web Semantico.

La realizzazione del Web Semantico si scontra con il fatto tale modello deve essere costruito su una infrastruttura già esistente, il Web. Risulta quindi più semplice perseguire i risultati attraverso la scomposizione di questo complesso problema in passi multipli.

Il Web Semantico lo si sta realizzando a strati, ereditando i risultati precedentemente ottenuti e adottando successivamente le soluzioni che meglio si adattano all'evoluzione delle nuove tecnologie.

Dando per scontato che il lettore conosca già i concetti basilari degli identificatori *URI* e del sistema di codifica *UNICODE*[10], con cui vengono pubblicati i documenti Web, il punto di partenza da cui hanno origine le attuali esperienze d'implementazione del Web Semantico è rappresentato dal linguaggio *XML*[11]. L'XML è un metalinguaggio di markup che consente di descrivere semanticamente le diverse parti di un documento. L'utilizzo del linguaggio XML per la realizzazione di contenuti Web offre la possibilità di interpretazione delle informazioni, oltre che da parte dell'utente umano, anche da parte di applicazioni software e fornisce inoltre un importante facilitazione nel caso in cui le informazioni di un documento debbano essere ristrutturate parzialmente o totalmente per l'adattamento a formati differenti.

Insieme al linguaggio XML è necessario introdurre i concetti di *XML Schema* e *XML namespace*. XML Schema è il linguaggio di descrizione del contenuto di un file XML. Come tutti i linguaggi di descrizione del contenuto, il suo scopo è delineare quali elementi sono permessi, quali tipi di dati sono ad essi associati e quali relazioni gerarchiche sussistono fra gli elementi contenuti in un file XML. Ciò permette principalmente la validazione del file XML, ovvero la verifica che i suoi elementi siano in accordo con la descrizione in linguaggio XML Schema.

Gli XML namespace vengono utilizzati per fornire nomi univoci di elementi o attributi all'interno di un'istanza XML e servono essenzialmente per risolvere le ambiguità che possono presentarsi in caso di omonimia. Si consideri ad esempio un file XML

[10] Unicode è un sistema di codifica che assegna un numero univoco ad ogni carattere usato per la scrittura di testi, in maniera indipendente dalla lingua, dalla piattaforma informatica e dal programma utilizzati.

[11] L'eXtensible Markup Language è un metalinguaggio di markup, ovvero un linguaggio marcatore che definisce un meccanismo sintattico che consente di estendere o controllare il significato di altri linguaggi marcatori.

all'interno del quale vengano specificati un cliente e un prodotto acquistato. Sia il prodotto che il cliente potrebbero avere un sottoelemento *ID_NUMBER* e qualsiasi riferimento a questo elemento risulterebbe ambiguo senza la sua risoluzione da parte di un namespace che ne specifichi il ruolo semantico.
Tale problema si evidenzia maggiormente nei casi in cui si effettua l'integrazione di due o più istanze XML che derivano da applicazioni differenti. Di seguito viene mostrata la soluzione a livello di codice che fa uso dei *namespaces* per risolvere le ambiguità.

```
1.  <?xml version="1.0"?>
2.  <TRANSACTION   xmlns:a="http://www.company/customer"
3.                      xmlns:b="http://www.company/product">
4.      <CUSTOMER>
5.          <a:ID_NUMBER>674686</a:ID_NUMBER>
6.          <FNAME>George</FNAME>
7.          <SNAME>Lucas</SNAME>
8.      </CUSTOMER>
9.      <PRODUCT>
10.         <b:ID_NUMBER>140964</b:ID_NUMBER>
11.         <NAME> Table</NAME>
12.         <QTY>3</QTY>
13.         <PRICE>112</PRICE>
14.     </PRODUCT>
15. </TRANSACTION>
```

Come si può notare alla riga 1, nell'elemento *TRANSACTION*, vengono definiti due namespaces attraverso l'attributo *xmlns* a cui viene attribuito un prefisso con la sintassi *xmlns:prefix*. Tali prefissi vengono utilizzati alla riga 5 e alla riga 10 per specificare con quale namespace devono essere disambiguati gli elementi *ID_NUMBER*.
Nonostante un documento XML sia un ottimo mezzo attraverso il quale trasferire informazioni, la natura distribuita e decentralizzata del Web fornisce un ostacolo ulteriore a quell'integrazione totale e globale delle informazioni a cui tende il Web Semantico.
Dal punto di vista pratico l'XML fornisce uno strumento semplice e intuitivo per descrivere ad esempio un entità all'interno di un documento, tuttavia questo linguaggio non definisce alcun meccanismo esplicito per qualificare le relazioni tra documenti e

non permette di definire quindi se la stessa entità è descritta anche in altri documenti presenti nel Web.

La reale evoluzione del Web nel Web Semantico inizia con la definizione, da parte del *W3C*[12], dello standard *Resource Description Framework*[13], una particolare applicazione XML che standardizza la definizione di relazioni tra informazioni ispirandosi ai principi della logica dei predicati e ricorrendo agli strumenti tipici del Web (URI, URL e URN) e dell'XML (namespace). In estrema sintesi, secondo la logica dei predicati le informazioni sono esprimibili con asserzioni costituite da triple formate da soggetto, predicato e valore.

RDF offre quindi quel meccanismo di cui era privo l'XML, e che risulta essere fondamentale per esprimere in modo condiviso e standardizzato relazioni fra entità e valori, ad esempio la relazione espressa dal predicato

"Barack Obama è il presidente degli stati Uniti" può essere formalizzata attraverso il seguente modo in RDF:

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF
3.      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.      xmlns:wikipedia="http://it.wikipedia.org/wiki/"

5.      <rdf:Description  rdf:about="http://www.barackobama.com">
6.          <wikipedia:presidente>
7.              Stati Uniti d'America
8.          </wikipedia:presidente>
9.      </rdf:Description>

10. </rdf:RDF>
```

Questo formalismo può essere utilizzato all'interno dei documenti per esprimere qualsiasi tipo di relazione fra qualsiasi tipo di entità o attributo e si basa essenzialmente su tre principi chiave:

- ✓ Qualunque cosa può essere identificata da un URI*;*

[12] World Wide Web Consortium, è un'associazione fondata nel 1994 da Tim Berners Lee in collaborazione con il CERN. Questa associazione, nata con lo scopo di migliorare gli esistenti protocolli e linguaggi per il WWW e di aiutare il Web a sviluppare tutte le sue potenzialità, rappresenta oggi il punto di riferimento a livello mondiale per quanto riguarda gli standard e i protocolli di comunicazione.

[13] Noto con l'acronimo RDF.

- ✓ *The least power*: utilizzare il linguaggio meno espressivo per definire qualunque cosa;
- ✓ Ogni risorsa può essere messa in relazione con qualsiasi altra risorsa.

Un modello RDF è inoltre rappresentabile da un grafo orientato sui cui nodi si trovano le risorse o i tipi primitivi mentre gli archi rappresentano le proprietà o le relazioni.
Al di sopra del livello RDF troviamo il livello *RDFS*, ovvero il *Resource Description Framework Schema*. L'RDFS è un linguaggio estensibile per la rappresentazione della conoscenza e rappresenta essenzialmente una collezione di informazioni riguardo alle classi di nodi, proprietà e relazioni esprimibili dal linguaggio RDF.
La logica predicativa del primo ordine, su cui si basa RDF, conduce a una complessità estremamente elevata quando si tenta di applicare processi di inferenza alle relazioni espresse nei documenti, perciò è stato realizzato un nuovo standard, *OWL*[14], che rappresenta una logica costituita da un sottoinsieme degli operatori della logica del primo ordine. OWL risulta essere più ricco ed espressivo di RDF e offre molti nuovi costrutti, due di questi, molto semplici da comprendere, sono l'equivalenza tra risorse e la relazione inversa.
Per equivalenza tra risorse si intende la possibilità di poter affermare che due o più URI rappresentano lo stesso elemento, mentre per relazione inversa si intende la possibilità di dire che se è vera la proposizione "soggetto, predicato, oggetto", allora è anche vera "oggetto, predicato inverso, soggetto".
OWL esiste in tre forme, caratterizzate da diversi gradi di complessità e di computabilità. *OWL-Light* è computabile, cioè è possibile trovare tutte le soluzioni in un tempo finito a discapito di una bassa espressività. *OWL-DL* è ugualmente computabile come OWL-Light ma risulta essere più ricco dal punto di vista espressivo. Infine esiste *OWL-Full*, che copre tutta la ricchezza della logica predicativa, ma non è computabile e non è quindi adatto al ragionamento automatico. Con OWL è possibile realizzare ontologie che descrivono la conoscenza che abbiamo di un certo dominio, tramite classi, relazioni fra classi e individui appartenenti alle classi. La conoscenza così formalizzata è processabile automaticamente da un calcolatore, tramite un algoritmo che implementi i processi inferenziali e deduttivi.

[14] Web Ontology Language, standard del World Wide Web Consortium.

Allo stesso livello di OWL, in Figura 1.1, troviamo i linguaggi *RIF/SWRL*[15] che, anche se presenti nell'elenco delle tecnologie che costituiscono il Web Semantico, non sono in realtà ancora stati standardizzati e implementati, ma ne sono solo stati delineati i concetti e gli scopi.

RIF è una componente proposta dal *World Wide Web Consortium* che si prefigge di realizzare un modello di interscambio per regole scritte in differenti linguaggi e per motori di inferenza eterogenei. Come detto in precedenza un'ontologia descrive un insieme di concetti con un formato usufruibile dalle applicazioni, mentre RIF descrive i metodi per inferire nuove informazioni da un'ontologia, combinarle in modo utile per permetterne l'utilizzo da parte di altre sorgenti informative.

SWRL[16] è un linguaggio proposto per l'implementazione di regole, usufruibili dal Web Semantico, che combina parti dei linguaggi OWL e del *Rule Markup Language*[17].

SPARQL, acronimo ricorsivo che significa SPARQL Protocol and RDF Query Language, è stato standardizzato dall' *RDF Data Access Working Group*, facente parte del W3C ed è considerato uno dei punti chiave a livello tecnologico per il Web Semantico. Questo linguaggio permette di effettuare richieste formalizzate in grado di ritornare una tripletta RDF, insiemi congiunti o insiemi disgiunti di risultati.

Di seguito viene mostrato un semplice esempio della sintassi impiegata per effettuare la richiesta di tutte le capitali di tutti gli stati dell'Africa tramite SPRQL:

