UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica

WordNet e sue applicazioni. Revisione e implementazione di un database di termini matematici.

Tesi di laurea di Federica Niero

Relatore: Chiar.ma prof. Laura Gilda Paccagnella

Anno Accademico 2005/2006

Ai miei genitori
e tutti coloro che mi hanno sostenuta

Indice

Introduzione

L'attività di ricerca e di sviluppo legata alle ontologie è stata, in questi ultimi anni, molto fervente, ricercando anche la condivisione della conoscenza e delle informazioni tramite l'utilizzo di strumenti già costruiti.

Base di molte ontologie lessicali per il linguaggio umano, come vedremo nel primo capitolo di questa tesi, è stata lo studio e l'implementazione di WordNet, dizionario lessicale per la lingua inglese sviluppato presso il Cognitive Scienze Laboratory dell'Università di Princeton. Di WordNet verrà data nel secondo capitolo la teoria su cui si basa e, nel capitolo 3, la struttura che ne contiene le informazioni.

Questo progetto ha generato altri progetti, che basandosi su di esso hanno portato alla realizzazione ad esempio di ontologie multilingui. EuroWordNet e MultiWordNet sono due realizzazioni di database multilingue che, come avremo modo di vedere nel capitolo 4, pur animate dallo stesso obiettivo, utilizzano due approcci completamente diversi.

Ulteriori applicazioni di WordNet sono delle ontologie lessicali basate su Domini Specifici, nel capitolo 5 avremo modo di vederne alcune realizzazioni nel panorama della lingua Italiana ed integrate ad EuroWordNet e MultiWordNet.

Nel panorama delle basi di dati specialistiche si colloca anche il dizionario di termini matematici sviluppato nell'anno 2002-2003 presso l'Università di Padova dall'Ing. Croin. La ristrutturazione del dizionario, l'aggiunta di alcuni elementi utili al lavoro di incremento delle voci in esso contenuto e al suo utilizzo sono stati introdotti con questo lavoro di tesi e saranno argomento del capitolo 7.

Capitolo1 - Le ontologie lessicali e WordNet

1.1 - Le ontologie lessicali

L’ontologia nasce ed è usualmente concepita come disciplina strettamente filosofica, lontana dal mondo dell’Information Technology.

In filosofia essa è una branca della metafisica e si riferisce allo studio dell’essere o dell’esistenza assieme alle sue categorie fondamentali.

Nell’ambito dell’Information Technology una ontologia è il tentativo di formulare uno schema concettuale esaustivo e rigoroso nell’ambito di uno specifico dominio.

La rete, e le comunicazioni che tramite essa avvengono, hanno reso strategici gli aspetti ontologici dell’informazione. Affinché sia possibile caratterizzare, reperire e organizzare le informazioni, diventa fondamentale il contenuto.

La definizione di ontologia più largamente accettata è quella di Tom Gruber: “an ontology is an explicit specification of a conceptualisation”.
Tralasciando un discorso molto ampio sulle ontologie, ci limiteremo a considerare le ontologie lessicali, ovvero quelle che caratterizzano un linguaggio, o più linguaggi o una loro parte.
Una ontologia lessicale è indipendente dal dominio ed esprime la semantica di elementi sintattici del linguaggio e la semantica di costrutti linguistici.
In termini schematici quello che si vuole esprimere con una ontologia lessicale sono le seguenti due caratteristiche:

1. una conoscenza lessicale, formata da un insieme di parole (intese come stringhe di caratteri)
2. una conoscenza semantica, che raccoglie in sé i significati delle parole e le relazioni che fra di esse intercorrono.

Come si vedrà nel paragrafo successivo, WordNet ricalca queste caratteristiche permettendo di recuperare non solo il significato di una parola (come nei dizionari classici) ma soprattutto fornisce relazioni basate sul significato, ovvero semantiche, con altre parole.

1.2 - WordNet

WordNet è un sistema di ricerca lessicale che si basa sulle attuali teorie psicolinguistiche formulate sulla linguistica umana. Sviluppato al Cognitive Science

Laboratory dell'Università di Princeton, e supervisionato dal Prof. George A.Miller oggi è alla versione 2.1.

A prima vista esso potrebbe sembrare un dizionario on line, in realtà viene sempre più spesso utilizzato come ontologia lessicale.

Un dizionario tradizionale è basato sulle procedure storiche di organizzazione delle informazioni lessicali. In esso le parole sono disposte alfabeticamente, per facilitarne la ricerca, ed i loro significati elencati tutti assieme in relazione all'uso più frequente. La ricerca può alle volte diventare tediosa ed il passo più semplice per renderla più veloce è stata quella di rendere disponibile il dizionario in formato on line.

In WordNet le informazioni sono memorizzate in base al loro significato ed alle loro categorie sintattiche e sono legate fra loro tramite diversi tipi di relazioni. WordNet divide il significato di parola in due concetti: la "*Word Form*", la forma scritta, e la "*Word Meaning*" ovvero il concetto espresso da tale parola. Quindi il punto d'inizio della classificazione delle parole secondo WordNet sono le relazioni che intercorrono fra lemma e significato.

La base della teoria sta nella Matrice Lessicale (Figura 1-1): nelle righe vengono elencati i significati delle parole e nelle colonne i lemmi. Ad esempio per la colonna relativa alla Word Form f*unction* i possibili Word Meaning, ovvero le righe della Matrice Lessicale, potrebbero essere: *mathematical relation*, *subroutine, religious cerimony*, quando *function* appartiene alla categoria lessicale dei nomi, oppure *operate* e *officiate*, nel caso sia utilizzata come verbo. La presenza di un valore non nullo di E_{ij} all'interno della matrice implica che la forma F_j viene espressa dal significato M_i. Se abbiamo più valori non nulli nella stessa riga siamo davanti a dei sinonimi: forme diverse hanno stesso significato; più valori non nulli nella stessa

colonna esprimono invece un concetto di polisemia: la stessa forma F_j ha più significati.

WordNet è quindi organizzato su Relazioni Semantiche che coinvolgono le relazioni fra significati (rappresentati, come vedremo, in synset) e su Relazioni Lessicali che stabiliscono le relazioni fra i singoli lemmi (le forme).

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 1-1 Matrice Lessicale di Word Net

1.3 - Estensioni: WordNet e Multilinguismo

Lo sviluppo di una ontologia linguistica contenente tutti i concetti immaginabili appartenenti ai diversi ambiti della conoscenza risulta molto complicato. La scelta di limitare i contenuti a termini generali e di uso comune, senza scendere in specializzazioni, rende l'ontologia di uso più flessibile per varie applicazioni.

Ciò nonostante il problema può essere risolto tramite estensioni delle ontologie stesse.

Una delle applicazioni di queste estensioni sono sicuramente le costruzioni di ontologie multilingui. Le naturali soluzioni al problema del multilinguismo possono essere le seguenti:

- Utilizzare una ontologia monolingue ed integrarla con delle traduzioni dei lemmi delle lingue diverse da quella standard
- Creare una ontologia multilingue

L'evoluzione ha dato ragione alla seconda delle ipotesi dando luogo però a due tipi di approccio: expand approach e merge approach.

L'***expand approach*** punta all'espansione dell'ontologia di base (WordNet) collegando fra loro synset di lingue diverse. Questa soluzione è più semplice ed efficiente dal punto di vista metodologico, compatibile con la struttura base e permette di collegare molte altre risorse.

Il ***merge approach*** mira a creare differenti wordnets, ognuno per ogni linguaggio specifico, ed allinearli in un secondo momento con WordNet attraverso la generazione di opportune traduzioni. Questo metodo si presenta come più complesso e più laborioso. La struttura è diversa da quella di WordNet ma si possono mantenere le caratteristiche specifiche di ogni linguaggio.

1.4 - Estensioni : WordNet e domini specifici

WordNet e gli altri database lessicali per lingue diverse dall'inglese si mantengono appositamente ad un livello di conoscenza generale. Database specifici invece si concentrano su una certo dominio, costruendo gerarchie fra concetti ad alta specializzazione con un uso limitato di relazioni linguistiche e lessicali. Sono stati implementati numerosi database lessicali per la lingua Italiana: **Economic WordNet** per il campo dell'economia, **Jur-WordNet** per il campo giuridico, **ArchiWordNet** per l'Architettura e il **Maritime-WordNet** per termini legati alla navigazione e al commercio marittimo .

In questo ambito si colloca il **database di termini matematici** sviluppato all'interno del gruppo di lavoro NLP presso il dipartimento di Matematica dell'Università di Padova. L'analisi e l'arricchimento di questo dizionario sarà argomento di questa tesi.

1.5 - Applicazioni di WordNet

WordNet è diventato il tool ideale in molti campi di ricerca grazie al fatto di essere open source, di essere ben documentato e di avere grande potenzialità nell'ambito del NLP (Natural Language Processing). Potremmo dividere lo sviluppo di WordNet in due grandi rami: da una parte l'implementazione e arricchimento dello strumento stesso, dall'altra il suo utilizzo da parte dei sistemi NLP.

1.5.1 Implementazione e arricchimento dello strumento

Per quanto riguarda il primo dei due ambiti si possono identificare due indirizzi principali: arricchimento del software, WordNet multilingue e domain specific.

1.5.1.1 Studi per l'arricchimento del software

Oltre che in termini di volume di lemmi contenuti nella banca dati stessa, si è pensato anche all'arricchimento della struttura stessa di WordNet. VerbNet ne è un esempio, implementato per riparare ad una mancanza di relazioni associative fra verbi e Lingua, una interfaccia per WordNet che contiene anche categorie di metonimia che non erano state ancora implementate.

1.5.1.2 WordNet Multilingue e Domain Specific

Anche se già citati in precedenza ci limitiamo a citare due grandi progetti nell'ambito dello sviluppo di wordnets per lingue europee: EuroWordNet e Multiwordnet. La tesi si svilupperà soprattutto in questo campo, illustrando i due progetti multilingui, cercando di illustrare alcune esperienze di WordNet per domini specifici ed infine approfondendo il WordNet sviluppato per il dominio matematico.

1.5.2 Applicazioni dello strumento

Le applicazioni che utilizzano WordNet nell'ambito di NLP sono le seguenti:

1.5.2.1 Information Retrival e Extraction

Queste operazioni sono strettamente legate all'organizzazione e rappresentazione del sapere nella rete di Internet. Una delle branche della ricerca è l'applicazione degli strumenti di intelligenza artificiale alla ricerca delle informazioni. WordNet viene usato come strumento di conoscenza linguistica atto a rappresentare ed interpretare il significato dell'informazione. Una sorta di aggiunta di "semantica" al processo di ritrovamento/caratterizzazione dell'informazione.

1.5.2.2 Disambiguation

La risoluzione di ambiguità semantiche dei termini consiste nel determinare in maniera automatica il significato più appropriato di una parola in base al contesto nel quale essa si trova. WordNet in quanto rete semantica viene utilizzato come strumento per determinare il senso di una parola basandosi sulle relazioni semantiche fra i "vicini" della parola stessa.

1.5.2.3 Distanza Semantica

Il concetto di distanza semantica è direttamente connesso a quello di similarità semantica che fornisce una misura della similarità semantica fra due lemmi . Si può affermare che il calcolo della similarità semantica fra due lemmi si basa sulla lunghezza del cammino necessario a percorrere la distanza che li separa dal concetto minimo comune.

Capitolo 2 – Genesi di WordNet

2.1 - Introduzione

WordNet nasce nel 1985 come risultato di un progetto al quale hanno partecipato linguisti e psicologi dell'Università di Princeton. Dall'idea iniziale di fornire una ulteriore risorsa on line rispetto ad una semplice ricerca di tipo alfabetico, si è arrivati ad un vero e proprio dizionario basato sui principi della psicolinguistica.

La differenza più lampante fra WordNet ed un dizionario classico è che il primo divide il lessico in 5 categorie sintattiche: sostantivi (o più banalmente nomi), verbi, aggettivi, avverbi e function words. Attualmente WordNet non tratta l'ultima delle categorie elencate. Come diverrà chiaro in seguito i sostantivi sono organizzati in una memoria lessicale come gerarchie di specializzazione (iperonimi/iponimi); i verbi sono strutturati in una gerarchia mediante la relazione di troponimia (un verbo è troponimo di un altro verbo quando esprime una particolare azione, per es. camminare/muoversi). Gli aggettivi dal canto loro si presentano invece come iperspazi n-dimensionali.

Nei prossimi paragrafi descriveremo nel dettaglio le teorie linguistiche sulle quali WordNet si fonda. Partiremo dalla Matrice Lessicale già vista nel Capitolo 1 che ne è la base.

2.2 - Forme di Parola e Significati

La semantica lessicale parte dal concetto chiave che una parola è una associazione fra l'espressione della parola (forma e pronuncia) ed i concetti che la parola stessa esprime. La corrispondenza fra la forma di parola ed il concetto che esprime sono rappresentate come abbiamo già visto nella matrice lessicale. Inoltre sinonimia e polisemia sono aspetti complementari della Matrice Lessicale. Le relazioni sono del tipo "molti a molti" e i concetti di polisemia e sinonimia possono essere facilmente compresi se si pensa ai processi mentali: un lettore (o una persona che ascolta) deve scontrarsi con la polisemia ovvero deve scegliere quale significato associare alla forma di parola che ha recepito e, a sua volta, colui che scrive (o parla) deve decidere quale forma rende in maniera appropriata il significato che intende esprimere.
Pariteticamente la matrice lessicale potrebbe essere rappresentata, in un diagramma, da bue blocchi con due frecce che partendo da questi vanno in entrambe le direzioni. I due blocchi saranno chiamati rispettivamente "Word Meaning" e "Word Form". Le due frecce indicheranno che colui che parla può partire da un significato "Word Meaning" e cercare una forma adeguata ad esprimerlo, oppure partendo da una "Word Form" ricercarne il giusto significato.
Da quanto visto WordNet quindi si è posto l'obiettivo di voler esprimere due diverse relazioni:

1. relazioni semantiche fra i significati
2. relazioni lessicali fra le forme.