```
1. PREFIX abc: <http://example.com/exampleOntology#>
2. SELECT ?capital ?country
3. WHERE {
4. ?x abc:cityname ?capital ;
5. abc:isCapitalOf ?y .
6. ?y abc:countryname ?country ;
7. abc:isInContinent abc:Africa .
8. }
```

Le variabili sono precedute dal simbolo "?" e come si può notare la sintassi fa uso della definizione di prefissi per rendere la richiesta maggiormente concisa.

[15] RIF è l'acronimo di Rule Interchange Format, mentre SWRL è l'acronimo di Semantic Web Rule Language.

[16] http://www.w3.org/Submission/SWRL

[17] Il Rule Markup Language (RuleML) è un linguaggio di markup sviluppato per esprimere sia regole induttive che deduttive in, http://ruleml.org.

Lo SPARQL *query processor* cercherà tutti gli insiemi di triplette del tipo "città *?capital* è capitale dello stato *?countryname*", con la condizione definita dall'utente che lo stato deve far parte del continente Africa. Un linguaggio per effettuare richieste di questo tipo risulta essere fondamentale per le applicazioni che verranno sviluppate sul Web Semantico.

Tutti i livelli che sovrastano quelli appena descritti non sono ancora stati sviluppati, non è quindi possibile parlare di sistemi specifici ma di idee che probabilmente verranno realizzate attraverso molti sistemi differenti.

Mentre risulta abbastanza semplice avere sistemi che comprendono concetti base come relazioni di gerarchia o relazioni inverse potrebbe risultare ancora più utile se potessimo definire numerosi principi logici e permettere ai computer di effettuare, grazie ad essi, ragionamenti basati sull'inferenza. Queste funzionalità dovranno essere implementate all'interno del livello *Unifying Logic.*

Ci si ponga ad esempio in un ambito aziendale che stabilisce la seguente regola: "se un dipendente effettua più di 100 vendite diventa membro del club super venditore". Un programma intelligente potrà seguire questa regola per fare semplici deduzioni: "Luca ha venduto 102 prodotti, quindi Luca è un membro del club super venditore".

Una volta realizzato un sistema che segue la logica potrebbe essere molto sensato utilizzarlo per provare affermazioni o in generale qualsiasi tipologia di dato verificabile. Il livello *Proof* dovrebbe poter fornire questa capacità, tuttavia nasce un dilemma non indifferente basato sulla fiducia che deve meritare un'affermazione provata con queste metodologie. È a questo punto che si inserisce il livello *Trust* dal momento che ogni prova dovrebbe supportata dalla fiducia.

L'idea di base è che ogni documento e ogni relazione RDF all'interno del Web Semantico debba essere firmata digitalmente dal proprio autore, in questo modo ognuno potrà definire il proprio livello di fiducia dichiarando a quali firme credere e a quali no.

Sono appena state descritte le basi di quello che viene definito il *Web of Trust.* La fiducia inoltre può essere soggetta alla proprietà transitiva, ovvero se io affermo di avere fiducia verso una certa persona che a sua volta ha fiducia verso un altro gruppo di persone allora probabilmente sarò portato ad allargare la mia fiducia anche a queste ultime. Oltre al concetto di fiducia può essere molto utile il concetto di sfiducia, se supponiamo infatti di trovare un documento a cui nessuno ha dato un'esplicita sfiducia,

ma nemmeno un'esplicita fiducia, è probabile che tale risorsa meriti più considerazione di un'altra che è stata esplicitamente sfiduciata.

Il livello trasversale *Cryptography*, che attraversa quasi l'intero *stack* proposto in Figura 1.1, ricorda che a supporto della fiducia delle informazioni contenute nel Web Semantico è necessario che vi sia un robusto meccanismo di protezione per l'inviolabilità dei dati e delle identità, altrimenti l'intero sistema risulterebbe inutile a causa della sua inattendibilità.

L'ultimo livello, *User interfaces and applications*, non è propriamente una parte del Web Semantico, è piuttosto una rappresentazione di tutte le future applicazioni che ne faranno uso, agendo ed interagendo con esso in modo del tutto trasparente agli occhi degli utenti.

Il Web, come si presenta oggi, richiede strumenti di lavoro più progrediti, per facilitare e velocizzare la navigazione attraverso gli innumerevoli documenti pubblicati. Per il futuro, il Web Semantico si propone di dare un senso alle pagine Web ed ai collegamenti ipertestuali, dando la possibilità di cercare solo ciò che è realmente richiesto. Non sempre la Rete ci porta dove ci attenderemmo e le difficoltà d'orientamento sono significative quando siamo alla ricerca di qualche cosa e non sappiamo dove reperirlo. Scorrere una lunga quantità di elenchi alla ricerca dell'informazione desiderata è ormai quotidianità, soprattutto quando la ricerca interessa un termine piuttosto comune. Con il Semantic Web possiamo aggiungere alle nostre pagine un senso compiuto, un significato che va oltre le parole scritte, una "personalità" che può aiutare ogni motore di ricerca ad individuare ciò che stiamo cercando semplicemente perché lo è, scartando, di fatto, gli altri che non soddisfano la nostra richiesta. Tutto questo non in virtù di sistemi di intelligenza artificiale, ma semplicemente in virtù di una marcatura dei documenti, di un linguaggio gestibile da tutte le applicazioni e dell'introduzione di vocabolari specifici, ossia insiemi di frasi alle quali possano associarsi relazioni stabilite fra elementi marcati. Il Web Semantico per funzionare deve poter disporre di informazione strutturata e di regole di deduzione per gestirla, in modo da accostare quelle informazioni che un'interrogazione ha richiesto. Tim Berners-Lee ha sottolineato che uno degli elementi fondamentali del Web Semantico sarà la compresenza di più ontologie integrate a livello logico, aspetto trattato nel prossimo paragrafo.

1.3 Linked Data

Il progetto Linked Data, come annunciato nel secondo paragrafo di questo capitolo, tenta di affrontare il problema della realizzazione del Web Semantico accettando come presupposto il fatto che esistano già numerose basi di conoscenza che contengono un'enorme mole di informazioni e relazioni fra le risorse pubblicate sul Web e che il passo necessario per la nascita del *Web 3.0*[18] sia la realizzazione di un livello intermedio che si ponga fra queste strutture e gli utenti che le utilizzano.

Più precisamente quindi Linked Data rappresenta un sottoargomento del Web Semantico, anche se questo termine viene utilizzato per descrivere il metodo di esposizione, condivisione e connessione dei sul Web attraverso l'utilizzo di URIs dereferenziabili. Questo progetto mira a fornire degli standard di utilizzo di strumenti come RDF e *HTTP*[19] per la pubblicazione di dati strutturati sul Web e per la loro connessione attraverso sorgenti dati eterogenee.

I principi base[20] del progetto Linked Data furono delineati per la prima volta da Tim Berners-Lee nel 2006 e provvedono a fornire una linea guida sulle metodologie e sulle regole a cui gli sviluppatori dovrebbero attenersi per raggiungere lo scopo finale, ovvero la nascita del Web Semantico.

I principi a cui si è fatto riferimento sono essenzialmente quattro e vengono presentati di seguito nello stesso semplice, ma efficace, modo con cui sono stati esposti da Tim Berners-Lee:

1. La prima regola è l'utilizzo degli URIs per identificare risorse o qualsiasi tipo di entità sul Web.
2. Le seconda regola è l'utilizzo di HTTP URIs. Anche se può sembrare ridondante è necessario specificare questo principio dal momento che fin dalla nascita del Web c'è stata una tendenza alla realizzazione di nuovi schemi URIs, come ad esempio *LSIDs*, *XRIs* e *DOIs*. Il problema nasce dal fatto che il DNS è composto da un complesso insieme di standard non rispettati completamente dai nuovi schemi, che risultano quindi non efficaci per identificare risorse generiche sul Web.

[18] Il Web 3.0 è un termine a cui corrispondono significati diversi volti a descrivere l'evoluzione dell'utilizzo del Web e l'interazione fra gli innumerevoli percorsi evolutivi possibili, uno dei quali è il Web Semantico.
[19] L'Hypertext Transfer Protocol è usato come principale sistema per la trasmissione di dati sul web.
[20] Articolo introduttivo a *Linked Data* reperibile a *http://www.w3.org/DesignIssues/LinkedData.html*

3. La terza regola invita ad aggiungere sempre maggiori informazioni agli URIs che vengono visitati. Tale principio viene seguito ad esempio dalle ontologie dal momento che permettono di valutare la classe e gli attributi di un'entità e permettono di navigare le relazioni, tuttavia esiste un'enorme quantità di basi di conoscenza, realizzate nel corso di progetti inerenti al Web Semantico, che non vengono esposte pubblicamente .
4. La quarta regola invita ad utilizzare i link URI di altre entità all'interno dei propri contenuti in modo da diffondere sempre più le relazioni esistenti nell'intero Web.

Come si può notare i principi base sono relativamente semplice tuttavia il non rispettarli, secondo Berners-Lee, implica una difficoltà sempre maggiore nella realizzazione del Web Semantico.

In linea di principio sarà possibile accedere al Web dei dati utilizzando un *browser Linked Data*, così come nel Web dei documenti si può accedere alle risorse tramite un *browser HTML*. La differenza risiede nel fatto che, gli utenti, invece di seguire i link attraverso le pagine HTML potranno navigare attraverso differenti sorgenti dati seguendo i link RDF.

Così come il Web tradizionale composto da documenti può essere scansionato seguendo i link ipertestuali, sarà possibile scansionare il Web dei dati seguendo i link RDF. Lavorando su tali scansioni i motori di ricerca potranno fornire capacità sofisticate di ricerca, simili a quelle fornite dai convenzionali database relazionali, e dal momento che i risultati delle *queries* saranno a loro volta dati strutturati, e non semplici link a pagine HTML, sarà possibile creare un insieme di nuovi applicazioni basate sul Web dei dati.

Parallelamente allo sviluppo di Linked Data, nel 2007, è nato il progetto *Linking Open Data*. Gli obbiettivi di tale lavoro sono l'identificazione di strutture e basi di conoscenza che siano disponibili sotto una licenza aperta, la loro ripubblicazione con linguaggio RDF e l'interconnessione fra tali strutture in modo da poter definire relazioni navigabili. Durante il 2007 la crescita delle relazioni RDF prodotte da questo progetto ha raggiunto quota due miliardi, utilizzando sorgenti dati appartenenti a diversi domini, come quello geografico, dell'informazione, delle persone, delle comunità online, del linguaggio umano, delle pubblicazioni scientifiche, dei films, della musica, dei libri, ecc…

Tutte queste sorgenti dati sono interconnesse da circa tre milioni di link RDF e in Figura 1.2 viene mostrata la complessa struttura che rappresenta una visione generale del progetto Linking Open Data.

La *nuvola* di sorgenti dati che partecipano al progetto rende immediatamente l'idea delle sue dimensioni e della portata di informazioni e relazioni RDF coinvolte. Come si può notare nel diagramma si possono distinguere *hub* principali come DBpedia e Geonames. DBpedia provvede ad estrarre triplette *RDF* dalle *Infoboxes* che normalmente vengono visualizzate alla destra degli articoli di *Wikipedia* e li rende disponibili sul Web nel formato RDF. Geonames, invece, provvede a fornire la descrizione RDF di milioni di località geografiche distribuite in tutto il mondo.

Gli hub assumono una maggiore importanza per il fatto che contengono la maggior parte delle entità a cui gli utenti intendono riferirsi nei loro contenuti e devono supportate quindi un numero maggiore di interconnessioni.

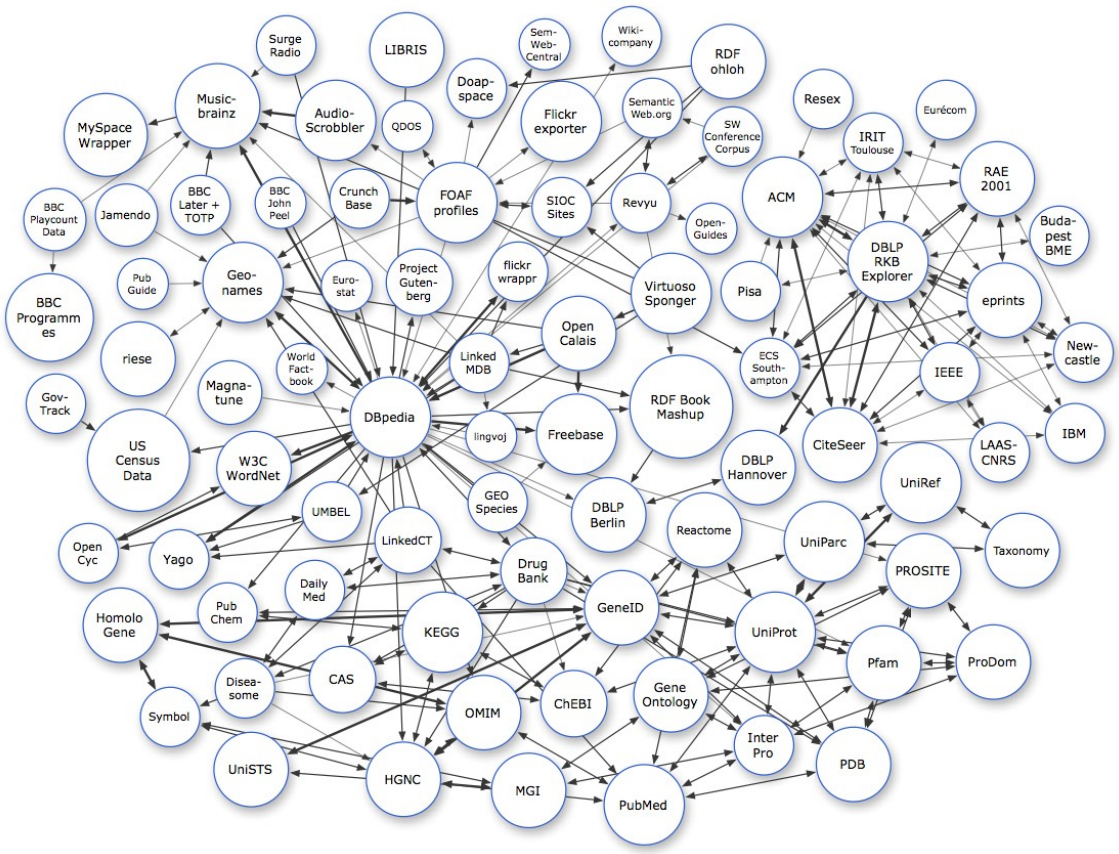

Figura 1.2 – Rappresentazione globale del progetto *Linking Open Data*

Oltre produrre un'integrazione di tutte le informazioni e relazioni presenti nelle basi di conoscenza coinvolte nel progetto, Linked Data si propone di sviluppare anche browsers[21], crawlers e motori di ricerca[22] che siano in grado di sfruttare al meglio le opportunità offerte dal Web dei dati. Esempi di browsers Linked Data sono *Tabulator*, *Disco*, il browser *OpenLink* e *Zitgist*. Esempi di motori di ricerca che sfruttano la tecnologia RDF sono *Falcons*, *Sindice*, *Swoogle* e *Watson*.
Questi strumenti permettono agli uomini e alle macchine di localizzare ed effettuare queries sull'infrastruttura creata da Linked Data.

1.4 IL progetto OKKAM

La presente tesi, sviluppata in seguito a un'esperienza di tirocinio semestrale presso l'azienda Expert System, si inserisce all'interno di un progetto proposto dalla Commissione Europea denominato OKKAM che vede coinvolte diverse realtà universitarie e aziendali e facente parte delle iniziative proposte dal *FP7* (*Seven Framwork Program*).
Di seguito, nella Tabella 1.1, vengono presentate in modo schematico le caratteristiche principali del progetto di ricerca OKKAM:

Titolo del progetto:	Enabling the Web of Entities. A scalable and sustainable solution for systematic and global identifier reuse in decentralized information environments
Acronimo del progetto:	OKKAM
Codice del progetto:	215032
Data di inizio:	2008-01-01
Data di fine:	2010-06-30
Durata:	30 mesi
Numero di partecipanti:	12 (5 Università, 4 PMI, 3 grandi compagnie)
Capitale umano:	763 mesi uomo (più di 63 anni uomo)
Finanziamento europeo:	5.125.000,00 Euro
Costo totale:	7.359.931,34 Euro

[21] http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients
[22] http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/SemanticWebSearchEngines

Partecipanti:	1. University of Trento, IT 2. L3S Hannover, DE 3. SAP Research, DE 4. Elsevier, NL 5. Expert System, IT 6. Europe Unlimited, BE 7. MAC, IE 8. EPFL, CH 9. DERI Galway, IE 10. University of Malaga, SP 11. INMARK, SP 12. ANSA, IT

Tabella 1.1 – Caratteristiche del progetto OKKAM

L'acronimo del progetto di ricerca deriva da un principio metodologico espresso nel XIV secolo dal filosofo Guglielmo da Ockham: *"Entia non sunt multiplicanda praeter necessitatem"* (trad. *"Non moltiplicare gli elementi più del necessario"*). Tale principio, conosciuto come rasoio di Ockham, nella sua forma più semplice suggerisce l'inutilità di formulare più assunzioni di quelle strettamente necessarie per spiegare un dato fenomeno. Il progetto OKKAM mira a rendere possibile un Web basato sulle entità, ovvero uno spazio virtuale in cui gli insiemi di dati e informazioni su ogni tipologia di entità (persone, località, eventi, organizzazioni, prodotti, ecc...) pubblicata sul Web possa essere integrata all'interno di un base di conoscenza univoca, aperta e decentralizzata. OKKAM contribuirà a realizzare tale scopo sostenendo la convergenza verso l'uso di un unico identificatore univoco in tutto il mondo per ciascuna entità che è pubblicata sul Web. L'intuizione del progetto è la concreta realizzazione di strumenti che permettano di mettere in atto il principio di Ockham, ovvero strumenti che permettano di tagliare alla radice il problema della proliferazione di nuovi ed inutili identificatori. A tale scopo il progetto metterà a disposizione degli editori di contenuti Web e degli sviluppatori un'infrastruttura globale e una serie di nuovi tools e plugins che offriranno un supporto intuitivo per trovare facilmente gli identificatori pubblici per le entità citate nei propri contenuti, per la creazione di annotazioni, o per la realizzazione di nuovi servizi basati sul Web che facciano uso di questi identificatori in un ambiente aperto come Internet.

Per raggiungere gli obbiettivi che il progetto OKKAM si è proposto sono stati stilati tre passi fondamentali da affrontare sequenzialmente:

- ✓ Fornire una infrastruttura scalabile e sostenibile, chiamata Entity Name System (ENS), per permettere che il riuso sistematico di identificatori globali e univoci delle entità non solo sia possibile, ma anche semplice e intuitivo. L'ENS sarà un servizio distribuito, paragonabile al DNS, che memorizzerà permanentemente gli identificatori delle entità e provvederà a fornire una serie di funzionalità come ad esempio l'*entity matching*, il *mapping* e la risoluzione di identificatori;
- ✓ permettere la crescita automatica e rapida del *Web of Entities* favorendo la creazione di contenuti *OKKAMizzati* [23] all'interno di applicazioni *OKKAM-empowered*[24];
- ✓ mostrare i vantaggi che il Web of Entities offre e, più in generale, mostrare gli aspetti positivi di un approccio orientato alle entità per la gestione dei contenuti e della conoscenza, realizzando nuove applicazioni ai vertici dell'infrastruttura in tre principali aree: ricerca semantica e *information retrieval*, l'*authoring* dei contenuti (in particolare nel settore delle pubblicazioni scientifiche e delle news) e la gestione della conoscenza organizzativa delle aziende per permettere un processo produttivo più controllato, rapido e integrato.

L'impatto dell'infrastruttura proposta può essere difficilmente sovrastimato in caso di successo del progetto. Non solo si fornirà un servizio pubblico per l'integrazione dei dati e dei servizi all'interno del Web of Entities ma si costituiranno anche le fondamenta per un intera generazione di nuove applicazioni e servizi che trarranno benefici dall'utilizzo di identificatori globali e contenuti OKKAMizzati.

Di seguito vengono mostrate alcune immagini dell'architettura logica e strutturale delle macrocomponenti che costituiscono il progetto OKKAM.

La prima componente presa in esame è il servizio *OkkamPUBBLIC*, visualizzato nella parte inferiore della Figura 1.3. Il servizio viene realizzato tramite una struttura cluster di nodi sincronizzati e indipendenti che rispondo a richieste provenienti non solo di utenti ma anche da applicativi software.

[23] Contenuti in cui le entità sono nominate o annotate con gli identificatori OKKAM
[24] Applicazioni che possono interagire con l'ENS per il recupero e il riutilizzo degli identificatori

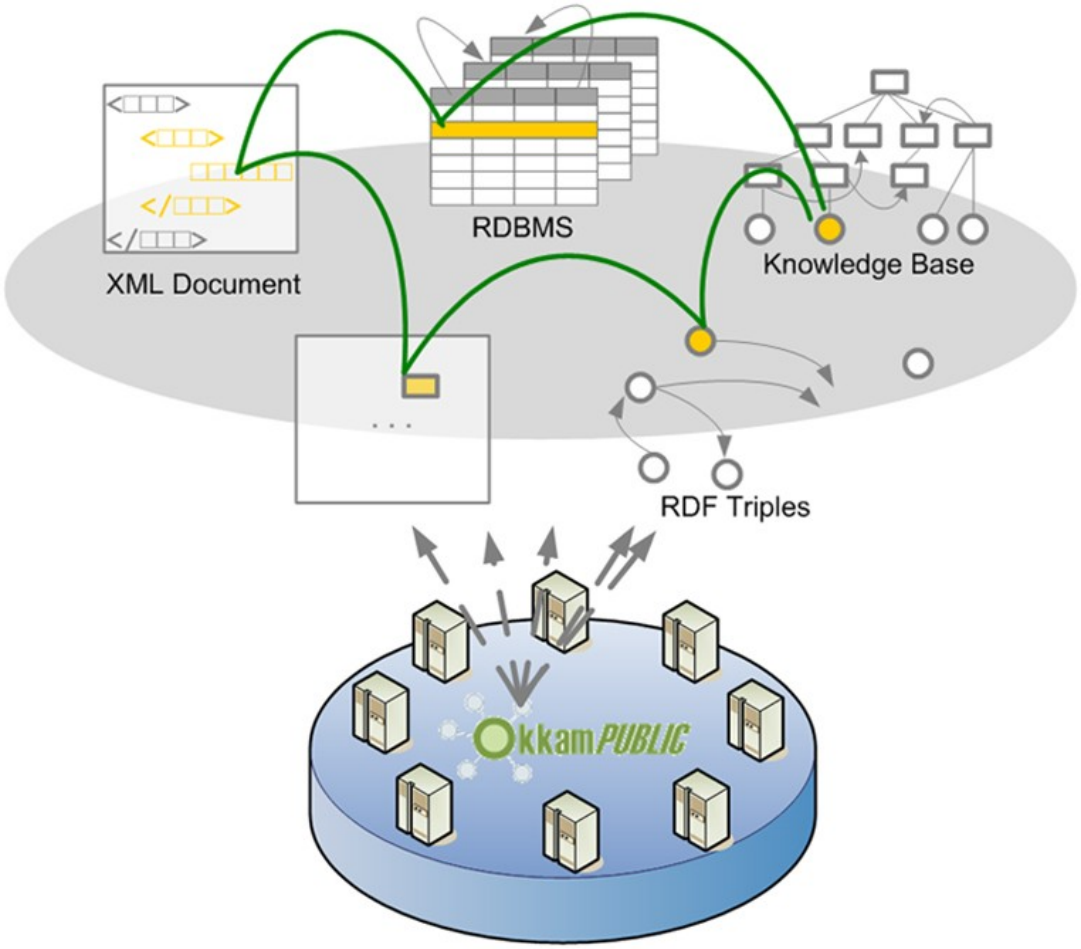

Figura 1.3 – Interazioni del servizio *OkkamPUBLIC*

Il servizio base, ovvero l'ENS con modalità simili al DNS, sarà accessibile da tutti gli utenti per richieste di risoluzione degli identificatori delle entità e delle relazioni che sussistono fra esse. Come mostrato in Figura 1.3 all'interno di un documento XML sarà possibile definire il ruolo semantico delle entità in esso contenute, anche nel caso in cui tali entità provengano ad esempio da una struttura *RDBMS*[25], da un'ontologia, oppure non provengano da nessuna fonte strutturata. Questa procedura di identificazione, che ha richiesto e richiederà sforzi enormi nella prima fase del progetto, permetterà di usufruire di tutte le relazioni semantiche contenute all'interno del repository OKKAM ed esposte tramite il formato RDF. Principio su cui si fonda l'intera architettura è naturalmente la scalabilità, per questo motivo i nodi che implementano il servizio *OkkamPUBBLIC* sono indipendenti l'uno dall'altro, per evitare la presenza di colli di bottiglia che potrebbero limitare una possibile espansione futura dell'ENS.

[25] Relational Database Management System

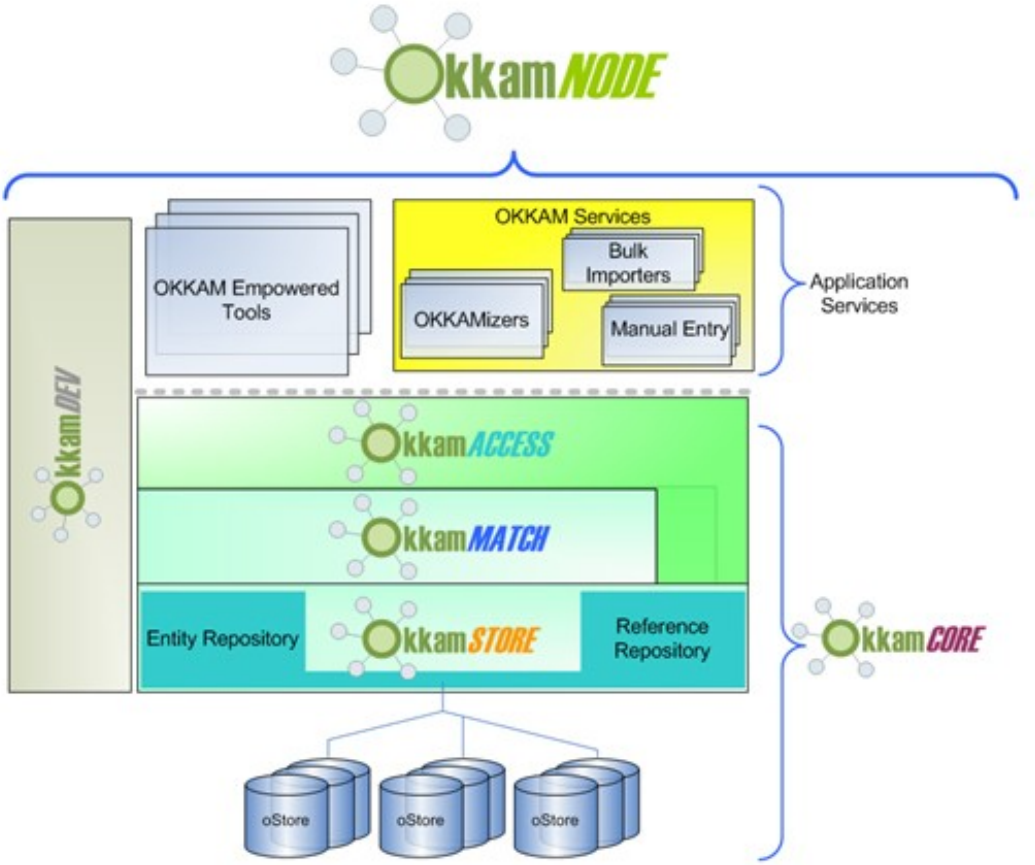

Figura 1.4 – Componenti dell'*OkkamNode*

Nella Figura 1.4 sono evidenziate le principiali componenti a livello applicativo di un *OkkamNODE*, ovvero l'*OkkamCore* e i servizi applicativi. L'OkkamCore, come è intuibile dal nome, rappresenta il nucleo di un nodo dell'architettura ed è a sua volta costituito dalle sottocomponenti *OkkamACCESS*, *OkkamMATCH* e *OkkamSTORE.*

OkkamACCESS fornisce un meccanismo di controllo e sicurezza tale per cui ogni richiesta rivolta a un OkkamNODE debba superare un primo livello di accesso per poter essere soddisfatta. E' stato previsto anche un secondo livello di sicurezza, che risiede fra il livello di *matching* e di *storing*, per garantire l'indipendenza di gestione dal primo livello di accesso e per garantire una maggior sicurezza dei livelli inferiori. Al di sotto del livello di accesso si trova l' OkkamMATCH che provvede a fornire funzionalità di alto livello per l'esecuzione di query e il confronto di entità. Uno dei principali obbiettivi di un OkkamNODE è quello di effettuare un confronto fra i dati derivanti da un'applicazione con tutte le entità presenti all'interno del repository e fornire una lista di candidati che possano soddisfare i criteri imposti nella richiesta. I rispettivi metodi che fornisco tali funzionalità risiedono proprio a questo livello.

Al livello OkkamSTORE è adibita la soluzione dei problemi che derivano dal *mapping* fra rappresentazione logica e fisica delle entità. A questo livello vengono inoltre memorizzati i puntatori che stabiliscono una relazione fra le entità contenute nel repository e i sistemi informativi esterni all'architettura, come ad esempio il *WWW*, le basi di conoscenza oppure i database relazionali. Dal punto di vista fisico questo livello si occupa di gestire l'immensa quantità di dati in modo distribuito con un paradigma *peer-to-peer* ed implementa a basso livello i meccanismi di risoluzione delle query.

Le seconda componente di un OkkamNODE sono i servizi esposti agli utenti come ad esempio gli OKKAMEmpowered Tools. Tali strumenti sono una serie di veri e propri applicativi e plugin integrabili ad applicazioni di uso comune che permettono alla comunità di utenti di creare e gestire contenuti OKKAMizzati.

E' stato ad esempio sviluppato un plugin integrabile con *Protégé*, uno degli editor di ontologie più diffusi, e sono in fase di studio ulteriori componenti per i più famosi applicativi di word-processing ed editor HTML affinché aumentino sempre più le opportunità di annotare le entità presenti nei documenti con gli identificatori di OKKAM. Per favorire la creazione e la proliferazione di questi strumenti il team OKKAM non farà affidamento solo sulle proprie capacità di sviluppo, ma verrà messa in atto una politica attiva di collaborazione con aziende fornitrici dei più famosi applicativi per coinvolgerle e permettere lo sviluppo integrato di estensioni OKKAM all'interno dei loro prodotti. Tale politica ha già dato buoni risultati con l'azienda *Microsoft®* e altri candidati, come *NeON* [26] ad esempio stanno pianificando la realizzazione di tali strumenti.

Vi sono inoltre ulteriori *OKKAM Services* che hanno principalmente lo scopo di permettere una veloce espansione di tutto il sistema. Esempi di questi servizi sono l'inserimento manuale di entità da parte degli utenti, l'inserimento aggregato di numerose entità attraverso l'utilizzo di un formato file specifico, oppure l'OKKAMizzazione automatica di contenuti attraverso analisi automatica del linguaggio.

[26] L'obbiettivo del progetto NeON è quello di produrre innovazioni nell'utilizzo di ontologie per l'utilizzo di applicazioni semantiche su larga scala. In particolare tenta di migliorare la capacità di gestire ontologie di rete multiple che appartengono a un particolare contesto, migliorandone la creazione, la gestione e la crescita. Per maggiori riferimenti http://www.neon-project.org.

Come si può notare in Figura 1.4 oltre ai livelli orizzontali implementati come servizi è stato previsto un *layer* trasversale, nominato *OkkamDEV*, che costituisce un *framework* di supporto per lo sviluppo di applicativi e costituito essenzialmente da APIs e una precisa documentazione. Anche se tale livello sembrerebbe fornire accesso a tutti gli strati della tecnologia occorre precisare che solo una parte delle funzionalità viene esposta pubblicamente, mentre le APIs relative allo sviluppo dell'OkkamCore rimarranno di uso esclusivo degli amministratori del sistema.

Dopo aver presentato il modello logico dell'architettura globale viene ora mostrata in Figura 1.5 l'infrastruttura fisica che implementa l'ENS. Dal punto di vista macroscopico, si ormai delineato come vi sia un sevizio OKKAM distribuito e globale, nominato OkkamPUBLIC, che provvede a realizzare l*'Entity Repository* e un'infrastruttura di servizi che permette a tools ed applicazioni di utilizzare questa nuova tecnologia.

Tuttavia oltre all'interazione diretta fra gli utenti e OkkamPUBBLIC, che non necessità di nessun particolare vincolo, se non la connessione alla Rete, è stata prevista la possibilità allocare OkkamNODE anche all'interno di organizzazioni ed enti privati che intendono usufruire del sistema di pubblicazione delle entità offerto da OKKAM.

Tale operazione non contribuisce in nessun modo alla diffusione di dati protetti e privati dal momento che l'intento principale è quello di strutturare i dati interni di un'organizzazione in modo che siano direttamente interfacciabili con il repository OKKAM, mentre il controllo della loro pubblicazione rimane di dominio dell'ente che possiede le informazioni. La sincronizzazione dei nodi OKKAM privati, definiti anche con il termine *corporate*, con il sistema centrale OkkamPUBLIC permette inoltre la precisa selezione delle informazioni che un certo ente desidera rendere disponibile pubblicamente.

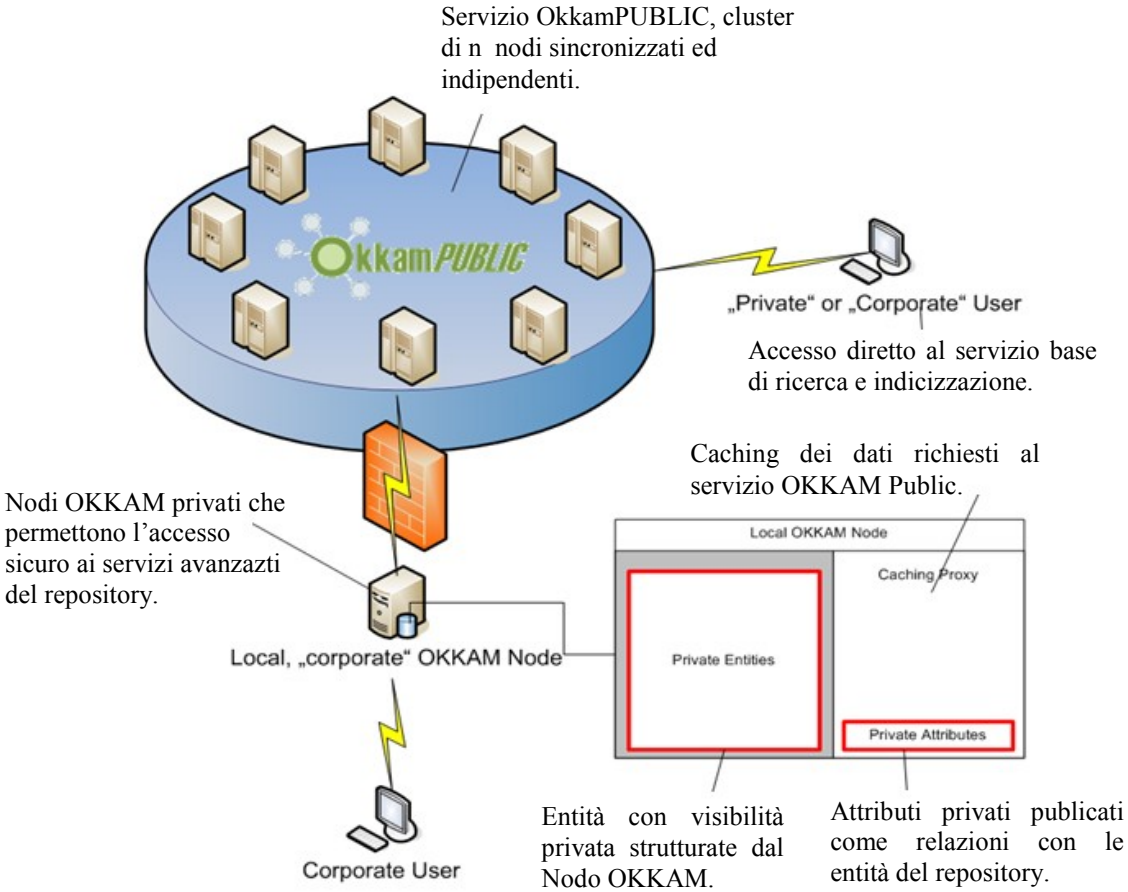

Figura 1.5 – Architettura ad alto livello del servizio *OkkamPUBLIC*

Le caratteristiche appena descritte completano la visione dell’intera architettura che il progetto OKKAM sta cercando di realizzare, è possibile comunque valutare gli sviluppi dei diversi *Work Package* e informarsi sugli eventi divulgativi visitando il portale *http://www.okkam.org*.

1.5 Confronto fra i progetti Linked Data e OKKAM

Dopo aver presentato le caratteristiche di due dei maggiori progetti in fase di sviluppo per l'avanzamento tecnologico in direzione del Web Semantico ne verrà ora effettuato un confronto per mettere in evidenza le differenze e i punti in comune.

Attualmente il progetto Linked Data è in fase di realizzazione più avanzata rispetto ad OKKAM e rappresenta il principale punto di riferimento a livello mondiale, sia per le sue dimensione sia per il fatto che è stato proposto dal massimo esponente del World Wide Web, Tim B. Lee. Tuttavia è possibile affermare che entrambi gli approcci sono basati su principi comuni.

Sia Linked Data che OKKAM fondano le proprie radici nel fatto che per raggiungere la visione del Web Semantico come una base di conoscenza distribuita e di enormi dimensioni è necessario trovare il modo di connettere gli insiemi di descrizioni o relazioni appartenenti agli stessi oggetti memorizzati all'interno di sorgenti dati RDF/OWL differenti.

Di seguito vengono riproposti in breve i principi base del progetto Linked Data per verificare le similarità con le linee guida del progetto OKKAM:

- ✓ Usare gli URIs per nominare le risorse;
- ✓ Usare gli URIs HTTP affinché le persone possano risolvere i nomi;
- ✓ La risoluzione di un URI deve portare a informazioni utili;
- ✓ Includere i link ad altri URIs affinché si possano scoprire nuove cose.

A un livello logico molto alto questi principi sono gli stessi seguiti dal progetto OKKAM infatti in esso:

- ✓ Gli URIs sono fondamentali è in più si tenta di evitarne la proliferazione;
- ✓ L'ENS provvede a fornire la risoluzione degli URIs basandosi sul protocollo HTTP;
- ✓ Quando un URI viene dereferenziato l'ENS provvede a fornire il profilo dell'entità a cui si riferisce, informazioni che vengono memorizzate per rendere tale entità recuperabile e differenziabile da tutte le altre;
- ✓ Vengono mantenuti nell'Entity Repository riferimenti a sorgenti dati esterne che contengono dati, informazioni e descrizioni delle entità.

È importante notare che all'interno del progetto OKKAM solitamente non si parla di risorse generiche, come quelle descritte all'interno di un documento RDF, ma si effettua la distinzione fra le entità reali e gli oggetti astratti, similarmente alla distinzione che si effettua nel linguaggio OWL fra oggetti individuali e classi.
La ragione di questa scelta deriva dal fatto che raramente le classi sono definite in modo univoco e precisamente distinto, mentre le entità rappresentano un risorsa atomica completamente distinguibile da tutte le altre.
Una delle principali critiche mosse contro il progetto Linked Data dai promotori di OKKAM riguarda il fatto che esso ha adottato un approccio *a posteriori* rispetto alla proliferazione degli identificatori di risorse ed entità.
Questa scelta, all'interno di un ambiente altamente dinamico a dai ritmi di crescita esponenziali come il Web, potrebbe risultare infelice dal momento che sarà necessario prevedere una ricerca, un'analisi e un allineamento continui dei nuovi dati.
Si deve tenere presente che l'operazione di notifica dei nuovi contenuti potrà essere eseguita in parte direttamente da chi li pubblica, in base alla propria sensibilità e conoscenza delle tecnologie per il Web Semantico, tuttavia questa opportunità non argina completamente i dubbi esposti precedentemente.
Inoltre implementare logiche di ragionamento che facciano uso della relazione di identità fra ontologie distribuite potrebbe risultare estremamente complesso.
Teoricamente per riconoscere tutte le identità presenti nel Web sarebbe necessario la computazione della chiusura transitiva su tutti i dati pubblicati, ed è noto che tale operazione, applicata a questa enorme quantità di informazioni, raggiunge una complessità superiore a quella effettivamente risolvibile dagli attuali sistemi in tempi ragionevoli.
E' evidente che l'approccio del progetto Linked Data inoltre non fornisce una soluzione al problema della presenza di identificatori multipli per la stessa entità, anzi concede la loro proliferazione.
Questo ha portato a pensare che tale progetto attualmente sia la soluzione più valida e realizzabile per un migliore supporto alle operazioni di *browsing* ma non fornisce un struttura efficiente per le operazioni di *reasoning* e *querying*.

D’altra parte il progetto OKKAM con la realizzazione dell’ENS punta immediatamente a evitare che vi sia una proliferazione degli identificatori e fornisce una soluzione strutturale affinché ciò non avvenga.

Una critica che può essere mossa alla struttura ENS riguarda il fatto che rappresenti un servizio centralizzato e quindi in opposizione alle caratteristiche del Web. Tuttavia gli sviluppatori hanno fatto notare che l’ENS non è stato a caso paragonato con il DNS, servizio che in passato ricevette giudizi negativi per gli stessi motivi che si sono poi mostrati superflui rispetto ai vantaggi apportati da esso.

Nello stesso modo gli sviluppatori giustificano le metodologie scelte per l’implementazione dell’ENS assicurando che esso potrebbe diventare una componente essenziale per rendere operativo il Web Semantico.

Questa analisi potrebbe indurre a pensare che i due progetti siano incompatibili tuttavia ciò non è vero, Linked Data potrebbe dare un contributo notevole nell’allineamento dei dati che risultano essere già pubblicati da tempo sul Web, mentre OKKAM potrebbe fornire un valido strumento per il riutilizzo degli identificatori evitandone in questo modo l’inutile proliferazione.

Questi due sforzi combinati forniscono il potenziale per raggiungere uno stato i cui vi sia una sempre minore necessità di eseguire l’allineamento dei dati presenti sul Web e in cui si possa raggiungere una reale integrazione di tutte le basi di conoscenza già esistenti all’interno di unico e universale *repository*.

2 GESTIONE E RAPPRESENTAZIONE DELLA CONOSCENZA

Non esiste un'unica e conclusiva definizione di knowledge management. In senso lato, il concetto può riferirsi alla preservazione e alla condivisione della conoscenza ed è portato avanti dall'antichità con lo sviluppo di biblioteche e strumenti di comunicazione. Nei tempi più recenti della rivoluzione digitale , chiamiamo knowledge management quel filone di ricerca teorica e applicativa che sviluppa il ciclo della conoscenza all'interno di una comunità o dominio.

Sempre più spesso nella società attuale l'informazione viene trattata come un prodotto e come tale è soggetta e diversi processi che la vedono coinvolta. Analiticamente è possibile descrivere l'intero ciclo a cui l'informazione e la conoscenza vengono sottoposte.

Essenzialmente tale ciclo contempla l'attività di acquisizione dell'informazione in qualche formato generico (audio, video, testuale, ecc…), la sua rappresentazione, che può prevedere la trasformazione in un formato differente da quello originale, e infine vi è il processo di utilizzo dell'informazione.

Nel 1986 Karl Wiig, autore nel 1993 del libro *Knowledge management foundations*, enuncia i principi del knowledge management, termine da lui coniato. Molte aziende, soprattutto multinazionali, mostrarono un forte interesse verso questa teoria.

L'obiettivo primario del knowledge management, infatti, è pragmatico: migliorare l'efficienza dei gruppi collaborativi esplicitando e mettendo in comune la conoscenza che ogni membro ha maturato durante il suo percorso professionale. I primi investimenti si concentrano soprattutto sullo sviluppo dei mezzi per rendere veloce e semplice l'archiviazione, la descrizione e la comunicazione di dati e informazioni. Questa prima fase tende a ridurre il knowledge mangement alla sua componente strumentale, l'*information technology*, che è fondamentale ma non ne esaurisce le potenzialità.

La seconda fase del knowledge management si focalizza su come poter mettere a servizio di tutti le conoscenze specifiche acquisite. Quello della conoscenza è un ciclo che può portare alla produzione di nuova conoscenza solo tramite la condivisione e l'elaborazione di informazioni.

In tale contesto ha assunto un ruolo sempre più fondamentale il formato dell'informazione, ovvero la disciplina nota con il termine di *Knowledge Representation*.

Prima di prendere in esame il connubio "rappresentazione della conoscenza e Web Semantico", ci soffermeremo sul contributo che la rappresentazione della conoscenza ha dato nell'ambito dell'Intelligenza Artificiale.

Questo permetterà di comprendere più a fondo i principi generali dell'argomento e fornisce le basi per poter effettuare un paragone con quanto sviluppato fino ad ora nell'ambito del Web Semantico.

Il settore dell'Intelligenza Artificiale (IA), conosciuto come rappresentazione della conoscenza, ebbe origine nella seconda metà degli anni sessanta.

Un sistema di Knowledge Representation deve soddisfare due requisiti primari:

- ✓ il linguaggio di rappresentazione: un insieme di strutture sintattiche, adatte a codificare le informazioni che si devono rappresentare e che possano essere implementate nella memoria di un computer;
- ✓ un insieme di regole che consentano di manipolare le strutture sintattiche: l'applicazione di tali regole deve consentire di ottenere le inferenze desiderate, inoltre le regole devono poter essere formulate sotto forma algoritmi implementabili su calcolatore.

Tali requisiti risultano essere soddisfatti dai sistemi formali sviluppati in logica matematica. Oggetto di studio della logica, disciplina assai più antica dell'IA, sono i nessi inferenziali tra enunciati. Col termine enunciato, o proposizione, ci si riferisce a "qualunque espressione linguistica che possa essere vera oppure falsa".

Un'inferenza, invece, è un processo che, a partire da alcuni enunciati di partenza, le premesse, consente di asserire un altro enunciato, la conclusione.

Tradizionalmente la logica si occupa solo di inferenze deduttivamente valide . In logica, lo studio delle inferenze valide si basa su di un processo di formalizzazione il quale si poggia su di:

- ✓ un linguaggio formale in cui esprimere come formule premesse e conclusioni;
- ✓ delle regole di inferenza che, operando sulle formule, consentano di derivare delle conclusioni dalle premesse.

La logica proposizionale cattura le forme più semplici di inferenza logica, ovvero quelle forme di inferenza verificabile senza prendere in considerazione la struttura interna delle proposizioni atomiche. Le proposizioni atomiche non possono essere scomposte in ulteriori proposizioni tuttavia possono essere combinate per creare delle proposizioni complesse.

Il linguaggio formale della logica proposizionale consente di rappresentare le proposizioni composte sfruttando dei connettivi proposizionali (negazione, congiunzione, disgiunzione, condizionale materiale).

Il significato dei connettivi può essere schematizzato attraverso le tavole di verità, nelle quali è possibile vedere come al variare del valore di verità delle formule A e B, varia il valore di verità delle formule ottenute utilizzando i diversi connettivi.

Tuttavia non tutte le inferenze valide possono essere formalizzate con gli strumenti della logica proposizionale. È necessario talvolta "smontare" le proposizioni atomiche e considerare la loro struttura interna. Per poter fare ciò si bisogno di un logica maggiormente espressiva quale la *logica dei predicati del primo ordine* che risulta essere caratterizzata da:

- ✓ un alfabeto di simboli;
- ✓ un insieme di termini (che dovrebbero denotare gli "oggetti" dell'insieme che si sta considerando);
- ✓ un insieme di formule rappresentato da un insieme di stringhe composte di simboli dell'alfabeto che vengono considerate sintatticamente corrette.

La logica dei predicati del primo ordine è una estensione della logica proposizionale. La differenza risiede nel fatto che il linguaggio consente di rappresentare la struttura interna delle proposizioni atomiche e che le regole di inferenza tengono conto di tale struttura. Inoltre per potenziare la capacità rappresentativa della logica dei predicati è necessario introdurre altri due elementi:

- ✓ variabili individuali che consentono di rappresentare individui generici del dominio;
- ✓ quantificatori: espressioni come "qualcosa" (quantificatore esistenziale) e "ogni cosa" (quantificatore universale).

I linguaggi formali risultano utili quando ci si trova a dover lavorare in un ambiente deterministico che possa essere descritto in modo agevole tramite enunciati logici. Agenti intelligenti informatici potrebbero essere in questo modo in possesso di strumenti per poter effettuare processi nettamente più complessi di quelli che possono essere eseguiti attualmente, ad esempio, dai tradizionali motori di ricerca.

2.1 I principali modelli di riferimento

Una delle principali alternative alla logica è rappresentata dalle *reti semantiche*. Le reti semantiche costituiscono una classe di sistemi di rappresentazione tipici dell'Intelligenza Artificiale. L'idea che ne è alla base è quella di utilizzare come strumento di rappresentazione un grafo, in cui ad ogni nodo è associata un'entità concettuale e in cui le relazioni fra le entità concettuali sono rappresentate mediante archi che connettono i nodi, come mostrato in Figura 2.1.

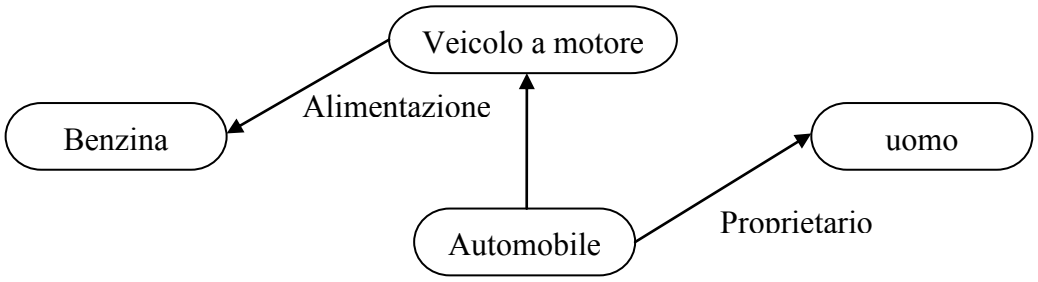

Figura 2.1 – Esempio di rete semantica.

Se la struttura generale può essere descritta agevolmente in questo modo bisogna anche precisare che sono stati sviluppati nel tempo diversi tipi di reti semantiche.

Rispetto ai modelli basati sulla logica, le reti semantiche avrebbero dovuto consentire di rappresentare la conoscenza con strutture associative simili a quelle ipoteticamente utilizzate dalla memoria umana. Nell'ambito dell'IA, le reti semantiche furono sviluppate a partire dal modello computazionale di M. Ross Quillian, i cui interessi erano di tipo prevalentemente psicologico sulla struttura della memoria e sulla rappresentazione della conoscenza lessicale. L'obiettivo di Quillian era quello di fornire un modello dell'organizzazione della memoria semantica di un essere umano, e dai suoi studi risulta che i concetti o significati delle parole nella mente sono interconnessi mediante una struttura reticolare. In tale modello erano presenti molte caratteristiche

tipiche dei sistemi a rete semantica successivi: ad esempio gli archi che rappresentano relazioni di sottoclasse fra concetti diventeranno un costrutto fondamentale di quasi tutte le reti semantiche. Tali meccanismi evolveranno in seguito nel principio dell'ereditarietà fra concetti, in base a cui i concetti più specifici ereditano le caratteristiche dei loro superconcetti più generali.

Allan M. Collins e Quillian nel 1970 presentarono un tipo di rete che ebbe grande importanza per gli sviluppi successivi della ricerca, privilegiando il ruolo degli archi dei nodi a livello più alto. Le proprietà più generali venivano introdotte ai livelli più alti della tassonomia e diventava esplicito in tal modo il meccanismo dell'ereditarietà, inoltre veniva esplicitato il meccanismo per cui proprietà definite a livelli più specifici potevano "cancellare" proprietà incompatibili derivanti dai livelli superiori, implementando in questo modo una sorta di polimorfismo. Ciò permette di modellare il fatto che non tutti gli individui appartenenti a una certa classe non siano obbligatoriamente conformi a uno stereotipo generalmente diffuso. Importanti innovazioni furono successivamente introdotte da Jaime R. Carbonell, quali la distinzione fra *concept units* e *example units*, ovvero nodi che rappresentano concetti generali e nodi che rappresentano istanze specifiche. Una caratterizzazione maggiormente dettagliata delle reti semantiche nell'ambito della linguistica verrà fornita nel paragrafo 2.4, insieme alla descrizione di *WordNet*.

Un diverso tipo di sistema di rappresentazione della conoscenza, i *frame*, fu proposto negli anni '70 da Marvin Minsky, uno dei padri fondatori dell'Intelligenza Artificiale. Essi costituiscono un formalismo strutturato per la rappresentazione della conoscenza che è strettamente imparentato con le reti semantiche, e che è centrato sulla rappresentazione di prototipi. L'idea di fondo è che la memoria umana sia strutturata in un insieme di schemi, di rappresentazioni standard di oggetti e situazioni. Posti di fronte a situazioni nuove, gli esseri umani identificano lo schema che meglio si applica ai dati disponibili, e agiscono sulla base delle informazioni che tale schema mette a disposizione e delle aspettative che esso comporta. I frame sono strutture dati complesse, che dovrebbero modellare questo genere di schemi. Un frame è costituito da diversi campi (*slot*), ognuno dei quali rappresenta una delle caratteristiche o degli attributi del prototipo rappresentato. Si riporta in Tabella 2.1 parte di un possibile frame che rappresenta il concetto "essere umano". Sulla sinistra sono riportati i nomi degli slot

e sulla destra i valori che fungono da valori degli slot, ed eventuali altre informazioni aggiuntive; tipicamente, restrizioni sui tipi di valori che uno slot può assumere. In generale, gli slot denominati "*è un"* indicano che un certo frame rappresenta un caso particolare di concetti rappresentati da altri frame. Nel nostro esempio, un essere umano è un tipo particolare di primate. Gli slot successivi rappresentano altri attributi del concetto. Ad esempio, lo slot "arto superiore" ha come valore "braccio", ad indicare che un essere umano ha degli arti superiori che sono braccia. L'espressione "*(card.: 2)"* esprime una restrizione di cardinalità sugli slot "arto superiore" e "arto inferiore": un essere umano ha esattamente due arti inferiori e due arti superiori. L'espressione "*(giorno/mese/anno)"* nello slot "*data di nascita*" indica quale deve essere il formato dei riempitori dello slot (in questo caso una data), e l'espressione "*(compresa tra 0 e 100)"* nello slot *"età"* indica l'intervallo dei valori che possono essere assunti da "*età"*.

nome del frame:	***essere umano***
è un:	***primate***
...	*...*
arto superiore:	***braccio*** *(card.: 2)*
arto inferiore:	***gamba*** *(card.: 2)*
...	*...*
data di nascita:	*(giorno/mese/anno)*
età (in anni):	*(compresa tra 0 e 100)*
...	*...*
genitore:	***essere umano*** *(card.: 2)*
antenato:	***essere umano***
...	*...*

Tabella 2.1 – Esempio di *frame* generico.

Un frame eredita gli slot con i relativi attributi dai frame più generali che sono riempitori degli slot "*è un*". Ad esempio, presumibilmente "essere umano" eredita tutta una serie di informazioni (inerenti, poniamo, la biologia) dal frame "primate". Analogamente, il frame "Davide Limentani", in Tabella 2.2, rappresenta una specifica istanza di "essere umano". In quanto tale, esso eredita gli slot di "essere umano" con le informazioni ad essi collegate. In taluni casi, queste informazioni vengono ulteriormente specificate (come nel caso della data di nascita, o del nome dei genitori).

nome del frame:	***Davide Limentani***
è un:	***essere umano***
...	*...*
data di nascita:	*4/ottobre/1956*
età (in anni):	*44*
...	*...*
genitore:	***Isacco Limentani***
	Sara Piperno
antenato:	***Daniele Limentani***
	Lia Sonnino
	Davide Piperno
...	*...*

Tabella 2.2 – Esempio di *frame* specifico.

Le informazioni associate agli slot possono essere intese come informazioni di default, che valgono nei casi tipici, e vengono cancellate in presenza di informazioni più specifiche. Ad esempio, la restrizione associata allo slot *"età"* ci dice che tipicamente gli esseri umani hanno un'età compresa tra 0 e 100 anni, ma ciò non esclude che possano esserci istanze eccezionali di "*essere umano"* con un valore di età superiore a 100. Nelle tabelle sopra riportate le scritte in grassetto corrispondono a nomi di altri frame. Così, ad esempio, si suppone che esista un frame che descrive il concetto "braccio" e un altro che descrive l'individuo "Sara Piperno*"*. In questo modo ciascun frame in una base di conoscenza rimanda ad un certo numero di altri frame, dando luogo ad una rete associativa molto simile a una rete semantica. Rispetto al formalismo delle reti semantiche, gli slot di tipo "*è un"* sono l'analogo degli archi di sussunzione e di individuazione, mentre gli altri slot corrispondono ad attributi. Una rete di questo tipo prende il nome di *frame system*. Un aspetto che caratterizza i frame è costituito dalla possibilità di introdurre delle forme di *collegamento procedurale* (*procedural attachment*), di collegare cioè delle procedure ad alcuni slot di un frame. Vediamo un semplice esempio. Nel frame *"essere umano"* sopra riportato sono presenti gli slot *"data di nascita"* ed "*età"*. Ma, per ogni istanza di "*essere umano"*, una volta dato il valore di "*data di nascita"*, il valore di "*età"* può essere facilmente calcolato a partire dal valore della data corrente. Si può allora collegare allo slot *"età"* una procedura che esegua questo semplice calcolo. Analogamente, dati i valori dello slot *"genitori"*, si possono ricavare gli antenati di un'istanza di *essere umano* andando a cercare nella base di conoscenza i genitori dei suoi genitori, e così via ricorsivamente. Così, anche i riempitori dello slot *"antenato"* possono essere calcolati per mezzo di un semplice collegamento procedurale. I frame sono un formalismo di rappresentazione di tipo

principalmente dichiarativo, tuttavia la tecnica del procedural attachment consente di integrare in essi elementi di rappresentazione procedurale della conoscenza.

Un punto di svolta nella ricerca sulla rappresentazione della conoscenza è costituito dai lavori di W.Woods. Quest'ultimo partì dalla constatazione che, al di là dell'idea generale della struttura a grafo per la rappresentazione della conoscenza, i vari sistemi fino ad allora proposti erano quanto di più eterogeneo si potesse pensare.

Non esisteva, secondo Woods, alcuna teoria delle reti semantiche e propose pertanto di introdurre nelle reti semantiche un formalismo rigoroso, limitando inoltre il tipo di relazioni rappresentabili tramite gli archi fra i nodi.

Come risposta alla richiesta di rigore di Woods, R. Brachman propose una classe di formalismi detti *reti semantiche ad ereditarietà strutturata*.

Verso la fine degli anni '70, seguendo i principi di queste ultime, R. J. Brachman sviluppò un sistema di rappresentazione della conoscenza, denominato *KL-ONE*, che ebbe grande influenza sugli sviluppi successivi della ricerca. Questo sistema può essere considerato il capostipite dei sistemi di rappresentazione della conoscenza basati sulle logiche descrittive. Per una migliore comprensione di *KL-ONE* e degli altri sistemi considerati si rimanda alle fonti riportate in bibliografia.

2.2 Informazioni strutturate e non strutturate

Verranno introdotti in questo paragrafo i concetti di informazioni o dati *strutturati* e *non strutturati*. Come appreso nel primo capitolo una condizione necessaria per la realizzazione del Web Semantico è la presenza di documenti autodescrittivi e che il formalismo utilizzato per l'elaborazione delle descrizioni sia standardizzato e comprensibile da parte di applicativi software.

Si stima, anche se misurazioni precise non sono effettuabili con certezza, che l'80% delle informazioni presenti nel Web non siano effettivamente rintracciabili per cause inerenti alla mancanza di una descrizione formalizzata. Tali dati vanno a costituire il cosiddetto *Web nascosto*, ovvero rappresentano risorse informative disponibili sul Web ma non rintracciabili dai motori di ricerca.

Le informazioni strutturate sono caratterizzate da un'organizzazione formale dei dati sia per quanto riguarda il ruolo semantico sia per quanto riguarda il tipo (numero, testo o altro). Altra caratteristica fondamentale dei dati strutturati è la precisa disambiguazione delle informazioni, ovvero la possibilità di interpretare in modo deterministico il significato di ogni dato.

Un esempio estremo di informazioni strutturate sono quelle contenute all'interno di un database, tali informazioni sono altamente usufruibili sia dal punto di vista atomico sia dal punto di vista delle relazioni instaurate fra le informazioni stesse.

Le persone utilizzano inconsciamente ogni giorno le informazioni non strutturare per la creazione, la memorizzazione o la ricerca di reports, email o altri tipi di documenti. Le informazioni non strutturate consistono in dati memorizzati senza una definizione concettuale dei contenuti e senza che venga definito il tipo dei dati.

Occorre porre attenzione nell'utilizzo del termine strutturato perché per definizione ogni file possiede una propria struttura interna, tuttavia non è ad essa che ci riferisce quando, in questo ambito, si parla di dati non strutturati.

I dati non strutturati sono essenzialmente di due tipologie:

1. *oggetti bitmap*: come ad esempio files immagini, video o audio;
2. *oggetti testuali*: come ad esempio files del noto applicativo Notepad.

Entrambi possono essere classificati come dati, tuttavia la tecnologia per sfruttare informazioni rilevanti da oggetti bitmap è ancora a livelli embrionali, mentre per quanto

riguarda gli oggetti testuali esistono numerosi strumenti, disponibili sul mercato, in grado di effettuare *data mining* e altre tipologie di analisi.

Le informazioni non strutturate mettono in evidenza le seguenti problematiche:

1. anche se i dati non strutturati sono rappresentati da un formato come *Microsoft Word template*, essi non sono utilizzabili da un punto di vista semantico;
2. anche nel caso in cui fosse possibile comprendere a livello informatico i dati non strutturati non è detto che se ne riesca a comprendere il contesto almeno che non ci si metta fisicamente a leggere il contenuto.
3. In ultimo, il modo di interpretare quello che leggiamo è estremamente soggettivo, caratteristica non plausibile per un sistema deterministico come un computer.

A livello tecnologico si stanno cercando metodologie per la comprensione automatica del linguaggio naturale, per realizzare ciò che in termine tecnico viene definito *POS (Part-of-Speech) tagging*. Questa elaborazione permette di marcare automaticamente, con il linguaggio XML, le parole di un testo piano assegnando ad esse ad esempio il ruolo grammaticale, logico o una relazione con la altre parole contenute nel testo.

Il POS tagging rappresenta quindi un punto cruciale per la conversione dei dati non strutturati in dati strutturati anche se risulta essere una sfida attualmente aperta.

A tale proposito un progetto molto interessante, già attualmente valutabile online, è *OpenCalais*[27]. OpenCalais è attualmente strutturato come un Web Service che crea automaticamente metadati che arricchiscono il testo inserito come input. Utilizzando l'analisi del linguaggio naturale, algoritmi di *machine learning* e altri metodi, Calais analizza il documento in testo piano e trova le entità celate al suo interno, come persone, aziende, luoghi e altre informazioni di grande utilità. Inoltre è possibile esportare tali informazioni in formato RDF, generato automaticamente dall'applicativo.

L'analisi di documenti non strutturati rappresenta il fulcro di questa tesi sviluppata presso *Expert System*, azienda leader nel settore dell'analisi linguistica.

[27] http://www.opencalais.com

2.3 Ontologie semantiche

Nella storia del pensiero occidentale sono state date diverse definizioni di ontologia, ognuna delle quali assume un'accezione differente a seconda del contesto considerato.
Dal punto di vista filosofico una delle prime definizione è stata data da Aristotele, il quale afferma che: "*l'ontologia è la scienza dell'essere in quanto essere*"[28].
Nel 1993 T. R. Gruber trasferì l'idea di ontologia al mondo dei sistemi informativi, dandone la seguente definizione:"*un'ontologia è una rappresentazione formale, condivisa ed esplicita di una concettualizzazione di un dominio di interesse*"[29]. Con "concettualizzazione" egli intende un modello astratto di un fenomeno che ne definisce gli aspetti più rilevanti.
Ed è proprio quest'ultima la definizione cui ci si ispira nell'ideazione di strumenti informatici basati su ontologie. Infatti, in questi casi, l'obbiettivo principale è proprio quello di rappresentare in maniera formale i concetti ascrivibili al dominio applicativo in esame.
Il corpo di tale rappresentazione formale è basata su una concettualizzazione e, cioè, sulla specificazione del significato inteso dei termini coinvolti nel dominio applicativo in esame. Specificare la concettualizzazione vuol dire attribuire un significato non ambiguo ai termini che definiscono la conoscenza in un preciso dominio.
Le ontologie sono nate nell'ambito dell'Intelligenza Artificiale con lo scopo di facilitare la condivisione e il riutilizzo della conoscenza su larga scala e per fornire una semantica alle informazioni, processabile dalle macchine e quindi comprensibile sia ad agenti umani che software. Dato un dominio, l'ontologia ne chiarisce la struttura della conoscenza, creando una sintassi da associare ai termini e da condividere con tutti coloro i quali devono interagire con lo stesso dominio, eliminando la duplicazione del processo di analisi.
A seconda del livello di generalità, si possono distinguere vari tipi di ontologie, che ricoprono ruoli differenti all'interno del processo di creazione di un *Knowledge Base System*:

- ✓ ontologie di dominio (*domain*) – riguardano un particolare campo di applicazione (elettronico, medico, ecc…). Bisogna tenere ben presente che la

[28] Aristotele, *Metafisica*, IV,1
[29] http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

modellazione delle ontologie è una modellazione per classi. Il meccanismo dell'ereditarietà consente di definire una unica volta gli attributi che classi ad uno stesso livello ereditano da un padre. La possibilità di definire come valore di un attributo un'altra classe consente di stabilire qualsiasi tipo di relazione fra classi;

- ✓ ontologie dei metadati (*Metadata*) – forniscono un vocabolario descrittivo del contenuto di informazioni on-line e un esempio ne è il sistema *Dublin Core*[30];
- ✓ ontologie generiche (*Upper* o *Top-level*) – sono valide attraverso vari domini perché riguardano concetti generici e di uso comune;
- ✓ ontologie di scopo (*Task* o *Method*) – riguardano concetti utilizzati per particolari funzioni ed attività, specificando i termini introdotti nelle ontologie top-level, rappresentano la struttura dei processi;
- ✓ ontologie *Application* – descrivono concetti dipendenti sia da un particolare dominio che da un task, e sono di solito una combinazione di tutte le sub-ontologie per applicazione.

Un'ontologia è quindi un insieme di concetti (entità, attributi, processi), di definizioni e di relazioni fra i concetti, le quali possono essere di vario tipo: tassonomico (*IS-A*), meronimico (*PART-OF*), telico (*PURPOSE-OF*) e così via.

Dunque è possibile vedere un'ontologia come una rete semantica di concetti appartenenti ad un dominio legati tra loro dalle suddette relazioni.

Formalmente un ontologia può essere definita come una Tripla O = {C, R, A}, dove C è un insieme di concetti del dominio di interesse, R è un insieme di relazioni tra i concetti appartenenti a C e A è un insieme di assiomi (Se A = ∅ l'ontologia si dice non assiomatizzata). Gli insiemi C ed R individuano un Grafo G = {(V,E)}, tale che: V ≡ C, E = {(c1, c2) ∈ C×C}}.

Un' ontologia può presentare vari livelli di formalizzazione, ma deve necessariamente includere un vocabolario di termini (*concept names*) con associate definizioni (*assiomi*), e relazioni tassonomiche, inoltre è sbagliato affermare che un'ontologia rappresenta autonomamente l'intera conoscenza di un dominio.

[30] Il Dublin Core è un sistema di metadati costituito da un nucleo di elementi essenziali ai fini della descrizione di qualsiasi materiale digitale accessibile via Rete. http://dublincore.org.

Per arrivare ad avere una *Knowledge Base* (KB) completa, ovvero la completa base di conoscenza che rappresenta fedelmente ed esaurientemente un determinato dominio d'interesse, dobbiamo aggiungere un insieme di fatti ed un insieme di regole.
Dunque possiamo definire Knowledge Base l'insieme costituito da un:

- ✓ ontologia: schema concettuale del dominio in questione, in cui si definiscono degli assiomi sui concetti da rappresentare;
- ✓ insieme di fatti: ovvero di istanze concrete dei concetti definiti nell'ontologia;
- ✓ regole: insieme di regole tramite le quali inferire nuova conoscenza.

Esistono essenzialmente tre approcci percorribili per la realizzazione di un'ontologia: l'approccio *Bottom-Up*, l'approccio *Top-Down* e l'approccio misto. Nell'approccio *Bottom-Up* la progettazione avviene dal basso verso l'alto, ovvero si individuano prima tutti i concetti più specifici, poi quelli più generali. Tale procedimento va iterato fino al raggiungimento di un livello di astrazione e generalizzazione sufficiente per gli scopi per cui viene progettata l'ontologia. I concetti saranno organizzati secondo una tassonomia che rappresenta la gerarchia degli stessi. Come per il livello di generalizzazione, il livello di specializzazione dei concetti individuati deve tener conto degli scopi dell'Ontologia.
Nell'approccio *Top-Down* la progettazione dell'ontologia avviene invece dall'alto verso il basso, ovvero individuando prima i concetti più generali, e poi quelli più specifici, arrivando ad un livello di specializzazione adeguato per gli scopi dell'ontologia. Anche qui i concetti saranno organizzati in una tassonomia che rispetta la gerarchia degli stessi.
Nell'approccio misto, come suggerisce il nome, si procede individuando prima tutti i concetti che si ritengono importanti per il dominio da rappresentare, poi i legami gerarchici che legano i concetti per organizzare gli stessi dal più generale al più specifico.
Attualmente sono in fase di studio anche metodologie semi-automatiche per la realizzazione di ontologie di dominio che vedono un inteso utilizzo di tecniche di analisi automatica del linguaggio naturale per recuperare i termini principali contenuti all'interno di numerosi documenti inerenti a un particolare settore. Tali tecniche non possono essere considerate totalmente automatiche dal momento che l'intervento di un

esperto del dominio è necessario per poter validare la varie fase di questo processo di auto-apprendimento dell'ontologia.

Uno degli aspetti più avvincenti resi possibili dalle ontologie è la reale automatica integrazione dei dati, ovvero il processo attraverso il quale dati provenienti da sorgenti dati eterogenee vengono combinati e restituiti all'utente attraverso una visione unificata.

Ci sono tre differenti vie di applicazione delle ontologie alla semantica delle informazioni e alla loro integrazione: l'approccio *single ontology*, l'approccio *multiple ontology* e l'approccio *hybrid ontology*. Gli schemi concettuali di questi approcci vengono mostrati in Figura 2.3.

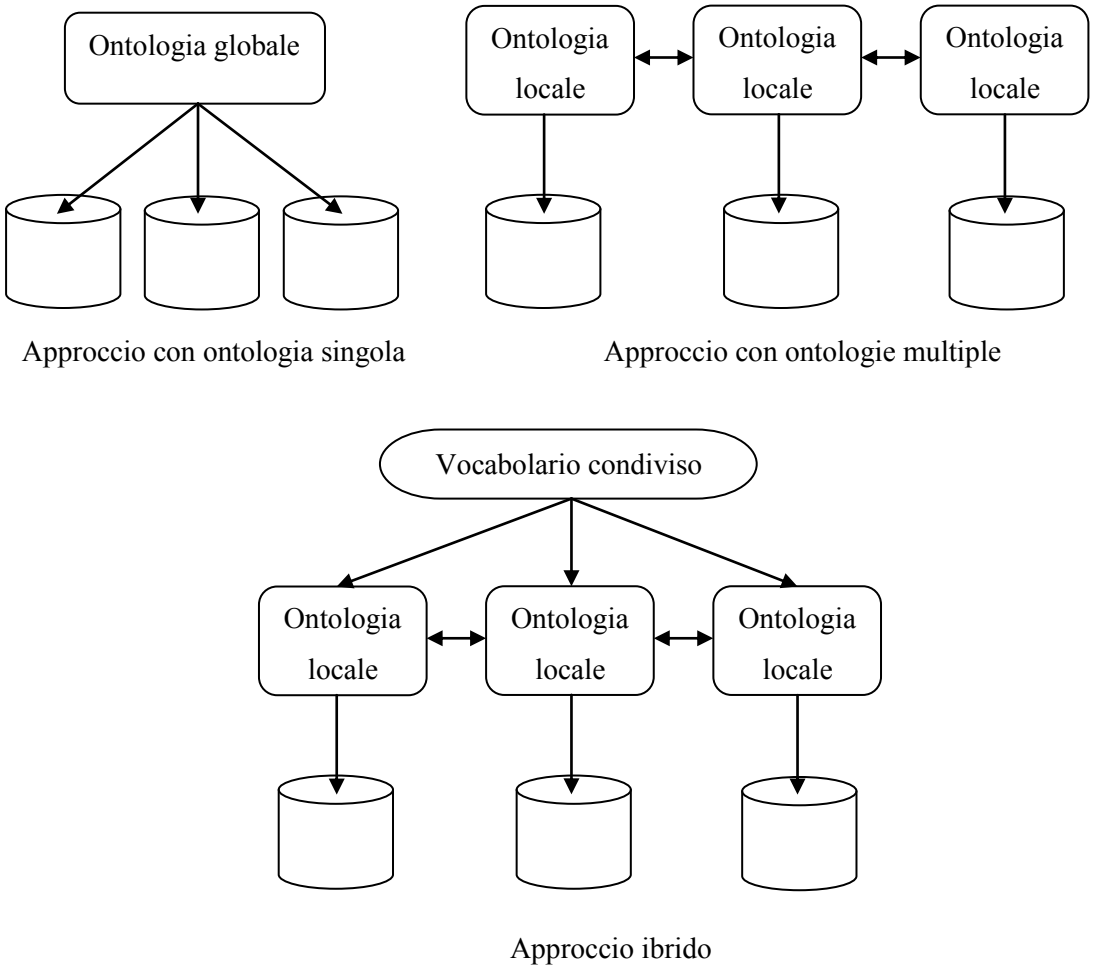

Figura 2.2 – Approcci di utilizzo delle ontologie.

L'approccio *single ontology* utilizza un'ontologia globale per la definizione di un vocabolario semantico condiviso. L'ontologia globale può essere ottenuta anche coniugando diverse ontologie in un'ottica di modularizzazione della conoscenza; in ogni caso questo sistema è adatto per problemi di integrazione di fonti di informazione

abbastanza omogenee tra di loro. Nell'approccio *multiple ontology*, vengono definite delle ontologie locali per ogni fonte di informazione e si applica una successiva fase di mappatura delle varie ontologie che non sempre risulta lineare e fattibile. L'approccio ibrido fonde le caratteristiche dei due precedenti, introducendo un vocabolario condiviso, che spesso è rappresentato da un'ontologia vera e propria, contenente i termini base del dominio di interesse, col cui linguaggio sono poi descritte le ontologie locali.

Parallelamente allo sviluppo della teoria delle ontologie sono nati linguaggi per la loro realizzazione e gestione. Questi linguaggi formali forniscono inoltre regole di ragionamento, basate sulla logica del primo ordine e sulla logica descrittiva, che supportano l'utilizzo avanzato della conoscenza presente nelle ontologie. Solitamente tali linguaggi sono di tipo dichiarativo e sono distinguibili in due categorie a seconda che fondino la propria semantica sulla sintassi o sulla struttura.

Linguaggi basati sulla struttura sono: *FLogic*, *OKBC* (*Open Knowledge Base Connectivity*), KL-ONE, *CycL* e *KIF* (Knowledge Interchange Format). Sono invece basati sulla sintassi i linguaggi: *OIL* (*Ontology Inference Layer*), OWL (Web Ontology Language), RDF (Resource Description Framework) e RDF Schema.

Quest'ultimi utilizzano il metalinguaggio XML per codificare la conoscenza e sono stati implementati a supporto del Web Semantico dato che esso necessita di schemi semplici e di basso livello per la trasmissione delle strutture ontologiche.

Oltre ai linguaggi per facilitare l'utilizzo delle ontologie sono nati editor che forniscono strumenti avanzati per la loro gestione, come ad esempio *Protégé*[31].

Protégé è una piattaforma accessibile ad una crescente comunità di utenti che fornisce gli strumenti per la costruzione di modelli di dominio e conoscenza tramite le ontologie. Protégé implementa un ricco set di strutture ed attività che supportano la creazione, la visualizzazione, e la manipolazione delle ontologie nelle varie configurazioni di rappresentazione. Partendo da un'ontologia, Protégé costruisce automaticamente un'interfaccia grafica di acquisizione che permette agli utenti di entrare nel dettaglio e definire specifiche applicazioni. Gli utenti possono personalizzare l'interfaccia configurando le entità grafiche in forme, che sono collegate ad ogni classe dell'ontologia, per l'acquisizione delle istanze. Con *Protégé-2000* gli elaborati possono

[31] http://protege.stanford.edu

essere salvati in molti formati, tra cui la sintassi RDF. Protégé supporta due processi di modellazione delle ontologie:

- ✓ il *Protégé-Frames Editor* consente la creazione di ontologie in concordanza con il protocollo Open Knowledge Based Connectivity (OKBC). In questo modello, un'ontologia consiste di un set di classi organizzato gerarchicamente, le cui proprietà e caratteristiche sono descritte dagli individui;
- ✓ il *Protégé-OWL Editor* consente la costruzione di ontologie per il Web Semantico tramite OWL. Un ontologia OWL può includere descrizioni di classi, proprietà ed esempi. Definita questa, la semantica formale specifica come dedurre le conseguenze logiche e tutto ciò che non è fisicamente presente nell'ontologia. Queste assunzioni possono fondarsi su un solo documento così come su documenti multipli che sono stati combinati usando ben definiti meccanismi di OWL.

2.4 Reti semantiche e *WordNet*

Si è già provveduto a fornire conoscenze di base sulle reti semantiche all'interno del paragrafo 2.1, tuttavia ora verranno esposti alcuni dettagli implementativi in modo maggiormente preciso e si provvederà a delineare le caratteristiche principali di un caso d'uso molto famoso, WordNet.

Le reti semantiche sono formalismi per la rappresentazione della conoscenza che sono stati sviluppati come alternative alla logica. In generale, una rete semantica è un grafo, di solito un grafo diretto, in cui i nodi rappresentano concetti, e gli archi rappresentano relazioni tra i concetti. L'idea di fondo è che tutte le informazioni relative a un dato concetto siano collegate al nodo che lo rappresenta nella base di conoscenza, e siano accessibili a partire da esso, in modo da agevolare le operazioni di reperimento delle informazioni e le inferenze. Esistono molteplici formalismi a rete semantica, spesso anche molto diversi tra loro. Qui di seguito riportiamo un esempio grafico sviluppato mediante un formalismo le cui caratteristiche sono comuni alla maggior parte dei sistemi esistenti.

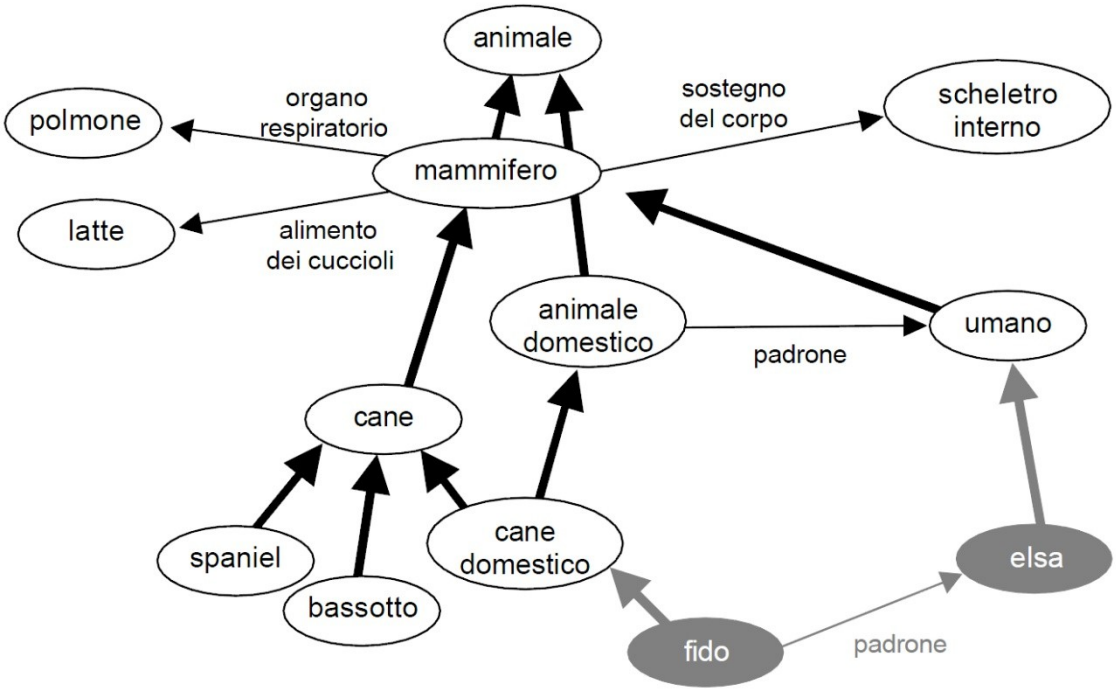

Figura 2.3 – Esempio di rete semantica complessa.

Nella rete semantica di Figura 2.4.1 vengono impiegati due diversi tipi di nodi:

- ✓ nodi che corrispondono a *concetti generici*, e che rappresentano classi di individui (come ad esempio *animale*, *mammifero*, *bassotto*, *polmone*, ecc…), raffigurati graficamente mediante ellissi bianche;
- ✓ nodi che corrispondono a *concetti individuali*, e che rappresentano individui specifici (in questo esempio *fido* e *elsa*), raffigurati graficamente mediante ellissi grigie.

I nodi della rete sono collegati da vari tipi di archi:

- ✓ archi di *sussunzione* (detti anche archi "*isa*", dall'inglese "is a", ossia "è un"), rappresentati graficamente mediante frecce nere spesse; gli archi di sussunzione collegano tra loro concetti generici; un arco di sussunzione da un concetto A a un concetto B indica che A è un tipo particolare, o una sottoclasse di B (ad esempio, *cane* è un tipo particolare di *mammifero*, *animale domestico* è un tipo particolare di *animale*). In questo caso, si dice che A è *sottoconcetto di* , o è *sussunto da* B, e che B è *superconcetto* di A.
- ✓ Archi che rappresentano gli *attributi di un concetto generico*, rappresentati graficamente mediante frecce nere sottili. Gli archi sono etichettati con il nome

dell'attributo. Un attributo di un concetto generico è caratterizzato per mezzo di un altro concetto. Ad esempio, nel caso di *animale domestico*, l'attributo *padrone* è caratterizzato per mezzo del concetto *essere umano* (il *padrone* di un *animale domestico* è un *essere umano*).

- ✓ Archi di *istanziazione*, rappresentati mediante frecce grigie spesse. Collegano un concetto individuale a un concetto generico. Un arco di individuazione da un concetto individuale A a un concetto generico B indica che A è un'istanza di B (*fido* è un'istanza di *cane domestico*, *elsa* è un'istanza di *umano*).
- ✓ Archi che rappresentano gli *attributi di un concetto individuale*, rappresentati graficamente mediante frecce grigie sottili. Sono il corrispettivo a livello di concetti individuali degli archi che rappresentano gli attributi dei concetto generico. Ad esempio, nel caso dell'individuo *fido* l'attributo *padrone* è soddisfatto dall'individuo *elsa*.

Così, la rete di Figura 2.4 descrive, tra le altre cose, i mammiferi come un tipo particolare di animali che hanno come sostegno del corpo uno scheletro interno, come organo respiratorio dei polmoni e i cui cuccioli si alimentano di latte, i cani come un tipo particolare di mammiferi, Fido come un particolare cane domestico che ha come padrone Elsa, e così via.

Uno dei meccanismi di inferenza tipici delle reti semantiche consiste nell'*ereditarietà*. L'idea di fondo è che le proprietà espresse per un concetto della rete vengano ereditate da tutti i suoi sottoconcetti. Ad esempio, nella rete della figura il concetto *cane*, in quanto sottoconcetto di *mammifero*, eredita tutti gli attributi che sono stati specificati per *mammifero*.

E' ammessa l'*eredità multipla*, ossia è ammesso che un concetto erediti da più superconcetti diversi. Così, nella figura, *cane domestico* eredita sia gli attributi di *cane*, sia quelli di *animale domestico*.

In generale, per ogni concetto, agli attributi ereditati dai superconcetti se ne possono aggiungere di nuovi. Inoltre, gli attributi ereditati possono essere ulteriormente specificati localmente.

Sebbene questo tipo di reti semantiche sia traducibile in logica dei predicati, ciò non comporta che in generale logica dei predicati e reti semantiche siano tra loro equivalenti. Vi sono innanzitutto cose che si possono esprimere in logica, ma che non si

possono esprimere con questo tipo di reti. Una differenza evidente consiste nel fatto che le reti semantiche non consentono di rappresentare direttamente relazioni fra più di due argomenti.

D'altra parte le reti semantiche vengono spesso usate per rappresentare informazioni ed effettuare inferenze che tradizionalmente non si possono esprimere in logica. Un esempio è la rappresentazione di eccezioni all'ereditarietà. Spesso nelle reti semantiche gli attributi di un concetto vengono usati per rappresentare caratteristiche che non valgono per tutte le sue istanze, ma solo le sue istanze tipiche. La struttura a grafo delle reti semantiche consente di gestire in modo agevole vari tipi di eredità con eccezioni.

Ora che si è provveduto a una descrizione maggiormente dettagliata della teoria sulle reti semantiche ne verrà descritto un esempio noto a livello mondiale, *WordNet*.

Le reti semantiche risultano essere particolarmente adatte per la rappresentazione delle relazioni fra il lessico del linguaggio naturale e vengono impiegate in diverse applicazioni NLP[32]. WordNet è una risorsa linguistica sviluppata da George Miller a partire dal 1985 presso l'Università di Princeton, che organizza, definisce, descrive i concetti rilevanti della lingua inglese. Il design di WordNet è ispirato agli studi di psicolinguistica per quanto riguarda la memoria lessicale umana.

L'idea iniziale era quella di fornire un aiuto per la ricerca concettuale all'interno di dizionari, per superare la semplice ricerca in ordine alfabetico, tuttavia dati gli sviluppi notevoli del progetto furono proposti obbiettivi maggiormente ambiziosi e WordNet ne è il risultato.

La differenza più evidente fra WordNet e un dizionario tradizionale è che nel primo il lessico viene suddiviso in cinque categorie principali: nomi, verbi, aggettivi, avverbi e in ultimo parole che hanno essenzialmente una funzione grammaticale e non semantica, come ad esempio articoli, pronomi, preposizioni, congiunzioni, ecc…

Il prezzo pagato per la scelta di questa categorizzazione del lessico è stata la presenza di un certo grado di ridondanza che solitamente è assente nei dizionari tradizionali. Infatti parole sintatticamente uguali compaiono all'interno di diverse categorie per il molteplice ruolo che possono ricoprire, come ad esempio "stato" che può essere un verbo o un sostantivo.

[32] Natural Language Processing

Per ciascuna categoria inoltre è stato adottato un differente schema di rappresentazione, i sostantivi sono stati organizzati gerarchicamente, i verbi attraverso un certo numero di implicazioni logiche legate alle relazioni fra essi, gli aggettivi e gli avverbi sono invece stati organizzati in uno spazio N-dimensionale. Ciascuna di queste strutture lessicali deriva dagli studi in campo psicolinguistico e soddisfa nel modo migliore presumibile la complessità con cui la mente umana gestisce e organizza le informazioni lessicali. Un unico schema di rappresentazione per tutte le categorie non avrebbe potuto condurre a risultati soddisfacenti.

La caratteristica più ambiziosa di WordNet è quella di cercare di organizzare le informazioni lessicali in termini di significato delle parole piuttosto che in base alla loro forma. Da questo punto di vista WordNet assomiglia più a un *thesaurus*[33] che a un dizionario.

La semantica lessicale inizia con il presupposto che una parola è una convenzionale associazione fra un concetto e un enunciato che gioca un ruolo sintattico. Dal momento che il termine "parola" viene utilizzato sia per indicare l'enunciato sintattico che il concetto associato è necessario, per ridurre l'ambiguità, utilizzare i termini "*forma della parola*" e "*significato della parola*". Come Geroge A. Miller afferma, una volta determinata questa differenza, "*il punto di partenza della semantica lessicale può essere ritenuta la mappatura fra forme e significati*".

In Figura 2.4 viene mostrata la nozione concreta di matrice lessicale. Le forme delle parole vengono elencate sulle righe mentre il significato viene posto sulla prima colonna. Una *entry* nella matrice implica che quella forma può essere utilizzata, all'interno di un contesto appropriato, per esprimere il significato che rappresenta quella riga.

[33] Secondo la definizione ISO il thesaurus è "un vocabolario di un linguaggio di indicizzazione controllato in maniera formalizzata in modo che le relazioni a priori tra i concetti sono rese esplicite".

Word Meanings	Word Forms				
	F_1	F_2	F_3	. . .	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2		$E_{2,2}$			
M_3			$E_{3,3}$		
⋮				. . .	
M_m					$E_{m,n}$

Figura 2.4 – Concetto di matrice lessicale

Se ci sono più *entries* all'interno della stessa colonna allora quella forma è soggetta a polisemia, ovvero può esprimere più significati, se invece sono presenti due forme sulla stessa riga allora le due forme sono sinonimi relativamente a un contesto. Polisemia e sinonimia possono essere visti come aspetti complementari di questa mappatura. Abitualmente gli psicolinguisti preferiscono rappresentare questi concetti attraverso uno schema costituito da nodi e archi. Con questa notazione la matrice lessicale potrebbe essere rappresentata attraverso due nodi e archi che collegano i nodi in entrambe le direzioni. Un nodo contiene il significato delle parole, mentre l'altro contiene la forma delle parole. In questo schema gli archi potrebbero collegare i significati a ciascuna forma che li rappresenta, o al contrario potrebbero collegare le forme a tutti i significati che esse possono assumere.

Come vengono quindi rappresentati i significati all'interno di WordNet?

Per rispondere a questa domanda si deve tenere conto del fatto che non è ancora stata considerata la rappresentazione delle relazioni che esistono fra i vari significati. Data questa nuova componente è necessario prevedere una forma compatta per rappresentare contemporaneamente sia la forma che il significato. Inizialmente si riteneva necessario che ciascun significato dovesse essere descritto in modo accurato, tuttavia considerando il numero enorme di concetti da descrivere ciò avrebbe causato problemi di dimensioni della struttura WordNet, si optò in seguito per la scelta che riteneva sufficiente un insieme di sinonimi per definire in modo sufficientemente preciso un concetto. In altre parole, prendendo in considerazione la Figura 2.4, il significato M_1 può essere

rappresentato elencando semplicemente le forme che possono essere usate per esprimerlo: $\{F_1, F_2, \ldots\}$.

Questo insieme di sinonimi va a costituire una struttura denominata *Synset* che rappresenta un concetto definito. Un *Synset* non spiega il senso del concetto, semplicemente identifica l'esistenza del concetto che rappresenta.

Tuttavia alcune volte è possibile incontrare concetti che sono descritti a livello culturale da un'unica forma, in questo caso il problema legato alla polisemia può essere risolto includendo all'interno del Synset una glossa.

WordNet è organizzato in base alle relazioni semantiche. Dal momento che una relazione semantica è una relazione fra significati e dato che i significati sono rappresentati dai Synset, viene naturale pensare le relazioni semantiche come puntatori fra i Synset.

Inoltre se esiste una relazione R fra i significati $\{x, x', \ldots\}$ e $\{y, y', \ldots\}$ allora R sarà usata anche per stabilire una relazione fra le singole forme che appartengono ai due Synset.

Ora che è stata delineata la struttura generica di WordNet ne viene completata di seguito la descrizione elencando i tipi di relazione che sussistono fra i Synset. Le relazioni semantiche variano in funzione della categoria a cui ciascun Synset appartiene, ovvero a seconda che un concetto sia un sostantivo, un verbo, un aggettivo o un avverbio.

Per i sostantivi le relazioni esistenti sono:

- ✓ *iperonimia*: A è un iperonimo di B se il significato di B è incluso in A;
- ✓ *iponimia*: A è un iponimo di B se il significato di A è incluso in B;
- ✓ *coordinazione*: A è un termine coordinato di B se A e B hanno un iperonimo in comune;
- ✓ *olonimia*: A è un olonimo di B se B è parte A;
- ✓ *meronimia*: A è un meronimo di B se A è parte B.

Per i verbi:

- ✓ *iperonimia* : il verbo A è un iperonimo del verbo B se l'attività B è inclusa in A (come viaggio rispetto a movimento);
- ✓ *troponimia* : il verbo A è un troponimo del verbo B se nel fare l'attività A si fa anche la B (come mormorare rispetto a parlare);

- ✓ *implicazione* : il verbo A è un'implicazione del verbo B se nel fare B uno deve per forza fare A (come russare rispetto a dormire);
- ✓ *coordinazione*: A è un termine coordinato di B se B e A hanno un iperonimo in comune.

Gli aggettivi sono classificati come:

- ✓ nomi relativi;
- ✓ simile a;
- ✓ participi dei verbi.

Gli avverbi seguono la classificazione dell'aggettivo da cui derivano.
Oltre a quelle elencate vi sono inoltre la relazione di sinonimia, implicitamente esistente fra tutte le parole contenute in un Synset, e la relazione di *antonimia* ovvero la relazione che identifica i contrari di un concetto.
WordNet memorizza il conto di tutti i significati delle parole polisemiche, ovvero il numero di Synset che contengono tale parola. Se una parola è inclusa all'interno di più Synset è molto probabile che alcuni significati siano molto più probabili di altri. WordNet quantifica questa caratteristica con un punteggio di frequenza calcolato sulla base di numerosi testi in cui le parole sono state marcate semanticamente.

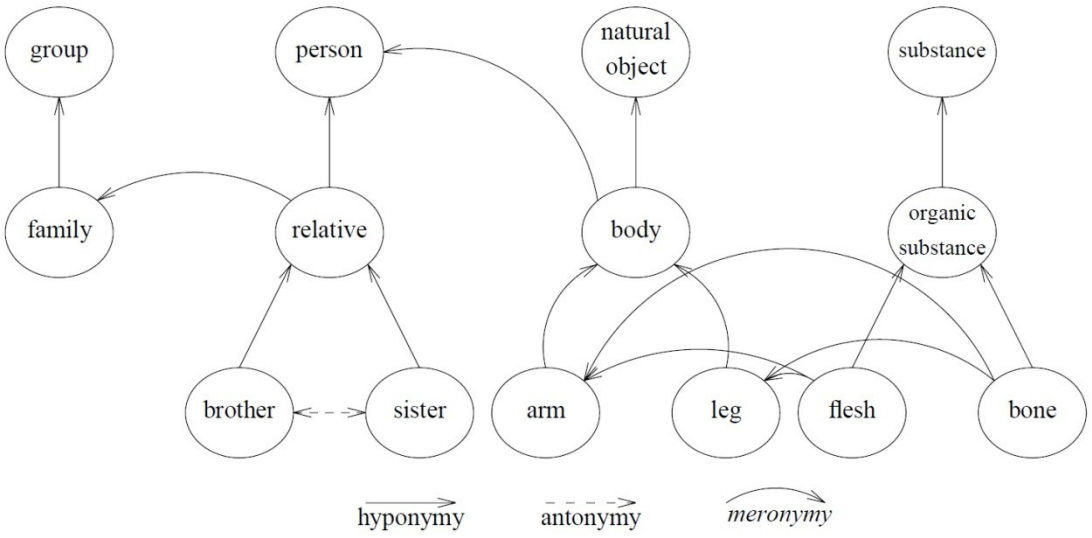

Figura 2.5 – Esempio di relazioni fra sostantivi in *WordNet*.

In Figura 2.5 viene mostrato un esempio delle relazioni che vengono instaurate fra alcuni sostantivi contenuti in *WordNet*.

Come specificato prima a seconda della categoria dei termini vengono utilizzati schemi di rappresentazione differenti. Per mostrare tale differenza in Figura 2.6 viene visualizzato un esempio delle relazioni che vengono instaurate fra alcuni aggettivi contenuti in WordNet.

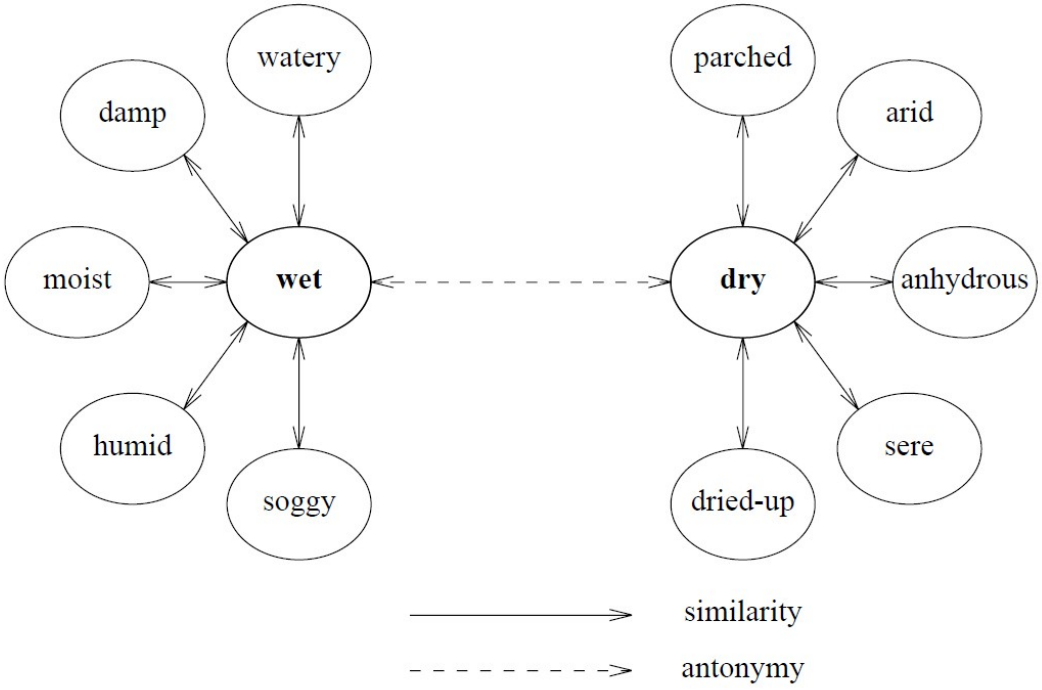

Figura 2.6 - Esempio di relazioni fra aggettivi in *WordNet*.

Come si può notare gli aggettivi vengono organizzati utilizzando due poli principali di senso opposto. Ciascuno di questi poli possiede un certo numero di satelliti costituiti da aggettivi che hanno un significato simile al polo a cui sono collegati.

WordNet è stato utilizzato per una serie di scopi differenti all'interno dei sistemi informativi come ad esempio: il WSD (Word Sense Disambiguation), l'*information retrieval*, la classificazione automatica del testo, la sintesi automatica di testi o addirittura la generazione automatica di cruciverba. Di seguito vengono riportate alcune statistiche del progetto WordNet tratte dal portale *http://wordnet.princeton.ed*.

Categoria	Stringhe univoche	Synset	Relazioni
Sostantivo	117.798	82.115	146.312
Verbo	11.529	13.767	25.047
Aggettivo	21.479	18.156	30.002
Avverbio	4.481	3.621	5.580
Totale	155.287	117.659	206.941

Tabella 2.3 – Statistiche di *WordNet*.

In risposta al progetto inglese l'Istituto di Linguistica Computazionale "Antonio Zampolli" sta sviluppando *ItalWordNet* (*IWN*), un database semantico-lessicale sviluppato nell'ambito di due progetti di ricerca distinti: *EuroWordNet* (*EWN*) e *Sistema Integrato per il Trattamento Automatico del Linguaggio* (*SI-TAL*), un progetto nazionale dedicato alla creazione di ampie risorse linguistiche e di strumenti software per l'elaborazione dell'italiano scritto e parlato.
IWN è strutturato nello stesso modo del Princeton WordNet, vale a dire attorno alla nozione di Synset o gruppo di sensi sinonimi tra loro.
Per maggiori informazioni riguardo al progetto italiano si invita a visitare il link *http://www.ilc.cnr.it/viewpage.php/sez=ricerca/id=834/vers=ita*.

Ù

3 NLP E TECNOLOGIE LINGUISTICHE

Il termine Natural Language Processing (NLP) è normalmente utilizzato per descrivere le funzionalità di un software o le componenti hardware all'interno di un computer che permettono di analizzare o sintetizzare il linguaggio scritto o parlato. L'epiteto "Natural" ha l'obiettivo di distinguere il linguaggio umano scritto o parlato da linguaggi notevolmente più formali, come ad esempio le notazioni matematiche o logiche, oppure i linguaggi di programmazione come Java o C++. In senso più stretto il concetto di *Natural Language Understanding* (*NLU*) viene associato ad obiettivi molto più ambiziosi, ovvero la capacità di un computer di interpretare e comprendere il linguaggio naturale con la stessa abilità di un essere umano.

E' ovvio che le macchine possono essere istruite per la comprensione dei linguaggi di programmazione, nel senso che vi è la possibilità di realizzare un interprete che possa permettere ad esempio ad un applet implementato in Java di essere eseguito correttamente all'interno di un browser. E' anche possibile programmare un computer per la risoluzione di numerosi e notevoli problemi matematici e logici, come la dimostrazione di teoremi, tuttavia l'analisi automatica del linguaggio naturale scritto o parlato rimane tutt'oggi una sfida aperta, che presenta notevoli problemi di complessità relativamente connessi alle numerose ambiguità che si presentano nel linguaggio umano.

Negli anni precedenti al processo di globalizzazione mondiale e alla grande diffusione dell'utilizzo del Web, che ha provocato un conseguente incremento di informazioni disponibili al costo di una connessione Internet, la risoluzione dei problemi di natura linguistica non attraevano il minimo interesse commerciale. Attualmente, invece, la situazione è letteralmente cambiata, infatti la necessità di informazioni rappresenta oggi il mercato con il tasso di crescita più alto.

Sempre più spesso ad esempio le informazioni di carattere finanziario sono disponibili online in un formato di testo piano, invece che in un formato tabellare simile allo schema di un database. La questione non è più la mancanza di informazioni, ma l'imbarazzo della scelta e la mancanza di strumenti per l'organizzazione delle informazioni. La maggior parte delle informazioni utili, inoltre, sono esposte proprio attraverso il linguaggio naturale e non tramite immagini, grafici, file audio e video.

Spesso le informazioni che risiedono all'interno dei database relazionali sono state estratte da documenti elettronici come ad esempio *memo*, fogli di calcolo e tabelle, e quasi sempre tale lavoro è stato effettuato a mano con un notevole utilizzo di risorse e di tempo.

Ciò che è stato appena descritto rappresenta la motivazione per cui il trattamento del linguaggio naturale sta assumendo un ruolo sempre più importante nel dominio delle tecnologie informatiche e giustifica gli sforzi messi in campo a livello internazionale, sia da parte di enti pubblici che da parte di aziende private, per giungere alla soluzione dei principali problemi che impediscono alle applicazioni NLP di rappresentare uno strumento affidabile con cui sostituire l'attuale lavoro svolto a mano da operatori specializzati.

3.1 Gli approcci storici al NLP

Nel settore della linguistica computazionale esistono essenzialmente due tipologie di approcci che sono stati adottati per cercare di modellare i problemi inerenti alla comprensione automatica dei testi piani, quello *razionalista* e quello *empirista*.

Fra il 1960 e il 1985 i settori della linguistica, della psicologia, dell'intelligenza artificiale e del trattamento del linguaggio naturale sono stati completamente dominati da un approccio di tipo razionalista. Tale punto di vista è caratterizzato dalla convinzione che una parte significante della conoscenza presente all'interno della mente umana non provenga dai sensi, ma sia presente a priori in essa per un fenomeno conosciuto con il termine di ereditarietà genetica. Tale approccio ha assunto una posizione dominate fra gli studiosi della linguistica grazie alla diffusa accettazione di alcune argomentazioni proposte da Noam Chomsky[34] a proposito della facoltà innata che gli esseri umani avrebbero per quanto riguarda, ad esempio, l'apprendimento del linguaggio.

All'interno del campo dell'intelligenza artificiale la posizione razionalista può essere interpretata come la convinzione che il tentativo di creare sistemi automatici intelligenti debba prevedere di inserire a priori all'interno di essi numerosi meccanismi di

[34] Avram Noam Chomsky (Filadelfia, 7 dicembre 1928) è uno scienziato, filosofo e teorico della comunicazione statunitense. Professore emerito di linguistica al Massachusetts Institute of Technology è riconosciuto come il fondatore della grammatica generativo-trasformazionale, spesso indicata come il più rilevante contributo alla linguistica teorica del XX secolo.

ragionamento e schemi cognitivi in modo da replicare ciò che, secondo i razionalisti, avviene nel cervello umano.

Noam Chomsky ha sempre avuto una propensione a sostenere questa tesi a causa di ciò che egli ha definito come "problema della povertà degli stimoli"[35]. Egli suggerisce che è molto difficile capire come un bambino possa apprendere qualcosa di così complesso come il linguaggio naturale attraverso il numero limitato di stimoli che esso può ascoltare nei primi anni di vita.

Studi recenti suggeriscono che alcune porzioni del cervello umano (che coinvolgono in modo cruciale l'Area di Broca, una porzione del giro frontale inferiore sinistro), siano selettivamente attivate nell'apprendimento dei linguaggi che presentano i requisiti della *Grammatica Universale*[36], mentre non si attiva quando si manipola artificialmente la grammatica delle lingue. Queste evidenze hanno messo in luce il nesso imprescindibile tra biologia e norme del linguaggio.

Anche l'approccio empirista allo studio del linguaggio ammette nelle sue assunzioni iniziali che nella mente umana vi siano alcune abilità cognitive innate. La differenza tra i due approcci non è pertanto assoluta, come si potrebbe pensare, poiché entrambe condividono la convinzione dell'esistenza di una *forma mentis* a priori presente nella mente umana, una struttura che induce le informazioni ad essere organizzate e generalizzate in un certo modo piuttosto che in un altro. Inoltre entrambi gli approcci concordano sul fatto che non sia possibile che un processo di apprendimento, come quello che avviene nel cervello umano, possa avere inizio a partire da una *tabula rasa*.

Tuttavia la corrente empiristica rifiuta di accettare che la mente sia inizializzata con una serie di principi e procedure dettagliate per l'apprendimento delle varie componenti del linguaggio naturale e di altri domini cognitivi. Vi è piuttosto l'accettazione diffusa che il cervello di un bambino possieda inizialmente la capacità di compiere operazioni generali e semplici per l'associazione delle informazioni, il confronto dei concetti e la generalizzazione dei termini, e che tali regole possano essere applicate ai numerosi input sensoriali per l'apprendimento della complessa struttura del linguaggio naturale.

La filosofia empiristica fu dominante nel settore linguistico e dell'intelligenza artificiale fra il 1920 e il 1960 e sta vivendo attualmente un'importante rinascita. Un approccio

[35] Noam Chomsky, 1986, *Knowledge of Language: Its Nature Origin, and Use,* Praeger, New York.

[36] La Grammatica Universale è una teoria linguistica la quale postula che i principi della grammatica siano condivisi da tutte le lingue, e siano innati per tutti gli esseri umani.

empirista al NLP suggerisce che si possano imparare le numerose e complesse strutture del linguaggio specificando un appropriato modello generale e introducendo successivamente i valori derivanti da processi di analisi statistica, *pattern recognition* e *machine learning* applicati ad una grande quantità di casi linguistici.

Proprio l'analisi statistica sta assumendo un ruolo sempre più importante nello studio del linguaggio naturale e un argomento a favore del suo utilizzo, per la comprensione scientifica del linguaggio, deriva dal fatto che, di per se la cognizione umana è intrinsecamente probabilistica e pertanto lo deve essere anche la lingua essendo parte integrante della cognizione.

A tale proposito si sottolinea la grande utilità che ha avuto in questo progetto di tesi il testo teorico *Foundations of Statistical Natural Language Processing*, realizzato da Chris Manning e Hinrich Schütze e pubblicato presso il *MIT Press*.

Questo libro è il primo in ordine di pubblicazione che tratta in maniera esaustiva i metodi statistici e gli algoritmi teorici per l'approccio al NLP. Provvede a fornire una copertura quasi totale dei fondamenti matematici e linguistici permettendo a chi affronta per la prima volta l'argomento di comprendere gli strumenti necessari per la realizzazione di applicativi per l'analisi automatica del linguaggio come ad esempio *word sense disambiguator*, *probabilistic parser* ed altre applicazioni.

3.2 Le problematiche legate al NLP

Verranno ora descritte alcune problematiche relative al trattamento automatico del linguaggio scritto. Si evidenzia infatti che oltre ai testi scritti l'NLP ambisce anche allo studio e alla comprensione del linguaggio parlato anche se questa forma presenta maggiori problemi rispetto al trattamento del linguaggio scritto. Tali problemi derivano essenzialmente dalla forma del messaggio e dalla sua corretta interpretazione da parte di un sistema informatico. Non banale ad esempio riuscire a segmentare correttamente in modo automatico un messaggio audio dato che i risultati possono dipendere da molti fattori quali la chiarezza dell'oratore, la sua pronuncia, la velocità del parlato e l'utilizzo in esso di pause o legature. Da questo punto di vista il linguaggio scritto offre sicuramente un punto di partenza più semplice per addentrarsi immediatamente negli aspetti specifici dell'NLP.

Il problema principale, quando si vuole avviare l'analisi computazionale del testo, è stabilire dei criteri di identificazione per quella che è la sua unità di base: la *parola*. Questo processo, più comunemente chiamato *segmentazione* o *tokenizzazione*, ovvero l'operazione mediante la quale si suddivide il testo in *token*, è relativamente semplice per lingue che, similmente all'italiano, adoperano gli spazi per delimitare le parole. Risulta invece molto complessa per lingue a sistema ortografico continuo, in questo caso l'operazione richiede algoritmi estremamente complicati. Se ci limitiamo al primo caso, il *token* è definibile semplicemente come una qualunque sequenza di caratteri delimitata dagli spazi, tuttavia, tale definizione lascia spazio a numerose eccezioni. Pensiamo ad esempio ai segni di punteggiatura, che compaiono attaccati alle parole: l'apostrofo compare di norma in mezzo a due parole diverse che, in virtù della definizione, verrebbero erroneamente identificate come una parola unica.

L'ambiguità della punteggiatura costituisce un problema anche quando dobbiamo identificare l'unità linguistica superiore alla parola, ovvero la *frase*. Potremmo definire le frasi come sequenze di parole separate da punto e spazio e comincianti con una maiuscola, ma ci sono anche abbreviazioni come "Mr. Johnson" che, secondo questa regola, verrebbero scisse in frasi distinte.

In seconda battuta dopo aver suddiviso il testo in unità base insorge la necessità di definire il ruolo di ogni token. Questo processo viene definito Part-of-speech tagging, e dal punto di vista pratico rappresenta la marcatura, ad esempio tramite XML, di ciascun

token, definendo inequivocabilmente il ruolo grammaticale o altre caratteristiche come la relazione con le parole adiacenti. È un'operazione simile a quella che ognuno di noi ha dovuto imparare durante i primi anni di scuola per effettuare esercizi di analisi grammaticale o logica attraverso i quali si analizza la struttura linguistica di un testo e si attribuisce a ogni parola o parte del discorso un ruolo ben definito.
Tuttavia l'uomo possiede una capacità fondamentale per effettuare questa operazione, la comprensione del *contesto*, che permette di interpretare immediatamente il giusto senso di un termine. Le macchine si trovano invece ad affrontare uno degli aspetti più ardui dell'NLP, la *disambiguazione*, argomento che verrà trattato in maggior dettaglio nel paragrafo 3.2.2.
Altro dettaglio non banale è la necessita da parte delle macchine di avere un sistema di rappresentazione del linguaggio che ne possa permettere un trattamento agevolato e che ne possa fornire una completa definizione a diversi livelli di astrazione. In risposta a questo problema nel paragrafo 3.2.1 viene esposta la soluzione proposta da Noam Chomsky.

3.2.1 Correttezza sintattica e correttezza semantica

Noam Chomsky del suo trattato *Syntactic Structures*[37] pose molta attenzione sulla distinzione fra quelle frasi che sono sintatticamente anomale e prive di senso:

> *"Furiously sleep ideas green colorless." (trad.)"Furiosamente dormo le idee verdi senza colore"*

e frasi che sono sempre prive di senso ma sono grammaticalmente ben formate come:

> *"Colorless green ideas sleep furiously." (trad.)"Le idee verdi senza colore dormono furiosamente"*

Chomsky propose questo ultimo esempio di frase grammaticalmente corretta ma priva di significato per dimostrare che un modello di analisi linguistica basato esclusivamente su fondamenti statistici risulta essenzialmente inadeguato se si fonda solo sulla

[37] Noam Chomsky, *Syntactic Structures*. The Hague: Mouton, 1957; Berlin and New York: 1985.

valutazione della correttezza grammaticale e che per raggiungere buoni risultati nel trattamento automatico del linguaggio occorrono modelli maggiormente strutturati.
Nonostante questa osservazione è innegabile affermare che per un processo automatico di analisi linguistica può risultare estremamente utile valutare se una frase è grammaticalmente corretta oppure no, anche se naturalmente tale approccio non sarebbe adeguato per l'analisi di testi poetici e di testi contenenti errori sintattici che tuttavia non compromettono il significato del contenuto.
Il fatto che si possano suddividere le regole di un linguaggio in due differenti categorie ha da sempre influenzato le metodologie di analisi, in particolare un'assunzione generale che viene effettuata dai sistemi NLP è quella di analizzare in prima battuta la *struttura sintattica* delle frasi, senza alcun riferimento al significato delle parole, e in seconda battuta viene analizzata la *struttura semantica*.
Per analizzare la struttura sintattica dei testi piani Chomsky ha introdotto un modello denominato *Phrase structure rules*. Questo modello, basato su una struttura ad albero, viene utilizzato per suddividere un frase esposta in linguaggio naturale nei suoi elementi costitutivi, come mostrato in Figura 3.1

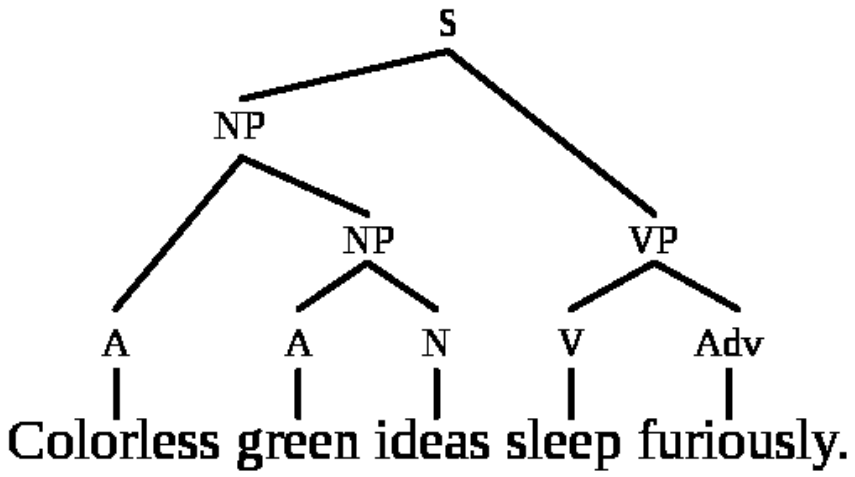

Figura 3.1 – Esempio di *Phrase structure rules*

Gli elementi costitutivi possono essere di due tipi, le *phrasal categories* e le *lexical categories*. Le prime includono *noun phrase*, *verb phrase*, and *prepositional phrase,* mentre le seconde includono *noun*, *verb*, *adjective*, *adverb*, e tutti gli altri ruoli grammaticali che possono assumere le parole all'interno di una frase. E' stata mantenuta

la terminologia in lingua inglese per mantenere una maggiore coerenza con il modello originale e per il fatto che i concetti relativi ad esso sono sufficientemente intuitivi.
Le regole di questo modello vengono esplicitate nel seguente modo:

- ✓ *S → NP VP*
- ✓ *NP → A NP*
- ✓ *NP → A N*
- ✓ *VP → V Adv.*

NP → A NP significa ad esempio la *noun phrase NP* è composta da un *adjective* A e un'ulteriore *noun phrase NP*. Come si può notare questo modello usa costrutti ricorsivi infatti come nell'esempio appena evidenziato un elemento può essere espresso in termini di sé stesso.
Un problema che affligge questo modo di rappresentare le strutture sintattiche è la possibilità frequente di poter rappresentare una stessa frase con più schemi, tuttavia ciò non impedisce di valutare se la sequenza delle parole compone una frase è ben formata dal punto di vista grammaticale.
Questo modello può sembrare eccessivamente semplicistico ma nonostante ciò è stato quello più adottato all'interno dei sistemi NLP, anche se con notevoli modifiche nel corso del tempo.

3.2.2 L'ambiguità nel linguaggio naturale

Anche se a volte le persone sono ritenute essere ambigue per il modo in cui utilizzano la lingua, l'ambiguità è una caratteristica intrinseca delle espressioni linguistiche. In generale una parola o una frase sono ritenute ambigue se sono suscettibili a più di una interpretazione, come ad esempio le seguenti frasi:

- ✓ *"Il **calcio** ormai riempie le domeniche degli italiani."*
- ✓ *"Il bandito ha colpito il cassiere con il **calcio** della pistola."*
- ✓ *"Il simbolo del **calcio** è Ca."*