La costruzione della base di dati si è scontrata con l'esistenza di due teorie: la teoria costruttiva e la teoria differenziale.

Secondo la teoria costruttiva un'accurata costruzione di un concetto deve essere supportata da un numero sufficiente di informazioni. Tali informazioni devono consentire di caratterizzarlo in modo da poterlo distinguere da altri possibili concetti lessicali e di fornirne una corretta definizione.

La teoria differenziale, molto meno rigida, esprime il fatto che la rappresentazione di un concetto possa essere fatta solo con elementi che permettano di distinguerlo da altri.

Per essere più chiari possiamo ricorrere ad un esempio: la parola ***pianta***.

Essa può avere i seguenti significati:

- *nome generico che indica qualsiasi vegetale fornito di organi specializzati*
- *proiezione orizzontale di un oggetto*
- *parte inferiore del piede*

Nella teoria costruttiva per differenziare i due significati dobbiamo fornire abbastanza informazioni in modo da distinguerli. In quella differenziale basta fornire una lista di forme che lo possano esprimere. Il significato M può essere espresso con una lista di forme (F1, F2, …). In questo modo abbiamo per ogni significato una lista di forme fra di loro in relazione di sinonimia. L'insieme viene indicato appunto come *Synonym Set*, o meglio conosciuto, nella sua forma contratta, *Synset*.

Ritornando al nostro esempio, per distinguere i due significati sarebbe stato sufficiente citarne due sinonimi: *vegetale* per il primo, e *mappa* per il secondo. Nel caso in cui non esista un sinonimo appropriato a differenziare quel significato da altri, si fa ricorso ad una glossa ovvero una breve spiegazione del significato. Per il terzo significato del nostro esempio si potrebbe utilizzare la glossa: *parte inferiore del piede*.

2.3 - Le relazioni alla base di WordNet

WordNet si basa sulle relazioni semantiche fra concetti, fra le quali la sinonimia gioca un ruolo fondamentale, ma non è l'unica ad essere utilizzata per costruirlo.
Le relazioni che descriveremo saranno le fondamentali, anche se non le uniche implementate da WordNet:

- *Sinonimia*
- *Antonimia*
- *Iponimia/Iperonimia*
- *Meronimia/Olonimia*
- *Relazioni Morfologiche*

2.3.1 - Sinonimia

La relazione di Sinonimia, come vedremo, ha una definizione, attribuita a Leibniz, molto rigida.

Definizione 1: Due concetti sono fra loro sinonimi se la sostituzione di uno con l'altro non cambia il valore di verità della frase nella quale viene fatta la sostituzione.

In base a questa definizione, due parole fra di loro legate da una relazione di sinonimia sono piuttosto rare. Si utilizza quindi una definizione più debole, non più legata alla frase ma al contesto a cui si fa riferimento.

Definizione 2: Due concetti sono fra loro sinonimi in un contesto linguistico C se la sostituzione di un concetto con l'altro nel contesto C non ne altera il valore di verità.

Così la sostituzione della parola "*pianta*" con "*mappa*" non ne altera il significato in topografia, ma in altri contesti la sostituzione potrebbe essere del tutto inappropriata.
Oltre a questa definizione in termini di vero/falso, esiste un'altra scuola filosofica di pensiero secondo la quale i sinonimi possono essere pensati anche secondo un concetto di similarità. Così una relazione di similarità semantica è sufficiente a rendere due concetti fra loro sinonimi.
Ritornando sempre al nostro esempio: *albero*, *arbusto* e *pianta* secondo le definizioni date in precedenza non possono essere fra loro sostituite, in quanto esistono anche piante che non sono necessariamente alberi o arbusti, ma secondo una definizione di similarità essi appartengono allo stesso synset, in quanto consentono la distinzione da altri significati.

2.3.2 - Antonimia

Una relazione alla quale non sempre è facile dare una definizione, ma che risulta molto familiare è l'antonimia.

L'antonimo di una parola *x* viene definito quasi sempre come *not-x.* Così ricco e povero sono fra loro antonimi anche se essere non ricchi non implica necessariamente essere poveri: infatti, si può essere né ricchi, né poveri.

2.3.3 - Iponimia/Iperonimia

Le relazioni di iponimia e iperonimia sono relazioni fra significati secondo la definizione seguente:

Definizione 3: Un concetto rappresentato dal synset {x1, x2, x3,…} viene detto iponimo del concetto rappresentato dal synset {y1, y2,…} se si può accettare una frase costruita come: *Un x è un (un tipo di) y.*

L'iponimia è transitiva ed è antisimmetrica; essa genera una struttura semantica gerarchica secondo la quale gli iponimi (concetto figlio) stanno sotto il proprio iperonimo (concetto padre).

2.3.4 - Meronimia/Olonimia

Essa è una relazione semantica, ed esprime il concetto di *parte di.*

Definizione 4: {x1, x2,…} è un meronimo di un concetto rappresentato da {y1, y2,…} se si possono accettare frasi scritte come *x è parte di y*.

La relazione di meronimia è transitiva (con le riserve che dopo spiegheremo) e antisimmetrica e può anch'essa essere usata per costruire relazioni gerarchiche.
Un esempio dell'applicazione del sistema gerarchico può essere dato da: *becco* ed *ala* sono meronimi di *uccello, canarino* è iponimo di *uccello*.
Quindi per il sistema gerarchico, *becco* e *ala* sono essi stessi anche meronimi di *canarino*.
L'applicazione della transitività invece ci porta a dire che se *dita* è meronimo di *mano* e *mano* è meronimo di *arto,* allora *dita* è meronimo di *arto,* ovvero le *dita* sono parte della *mano* e quindi anche dell'*arto*.
Questo non è sempre vero è dipende dal tipo di relazione *parte di* che si instaura fra le parti.
Per chiarire quanto affermato ci avvarremo di un esempio: *maniglia* è meronimo di *porta, porta* è meronimo di *casa,* dire che *maniglia* è meronimo di *casa* non risulta certo molto appropriato. Risulterebbe abbastanza sbagliato affermare che *maniglia è parte di casa*.
Si intuisce quindi che esistono varie tipologie di relazioni *parte di*, per l'esattezza ne sono state individuate 6:

- componente/oggetto
- elemento/insieme
- porzione/intero
- materiale/oggetto
- azione/attività
- località/area

e un ultimo aggiunto più tardi nel tempo:

- fase/processo

2.3.5 - Relazioni Morfologiche

Una importante classe di relazioni lessicali sono quelle morfologiche fra forme di parola.

Un esempio è quello del plurale dei nomi. Se un utente inserisce la parola *trees*, e lancia la ricerca, il programma non dovrebbe dare come risultato l'assenza della parola del DataBase. Se la parola *trees* è in relazione morfologica con il suo singolare, la ricerca viene fatta sul termine *tree*.

Per evitare inutili ridondanze, nella costruzione di WordNet, gli implementatori si resero conto della necessità di introdurre nel software una parte che gestisse le relazioni morfologiche. Il lavoro non si presentò comunque così semplice, per le numerose forme irregolari di nomi e verbi.

2.4 - I nomi in WordNet

La definizione comune di un nome è in genere data attraverso un termine più generale che lo descrive (iperonimo) e dall'elenco di caratteristiche che lo distinguono da altri.

La relazione di iponimia introduce quindi il concetto di ereditarietà. Un figlio eredita dal padre tutte le caratteristiche aggiungendone altre che lo classificano rispetto a questo e che lo distinguono dagli altri figli. Così ad esempio un ciliegio è un albero e si distingue dagli altri alberi per la durezza del legno, il fatto che produce frutti, la forma delle foglie, il tipo di radici, ecc.

Questa relazione è la base della strutturazione dei nomi in WordNet.

2.4.1 - Il sistema gerarchico

La relazione di iperonimia/iponimia è la relazione sulla quale gli studiosi basano la teoria sul sistema di memoria semantica. Esso è di tipo gerarchico e può essere schematizzato come un albero (nel senso grafico del termine).

La proprietà fondamentale di un albero è che un cammino dalla radice alle foglie non debba mai essere un loop. L'albero viene costruito seguendo la catena di termini in relazione di iponimia.

La struttura creata è una sequenza di livelli che va da molti termini specifici al livello più basso a pochi termini generici a livello più alto.

Con questo sistema si ovvia al problema della ridondanza, soprattutto per basi di dati che contengono molti termini: il termine al livello *n* ha tutte le proprietà del termine

al livello *n-1* ad esso collegato e ne aggiunge delle altre. E così via scendendo o salendo nella scala gerarchica. Basta quindi memorizzare solo le informazioni che caratterizzano l'oggetto stesso, mentre si possono tralasciare quelle che già sono memorizzate per l'oggetto padre.

A questo punto per capire meglio come siano organizzati i nomi in WordNet, risulta utile introdurre i simboli puntatore utilizzati per rappresentare le relazioni fra lemmi, anche se più avanti si darà una caratterizzazione più completa della base di dati.

Per le relazione di iperonimia/iponimia si utilizzano i simboli @ e ~: se W_h @-> W_s allora esiste la relazione inversa W_s~->W_h. Ciò significa che se la parola W_s è un iperonimo del nome W_h allora W_h è un iponimo di W_s.

La relazione semantica @-> indica una generalizzazione, ovvero si va da un termine specifico ad uno più generale. La relazione inversa ~->, invece indica una specializzazione, da un termine generico si va ad un termine specifico.

Per fare un esempio, vediamo come potrebbe essere il synset della parola albero:

{albero, pianta, @ conifera, ~….}

Qui vediamo che pianta è un iperonimo di albero, equivale quindi a scrivere: albero@->pianta. A sua volta nel synset relativo alla parola pianta dovremmo trovare rappresentata la relazione inversa pianta~->albero.

Il synset della parola pianta potrebbe essere:

{pianta, flora, @ organismo, @ albero, ~…}

i puntini … indicano semplicemente che la lista di relazione potrebbe continuare con altri termini ed i relativi puntatori.

2.4.2 - Le radici del sistema gerarchico

Secondo la teoria del sistema gerarchico, la costruzione dell'albero dovrebbe partire da un'unica gerarchia. Se così fosse, il livello più generico, la radice, sarebbe semanticamente pieno. In linea di principio si potrebbe mettere come radice un termine astratto come {entità} e mettere {oggetto, cosa} e {idea} come suoi immediati iponimi. In pratica questo porta a pochissimo contenuto semantico.

L'alternativa è quella di partizionare i nomi in un insieme di synset primitivi, selezionando un numero relativamente piccolo di concetti generici da utilizzare come radici semantiche per costruire altrettante gerarchie separate.

In Tabella 2-1 vengono elencati i 25 synset primitivi; all'interno degli alberi generati da tali radici devono essere contenuti tutti i vocaboli appartenenti alla categoria dei nomi della lingua inglese.

Le gerarchie create variano in grandezza (in termine di vocaboli contenuti) e non sono fra loro mutuamente esclusive: sono presenti anche riferimenti incrociati. Una volta scelti i concetti primitivi, ci si è accorti che fra di essi esistevano delle relazioni.

{act, action, activity}	*{natural, object}*
{animal, fauna}	*{natural phenomenon}*
{artifact}	*{person, human being }*
{attribute, property}	*{plant, flora}*
{body, corpus}	*{possession}*
{cognition, knowlwdge}	*{process}*
{communication}	*{quantity, amount}*
{event, happening}	*{relation}*
{feeling, emotion}	*{shape}*
{food}	*{state, condition}*
{group, collection}	*{substance}*
{location, place}	*{time}*
{motive}	

Tabella 2-1 I synset primitivi

Una schematizzazione della relazione più evidente è rappresentata in Figura 2-1. Queste relazioni vengono mantenute in un file “Tops” all’interno di WordNet.

In linea di principio non esiste limite in termini di numero di livelli del sistema gerarchico relativi ad ogni radice.

In pratica si può notare che il loro numero può variare a seconda del synset primitivo. Comunque alle volte si può andare anche oltre i dieci livelli, oltre i quali si entra nella specificità tecnica, che spesso non fa parte del vocabolario di tutti i giorni.

{plant, flora}

{living thing, organism} *{animal, fauna}*

{person, human being}

{thing, entity}

{natural object}

{artifact}

{non-living thing, object} *{substance}*

{food}

Figura 2-1 Relazioni fra concetti primitivi

2.4.3 - Le caratteristiche di distinzione

Come si può intuire facilmente, sebbene la struttura gerarchica generale venga costruita attraverso la relazione di iponimia, nel dettaglio, le caratteristiche proprie di un concetto sono quelle che lo distinguono da un altro.

Ogni concetto può essere distinto da un altro essenzialmente per tre caratteristiche:

1. attributi
2. parti
3. funzioni

Per rappresentarli sono necessarie anche relazioni incrociate fra categorie lessicali diverse. Esse saranno del tipo:

1. nome / aggettivo
2. nome / nome (meronimia)
3. nome / verbo

La prima e terza relazione non sono implementate. Mentre la seconda è la relazione di meronimia.