Come è ben riscontrabile nella prima frase il termine calcio viene utilizzato per definire l'attività sportiva calcistica, nella seconda per indicare la parte posteriore di un pistola, mentre nell'ultima frase viene utilizzato per indicare l'elemento chimico. Per un qualsiasi lettore risulta sicuramente facile poter interpretare nel modo corretto il reale

significato a cui la parola *calcio* fa riferimento in ciascuna delle tre frasi, tuttavia questo risulta un dei compiti più ardui per il trattamento automatico del linguaggio naturale.

Esistono essenzialmente due tipologie di ambiguità, quella lessicale, già presentata negli esempi precedenti e quella sintattica o strutturale. Per quanto riguarda l'ambiguità lessicale oltre ai casi di omonimia che fanno riferimento allo stesso tipo grammaticale è necessario considerare i casi di omonimia che risultano essere espressione di diversi tipi grammaticali. Un esempio di quanto appena detto può essere riscontrato nelle seguenti frasi:

- ✓ *"Il vento **porta** via le nuvole."*
- ✓ *"Ho dimenticato di chiudere la **porta**."*

Nel primo caso il termine porta risulta essere le seconda persona singolare del verbo portare coniugato al presente, mentre nella seconda frase indica la classica porta di casa ed è quindi inteso come sostantivo.

L'ambiguità strutturale si ha quando una frase può avere multiple interpretazioni sintattiche, da cui derivano significati diversi. Ad esempio:

"Chiara ha visto Luca in giardino con il cannocchiale."

In questo caso linguistico un'applicazione per il trattamento automatico del linguaggio potrebbe incontrare non poche difficoltà per comprendere se Chiara ha visto Luca attraverso un cannocchiale perché il giardino è lontano dal suo punto di vista, oppure se semplicemente Chiara ha visto ad occhi nudi Luca che reggeva un cannocchiale in mano mentre era in giardino. Un essere umano può agevolmente sciogliere le ambiguità di tipo lessicale in base al contesto della frase in cui tale ambiguità è inserita, mentre le ambiguità di tipo strutturale non possono essere disambiguate neanche dal lettore umano se non vi sono ulteriori riferimenti nei testo che stiamo leggendo.

I due principali approcci storici, utilizzati nel settore del trattamento automatico del linguaggio, per risolvere queste tipologie di ambiguità, ovvero per svolgere quella fase dell'analisi che viene definita Word Sense Disambiguation (WSD), si basano essenzialmente sul modello definito con il termine Knowledge-based e sul modello statistico. Nel primo caso gli sviluppatori di sistemi NLP devono cablare all'interno dell'applicazione una enorme quantità di conoscenza che riguarda il mondo reale e devono essere sviluppate inoltre le procedure e gli algoritmi per utilizzare tale conoscenza nella giusta determinazione del significato di una parola all'interno di un

testo. Un esempio di soluzione riconducibile a questo caso è quello in cui viene utilizzata la conoscenza contenuta nella rete semantica linguistica WordNet.

Per quanto riguarda l'approccio statistico alla disambiguazione linguistica invece si necessita di una grande quantità di testi in cui siano annotate tutte le soluzioni delle ambiguità presenti, in modo che un sistema informatico possa apprendere automaticamente i casi più frequenti e dedurre semplici regole di disambiguazione di cui di seguito ne viene dato un esempio:

- ✓ *"Quel musicista è veramente bravo a suonare il **basso** elettrico."*
- ✓ *"Temo di aver preso un voto molto **basso** in matematica."*

Nella prima frase è ovvio che con il termine basso ci si sta riferendo allo strumento musicale e una regola che potrebbe aiutare un sistema NLP a comprendere il giusto significato di basso potrebbe essere: "Se il termine basso si trova vicino all'aggettivo elettrico o al verbo suonare allora interpretalo come strumento musicale". Nel secondo caso invece potrebbe essere dedotta la regola: "Se il termine basso si trova vicino al sostantivo voto, all'avverbio molto e al sostantivo matematica allora interpretalo come aggettivo qualificativo."

Con il metodo appena mostrato, ovvero con l'approccio statistico alla disambiguazione, si sono ottenuti buoni risultati pratici nonostante la sua efficacia teorica sia in realtà molto inferiore rispetto all'approccio basato sulla conoscenza.

Questo approccio solitamente impone la definizione della dimensione della finestra di parole che vengono utilizzate per la disambiguazione e successivamente si utilizzano strumenti statistici come i *classificatori Bayesiani* e gli *alberi di decisione*.

3.3 Gli obiettivi del NLP

L'applicazione primaria del trattamento automatico dei testi presenti sul Web risulta essere ancora il *document retrieval*, ovvero la ricerca di documenti che sono rilevanti in base alle richieste degli utenti. Vi è la possibilità di effettuare document retrieval anche senza effettuare rilevanti operazioni di NLP, come numerosi motori di ricerca continuano a fare, tuttavia il trend che si è delineato dagli '90 fino ai giorni nostri è quello di un processo di indicizzazione, identificazione e presentazione dei documenti sempre più complesso e strettamente dipende dall'analisi linguistica dei contenuti.

Un obiettivo correlato ma non identico al document retrieval è il *document routing*; questo processo dovrebbe permettere in futuro, soprattutto all'interno di reti informative aziendali, di indirizzare automaticamente i documenti ai corretti profili aziendali per ridurre il tempo di smistamento quando le competenze necessarie per la risoluzione di un problema non sono note a priori.

Un'ulteriore obbiettivo perseguito dalle applicazioni NLP è quello della *classificazione*. Con classificazione si intende l'assegnamento di ciascun documento ad una particolare classe in base ai suoi contenuti. Nel caso più generale un documento potrebbe essere assegnato a più di una classe e le classi potrebbero essere organizzate in un struttura gerarchica complessa. Va precisato per quanto riguarda la classificazione dei documenti che il risultato di tale operazione dipende estremamente dalla tipologia dei documenti analizzati e dalla loro dimensione. Se ipotizziamo infatti di processare tramite un'applicazione NLP un testo molto lungo che tratta numerosi argomenti estremamente eterogenei fra loro è facile intuire la difficoltà che si avrà nel classificare tale documento. Mentre nel caso di analisi di testi relativamente brevi che trattano uno o pochi argomenti risulta evidente che il processo di classificazione risulterà estremamente più semplice e con buona probabilità corretto.

Sempre più frequentemente l'attenzione si sta spostando dalla ricerca di un documento alla ricerca di una particolare informazione all'interno di un preciso documento e di un insieme di documenti. Per esempio, dato un insieme di articoli giornalistici che trattano l'acquisizione di aziende, si potrebbe pensare di voler ottenere una sintesi di tali articoli che includa solo chi ha acquisito chi, escludendo tutte le informazioni di contorno che potrebbero non essere rilevanti per un operatore del settore.

Tale operazione è solitamente chiamata *Information Extraction* e fornisce un metodo automatico per la generazione di metadati, processabili da ulteriori software, che racchiudono le informazioni più rilevanti contenute all'interno di un testo piano.

L'*Automatic Summarization* rappresenta la creazione di versioni riassunte di un testo in modo tale che vengano mantenuti solo i punti più importanti del testo originale. Tale operazione differisce dall' Information Extraction per il fatto che non necessariamente le informazioni recuperate siano formattate in modo che risultino processabili da ulteriori applicazioni.

L'*Automatic Translation*, di cui esistono già numerosi esempi online, sfrutta il trattamento del linguaggio naturale per effettuare la traduzione da una lingua all'altra non sostituendo semplicemente un parola con la relativa traduzione, ma verificando se vi è la presenza di anomalie linguistiche come ad esempio idiomi o proverbi che non avrebbero alcun senso se tradotti parola per parola.
L'elenco degli obbiettivi perseguibili con il trattamento del linguaggio naturale sarebbe maggiormente espandibile, tuttavia non si andrà ulteriormente in dettaglio dal momento che molti di essi rappresentano generalizzazioni o specializzazioni di quelli citati

3.4 Le tecnologie linguistiche di Expert System

Expert System è un'azienda con esperienza decennale che opera nel settore informatico principalmente a livello di Gestione della conoscenza e Natural Language Processing.
Nasce nel 1989 e dopo l'iniziale sviluppo di applicazioni software tradizionali, l'interesse di Expert System si sposta sul mondo dell'analisi semantica, in cui l'azienda ottiene eccezionali risultati riuscendo a conquistare la fiducia di Microsoft, per la quale ha prodotto i correttori grammaticali e ortografici per la versione italiana di Microsoft Office.
Lo sviluppo della tecnologia per l'analisi linguistica *COGITO*®, di cui Expert System è proprietaria, è continuata nel tempo e si è attestata ben presto come *core* aziendale, attorno al quale vengono sviluppate applicazioni business di categorizzazione, analisi semantica, e ricerca semantica dei dati e delle informazioni, sia che esse derivino da fonti strutturate o non strutturate.
Il vero potere derivante dall'utilizzo di tecnologie di analisi linguistica deriva proprio dal fatto di poter convertire dati contenuti all'interno di fonti non strutturate (testi piani, email, contenuti Web, ecc…) in informazioni utili e affidabili che possono costituire un importante supporto alla decisione per gli organi dirigenziali di una generica azienda che operi in un settore altamente competitivo e dinamico.
L'importanza della analisi linguistica e semantica delle informazioni trova un importante fondamento anche negli sviluppi attuali delle tecnologie per la comunicazione. La continua evoluzione di Internet, e con esso del World Wide Web, sta elevando sempre più il concetto di ricerca delle informazioni a supporto dell'utente

finale, che sempre più spesso si trova in possesso di strumenti troppo deboli per effettuare ricerche di informazioni in modo rapido ed efficace all'interno dell'universo informativo pubblicato sulla Rete.
La risposta che la ricerca internazionale sta tentando di fornire a questa tipologia di problematiche ruota proprio attorno alla capacità di far comprendere alle macchine il linguaggio umano, in tutta la sua capacità espressiva e polimorfica, e in questo Expert System è sicuramente stato un precursore e un importante modello di riferimento a livello internazionale.

A tale proposito viene riportato di seguito un estratto tratto da un articolo comparso sul Sole 24 Ore:

> *"Quello di Expert System è un caso interessante per due motivi. Da un lato la capacità di crescere nel settore del software sul mercato italiano che non è uno degli ambienti più facili e aperti al mondo. (...) Dall'altro lato, Expert System, che ha giocato le sue carte nell'informatica semantica, è interessante perché adesso si trova al centro di uno dei settori più caldi del mondo del software"*
>
> *nòva24 – 7 giugno 2007*

Con l'inarrestabile affermarsi di Internet e delle tecnologie di distribuzione di massa dei contenuti, la quantità di documenti disponibili all'utenza è letteralmente esplosa e sembra non esserci limite alla proliferazione dei dati potenzialmente interessanti; grazie al grande successo della Rete tutti hanno a disposizione una quantità immensa di testi in formato elettronico.
Allo stesso modo, all'interno delle Intranet aziendali e nei documenti e email personali la quantità di materiale da gestire aumenta in modo esponenziale, rendendo sempre più difficile ritrovare i documenti che servono.
I sistemi per la gestione delle informazioni testuali usati fino all'avvento delle tecnologie semantiche non sono in grado di risolvere in modo soddisfacente il problema di separazione delle informazioni utili dal rumore di fondo.

Gli utenti si trovano sempre più spesso in queste situazioni critiche:

- troppe risposte (*overload*): il sistema non è in grado di catalogare e ordinare le informazioni in modo utile, quindi produce un grande quantità di risposte non rilevanti, all'interno delle quali, da qualche parte, si trova la risposta cercata.
- poche o nessuna risposta (*underload*): per l'inefficacia della fase di catalogazione o la scarsa capacità di discriminare del motore di ricerca, il sistema non capisce la domanda o non è in grado di ricondurla ad alcuna risposta.

Internet e le tecnologie collegate non potranno esprimere al meglio le proprie potenzialità finché questi problemi non saranno risolti.

COGITO® è un sistema software che "capisce" la lingua in un modo analogo a quanto fanno gli esseri umani, perché cattura tutti gli aspetti strutturali e lessicali del testo per comprenderne il significato.

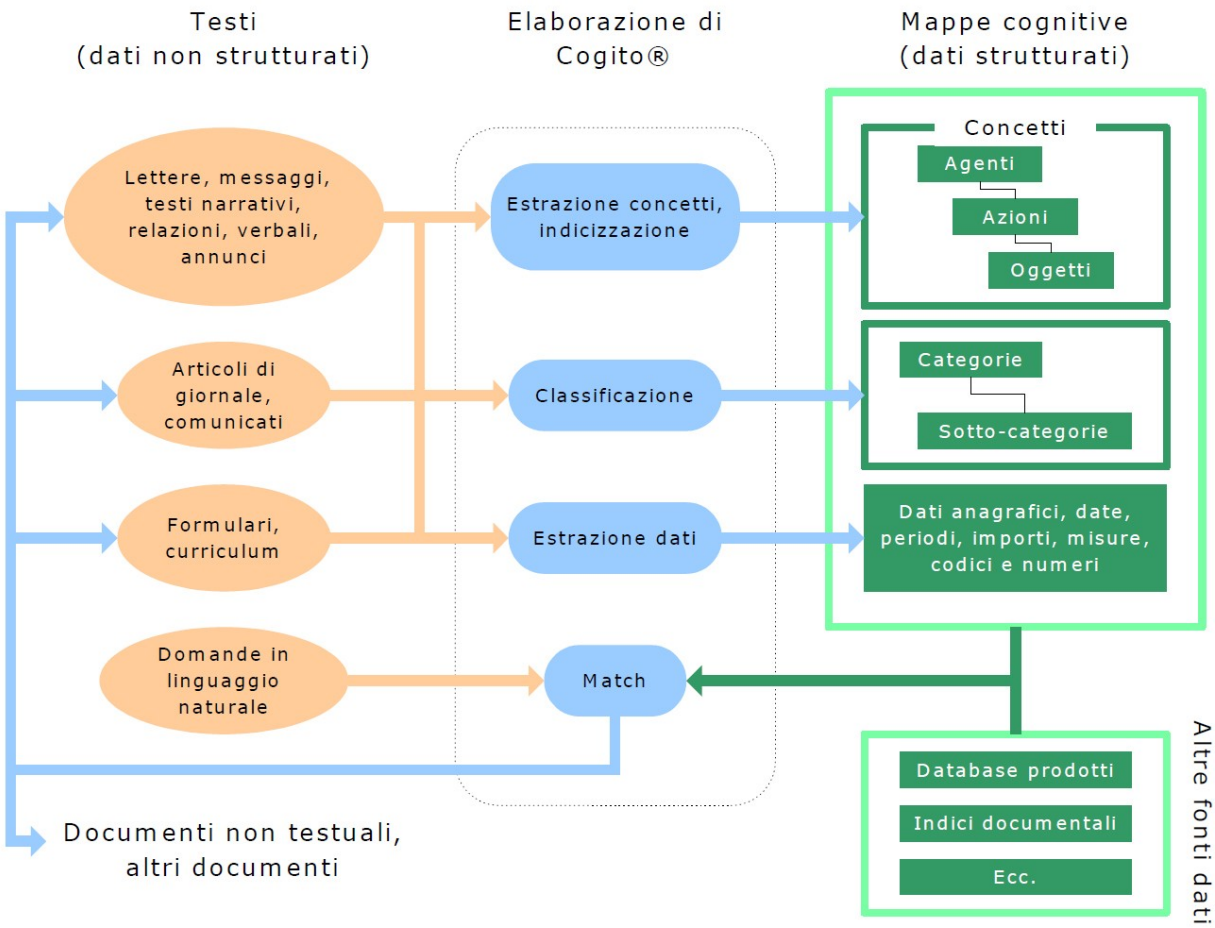

Figura 3.2 - Schema concettuale di *COGITO*®

Il risultato dell'elaborazione di COGITO® è una mappa cognitiva e concettuale, vale a dire una rappresentazione strutturata degli aspetti qualificanti del flusso di dati non strutturati in ingresso. La strutturazione dell'output consente ogni tipo di trattamento automatico degli elementi più significativi dei documenti.

Spesso per l'implementazione di questa tecnologia sono state affrontate scelte in controtendenza con quelle più diffuse a livello internazionale. Ad esempio i sistemi NLP sono programmati per ignorare le cosiddette *stop words*, parole che (come gli articoli o le preposizioni) non hanno un significato proprio; eppure sono elementi fondamentali per dare senso compiuto al discorso. COGITO®, invece, prende in considerazione tutte le parole, così come fa una persona che legge o scrive.

Per capire quanto sia importante questa caratteristica, consideriamo due frasi:

*"Mi hanno fornito **una** carta **di** credito."*

*" Mi hanno fornito **della** carta **a** credito."*

Queste due frasi differiscono solo per un articolo ed una preposizione, ma hanno un significato completamente diverso. Nella prima frase le parole "carta" e "credito" fanno riferimento ad un preciso strumento di pagamento, mentre nella seconda frase "carta" ha il suo significato più comune e "credito" si riferisce all'accordo intercorso tra il fornitore e chi ha ottenuto la carta. Per una persona, questa differenza di significato è ovvia, ma non è così per i programmi attuali di gestione del testo.

COGITO® è una tecnologia generale, scalabile e personalizzabile che può essere utilizzata per elaborare contenuti appartenenti a qualunque dominio semantico o settore di attività: documentazione tecnica specialistica, materiale enciclopedico, informazioni finanziarie, articoli di giornale, manualistica e così via.

Gli esempi che seguono e che illustrano il funzionamento di COGITO® rispetto agli altri metodi di elaborazione dei testi sono per la lingua italiana, seppure oggi la tecnologia linguistica di Expert System sia disponibile anche per l'inglese e il francese e in futuro lo sarà per il tedesco e l'arabo.

COGITO® è il risultato dell'integrazione di diverse sottocomponenti che vengono elencate di seguito:

- ✓ il *parser*: analizza la frase dal punto di vista morfologico, grammaticale e sintattico;

- ✓ il *lessico*: risorse per il riconoscimento delle parole e dei possibili significati;
- ✓ la *memoria*: storia delle analisi precedenti;
- ✓ la *conoscenza*: risorse per rappresentare la conoscenza del mondo reale;
- ✓ la *rappresentazione del contenuto*: il testo in forma cognitiva e strutturata;

Per comprendere il significato di una frase bisogna determinare innanzitutto il ruolo grammaticale di ogni parola e la componente addetta a questo ruolo è il *parser*. Esso gestisce in modo completo ed ottimale tutte le caratteristiche grammaticali della frase, effettua l'analisi logica e fornisce una base solida ai moduli che analizzano il contenuto.

Le informazioni sui significati possibili delle parole sono essenziali per la corretta interpretazione del contenuto di un testo. Queste informazioni sono memorizzate in una serie di reti semantiche realizzate in modo specifico per l'elaborazione automatica dei testi. Tali reti semantiche non sono solo semplici dizionari di termini, ma fitte reti di collegamenti e dati che consentono di rappresentare informazioni complesse, indispensabili per la disambiguazione. Grazie a queste informazioni, COGITO® sa che forme diverse (come "disastro aereo" e "sciagura aerea" oppure "motorino" e "ciclomotore") rappresentano in realtà lo stesso concetto, un'operazione impossibile per i sistemi che si limitano ad agire sulle parole e non sui concetti.

Quando leggiamo, compiamo inconsciamente una serie di operazioni che ci consentono di giungere alla comprensione del testo, una di queste attività è la memorizzazione persistente dei concetti significativi delle frasi lette in precedenza. Durante l'analisi dei documenti, COGITO® impiega una tecnica paragonabile per determinare il contesto semantico ideale per la disambiguazione e, successivamente, estrarre i concetti in modo preciso.

La cultura è un elemento chiave per capire ciò che si legge. Quando una persona con una buona cultura generale legge un testo specialistico sulla teoria della relatività ristretta di Einstein, comprende senza difficoltà le parole e l'impostazione generale del discorso, ma non riesce a capire veramente la sostanza di quanto letto a causa della mancanza della conoscenza specifica.

COGITO® contiene una conoscenza vasta e bilanciata del mondo reale, implementata sotto forma di regole descrittive che sono applicate durante l'analisi, con un meccanismo assimilabile al "buon senso" umano.

COGITO® è anche in grado di riconoscere gli elementi *particolari* (le date e i termini temporali, importi in denaro, quantità, ecc…) e di gestirli a livello concettuale.
La base di conoscenza generica di COGITO® può essere arricchita, tramite un meccanismo di apprendimento guidato, con le conoscenze specifiche di particolari domini cognitivi.
Il risultato dell'analisi di COGITO® è una mappa cognitiva del contenuto del testo, dove:

- ✓ ogni *concetto* è memorizzato in modo indipendente dalle parole usate per rappresentarlo;
- ✓ ogni *agente* è associato all'azione compiuta;
- ✓ ogni *oggetto* è collegato all'azione relativa;
- ✓ a ogni documento vengono attribuiti un argomento principale (ad esempio "sport" oppure "finanza") e gli eventuali argomenti secondari ("tennis" oppure "borsa"), le informazioni temporali, ed altre eventuali informazioni significative che sono memorizzate in questa rappresentazione.

La caratteristica fondamentale della mappa cognitiva è quella di essere un insieme strutturato di dati, per questo si presta facilmente ad ogni genere di elaborazione formale come ricerche, classificazione, sintesi, traduzioni ed altro ancora.

3.5 NLP e Web semantico

Con l'avanzamento delle tecnologie e i notevoli risultati ottenuti si delineano in modo sempre più definito i punti di contatto fra il Web Semantico e il trattamento automatico del linguaggio. La sinergia più nota è probabilmente quella che riguarda la progettazione, la creazione e in parte anche l'utilizzo delle ontologie.
Tuttavia tale aspetto può essere affrontato da due differenti e opposti punti di vista che rispondono alle seguenti domande:

- ✓ "Come può il Web Semantico migliorare le tecnologie NLP?"
- ✓ "Come possono le tecnologie NLP contribuire alla realizzazione del Web Semantico?"

Per rispondere alla prima domanda si pensi che in futuro tutte le risorse disponibili sulla Rete saranno etichettate attraverso un linguaggio indipendente dai sistemi, l'XML, e che

molte relazioni fra queste risorse saranno rese esplicite grazie alla semantica proposta dall'RDF. Se si concentra l'attenzione su un particolare tipo di risorse, le pagine Web, è opportuno pensare che l'esistenza di queste informazioni autodescrittive potranno essere di enorme aiuto per i sistemi NLP che si basano su algoritmi di apprendimento automatico.

Fino ad ora i ricercatori hanno utilizzato il Web solo come fonte di documenti per testare algoritmi statistici. Questo primo tipo di utilizzo ha evidenziato grandi vantaggi dal momento che la caratteristica innata del Web nel categorizzare e classificare le risorse permette di recuperare *corpora* specifici suddivisi per dominio, molto utili nell'ambito dell'analisi e test di applicazioni NLP.

Dato questo primo supporto del Web tradizionale a favore dell'NLP non è difficile immaginare che il Web Semantico garantirà risultati ancora migliori nel momento in cui le applicazioni NLP saranno in grado di sfruttare la conoscenza *cablata* all'interno delle risorse RDF, immagazzinando le informazioni e riutilizzandole in futuro per nuove analisi.

Attualmente si stanno spendendo grandi risorse per realizzare basi di conoscenza generalizzate della lingua e della cultura umana, utilizzate nel processo di disambiguazione di un testo da parte di un'applicazione NLP, il Web Semantico offre e offrirà in modo sempre più consistente una fonte vastissima di tale conoscenza gratuitamente.

Veniamo ora alla risposta della seconda domanda. Esiste un numero notevole di *papers* che analizzano la possibilità di utilizzare i sistemi NLP per effettuare *tagging* automatico delle pagine Web attraverso descrizioni RDF.

Dal punto di vista operativo la marcatura dei contenuti di un documento è sempre stato uno degli obbiettivi principali dell'NLP, ed è facile pensare che in futuro questo sarà uno dei suoi impieghi più diffusi nell'ambito del Web Semantico.

In questa visione ci sono tuttavia alcuni punti deboli che è necessario analizzare facendo un confronto con la storia dei database relazionali. I primi database commerciali comparvero all'inizio degli anni '80 e all'epoca sembrava ragionevole pensare che il passaggio da dati non strutturati a dati strutturai avrebbe potuto fornire un'ottima motivazione di una rapida adozione delle tecnologie NLP da parte dell'industria. Nonostante tutto ciò non avvenne in modo significativo e l'inserimento manuale dei dati

non fu mai sostituito completamente, anzi fu supportato con l'introduzione di nuovi strumenti che facilitavano l'inserimento di dati anche da parte di utenti privi di esperienza e competenze informatiche. Il successo dei *forms* nell'ambito del Web ne è un esempio lampante.

Allo stesso modo viene da chiedersi: "Perché le persone dovrebbero voler marcare automaticamente le proprie pagine Web dal momento che realizzare una descrizione RDF è troppo complesso per l'utente medio?".

Potrebbe essere sufficiente l'insorgere di una nuova tecnologia che faciliti questa operazione, come i *forms* hanno facilitato l'inserimento manuale dei dati.

Affinché tale scenario non si verifichi, destinando un allontanamento fra Web Semantico e NLP Luca Dini[38] ha evidenziato la necessità delle seguenti tre condizioni:

1. l'NLP deve fornire descrizioni RDF che siano utili anche per gli utenti umani e non solo per iterazioni macchina a macchina;
2. è necessario che almeno un motore di ricerca famoso e accreditato introduca la possibilità di effettuare efficacemente richieste innovative sfruttando il Web Semantico;
3. gli algoritmi di *tagging*, di qualsiasi tipologia, devono migliorarsi in accuratezza.

L'ultimo punto è forse il più importante, infatti anche se al giorno d'oggi la classificazione automatica ha raggiunto un buon livello di precisione, è necessario far avanzare la ricerca in questo ambito.

Nella visione del Web Semantico non è infatti sufficiente poter affermare che una certa pagina parli di una persona, vi è la necessità di qualificare la risorsa in modo nettamente più dettagliato. Nell'esempio proposto è necessario che di tale persona vengano caratterizzate le abilità, il luogo di residenza, l'età e così via.

Per raggiungere tale scopo bisogna che le applicazioni che effettuano tagging siano in grado di recuperare le informazioni mancati da pagine Web alternative realizzando nuovi collegamenti fra esse.

[38] Luca Dini è il Presidente della CELI srl e della CELI Associazione. In precedenza è stato CEO della DIMA Logic. Ha conseguito il dottorato in Linguistica Generale presso la Scuola Normale Superiore di Pisa.

4 IMPLEMENTAZIONE DI UN MODELLO PER *INFORMATION EXTRACTION*

Come anticipato nell'introduzione il progetto descritto all'interno di questo elaborato ha come scopo principale l'implementazione di un sistema automatico per l'estrazione di informazioni inerenti a *persone*, *luoghi* e *organizzazioni*. Per informazioni si intende qualsiasi insieme di vocaboli singoli o multipli che possano caratterizzare in modo significativo l'entità a cui si riferiscono.
Il nome scelto per l'applicazione, *OKKAM-POP*, è motivato da due fattori principali. Il primo è inerente al fatto che i dati estratti non sono fini a se stessi ma hanno lo scopo di contribuire al popolamento dell'*Entity Rrepository* dell'*OkkamCORE* descritto al paragrafo 1.4. L'apporto di una migliore descrizione degli elementi all'interno dell'infrastruttura *OKKAM* è giustificata dal fatto che in questo modo sarà possibile evitare sempre più l'ambiguità fra le entità e permetterà inoltre di instaurare sempre maggiori relazioni fra le risorse rendendo più evidente l'utilità dell'*ENS* agli occhi degli utenti del Web.
Il secondo fattore deriva dal fatto che il presente progetto si limita alla caratterizzazione di *people*, *organizations* e *places*, tripletta che viene riassunta appunto dall'acronimo *POP*.

4.1 Descrizione del dominio del problema

E' opportuno innanzitutto definire con precisione il concetto di Information Extraction. Questo processo ha lo scopo di estrarre informazioni strutturate, contestualmente e semanticamente ben definite, a partire da un certo dominio, come documenti non strutturati in formato comprensibile da una macchina.
Un'operazione di Information Extraction può essere applicata alla seguente frase:

> *"Con una transazione pari a 3,4 miliardi di dollari Adobe ha acquisito Macromedia, un vero e proprio terremoto che non coglie di sorpresa i più informati."*

A livello concettuale la parti più interessanti di questa affermazione sono in prima battuta "*Adobe ha acquisito Macromedia*" e successivamente potrebbe essere importante riportare il valore finanziario dell'acquisizione "*3,4 miliardi di dollari*".
Ottenere un sistema che effettui automaticamente questa operazione così naturale per la mente umana presenta numerose difficoltà legate soprattutto al complessità del linguaggio naturale, si pensi ad esempio alla stessa frase riproposta nel seguente modo:

> *"Adobe ha sfidato la concorrente Microsoft acquisendo Macromedia. Un vero e proprio terremoto che non coglie di sorpresa i più informati. Costo dell'operazione 3,4 miliardi di dollari."*

Concettualmente i contenuti sono molto simili a quelli precedentemente espressi tuttavia l'aumento di complessità della struttura sintattica e la suddivisione in più periodi dell'informazione potrebbe creare non pochi problemi a un sistema automatico.
Gli obiettivi associati all'Information Extraction sono:

- ✓ il riconoscimento di entità principali: essenziale all'interno di un documento è il riconoscimento delle entità principali come ad esempio i nomi delle persone, delle organizzazioni, dei luoghi oppure altre tipologie di dati tipicamente definiti da espressioni, come ad esempio i numeri, le date, ecc…
- ✓ identificazione di concetti, descrizioni e *coreferenze*: la complessità del linguaggio naturale si evidenzia nel caso in cui sia necessario valutare a quale entità si riferisce una certa apposizione oppure nel caso in cui si debba identificare il riferimento di una particolare *anafora*[39].
- ✓ estrazione terminologica: spesso può essere rilevante catturare all'interno di un testo un serie di termini specifici che assumono una grande importanza all'interno di un determinato dominio.
- ✓ estrazione delle relazioni: oltre all'identificazione delle singole entità può essere molto interessante interpretare le relazione che sussistono fra esse come ad esempio comprendere che una persona si trova in certo luogo dalla frase "Daniele a Londra sta incontrando più problemi del previsto".

[39] L'anafora è una tecnica retorica che consiste nel richiamare un concetto precedentemente espresso con l'utilizzo di un termine, specialmente di un pronome. Ad esempio "Tutto ciò è stato causato da lui, questo è disdicevole".

La complessità e la precisione nell'estrarre informazioni, come verificato durante il progetto, è strettamente dipendente dal formato e dalla tipologia dei dati su cui viene effettuata tale operazione. Un impatto notevole è dato ad esempio dall'omogeneità del formato e dello stile dei documenti non strutturati che vengono analizzati, soprattutto se le metodologie scelte per effettuare *Information Extraction* si basano su modelli empirici come quello che verrà esposto in questo capitolo.

La tipologia di documenti analizzabili è estremamente vasta, tuttavia è possibile fornirne una categorizzazione generalizzata basandosi su due principali unità di misura, la granularità e l'eterogeneità.

La granularità viene intesa come la dimensione strutturale del documento considerata per l'estrazione delle informazioni. La forma più popolare di estrazione è quella che considera una frase come elemento processabile per la ricerca di *pattern* strutturati e riconoscibili come ad esempio indirizzi, numeri telefonici, indirizzi email, oppure di intere porzioni di testo ritenute estremamente pertinenti con le entità riscontrate. In quest'ultimo caso, che rientra negli scopi del progetto, le porzioni di testo possono essere considerate come una sequenza di campi a loro volta strutturati e concatenati insieme per formare un certo concetto o modello descrittivo. In questo modo il processo di estrazione può basarsi su un'operazione preliminare di segmentazione delle parole e su una seconda fase di riconoscimento di certe sequenze presenti in prossimità di un'entità.

Altre tipologie di granularità che possono essere adottate sono il paragrafo o l'intero documento. Come è intuibile estrazioni che considerano uno spettro d'ampiezza più ampio risultano estremamente più complesse tuttavia risultano necessarie nel caso in cui si debba considerare il contesto di più frasi o di un intero documento per ottenere estrazioni significative. Si prenda come esempio un sistema di estrazione delle informazioni che miri alla descrizione di eventi generici. E' molto probabile che la distribuzione degli elementi significativi di un evento, come data, luogo, elenco dei partecipanti e attività svolte siano distribuite in modo sufficientemente omogeneo all'interno di un intero testo. Una tecnica molto diffusa per effettuare estrazioni considerando come granularità di riferimento il paragrafo o l'intero documento è quella di implementare tecniche di filtraggio che permettano di considerare solo le parti

rilevanti di un documento relativamente lungo in modo da facilitare la ricerca delle informazioni inerenti alla stessa entità.

Come detto precedentemente la seconda unità di misura secondo la quale è possibile categorizzare le tipologie di documenti è l'eterogeneità dei contenuti. Questo fattore oltre ad essere importante per l'Information Extraction lo è anche per quelle tipologie di applicazione il cui scopo è attribuire un certo dominio di appartenenza a testi generici. Si pensi a un primo caso che consideri come testi di riferimento per l'estrazione una serie di news giornalistiche e un secondo che utilizzi un intero libro di narrativa per effettuare la stessa operazione.

L'analisi delle news presenta un gran vantaggio per l'*Information Extraction*, infatti per loro stessa natura spesso sono monotematiche e concentrano la propria attenzione su poche entità coinvolte in relazioni chiare e frequentemente ben descritte, dato che lo scopo di un estratto giornalistico è proprio quello di informare in modo conciso e il più completo possibile il lettore.

Nel secondo caso, invece, l'eterogeneità degli argomenti contenuti nell'intero testo rendono difficile e imprecisa l'attribuzione di un unico dominio, almeno che non si utilizzi una tecnica di segmentazione che scomponga la narrazione in parti maggiormente omogenee. Tuttavia anche in questo caso non si risolve il problema della dispersione delle descrizioni inerenti alla stessa entità come ad esempio un personaggio del racconto narrativo.

Il progetto OKKAM-POP ha come scopo proprio l'analisi linguistica e semantica di news giornalistiche per l'estrazione automatica di concetti e caratterizzazioni delle entità *persone, organizzazioni* e *luoghi*.

Di seguito viene riportato un esempio di tali notizie, evidenziando in grassetto le entità di interesse e sottolineando i concetti che si desidera estrarre:

> *TORINO, 18 GIU - I consigli di fabbrica di **Mirafiori** hanno deliberato per mercoledì due ore di sciopero "contro lo scalone e gli scalini e la revisione al ribasso dei coefficienti" per il calcolo delle pensioni e hanno promosso una raccolta di firme per invitare i segretari generali nello stabilimento **Fiat** (Fabbrica Italiana Automobili Torino) prima della conclusione della trattativa e per sottoporre l'eventuale intesa al voto dei*

lavoratori. Lo ha reso noto il sindacato ***Fiom-Cgil****. Le* ***Rsu*** *aziendali hanno deliberato unitariamente due ore di sciopero, per ogni turno, dalle 10 alle 12 e dalle 14 alle 16, con iniziative fuori dai cancelli. Intanto domani, scioperera'* ***l'Avio Group****, ex Fiat Avio, di Rivalta, con corteo fuori dallo stabilimento. "Il governo e i sindacati ascoltino i buoni consigli dei delegati di* ***Mirafiori*** *e usino la forza che a loro viene data. Saremo tutti giudicati dall'esito di questo negoziato", afferma il segretario generale della Fiom torinese,* ***Giorgio Airaudo****.*

La prima operazione che deve essere effettuata sul testo appena presentato è il riconoscimento di quelle che possono essere considerate entità rilevanti, cercando di evitare il più possibile rumore molto frequente all'interno di un testo scritto in linguaggio naturale. Ad esempio *Rsu*, acronimo di rappresentanze sindacali unitarie, potrebbe non essere di interesse dato che non rappresenta un'entità specifica ma un insieme di organi diffusi all'interno di molte aziende. In seconda battuta è necessario verificare se nei pressi di un'entità esista una possibile descrizione utile. La scelta di osservare un intorno definito delle entità per la rilevazione dei concetti è posto come vincolo per aumentare la precisione delle informazioni estratte. È plausibile che vengano perse in questo modo alcune informazioni tuttavia dato l'enorme numero di news che verranno analizzate, circa 1.160.000, è stato scelto questo compromesso onde evitare una proliferazione incontrollata del rumore, ovvero di informazioni non pertinenti con gli scopi del progetto.

4.2 Metodi per l'*Information Extraction*

Le metodologie attraverso le quali è possibile implementare sistemi di *Information Extraction* sono valutabili lungo due dimensioni principali: l'automatizzazione delle operazioni e la tipologia di algoritmo che valuta le estrazioni. A ciascuna di queste due dimensioni possono essere attribuiti, in base all'esperienza acquisita durante il progetto, due valori discreti che contribuiscono a realizzare lo schema concettuale presentato in Figura 4.1.

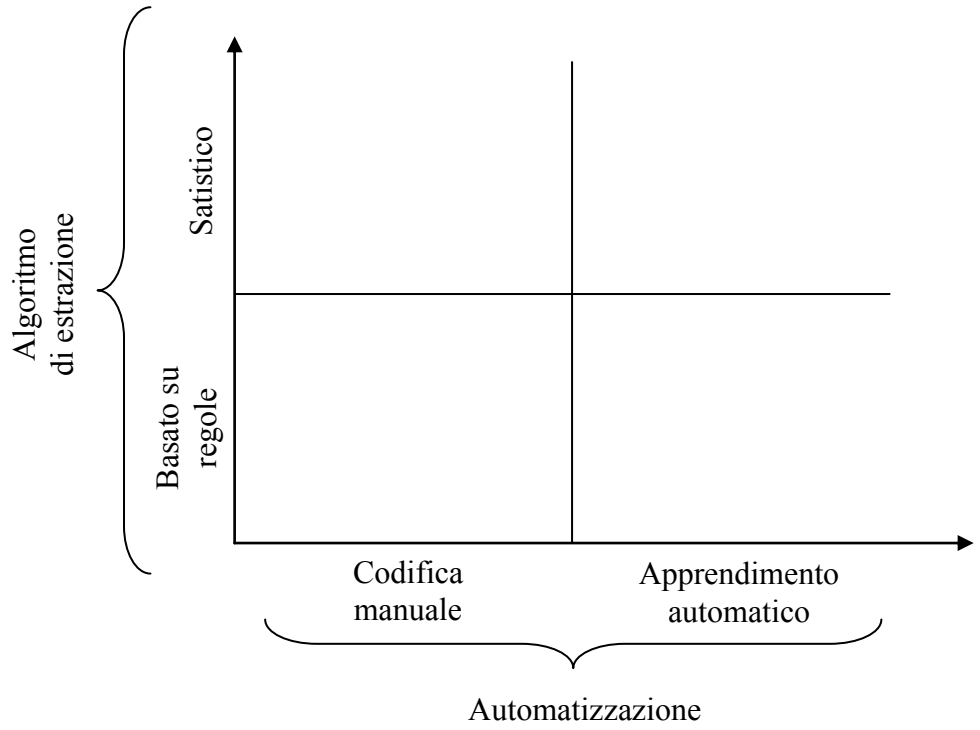

Figura 4.1 – Classificazione dei metodi per *Informaztion Extraction*

E' possibile quindi distinguere quattro principali tipologie di metodi per *l'Information Extraction* risultanti dalla combinazione delle due dimensioni del grafico in Figura 4.1.
Analizzando il livello di automazione un sistema a codifica manuale richiede le competenze di una persona per la definizione di regole ed espressioni regolari o di metodi basati sulla statistica. Tale persona deve essere un esperto di dominio, un programmatore e in più deve possedere conoscenze linguistiche per poter sviluppare metodi robusti di estrazione.
Al contrario i metodi automatici necessitano di esempi etichettati manualmente che costituiscono il *training set*[40] di partenza. Anche in questo caso è necessario l'apporto di esperienza da parte di un uomo nell'identificazione di quelli che possano essere esempi rappresentativi per l'apprendimento di un sistema automatico. L'utilizzo di modelli automatici si scontra inoltre con il fatto che devono essere previste e definite funzionalità precise per la frequente possibilità di incontrare dati mai modellati precedentemente.

[40] Insieme di dati omogenei che vengono utilizzati per le fasi iniziali dell'auto-apprendimento di sistemi automatici.

La decisione fra l'impiego di modelli a codifica manuale o di modelli automatici deve essere presa in base agli obiettivi che si vogliono raggiungere con l'operazione di *Information Extraction* e in base all'ammontare di rumore presente all'interno dei dati che devono essere sottoposti ad analisi.

Analizzando la seconda dimensione, ovvero l'algoritmo di estrazione, la scelta può ricadere su modelli che si basano su regole e modelli statistici. I metodi di estrazione basati sulle regole rappresentano un modo molto statico di affrontare il problema e risultano più semplici da implementare rispetto a modelli statistici. I modelli statistici risultano tuttavia più robusti al caratteristico rumore presente all'interno dei dati non strutturati. I modelli basati su regole sono molto utili nel caso in cui si abbia a che fare con domini ristretti, in cui la componente umana sia fondamentale per la valutazione dei risultati, mentre all'interno di domini aperti i modelli statistici si comportano in modo nettamente migliore e risultano più appropriati.

Come è intuibile vi è un aumento di complessità nell'implementazione dei sistemi di estrazione a seconda delle scelte effettuate lungo le due dimensioni, in particolare sistemi basti su regole e a codifica manuale risultano i più semplici, mentre sistemi che coivolgono modelli statistici e algoritmi di apprendimento automatico sono i più complessi.

Dato che questo progetto ha rappresentato per me il primo approccio al mondo delle applicazioni *NLP* e visto che ritengo sia sempre necessario, quando si affronta un nuovo problema, considerare in primo luogo la soluzione più semplice e realizzabile, ho deciso di indirizzare la mia scelta su un modello basato su regole e codifica manuale, mentre nell'ultimo capitolo di questo elaborato verrà presentato solo a livello teorico un modello statistico che fa uso di un algoritmo di apprendimento e decisione automatico.

Iniziano quindi a delinearsi le prime caratteristiche dell'applicazione *OKKAM-POP*, la quale, anticipando i commenti finali, ha completamente esaudito le specifiche iniziali del progetto grazie soprattutto all'impiego delle tecnologie linguistiche *Expert System*.

4.3 Gli strumenti linguistici *Expert System*

Come descritto nel paragrafo 3.4 *Expert System* sviluppa numerose applicazioni per l'analisi e la gestione di dati non strutturati, ognuna con caratteristiche specifiche che si adattano alle necessità del cliente, tuttavia la quasi totalità dei prodotti fondano il proprio potenziale sulla tecnologia linguistica *COGITO*® .

Questa tecnologia, sviluppata nell'arco di circa 15 anni, ha un'architettura realizzata a strati le cui colonne portanti sono il *SENSIGRAFO*® e il *DISAMBIGUATORE*.

Il *SENSIGRAFO*® è rappresentato da una rete semantica lessicale che si ispira alla struttura di *WordNet* descritta precedentemente. La differenza con l'analogo open source è data dalla ricchezza dei suoi contenuti, dalla presenza di glosse molto dettagliate per ciascun concetto, dalla maggiore complessità delle relazioni instaurate fra i diversi concetti, dalla presenza di dati sulla frequenza di utilizzo di ciascun concetto e dalla sua efficienza in termini di prestazioni durante la fase di analisi dei documenti.

Il *SENSIGRAFO*® nel suo complesso costituisce una base di conoscenza che fornisce con una buona copertura la rappresentazione del mondo conosciuto e la sua ricchezza è percepibile dal numero di concetti e relazioni presenti al suo interno.

Il *SENSIGRAFO*® contiene 350.000 lemmi e forme fraseologiche di cui circa 240.000 sostantivi, 50.000 verbi e 60.000 aggettivi, inoltre al suo interno sono presenti più di 2 milioni e 800 mila connessioni rappresentate da relazioni di iponimia, iperonimia, troponimia[41], ecc…

La presenza di queste relazioni è fondamentale per il processo di disambiguazione di un testo, si pensi ad esempio alla frase "Sposta il cavallo, è una mossa conveniente". Per un sistema automatico che cerchi di comprendere se nella frase si parla dell'animale cavallo o della pedina degli scacchi è una priorità il poter far affidamento su una rete semantica come il *SENSIGRAFO*®. Se in fatti in esso viene mantenuta una relazione fra il verbo "spostare", il concetto appartenente al dominio degli scacchi "cavallo" e il sostantivo "mossa" sarà molto probabile che l'operazione di disambiguazione abbia successo.

[41] È simile alla relazione di iponimia con riferimento alla classe dei verbi.

Inoltre grazie a tali relazioni è possibile utilizzare tecniche di *backtracking*[42] per correggere disambiguazioni errate durante l'analisi del testo. Il linguaggio naturale viene infatti analizzato in modo sequenziale e, ritornando all'esempio precedente, è quindi possibile che fino all'analisi del primo periodo "Sposta il cavallo" venga considerato il concetto di animale e proseguendo successivamente fino alla parola "mossa" venga invece riconsiderato il concetto appartenente al mondo degli scacchi.

Il *SENSIGRAFO*® tuttavia è una condizione necessaria ma non sufficiente per riuscire ad effettuare le operazioni linguistiche appena descritte. La componente che completa il quadro fondamentale della tecnologia *COGITO*® è il *DISAMBIGUATORE*.

Esso implementa gli algoritmi necessari per effettuare tutte le operazioni caratteristiche dell'analisi linguistica: *parsing* morfologico e grammaticale, disambiguazione dei significati, analisi logica e sintattica e controlli di congruenza semantica. Oltre alle caratteristiche informazioni grammaticali il *DISAMBIGUATORE* ritorna anche interessanti dati inerenti al contesto di un documento come ad esempio i lemmi e i domini principali trattati, elencati per ordine di importanza e in più corredati di una percentuale di pertinenza.

Expert System afferma che, grazie alle proprie tecnologie, in contesti liberi ottiene una precisione nella comprensione del linguaggio naturale del 90%.

L'utilizzo della tecnologia costituita da *SENSIGRAFO*® e *DISAMBIGUATORE* è semplificata grazie alla presenza di librerie aziendali di livello logico superiore che forniscono un primo accesso alle funzionalità di analisi linguistica. Come verrà evidenziato successivamente durante il progetto è stata studiata anche una seconda tipologia di accesso alla tecnologia *COGITO*® di più alto livello.

Il nucleo delle librerie aziendali è una classe che rappresenta una struttura ad albero di un testo disambiguato. Vengono fornite inoltre una serie di strutture che permettono di navigare tale albero in modo semplice e intuitivo.

In Figura 4.2 viene mostrata una semplice rappresentazione generalizzata di una struttura ad albero di un testo disambiguato.

---

[42] Il *backtracking* è una tecnica per trovare soluzioni a problemi in cui devono essere soddisfatti dei vincoli. Con questa tecnica si considerano successivamente tutte le possibili soluzioni, scartando man mano le condizioni che non soddisfano i vincoli.
Una tecnica classica consiste nell'esplorazione di strutture ad albero e tenere traccia di tutti i nodi e i rami visitati in precedenza, in modo da poter tornare indietro al più vicino nodo che conteneva un cammino ancora inesplorato nel caso che la ricerca nel ramo attuale non abbia successo.

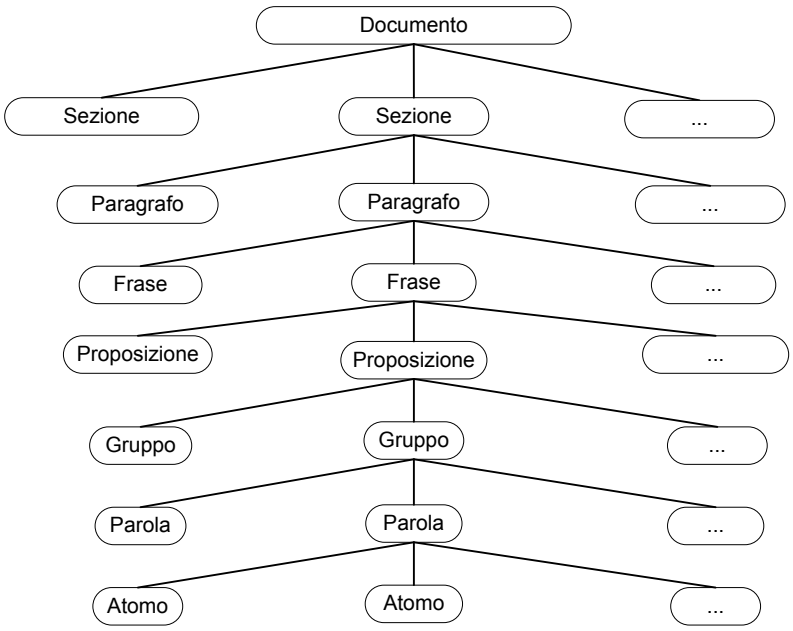

Figura 4.2 – Struttura ad albero di un testo disambiguato

L'organizzazione del linguaggio naturale attraverso questa struttura ad albero permette di navigare in nodi sia in modo verticale che in modo orizzontale permettendo analisi che possano sfruttare la generalizzazione e la specializzazione degli elementi del testo o analisi che mantengano la naturale sequenzialità delle parole.

Ogni nodo di tale struttura mantiene numerose informazioni di tipo grammaticale e strutturale tuttavia dato il carattere privato di questa tecnologia di seguito vengono descritte solo le informazioni del nodo "parola" utilizzato in modo intensivo all'interno di questo progetto.

Tale nodo permette di recuperare:

- ✓ la posizione di inizio dell'elemento nel testo utilizzato per costruire l'albero;
- ✓ la lunghezza dell'elemento;
- ✓ la posizione di fine dell'elemento;
- ✓ il valore di "Synset" associato all'elemento;
- ✓ il valore del lemma principale associato all'elemento;
- ✓ il tipo grammaticale della parola che può assumere i seguenti valori:
  - ING, ignoto
  - ART, articolo

- PRE, preposizione
- SOS, sostantivo
- AGG, aggettivo,
- VER, verbo
- AUX, componente ausiliaria o servile dei verbi
- CON, congiunzione
- AVV, avverbio
- PRO, pronome
- NPR, nome proprio
- NPH, nome proprio di persona (anche presunta tale)
- DAT, data
- PNT, punteggiatura
- ADR, indirizzi
- MON, valore monetario

Si è accennato prima a una seconda tipologia di accesso alla tecnologia linguistica *COGITO*®, tale metodologia fa uso di un paradigma *client-server* per fornire funzionalità di analisi linguistica ad alto livello pilotabili tramite una richiesta inviata al server in linguaggio *XML*.

Il client verrà d'ora in poi chiamato *client linguistico* e si farà riferimento al server con il termine *server linguistico*. Per completare il quadro di questa architettura è necessario introdurre un ulteriore elemento che definiremo *coordinatore delle richieste*. Tale componente fornisce un livello di astrazione fra il *client linguistico* e il *server linguistico* che svincola il primo dalla necessità di sapere dove sia ubicato il secondo e permette di espandere in modo naturale il paradigma *client-server* in paradigma *multi-client* e *multi-server*. Il *coordinatore delle richieste* fornisce un unico punto di accesso a tutte le richieste provenienti dai *client linguistici* e si occupa della redirezione bilanciata di tutte le richieste verso le istanze di *server linguistici* ad esso collegati.

Il *server linguistico* è implementato come una vera e propria applicazione stand-alone che fornisce un servizio in ascolto su un certa porta, tale servizio rimane in attesa che venga inviata una richiesta di analisi. L'unica ulteriore informazione che necessita il *server linguistico* è l'identificazione del *coordinatore delle richieste* a cui deve essere collegato. La tipologia di analisi linguistica eseguibile dal *server linguistico*, oltre che

essere che essere pilotabile tramite una richiesta *XML* del *client linguistico*, è configurabile e personalizzabile attraverso altre due modalità. La prima è rappresentata da regole scritte con un linguaggio proprietario di *Expert System*, la seconda invece vede la possibilità dell'inserimento all'interno del *server linguistico* di un *plugin di post-processing* che può rielaborare in modo completo la risposta standard in termini di contenuti e formato *XML*. Per chiarire maggiormente il funzionamento del *plugin di post-processing* di seguito viene mostrato un *activity diagram* che evidenzia le fasi principali di un'analisi linguistica e la sequenza delle operazioni svolte a partire dall'arrivo di una richiesta da parte di un *client linguistico*.

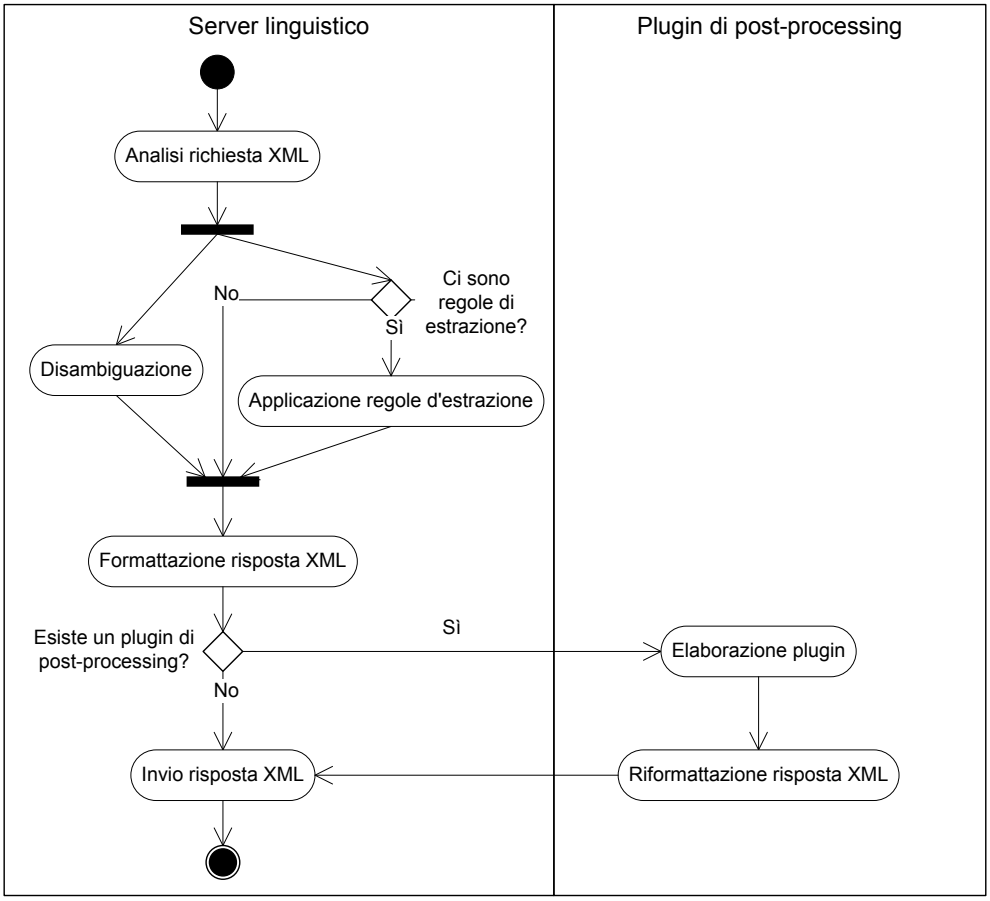

Figura 4.3 – Activity diagram delle operazioni svolte da un server linguistico

Come si vedrà nel prossimo paragrafo l'implementazione di un *plugin di post-processing* per il *server linguistico* rappresenta uno degli sforzi maggiori di questo progetto dato che proprio al suo interno sono state implementate le regole di estrazione dei concetti relativi alle entità.

Il *client linguistico* rappresenta un'applicazione che formatta in modo opportuno un messaggio contenente la richiesta di analisi in linguaggio *XML* e la invia al *coordinatore delle richieste*. *Expert System* fornisce le specifiche per l'implementazione di un *client linguistico* attraverso diversi linguaggi di programmazione, come ad esempio Java, C#, C++, infatti l'astrazione fornita dalla formattazione *XML* della richiesta permette di svincolare i formalismi e problemi di compatibilità legati ai linguaggi di programmazione, permettendo di scegliere quello che meglio si adatta alle necessità del caso.

Di seguito viene mostrato un semplice esempio di richiesta *XML* che può essere inviata al *server linguistico*. Vengono omesse le parti inerenti alla tecnologia *COGITO*®.

```
<?xml version="1.0" encoding="Windows-1252" ?>
<RICHIESTA>
   <DOCUMENTO TIPO="XML">
      <SEZIONE NOME="BODY">
         L'amministratore delegato Sergio Marchionne  ha affermato che la FIAT,
         azienda automobilistica di Torino, è in ripresa
      </SEZIONE>
   </DOCUMENTO>
   <ANALISI>
      …
   </ANALISI>
</RICHIESTA>
```

Tale richiesta è stata proposta per fornire un esempio concreto ma non verrà spiegata in modo dettagliato, si evidenzia solo che tramite tali richieste è possibile esplicitare la tipologia di output che si vuole ottenere. Sono numerose infatti la caratteristiche linguistiche che possono essere ritornate da un *server* tuttavia, a seconda degli scopi, potrebbe essere sufficiente ottenerne solo una parte nella risposta ritornata dall'analisi.

Una possibile risposta alla precedente richiesta può essere ad esempio la seguente porzione di codice XML.

```
<?xml version="1.0" encoding="Windows-1252" ?>
<RISPOSTA>
      <ESTRAZIONI TIPO="PEOPLE">
         <ENTITA NOME="Sergio Marchionne">
            …
         </ENTITA>
      </ ESTRAZIONI>
      < ESTRAZIONI …>
         …
      </ ESTRAZIONI>
</ RISPOSTA >
```

Di seguito viene infine rappresenta un'architettura tipicamente utilizzata per effettuare analisi linguistiche su una grande quantità di dati come ad esempio il corpus di notizie giornalistiche che è stato processato nell'ambito di questo progetto.

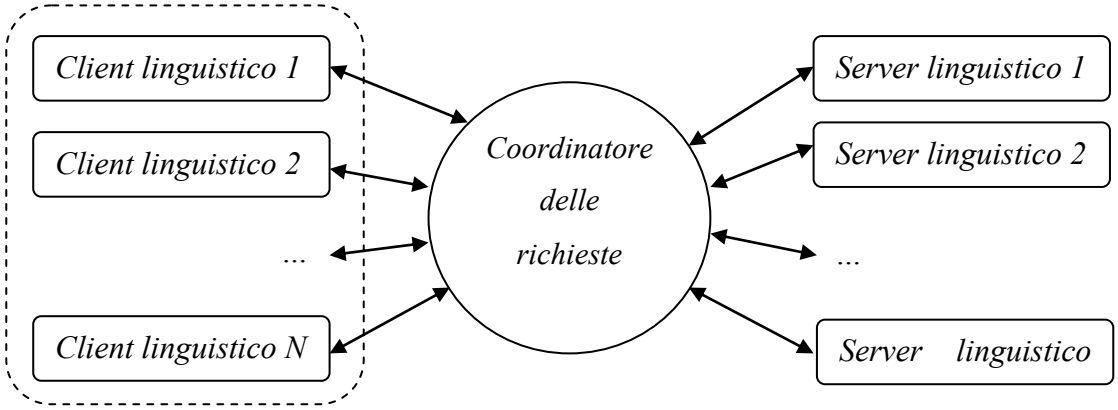

Figura 4.4 – Architettura scalabile per analisi linguistiche

L'architettura presentata risulta estremamente scalabile dal punto di vista pratico, anche se dal punto di vista teorico il *coordinatore delle richieste* potrebbe rappresentare un collo di bottiglia. Tuttavia ciò non avviene, entro certi limiti nella proliferazione di *client e server,* dal momento che ciascuna richiesta effettuata da un *client linguistico* risulta essere bloccante, ovvero il *client* deve attendere una risposta o lo scadere di un timeout prima di poter inviare un'ulteriore richiesta e dal momento che le operazioni di analisi linguistica da parte di un *server* risultano decisamente più lente rispetto alle altre operazioni di coordinazione e di invio e ricezione dei messaggi.

4.4 Modello teorico di soluzione del problema

Lo studio del problema affrontato in questo progetto ha evidenziato la possibilità di una sua scomposizione in diverse sottocomponenti. Si è deciso in seguito di affrontare questa strada per diverse motivazioni. Un flusso dei dati suddiviso in diverse fasi offre innanzitutto un aumento della *manutenibilità* dell'intero processo. Infatti la possibilità di mantenere e riutilizzare risultati parziali rappresenta un evidente vantaggio data l'enorme quantità di dati che vengono processati dall'interno sistema. Come verrà mostrato fra poco infatti anche semplici elaborazioni, come ad esempio l'estrazione di contenuti da strutture *XML*, dovendo essere ripetute su circa 1.160.000 documenti, impiegano tempi non sottovalutabili, indicativamente fra le 5 e le 10 ore.

Inoltre la suddivisione del flusso dei dati in diverse fasi ha permesso un maggiore controllo durante lo sviluppo del sistema garantendo un'analisi più approfondita sulle scelte effettuate.

In Figura 4.5 viene mostrata l'architettura dell'intera applicazione *OKKAM-POP*.

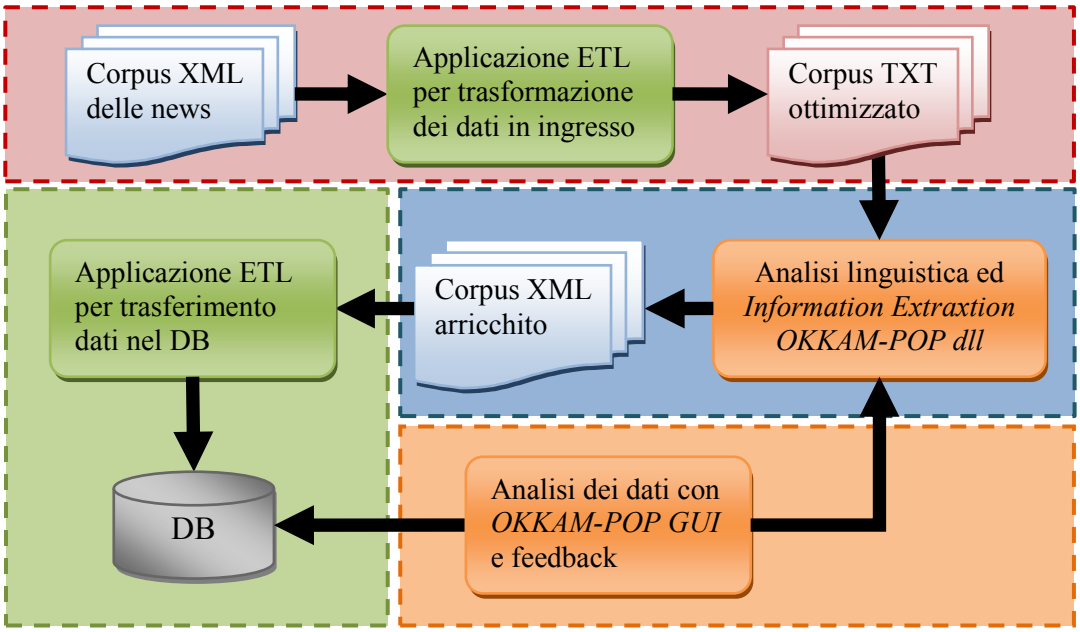

Figura 4.5 – Architettura dell'applicazione *OKKAM-POP*

Come si può notare le diverse componenti sono raggruppate in macroblocchi che rappresentano le principali fasi di elaborazione.

Il primo macroblocco, di colore rosso, rappresenta la fase preliminare di formattazione dei dati che permette di ottimizzare il corpus per le successive elaborazioni ed è

costituito essenzialmente da un'applicazione *ETL*. *ETL* è l'acronimo di *Extract, Transform, Load* e rappresenta un classe di applicazioni che hanno lo scopo di trasformare il formato dei dati provenienti da un sorgente per il successivo inserimento degli stessi all'interno di un ulteriore sistema. Le operazioni più comuni di un programma *ETL* sono:

- ✓ la selezione dei dati di interesse;
- ✓ la normalizzazione dei dati, per esempio l'eliminazione dei duplicati;
- ✓ la traduzione dei dati codificati;
- ✓ la derivazione di nuovi dati calcolati;
- ✓ l'accoppiamento tra dati recuperati da sorgenti differenti
- ✓ il raggruppamento dei dati.

Inizialmente le notizie giornalistiche sono contenute all'interno di files *XML* organizzati all'interno del file system nel seguente modo: il corpus contiene tre directory che rappresentano 3 differenti anni, ciascun anno contiene altrettante directories quanti i mesi dell'anno e a loro volta ciascun mese contiene tante directories quanti i giorni di ciascun mese.

I files *XML* che contengono le *news* sono strutturati secondo lo standard *NewsML*. Questo formato, proposto dall'*IPTC*[43], ha lo scopo di fornire un *XML Schema* generalizzato che regola la descrizione delle notizie con metadati appositamente studiati per facilitare la condivisione e lo scambio delle informazioni fra agenzie di stampa distribuite su tutto il territorio mondiale. Il risultato finale di questa prima elaborazione è un corpus in formato *TXT* strutturato nello stesso modo del corpus *XML* iniziale ma che mantiene solo le informazioni utili per l'analisi linguistica eliminando tutti i metadati superflui.

Il secondo macroblocco, evidenziato in blu, rappresenta la fase più importante del flusso dei dati ed è costituito principalmente dall'applicazione *OKKAM-POP*. Tale applicazione dal punto di vista funzionale è costituita a sua volta da un *client linguistico* che interagisce con un *server linguistico* al cui interno viene integrato un *plugin di post-processing* realizzato sfruttando le librerie aziendali di *Expert System*. All'interno di tale *plugin* sono implementate le regole di estrazione che catturano le informazioni utili per

[43] L'International Press Telecomunications Council è un consorzio costituito dalle più famose agenzie di stampa mondiali costituito per realizzare e mantenere standard tecnici per lo scambio di news.

la descrizione delle entità. Il risultato finale di questa fase è un corpus strutturato come il corpus iniziale, ma al cui interno sono presenti files *XML* che mantengono le informazioni linguistiche e i concetti relativi alle entità.

La terza fase, rappresentata dal macroblocco di colore verde, si occupa di trasferire i dati dal corpus *XML* arricchito all'interno di un database appositamente strutturato. Tale operazione viene effettuata da un'applicazione *ETL* che come si vedrà successivamente è stata integrata all'interno di un'intuitiva interfaccia grafica.

L'ultima fase, rappresentata dal macroblocco arancione, è costituita dall'analisi dei dati estratti. Tale operazione è effettuabile tramite un'applicazione grafica, denominata *OKKAM-POP GUI*, che si interfaccia direttamente al database e ne permette una comoda navigazione. Attraverso *OKKAM-POP GUI* è possibile valutare l'efficacia delle regole implementate per l'estrazione delle informazioni, per questo motivo viene integrato all'interno di questa fase un processo di feedback manuale atto a produrre un raffinamento continuo delle regole.

Il modello appena proposto si è dimostrato pratico ed efficace per il raggiungimento degli obbiettivi anche se si è resa evidente la mancanza di automazione nel processo di apprendimento delle regole di estrazione. Tale limite era comunque stato previsto dalle ipotesi iniziali, dopotutto l'implementazione di un sistema basato su regole e a codifica manuale era quello che meglio si adattava alla mia scarsa esperienza nell'ambito delle applicazioni *NLP* ed ha permesso una maggiore comprensione dei problemi relativi all'analisi del linguaggio naturale fornendo inoltre un controllo diretto dei risultati ottenuti.

4.5 Strumenti utilizzati per lo sviluppo delle applicazioni

All'interno di questo progetto sono stati utilizzi diversi strumenti software e linguaggi di programmazione. La motivazione è legata al fatto che in fase di sviluppo sono state fatte differenti scelte metodologiche per verificare quali meglio si adattassero agli scopi del progetto. Inizialmente ad esempio si era pensato di lavorare solo con gli strumenti linguistici di alto livello di *Expert System* è ciò ha implicato l'utilizzo della tecnologie *Java* e *MySql*, in seguito invece si è deciso di ricercare soluzioni più incisive impiegando le librerie aziendali, che offrono un accesso di basso livello alle tecnologie

linguistiche, e ciò ha reso necessario uno sviluppo basato sui linguaggi *C++, C#* accoppiati a *MS Sql Server*.

4.5.1 Eclipse e la tecnologia Java

Eclipse è un progetto open source legato alla creazione e allo sviluppo di una piattaforma di sviluppo ideata da un consorzio di grandi società quali Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP e Serena Software, chiamato Eclipse Foundation.

Pur essendo orientato allo sviluppo del progetto stesso, questo IDE[44] è utilizzato anche per la produzione di software di vario genere. Si passa infatti da un completo IDE per il linguaggio Java ad un ambiente di sviluppo per il linguaggio C++ e a plugin che permettono di gestire XML, PHP e persino di progettare graficamente una GUI per un'applicazione JAVA, rendendo di fatto Eclipse un ambiente RAD[45].

Il programma è scritto in linguaggio Java, ma anziché basare la sua GUI su Swing, il toolkit grafico di Sun Microsystems, si appoggia a SWT, librerie di nuova concezione che conferiscono ad *Eclipse* una straordinaria reattività.

La piattaforma di sviluppo è incentrata sull'uso di plugin, componenti software ideati per uno specifico scopo, per esempio la generazione di diagrammi UML. In effetti tutta la piattaforma è un insieme di plugin, versione base compresa, che chiunque può sviluppare e modificare. Nella versione base è possibile programmare in *Java*, usufruendo di comode funzioni di aiuto quali: completamento automatico, suggerimento dei tipi di parametri dei metodi, possibilità di accesso diretto a CVS[46] e riscrittura automatica del codice in caso di cambiamenti nelle classi, funzionalità questa detta di *Refactoring*. Per maggiori informazioni sul progetto *Eclipse* e per il download dei sorgenti visitare il link *http://www.eclipse.org*.

Java è un linguaggio di programmazione orientato agli oggetti, derivato dallo Smalltalk e creato da James Gosling e altri ingegneri di Sun Microsystems. Lo sviluppo di Java iniziò nel 1991 e i motivi principali per cui venne creato erano quelli di fornire un linguaggio di programmazione:

- ✓ completamente orientato agli oggetti;

[44] Integrated development environment

[45] Rapid Application Development

[46] Concurrent Versions System

- ✓ indipendente dalla piattaforma;
- ✓ contenente strumenti e librerie per il networking;
- ✓ progettato per eseguire codice da sorgenti remote in modo sicuro.

Data la diffusione mondiale della tecnologia *Java* non si proseguirà nella sua descrizione e si rimanda al lettore la possibilità di reperire ulteriori informazioni sulle sue funzionalità e modalità di impiego.

4.5.2 Visual Studio 2005 e MS Sql Server 2005 Express

Visual Studio 2005 è un ambiente di sviluppo integrato sviluppato da Microsoft, che supporta diversi tipi di linguaggio, quali *C++, C#, J#, Visual Basic .Net* e *ASP .Net*, e che permette la realizzazione di applicazioni, siti web, applicazioni web e servizi web.
È inoltre un RAD, ovvero un'applicazione atta ad aumentare la produttività aiutando il programmatore con mezzi come l'IntelliSense o un designer visuale delle forms.
Visual Studio 2005 è inoltre multipiattaforma: con esso è possibile realizzare programmi per server, workstation, pocket PC, smartphone e, naturalmente, per i browser. A differenza dei compilatori classici, quello disponibile col .NET Framework converte il codice sorgente in codice *IL* (Intermediate Language).
IL è un nuovo linguaggio progettato per essere convertito in modo efficiente in codice macchina nativo su differenti tipi di dispositivi. Intermediate Language è un linguaggio di livello più basso rispetto a Visual Basic o C#, ma è a un livello di astrazione più alto rispetto ai linguaggi *assembly* o linguaggi macchina.
Microsoft mette a disposizione un versione gratuita di questo prodotto, tale versione e denominata *Express Edition* e possiede le funzionalità base generalmente sufficienti per gestire la maggior parte dei progetti.
Questo ambiente di sviluppo è stato utilizzato sia per la programmazione in C++ che per la programmazione in C#. Il primo linguaggio è stato impiegato per la realizzazione del *plugin di post-processing* integrato all'interno del *server linguistico* del progetto. Tale scelta è scelta è stata obbligata dal fatto che le librerie aziendali sono state implementate con lo stesso linguaggio e compilate come *dll*[47].
Per l'utilizzo delle librerie linguistiche aziendali ho dovuto approfondire le mie conoscenze per quanto riguarda gli aspetti più caratteristici della programmazione in

[47]Dynamic Linked Library

C++ e di seguito vengono riportate le nozioni più interessanti che si sono dimostrate di estrema utilità.

La *Standard Template Library*[48] (STL) è una libreria software inclusa nella libreria standard del linguaggio C++ e definisce strutture dati generiche, iteratori e algoritmi generici. La Standard Template Library costituisce uno strato software ormai divenuto fondamentale per i programmatori C++, cui fornisce un set precostituito di classi comuni, come container e array associativi, che hanno la caratteristica di poter operare con qualsiasi tipo di dato, sia primitivo che definito dall'utente, richiedendo allo sviluppatore il rispetto di pochi vincoli, come ad esempio l'implementazione di operatori o funzioni di assegnamento o confronto, offrendogli in cambio classi complete di tutte le funzioni e operazioni elementari: copia, assegnamento, inserimento/rimozione, iterazione tra gli elementi, ecc... Inoltre STL fornisce numerosi algoritmi per eseguire operazioni come la ricerca e l'ordinamento all'interno di strutture dati come vettori, liste, insiemi e mappe.

Le *Librerie C++ Boost*[49] sono una collezione di librerie open source che estendono le funzionalità del C++. Molte di esse sono licenziate sotto la *Boost Software License*, designata per permette alle *Boost* di essere usate sia in progetti open sia in progetti closed source. Diverse librerie *Boost* sono ormai diventato uno standard *de facto* nella comunità che si occupa dell'evoluzione del linguaggio C++ e sono state accettate per l'incorporazione all'interno delle prossime versioni base della tecnologia C++.

Per assicurare efficienza e flessibilità, *Boost* fa un estensivo utilizzo della programmazione basata su template, e quindi sulla programmazione generica e metaprogrammazione. Le librerie sono destinate ad una vasta gamma di utenti C++ e campi di applicazione, come la gestione del filesystem, delle espressioni *regex*, dell'algebra lineare, del multithreading, delle immagini, e così via, contando circa 80 librerie individuali.

Durante il progetto si è dovuto frequentemente lavorare con strutture dati *XML* e si è notato un basso supporto per la gestione di tale formato da parte delle librerie base del linguaggio C++. A tale proposito sono state utilizzate le librerie *CMarkup*[50], la cui versione base viene rilasciata gratuitamente e fornisce utili funzionalità per il *parsing* e

[48] http://www.sgi.com/tech/stl
[49] http://www.boost.org
[50] http://www.firstobject.com

la navigazione controllata di una struttura *XML*. Tali librerie non hanno un supporto a linguaggi di *XML query* come ad esempio *XPath*, permettono tuttavia una comoda lettura, generazione e gestione dei file *XML*, aspetto che manca totalmente nelle attuali versioni base di C++.

Per l'aspetto di memorizzazione dei dati, visto il principale impiego di linguaggi della famiglia Microsoft si è deciso di utilizzare la piattaforma *Ms Sql Server 2005 Express Edition*, disponibile gratuitamente con l'unica limitazione sulla dimensione massima raggiungibile da un database di 4096 MB. *Microsoft SQL Server* è un *DBMS*[51] relazionale iniziato a sviluppare nel 1989, inizialmente impiegato per basi dati medio-piccole, ma a partire dalla versione 2000 è stato utilizzato anche per la gestione di basi dati di grandi dimensioni. *SQL Server Express* è inoltre dotato di potenti funzionalità, ad esempio *SQL Server Management Studio Express* è una comoda interfaccia grafica che permette una gestione avanzata e semplificata dell'intero *DMBS*. Tale tool permette di eseguire tutte le operazioni di gestione normalmente effettuabili in linguaggio *SQL*, dalle creazione dei database, alla gestione degli utenti e dalla visualizzazione dei dati all'esecuzione di operazioni di backup e restore sui db.

SQL (*Structured Query Language*) è un linguaggio strutturato sviluppato nel 1974 nei laboratori IBM per l'accesso alle informazioni memorizzate in una base di dati. Inizialmente chiamato *SEQUEL*, il nome fu poi cambiato in *SQL* nel 1977 per motivi legali, e nel 1983, con il rilascio da parte di IBM del DBMS relazione *DB2*, *SQL* divenne lo standard *de facto* (e *de iure* a partire dal 1986 con la standardizzazione ANSI, e in seguito ISO, con le varie standardizzazioni SQL86, SQL92 ed SQL2003) per la gestione dei dati memorizzati su una qualsiasi base di dati relazionale. L'obiettivo della standardizzazione ISO era quello di creare un linguaggio per la manipolazione di query che funzionasse su qualsiasi *DBMS* relazionale. Quest'obiettivo purtroppo non è mai stato raggiunto, in quanto, nonostante i vari produttori di *DBMS* abbiano adottato la base del linguaggio, di fatto ogni produttore ha poi elaborato un proprio "dialetto" *SQL*.

Non si andrà maggiormente nello specifico della descrizione delle tecnologie *Microsoft* utilizzate dato che il lettore può approfondire l'argomento grazie numerose fonti che trattano l'argomento.

[51] Database Management System

4.6 Implementazione del sistema *OKKAM-POP*

Il sistema di estrazione *OKKAM-POP*, come mostrato nello schema concettuale presentato in Figura 4.5 del paragrafo 4.4, è costituito da un insieme di sotto-applicazioni, ciascuna adibita ad uno scopo preciso. In totale sono state sviluppate due *console application* ETL, una *dll*, che costituisce il *plugin di post-processing* integrato all'interno del server linguistico, adibito all'estrazione delle informazioni, e infine è stata sviluppata una *window application, OKKAM-POP GUI,* che permette di visualizzare i risultati ottenuti e memorizzati all'interno di un'apposita struttura db.

4.6.1 L'applicazione ETL *XMLtoTXT*

Questa semplice applicazione è coinvolta nella fase iniziale di preparazione dei dati per la successiva analisi linguistica. *XMLtoTXT* è stato implementato con C#, scelta fatta per il comodo supporto che le librerie di questo linguaggio forniscono per l'analisi dei documenti *XML* attraverso *XPath*[52]. Come spiegato in precedenza i documenti originali dai quali devono essere estrapolati gli estratti giornalistici sono strutturati secondo lo standard *NewsML* dell'*IPTC*. Di seguito viene fornita un'istanza *XML* di questo standard, evidenziando in giallo i dati di interesse che vengono estratti dall'applicazione *XMLtoTXT*.