2.4.4 - Parti - Meronimia

I tipi di meronimia presi in considerazione nello sviluppo dei nomi in WordNet sono tre:

1. *componente/oggetto*: il componente è parte dell'oggetto, da questo può esserne separato ed ha una specifica funzione (es. ruota/bicicletta)
2. *elemento/insieme*: l'elemento appartiene all'insieme, appartenenza intesa come nell'omonimo concetto matematico ed è separabile dal tutto (es. albero/foresta)
3. *materiale/oggetto*: il materiale è parte integrante e non separabile dell'oggetto (es. vetro/bicchiere)

Il primo di questi tipi è statisticamente il più frequente.
Un'ultima considerazione sulla meronimia si riferisce alla relazione *materiale/oggetto*: con le recenti tecnologie, si è in grado scomporre un oggetto in elementi mano a mano più piccoli, fino ad arrivare alla particella più piccola finora conosciuta, l'atomo. La parola atomo è quindi, secondo la definizione che abbiamo

dato, meronimo di ogni parola che rappresenta un oggetto concreto. Il limite, dato dal buonsenso, è quello per il quale la relazione *parte di* serve a distinguere questo oggetto da altri con i quali potrebbe essere confuso.

2.4.5 - Antonimia

Come abbiamo già introdotto nel Cap. 2.3.2 la relazione di antonimia è una relazione lessicale fra parole.
Essa non è fondamentale nel sistema di relazioni rappresentate in WordNet, ma viene comunque esplicitata.
Un esempio di nomi in relazione di antonimia è la coppia uomo/donna oppure fratello/sorella e altri termini utilizzati per descrivere le parentele famigliari.

Abbiamo così esaurito le relazioni utilizzate in WordNet per i nomi, la Figura 2-2 ci permette di dare una rappresentazione grafica abbastanza chiara di quale possa essere il risultato della loro applicazione.

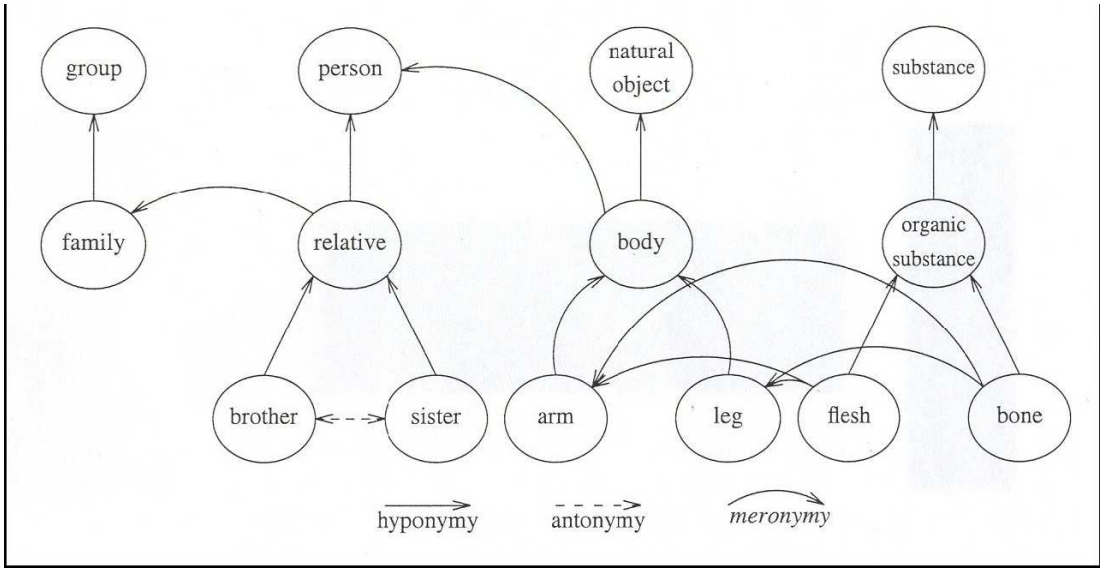


Figura 2-2 Rappresentazione di relazioni semantiche

2.5 - Gli aggettivi in WordNet

La trattazione degli aggettivi sarà meno particolareggiata, in quanto molti concetti sulle relazioni sono già stati ampiamente spiegati nei paragrafi precedenti.

La categoria semantica degli aggettivi viene divisa, secondo i dettami di WordNet in tre grandi gruppi:

- Aggettivi descrittivi
- Aggettivi Reference-Modifying
- Aggettivi relazionali

2.5.1 - Aggettivi descrittivi

Gli aggettivi descrittivi sono quelli ai quali di solito si pensa più spesso. Ciò che essi specificano è il valore di una caratteristica associata ad un nome, per esempio: un pacco *pesante*. Con l'aggettivo pesante si intende dare un valore alla caratteristica "peso" del pacco.

La caratterizzazione è quindi completamente diversa da quella utilizzata per la descrizione dei nomi: nessuna relazione di iponimia/iperonimia e quindi nessun sistema di tipo gerarchico. Possiamo immaginare una schematizzazione degli aggettivi come uno spazio dimensionale più che come un albero.

La relazione fondamentale è l'antonimia.

2.5.2 - Aggettivi Reference-Modifying

Essa è una categoria molto piccola di aggettivi. Per spiegarla partiamo introducendo un esempio: "quello è un mio *vecchio* amico". Per l'aggettivo *vecchio* abbiamo due possibili interpretazioni: una prima riguarda l'età anagrafica dell'amico (referent) una seconda si riferisce all'amicizia (reference), "di vecchia data", ed è appunto la tipologia in esame.

2.5.3 - Aggettivi Relazionali

La loro caratteristica principale è quella di derivare dai nomi. Possiamo citare alcuni esempi: atomico/atomo, fraterno/fratello, musicale/musica, nasale/naso. Per essi la relazione di antonimia non si applica mentre viene mantenuto un puntatore al nome da cui essi derivano.

2.6 - I verbi in WordNet

I verbi sono organizzati in WordNet secondo uno schema gerarchico. Nella categoria vengono inclusi anche i "phrasal verbs" (*stare per, accingersi a,…*).

La loro organizzazione è basata su una suddivisione in 15 files diversi: 14 files relativi a 14 gruppi semantici contengono i verbi che denotano azioni o eventi, un file per verbi che denotano stati.

Le relazioni fra verbi si basano su implicazione e opposizione. Per quanto riguarda l'implicazione, il tipo a cui si fa più spesso riferimento è la relazione di troponimia, che è la corrispondente all'iponimia per la categoria lessicale dei nomi. Si parla di troponimia fra due verbi quando l'uno è un "modo" dell'altro. Ad esempio *zoppicare* è un modo di *camminare.*

L'informazione memorizzata per i verbi è del tipo predicato-argomento. Infine per ogni verbo nel synset vengono inserite una o più "frames" che indicano le frasi in cui possono comparire.

Capitolo 3 - Struttura del DataBase WordNet e algoritmi di ricerca

3.1- Introduzione

Il metodo di lavoro utilizzato per l'implementazione di WordNet consiste di due parti fra loro distinte, che possono essere comparate alle due operazioni di scrittura e di stampa di un dizionario.

Da una parte si trattò di scrivere i Source Files che contenessero la base di dati lessicale, il contenuto di tali file costituisce la sostanza di WordNet, e possono essere arricchiti nel tempo con l'aggiunta di ulteriori lemmi.

Dall'altra, si trattò di implementare i programmi che accettassero in input la parola ricercata dall'utente e, utilizzando i Source File, fornissero a video il risultato del lavoro di ricerca.

In realtà le due parti divennero quattro:

1. la scrittura dei Source Files
2. il codice per trasformare questi file nel Database WordNet
3. il Database vero e proprio

4. gli strumenti software per accedere al Database

Come si avrà modo di notare nei paragrafi che seguono, in WordNet ci si può ricondurre ad una forma di parola tramite quattro specifiche:

1. la rappresentazione ortografica, o sintassi della parola
2. la categoria sintattica di appartenenza: nomi, verbi, avverbi, aggettivi
3. il campo semantico
4. il sense number (il numero che identifica il significato)

L'insieme di questi dati, forma la "chiave" che identifica univocamente ogni forma di parola nel database.
Quanto illustrato in questo capitolo è stato tratto dalla documentazione inclusa in WordNet 2.1, nella directory ...*WordNet\2.1\doc* (path relativo).

3.2 - L'indice di familiarità

Uno dei risultati fondamentali a cui si è arrivati nello studio della psicolinguistica e dei processi mentali lessicali, è il fatto che alcune parole sono molto più familiari di altre. La familiarità di una parola influenza in maniera preponderante almeno quattro variabili:

- La velocità di lettura
- La velocità di comprensione del testo
- La facilità di ricordare la parola

- La probabilità di riutilizzo della parola

Gli effetti dell'applicazione del concetto di familiarità sono molto importanti e sono stati presi in considerazione anche in WordNet.

La definizione più immediata di familiarità si basa su una relazione di proporzionalità diretta con la frequenza d'uso di una parola: più una parola viene utilizzata, ad esempio in un testo, più il suo indice di familiarità sarà alto. Tramite questa definizione si potrebbe, dato un testo di riferimento, calcolare la frequenza di una certa parola ed in base a questa definire l'indice di familiarità.

Questo genere di approccio risultò impensabile per uno strumento esteso come WordNet.

Fortunatamente una definizione alternativa, data da Zipf nel 1945, si basa sulla considerazione che la frequenza di utilizzo è direttamente legata alla polisemia.

Ciò equivale ad affermare che, in media, le parole con maggior frequenza di utilizzo, sono quelle che hanno un maggior numero di significati diversi in un dizionario. Invece di usare il concetto di proporzionalità alla frequenza di occorrenza in un testo come indice di familiarità, WordNet utilizza la polisemia.

Ciò che si può fare per calcolare l'indice di familiarità è abbastanza semplice. Basta prendere in considerazione un qualsiasi dizionario ondine e assegnare ad ogni parola un indice che valga 0 (zero) se la parola non compare nel dizionario, e un numero, maggiore di 1, pari al numero di significati diversi assunti dalla parola nel dizionario.

Un esempio di come può essere utilizzato l'indice di familiarità è rappresentato dalla parola *bronco* (inglese), il seguente elenco rappresenta le parole in relazione di iperonimia e accanto, fra parentesi tonde, il contatore di polisemia:

bronco (1)

=> *mustang (1)*

=> pony (5)

=> horse (14)

=> *equine (0)*

=> *odd-toed ungulate (0)*

=> *placental (0)*

=> *mammal (1)*

=> *vertebrate (1)*

=> *chordate (1)*

=> animal (4)

=> organism (2)

=> entity (3)

Se si omettono i termini con indice di familiarità pari a zero (0) o uno (1), indicati sopra in carattere corsivo, si ottiene la seguente lista di iperonimi:

bronco (1)

=> pony (5)

=> horse (14)

=> animal (4)

=> organism (2)

=> entity (3)

Il risultato ottenuto è più simile a quello che un normale utente che interroga WordNet si aspetta inserendo la parola *bronco*, infatti i termini omessi appartengono ad una terminologia tecnica, e non d'uso comune.

WordNet utilizza come dizionario di riferimento il *Dictionary of the English Language* di *Collins* per individuare, e memorizzare, il numero di significati associati ad ogni parola, relativamente ad ogni categoria sintattica (nomi, verbi, avverbi o aggettivi).

3.3 - I Source Files

I Source Files sono il prodotto di una dettagliata analisi semantica: vengono rappresentate una grande varietà di relazioni semantiche e lessicali per riprodurre il complesso sistema di conoscenza lessicale.

WordNet organizza nomi, verbi, aggettivi e avverbi in synset che vengono esplicitati nei Source Files secondo la loro categoria sintattica ed altri criteri.

I nomi e i verbi vengono divisi in base al campi semantici, gli avverbi vengono mantenuti in un solo file, gli aggettivi sono divisi in due file: uno per gli aggettivi descrittivi, uno per gli aggettivi relazionali.