```
<?xml version="1.0" encoding="utf-8" ?>
<newsItem standard="NewsML-G2" standardversion="2.1"
conformance="power" guid="tag:bdm.it,2007-01-
01:00361:e419989b37c40b7a34df0e949b87c7bb"
version="1" xml:lang="it" xmlns:a="http://bdmit/NewsML-G2/ns/content"
xmlns="http://iptc.org/std/nar/2006-10-01/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://iptc.org/std/nar/2006-10-01/ xsd\NewsML-
G2_2.0-spec-NewsItem-Power.xsd">
    <catalogRef href="http://www.iptc.org/std/catalog/catalog.IPTC-G2-
    Standards_3.xml" />
    <catalogRef href="http://www.news.it/newsml/catalog-2008-04-
```

[52] XPath è un linguaggio parte della famiglia XML che permette di individuare i nodi all'interno di un documento XML. Le espressioni XPath, a differenza delle espressioni XML, non servono a identificare la struttura di un documento, bensì a localizzarne con precisione i nodi.

```
30.xml" />
<itemMeta>
    <itemClass qcode="ninat:text" />
    <versionCreated>2007-01-01T17:07:00+02:00</versionCreated>
    <pubStatus qcode="stat:usable" />
    <generator versioninfo="1.0">BDM-TextConverter</generator>
    <signal qcode="pid:GENERALE" />
</itemMeta>
<contentMeta>
    <contentCreated>2007-01-01T17:07:00+02:00</contentCreated>
    <located type="cptype:city" qcode="city:GAZA">
        <broader type="cptype:geoarea" qcode="dea-geoarea:EGITTO
        (ISRAELE) AFRICA" />
    </located>
    <infoSource literal="NA" />
    <creator literal="XBU" />
    <language tag="it" />
    <subject type="cptype:category" qcode="dea-category:ESTERO" />
    <slugline>MO</slugline>
    <headline>MO: GAZA; FOTOGRAFO RAPITO, SCATTANO LE
    RICERCHE </headline>
</contentMeta>
<contentSet>
    <inlineXML contenttype="application/nitf+xml" xml:lang="it"
    rendition="rnd:web">
        <nitf xmlns="http://iptc.org/std/NITF/2006-10-18/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://iptc.org/std/NITF/2006-10-18/
        nitf-3-4.xsd">
            <body>
                <body.content>
                    <block>
                        <key-list>
                            <keyword key="MO" />
                        </key-list>
                        <byline>
                            <person value="XBU" />
```

```
            </byline>
            <dateline>
                <location>GAZA</location>
                <story.date norm="20070101T1707Z" />
            </dateline>
            <abstract>
                <hl2>MO: GAZA; FOTOGRAFO RAPITO,
                SCATTANO LE RICERCHE</hl2>
                <hl3>GAZA, 1 GEN - Il presidente
                palestinese Abu Mazen ha</hl3>
            </abstract>
            <p>Il presidente
palestinese Abu Mazen ha ordinato a tutti i
servizi di sicurezza di lanciarsi alla ricerca
dei sequestratori del fotoreporter peruviano
Jaime Razuri. Lo riferiscono fonti dell'Anp.
Razuri lavora per la agenzia di stampa
francese Afp.
            </p>
        </block>
    </body.content>
</body>
</nitf>
</inlineXML>
</contentSet>
</newsItem>
```

Come si può notare i metadati associati a ciascun articolo non vengono considerati nell'ambito di questo progetto. Ciò è stato scelto di proposito dal momento si intende connotare *OKKAM-POP* con caratteristiche che siano il più generalizzate possibile. L'utilizzo dei metadati di questa struttura *XML* lo vincolerebbe allo standard *NewsML* mentre l'intento futuro è quello di utilizzare questa applicazione per analizzare anche altre tipologie di testi.

E' opportuno specificare che a volte, anche se raramente all'interno del corpus analizzato, un'unica struttura *NewsML* contiene più di un articolo giornalistico, in

questo caso l'applicazione ETL li scompone e li associa in uscita a due differenti files TXT.

Di seguito, in Figura 4.6, viene mostrato un *activity diagram* che mostra le principali operazioni svolte da *XMLtoTXT*. Essenzialmente *XMLtoTXT* effettua un controllo ricorsivo nel file system a partire dalla cartella principale che racchiude tutto il corpus *XML*.

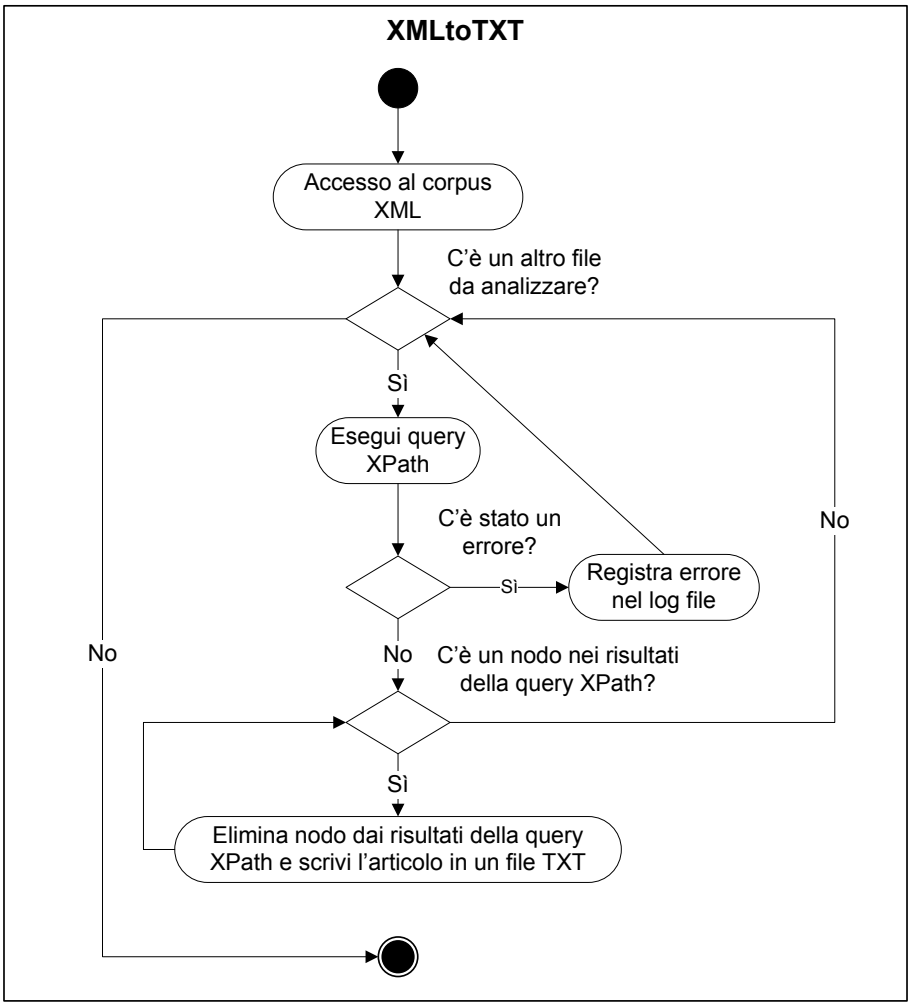

Figura 4.6 – Activity diagram delle operazioni svolte da *XMLtoTXT*

Ad ogni file di estensione *XML* trovato viene applicata una query XPath di questo tipo: *//a:p*, dove *a* rappresenta un prefisso che fa riferimento al seguente namespace *http://iptc.org/std/NITF/2006-10-18/*. In caso errore nell'esecuzione della query, che può ad esempio verificarsi per una malformazione della struttura *XML*, il path assoluto del file che ha provocato tale errore viene registrato nel log file e si prosegue l'analisi del prossimo file XML. Nel caso in cui non vi siano errori nell'esecuzione della *query XPath* si procede al memorizzazione di ciascun nodo *p* all'interno di singoli files TXT.

In conclusione si ottiene un corpus TXT che ha la stessa struttura del corpus XML originale con file che contengono al massimo una news giornalistica. Per completezza si riporta che le librerie del *.NET Framework 2.0* che forniscono un supporto al linguaggio *XPath* o all'*XML* in generale si trovano all'interno del namespace *System.Xml* e *System.Xml.XPath* e in dettaglio le classi utilizzate sono: *XPathNavigator*, *XPathDocument*, *XPathNodeIterator* e *XmlNamespaceManager*.

4.6.2 Il client e il server linguistico di *OKKAM-POP*

La fase di analisi linguistica e di *Information Extraction*, come specificato nel paragrafo 4.3 è composta da due applicazioni, o, per essere ancora più precisi, viene realizzata tramite l'interazione fra due differenti componenti, il *client linguistico* e il *server linguistico*.

Il *client linguistico* è stato implementato con il linguaggio *Java* e la motivazione è legata al fatto tale componente è stata una delle prime ad essere sviluppata, quando ancora il progetto prevedeva di dover utilizzare come base di partenza un ulteriore progetto in fase di sviluppo all'interno dell'azienda *Expert System*.

In seguito tale prospettiva è stata abbandonata, essendomi stato affidato il compito di progettare un apposito *server linguistico*, tuttavia il *client linguistico* già implementato risultava essere ancora utilizzabile ed è stato quindi mantenuto nella sua forma originale. Il *client linguistico* è strutturato secondo il paradigma *multithreading* per due motivazioni essenziali. La prima riguarda la possibilità di poter effettuare richieste multiple di analisi linguistica contemporaneamente attraverso un'unica istanza di un *client linguistico* e la seconda invece è inerente all'efficienza della gestione della memoria allocabile dinamicamente da un'applicazione *Java*. Per non incorrere infatti in problemi legati al superamento dei limiti di memoria utilizzabili dalla *Java Virtual Machine*, errore noto come *OutOfMemoryError*, si è fatto uso di una lista finita per implementare una risorsa condivisa sincronizzata atta a risolvere un problema simile a quello noto nell'ambito informatico come *problema del produttore/consumatore*, classico esempio si sincronizzazione fra processi.

Di seguito, in Figura 4.7, mostrato un *class diagram* che rappresenta le classi principali del *client linguistico*. Si è deciso di utilizzare questa rappresentazione perché risulta più semplice rispetto ad un *activity diagram* e permette inoltre di evidenziare in modo più

evidente le relazioni di cardinalità fra i diversi *thread* che compongono l'applicazione in fase di esecuzione. Il *client linguistico* è configurabile attraverso un file *XML* che viene letto all'avvio dell'applicazione. I principali parametri configurabili sono il numero di thread della classe *OkkamExp*, ovvero il numero di thread che effettuano richieste simultanee verso i server linguistici e la dimensione della risorsa sincronizzata *SynchronousSource* che contiene la lista di file TXT da cui recuperare le news.

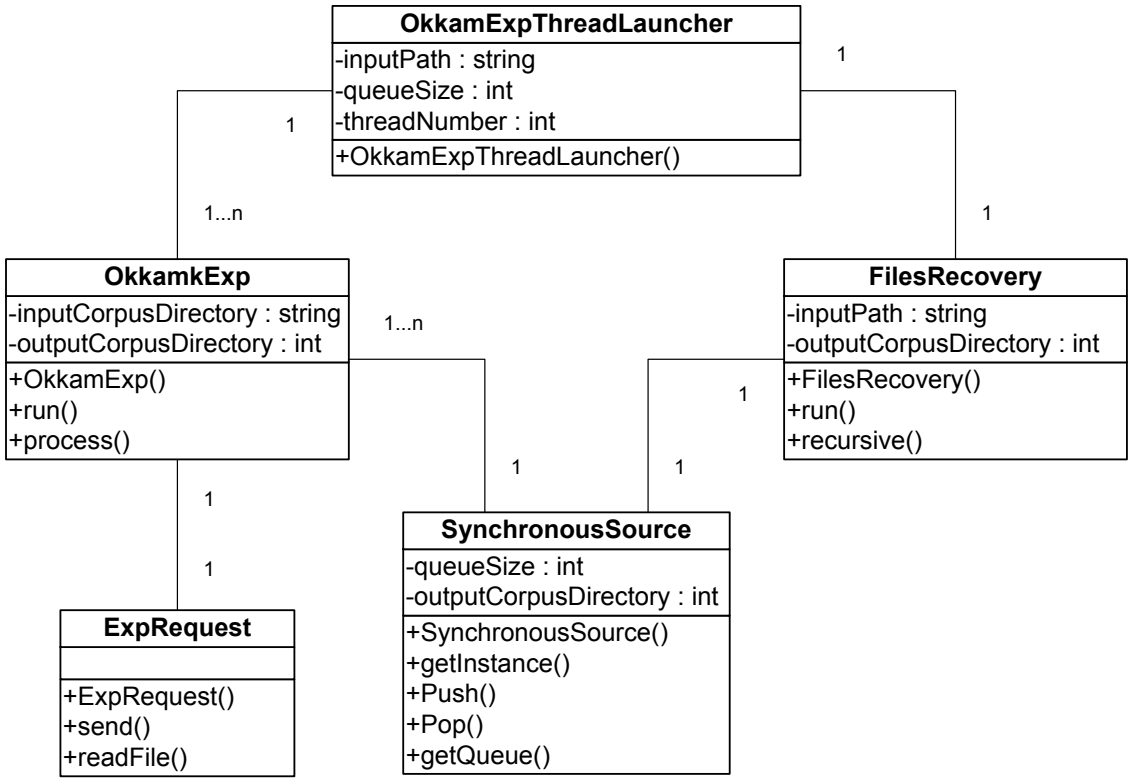

Figura 4.7 – Class Diagram del client linguistico

Il *class diagram* mostrato in Figura 4.7 è una versione semplificata rispetto a quello proposto dal linguaggio UML[53], non vengono infatti mostrati né i parametri dei metodi nè i loro valori di ritorno. La classe *OkkamExpThreadLauncher* rappresenta l'entry point dell'applicazione e si occupa essenzialmente di leggere il file di configurazione, istanziare un certo numero di thread che eseguono il metodo *run* della classe *OkkamExp* e istanziare un thread che esegue il metodo *run* della classe *FilesRecovery*. Quest'ultima implementa un algoritmo ricorsivo che recupera i nomi dei files TXT da analizzare inserendoli, con il metodo *pop* della classe *SynchronousSource*, all'interno della lista condivisa. *SynchronousSource* viene gestita come una classe *singleton*, ovvero come

[53] Unified Modeling Language

una classe che può essere istanziata una ed una sola volta all'interno dell'intera applicazione. I metodi *push e pop* della classe *SynchronousSource* sono naturalmente metodi sincronizzati, ovvero metodi il cui codice è accessibile da un solo thread alla volta. Ciascun thread che rappresenta un'istanza della classe *OkkamExp* si occupa di recuperare il path assoluto di un file estraendolo dalla lista sincronizzata attraverso il metodo *pop* e di inviare il testo contenuto all'interno di tale file al *server linguistico* tramite le funzionalità esposte dalla classe ausiliaria *ExpRequest*. Una volta terminata l'analisi linguistica la risposta viene memorizzata all'interno di un file *XML*. Tale processo viene iterato fino a che tutti i file presenti nel corpus TXT non sono stati processati.

Si passa ora alla descrizione del *server linguistico*, o meglio alla descrizione del *plugin di post-processing* integrato all'interno del *server linguistico*. Tale plugin è stato implementato in C++ dovendo fare largo uso delle librerie aziendali realizzate con questo linguaggio di programmazione ed è stato compilato come DLL. In informatica, una *dynamic-link library* è una libreria software che viene caricata dinamicamente in fase di esecuzione, invece di essere collegata staticamente ad un eseguibile in fase di compilazione. Queste librerie sono note appunto con l'acronimo DLL, che è l'estensione del file che hanno nel sistema operativo Microsoft Windows, o anche con il termine librerie condivise, da *shared library*, usato nella letteratura dei sistemi Unix.

All'interno di un *server linguistico* Expert System un *plugin di post-processing* ha il completo controllo sia delle informazioni linguistiche che derivano dalla disambiguazione sia dell'output che viene ritornato ai *client linguistici*, come mostrato in Figura 4.3. In particolare l'elaborazione della DLL si basa sui seguenti dati in ingresso: l'output standard di un server linguistico e la struttura ad albero che rappresenta un testo disambiguato. A partire da questi dati il plugin effettua tutte le operazioni di *Information Extraction* basandosi su regole che verranno adeguatamente descritte nel prossimo paragrafo. Il plugin è strutturato in classi e di seguito vengono mostrate quelle principali coinvolte nelle operazioni di estrazione delle informazioni. In Figura 4.8 viene inoltre evidenziato il flusso dei dati e l'interazione fra le diverse classi. Non vengono visualizzate in questo schema tutte quelle classi che forniscono funzionalità di supporto come ad esempio le utilità di gestione del file di log, di lettura del file di configurazione e di lettura e scrittura di strutture *XML*.

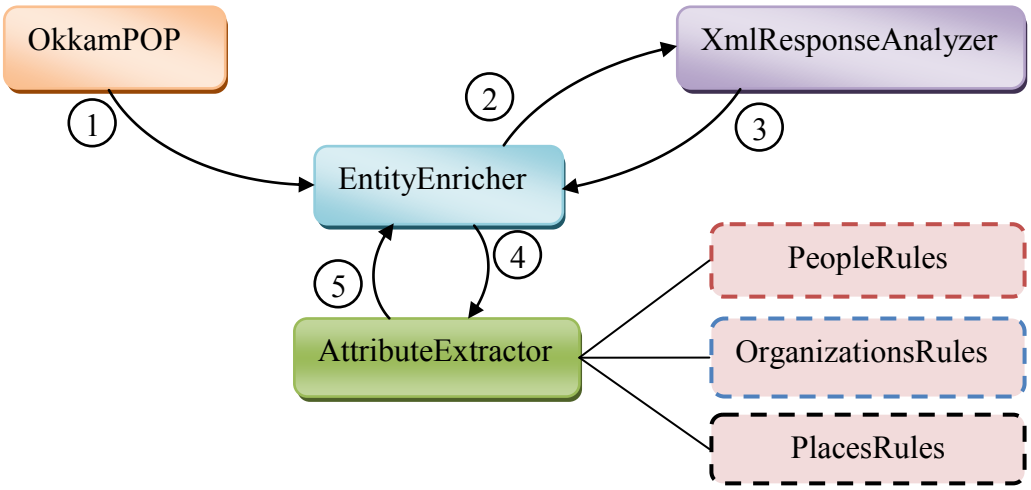

Figura 4.8 – Schema delle classi principali del plugin di post-processing

La classe *OkkamPOP* rappresenta l'entry poi della DLL e ha il compito di effettuare le operazioni di inizializzazione come ad esempio la lettura del file di configurazione.
Successivamente (*1*) il controllo passa alla classe *EntityEnricher* che è la più importante a livello funzionale di tutta la DLL. Al suo interno vengono svolte due operazioni molto rilevanti, la navigazione dell'albero che rappresenta il testo disambiguato e la formattazione dello schema *XML* che viene restituito in output.
Prima di tutto l'istanza della classe *EntityEnricher* richiama (*2*) le funzionalità della classe *XMLResponseAnalyzer*. Come detto in precedenza uno dei dati in ingresso alla DLL è l'output XML standard fornito da un server linguistico privo di *plugin di post-processing*. L'istanza della classe *XMLResponseAnalyzer* si occupa appunto di analizzare tale output XML standard per recuperare al suo interno le seguenti informazioni linguistiche: tutte le entità persona, organizzazione e luogo riscontrate e tutte le triplette *Soggetto*, *Azione*, *Oggetto (SAO)* presenti all'interno del testo analizzato. Tali informazioni vengono organizzate all'interno di una struttura dati appositamente realizzata e restituite (*3*) all'istanza della classe *EntityEnricher*.
Il motivo per cui l'estrazione delle entità viene effettuato tramite l'analisi dell'output XML standard è giustificato dal fatto che in questo modo è possibile sfruttare gli algoritmi e le regole linguistiche implementate all'interno di un *server linguistico* standard. Infatti tramite la semplice analisi della struttura ad albero, effettuata dalla classe *EntityEnricher*, è possibile riconoscere facilmente quali vocaboli rappresentano

un nome proprio di persona (NPH) o un nome proprio generico (NPR), ma non esiste una descrizione ulteriormente dettagliata. Occorrerebbe quindi implementare un algoritmo che riesca a comprendere quali nomi propri si riferiscono a un luogo e quali nomi propri si riferiscono a un'organizzazione. Ma dato che tali caratteristiche sono già implementate all'interno della tecnologia *COGITO*® si è deciso di utilizzarle con le funzionalità sviluppate all'interno della classe *XMLResponseAnalyzer*.

Continuando con il flusso informativo della DLL successivamente l'istanza della classe *EntityEnricher* si occupa di percorrere in modo iterativo la struttura ad albero del testo disambiguato alla ricerca di tutte le entità. Una volta identificata un'entità a livello del nodo "parola" (Figura 4.2) dell'albero si attiva il processo vero e proprio di *Information Extraction*.

A questo punto il controllo passa (*4*) all'istanza della classe *AttributeExtractor* le cui principali funzionalità riguardano la valutazione dell'intorno dell'entità in cui vengono applicate le regole di estrazione e la giusta invocazione delle regole a seconda che si tratti di un'entità persona, organizzazione o luogo.

Dopo l'applicazione delle regole di estrazione implementate all'interno di una delle classi *PeopleRules*, *OrganizationsRules* e *PlacesRules* viene restituita (*5*) all'istanza della classe *EntityEnricher* una struttura dati che contiene tutte le estrazioni effettuate.

Le successive fasi di elaborazione sono rappresentate essenzialmente da una riorganizzazione dei contenuti dell'output di cui ne viene fornito un esempio di seguito.

La risposta linguistica proposta corrisponde all'analisi della seguente frase "Michele Vitali, sviluppatore software, va in pausa pranzo."

```
?xml version="1.0" encoding="Windows-1252" ?>
<RISPOSTA>
   <ESTRAZIONI TIPO="PEOPLE">
      <ENTITA NOME="Michele Vitali">
         <APPOSIZIONI>
            <APPOSIZIONE REGOLA="people_NPH_PNT_SOS"
            VALORE="sviluppatore software">
               <TOKEN NOME="sviluppatore software" TIPO="SOS" />
            </APPOSIZIONE>
         </APPOSIZIONI>
         <SAO>
```

```
            <RECORD MODELLO="SVO">
              <FIELD NOME="S" VALORE="Michele Vitali">
              </FIELD>
              <FIELD NOME="V" VALORE="andare">
              </FIELD>
            </RECORD>
          </SAO>
        </ENTITA>
    </ESTRAZIONI>
    <INFO TIPO="MAINLEMMAS">
      <LEMMA NOME="Michele Vitali" SCORE="39.4" />
    </INFO>
    <INFO TIPO="DOMAINS" />
    <INFO TIPO="CONTEXTS">
      <ENTITA NOME="Michele Vitali">
        <CONTESTO NOME="sviluppatore software" />
        <CONTESTO NOME="pausa pranzo" />
      </ENTITA>
    </INFO>
</RISPOSTA>
```

Sono state evidenziate con diversi colori le principali parti che compongono un risposta del server linguistico. Nella parte in giallo, corrispondente al contenuto del TAG *APPOSIZIONI*, sono presenti tutte le estrazioni effettuate dal *plugin di post-processing* attraverso le regole da me implementate. La struttura di tali regole verrà spiegata adeguatamente nel prossimo paragrafo.

Nella parte in verde, corrispondente al contenuto del TAG *SAO*, sono presenti tutte le triplette *soggetto, azione, oggetto* riscontrate nel testo.

Infine nelle ultime sezioni evidenziate sono contenute informazioni che riguardano rispettivamente i lemmi principali, i domini, e i contesti del testo analizzato. I concetti di lemma principale e dominio risultano abbastanza chiari anche se in questo caso non viene riscontrato nessun dominio data la sintesi del testo analizzato. Richiede invece una spiegazione il concetto di contesto. Viene interpretato come contesto di un'entità qualsiasi sostantivo che compare nella stessa frase dell'entità.

Il risultato della fase di analisi linguistica e *Information Extraction* dell'applicazione *OKKAM-POP* è un corpus di documenti strutturati con lo schema XML appena descritto.

Si è parlato precedentemente della possibilità di impostare alcuni parametri della DLL attraverso file di configurazione, di seguito vengono riportati i valori configurabili:

- ✓ *okkam.mode.verbose*: con questo parametro, costituito da un valore booleano, è possibile impostare due differenti modalità di funzionamento della DLL, una con output sintetico una con output più dettagliato.
- ✓ *okkam.AttributeExtractor.halfWindow*: permette di impostare la dimensione dell'intorno dell'entità a cui applicare le regole di estrazione. In realtà questo parametro permette di impostare il numero di parole precedenti e il numero di parole successive che vengono considerate rispetto all'entità, quindi nell'esempio mostrato un valore del parametro impostato a 5 indica una dimensione reale della finestra di controllo di 10 parole, escludendo l'entità.
- ✓ *okkam.AttributeExtractor.maxDeep*: rappresenta un valore di profondità per il controllo delle relazioni all'interno del *SENSIGRAFO*®.
- ✓ *okkam.AttributeExtractor.People.RightSynconList*: lista di interi che identifica i Synset correlati al concetto di uomo.
- ✓ *okkam.AttributeExtractor.People.WrongSynconList*: lista di interi che identifica i Synset che si vogliono esplicitamente eliminare per il controllo della correlazione con il concetto uomo.
- ✓ *okkam.AttributeExtractor.Places.RightSynconList*: lista di interi che identifica i Synset correlati al concetto di luogo.
- ✓ *okkam.AttributeExtractor.Places.WrongSynconList*: lista di interi che identifica i Synset che si vogliono esplicitamente eliminare per il controllo della correlazione con il concetto luogo.
- ✓ *okkam.AttributeExtractor.Organizations.RightSynconList*: lista di interi che identifica i Synset correlati al concetto di organizzazione.
- ✓ *okkam.AttributeExtractor.Organizations.WrongSynconList*: lista di interi che identifica i Synset che si vogliono esplicitamente eliminare per il controllo della correlazione con il concetto organizzazione.

Il senso degli ultimi sette parametri potrà essere maggiormente chiarito con la lettura del prossimo paragrafo.

4.6.3 Implementazione delle regole di estrazione

L'aspetto sicuramente più complicato di questo progetto, una volta decisa l'impostazione generale per il sistema di estrazione, è stato trovare un metodo efficace e preciso per intercettare all'interno di un testo, scritto in linguaggio naturale, tutte quelle descrizioni e concetti che possono caratterizzare bene un'entità. Si ribadisce che il sistema *OKKAM-POP* si basa su un modello a codifica manuale che fa uso di regole. L'aspetto della codifica manuale ha innanzitutto richiesto uno studio approfondito del dominio, che è stato basato su numerose analisi preliminari di dati estratti dai testi, per verificare se vi fosse un qualche fattore comune a tutte quelle porzioni di testo che rappresentano apposizioni delle entità e concetti descrittivi.
Da subito l'attenzione è stata attratta da un'analisi specifica sulla struttura sintattica del testo. Per chiarire meglio questo concetto si veda il seguente esempio di analisi della struttura sintattica di un testo.

> *"Il[**ART**] presidente[**SOS**] della[**PRE**] Commissione[**SOS**] di[**PRE**] Vigilanza[**SOS**] RAI[**NPR**],[**PNT**] Sergio Zavoli[**NPH**], incontrerà[**VER**] Paolo Romani[**NPH**], [**PNT**] Viceministro[**SOS**] alle[**PRE**] Comunicazioni[**SOS**], presso[**PRE**] palazzo[**SOS**] Grazioli[**NPR**] a[**PRE**] Roma[**NPR**] per[**PRE**] esporre[**VER**] il[**ART**] programma[**SOS**] di[**PRE**] sviluppo[**SOS**]. [**PNT**]"*

Come si può notare vi sono degli specifici pattern sintattici che precedono o seguono le entità che sono riconducibili a diffusi modelli sintattici usati per costruire apposizioni.
Ad esempio *ART|SOS|PRE|SOS|PRE|SOS|NPR|PNT* può essere riconosciuto come pattern sintattico che descrive l'apposizione "Il presidente della commissione di Vigilanza RAI," riferita all'entità persona "Sergio Zavoli". Allo stesso modo *SOS|PRE|SOS* è il pattern sintattico che descrive "Viceministro alle Comunicazioni" riferito a "Paolo Romani" e *PRE|NPR* descrive "a Roma" che localizza "Grazioli".

Questo rappresenta un primo metodo relativamente semplice per filtrare parti di testo che possono rappresentare il *target* delle estrazioni.
Per rilevare in modo semiautomatico una discreta quantità di pattern sintattici utili allo scopo si è proceduto inizialmente a un estrazione di tutte le 10 parole, punteggiatura esclusa, che precedono le entità e di tutte le 10 parole che le seguono. Naturalmente sono stati rispettati in tale estrazione i vincoli di inizio frase e fine frase dal momento che non avrebbe senso considerare vocaboli che appartengono alla frase precedente o alla frase successiva. In seguito su tali dati è stata effettuata un analisi statistica basata sia sulla frequenza con cui i pattern si presentano sia sulla capacità che tali pattern hanno nel rappresentare informazioni di interesse.
Tale metodo di estrazione ha però fatto rilevare un problema non sottovalutabile, la presenza di eccessivo rumore all'interno dei dati estratti. Per rumore in questo ambito si intendono tutte quelle estrazioni che rispettano i vincoli finora proposti dalle regole ma che non rispettano il contenuto informativo voluto.
Di seguito viene fornito un esempio con permette di capire meglio la natura del problema.

Si considerino le seguenti frasi:

> *"Il[**ART**] premier[**SOS**] israeliano[**AGG**] Ehud Olmert[**NPH**] ha[**AUX**] respinto[**VER**] le[**ART**] accuse[**SOS**] palestinesi[**AGG**]"*

> *"Nei[**PRE**] giorni[**SOS**] scorsi[**AGG**] Ehud Olmert[**NPH**] ha[**AUX**] respinto[**VER**] le[**ART**] accuse[**SOS**] palestinesi[**AGG**]"*

Fatta eccezione per il tipo grammaticale della prima parola queste due frasi hanno una struttura sintattica identica. Si consideri ora di effettuare un'estrazione secondo le metodologie descritte precedentemente con il seguente pattern sintattico: *SOS|AGG|NPH*. Nel primo caso otteniamo un risultato positivo con l'estrazione di "premier israeliano" riferito a "Ehud Olmert", nel secondo caso invece si ottiene "giorni successivi" riferito a "Ehud Olmert" che non è di nessuna utilità per gli scopi del progetto, anzi contribuisce ad aumentare il rumore dei dati.

Per ovviare a questo problema è stato quindi necessario porre un ulteriore vincolo all'interno delle regole. Dato che la sintassi non può per definizione cogliere il significato dei termini è subito risultato evidente che questo nuovo vincolo dovesse essere di tipo semantico.

A tale scopo sono stati di fondamentale importanza gli strumenti linguistici Expert System. Per implementare infatti questi nuovi vincoli all'interno delle regole è stata utilizzata la rete linguistico-semantica *SENSIGRAFO*®.

Tale rete essendo strutturata, come descritto nei precedenti capitoli, in modo gerarchico permette di verificare se ad esempio un generico sostantivo sia figlio o oppure no del concetto più generale di "uomo".

Effettuando tale controllo per qualsiasi pattern sintattico utilizzato in fase di estrazione è possibile eliminare a priori una notevole quantità di rumore, nell'esempio precedente non verrebbe ad esempio considerato "giorni successivi" riferito a "Ehud Olmert", dato che il sostantivo "giorni" non ha nessuna relazione di iponimia con il concetto "uomo".

Si può cogliere ora il significato dei parametri della DLL non descritti in modo preciso nel precedente paragrafo. *AttributeExtractor.maxDeep* rappresenta la massima distanza, in termini di relazioni nel *SENSIGRAFO*®, che un sostantivo deve avere rispetto ai concetti "uomo", "organizzazione" o "luogo" affinché il pattern all'interno del quale è contenuto possa essere considerato valido in fase di estrazione.

Il parametro *okkam.AttributeExtractor.People.RightSynconList* e gli omologhi per organizzazioni e luoghi rappresentano il synset di riferimento rispetto al quale viene valutata la distanza semantica dei sostantivi.

Il parametro *okkam.AttributeExtractor.People.WrongSynconList* e gli omologhi per organizzazioni e luoghi, che non sono stati utilizzati nella versione finale di *OKKAM-POP*, rappresentano invece synset di riferimento che non devono essere genitori di un certo sostantivo affinché il pattern che lo contiene venga considerato valido.

Le regole sintattico-semantiche sono state incluse, come mostrato in Figura 4.8, all'interno di tre differenti classi della DLL realizzata e vengono elencate di seguito.

Regole per l'estrazione dei concetti relativi alle persone:

```
/*0*/   People_NPH_SOS
/*1*/            People_NPH_SOS_PRE_SOS
/*1*/            People_NPH_SOS_PRE_NPR
/*1*/            People_NPH_SOS_AGG
/*2*/                     People_NPH_SOS_AGG_PRE_SOS
/*2*/                     People_NPH_SOS_AGG_PRE_NPR
```

```
/*0*/   People_NPH_PNT_SOS
/*1*/            People_NPH_PNT_SOS_PRE_SOS
/*1*/            People_NPH_PNT_SOS_PRE_NPR
/*1*/            People_NPH_PNT_SOS_AGG
/*2*/                     People_NPH_PNT_SOS_AGG_PRE_SOS
/*3*/                              People_NPH_PNT_SOS_AGG_PRE_SOS_PRE_NPR
/*3*/                              People_NPH_PNT_SOS_AGG_PRE_SOS_PRE_SOS
/*2*/                     People_NPH_PNT_SOS_AGG_PRE_NPR
/*3*/                              People_NPH_PNT_SOS_AGG_PRE_NPR_PRE_NPR
/*3*/                              People_NPH_PNT_SOS_AGG_PRE_NPR_PRE_SOS

/*0*/   People_NPH_AGG_SOS
/*1*/            People_NPH_AGG_SOS_AGG
/*2*/                     People_NPH_AGG_SOS_AGG_PRE_NPR
/*2*/                     People_NPH_AGG_SOS_AGG_PRE_SOS
/*1*/            People_NPH_AGG_SOS_PRE_SOS
/*1*/            People_NPH_AGG_SOS_PRE_NPR

/*0*/   People_NPH_PNT_AGG_SOS
/*1*/            People_NPH_PNT_AGG_SOS_AGG
/*2*/                      People_NPH_PNT_AGG_SOS_AGG_PRE_NPR
/*1*/            People_NPH_PNT_AGG_SOS_SOS
/*2*/                     People_NPH_PNT_AGG_SOS_SOS_PRE_NPR
/*1*/            People_NPH_PNT_AGG_SOS_PRE_NPR
/*1*/            People_NPH_PNT_AGG_SOS_PRE_SOS

/*0*/   People_SOS_NPH
/*1*/            People_SOS_PRE_SOS_NPH
/*2*/                     People_ART_SOS_PRE_SOS_NPH
/*2*/                     People_AGG_SOS_PRE_SOS_NPH
/*3*/                     People_ART_AGG_SOS_PRE_SOS_NPH
/*1*/            People_AGG_SOS_NPH
/*2*/                     People_ART_AGG_SOS_NPH
/*1*/            People_ART_SOS_NPH
/*1*/            People_SOS_SOS_NPH
/*2*/                     People_ART_SOS_SOS_NPH

/*0*/   People_SOS_AGG_NPH
/*1*/            People_ART_SOS_AGG_NPH

/*0*/   People_SOS_PRE_NPR_NPH
/*1*/            People_ART_SOS_PRE_NPR_NPH
/*1*/            People_AGG_SOS_PRE_NPR_NPH
/*2*/                     People_ART_AGG_SOS_PRE_NPR_NPH

/*0*/   People_SOS_PRE_NPR_PNT_NPH
/*1*/            People_ART_SOS_PRE_NPR_PNT_NPH
/*1*/            People_AGG_SOS_PRE_NPR_PNT_NPH
/*2*/                     People_ART_AGG_SOS_PRE_NPR_PNT_NPH

/*0*/   People_SOS_PRE_SOS_PNT_NPH
/*1*/            People_ART_SOS_PRE_SOS_PNT_NPH
/*1*/            People_AGG_SOS_PRE_SOS_PNT_NPH
/*2*/                     People_ART_AGG_SOS_PRE_SOS_PNT_NPH
```

Regole per l'estrazione dei concetti relativi alle organizzazioni:

```
/*0*/  Organizations_NPR_PNT_SOS
/*1*/          Organizations_NPR_PNT_SOS_SOS
/*2*/                  Organizations_NPR_PNT_SOS_SOS_PRE_SOS
/*1*/          Organizations_NPR_PNT_SOS_AGG
/*2*/                  Organizations_NPR_PNT_SOS_AGG_AGG
/*3*/                          Organizations_NPR_PNT_SOS_AGG_AGG_AGG_PRE_SOS
/*3*/                          Organizations_NPR_PNT_SOS_AGG_AGG_AGG_PRE_NPR
/*2*/                  Organizations_NPR_PNT_SOS_AGG_PRE_ART_SOS
/*2*/                  Organizations_NPR_PNT_SOS_AGG_PRE_ART_NPR
/*2*/                  Organizations_NPR_PNT_SOS_AGG_PRO_PRO_VER_PRE_SOS
/*2*/                  Organizations_NPR_PNT_SOS_AGG_PRO_PRO_VER_PRE_NPR
/*2*/                  Organizations_NPR_PNT_SOS_AGG_SOS
/*3*/                          Organizations_NPR_PNT_SOS_AGG_SOS_PRE_SOS
/*3*/                          Organizations_NPR_PNT_SOS_AGG_SOS_PRE_NPR
/*1*/          Organizations_NPR_PNT_SOS_PRE_NPR
/*2*/                  Organizations_NPR_PNT_SOS_PRE_NPR_PRE_NPR
/*1*/          Organizations_NPR_PNT_SOS_PRE_NPH
/*1*/          Organizations_NPR_PNT_SOS_PRE_ART_SOS
/*2*/                  Organizations_NPR_PNT_SOS_PRE_ART_SOS_PRE_SOS
/*1*/          Organizations_NPR_PNT_SOS_PRE_AGG_SOS
/*1*/          Organizations_NPR_PNT_SOS_PRE_SOS
/*2*/                  Organizations_NPR_PNT_SOS_PRE_SOS_AGG
/*3*/                          Organizations_NPR_PNT_SOS_PRE_SOS_AGG_NPR
/*3*/                          Organizations_NPR_PNT_SOS_PRE_SOS_AGG_PRE_SOS
/*3*/                          Organizations_NPR_PNT_SOS_PRE_SOS_AGG_PRE_NPR
/*2*/                  Organizations_NPR_PNT_SOS_PRE_SOS_CON_SOS
/*3*/                          Organizations_NPR_PNT_SOS_PRE_SOS_CON_SOS_PRE_SOS
/*2*/                  Organizations_NPR_PNT_SOS_PRE_SOS_NPR
/*2*/                  Organizations_NPR_PNT_SOS_PRE_SOS_PRE_NPR
/*2*/                  Organizations_NPR_PNT_SOS_PRE_SOS_PRE_SOS
/*3*/                      Organizations_NPR_PNT_SOS_PRE_SOS_PRE_SOS_AGG
/*3*/                      Organizations_NPR_PNT_SOS_PRE_SOS_PRE_SOS_PRE_SOS
/*3*/                     Organizations_NPR_PNT_SOS_PRE_SOS_PRE_SOS_PRE_NPR
/*3*/                     Organizations_NPR_PNT_SOS_PRE_SOS_PRE_SOS_PRE_SOS_AGG
/*1*/          Organizations_NPR_PNT_SOS_PRO_VER_NPR
/*1*/          Organizations_NPR_PNT_SOS_PRO_VER_SOS
/*1*/          Organizations_NPR_PNT_SOS_PRO_PRO_VER_PRE_SOS
/*1*/          Organizations_NPR_PNT_SOS_PRO_VER_ART_SOS_PRE_SOS
/*1*/          Organizations_NPR_PNT_SOS_PRO_VER_ART_SOS_PRE_NPR
/*1*/          Organizations_NPR_PNT_SOS_PRO_VER_PRE_SOS_PRE_SOS
/*1*/          Organizations_NPR_PNT_SOS_PRO_VER_PRE_SOS_PRE_NPR

/*0*/          Organizations_NPR_PAR_SOS_PAR
```

Regole per l'estrazione dei concetti relativi ai luoghi:

```
/*0*/  Places_NPR_PAR_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_PRE_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_AGG_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_ART_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_SOS_PRE_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_PRE_AGG_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_NPR_PNT_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_PRE_SOS_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PAR_SOS_AGG_PNT_SOS_PRE_NPR_PAR
/*0*/  Places_NPR_PNT_PRE_SOS_PRE_NPR
```

```
/*0*/    Places_NPR_PNT_AGG_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_SOS_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_PRE_AGG_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_ART_AGG_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_ART_SOS_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_PRE_SOS_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_SOS_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_AGG_SOS_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_SOS_PRE_AGG_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_ART_SOS_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_SOS_AGG_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_PRE_SOS_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_ART_SOS_PRE_AGG_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_ART_SOS_AGG_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_ART_AGG_SOS_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_AGG_SOS_SOS_PRE_SOS_PRE_NPR
/*0*/   Places_NPR_PNT_SOS_PRE_SOS_PRE_SOS_PRE_NPR
/*0*/    Places_NPR_PNT_PRE_SOS_PRE_ART_SOS_PRE_NPR
/*0*/   Places_NPR_PNT_PRE_SOS_AGG_PRE_SOS_PRE_NPR
```

Come si può notare le regole costituite dai pattern sintattici sono organizzate in modo gerarchico per evidenziare le dipendenze durante le fasi di analisi linguistica. Se ad esempio durante l'analisi viene individuato un pattern di livello *n*, per ottimizzare il processo, si evita di eseguire l'analisi di tutte le regole di livello inferiore a *n* appartenenti alla stessa gerarchia. Ciò permette di evitare elaborazioni inutili e di non avere in output ridondanze di contenuto informativo inferiore. Se ad esempio il *plugin di post-processing* individua per l'entità "Romano Prodi'' l'apposizione "presidente del partito democratico" si evita di estrarre anche l'apposizione "presidente del partito".

Oltre al metodo di estrazione appena descritto ne è stato realizzato un altro di diversa natura che si basa su una caratteristica intrinseca del linguaggio naturale, ovvero l'espressione di una qualità del soggetto di una frase attraverso l'utilizzo del predicato nominale. Esso è una delle due forme in cui può presentarsi il predicato, l'altra è il predicato verbale. Il predicato nominale può essere espresso in due modi:

- ✓ con il verbo "essere" opportunamente coniugato, che funge da *copula*, e da una *parte nominale*, o *nome del predicato* ovvero un aggettivo, un sostantivo o entrambi.
- ✓ con un verbo *copulativo* (*sembrare*, *parere* e, con alcune sfumature di significato, anche *nascere*, *morire*, ecc…) e un complemento predicativo del soggetto o dell'oggetto che ne completi il significato.

Un esempio del primo modo di formare il predicato nominale è "Il giardiniere è bravo". In questo caso il verbo "è" costituisce la copula e l'aggettivo "bravo" la parte nominale.
Un esempio del secondo modo è "Il giardiniere sembra bravo" o "Il giardiniere nacque povero". In questo caso "sembra" o "nacque" rappresentano il verbo copulativo e "bravo" è il complemento predicativo del soggetto.
Per estrarre i predicati nominali in cui sono coinvolte le entità rilevate nel testo si è fatto uso delle funzionalità di analisi logica esposte dalle librerie aziendali *Expert System*.
Queste funzionalità permettono di recuperare, a livello del nodo "gruppo" della struttura ad albero mostrata in Figura 4.2, alcune delle relazioni logiche che sussistono fra le varie componenti di una frase, fra cui anche le relazione che lega un soggetto e il suo predicato nominale.
I risultati derivanti dalle estrazioni dei predicati nominali sono abbastanza interessanti ma afflitti da notevole rumore.
Un critica che può essere mossa al metodo di estrazione basato sui pattern sintattici e che sia una tecnica eccessivamente manuale, tuttavia ha permesso di ottenere risultati soddisfacenti, come verrà mostrato nel paragrafo finale di questo capitolo, ed è stato scelto perché effettivamente implementabile entro tempi previsti per il progetto.
Inoltre il potenziale delle regole basate su pattern sintattici con vincoli semantici è evidente dato che con esse è possibile estrarre veri e propri concetti e non solo singoli termini che possiedono un certo valore di correlazione con le entità.

4.6.4 L'applicazione ETL *XMLtoDB*

Una volta processate tutte le news giornalistiche presenti nel corpus, che ammontano circa a 1.160.000, si è in presenza di corpus XML arricchito con le informazioni semantiche relative a tutte le entità riscontrate, tuttavia i dati in questo formato non sono comodamente visualizzabili né tanto meno è possibile effettuare efficientemente su di essi ragionamenti statistici. E' necessario quindi esportare i dati all'interno di un database relazionale con indici ben strutturati, data la grande quantità di record, per poter effettuare un'analisi accurata sulle estrazioni.
L'operazione di trasferimento dei dati dal corpus XML al database viene eseguita dall'applicazione ETL *XMLtoDB*. Questa applicazione, sviluppata in C#, è del tutto

simile all'applicazione ETL *XMLtoTXT*, con la differenza che i dati in fase di *load* non vengono memorizzati all'interno del file system ma all'intero di uno schema database implementato con *MS Sql Server 2005 Express*. Lo schema di tale database viene descritto in modo dettagliato di seguito fornendone il modello , la relativa traduzione in linguaggio relazionale e un'esplicitazione in forma tabellare di tutti i campi appartenenti a ciascuna tabella.

Come si può notare in Figura 4.9, lo schema del database contiene due tabelle principali: DOCUMENTS che è in relazione n-aria con le tabelle RELEVANTS_LEMMAS, RELEVANTS_DOMAINS e ENTITIES e la tabella ENTITIES che è in relazione n-aria con le tabelle APPOSITIONS, SAO, PROPS e RELEVANTS_CONTEXTS.

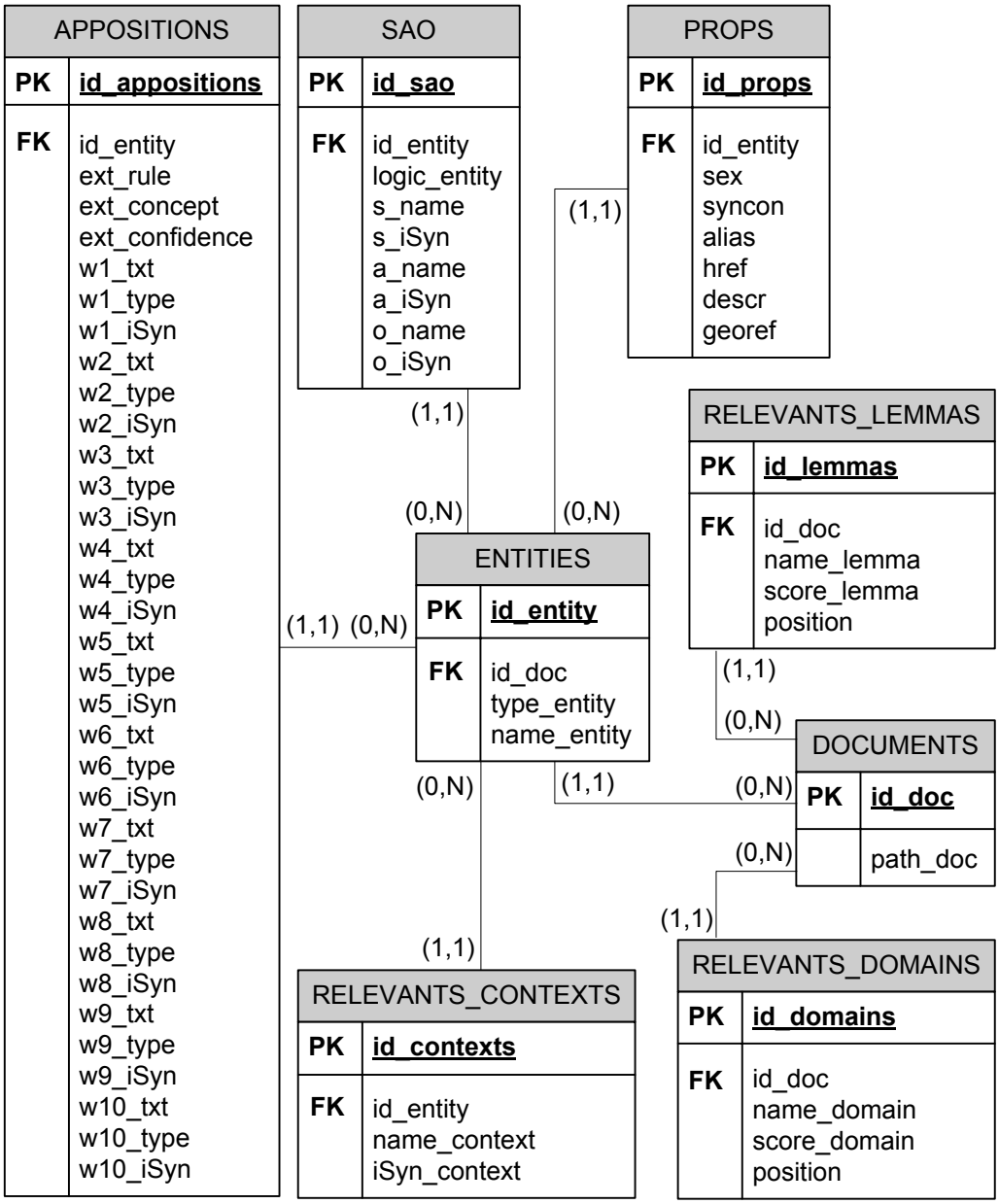

Figura 4.9 – Modello ER dello schema DB primario

Di seguito viene fornita la descrizione relazionale dello schema che permetta di mettere in maggiore evidenza le *foreign key* delle tabelle secondarie.

DOCUMENTS (id_doc, path_doc)

ENTITIES (id_entity, id_doc, type_entity, name_entity)

FK: id_doc REFERENCES DOCUMENTS

APPOSITIONS (id_appositions, id_entity, ext_rule, ext_concept, ext_confidence, w1_txt , w1_type, w1_iSyn, w2_txt , w2_type, w2_iSyn, w3_txt , w3_type, w3_iSyn, w4_txt ,

w4_type, w4_iSyn, w5_txt , w5_type, w5_iSyn, w6_txt , w6_type, w6_iSyn, w7_txt , w7_type, w7_iSyn, w8_txt , w8_type, w8_iSyn, w9_txt , w9_type, w9_iSyn, w10_txt , w10_type, w10_iSyn,)

FK: id_entity REFERENCES ENTITIES

SAO (id_sao, id_entity, logic_entity, s_name, s_iSyn, a_name, a_iSyn, o_name, o_iSyn)

FK: id_entity REFERENCES ENTITIES

PROPS (id_props, id_entity, sex, syncon, alias, href, descry, georef)

FK: id_entity REFERENCES ENTITIES

RELEVANTS_CONTEXTS (id_contexts, id_entity, name_context, iSyn_context)

FK: id_entity REFERENCES ENTITIES

RELEVANTS_LEMMAS (id_lemmas, id_doc, name_lemma, score_lemma, position)

FK: id_doc REFERENCES DOCUMENTS

RELEVANTS_DOMAINS (id_domains, id_doc, name_domain, score_domain, position)

FK: id_doc REFERENCES DOCUMENTS

Viene fornita infine una descrizione più dettagliata delle tabelle e dei campi che forniscono il livello di dettaglio dei *records*. Viene inoltre caratterizzato il tipo di dato, il vincolo di nullità e il tipo di indicizzazione di ciascun campo.

La tabella DOCUMENTS mantiene un riferimento a tutti i documenti presenti all'interno del corpus XML arricchito.

TABELLA DOCUMENTS				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_doc	integer	NOT NULL	CLUSTER	Chiave primaria.
path_doc	varchar(255)	NOT NULL		Path relativo di un documento riferito al path assoluto del corpus.

La tabella ENTITIES contiene tutte le entità persone, organizzazioni e luoghi estratte all'interno dei documenti

TABELLA ENTITIES				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_entity	integer	NOT NULL	CLUSTER	Chiave primaria.
id_doc	integer	NOT NULL		Chiave esterna per riferimento al documento in cui è contenuta l'entità.
type_entity	varchar(15)	NOT NULL		Tipo dell'entità (PEOPLE, ORGANIZATIONS, PLACES).
name_entity	varchar(127)	NOT_NULL	NOT CLUSTER	Testo che rappresenta l'entità.

La tabella APPOSITIONS contiene tutte le estrazioni effettuate tramite le regole implementate all'interno della DLL di post-processing. Tale tabella risulta essere la più vasta in termini di numero dei campi e ciò è dovuto al fatto che per ogni termine contenuto all'interno del concetto estratto vengono memorizzate le seguenti informazioni: testo, tipo grammaticale e Synset. Dopo quanto appena detto potrebbe risultare superflua la memorizzazione del campo *ext_concept*, che contiene appunto il testo del concetto estratto, e che potrebbe essere quindi calcolato con un'operazione di concatenazione dei campi *w1_txt, w2_txt, ..., w10_txt*. Si è deciso tuttavia di mantenerlo dal momento che tale operazione di concatenazione effettuata su un grande numero di record degrada enormemente le prestazioni di navigazione del database. Il fatto che un concetto possa contenere al massimo 10 termini, punteggiatura inclusa, evidenzia la scelta fatta a livello progettuale di attribuire alla finestra di controllo centrata sull'entità un valore di 21 termini, entità inclusa. La scelta di questo valore deriva da studi e prove effettuate preliminarmente all'elaborazione finale del corpus.

TABELLA APPOSITIONS				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_appositions	integer	NOT NULL	CLUSTER	Chiave primaria.
id_entity	integer	NOT NULL	NOT CLUSTER	Chiave esterna per riferimento all'entità descritta dall'estrazione.
ext_rule	varchar(50)	NOT NULL	NOT CLUSTER	Regola che ha prodotto l'estrazione.
ext_concept	varchar(2048)	NOT_NULL		Testo che rappresenta l'estrazione.
ext_confidence	float	NULL		Valore numerico di scoring nel caso in cui l'estrazione sia un predicato nominale.
w1_txt	varchar(50)	NULL		Testo della prima parola dell'estrazione.
w1_type	varchar(5)	NULL		Tipo grammaticale della prima parola dell'estrazione.
w1_iSyn	integer	NULL		Synset della prima parola dell'estrazione.
w2_txt	varchar(50)	NULL		Testo della seconda parola dell'estrazione.
w2_type	varchar(5)	NULL		Tipo grammaticale della seconda parola dell'estrazione.
w2_iSyn	integer	NULL		Synset della seconda parola dell'estrazione.
...	…	…	…	…
w10_txt	varchar(50)	NULL		Testo della decima parola dell'estrazione.
w10_type	varchar(5)	NULL		Tipo grammaticale della decima parola dell'estrazione.

w10_iSyn	integer	NULL		Synset della decima parola dell'estrazione.

La tabella SAO contiene tutte le triplette soggetto, azione, oggetto riscontrate nel testo che includono all'interno del soggetto o dell'oggetto una delle entità estratte nei documenti. Anche in questo caso il campo *logic_entity*, che mantiene il ruolo logico all'interno di ciascuna tripletta dell'entità riferita, potrebbe sembrare superfluo, è stato tuttavia mantenuto perché permette di eseguire un'operazione di *JOIN* in meno fra la tabella SAO e la tabella ENTITIES nel caso in cui si vogliano ad esempio recuperare tutte le entità che compaiono come soggetto o come oggetto all'interno delle triplette SAO.

TABELLA SAO				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_sao	integer	NOT NULL	CLUSTER	Chiave primaria.
id_entity	integer	NOT NULL	NOT CLUSTER	Chiave esterna per riferimento all'entità.
logic_entity	char(1)	NOT NULL	NOT CLUSTER	Carattere settato a 'S' oppure 'O' a seconda che l'entità riferita sia rispettivamente soggetto o oggetto di una tripletta SAO.
s_name	varchar(127)	NULL		Testo che rappresenta il soggetto del SAO.
s_iSyn	integer	NULL		Synset del soggetto del SAO.
a_name	varchar(127)	NULL		Testo che rappresenta l'azione del SAO.
a_iSyn	integer	NULL		Synset dell'azione del SAO.
o_name	varchar(127)	NULL		Testo che rappresenta l'oggetto del SAO.
o_iSyn	integer	NULL		Synset dell'oggetto del SAO.

La tabella PROPS mantiene memorizzate le informazioni linguistiche sulle entità che sono fornite in modo automatico dalla tecnologia linguistica *COGITO*®. Si è deciso di mantenere queste informazioni per poter effettuare un confronto con quelle estratte dal sistema *OKKAM-POP*. Alcuni campi di questa tabella come *synset, alias, descr* e *georef* sono informazioni che vengono recuperate dal *SENSIGRAFO*®, mentre altri come *sex* e *href* vengono estratti con metodi euristici dalla tecnologia linguistica *Expert System*.

TABELLA PROPS				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_props	integer	NOT NULL	CLUSTER	Chiave primaria.
id_entity	integer	NOT NULL	NOT CLUSTER	Chiave esterna per riferimento all'entità.

sex	char(1)	NULL		Carattere settato a 'M' oppure 'F' per rappresentare il sesso dell'entità nel caso in cui sia di tipo PEOPLE
synset	integer	NULL		Synset dell'entità riferita.
alias	varchar(127)	NULL		Testo alternativo per l'entità riferita.
href	varchar(255)	NULL		Testo che caratterizza l'entità riferita.
descr	varchar(255)	NULL		Descrizione dell'entità riferita.
georef	varchar(255)	NULL		Descrizione geografica dell'entità riferita nel caso in cui sia di tipo PLACE.

La tabella RELEVANTS_CONTEXTS contiene tutti i contesti relativi a una certa entità. Come spiegato precedentemente per contesto si intende un qualsiasi sostantivo che sia presente all'interno della stessa frase in cui compare un entità. Queste informazioni non hanno molta rilevanza se non si osservano da un punto di vista statistico. Affinché assumano un senso significativo occorre processare numerosi documenti all'interno dei quali compare una certa entità ed in seguito bisogna filtrare in base alla frequenza i contesti per verificare quali veramente hanno una forte correlazione con l'entità stessa.

TABELLA RELEVANTS_CONTEXTS				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_contexts	integer	NOT NULL	CLUSTER	Chiave primaria.
id_entity	integer	NOT NULL	NOT CLUSTER	Chiave esterna per riferimento all'entità.
name_context	varchar(127)	NOT NULL		Testo che rappresenta il contesto presente all'interno della frase che contiene l'entità riferita.
iSyn_context	integer	NULL		Synset del contesto.

La tabella RELEVANTS_LEMMAS memorizza i lemmi principali riscontrati all'interno dei documenti. A differenza delle tabelle precedenti questa non è posta in relazione con la tabella ENTITIES ma con la tabella DOCUMENTS.

TABELLA RELEVANTS_LEMMAS				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_lemmas	integer	NOT NULL	CLUSTER	Chiave primaria.
id_doc	integer	NOT NULL	NOT CLUSTER	Chiave esterna per riferimento al documento.
name_lemma	varchar(127)	NOT NULL		Testo che rappresenta il lemma presente all'interno del documento riferito.
score_lemma	float	NULL		Valore numero dello score del lemma all'interno del documento riferito.
position	integer	NOT NULL	NOT CLUSTER	Posizione del lemma all'interno

				del documento riferito in base allo score.

Infine la tabella RELEVANTS_DOMAINS memorizza i domini attribuiti a ciascun documento dalla tecnologia Expert System. Queste informazioni assumono un senso importante in virtù della tipologia di testi analizzati in questo progetto, infatti trattandosi di articoli relativamente brevi e monotematici aumenta notevolmente il grado di attinenza fra il dominio attribuito al documento e le entità in esso contenute. Inoltre queste informazioni, come verrà mostrato successivamente, permettono di effettuare un'analisi molto interessanti dal punto di vista dei domini.

TABELLA RELEVANTS_DOMAINS				
CAMPO	**TIPO DATO**	**NULL**	**INDICE**	**DESCRIZIONE**
id_domains	integer	NOT NULL	CLUSTER	Chiave primaria.
id_doc	integer	NOT NULL	NOT CLUSTER	Chiave esterna per riferimento al documento.
name_domain	varchar(127)	NOT NULL		Testo che rappresenta il dominio presente all'interno del documento riferito.
score_domain	float	NULL		Valore numero dello score del dominio all'interno del documento riferito.
position	integer	NOT NULL	NOT CLUSTER	Posizione del dominio all'interno del documento riferito in base allo score.

4.6.5 L'applicazione *OKKAM-POP GUI*

A questo punto del progetto si è ottenuto un database notevolmente popolato di informazioni semantiche sul corpus analizzato. Ben presto tuttavia si è resa evidente la necessità di uno strumento software che potesse automatizzare ed elevare il livello di astrazione delle analisi effettuate sui dati. Tali analisi sono fondamentali sia per comprendere come migliorare le regole di estrazione sia per valutare la percentuale di informazioni realmente utili ai fini del progetto.

Da queste necessità è nato lo sviluppo dell'applicazione *OKKAM-POP GUI*, una *window application* realizzata in C#, che offre una visione rapida ed integrata dei dati presenti all'interno delle tabelle del database.

Inizialmente questa applicazione ha costituito un modo semplice ed efficace di effettuare query automatiche sul db tuttavia col tempo si è evoluta in uno strumento più complesso a livello funzionale, in grado di realizzare analisi statistiche sui dati e

utilizzabile da qualsiasi utente anche se non in possesso di particolari conoscenze tecniche riguardo al progetto.

In *OKKAM-POP GUI* è stata integrata inoltre una funzione che permette di creare in modo automatico la struttura del database presentata nel paragrafo precedente, ed effettua, richiamando internamente la console application *XMLtoDB*, l'importazione nel database dei dati contenuti nel corpus XML arricchito.

Si fa presente che l'applicazione è stata realizzata utilizzando uno stile di programmazione multithreading per gestire in modo efficiente la componente grafica e la componente logica del software. Infatti a causa del notevole tempo di elaborazione che le query eseguite sul database comportano è stato necessario separare su differenti threads la gestione delle *forms* e del codice che esegue le operazioni di recupero dei dati. In questo modo l'applicazione rimane sempre reattiva alle operazioni eseguite dall'utente senza causare disagi e fastidiose attese nel suo utilizzo.

Per un ulteriore miglioramento dal punto di vista delle prestazioni è stato inoltre realizzato un database secondario con funzionalità di supporto per la memorizzazione di elaborazioni parziali sui dati. Come si comprenderà fra poco infatti ogni analisi sui dati è caratterizzata da una prima operazione SQL di raggruppamento che dovendo essere effettuata su una notevole quantità di records rallenta in modo consistente l'intero processo.

Concretamente il database di supporto realizzato è costituito da *viste materializzate*[54] rappresentate in figura 4.10.

[54] Alcuni DBMS, come Oracle, supportano le viste materializzate. Si tratta di viste che vengono scritte fisicamente su disco per consentirne una lettura più rapida. I dati ivi contenuti vengono aggiornati automaticamente a intervalli regolari dal DBMS. Queste viste vengono utilizzate di solito per applicazioni di datawarehousing.

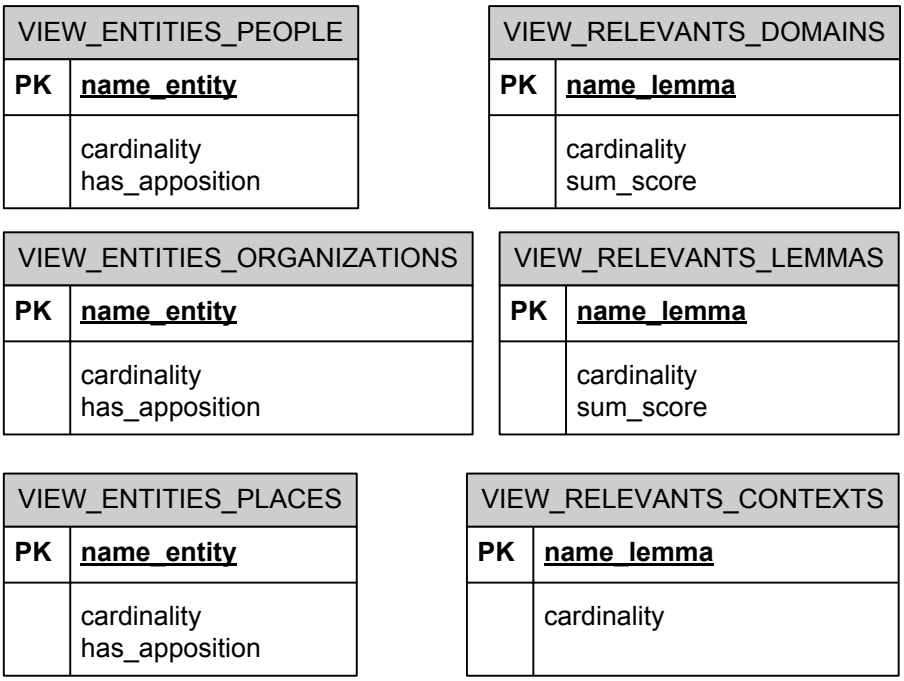

Figura 4.10 – Schema del DB di secondario di supporto

La tabella VIEW_ENTITIES_PEOPLE deriva dalla seguente query eseguita sulla tabella ENTITIES:

```
INSERT INTO dbo.view_entities_people
SELECT name_entity, count(*), 0
FROM dbo.entities
WHERE type_entity = 'PEOPLE'
GROUP BY name_entity
```

Essenzialmente rappresenta un'operazione di raggruppamento su tutte le entità PEOPLE che hanno lo stesso nome e che sono quindi accumunabili, eccetto i casi di omonimia, alla stessa persona. Il campo *cardinalità*, comune a tutte le tabelle contiene la frequenza con cui una certa entità, dominio, lemma o contesto si sono presentati nell'intero corpus.

La tabella VIEW_ENTITIES_ORGANIZATONS deriva dalla seguente query eseguita sulla tabella ENTITIES:

```
INSERT INTO dbo.view_entities_organizations
SELECT name_entity, count(*),0
FROM dbo.entities
WHERE type_entity = 'ORGANIZATIONS'
GROUP BY name_entity
```

La tabella VIEW_ENTITIES_PLACES deriva dalla seguente query eseguita sulla tabella ENTITIES:

```
INSERT INTO dbo.view_entities_places
SELECT name_entity, count(*), 0
FROM dbo.entities
WHERE type_entity = 'PLACES'
GROUP BY name_entity
```

Come si può notare nelle tre query appena presentate l'ultimo valore inserito nelle viste materializzate e settato a 0 di default. Questo valore corrisponde al campo *has_apposition* delle tre tabelle VIEW_ENTITIES_PEOPLE, VIEW_ENTITIES_ORGANIZATONS e _ENTITIES_PLACES. Tale campo è un flag booleano che memorizza quali entità possiedono almeno un'estrazione effettua tramite le regole implementate in questo progetto e serve per escludere velocemente durante le analisi sui dati tutte le entità che non sono state descritte.

La tabella VIEW_RELEVANTS_DOMAINS deriva dalla seguente query eseguita sulla tabella RELEVANTS_DOMAINS:

```
INSERT INTO view_relevants_domains
SELECT r.name_domain, count(*), sum(r.score_domain)
FROM dbo.relevants_domains r
GROUP BY r.name_domain
```

La tabella VIEW_RELEVANTS_LEMMAS deriva dalla seguente query eseguita sulla tabella RELEVANTS_LEMMAS:

```
INSERT INTO view_relevants_lemmas
SELECT r.name_lemma, count(*), sum(r.score_lemma)
FROM dbo.relevants_lemmas r
GROUP BY r.name_lemma
```

Sia nella tabella VIEW_RELEVANTS_DOMAINS che nella tabella VIEW_RELEVANTS_LEMMAS il campo *sum_score* contiene la somma di tutti gli score che un certo dominio o un certo lemma hanno ottenuto dall'analisi dell'intero corpus. Queste informazioni servono per attribuire un grado di importanza a ciascun dominio o lemma, mentre per le entità questa classifica viene stilata in base alla frequenza.

La tabella VIEW_RELEVANTS_CONTEXTS deriva dalla seguente query eseguita sulla tabella RELEVANTS_CONTEXTS :