I Source Files contenuti nell'ultima versione scaricabile di WordNet (ver. 2.1), secondo la documentazione allegata, sono elencati in Tabella 3-1:

File Number	**Name**	**Contents**
00	adj.all	all adjective clusters
01	adj.pert	relational adjectives (pertainyms)
02	adv.all	all adverbs
03	noun.Tops	unique beginner for nouns
04	noun.act	nouns denoting acts or actions

05	noun.animal	nouns denoting animals
06	noun.artifact	nouns denoting man-made objects
07	noun.attribute	nouns denoting attributes of people and objects
08	noun.body	nouns denoting body parts
09	noun.cognition	nouns denoting cognitive processes and contents
10	noun.communication	nouns denoting communicative processes and contents
11	noun.event	nouns denoting natural events
12	noun.feeling	nouns denoting feelings and emotions
13	noun.food	nouns denoting foods and drinks
14	noun.group	nouns denoting groupings of people or objects
15	noun.location	nouns denoting spatial position
16	noun.motive	nouns denoting goals
17	noun.object	nouns denoting natural objects (not man-made)
18	noun.person	nouns denoting people
19	noun.phenomenon	nouns denoting natural phenomena
20	noun.plant	nouns denoting plants
21	noun.possession	nouns denoting possession and transfer of possession
22	noun.process	nouns denoting natural processes
23	noun.quantity	nouns denoting quantities and units of measure
24	noun.relation	nouns denoting relations between people or things or ideas
25	noun.shape	nouns denoting two and three dimensional shapes
26	noun.state	nouns denoting stable states of affairs
27	noun.substance	nouns denoting substances
28	noun.time	nouns denoting time and temporal relations
29	verb.body	verbs of grooming, dressing and bodily care
30	verb.change	verbs of size, temperature change, intensifying, etc.
31	verb.cognition	verbs of thinking, judging, analyzing, doubting
32	verb.communication	verbs of telling, asking, ordering, singing
33	verb.competition	verbs of fighting, athletic activities
34	verb.consumption	verbs of eating and drinking
35	verb.contact	verbs of touching, hitting, tying, digging
36	verb.creation	verbs of sewing, baking, painting, performing
37	verb.emotion	verbs of feeling
38	verb.motion	verbs of walking, flying, swimming
39	verb.perception	verbs of seeing, hearing, feeling
40	verb.possession	verbs of buying, selling, owning
41	verb.social	verbs of political and social activities and events
42	verb.stative	verbs of being, having, spatial relations
43	verb.weather	verbs of raining, snowing, thawing, thundering
44	Adj.ppl	participial adjectives

Tabella 3-1 Source files

Essi non vengono distribuiti con il pacchetto WordNet.

La forma del nome dei file sorgente è del tipo:

pos.suffix

dove pos indica la categoria sintattica, noun, verb, adj, adv; suffix viene utilizzato per organizzare i source file in più file specifici per ogni categoria.

Vediamo con un esempio come la stessa forma di parola si possa trovare, a seconda del senso che essa assume, in file sorgente diversi.

Prendiamo in esame la word form *box*, essa ha indice di polisemia (o indice di familiarità) pari a 10 (quindi esistono 10 synset relativi al nome box).

Fra questi:

- *box#1 -- (a (usually rectangular) container; may have a lid; "he rummaged through a box of spare parts")* si trova nel file *<noun.artifact>*
- *box#3, -- (the quantity contained in a box; "he gave her a box of chocolates")* si trova nel file *<noun.quantity>*
- *box#4 -- (a predicament from which a skillful or graceful escape is impossible; "his lying got him into a tight corner")* si trova nel file *<noun.state>*
- *box#5 -- (a rectangular drawing; "the flowchart contained many boxes")* si trova nel file *<noun.shape>*
- *box#6, -- (evergreen shrubs or small trees)* si trova nel file *<noun.plant>*

Il verbo box invece ha indice di polisemia 3 e tutti i synset si trovano nel file sorgente denominato *<verb.contact>*.

Se invece prendiamo in considerazione la word form *home* nella categoria sintattica degli aggettivi abbiamo:

- *home#1 -- (used of your own ground; "a home game")* si trova nel file *<adj.all>*
- *home#2 -- (relating to or being where one lives or where one's roots are; "my home town")* si trova nel file *<adj.pert>*

3.4 - Il formato dei file sorgente

Nei file sorgente ogni synset occupa una riga, e ogni riga termina con il carattere *newline* (a capo). Una linea può essere lunga quanto necessario, ma nessun synset può occupare più di una linea (ad esempio andando a capo nel mezzo della stringa che lo rappresenta).
All'interno di un synset possono essere usati spazi e tabulazioni per la separazione degli elementi.
La sintassi generale è la seguente:

*{ parole puntatori **(** glossa **)** }*

Affinché il synset sia valido deve contenere almeno una parola e la glossa.
Nel caso specifico, i synset relativi alla categoria lessicale dei verbi devono rispettare la sintassi espressa dalla forma:

*{ parole puntatori frames **(** glossa **)** }*

mentre gli aggettivi prevedono l'organizzazione in cluster contenenti uno o più synset principali (*head synset*) e synset satellite (*satellite synset*) opzionali.

I cluster sono gruppi di synset relativi ad aggettivi organizzati attorno a coppie o triplette di antinomi. Ogni cluster contiene uno o più *head synset* che rappresentano il concetto in relazione di antinomia e ognuno di questi ha uno o più *satellite synset* rappresentanti i concetti simili all'*head synset.*

I cluster sono espressi nella forma:

[

head synset

[satellite synsets]

[-]

[additional head/satellite synsets]

]

Ogni cluster viene esplicitato fra parentesi quadre e può avere una o più parti. Le parti sono fra loro separate da uno o più simboli "-".

Gli *head* e *satellite synset* seguono la sintassi generale, tuttavia l'*head synset* deve obbligatoriamente contenere il puntatore **&**, *similar to*, per ognuno dei *satellite synsets*.

Per gli aggettivi relazionali, invece, la forma rimane quella di base.

3.5 - La sintassi di parola

Le parole all'interno di un synset possono essere espresse in due tipi di forme:

- Parola semplice

- Parola / Puntatore

La prima deve avere la seguente sintassi:

word [(marker)] [lex_id]

Se una forma risulta composta da più parole, queste ultime devono essere unite fra di loro dal simbolo **_**, *underscore*; se invece si devono inserire delle cifre numeriche all'interno di una forma ma anche come singola parola, esse devono essere seguite dal simbolo **"**, *double quote*.

Lex_id è un numero compreso fra 1 e 15 ed è utilizzato per distinguere differenti significati della parola. La numerazione di solito, ma non obbligatoriamente, è assegnata in maniera crescente, lo 0 è il *lex_id* di default e non deve essere specificato.

La seconda, per i nomi, ha la sintassi:

[word [(marker)] [lex_id],pointers]

Per i verbi la sintassi parola/puntatore viene esplicitata invece nella seguente forma:

[*word* **,** *[pointers] frames (gloss)* **]**

questo permette di specificare verbo e frames; in tale caso i puntatori sono opzionali.

3.6 - La sintassi dei puntatori

I puntatori all'interno di un synset non sono obbligatori. Se il puntatore è specificato all'esterno del word/pointer set, la relazione è applicata a tutte le forme di parola contenute nel synset e riguarda tutti i significati, incluse tutte le parole specificate usando la sintassi word/pointer.

Se specificato all'interno del word/pointer set, la relazione si riferisce solamente alla parola nell'insieme ed è quindi una relazione lessicale.

Un puntatore è nella forma:

[lex_filename **:** *]word[lex_id]* **,** *pointer_symbol*

o nella forma:

[lex_filename **:** *]word[lex_id]* ^ *word[lex_id]* **,** *pointer_symbol*

Nei puntatori, *word* indica una parola in un altro synset.

Quando si utilizza la seconda forma illustrata, la prima *word* indica una parola in un head synset, la seconda è una parola in un satellite di quel cluster.

L'indice *lex_id* può seguire *word*, e viene usato per collegare il puntatore al synset corretto.

Se il synset contenente la *word* risiede in un altro file sorgente, *word* è preceduta dal nome del file: *lex_filename*.

I puntatori vengono utilizzati per rappresentare le relazioni fra parole in synset diversi. Come abbiamo già visto esistono due tipi di relazioni, semantiche e lessicali. Per alcune categorie sintattiche alcuni tipi di puntatori non vengono usati, di seguito verranno elencati tutti i tipi di puntatore rappresentati in WordNet, alcuni dei quali non essendo fondamentali, non sono stati illustrati nel capitolo precedente.

I puntatori per i nomi sono:

! Antonym

@ Hypernym

@i Instance hypernym

~ Hyponym

i Instance hyponym

#m Member holonym

#s Substance holonym

#p Part holonym

%m Member meronym

%s Substance meronym

%p Part meronym

= Attribute

+ Derivationally related form

;c Domain of synset - TOPIC

-c Member of this domain - TOPIC

;r Domain of synset - REGION

-r Member of this domain - REGION

;u Domain of synset - USAGE

-u Member of this domain – USAGE

I puntatori per i verbi sono:

! Antonym

@ Hypernym

~ Hyponym

***** Entailment

> Cause

^ Also see

$ Verb Group

+ Derivationally related form

;c Domain of synset - TOPIC

;r Domain of synset - REGION

;u Domain of synset - USAGE

I puntatori per gli aggettivi sono:

! Antonym

& Similar to

< Participle of verb

\ Pertainym (pertains to noun)

= Attribute

^ Also see

;c Domain of synset - TOPIC

;r Domain of synset - REGION

;u Domain of synset - USAGE

I puntatori per gli avverbi sono:

! Antonym

\ Derived from adjective

;c Domain of synset - TOPIC

;r Domain of synset - REGION

;u Domain of synset - USAGE

Molti puntatori godono della proprietà riflessiva, quindi nel seguente elenco troviamo a sinistra la relazione e a destra la sua riflessa, e viceversa:

Antonym - Antonym

Hyponym - Hypernym

Hypernym - Hyponym

Instance Hyponym - Instance Hypernym

Instance Hypernym - Instance Hyponym

Holonym - Meronym

Meronym - Holonym

Similar to - Similar to

Attribute - Attribute

Verb Group - Verb Group

Derivationally Related - Derivationally Related

Domain of synset - Member of Domain

3.7 - La glossa

Una glossa può essere inclusa in un qualsiasi synset. La stringa che la rappresenta può essere della lunghezza che si desidera.

Una glossa è semplicemente una stringa chiusa fra parentesi tonde nella quale non deve essere presente alcun carattere di *carriage returns*. Essa fornisce una definizione di cosa il synset rappresenta e/o una frase di esempio.

Per essere più chiari prendiamo ad esempio la word form *wall*, per la categoria sintattica dei nomi. Essa ha indice di familiarità pari a 8.

Per distinguerne i synset relativi a *lex_id* diversi, vengono utilizzate le seguenti glosse all'interno dei synset:

1. *wall#1 -- (an architectural partition with a height and length greater than its thickness; used to divide or enclose an area or to support another structure; "the south wall had a small window"; "the walls were covered with pictures")*
2. *wall#2 -- (an embankment built around a space for defensive purposes; "they stormed the ramparts of the city"; "they blew the trumpet and the walls came tumbling down")*
3. *wall#3 -- (anything that suggests a wall in structure or function or effect; "a wall of water"; "a wall of smoke"; "a wall of prejudice"; "negotiations ran into a brick wall")*

4. *wall#4 -- (a masonry fence (as around an estate or garden); "the wall followed the road"; "he ducked behind the garden wall and waited")*
5. *wall#5, paries#1 -- ((anatomy) a layer (a lining or membrane) that encloses a structure; "stomach walls")*
6. *wall#6 -- (a vertical (or almost vertical) smooth rock face (as of a cave or mountain))*
7. *wall#7 -- (a layer of material that encloses space; "the walls of the cylinder were perforated"; "the container's walls were blue")*
8. *wall#8 -- (a difficult or awkward situation; "his back was to the wall"; "competition was pushing them to the wall")*

3.8 - Grinder utility

La funzione Grinder ha lo scopo, in primo luogo, di prendere i file sorgente e riorganizzarli in forma di database che faciliti il reperimento delle informazioni in WordNet.

Per implementare il database WordNet tutti i file sorgente devono essere analizzati contemporaneamente.

Lo scopo secondario, ma di non minore importanza, è quello di verificare che nei file sorgente venga mantenuta l'integrità sintattica.

La prima parte della funzione implementata dal Grinder ha lo scopo di verificare che la sintassi dei file di input sia conforme alle specifiche e che non vi siano errori strutturali. Gli errori di tipo strutturale si riferiscono al fatto che un oggetto specificato da un puntatore non possa essere trovato. Di solito questo tipo di errori sorgono a causa di distrazioni nella battitura.

La seconda parte della funzione si occupa di “risolvere” tutti i puntatori. Per fare questo, i puntatori specificati in ogni synset vengono esaminati uno ad uno e viene ricercato il valore puntato da ogni puntatore. Il puntatore viene quindi “risolto” memorizzando il “luogo fisico” dove si trova il valore puntato nella struttura dati interna .

Un ultimo passo aggiunge l’indice di polisemia ad ogni forma di parola trovata in un dizionario on-line. Forme di parola uguali in categorie sintattiche diverse hanno indici di polisemia disgiunti.

Il passo finale è quello di creare il WordNet database.

Capitolo 4 - Applicazione di WordNet: il multilinguismo

4.1- Introduzione

Come già introdotto nel Capitolo 1, due delle prime applicazioni di WordNet sono stati dei progetti volti alla creazione di ontologie multilingui secondo i canoni introdotti da WordNet.

Da questo lavoro sono emersi due studi che hanno come fondamento la medesima struttura di WordNet ma affrontano il multilinguismo con approcci completamente diversi: ciò che cambia è il modello di integrazione della struttura madre con le strutture di altri contesti linguistici.

Sempre nel Capitolo 1 abbiamo accennato che i due approcci prendono il nome di Merge Model ed Expand Model. Nel corso della discussione, in questo capitolo, vedremo che sulla scia di questi due modelli sono nate due strutture dati rappresentanti ontologie multilingui: EuroWordNet e MultiWordNet.

4.2 - Il database lessicale EuroWordNet

Obiettivo del progetto, finanziato dalla Comunità Europea, iniziato nel Marzo 1996 e terminato nel 1999, era la costruzione di un database lessicale multilingue coerente ed affidabile, e allo stesso tempo in grado di conservare le diversità e le ricchezze delle diverse lingue.

Il modello scelto fu il Merge Model secondo il quale il metodo da seguire è quello di tenere separate le strutture semantiche dei diversi linguaggi.