```
INSERT INTO view_relevants_contexts
SELECT r.name_context, count(*)
FROM dbo.relevants_contexts r
GROUP BY r.name_context
```

A causa delle dimensioni limitate a 4096 MB che un database può raggiungere in *MS Sql Server 2005 Express* e data la grande mole di dati estratti, che verrà quantificata nel paragrafo 4.8, è stato necessario allocare le viste materializzate appena descritte su un database differente da quello primario, l'applicazione *OKKAM-POP GUI* astrae questa suddivisione permettendo di utilizzare un unico nome di database, quello primario, mentre quello secondario di supporto viene gestito in modo totalmente automatico e trasparente.

4.6.6 Documentazione dell'applicazione *OKKAM-POP GUI*

In questo paragrafo verrà descritto l'utilizzo dell'applicazione *OKKAM-POP GUI* mostrando le funzionalità che espone e alcuni tipici scenari di analisi sui dati.

In Figura 4.11 viene mostrata l'interfaccia principale che si apre all'avvio dell'applicazione. Essa è costituita da un classico menù orizzontale superiore e un componente *TabPage* che contiene nella *Start page* sei moduli che corrispondo a sei diverse tipologie di analisi. I tre moduli superiori permettono un'analisi sui dati dal punto di vista delle entità, mentre quelli inferiori permettono un'analisi dal punto di vista dei domini, dei lemmi e dei contesti.

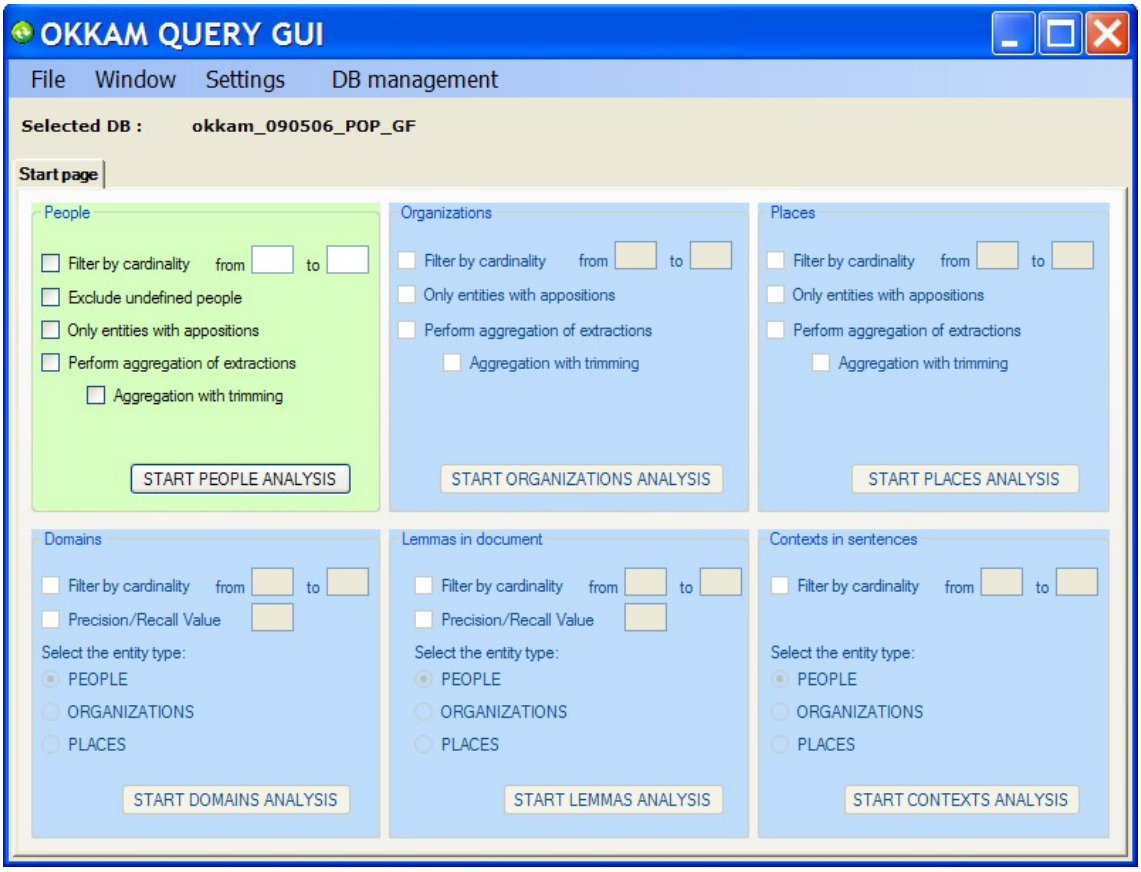

Figura 4.11 - *OKKAM-POP GUI, interfaccia principale*

Ogni modulo viene attivato, ed evidenziato con il colore verde, premendo su di esso con il mouse ed ognuno presenta un certo numero di configurazioni che verranno descritte in dettaglio fra breve.

Fra il menù e la *TabPage* troviamo inoltre una stringa, *Selected DB*, che mostra il nome del database che si sta utilizzando.

Prima di procedere con le analisi vediamo come è possibile creare il database primario e quello secondario di supporto. Per costruire automaticamente lo schema de DB descritto nel paragrafo 4.6.4 occorre premere su "DB Management" nel menù dell'applicazione e poi selezionare "Create and fill DB", come mostrato in Figura 4.12.

Si precisa che le operazioni descritte fra breve presumono che sul computer utilizzato sia presente ed attivo un servizio MS Sql e che l'utente sia in possesso delle credenziali con i diritti da amministratore per l'accesso a tale servizio. Questo è necessario perché come tutti i DBMS anche MS Sql Server mette in atto sistemi di protezione che impediscono ad utenti non autenticati di poter creare, modificare o cancellare i database.

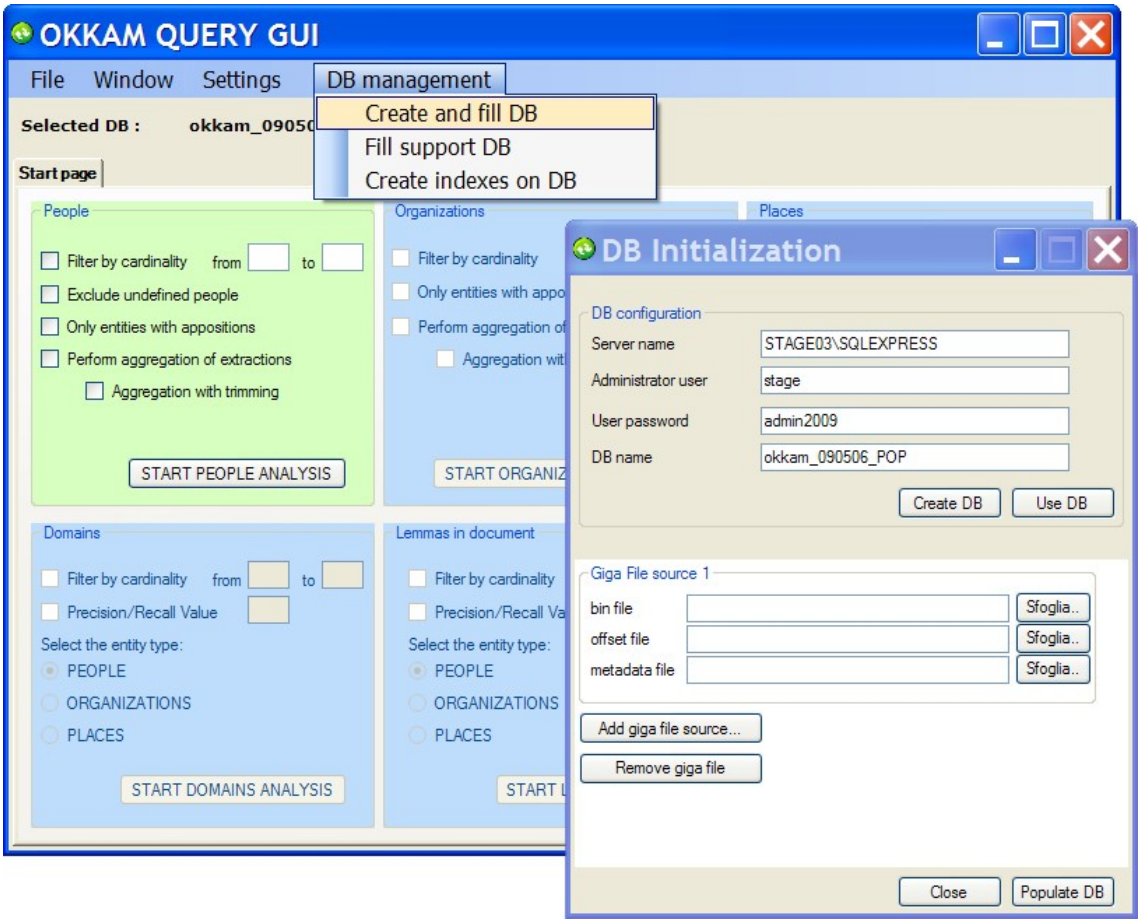

Figura 4.12 - *OKKAM-POP GUI*, creazione dei database primario e secondario

Il form "DB initialization" è composto da due sezioni principali, "DB configuration", che permette di inserire i dati per la creazione o la selezione dei DB da utilizzare per l'operazione di trasferimento dei dati da sorgente XML e la sezione "Giga File Source". All'interno della sezione "DB configuration" è obbligatorio inserire il nome di un'istanza attiva di MS Sql Server, il nome e la password di un utente di tale e infine deve essere inserito il nome del DB che si vuole creare o che si vuole utilizzare nel caso in cui tale DB esista già. Premendo sul pulsante "Create DB" si lanciano le operazioni per la creazione del DB primario e del DB secondario di supporto, nominati rispettivamente nel seguente modo: "DB name" e "DB name_SUPPORT".

Alternativamente si vuole utilizzare un DB già esistente occorre premere il pulsante "Use DB". La possibilità di utilizzo di un DB già esistente permette di popolare un DB in fasi successive aumentando in modo incrementale il suo contenuto informativo. La pressione di tali pulsanti può generare delle eccezioni che vengono gestite e descritte attraverso apposite *Message Box* che indicano la tipologia di errore riscontrato.

Nella sezione “Giga File Source” dovranno essere inseriti i path assoluti dei files che costituiscono la struttura completa di un *GIGAFILE* derivante da analisi linguistica effettuata con il server linguistico descritto nel paragrafo 4.6.2. Un *GIGAFILE* è una struttura di memorizzazione dei corpus che permette di unificare più file in un'unica soluzione. E’ possibile inserire una o più sorgenti *GIGAFILE* premendo intuitivamente sul bottone “Add giga file source”. Viceversa è possibile eliminare una sorgente *GIGAFILE* premendo sul pulsante “Remove giga file”.

Il pulsante “Populate DB” lancia le operazioni di trasferimento dei dati da *GIGAFILE* al DB, tuttavia è necessario accertarsi di avere selezionato o creato un DB prima della sua pressione. Inoltre è necessario che tutti i campi di tutte le sorgenti giga file siano completi e validi.

L’effetto concreto della pressione del pulsante Populate DB” è l’avvio di un’applicazione ETL *XMLtoDB* che analizza i dati contenuti all’interno del *GIGAFILE*, li rielabora dal punto di vista strutturale e li inserisce opportunamente all’interno del DB. Si precisa che tale applicazione ETL ha il compito di mantenere l’integrità e la coerenza delle chiavi primarie utilizzate.

L’esecuzione del trasferimento da *GIGAFILE* a DB può impiegare molto tempo a seconda delle dimensione della sorgente, ad esempio il trasferimento dei dati elaborati in questo progetto ha impiegato circa 10 ore.

Una volta terminate le operazioni di trasferimento dei dati è necessario popolare il database secondario di supporto tramite il form “Support DB filler” mostrato in Figura 4.13.

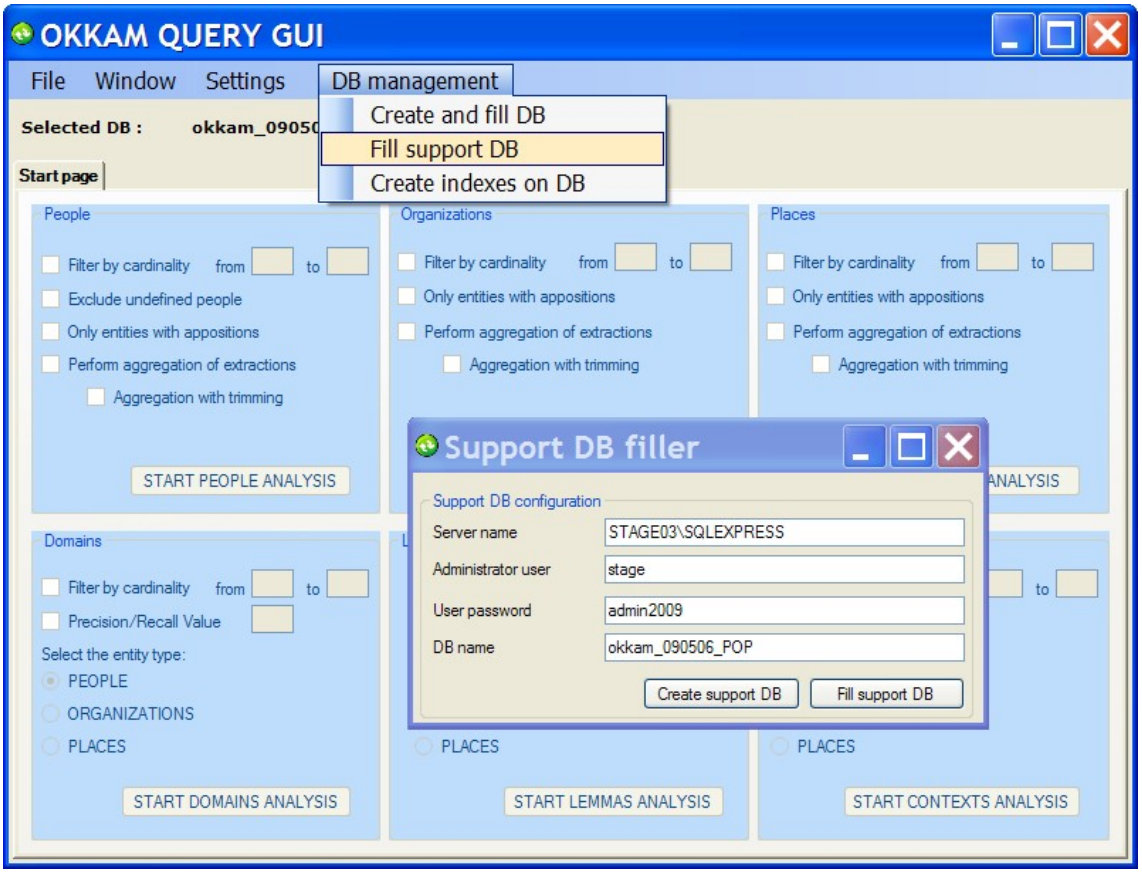

Figura 4.13 – *OKKAM-POP GUI*, popolamento del database secondario di supporto

Per effettuare tale operazione premere su "DB Management" nel menù dell'applicazione e poi selezionare "Fill support DB". Si aprirà il form "Support DB filler" che intuitivamente assomiglia molto al sezione "DB configuration" del form mostrato in Figura 4.12. Dopo aver inserito correttamente tutte le informazioni necessarie, ovvero, il nome di un'istanza attiva di MS Sql Server, il nome e la password di un utente di tale server che possieda i diritti da amministratore e il nome del DB primario, sarà possibile effettuare due operazioni: creare lo schema del DB secondario nel caso in cui per qualche motivo sia stato cancellato, oppure popolarne uno già esistente premendo rispettivamente sui pulsanti "Create support DB" o "Fill support DB". Si precisa all'interno del campo "DB name" dovrà essere inserito il nome di un DB primario già esistente e popolato. Si è scelta questa strategia così l'utente dovrà gestire per l'applicazione un solo nome del DB senza doversi ricordare entrambi i nomi del DB primario e di quello secondario, sarà l'applicazione a gestire l'utilizzo dei due database in modo totalmente trasparente rispetto all'utente.

Come riferimento si riporta che il popolamento del DB DI SUPPORTO per il il corpus utilizzato in questo progetto ha impiegato circa 45 minuti.

Una volta terminato il riempimento sia del DB primario che del DB secondario di supporto sarà possibile creare tutti gli indici necessari su entrambi i database per ottimizzare l'esecuzione delle queries effettuate dall'applicazione. Effettivamente è possibile creare tali indici anche subito dopo la creazione dei DB, tuttavia si sconsiglia tale approccio dato che si avrebbe un notevole calo di prestazioni durante la fase di trasferimento dei dati.

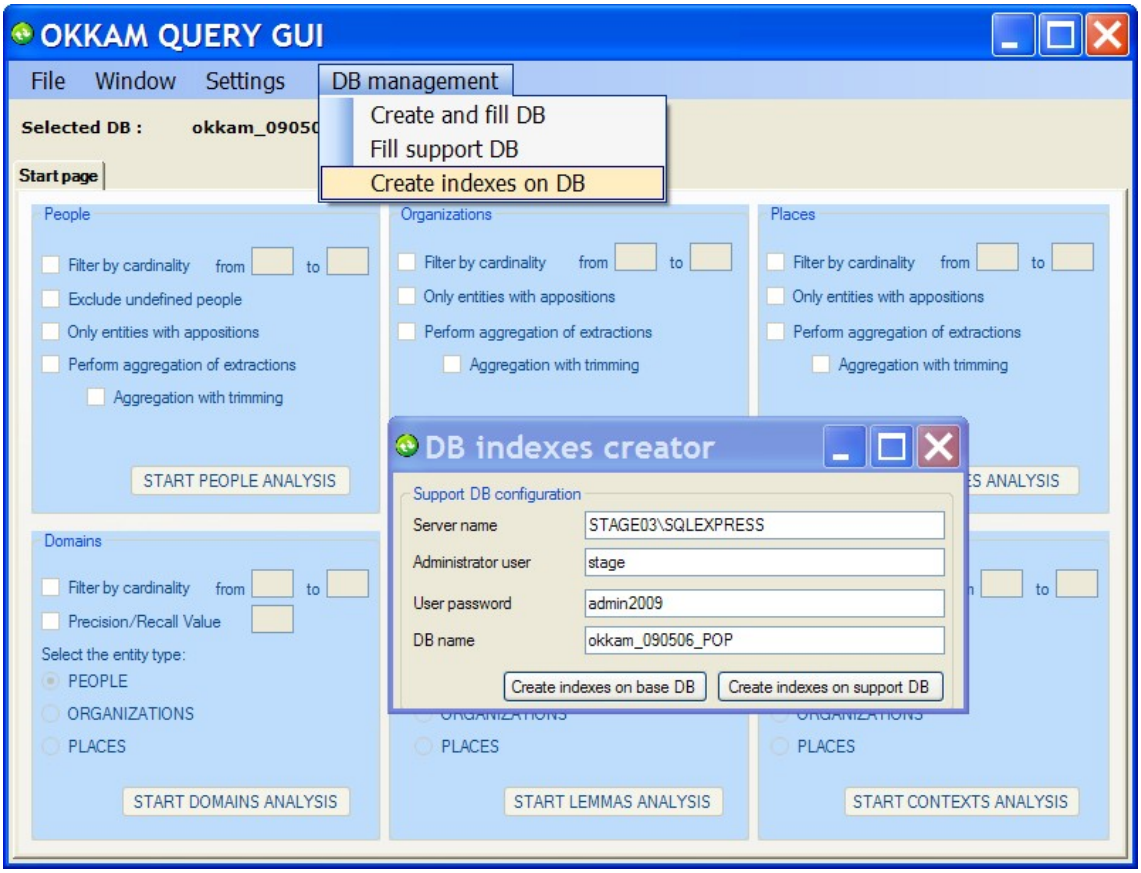

Figura 4.14 – *OKKAM-POP GUI*, creazione degli indici dee database

Per aprire il form di creazione degli indici premere su "DB Management" nel menù dell'applicazione e selezionare "Create indexes on DB", come mostrato in Figura 4.14. I campi da completare sono del tutto simili ai campi dei due precedenti form descritti e una volta inseriti correttamente i dati sarà possibile creare automaticamente gli indici sul database primario premendo sul pulsante "Create indexes on base DB" e identicamente creare gli indici sul database secondario premendo sul pulsante "Create indexes on support DB".

Nel caso specifico di questo progetto la creazione degli indici sul DB primario ha impiegato circa 15 minuti mentre sul DB secondario ha impiegato circa 4 minuti.
Una volta eseguite tutte le operazioni descritte si hanno a disposizione tutti gli strumenti per poter effettuare analisi sui dati estratti.
Le tipologie di analisi effettuabili sui dati sono essenzialmente 6, ciascuna inizializzabile attraverso i 6 moduli principali presenti nella "Start Page" dell'interfaccia principale.
Come spiegato precedentemente i tre moduli superiori permettono un'analisi dal punto di vista delle entità, mentre i tre moduli inferiori permettono un'analisi dal punto di vista dei domini e dei lemmi presenti all'interno dei documenti o dal punto di vista dei contesti. La differenza fra queste due macro tipologie di analisi sui dati risiede nel fatto che per la prima, dal punto di vista delle entità, il passo successivo mostrerà direttamente tutte le entità presenti all'interno del database raggruppate per il nome dell'entità ed ordinate in base alla frequenza con cui ciascuna entità compare, mentre per la seconda il passo successivo mostrerà un elenco dei domini, lemmi o contesti, sempre ordinati in base alla frequenza, che permetterà un'ulteriore successiva analisi dal punto di vista delle entità che hanno però questa volta una particolare attinenza con il dominio, lemma o contesto selezionato.
In Figura 4.15 viene mostrato un activity diagram che evidenza le interazioni fra utente e applicazione durante la varie fasi che costituiscono le due macro tipologie di analisi. Per mettere maggiormente in evidenza i percorsi di ciascun tipo di analisi sono stati usati dei colori, in particolare in azzurro sono rappresentate le azioni che sono coinvolte nelle analisi svolte dal punto di vista delle entità, in verde sono rappresentate le azioni coinvolte nelle analisi svolte dal punto di vista dei domini, lemmi o contesti .
Come si può notare entrambi i flussi iniziali convergono nell'azione evidenziata in rosso che rappresenta la visualizzazione integrata delle apposizioni, dei SAO e delle proprietà estratte in modo automatico con la tecnologia COGITO®.

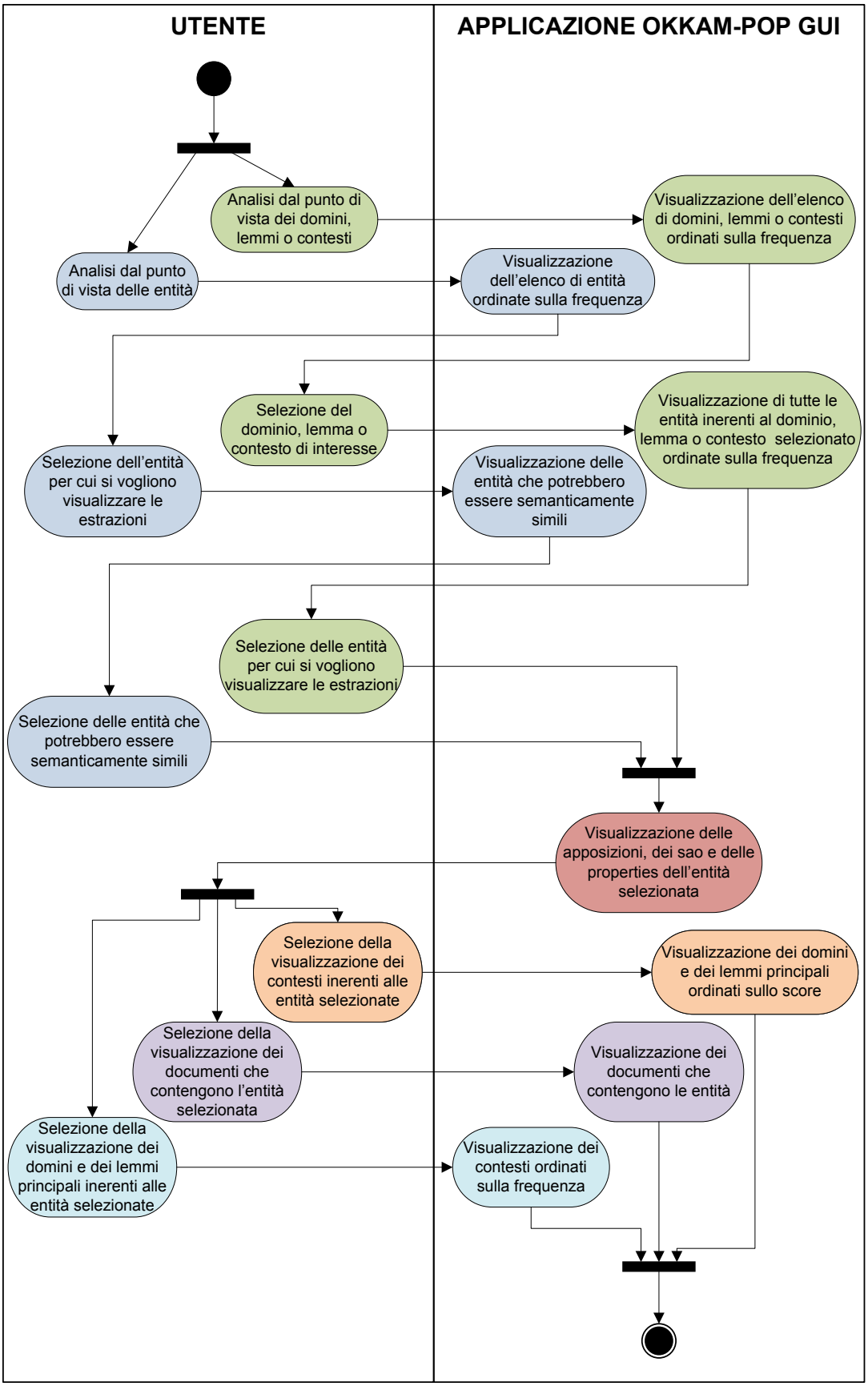

Figura 4.15 – Activity diagram delle analisi sui dati

Entrando in un caso specifico viene descritto fra breve un esempio di analisi effettuata dal punto di vista delle entità persona, prima però è necessario contestualizzare i parametri configurabili all'interno dei moduli di analisi.

Figura 4.16 – *OKKAM-POP GUI*, parametri per le analisi dal punto di vista delle entità

In Figura 4.16 possiamo osservare il modulo per l'analisi sulle entità persona che presenta i seguenti parametri:

- ✓ *Filter by cardinality* : se selezionato e impostato correttamente, inserendo un numero intero in "from" minore del numero intero inserito in "to", permette di eliminare dalla visualizzazione tutte le entità che non hanno una frequenza appartenente al range specificato. Nel caso in cui venga inserito un valore numerico solo in "from" il filtro ha un effetto "passa alto", mentre se viene impostato solo il valore numerico "to" il filtro ha un effetto passa basso;
- ✓ *Exclude undefined people* : se selezionato permette di eliminare dai dati visualizzati tutti i nomi di battesimo delle persone. E' stata fornita questa possibilità dal momento che i nomi di persona senza la presenza del cognome o altre caratterizzazioni non permettono di identificare univocamente un'entità e quindi vanno ad assorbire apposizioni, sao, o properties appartenenti a diverse entità;
- ✓ *Only entity with apposition* : se selezionato permette di visualizzare solo le entità per cui è stata estratta almeno un'apposizione dal plugin di post-processing;
- ✓ *Perform aggregation of extractions* : questo parametro, a differenza di quelli precedenti, agisce sulla fase evidenziata in rosso nella Figura 4.15, permettendo di effettuare un'elaborazione ulteriore sui dati prima della loro effettiva visualizzazione. Essenzialmente selezionando questa proprietà si chiede all'applicazione di aggregare tutte le estrazioni (apposizioni, sao e properties) in

base al testo contenuto in esse e viene fornito anche il numero di frequenza per ciascuna estrazione univoca. La selezione di questo parametro aiuta quindi a visualizzare i dati da punto di vista maggiormente statistico anche se a discapito di ciò vengono perse alcune informazioni, come ad esempio il nome della regola che ha permesso l'estrazione di una apposizione per una certa entità.

- ✓ *Aggregation with trimming* : questo parametro risulta essere strettamente dipendente rispetto a quello precedente, infatti può essere selezionato solo se viene selezionato anche il parametro "Perform aggregation of extractions". Questa funzionalità agisce sempre nella fase evidenziata in rosso nella Figura 4.15 e permette di effettuare un'operazione di *trimming* sulle apposizioni rilevate per una certa entità. L'operazione di *trimming* si occupa essenzialmente di eliminare particolari sequenze di caratteri presenti all'inizio e alla fine delle apposizioni. Tali sequenze di caratteri attualmente sono rappresentati da punteggiatura, articoli determinativi e indeterminativi, preposizioni semplici e preposizioni articolate. Questa pulizia del testo permette di rendere ancora più significativa l'operazione di aggregazione, infatti eliminando le sequenze indicate, che non possiedono un contenuto informativo rilevante, il concetto "il presidente del consiglio" verrà ad esempio considerato del tutto identico al concetto "presidente del consiglio".

Ipotizziamo ora di effettuare un'analisi selezionando tutti i parametri tranne "Filter by cardinality" e osserviamo il risultato mostrato in Figura 4.17 ponendo per il momento attenzione al form in secondo piano.

Viene automaticamente aggiunta una nuova *Page* all'interno del componente *TabPage* dell'interfaccia, il cui nome è uguale al tipo di entità su cui stiamo effettuando l'analisi, PEOPLE in questo caso. Viene mostrata una lista di entità persona ordinate in base alla frequenza con cui sono presenti all'interno del corpus e come si può notare, dato che il corpus analizzato è costituito da articoli giornalistici, le entità con frequenza più alta sono quelle inerenti alla politica.

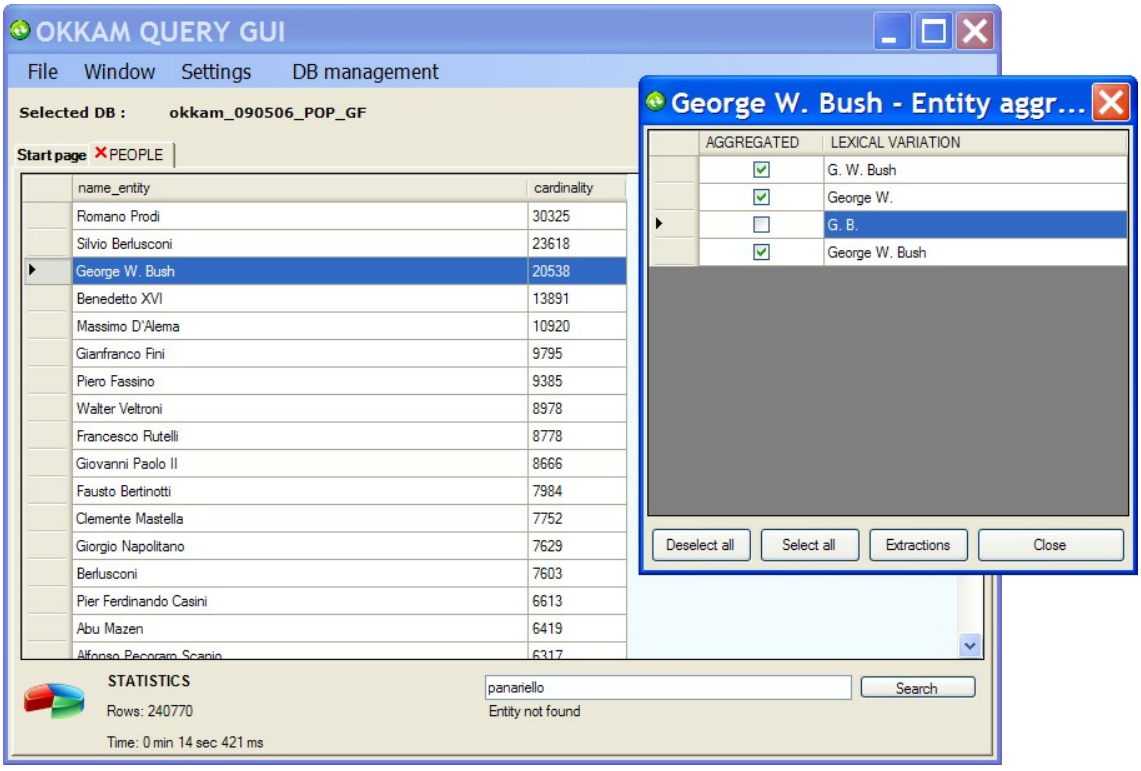

Figura 4.17 – *OKKAM-POP GUI*, esempio di analisi sulle entità persona

Per proseguire con l'analisi è necessario premere due volte su una delle entità elencate, azione che causa l'apertura del form mostrato in primo piano in Figura 4.17. Questo form, nominato "Entity aggregator" permette di verificare se nella lista di entità visualizzate precedentemente ve ne siano alcune espresse da un testo differente ma che hanno in realtà lo stesso significato di quella selezionata per l'analisi. Ad esempio nel caso specifico è molto plausibile che "G. W. Bush", "George W." e "George W. Bush" rappresentino in realtà la stessa entità, mentre "G. B." potrebbe essere troppo generico per provarne l'attinenza. Una volta effettuate le opportune selezioni nell'"Entity Aggregator" si può procedere con l'analisi premendo il pulsante "Extractions".

Termina in questo momento la fase vera e propria di configurazione dell'analisi ed inizia l'esecuzione della query sul database primario per recuperare tutte le informazioni estratte sull'entità selezionata. Il risultato di questa query, formattato in base alla parametrizzazione iniziale viene mostrato in Figura 4.18.

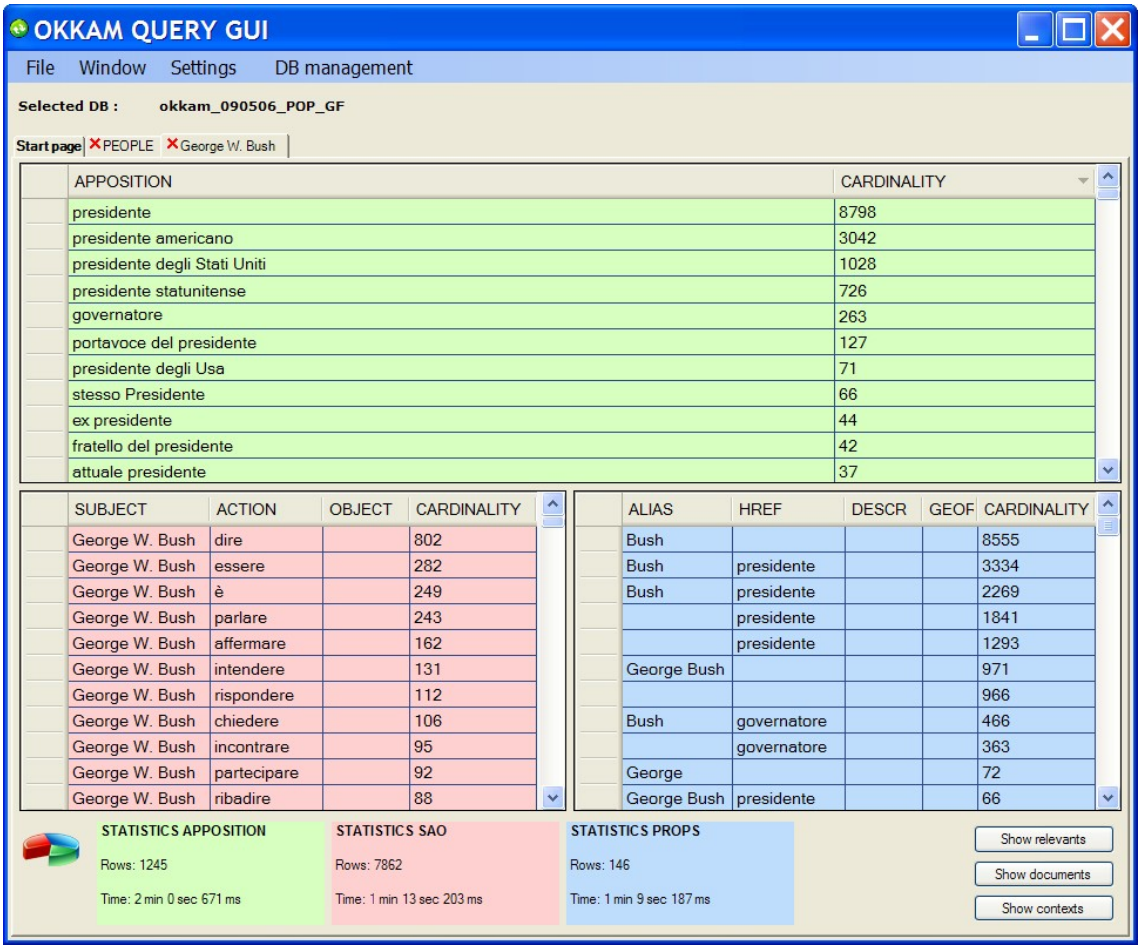

Figura 4.18 – *OKKAM-POP GUI*, dati estratti su un'entità

L'operazione di recupero dei dati può richiedere un tempo dell'ordine di qualche minuto, a seconda dell'importanza dell'entità in termini di frequenza all'interno del corpus. Il numero di record e i tempi individuali per il recupero delle apposizioni, dei SAO delle properties sono visualizzati nella parte inferiore della *Page* che contiene i risultati. In verde vengono evidenziati i dati relativi alle apposizioni, in rosa i dati relativi alle triplette SAO e in azzurro le proprietà estratte da COGITO®.

Nell'esempio mostrato si può notare che a volte le stesse informazioni vengono estratte sia dalle regole sintattiche realizzate per il progetto che dalla tecnologia COGITO®. In questo caso "presidente" e "governatore" compaiono in entrambe le sezioni, tuttavia le apposizioni forniscono più informazioni grazie a descrizioni più dettagliate. Indicativamente le informazioni estratte da COGITO® hanno minore contenuto informativo ma una certezza prossima al 100% mentre quelle estratte con le regole sintattiche sono più descrittive a discapito di un precisione inferiore e difficilmente prevedibile.

In basso a destra si possono notare tre pulsanti che permettono di visualizzare ulteriori informazioni. “Show relevants” visualizza, come mostrato in Figura 4.19, i domini e i lemmi più importanti riscontrati all’interno degli stessi documenti in cui compare l’entità analizzata.

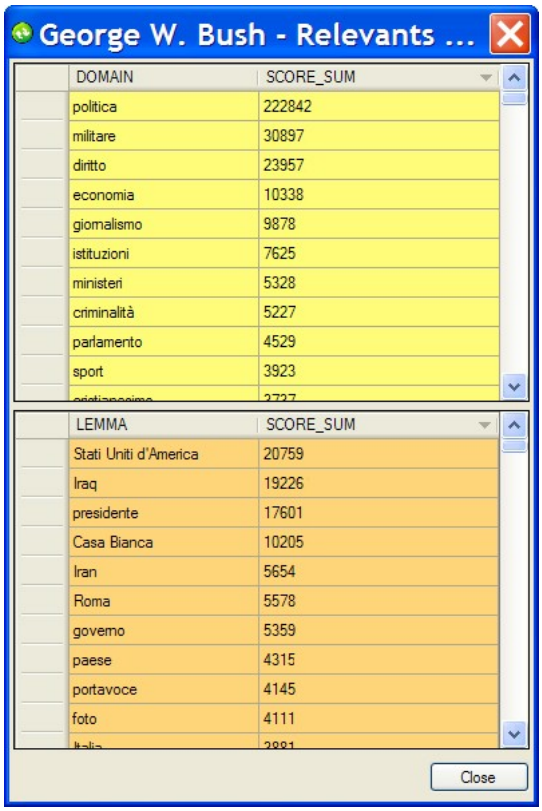

Figura 4.19 – *OKKAM-POP GUI*, visualizzazione dei domini e dei lemmi

Il pulsante “Show documents” permette di recuperare le news giornalistiche all’interno delle quali è presente l’entità, mentre il pulsante “Show Contexts” visualizza i contesti inerenti all’entità analizzata in un modo equivalente a quello visualizzato in Figura 4.20. I contesti non vengono ordinati in base allo score acquisito nell’interno corpus come i domini e lemmi, ma in base alla frequenza.

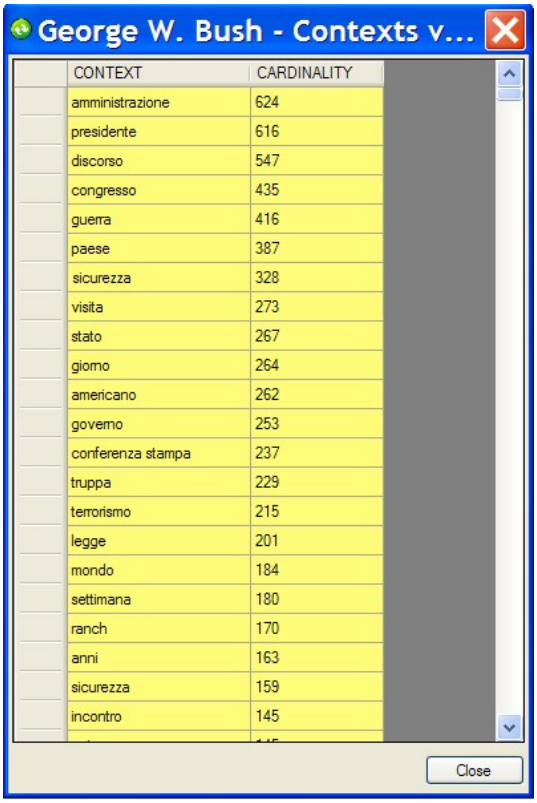

Figura 4.20 – *OKKAM-POP GUI*, visualizzazione dei contesti

Si conclude a questo punto l'esempio di analisi proposto e si rinvia al paragrafo successivo per una presentazione di alcuni risultati.

L'analisi effettuata dal punto di vista dei domini, dei lemmi e dei contesti non verrà tratta in dettaglio, ma si provvederà a descrivere i parametri di configurazione mostrati in Figura 4.21.

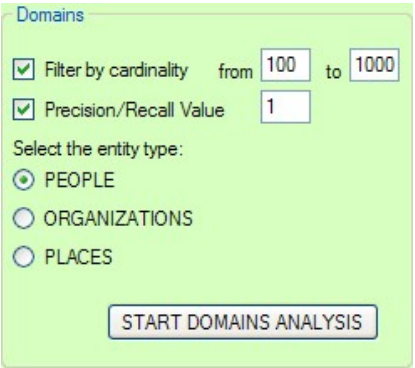

Figura 4.21 – *OKKAM-POP GUI*, parametri per le analisi dal punto di vista dei domini

Il modulo per l'analisi sui domini presenta i seguenti parametri:

- ✓ *Filter by cardinality*: se selezionato e impostato correttamente, inserendo un numero intero in "from" minore del numero intero inserito in "to", permette di eliminare dalla visualizzazione tutti i domini che non hanno una frequenza

compresa nell'intervallo specificato. Nel caso in cui venga inserito un valore numerico solo in "from" il filtro ha un effetto "passa alto", mentre se viene impostato solo il valore numerico "to" il filtro ha un effetto passa basso;

- ✓ *Precision/Recall Value*: se selezionato e impostato con un valore numerico intero positivo questo parametro rappresenta un filtro che permette di visualizzare le entità che hanno un definito valore di attinenza con il dominio selezionato per l'analisi. Come detto precedentemente l'analisi linguistica di un documento fornisce una lista di domini, ognuno con un preciso score, a cui tale documento appartiene. Impostando questo parametro a 1 è possibile selezionare tutte e solo le entità che compaiono all'interno di documenti il cui primo dominio è quello selezionato per l'analisi. Impostando questo parametro a *n* si intende invece selezionare tutte le entità che compaiono all'interno di documenti a cui è stato attribuito il dominio selezionato nelle prime *n* posizioni. In definitiva un valore basso di questo parametro aumenta la precisione in termini di attinenza fra il dominio selezionato e le entità visualizzate mentre all'aumentare di questo parametro aumenta il *recall* delle entità con una conseguente diminuzione della precisione.
- ✓ *Select the entity type*: questo parametro permette di selezionare in modo esclusivo il tipo di entità su cui si vuole effettuare l'analisi.

Si vedano ad esempio i risultati visualizzati in Figura 4.22 che derivano dall'analisi effettuata sul dominio "eventi televisivi" con valore di *precision/recall* impostato a 1.
Come risulta evidente il punto di vista dei domini permette di effettuare in modo semplice ma efficace una *clusterizzazione* delle entità che potrebbe risultare molto utile, nell'ambito di studi futuri, per porre in relazione le entità appartenenti ad un ambito comune.
Partendo dai dati visualizzati in Figura 4.22 è possibile inoltre effettuare un normale analisi dal punto di vista delle entità semplicemente selezionando una delle entità presenti nella lista.

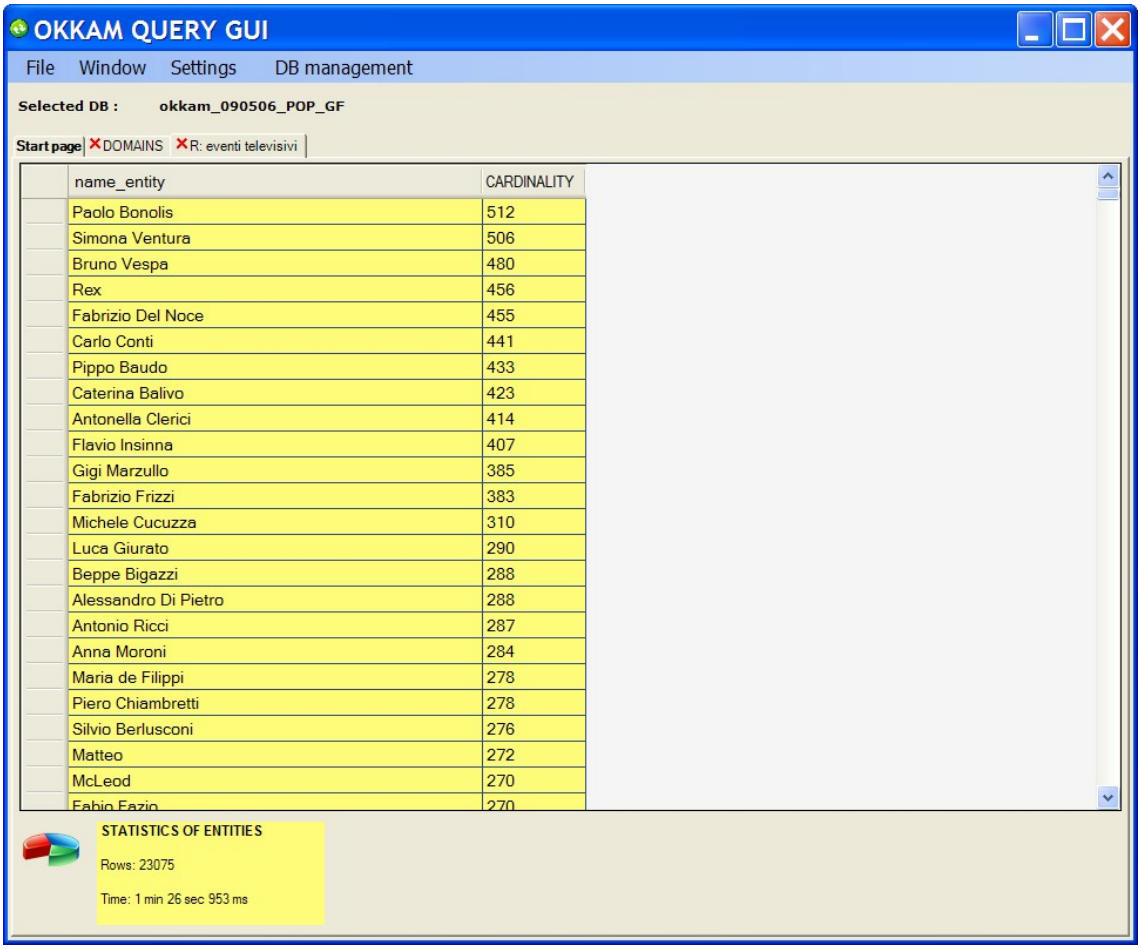

Figura 4.22 - *OKKAM-POP GUI*, analisi sul domino "eventi televisivi"

Si conclude la descrizione dell'applicazione *OKKAM-POP GUI* facendo presente che le informazioni visualizzate in Figura 4.18 nell'area verde delle apposizioni potrebbero essere oggetto di nuove elaborazioni di tipo semantico. Si potrebbero ad esempio affrontare studi per cercare di effettuare aggregazioni più significative, che sfruttino metodologie più raffinate di quella lessicale, utilizzata per motivi di ridotta complessità compatibile con i termini temporali del progetto.

4.7 Risultati ottenuti

Si conclude il presente capitolo riportando un serie di interessanti risultati ottenuti grazie al progetto affrontato presso la Expert System.

Di seguito vengono riportati i numeri che caratterizzano il database realizzato, evidenziando che lo sviluppo di applicativi per l'elaborazione di ingenti quantità di dati ha richiesto una attenta valutazione a livello progettuale per la scelta di strategie di memorizzazione dei dati che potessero migliorare il tempo complessivo di elaborazione.

La tabella 4.1 mostra alcuni dati relativi ai documenti analizzati a alle entità .

Numero di documenti elaborati	1.158.841
Numero di entità generiche globali estratte	10.469.679
Numero di entità generiche per documento	9,034
Numero di entità generiche distinte estratte	653.766
Frequenza media di ciascuna entità generica	16,014

Tabella 4.1 – Statistiche su documenti ed entità globali

Si fa presente che le frequenze medie che vengono riportate non possono essere ritenute una descrizione precisa della distribuzione dei dati, infatti come mostrato in Figura 4.23, vi sono pochi valori che si manifestano con un frequenza molto alta, mentre la maggior parte delle entità si manifesta una o al massimo due volte.

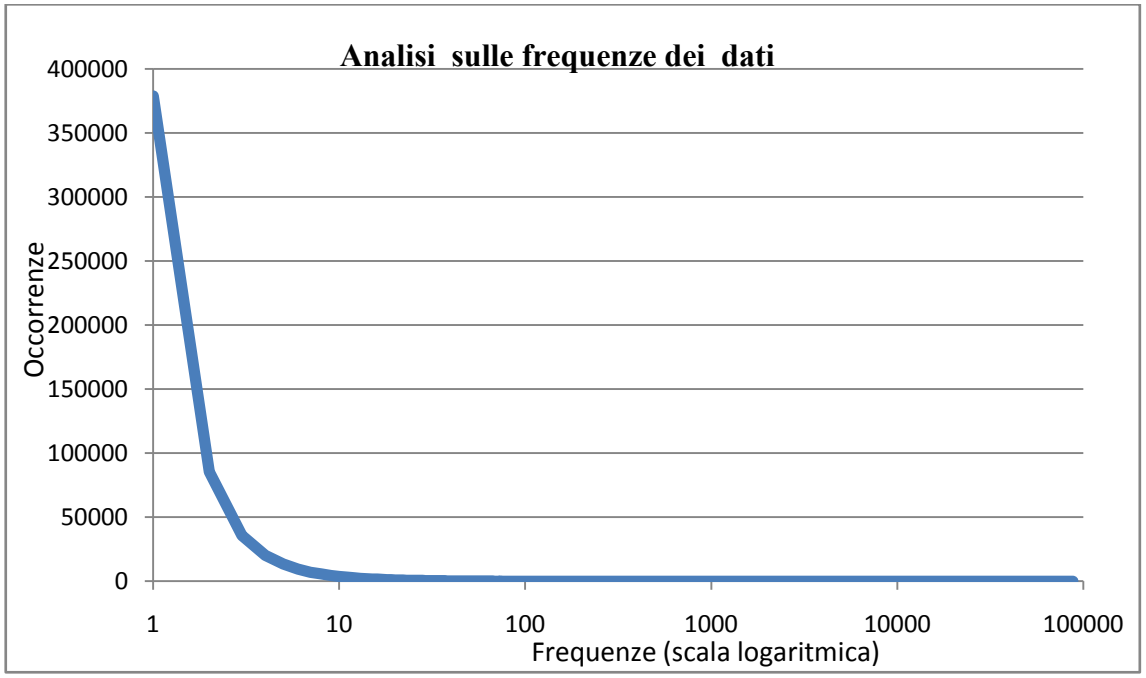

Figura 4.23 – Analisi sulle frequenze dei dati

Dati con frequenza molto alta sono ad esempio all'interno del corpus "Roma", "Italia", "Romano Prodi" e "Silvio Berlusconi", ovvero entità di indiscussa importanza sociale che ne giustifica la grande ricorrenza all'interno degli articoli giornalistici.

Tuttavia se ci sofferma a pensare al fatto che forse sia più utile descrivere entità che siano poco conosciute piuttosto che entità già note e facilmente riconoscibili rappresenta già un buon punto di partenza il fatto che la maggior parte delle entità abbia un frequenza bassa ovvero una bassa notorietà. Resta da valutare ora come le regole di estrazione abbiano operato su tali entità.

Di seguito in tabella 4.2 vengono riportati in dettaglio alcuni dati sulle entità persone.

Numero totale di persone estratte	4.149.241
Numero di persone per documento	3,581
Numero di persone distinte estratte	465.107
Frequenza media di ciascuna persona	8,921

Tabella 4.2 – Statistiche sulle entità persona

In tabella 4.3 sono mostrati alcune informazioni in dettaglio sulle entità organizzazioni.

Numero totale di organizzazioni estratte	3.352.119
Numero di organizzazioni per documento	2,893
Numero di organizzazioni distinte estratte	117.113
Frequenza media di ciascuna organizzazione	28,623

Tabella 4.3 – Statistiche sulle entità organizzazione

In tabella 4.4 infine sono mostrati alcune informazioni sulle entità luoghi.

Numero totale di luoghi estratti	2.968.319
Numero di luoghi per documento	2,561
Numero di luoghi distinti estratti	71.546
Frequenza media di ciascuno luogo	41,488

Tabella 4.4 – Statistiche sulle entità luogo

Con la successiva tabella si passa ad un'analisi sulle estrazioni effettuate tramite le regole implementate nell'ambito del progetto e ci si rende immediatamente conto su come la maggior parte dei concetti rilevati sia inerente alle persone piuttosto che alle organizzazioni o ai luoghi.

Numero totale di concetti estratti	1.744.256
Concetti estratti per le persone	1.672.655
Numero medio di concetti per persona	3,596
Concetti estratti per le organizzazioni	39.754
Numero medio di concetti per organizzazione	0,339
Concetti estratti per i luoghi	31.847
Numero medio di concetti per luogo	0,445

Tabella 4.5 – Statistiche sui concetti estratti tramite le regole

Ciò risulta estremamente plausibile dal momento che lo stile giornalistico tende a descrivere in modo molto più dettagliato le persone coinvolte negli eventi piuttosto che le altre tipologie di entità. Anzi a volte luoghi e organizzazioni vengono utilizzati per caratterizzare le persone, si pensi ad esempio alle frasi "Valentino Rossi, residente a Tavullia …" o "Guido Bertolaso, direttore del dipartimento della Protezione Civile …".
Di seguito in tabella 4.6 vengono mostrati alcuni dettagli statistici sulle triplette soggetto, azione, oggetto riscontrate nel testo.

Numero totale di triplette SAO estratte	4.757.225
SAO estratti per le persone	3.225.727
Numero medio di SAO per persona	6,935
SAO estratti per le organizzazioni	625.465
Numero medio di SAO per organizzazione	5,341
SAO estratti per i luoghi	906.033
Numero medio di SAO per luogo	12,664

Tabella 4.6 – Statistiche sulle triplette SAO

Le triplette SAO non sempre sono completamente definite, in alcuni casi si verifica la sola presenza del soggetto e dell'azione, ad esempio quando si hanno verbi intransitivi, in altri invece si ha solo l'azione e l'oggetto.
Si prenda come esempio la frase "Giancarlo Fisichella guida la sua vettura …". In questo caso si sta parlando di un pilota di F1 e vi è quindi una grande attinenza con il verbo "guidare" e l'oggetto "vettura", tuttavia la maggior parte della popolazione può guidare una vettura, in mancanza di ulteriori specificazioni è quindi difficile poter utilizzare SAO di questo tipo per caratterizzare in modo efficace un'entità. Queste informazioni possono invece assumere un senso maggiore se introdotte come relazioni all'interno della rete semantica SENSIGRAFO®, se è possibile infatti attribuire una certa azione ricorrente ad una persona potrebbe essere facilitata l'operazione di disambiguazione linguistica su future analisi.
In tabella 4.7 si mostrano alcune caratteristiche delle proprietà estratte da COGITO®.

Numero totale di PROPS estratte	10.135.012
PROPS estratte per le persone	3.182.192
Numero medio di PROPS per persona	6,842
PROPS estratte per le organizzazioni	2.967.649
Numero medio di PROPS per organizzazione	25,340
PROPS estratte per i luoghi	3.985.171
Numero medio di PROPS per luogo	55,701

Tabella 4.7 – Statistiche sulle proprietà estratte da COGITO®

Le proprietà, come detto precedentemente vengono recuperate da COGITO® sia attraverso algoritmi euristici, sia attraverso informazioni già presenti all'interno del SENSIGRAFO®. Nel secondo caso è evidente la necessità di una disambiguazione pressoché certa per poter ritenere valide le proprietà estratte.

Effettuando un confronto dei valori che caratterizzano i concetti estratti, tabella 4.5, e le proprietà si può notare che COGITO® è in grado, in media, di fornire maggiori informazioni per le organizzazioni e per i luoghi, mentre le regole, sempre in media, riescono a fornire maggiori informazioni per le persone.

Questo dato deriva dal fatto che per le organizzazioni e soprattutto per i luoghi COGITO® riporta spesso proprietà che non sono presenti nel testo che si sta analizzando ma informazioni strutturate nel SENSIGRAFO®.

Per quanto riguarda i domini, i lemmi principali e i contesti vengono riportati di seguito i primi 50 valori più rilevanti per ciascuna categoria

DOMINI		LEMMI		CONTESTI	
VALORE	**SCORE**	**VALORE**	**SCORE**	**VALORE**	**FREQUENZA**
politica	4594884	Roma	390317	presidente	393857
diritto	1338775	governo	319575	governo	216870
sport	1186860	presidente	307223	paese	166639
economia	1048511	ministro	261470	ministro	163921
giornalismo	658753	Italia	209275	giorno	149768
calcio	582886	Milano	198921	gruppo	118091
militare	497044	Romano Prodi	167790	anno	109670
criminalità	395332	donna	152956	incontro	95169
cristianesimo	342697	paese	147237	persona	91885
medicina	304924	uomo	143309	sindaco	91245
diritto penale	290353	Unione Europea	136471	partito	90832
finanza	276993	Silvio Berlusconi	130267	commissione	90279
lavoro	275634	persona	125178	mese	89936
commercio	262508	partito	123558	leader	81746
ministeri	257555	milione	121752	lavoro	80621
borsa	230258	radiotelevisione	114499	segretario	80515
istituzioni	220501	Torino	111093	città	77650
polizia	200510	Stati Uniti d'America	101341	società	76823
impresa	186922	riga	99405	regione	73832
eventi televisivi	175198	Napoli	98014	giornalista	72860
parlamento	174207	sindaco	96766	programma	71697
trasporti	146281	George W. Bush	94719	operazione	69172
cinema	139124	squadra	93619	decisione	68861
amministrazione pubblica	130548	carabiniere	92891	problema	67733
sicurezza sociale	128722	società	92576	politica	66969
musica	118394	legge	91755	uomo	66901
spettacolo	105832	New York	90633	mondo	64823
autoveicoli	86545	Alleanza Nazionale	86697	settimana	63788

religione	79351	portavoce	85618	intervento	63300
strada	79072	giornalista	83776	nota	63270
arte	75475	dato	83655	stato	63075
sindacati	74544	film	82727	premier	62282
legislazione	68208	rialzo	82459	famiglia	61621
marina	67716	Forza Italia	82185	giornata	61074
banca	66968	Polizia di Stato	80367	donna	60830
scuola	65768	premier	78761	punto	60382
idrografia	62940	ora	75921	progetto	60334
meteorologia	62128	politica	75618	centro	59892
teatro	61078	Iraq	74671	legge	59859
editoria	60772	problema	74226	sera	58635
letteratura	57140	sport	73052	accordo	58507
sociologia	55266	progetto	72709	parte	57613
fisco	52909	radiogiornale	71539	comune	57373
armi	51956	voto	70989	rapporto	57069
anatomia	51722	bambino	70694	iniziativa	56548
industria	51681	candidato	70662	ora	55980
alimenti	49765	maggioranza	70562	deputato	55212
edilizia	49552	Democratici di Sinistra	70384	proposta	55014
diritto pubblico e amministrativo	49439	anni	68793	senatore	54696
ferrovia	48551	accordo	67779	caso	54446

Tabella 4.8 – Domini, lemmi e contesti più frequenti

Il valore di *score* per i domini e i lemmi corrisponde alla somma di tutti i valori attribuiti dalla tecnologia COGITO® all'interno di ciascun documento del corpus, mentre per i contesti viene riportato il numero totale di occorrenze.

Vengono proposti ora alcuni risultati ottenuti con l'estrazione di concetti tramite le regole sintattiche. In tabella 4.9 compaiono esempi di caratterizzazioni per le entità persona.

REGOLA people_ART_SOS_AGG_NPH	
PERSONA	**CONCETTO**
Ann Keren	La cantautrice francese
Ann Louise Bardach	la giornalista americana
Annie Lennox	la popstar britannica
Anp Abu Mazen	il presidente palestinese
Antoine Deneriaz	il campione olimpico
Antonella De Lillo	la regista italiana
Antonio Capriati	il boss barese
Antonio Giaconi	il pm livornese
Antonio Meucci	l' inventore fiorentino
Roberto Bustinello	il penalista veronese

REGOLA people_ART_AGG_SOS_PRE_NPR_NPH	
PERSONA	**CONCETTO**
Fabrizio Cicchito	il vice coordinatore di Forza Italia
Pantaleo Macarelli	l' unico consigliere regionale del Cdu
Francesco Saverio Borrelli	L' ex procuratore di Milano
Benigno Bartoletti	l' ex medico della Ferrari
Luca Rajola Pescarini	L' ex colonnello del Sismi
Ennio Antonelli	il cardinale arcivescovo di Firenze
Angelo Bottini	Il nuovo soprintendente archeologico di Roma
Carlo Freccero	l' ex direttore di Raidue
Kofi Annan	l' ex segretario generale delle Nazioni Unite
Mitt Romney	l' ex governatore del Massachusetts
REGOLA people_NPH_PNT_ART_SOS_AGG_PRE_SOS_PRE_SOS	
PERSONA	**CONCETTO**
Maurizio Mattei	il designatore unico degli arbitri di serie A
Domenico Miceli	il medico chirurgo accusato di concorso in associazione mafiosa
Alessandro Amadori	l' allenatore storico della nazionale di pallavolo
Massimo Vannucci	il presidente nazionale della Lega delle autonomie
Riad Jawad Taki	un deputato sciita dell' Alleanza degli iracheni
Benedetta Ceccarelli	la primatista italiana dei 400 con barriere
Lamont Ned	un imprenditore milionario del settore delle telecomunicazioni
Dounia Ettaib	la vicepresidente lombarda dell' Associazione delle donne
John Griffin	uno scienziato esperto nei sistemi di difesa
Marco Lanzetta	il chirurgo esperto di trapianto di arti
REGOLA people_NPH_PNT_SOS_AGG_PRE_NPR_PRE_NPR	
PERSONA	**CONCETTO**
Angelino Alfano	coordinatore regionale di Forza Italia in Sicilia
Nicola Adamo	segretario regionale della Calabria dei Ds
John R. Bolton	rappresentante permanente designato degli Usa all' Onu
Alfio Nicotra	responsabile nazionale del Dipartimento Pace del PRC
Liberato Del Mastro	segretario generale provinciale del Siulp di Napoli
Maria Stella Gelmini	coordinatrice regionale della Lombardia di Forza Italia
Francesco Mario Polella	comandante interregionale della Guardia di finanza dell' Italia meridionale
Eileen Daly	coordinatrice medica del Cicr a Gaza
Giuseppe Scaramuzza	segretario regionale del Lazio di Cittadinanzattiva-Tribunale
Giovanni Collino	responsabile nazionale del Dipartimento Enti Locali di Alleanza Nazionale

REGOLA people_NPH_SOS_PRE_NPR	
PERSONA	**CONCETTO**
Roberto Villetti	vice presidente dello Sdi
Henryk Muszynski	arcivescovo di Gniezno
Clemente Mastella	segretario dei Popolari
Akihito	imperatore del Giappone
Alessandro Ciarlo	presidente dell' Aicun
Mario Monti	ministro dell' Ulivo
Gianni Fanecco	segretario della Fim
Franco Giordano	capogruppo del Prc
Francesco Nespega	amministratore delegato di Jetix Europa
Franco Mandelli	presidente dell' Ail

Tabella 4.9 – Estrazioni effettuate tramite regole per le persone

Di seguito in tabella 4.10 sono visualizzate alcune estrazioni effettuate per le entità organizzazione.

REGOLA organizations_NPR_PNT_SOS_AGG	
ORGANIZZAZIONE	**CONCETTO**
Intel	colosso mondiale
British Rail	società pubblica
Financial Services Authority	organo di controllo britannico
Ryanair	compagnia aerea irlandese
Winn-Dixie Stores	Gruppo statunitense
Francese Leclerc	catena leader
Kroymans Corporation	azienda internazionale
BirdLife International	network mondiale
Democrazia Cristiana	formazione politica
Franco Ziche	azienda vicentina
REGOLA organizations_NPR_PNT_ART_SOS_PRE_SOS_AGG	
ORGANIZZAZIONE	**CONCETTO**
Doppler	un gruppo di elettro-amatori romani
Enit	l' Ente nazionale del turismo italiano
Global Strategy Group	un' organizzazione di sondaggi newyorchese
Sea	la società di gestione degli aeroporti milanesi
Mdl	il partito della minoranza turca
Upi	l' Unione delle Province italiane
Assiom	l' associazione degli operatori italiani
Merloni Progetti	la società di ingegneria del gruppo fabrianese
Dap	il Dipartimento dell' amministrazione penitenziaria
National Basketball Association	la lega del basket professionistico

REGOLA organizations_NPR_PAR_SOS_PAR	
ORGANIZZAZIONE	**CONCETTO**
Fipe	(Federazione Italiana piccoli esercizi)
Spla	(Movimento popolare di liberazione del Sudan)
Epp-Ed	(Partito popolare-Democratici europei)
Grtn	(Gestore della rete di trasmissione nazionale)
Toroc	(comitato organizzatore dei Giochi)
Lac	(Lega abolizione caccia)
Amga	(Azienda mediterranea Gas e Acqua)
Smat	(Società Metropolitana Acque Torino)
Npd	(partito nazionaldemocratico tedesco)
Arpav	(Agenzia regionale per l' ambiente)
REGOLA organizations_NPR_PNT_ART_SOS_PRE_SOS_PRE_SOS_AGG	
ORGANIZZAZIONE	**CONCETTO**
Crui	il Comitato dei Rettori delle università italiane
Asstra	l' associazione delle aziende del trasporto locale
Finsoe	la holding di controllo del gruppo assicurativo
Finpart	la holding della moda in difficoltà finanziarie
Ucimu	l' associazione dei costruttori di macchine utensili
Hamas	il movimento integralista al governo nei territori palestinesi
Buonitalia	la società del Ministero delle politiche agricole
South Sydney Rabbitohs	la squadra di rugby del campionato australiano
Aiea	il consiglio dei governatori dell' Agenzia atomica internazionale
Nev	l' agenzia della Federazione delle chiese evangeliche
REGOLA organizations_NPR_PNT_SOS_PRO_VER_ART_SOS_PRE_SOS	
ORGANIZZAZIONE	**CONCETTO**
Fortune Brands.	gruppo che possiede la marca di bourbon
Assograniti	associazione che tutela gli interessi dei cavatori
European Gay Police Association	rete che riunisce le associazioni di gay
Neurothon onlus	associazione che finanzia la ricerca scientifica sulle staminali
Helen Bamber	associazione che sostiene le vittime di genocidio
Gori	società che gestisce le risorse idriche del Nolano
Honda	azienda che fornisce le moto al team
Trambus	società che gestisce il trasporto pubblico di superficie
Confindustria Ancma	associazione che riunisce i produttore di due ruote
Sundance Channel	canale tv che tratta i rapporti tra finanza

Tabella 4.10 – Estrazioni effettuate tramite regole per le organizzazioni

Infine in tabella 4.11 compaiono alcune estrazioni effettuate per le entità luogo.

REGOLA places_NPR_PNT_SOS_PRE_SOS_PRE_NPR	
PERSONA	**CONCETTO**
Resistencia	capoluogo della provincia del Chaco
Taipei	capitale dell' isola di Taiwan
Pascarola	frazione del Comune di Caivano
Bassora	città del sud dell' Iraq
Vada	frazione del comune di Rosignano
Mossul	capoluogo della provincia di Ninive
Dossi	frazione del Comune di Sabbioneta
Roselle	frazione del comune di Grosseto
Hilla	capoluogo della provincia di Babilonia
Falluja	città a ovest di Baghdad
REGOLA places_NPR_PNT_AGG_SOS_PRE_NPR	
PERSONA	**CONCETTO**
Latikia	nota località balneare sul Mediterraneo
Adria	antico porto sull' Adriatico
Tavazzano con Villavesco	piccolo comune alle porte di Lodi
Darfur	immensa zona del Sudan
Gavarno	popolosa frazione di Nembro
Kalsa	miserabile quartiere di Palermo
Osaka	seconda città del Giappone
Iglesias	storico feudo dell' Udc
Stati Uniti d'America	unico paese membro dell' Aiea
Illegio	piccolo paesino della Carnia
REGOLA places_NPR_PAR_PRE_SOS_PRE_NPR_PAR	
PERSONA	**CONCETTO**
Zaria	(nello stato del Kaduna)
Dorset	(nel sud dell' Inghilterra)
Karachi	(nel sud del Pakistan)
Osaka	(ad ovest del Giappone)
Iasi	(nel nord-est della Romania)
Pocenia	(in provincia di Udine)
San Vito al Tagliamento	(in provincia di Pordenone)
Ramadi	(a ovest di Baghdad)
Pompei	(in provincia di Napoli)
Padula	(in provincia di Salerno)

REGOLA places_NPR_PNT_ART_AGG_SOS_PRE_SOS_PRE_NPR	
PERSONA	**CONCETTO**
Pontos	una piccola località in provincia di Barcellona
Genoni	un piccolo paesino in provincia di Nuoro
Kandahar	la grande città del sud dell' Afghanistan
Mugnano del Cardinale	l' ultimo comune della provincia di Avellino
Palestro	l' ultimo paese della provincia di Pavia
Porto Seguro	una nota località balneare nello stato di Bahia
Karachi	il grande porto del sud del Pakistan
Darfur	la martoriata regione dell' ovest del Sudan
New Orleans	il delicato ambiente naturale del Delta del Mississippi
Port Arthur	un importante centro di raffinerie in Texas
REGOLA places_NPR_PNT_SOS_PRE_AGG_SOS_PRE_NPR	
PERSONA	**CONCETTO**
Banda Aceh	capitale dell' omonima provincia a Sumatra
Gilan	capoluogo dell' omonima provincia sul Mar Caspo
Trivandrum	regione dell' estremo sud dell' India
Nuku'alofa	capitale del piccolo regno delle isole Tonga
Yemen	paese dell' estremo sud della penisola arabica
Pivin	paese nelle strette vicinanze di Prostejov
Ternate	capitale dell' omonima isola delle Molucche
Sciarborasca	piccolo centro nell' immediato entroterra di Cogoleto
Oaxaca	capitale dell' omonimo stato del Messico
Montenegro	località del piccolo dipartimento di Quindio

Tabella 4.11 – Estrazioni effettuate tramite regole per i luoghi

I risultati mostrati evidenziano il potenziale delle regole di estrazione implementate in questo progetto, tuttavia un problema per cui non si è riusciti a trovare una soluzione efficace è la capacità di distinguere in modo automatico il rumore dalle estrazioni pertinenti. Tale problema potrebbe essere risolto con un'ulteriore analisi che tenti di effettuare un'aggregazione semantica dei concetti estratti. In questo modo si potrebbe utilizzare la frequenza di ciascun concetto per discriminare con una maggiore precisione le estrazioni significative. All'interno di *OKKAM-POP GUI* sono state implementate alcune funzionalità di aggregazione dei concetti simili, tuttavia si sono utilizzate metodologie basate puramente sull'aspetto lessicale che offrono risultati discreti ma nettamente migliorabili con un approccio semantico.

5 POSSIBILI SVILUPPI FUTURI

Come esposto nel paragrafo 4.2 esistono quattro principali tipologie di modelli per l'implementazione di sistemi di Information Extraction:

- ✓ modello a codifica manuale basato su regole;
- ✓ modello a codifica manuale basato su metodi statistici;
- ✓ modello ad apprendimento automatico basato su regole;
- ✓ modello ad apprendimento automatico basato su metodi statistici.

OKKAM-POP, per motivi legati alla realizzabilità del progetto, si ispira alla prima tipologia, relativamente meno complessa rispetto alle altre ma che richiede requisiti maggiori in termini di competenze linguistiche.

Inoltre ogni modello risulta adeguato ad un particolare campo di applicazione, in particolare i primi due ottengono risultati migliori se applicati a domini ben definiti, ovvero corpus di documenti omogenei, i cui documenti mostrano una certa coerenza per stile linguistico e lunghezza. Gli ultimi due modelli, invece, in virtù degli algoritmi automatici di apprendimento, si adattano più efficacemente all'analisi di contesti liberi, anche se la qualità dei risultati è strettamente correlata alla possibilità di fare affidamento su training set che coprano in buona percentuale la maggior parte dei casi linguistici riscontrabili.

Un'ulteriore differenza che caratterizza i primi due metodi rispetto agli ultimi due è la capacità di offrire in generale migliori risultati a discapito di un minor tasso di estrazione e di una maggiore staticità nell'evoluzione del sistema, infatti il miglioramento delle regole o dei parametri statistici richiede in modo imprescindibile l'intervento di un esperto del dominio che effettui le dovute analisi sui risultati ottenuti.

Nel caso specifico risultano quindi adeguate le scelte effettuate per l'applicazione OKKAM-POP, tuttavia per completare lo scenario disponibile nel prossimo paragrafo vengono delineate le caratteristiche principali di un sistema di estrazione riconducibile all'ultima tipologia, ovvero un sistema ad apprendimento automatico basato su metodi statistici.

5.1 Modello statistico per Information Extraction basato su Fuzzy C-Means

Per l'introduzione di questo modello statistico di estrazione delle informazioni occorre innanzitutto definire il concetto di *cluster analysis*.

Il *clustering* o analisi dei cluster è un insieme di tecniche di analisi multivariata dei dati[55] volte alla selezione e raggruppamento di elementi omogenei in un insieme di dati. Tutte le tecniche di *clustering* si basano sul concetto di distanza tra due elementi. Infatti la bontà delle analisi ottenute dagli algoritmi di *clustering* dipende essenzialmente da quanto è significativa la metrica, e quindi da come è stata definita la distanza.

La distanza è un concetto fondamentale, dato che gli algoritmi di *clustering* raggruppano gli elementi a seconda della distanza, e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme.

Le tecniche di *clustering* si possono basare principalmente su due filosofie:

- ✓ Dal basso verso l'alto (*Bottom-Up*): questa filosofia prevede che inizialmente tutti gli elementi siano considerati cluster a sé, e poi l'algoritmo provvede ad unire i *cluster* più vicini. L'algoritmo continua ad unire elementi al *cluster* fino ad ottenere un numero prefissato di *cluster*, oppure fino a che la distanza minima tra i *cluster* non supera un certo valore.
- ✓ Dall'alto verso il basso (*Top-Down*): all'inizio tutti gli elementi sono un unico *cluster*, e poi l'algoritmo inizia a dividere il *cluster* in tanti *cluster* di dimensioni inferiori. Il criterio che guida la divisione è sempre quello di cercare di ottenere elementi omogenei. L'algoritmo procede fino a che non ha raggiunto un numero prefissato di *cluster*.

La tecniche di *clustering* vengono utilizzate generalmente quando si hanno tanti dati eterogenei, e si è alla ricerca di elementi anomali. Per esempio, le compagnie telefoniche utilizzano le tecniche di *clustering* per cercare di individuare in anticipo gli utenti che diventeranno morosi. Normalmente questi utenti hanno un comportamento nettamente diverso rispetto alla maggioranza degli utenti telefonici, e le tecniche di

[55] Analisi statistica che considera le variabili rilevate su un insieme di unità statistiche a coppie o a gruppi al fine di evidenziarne le relazioni. L'analisi multivariata è una branca molto vasta della statistica che, da un punto di vista formale, esamina le distribuzioni multiple e che, operativamente, contempla diverse procedure di analisi.

clustering riescono spesso ad individuarli, o comunque definiscono un *cluster* dove vengono concentrati tutti gli utenti che hanno un'elevata probabilità di diventare utenti morosi.

Una prima categorizzazione degli algoritmi di *clustering* è effettuabile in base alla possibilità che ogni elemento possa o meno essere assegnato a più *clusters*. In base a questa definizione viene effettuata la distinzione fra *Hard clustering*, o *clustering esclusivo* e *Soft clustering*, o *clustering non-esclusivo*. Nel primo caso ogni elemento può essere associato ad esattamente un solo gruppo mentre nel secondo caso un elemento può appartenere a più *clusters* con diversi gradi di appartenenza.

Uno dei più noti algoritmi di *Soft clustering*, che verrà preso in considerazione per effettuare *Information Extraction*, è il *Fuzzy C-Means*.

Questo algoritmo, conosciuto anche con l'acronimo *FCM*, è stato sviluppato da Joe Dunn nel 1973 e migliorato da Bezdek nel 1981, ed è frequentemente utilizzato nell'ambito della *pattern recognition*[56] e si basa sulla minimizzazione della seguente funzione obiettivo.

Funzione obiettivo : $J_m = \sum_{i=}^{N} \sum_{j=}^{C} \left\| x_i - \right\|^2$, con $1 \le m < \infty$

N => cardinalità degli elementi del sistema

C => numero di *cluster*

u_{ij} => valore di appartenenza dell' i-esimo elemento al j-esimo *cluster*

x_i => i-esimo elemento del sistema

c_j => centro del j-esimo *cluster*

m => valore di *fuzziness* dei dati del sistema.

La definizione di *fuzziness* più nota a livello internazionale è "*the quality of being indistinct and without sharp outlines*", ovvero la proprietà, in questo caso di un dato, di essere inclassificabile e di non avere forti contorni di distinzione. Il valore *m* viene solitamente posto uguale a 2, tuttavia può assumere un valore qualsiasi nel seguente

[56] l riconoscimento di pattern è una sottoarea dell'apprendimento automatico che consiste nell'analisi e identificazione di pattern all'interno di dati grezzi al fine di identificarne la classificazione.

intervallo [1, ∞[. Per m = 1 il modello descritto si comporta come il classico algoritmo di *Hard clustering* K-Means, mentre per *m* che tende a infinito i contorni dei cluster, o meglio il concetto di appartenenza ad un cluster, assume un aspetto sempre più sfumato. Di seguito viene fornita la definizione matematica dei valori u_{ij} e c_j:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \times x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

Il modello *Fuzzy C-Means* gode inoltre di ulteriori proprietà che vengono riassunte di seguito.

$$u_{ij} \in [0,1] \qquad \forall \; j$$

$$\sum_{j=1}^{C} u_{ij} = 1 \qquad \forall$$

$$0 < \sum_{i=1}^{N} u_{ij} \leq N \qquad \forall$$

Tali proprietà sono conseguenze matematiche che derivano dalla definizione del modello. Ad esempio il valore di appartenenza di un elemento ad un determinato *cluster* essendo un valore normalizzato e conseguentemente la somma di tutti i valori di appartenenza di un certo elemento deve essere uguale alla probabilità totale. Inoltre la somma di tutti i coefficienti di appartenenza degli elementi di un certo *cluster* non può superare in valore il numero di elementi del *cluster*.

Di seguito viene descritto per passi l'algoritmo che permette di raggiungere a un sistema che implementa questo modello di raggiungere un punto di equilibrio che minimizza la funzione obiettivo rispetto a una determinata soglia.

Algoritmo di clusterizzazione :

1. Inizializzazione della matrice $U^{(0)}$ che contiene i valori iniziali di appartenenza di ciascun elemento a ciascun *cluster* del sistema.

$$U^{(k)} = \begin{bmatrix} u_{11} & \dots & u_{1j} \\ \dots & \dots & \dots \\ u_{i1} & \dots & u_{ij} \end{bmatrix} \qquad i \in [1, N],\ j \in [1, C]$$

   L'indice k rappresenta il passo attuale dell'algoritmo ed in fase di inizializzazione della matrice U assume valore uguale a 0. Il numero di righe della matrice corrisponde al numero di elementi del sistema statistico, mentre il numero di colonne corrisponde al numero di *clusters* considerati.

2. Al *k-esimo* passo: calcolare il vettore dei centri $C^{(k)}$ utilizzando la matrice $U^{(k)}$.

$$j \in [1, C] \qquad C^{(k)} = [c_1 \quad \dots \quad c_j]$$

3. Aggiornamento di $U^{(k)}$, $U^{(k+1)}$

4. Se $\left\| U^{(k+1)} - U^{(k)} \right\| < \varepsilon$, dove $\varepsilon \in [0,1]$ rappresenta il criterio di terminazione, allora l'algoritmo ha raggiunto un punto di equilibrio e si ferma, altrimenti occorre iterare il procedimento ripartendo dal punto 2.

5. L'algoritmo di *clusterizzazione C-Means* converge a un minimo locale, detto anche punto di sella della funzione obiettivo J_m. E' importante sottolineare che il costo computazionale dell'algoritmo *FCM* dipende dal criterio di terminazione. Più ε è piccolo maggiore sarà il numero di passi necessari per giungere al punto di stabilità. Inoltre differenti inizializzazioni possono determinare una differente evoluzione del algoritmo, potrebbero essere ad esempio necessari un numero di passi diverso per raggiungere la stabilità.

Ora che è stato definito in modo dettagliato il modello statistico se ne descrive l'impiego nell'ambito dell'*Information Extraction*. Dato che il progetto considerava come entità di riferimento le persone, le organizzazioni e i luoghi manteniamo questa ipotesi e si pensi a un'organizzazione in uno spazio tridimensionale degli attributi o dei concetti che descrivono tali entità, come mostrato in Figura 5.1.
Ogni elemento dell'insieme sarebbe caratterizzato da tre coefficienti di appartenenza ognuno corrispondente a uno dei tre *clusters* individuati, e tali coefficienti denoterebbero la posizione specifica di ciascun elemento all'interno dello spazio tridimensionale che sottende le linee tratteggiate in Figura 5.1.

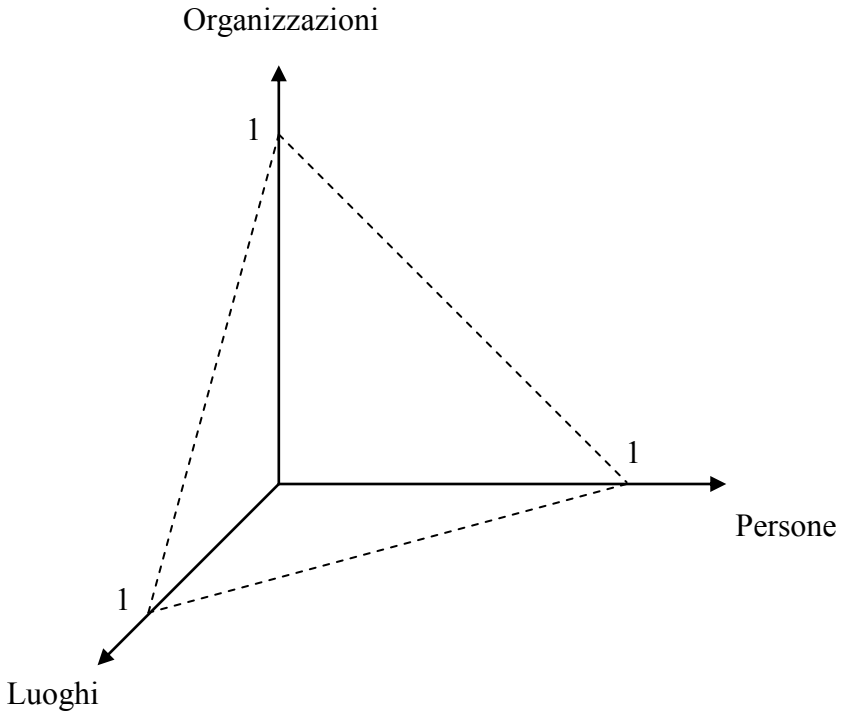

Figura 5.1 - Rappresentazione in uno spazio tridimensionale dei clusters.

I vertici individuati sugli assi cartesiani rappresentano i punti di massima discriminazione dei *clusters* ai quali è plausibile che tendano gli elementi dell'insieme dopo l'applicazione dell'algoritmo di clusterizzazione.
Come spiegato il primo passo da effettuare corrisponde all'inizializzazione della matrice $U^{(0)}$ e tale operazione potrebbe essere sfruttata utilizzando i dati estratti con

l'applicazione *OKKAM-POP*. Tale scelta è giustificata dal fatto che in essi è riscontrabile una discreta precisione e ciò aiuterebbe considerevolmente il modello statistico nell'avvicinarsi rapidamente ad un insieme di risultati il più realistico possibile.

Ad esempio gli elementi del modello statistico potrebbero essere rappresentati da sostantivi e aggettivi presenti all'interno di pattern sintattici che descrivono o caratterizzano differenti tipologie di entità.

Ad ogni elemento potrebbe essere inizialmente attribuito un valore di appartenenza a ciascun *cluster* in base al fatto che si trovi all'interno di un pattern estratto da *OKKAM-POP* che descrive una persona, un'organizzazione o un luogo e attraverso una funzione matematica che consideri la vicinanza di un elemento rispetto all'entità all'interno dello stesso pattern sintattico.

E' importante sottolineare che la realizzazione di tale sistema statistico non ha solo lo scopo di realizzare una segmentazione statica dei dati già estratti, ma potrebbe essere utilizzato in futuro per l'estrazione di nuovi termini e sarebbe in grado di adattarsi, o meglio di adattare il proprio punto di equilibrio, reagendo all'estrazione di termini già presenti nel sistema o di nuovi termini, a cui sarà possibile dare con una buona attendibilità un *cluster* di riferimento.

Tale modello ha quindi lo scopo di fornire un supporto alla decisione per comprendere nel modo più preciso possibile se l'entità che si trova in prossimità di particolari termini sia una persona, un luogo ed una organizzazione. Inoltre in base al valore di confidenza che ciascun sostantivo o aggettivo ha rispetto a ciascun *cluster* sarà possibile verificare ed estrarre concetti che possono aiutare a caratterizzare tale entità.

6 CONCLUSIONI

Il progetto affrontato presso la *Expert System* ha dato la possibilità di approfondire argomenti di estrema attualità: il *Web Semantico*, il *Natural Language Processing* e l'importazione automatica delle entità all'interno del repository *OKKAM* che rappresenta l'anello di congiunzione fra i primi due aspetti.

Lo sviluppo di queste tecnologie offre notevoli spunti di riflessione e assume un senso estremamente significativo nel panorama mondiale che vede coinvolti numerosi progetti atti a rendere operativa la tanto blasonata evoluzione del *Web 2.0* nel *Web 3.0*. Evoluzione che promette di offrire nuovi e innovativi modi di utilizzare la Rete, ma che tarda a manifestarsi a causa delle difficoltà tecniche incontrate lungo il suo percorso.

All'interno di questo progetto di ricerca si è tentato di dare una risposta concreta alle problematiche legate all'automazione dei processi di comprensione dei contenuti *Web* non strutturati. Le strade percorribili per il successo tecnologico del *Web Semantico* sono essenzialmente due:

1. L'intera comunità mondiale che utilizza il Web deve cambiare approccio nella pubblicazione dei contenuti adottando gli standard proposti dal *W3C* per la realizzazione di documenti *autodescrittivi*. Tali standard rendono i contenuti interpretabili automaticamente da agenti intelligenti. Queste entità software sono in grado sfruttare l'infrastruttura fornita dalle tecnologie *RDF/OWL* per creare relazioni non note a priori tramite tecniche di *automatic reasoning*.
2. La ricerca scientifica che copre i settori dell'*Intelligenza Artificiale* e del *NLP* deve essere in grado di fornire strumenti avanzati ed affidabili che permettano di effettuare in modo automatico tutte le operazioni altrimenti svolte manualmente. Queste operazioni sono ad esempio la marcatura dei contenuti o l'integrazione delle relazioni *RDF* all'interno delle pagine *Web*.

Queste due opportunità sono state volutamente esposte come gli estremi punti di vista su quello che è un problema univoco e ancora aperto. Il primo approccio è sicuramente quello più semplice sotto l'aspetto teorico ma purtroppo è irrealizzabile a causa della grande inerzia che la comunità mondiale oppone alla diffusione e all'uso degli standard e delle tecnologie già descritte. Dopotutto non è credibile uno scenario in cui tutti gli utenti, o almeno la maggior parte, impieghino tempo e risorse maggiori per fare

essenzialmente le stesse cose che attualmente si possono fare in meno tempo. Ciò non è sicuramente vero nel lungo termine, infatti il progressivo aumento di informazioni strutturate sul *Web* potrebbe agevolare in futuro gli utenti nella pubblicazione e ricerca di contenuti. Inoltre questo modo di affrontare la questione non risponde alla seguente domanda: "Come fare per utilizzare l'enorme quantità di dati, che in questi anni di vita fortunata del *Web*, sono stati pubblicati secondo le classiche metodologie prive di semantica?".

Il secondo approccio potrebbe in qualche modo offrire un risposta a questo quesito proponendo inoltre una soluzione semplice e quindi più appetibile per il problema legato alla pubblicazione di contenuti *autodescrittivi*. Questo dovrebbe essere il *Web semantico*, uno strumento semplice e flessibile come è stato in passato il *Web* tradizionale.

Occorre però accettare il seguente compromesso, operando con sistemi statistici automatici, come quello introdotto nel paragrafo 5.1, deve essere abbandonata l'idea di verità assoluta in cambio di un certo grado di incertezza delle informazioni.

Per fare un esempio concreto di quanto appena detto si pensi ad esempio ai tradizionali motori di ricerca basati su *keyword*. Essi rispondono attualmente alle richieste fornendo risultati con una totale certezza lessicale, ovvero assicurano che i *links* proposti contengano la parola o gli elementi di una frase inseriti come chiave di ricerca, ma forniscono poche indicazioni sulla certezza semantica, ovvero sul fatto che i risultati corrispondano a ciò che stavamo realmente cercando. I motori di ricerca semantici invece promettono, tramite strumenti *NLP*, di interpretare semplici richieste formulate con il linguaggio naturale e di fornire risposte coerenti al significato e non al lessico. Tali risposte non sono costituite esclusivamente da *links*, ma anche da informazioni che derivano dall'integrazione di sorgenti dati multiple e che contribuiscono insieme ad aumentare la qualità dei risultati. Alcuni motori di ricerca semantici noti sono *Hakia, SenseBot, Powerset, Cognition,* si consiglia inoltre di valutare le funzionalità offerte da innovativi servizi *Web* come *Wolfram Alpha* e *OpenCalais*, che fanno ben sperare in una imminente nascita di nuovi strumenti pubblici. Tuttavia se non fosse per questi esempi, non si potrebbe avere la percezione del cambiamento radicale del *Web*.

Dopo l'esperienza fatta posso affermare che l'*analisi linguistica automatica* è un punto chiave per la fornitura di nuovi servizi nella pubblicazione dei contenuti. Quello che si è

cercato di fare durante questo progetto è essenzialmente estrarre in modo automatico le proprietà principali, contenute all'interno di testi non strutturati, di persone, organizzazioni e luoghi. Queste tre tipologie di entità non devono essere considerate in modo distinto. Uno studio che purtroppo non è stato affrontato all'interno del progetto è quello delle relazioni che sussistono fra queste tre tipologie di entità.

L'operazione di descrizione delle entità presenti nel *repository OKKAM* è essenziale per aumentare il grado di discriminazione fra gli elementi in esso memorizzati. Concetti ben definiti in *OKKAM* corrispondono a un grado di *precision* e *recall* migliore delle risposte fornite da questa innovativa infrastruttura.

I risultati mostrati nel paragrafo 4.7 non sono fine a se stessi, sono stati perseguiti per una successiva loro importazione nell'architettura *OKKAM* tramite i servizi offerti di *Bulk importing*. Questo processo potrebbe fornire gli spunti per un nuovo progetto di ricerca che si occupi di verificare l'efficacia del progressivo popolamento del *repository OKKAM*.

In definitiva gli obbiettivi prefissati sono stati raggiunti, e l'approccio pratico con il quale si è cercato di affrontare le problematiche in ambito aziendale ha contribuito a fornire nuove competenze progettuali e sugli aspetti implementativi di un'applicazione *NLP*.

BIBLIOGRAFIA

Testi di riferimento

- ✓ Christopher Manning & Hinrich Schütze*, Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, Maggio 1999.
- ✓ Peter Jackson & Isabelle Moulinier, *Natural Language Processing for Online Applications Text Retrieval,Extraction and Categorization*, John Benjamins Publishing Co, Luglio 2002.
- ✓ Lucja M. Iwanska & Stuart C. Shapiro, *Natural Language Processing and Knowledge Representation*, AAAI Press, prima edizione 7 luglio 2000.
- ✓ Marie-Francine Moens, *Information Extraction: Algorithms and Prospects in a Retrieval Context*, Springer, 19 ottobre 2006.
- ✓ Anne Kao, Steve R. Poteet, *Natural Language Processing and Text Mining*, Springer, 14 novembre 2006.
- ✓ Scott Meyers, *Effective C++: 50 Specific Ways to Improve Your Programs and Design*, Addison-Wesley Professional, 2 settembre 1997.

Articoli scientifici

- ✓ Christian Bizer, Tom Heath, Kingsley Idehen, Tim Berners-Lee*, Linked Data on the Web*, Beijing, 21-25 aprile 2008.
- ✓ Tom Heath, *An Introduction to Linked Data*, Austin, Texas, 13-14 febbraio 2009
- ✓ Paolo Bouquet, Heiko Stoermer, Daniele Cordioli, *An Entity Name System for Linking Semantic Web Data*, 28 febbraio 2008
- ✓ Paolo Bouquet, Heiko Stoermer, Barbara Bazzanella, *An Entity Name System (ENS) for the Semantic Web*, 14 marzo 2008.
- ✓ Themis Palpanas, Junaid Chaudhry, Periklis Andritsos, Yannis Velegrakis, *Entity Data Management in OKKAM*, 30 maggio 2008.
- ✓ Sunita Sarawagi, *Information Extraction*, Mumbai, 29 novembre 2008.
- ✓ Luca Dini, *NLP Technologies and the Semantic Web: Risks, Opportunities andChallenges*, agosto 2008.
- ✓ Paul Buitelaar, Thierry Declerck*, Linguistic Annotation for the Semantic Web*, 2008.

- ✓ Linguistic Annotation for the Semantic Web, Linguistic Annotation for the Semantic Web, 13 giugno 2008.
- ✓ Gàbor Pròszeky, Morphological Analyzer as Syntactic Parser, 1 ottobre 2002.
- ✓ Rada Mihalcea, Courtney Corley, Carlo Strapparava, *Corpus-based and Knowledge-based Measures of Text Semantic Similarity*, 2006.
- ✓ Jay J. Jiang David W. Conrath, *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*, Taiwan, 1997.
- ✓ Felice dell'Orletta, Alessandro Lenci, Simone Marchi, Simonetta Montemagni, Vito Pirrelli, Giulia Venturi, *Dal testo alla conoscenza e ritorno: estrazione terminologica e annotazione semantica di basi documentali di dominio*, 23 dicembre 2008.
- ✓ Simonetta Montemagni, *Acquisizione automatica di termini da testi: primi esperimenti di estrazione e strutturazione di terminologia metalinguistica*, 16 gennaio 2008.
- ✓ Joe Zhou, Pete Dapkus, *Automatic Suggestion of Significant Terms for a Predefined Topic*, 3 settembre 2002.
- ✓ Sergio Bolasco, *Statistica testuale e text mining: alcuni paradigmi applicativi*, 2005.
- ✓ Ralf Krestel, Ren´e Witte, Sabine Bergler, *Fuzzy Set Theory-Based Belief Processing for Natural Language Texts*, 5 gennaio 2008.
- ✓ Bernardo Magnini, Carlo Strapparava, *Costruzione di una base di conoscenza lessicale per l'italiano basata su WordNet*, 28 luglio 2004.
- ✓ George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller, *Introduction to WordNet: An On-line Lexical Database*, agosto 1993.

Siti Web consultati

- ✓ http://www.expertsystem.it/
- ✓ http://www.okkam.org/
- ✓ http://it.wikipedia.org/
- ✓ http://www.websemantico.org/
- ✓ http://www.iptc.org/

- ✓ *http://portal.acm.org/*
- ✓ *http://cwl-projects.cogsci.rpi.edu/msr/*
- ✓ *http://nlp.stanford.edu/fsnlp/*
- ✓ *http://www.eclipse.org/*
- ✓ *http://wordnet.princeton.edu/*
- ✓ *http://www.elearninglab.eu/studying/sw/sw_program.html*
- ✓ *http://www.ilc.cnr.it/viewpage.php/sez=ricerca/id=834/vers=ita*
- ✓ *http://europa.eu/languages/it/chapter/17*
- ✓ *http://www.aclweb.org/*
- ✓ *http://groups.csail.mit.edu/medg/ftp/psz/k-rep.html*
- ✓ *http://www.information-management.com/issues/20040901/1009161-1.html*
- ✓ *http://www-ksl.stanford.edu/kst/what-is-an-ontology.html*
- ✓ *http://www.ittig.cnr.it/Ricerca/materiali/JurWordNet/DisambiguationOfPolysemies.htm*
- ✓ *http://online.sfsu.edu/~kbach/ambguity.html*
- ✓ *http://dissertations.ub.rug.nl/faculties/arts/2004/t.gaustad/*
- ✓ *http://www.chomsky.info/onchomsky/19720629.htm*