Secondo questa logica, il primo passo è quello di sviluppare un database per ogni lingua: questa operazione può essere fatta autonomamente. Il passo successivo prevede lo sviluppo di una parte interlinguistica che metta in relazione i Synset dei diversi wordnet con quelli di WordNet di Princeton che più si avvicinano nel significato.

Le lingue coinvolte in questo progetto sono in tutto otto: olandese, inglese, italiano, tedesco, spagnolo, presenti fin dall'inizio del progetto, e francese, ceco, estone, introdotte in un secondo momento.

I seguenti istituti sono responsabili dei singoli WordNet:

- *Olandese*: the University of Amsterdam (coordinatore di EuroWordNet), NL.
- *Spagnolo*: the 'Fundacíon Universidad Empresa', ES.
- *Italiano*: Istituto di Linguistica Computazionale, C.N.R., Pisa., IT.
- *Inglese*: University of Sheffield, GB.
- *Francese:* Université d'Avignon and Memodata at Avignon, FR.
- *Tedesco*: Universität Tübingen, DE.
- *Ceco*: University of Masaryk at Brno, CZ.

- *Estone*: University of Tartu, EE.

Ogni WordNet specifico del linguaggio considerato è strutturato secondo le stesse linee guida di WordNet, cioè i sinonimi sono raggruppati in synset, i quali a loro volta sono legati fra di loro da relazioni semantiche. In aggiunta ogni significato viene associato ad un synset di Princeton WordNet tramite una relazione di equivalenza, in modo da creare così un database multilingue.

La previsione all'inizio del lavoro era quella di inserire nella base di dati approssimativamente 25000 synset per ogni linguaggio. Il vocabolario doveva contenere tutte le forme di parola di base per ogni lingua, e in più dei sotto-vocabolari per domini specifici in modo da illustrare la possibilità di integrarne la terminologia al lessico generico. La fase di test era prevista su una applicazione di information-retrieval esistente.

Per gli obiettivi del Merge Model, introdotto in precedenza, ogni WordNet specifico viene mantenuto in maniera del tutto indipendente in un database lessicale centrale mentre le relazioni di equivalenza fra significati in lingue diverse vengono mantenute attraverso le relazioni di equivalenza con i synset di WordNet.

Vediamo ora come può essere schematizzata tramite un diagramma di flusso l'implementazione di EuroWordNet.

La costruzione dei wordnet per ogni lingua si articola in due cicli principali indicati in Figura 4-1 con i numeri romani I e II. Ogni ciclo consiste di una fase di costruzione e una fase di confronto. Partendo da risorse già esistenti (dizionari elettronici, altri wordnet,…), viene selezionato un sottoinsieme di significati, per i quali vengo estratti tutti i lemmi ad essi collegati. All'interno del sottoinsieme vengono individuate le relazioni semantiche per ogni linguaggio: si ottiene in questa

maniera una suddivisione in synset, che ora possono essere collegati tramite relazioni di equivalenza, ai synset di WordNet più vicini. Questa operazione viene iterata ogni qualvolta si intenda inserire nuovi wordnet in EuroWordNet, o quando si intenda arricchire wordnet già collegati.

In seguito vengono eseguiti test di coerenza interlinguistica: gli errori che emergono in questo passaggio vengono corretti ristrutturando le gerarchie fra synset e le relazioni che li legano a WordNet.

Si costruisce infine una top-ontology comune in base alle gerarchie appena citate e ai nodi più frequentemente coinvolti nelle relazioni.

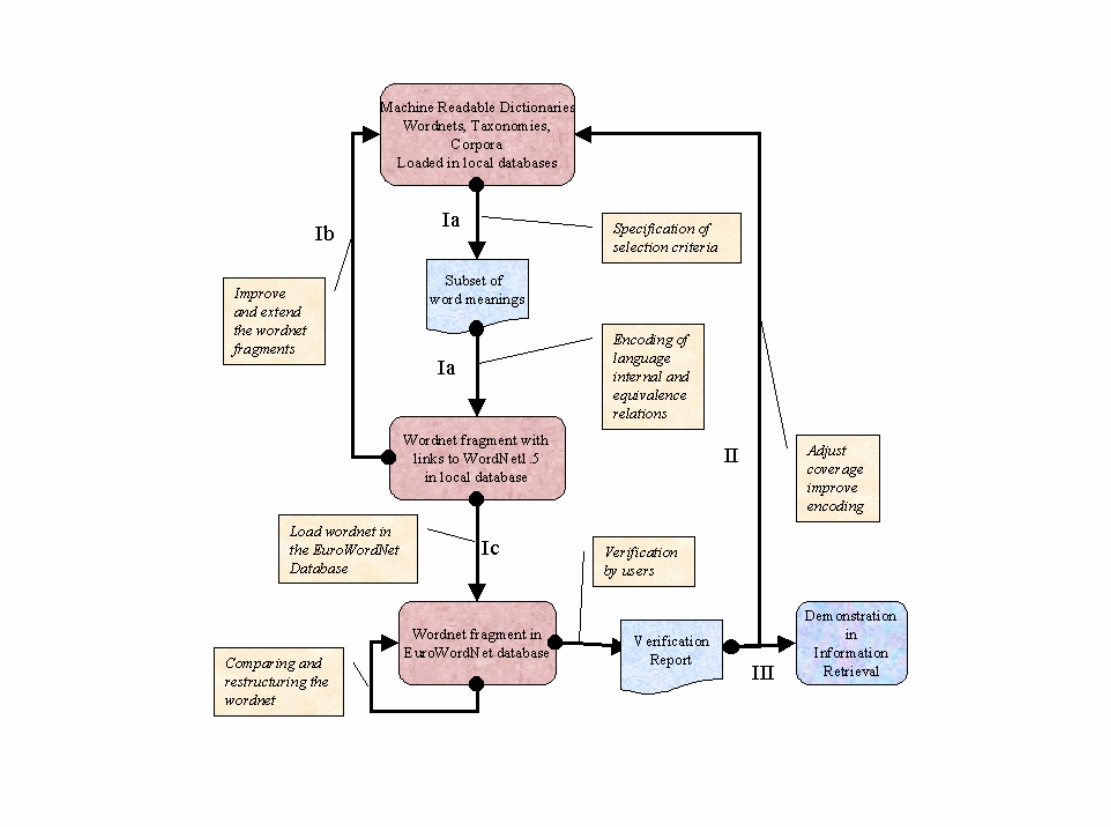


Figura 4-1 Schema per la costruzione di EuroWordNet

La fase III mira a testare EuroWordNet tramite Information-Retrieval.

Si delinea, a questo punto, la presenza di due strutture costitutive fondamentali:

1. un insieme di moduli specifici, uno per ogni linguaggio

2. un modulo indipendente dal linguaggio

Continuiamo l'analisi di EuroWordNet attraverso lo schema dell'architettura utilizzata illustrato in Figura 4-2.

Le relazioni di equivalenza fra synset di lingue diverse vengono esplicitate attraverso una struttura chiamata Inter-Lingual-Index (ILI).

Ogni synset in un wordnet monolingue avrà almeno una relazione di equivalenza con un record della struttura ILI. Secondo lo schema i synset presenti in wordnet di linguaggi diversi collegati allo stesso record ILI dovrebbero essere fra loro equivalenti (in senso lessicale).

La struttura iniziale dell'Inter-Lingual-Index contiene una lista di synset equivalente a quelli esistenti in WordNet; essa è comunque destinata ad incrementare il numero di record con l'introduzione di concetti specifici di altre lingue.

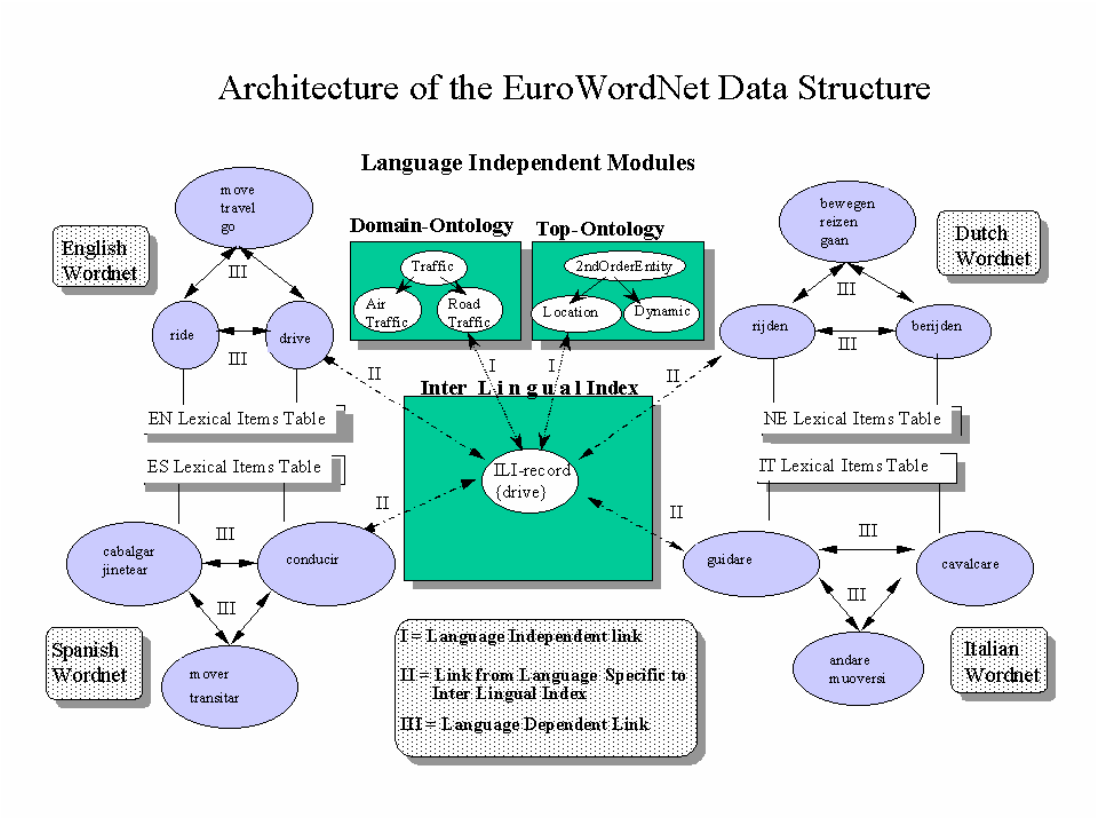


Figura 4-2 Schema di MultiWordNet per la parola "drive"

Sempre in Figura 4-2 si può vedere come synset di linguaggi diversi sono connessi allo stesso record della struttura ILI; nello specifico, è presente un esempio relativo al record *drive*. Inoltre, sempre la stessa figura ci dà una rappresentazione schematica dei differenti moduli e delle relazioni che intercorrono fra di essi. Nel centro si trovano i moduli indipendenti dal linguaggio (in verde), attorno ai quali troviamo quelli di ogni specifico linguaggio. Non esiste alcuna relazione interna fra record della struttura ILI, essa serve solamente come elemento di connessione.

La scelta della lingua inglese per la costruzione della struttura ILI è una diretta conseguenza del fatto che costruire una ontologia neutrale dal punto di vista del linguaggio è stata troppo complessa e costosa in termini di tempo, soprattutto avendo a disposizione una risorsa già disponibile.

I vantaggi dell'architettura che abbiamo appena visto sono i seguenti:

- Le relazioni multilingui non devono essere considerate una per una, ma vengono ridotte a relazioni fra significati.
- Estensioni future del database (come ad esempio l'introduzione di una nuova lingua) non rimettono in discussione tutto il database, avvengono attraverso l'utilizzo dell' ILI, inteso come insieme di concetti a cui far riferimento per l'introduzione di un nuovo wordnet.
- Per aumentare l'efficienza della ricerca interlinguistica, e quindi dell'interconnessione fra wordnet diversi, è sufficiente agire su una unica struttura.

Entrando più nel dettaglio e riferendoci sempre allo schema dell'architettura di EuroWordNet si può notare che esistono due ulteriori strutture indipendenti dal

linguaggio, rappresentanti altrettante diverse ontologie, a cui possono essere collegati i record della struttura ILI:

- La Top-Concept Ontology (TCO o TO) che è una rappresentazione strutturata secondo tre livelli (ordini) dei concetti indipendenti dalle lingue. Ad esempio: Object, Location, Dynamic, Static;
- La Domain Ontology (DO), una gerarchia che divide i significati per campo semantico, cioè "argomento". Ad esempio: Traffic, Road-Traffic, Air-Traffic.

I record di entrambe le strutture possono essere collegati ai significati specifici di ogni linguaggio attraverso le relazioni di equivalenza rese esplicite dai riferimenti presenti nella struttura ILI. Così, come si può vedere nella Figura 4-2, i concetti *Location* e *Dynamic* presenti nel secondo ordine della Top Concept Ontology sono direttamente collegati al record *drive* della ILI. Attraverso le relazioni di equivalenza essi sono indirettamente collegati ai concetti relativi ad altre lingue, nel caso della lingua italiana: *andare*, *cavalcare*, *guidare*, *muovere*.

Lo scopo principale della TCO è di fornire una struttura comune per i concetti più importanti di tutti i wordnets collegati. Essa consiste di 63 gruppi semantici di base, che classificano un insieme di 1310 concetti fondamentali comuni a tutti le lingue, individuati secondo i seguenti criteri:

- Numero delle relazioni ad essi associate
- Posizione nella gerarchia tassonomica
- Frequenza in un corpus

La Domain Ontology gioca un ruolo fondamentale e può essere utilizzata direttamente nell'information-retrieval per raggruppare i concetti in modi differenti. Inoltre può essere usata per separare i vocabolari specifici da quelli generici. Questo risulta fondamentale per controllare i problemi di ambiguità.
La Figura 4-3 mostra in maniera abbastanza fedele la struttura modulare che sta sotto il database EuroWordNet, analogamente a quanto appena visto:

1. ci sono dei moduli linguistici contenenti le relazioni che coinvolgono il lessico proprio della lingua,
2. esiste un modulo indipendente dai linguaggi, che comprende tre strutture: ILI, TCO e DO.

Nel centro troviamo il modulo indipendente dalla lingua, i record appartenenti alla IRI sono collegati ai word-meaning (M) nei moduli dipendenti dai linguaggi, e a uno o più elementi del TC (Top-concept), così come (possibilmente) a un elemento appartenente a D (Domain).
Anche gli elementi, Instances (I), che appartengono solamente ad un determinato dominio linguistico vengono memorizzati come concetti specifici del linguaggio e legati ad un record ILI.
Consideriamo ora un esempio che ci aiuti a focalizzare meglio i legami fra gli elementi delle strutture che compongono l'architettura. Possiamo vedere in Figura 4-4 come la parola Italiana *dito* e la parola Spagnola *dedo* possano essere entrambe riferite a *fingers (dita della mano)* e *toes (dita del piede).* Le parole in Italiano e Spagnolo, che hanno un significato più generale di quello espresso dalle parole Inglesi, possono essere collegata ad un record della ILI in virtù dell'esistenza di una

relazione di iponimia (chiamata eq_hyponym per distinguerla dalle relazioni interne di ogni database) fra i significati.

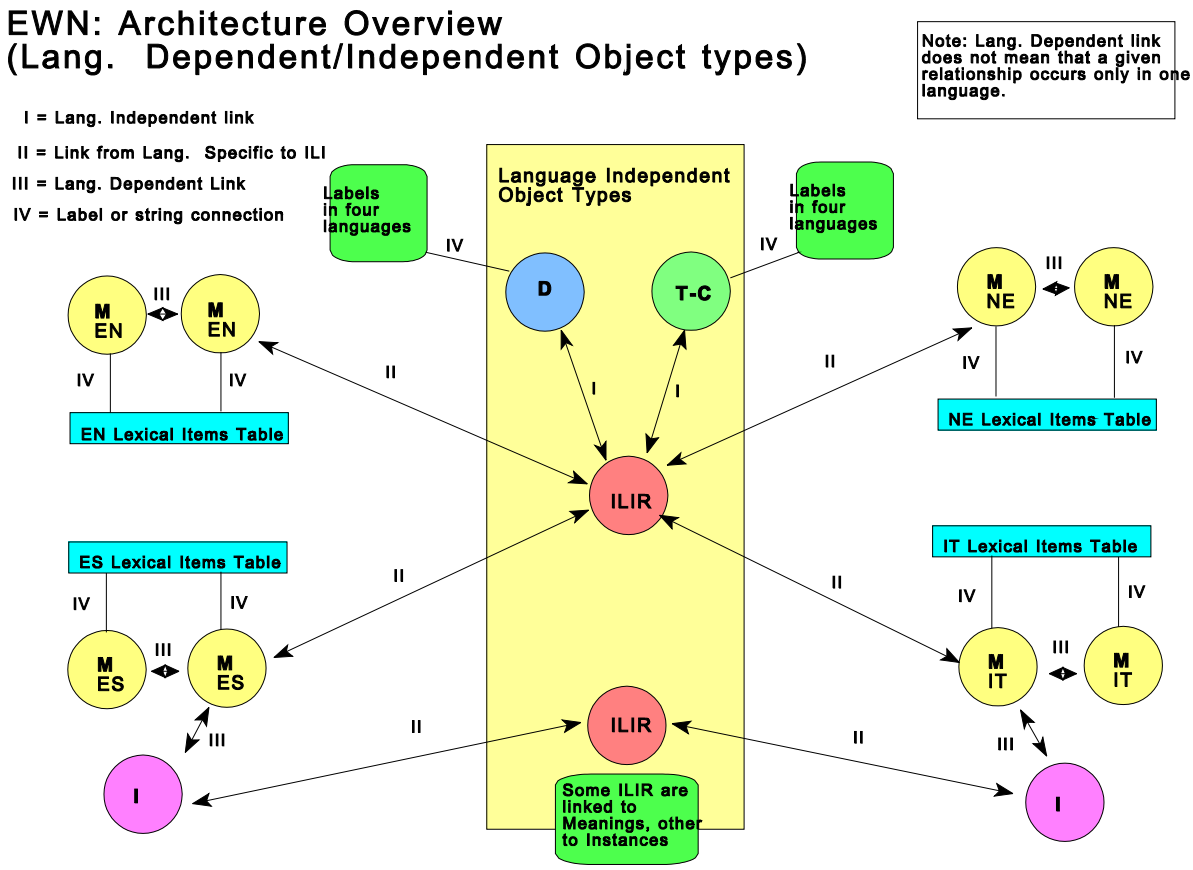


Figura 4-3 Architettura di EuroWordNet

Le parole in lingua Olandese *hoofd* (testa appartenente ad una persona umana) e *kop* (testa appartenente ad un animale) sono d'altra parte più specifiche della inglese *head* (testa) e possono essere collegate ad essa in base alla relazione di iperonimia (eq_hyperonym).

In tutti e tre i casi la relazione diretta (eq_synonym) viene resa esplicita con l'aggiunta di un nuovo record (non-english) alla ILI. Quest'ultimo è necessario

affinché si possa esprimere la relazione di equivalenza fra due significati di lingue diverse dall'inglese quando non esista un lemma inglese equivalente.

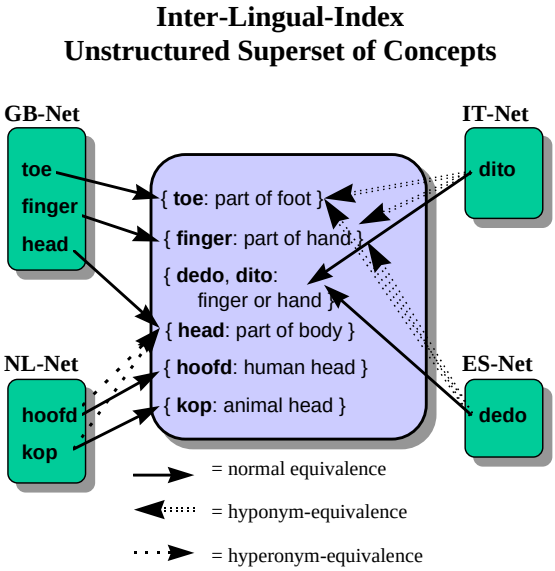


Figura 4-4 ILI Record per la parola "dito"

Per concludere, lo schema scelto per il database multilingue ha i seguenti vantaggi:

- è possibile utilizzare il database per l'information retrieval, collegando parole in una data lingua alle relative equivalenti in altre lingue attraverso i record della Inter-Lingual-Index.
- Wordnets differenti possono essere confrontate e si possono fare ricerche linguistiche incrociate che le rendano più compatibili.
- Differenze dipendenti dal linguaggio specifico non vengono perse.
- È possibile implementare i wordnet in luoghi fisici diversi e relativamente indipendenti.
- Informazioni indipendenti dal linguaggio come ad esempio DO e TCO, sono unici e quindi non è necessario riportarli per ogni linguaggio, eliminando in

tal modo ridondanze e raggiungibili attraverso i collegamenti via elementi del ILI.

- Il database può essere adattato a misura delle esigenze dell'utente modificandone la TCO e il DO senza andare a toccare i wordnet specifici di ogni lingua.

4.3 - Il database lessicale MultiWordNet

Il progetto MultiWordNet, sviluppato presso l'istituti ITC-IRST di Povo (TN), mira a creare un WordNet per la lingua italiana strettamente allineato a quello di Princeton proponendo indirettamente anche un modello a cui ispirarsi per l'interlinguismo. Il sito internet del progetto è http://multiwordnet.itc.it .

Il modello di riferimento che MultiWordNet mira ad implementare è l'expand-model. Nonostante quindi l'obiettivo primario fosse l'implementazione di una ontologia lessicale per l'Italiano simile a quella fatta a Princeton per l'Inglese, l'architettura è stata pensata in modo da renderla flessibile all'inserimento di nuove lingue.

Secondo questo tipo di modello la scrittura della base di dati lessicale deve necessariamente essere fatta ex-novo, per renderla perfettamente aderente all'equivalente inglese non si potranno in questa maniera utilizzare risorse eventualmente già esistenti.

Seguendo la teoria dell'expand-model, si riduce in maniera drastica il numero di discrepanze strutturali dovuto al fatto che si utilizzano strutture simili, ma si deve tenere conto, nel progettare la struttura finale, del fatto che la forzatura della struttura

semantica della lingua inglese sulla lingua italiana (o di altre lingue) potrebbe rappresentare un problema.

Il progetto di MultiWordNet si è basato sull'ipotesi teorica secondo cui le strutture concettuali di livello lessicale di lingue diverse ma appartenenti a culture affini siano confrontabili e in gran parte sovrapponibili. Per giustificare una tale affermazione si assume che i parlanti di culture affini, in questo caso, culture dell'area Europea, condividono gran parte dei concetti lessicali e delle relazioni che esistono tra questi concetti.

Per la piccola percentuale di non sovrapponibilità, nel progetto si è tenuto conto della necessità di rappresentare le differenze semantiche proprie di una lingua rispetto alla lingua inglese, permettendo ai singoli wordnet di "divergere" dalla struttura madre quando necessario.

Altro particolare che si è tenuto in considerazione è l'esistenza dei lexical-gap (letteralmente "lacune lessicali") ovvero concetti che in una lingua non sono esprimibili con una sola parola, ma con un insieme di termini.

Dal punto di vista dell'architettura, MultiWordNet prevede la realizzazione di una *matrice lessicale multilingue* come estensione della matrice lessicale bidimensionale, base di quella implementata in WordNet. Viene aggiunta una terza dimensione, sulla quale è possibile considerare le altre lingue. Nello specifico, la lingua italiana.

La Figura 4-5 visualizza le tre dimensioni della matrice (parole, significati e linguaggi) assieme alle principali relazioni lessicali e semantiche. Per realizzarla occorre ri-mappare le forme lessicali italiane con i significati corrispondenti, M_i, costruendo l'insieme di synset per l'italiano, ovvero determinando gli E_{ij} della lingua italiana. Il risultato è una completa ridefinizione delle relazioni lessicali (ovvero le

relazioni che mettono in corrispondenza lemma e synset), mentre per le relazioni semantiche vengono sfruttate quelle originariamente definite per l'inglese.

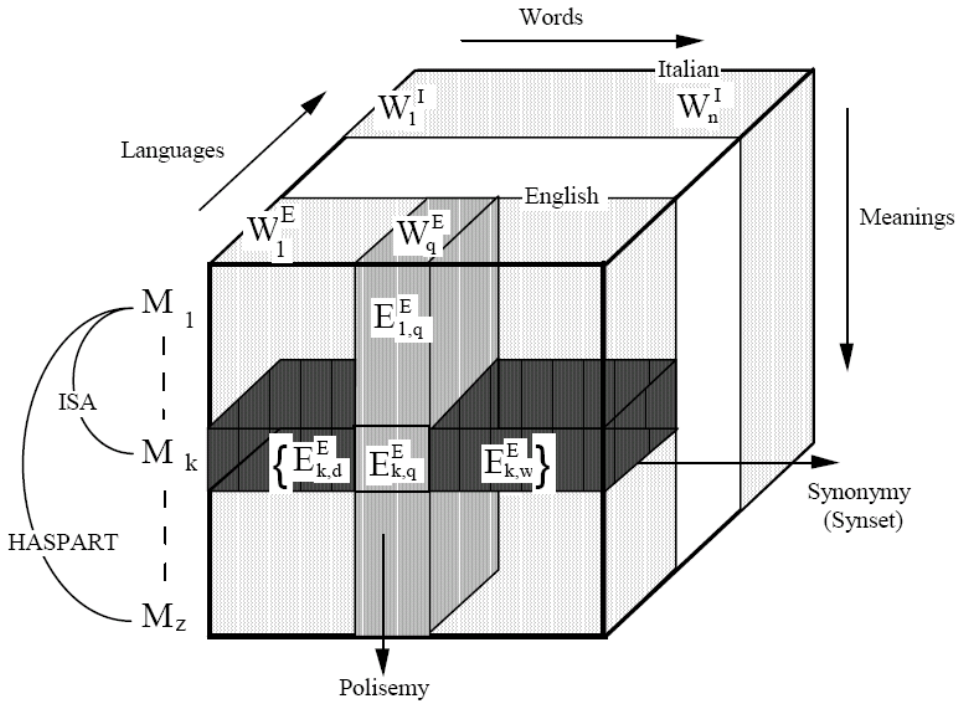


Figura 4-5 La matrice lessicale multilingue di MultiWordNet

Se per un certo M_k si ottiene E_{ik}^{L} =(0,0,….,0) significa che per il linguaggio L non esiste alcuna parola che realizza lessicalmente quel significato.

Nella matrice lessicale multilingue il concetto di synset si evolve, dunque nel multisynset: un significato comune a due o più lingue che viene rappresentato e individuato univocamente nella base di dati. Il multisynset non riguarda più relazioni di sinonimia fra lemmi di una stessa lingua: individua, invece, una relazione di sinonimia in senso lato tra synset equivalenti in lingue diverse.

La struttura che si può vedere in Figura 4-6 è organizzata come un insieme di reti monolingue unite per mezzo di una gerarchia comune (denominata WN-comune). Quest'ultima costituisce la struttura concettuale su cui si basano le altre lingue prese in considerazione. Per quanto riguarda le reti monolingui, la base rimane la stessa

della WordNet inglese, ma ogni singola parola può differenziarsi in base a peculiarità proprie.

A livello implementativo esse contengono l'informazione relativa alle relazioni lessicali, mentre per quanto riguarda le relazioni semantiche si hanno due casi:

1. se la gerarchia combacia perfettamente con quella comune le informazioni vengono ereditate da quest'ultima
2. quando la gerarchia è modificata si rende necessaria la memorizzazione di informazioni specifiche.

Tutti i nodi di una gerarchia monolingue devono essere collegati ai nodi della gerarchia comune in tutti i punti nei quali le gerarchie combaciano.

Per questo motivo è necessaria in quei punti l'introduzione di specifiche relazioni che colleghino i nodi propri di una rete con i synset del WN-Comune.

Osserviamo anche come i DB specifici delle varie lingue risultino costituiti da due parti. La parte costituita da syneset in relazione di equivalenza con quelli della gerarchia comune contiene solo il synset e puntatori ai synset della gerarchia comune equivalenti. Per le parti di gerarchia specifiche per le singole lingue (WN-Differenza), è necessario specificare una serie di informazioni in più.

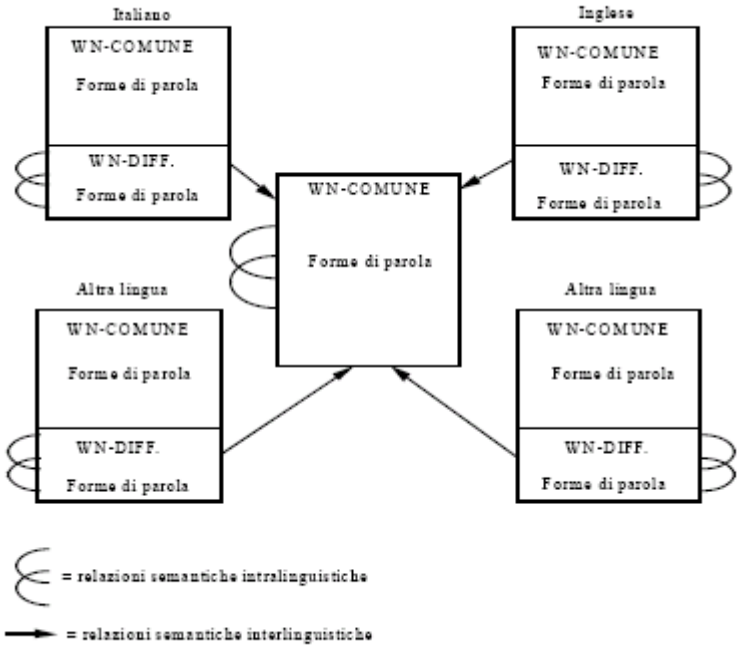


Figura 4-6 Schema di MultiWordNet

La Figura 4-7 mette in risalto come sono stati praticamente raggiunti gli obiettivi appena introdotti. La soluzione consiste di una struttura a moduli di tipo ADD-ON che sovrascrivono (nel senso delle frecce) in vari strati il WordNet senza però modificarlo fisicamente. Il primo di questi moduli semantic ADD-ON è incapsulato nel Common-DB e si pone direttamente sopra WordNet. Inoltre, ogni language-DB contiene un language-ADD-ON che specifica le relazioni semantiche incompatibili con quella determinata lingua.

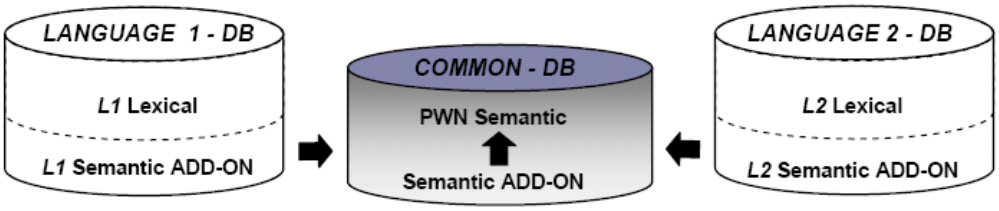


Figura 4-7 Schema del database MultiWordNet

Le informazioni contenute in questi ultimi ADD-ON sono essenzialmente di due tipi:

- relazioni semantiche specifiche di una lingua
- lexical gap.

Come abbiamo già detto prima i *lexical-gap* si hanno quando un determinato concetto, che in una lingua è associato ad un synset non vuoto, in un'altra lingua non trova un synset corrispondente ma può essere tradotto solo con una frase o con un significato più generico; in questo caso si parla più propriamente di *denotation difference*.

Per essere più chiari consideriamo i seguenti due esempi:

- la parola inglese *borrower* può essere espressa in italiano solamente con una espressione del tipo *colui che prende in prestito*
- la parola inglese *bell* può avere in italiano significati espressi da lemmi diversi: *campanello* (small/electric bell), *campana* (church bell), *sonaglio* (on cat's neck); la parola inglese può essere considerata quindi un iponimo interlinguistico dei lemmi italiani elencati, ovvero esprime un concetto più generale.

Il riconoscimento dei lexical-gap è stato fatto implementando una procedura semiautomatica. Ugualmente, per la costruzione del database interlinguistico, le procedure di estrazione dell'informazione sono basate su automatismi.

L'idea è stata di usare un dizionario in formato elettronico, da cui estrarre informazioni semantiche (contenute come testo). Un processo analogo, viene

utilizzato sulle definizioni associate ai synset inglesi, ovvero le short gloss. L'ultima procedura è quella di determinare tra l'insieme di frames ricavati da WordNet quelli che soddisfano meglio una corrispondenza con il singolo frame ricavato dall'italiano. Il matching farà quindi uso di un dizionario bilingue, tramite il quale verificare la corrispondenza tra due termini nelle due lingue. La fase di automazione produrrà un insieme di possibili agganci tra una parola italiana e significati nella rete WordNet per i quali l'algoritmo ha superato una soglia prefissata. La validazione delle scelte proposte deve, a questo punto, passare attraverso l'intervento di un lessicologo.

Capitolo 5 - Domain-Specific lexical database : alcuni progetti italiani

5.1 - Introduzione

Nel precedente capitolo abbiamo parlato di ontologie multilingui sviluppate sul modello di WordNet: EuroWordNet e MultiWordNet. Per la lingua italiana ad esse fanno capo i database ItalWordNet e ItalianWordNet.

Partendo da questi ultimi ne illustreremo le modalità di integrazione con ontologie lessicali implementate per domini specifici.

Vedremo tre esempi che utilizzano come dominio generico EuroWordNet ed uno che utilizza MultiWordNet. Le procedure di integrazione dei database semantici utilizzano lo stesso tipo di approccio, con alcune variazioni in base al database generico. L'approccio utilizzato viene indicato con il nome di plug-in.

5.2 - L'approccio di tipo plug-in

Nell'affrontare il problema dell'integrazione di wordnet generici e specialistici esistono tre considerazioni fondamentali:

- Coverage: nel wordnet integrato tutti i synset a livello più basso del wordnet specializzato devono essere accessibili, così nessuna informazione "specialistica" viene perduta
- Precedence criteria: nel wordnet integrato, il punto di vista "esperto" deve avere la precedenza se l'informazione interessa il dominio specifico; questo assicura la coerenza del wordnet integrato
- Modularità: le due risorse non devono essere modificate dalla procedura di connessione plug-in. Questo garantisce che dopo una consultazione della struttura ottenuta con l'integrazione, entrambe le risorse mantengano le loro informazioni originali e possano essere utilizzate in maniera indipendente.

L'intero apparato è basato sull'uso di *relazioni plug-in* (PLUG-SYNONYMY, PLUG-NEAR-SYNONYMY, PLUG-HYPONYMY) che connettono i synset del wordnet specialistico ai corrispondenti synset generici e sull'utilizzo di *eclipsing procedures*, che nascondono un determinato synset.

Una relazione plug-in collega direttamente coppie di synset corrispondenti, l'una appartenente al wordnet generico e l'altra a quello specialistico. L'effetto primario della definizione di questo tipo di relazioni è la creazione di uno o più synset che vanno a sostituire i synset connessi (ad esempio due synset direttamente coinvolti nella relazione).

D'ora in poi faremo riferimento all'approccio utilizzato per l'integrazione con il database lessicale EuroWordNet mentre per quanto riguarda MultiWordNet, essendo comunque per molti versi simile, ne daremo un analisi un po' più approfondita nel paragrafo relativo ad ArchiWordNet.

5.2.1 - Strumenti Plug-in

Vediamo ora in dettaglio le relazioni plug-in. Per farlo però introduciamo qualche definizione utile.

Nel descrivere le relazioni si farà uso della seguente classificazione (Hirst & St-Onge, 1998): si parla di connessioni verso l'alto (*upward links*) di un synset quando il synset puntato è più generale di quello di origine (esempio le relazioni di iperonimia), di connessione verso il basso (*downward links*) quando l'origine ha un significato più specialistico del synset puntato (esempio l'iponimia) e di connessione orizzontale (*horizonthal links*) per tutti gli altri tipi di relazioni (esempio parte-di, causa, derivazione).

Inoltre indicheremo con GWN il WordNet Generico e con SWN quello specialistico.

La PLUG-SYNONYMY è usata quando, nel momento di fondere i due wordnet, ci si trova davanti synset che si sovrappongono, ovvero che hanno il medesimo significato pur appartenendo a basi di dati diverse. Stabilire questa relazione fra due synset $\{a\}^{GWN}$ e $\{a_1\}^{SWN}$ implica la creazione del nuovo synset $\{a_1\}^{plug}$ che eredita i sinonimi dal SWN così come i downward links (in maniera da dare la priorità alle informazioni contenute nella base di dati specifica) mentre gli upward links vengono ereditati dalla GWN. Gli horizontal links vengono ereditati dalla SWN ed anche dalla GWN nel caso però non vi siano incompatibilità. L'effetto secondario di questo collegamento è che gli iperonimi di $\{a_1\}^{SWN}$ e gli iponimi di $\{a\}^{GWN}$ vengono nascosti.

La PLUG-NEAR-SYNONYMY si usa per collegare due synset che hanno significato molto simile ma non intercambiabile nel contesto. In pratica si utilizza quando per un synset in SWN si individuano più corrispondenze con vari synset in GWN.

Stabilire questo tipo di relazione ha gli stessi effetti visti per la precedente e si possono vedere anche in Tabella 5-1.

	PLUG-SYNONYMY	**PLUG-NEAR-SYNONYMY**	**PLUG-HYPONYMY**	
	$\{a_1\}^{plug}$	**$\{a_1\}^{plug}$**	**$\{a\}^{plug}$**	**$\{a_1\}^{plug}$**
Variants	SWN	SWN	GWN	SWN
Upward links	GWN	GWN	GWN	$\{a\}^{plug}$
Downward Links	SWN	SWN	GWN+$\{a_1\}^{plug}$	SWN
Horizonthal links	GWN +SWN	GWN +SWN	GWN	SWN

Tabella 5-1 Regole di integrazione nelle relazioni plug-in

La PLUG-HYPONYMY si utilizza per connettere un synset di SWN ad un synset di GWN con un significato più generale o quando non ci sia il synset corrispondente nella GWN, quando cioè esiste un gap lessicale.

L'effetto principale dello stabilire questo tipo di relazione è la creazione di due nuovi synset $\{a_1\}^{plug}$ e $\{a\}^{plug}$ con le seguenti caratteristiche:

- $\{a\}^{plug}$ eredita gli upward links da GWN, $\{a_1\}^{plug}$ come iponimi in aggiunta a quelli di $\{a\}^{GWN}$ e horizontal links dalla GWN.
- $\{a_1\}^{plug}$ ha $\{a\}^{plug}$come iperonimo, e downward link e horizontal links ereditati da SWN.

L'effetto secondario è che gli iperonimi di $\{a_1\}^{plug}$ vengono nascosti.

Nei prossimi paragrafi, descrivendo alcuni database specifici, vedremo qualche esempio di relazione plug-in applicata al lessico del dominio implementato.

5.2.2 - Eclipsing

Come abbiamo appena avuto modo di vedere, le relazioni plug-in fra synset diversi che condividono alcune parti del loro significato hanno come effetto secondario la necessità di eliminare, o meglio nascondere (in virtù del principio della modularità), alcune informazioni che creerebbero un conflitto con ciò che si va ad integrare. Questo è reso possibile dall'esistenza di alcune funzioni, citate anche all'inizio del paragrafo, dette *eclipsing procedure* che hanno il compito di eliminare un determinato synset così come la relazione che lo aveva originato.

La loro funzione principale è appunto, nella costruzione di relazioni plug-in, quella di nascondere per esempio gli iponimi di un determinato synset o gli iperonimi in base al tipo di relazione, ma questo non è il loro unico campo di utilizzo.

Esse si rendono necessarie anche per mantenere la coerenza semantica della struttura integrata.

Un tipico esempio è dato dalla forma di parola "balena", essa potrebbe essere presente nella gerarchia della parola "pesce", mentre in un dizionario ontologico scientifico, essa sarà in relazione semantica con la parola "mammifero".

5.2.3 – Integrazione

Il processo di integrazione può essere schematizzato in quattro passi

(1) Identificazione dei synset di base. Viene scelto all'interno dell'insieme dei synset della SWN un sottoinsieme di informazioni che siano rappresentative

del dominio. I synset scelti devono essere fra loro disgiunti e assicurare una copertura completa del dominio specifico.

(2) *Allineamento.* Ogni synset scelto prima viene collegato ai synset della gerarchia della GWN utilizzando le relazioni plug-in viste in precedenza.

(3) *Integrazione.* Per ogni relazione plug-in configurata viene ricostruita la porzione di wordnet integrato, e un algoritmo si occupa di controllare l'esistenza di incongruenze.

(4) *Risoluzione delle incongruenze.* Nel caso si presentino incongruenze sui dati, la risoluzione deve avvenire tramite un esperto che riconsideri quale configurazione abbia la priorità.

In

Figura 5-1 si vede come appaiono i due WordNet prima dell'integrazione e in

Figura 5-2 come appaiono dopo l'integrazione, tramite la sovrapposizione delle zone b e b1.

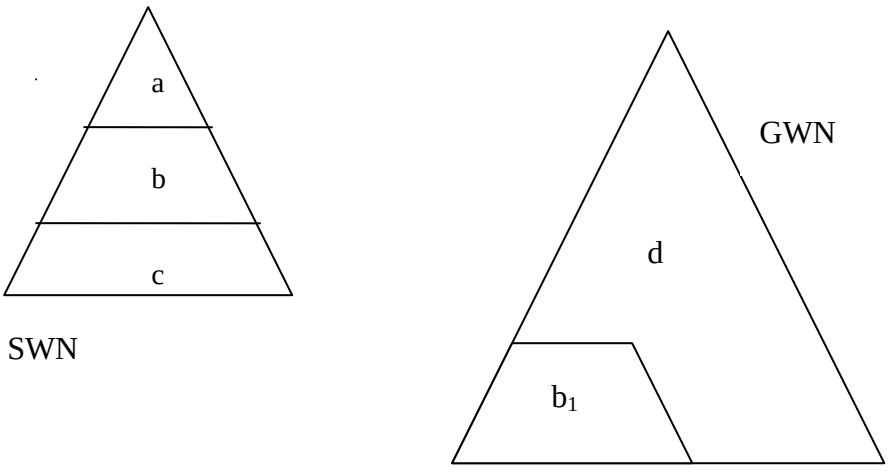


Figura 5-1 SWN e GWN prima dell'integrazione : b e b1 sono le zone di sovrapposizione

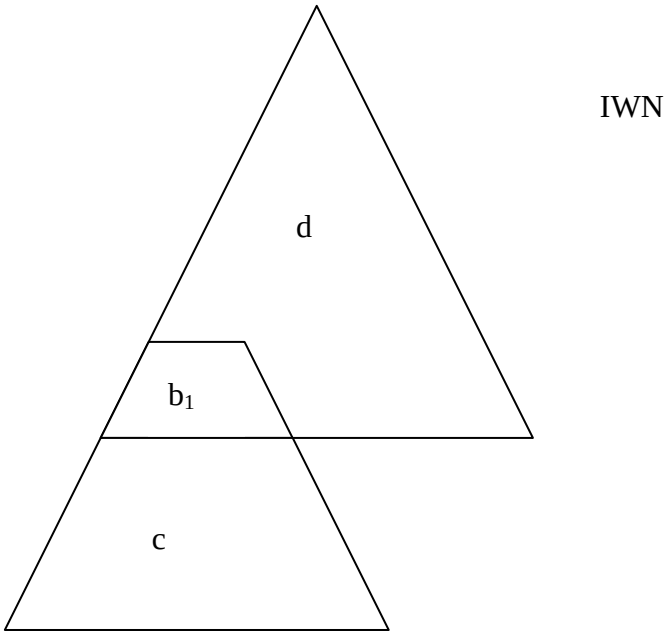


Figura 5-2 WordNet Integrata

5.3 - WordNet generici e specialistici

Quando si parla di ontologia lessicale generica, si fa riferimento a database lessicali che tipicamente contengono un livello di conoscenza senza competenza specifica. Ci sono parecchi esempi di ontologie generiche. Quelli affrontati nella presente tesi, WordNet, EuroWordNet, MultiWorNet, ne sono un esempio come quelli sviluppati sviluppati per la lingua Italiana: ItalWordNet e ItalianWordNet.

Le basi di dati specializzate si concentrano su un dominio specialistico, fornendo gerarchie lessicali per concetti specialistici. L'insieme dei synset in essi descritti tende ad assumere la forma di parole composte ed è fondamentale il ruolo di un esperto per riuscire a stabilire relazioni corrette fra gli stessi.

Sono stati sviluppati parecchi esempi di dizionari lessicali specializzati: i più famosi, a livello di lingua inglese, sono stati sviluppati per il dominio medico (Gangemi et

al., 1999), per l'arte e l'architettura (Art and Architecture Getty Thesaurus) e per la geografia (Getty Thesaurus of Geographical Names).

Per la lingua italiana, invece abbiamo scelto alcuni dei database sviluppati : Economic-WordNet, per il dominio economico finanziario, un database lessicale per termini Marittimi, specializzato per termini appartenenti al dominio specifico della navigazione e del trasporto marittimo, ArchiWordNet, sviluppato per l'architettura e Jur-WordNet per il dominio legale. Oltre a questi, il WordNet Matematico, ontologia lessicale per il dominio della matematica, che descriveremo in dettaglio nei capitolo che seguono.

Nei prossimi paragrafi vedremo queste basi di dati suddivise in base all'ambito specialistico che vanno a illustrare e in base al dominio generico di integrazione.

5.3.1 - Il Dominio Marittimo

Il database semantico costruito per il dominio marittimo è stato implementato presso l'Istituto di Linguistica Computazionale dell'Università di Pisa. Esso è stato costruito secondo i principi dei wordnet generici, utilizzando i costrutti specifici utilizzati in EuroWordNet per relazionare synset di lingue diverse; in questo caso per collegare termini specifici a synset di WordNet 1.5.

Il lavoro di elaborazione della base di dati è iniziato identificando i concetti più rappresentativi del dominio in questione, ovvero i concetti di base (li chiameremo BCs , base concepts). La scelta dei BCs è stata fatta secondo diversi criteri, in particolare la scelta si è diretta verso quei concetti che in entrambi i dizionari, generico e specialistico, mostrassero un gran numero di concetti iponimi, e che

fossero utilizzati con maggior frequenza nel campo del trasporto marittimo e della navigazione.

Si è giunti dopo diverse analisi all'identificazione di un primo nucleo di termini abbastanza generici che costituissero i nodi radice del database che si andava ad implementare. Per essere più chiari possiamo elencarne qualcuno: *nave*, *vela*, *porto*, *ancora* per i nomi, *navigare*, *salpare*, *manovrare*, *stivare*, per i verbi.

Oltre a questi esistono anche altri BCs che però non erano presenti in ItalianWordNet con il senso specifico del dominio marittimo: *armatore*, *nolo*, *classe*, *punto*, *spedizioniere*.

Partendo da questi termini di base, è stato implementato il dizionario codificando i synset, le relazioni di iponimia e le relazioni proprie di WordNet già viste nei capitoli ad esso dedicati.

Lo sviluppo ha incontrato alcune difficoltà dovute alla complessità del dominino in questione: in particolare esso coinvolgeva molti altri campi di conoscenza quali la geografia, la cartografia, l'astronomia e la meteorologia, e inoltre la tecnologia legata al trasporto e la legislatura propria dei contratti marittimi.

Per questo sono stati inclusi anche termini specifici di altri campi e i differenti livelli di specificità dipendono dal dominio di appartenenza. Molta importanza si è data ai termini legati ai fenomeni atmosferici e alla geografia.

Sono state inserite le seguenti categorie lessicali (tra parentesi si indica il numero di elementi appartenenti all'insieme presenti nella base di dati):

- Nomi (1803)
- Verbi (258)
- Aggettivi (43)

- Avverbi (23)
- Nomi Propri (249)

La base di dati è stata collegata ad EuroWordNet tramite le relazioni di equivalenza con i record della struttura Inter-Lingual-Index. Qualora il sinonimo inglese di un termine non fosse presente nella ILI, il termine veniva messo in relazione con il suo iperonimo tramite la relazione di iperonimia (*eq_has_hyperonym*) ed il sinonimo inglese veniva aggiunto ad una lista.

Nei casi in cui il termine inglese fosse ugualmente conosciuto e frequentemente usato al posto di quello italiano, nel synset sono stati inseriti entrambi.

Una caratteristica interessante del dominio marittimo è che la categoria lessicale dei verbi contiene una altissima percentuale di termini appartenenti esclusivamente al dominio specifico. Questi verbi per la maggior parte rappresentano azioni e movimenti e possono essere suddivisi come segue:

- verbi che possono essere riferiti ai tipi di navigazione, come *salpare, fare scalo, approdare*
- verbi che rappresentano azioni che precedono o rendono possibile la navigazione come *armare, varare, carteggiare*
- verbi propri della navigazione a vela come *filare, lascare, orzare*
- verbi riferiti strettamente al trasporto marittimo come *rizzare, sollevare, zavorrare.*

Tutti questi verbi beneficiano della struttura semantica del modello, che permette di codificare le relazioni che fra di essi intercorrono con un buon grado di granularità.

Consideriamo a titolo di esempio il quarto gruppo di verbi: verbi propri della navigazione. I 52 elementi di questo gruppo sono tutti iponimi del verbo *manovrare* e del verbo *condurre*, e denotano una lunga serie di manovre ed azioni funzionali alla navigazione, ciò significa che essi caratterizzano azioni che hanno un concreto e preciso scopo. Per questo motivo nell'uso comune questi verbi si accompagnano ad un sostantivo: *gettare l'ancora, lascare una cima*.

Una ulteriore particolarità dei verbi appartenenti a questo dominio è che molti vengono utilizzati nella loro forma transitiva e intransitiva: *sbarcare* nel senso di *scendere a terra* e *sbarcare* nel senso di *porre a terra*.

Il processo di sviluppo della base di dati è iniziato, come abbiamo già visto, con un procedimento top-down attraverso la definizione dei concetti più generali nel dominio ne sono stati definiti i sottoinsiemi di specializzazione; allo stesso tempo si è seguito anche un processo bottom-up individuando i concetti più specifici e raggruppandoli in base ai concetti più generali.

In Tabella 5-2 abbiamo riportato la gran parte delle relazioni definite internamente al database lessicale e la loro frequenza di utilizzo. Per cercare di capire come esse sono utilizzate vedremo due esempi di verbi: *allascare*, *lascare* e *stivare*.

Allascare*, *lascare

antinomy : ***cazzare***, ***tesare***, ***bordare***

involved_instrument : ***scotta***

sub_event : ***navigare a vela***

causes : ***lasco***

hyperonymy : ***manovrare***

Stivare

Xpos_near_synonym : ***stivaggio***

Hyperonymy : ***porre***, ***situare***, ***mettere***

Involved_agent : ***stivatore***

Involved_patient : ***carico***

Involved_location : ***stiva***

Is_pourpose_of : ***avvolgere***

HAS_HYPERONYM	1048	INVOLVED	67
HAS_HYPONYM	1048	ROLE	67
BELONGS_TO_CLASS	211	ROLE_INSTRUMENT	43
HAS_INSTANCE	211	ROLE_INSTRUMENT	43
FUZZYNYM	156	HAS_SUBEVENT	40
HAS_HOLO_PART	117	IS_SUBEVENT_OF	40
HAS_MERO_PART	117	ROLE_LOCATION	24
HAS_MERO_LOCATION	35	INVOLVED_LOCATION	24
HAS_HOLO_LOCATION	35	PLUG_IN RELATION	228
XPOS_NEAR_SYNONYM	102	ANTONYM	56

Tabella 5-2 Relazioni interne e relazioni plug_in con frequenza di utilizzo nella bancadati

5.3.2 - Il Dominio Giuridico

Il punto di partenza per lo studio di una risorsa linguistica in campo giuridico è stato il progetto *Norme in rete*, lanciato nel 1999, parte del *E-governement plan* italiano.

Norme in rete coinvolge le più importanti istituzioni italiane con lo scopo di "*creare un portale, attraverso il quale una singola interfaccia semplice permetta la ricerca di tutta la documentazione di interesse normativo pubblicata liberamente in rete, in specialmente nei siti istituzionali*".

Jur-WordNet è una risorsa lessicale multistrato. Il dominio legale si adatta molto bene al modello utilizzato per EuroWordNet. La scelta è stata quella di partire da un dominio general-purpose, ItalWordNet, per poi "specializzare" i termini appartenenti al dominio giuridico.

Il metodo di popolamento della base di dati seguito si è basato su un modello di tipo bottom-up utilizzando risorse già esistenti e partendo dai termini più frequenti utilizzati nelle query dei maggiori sistemi di information-retrieval.

Si sono usati:

- per l'identificazione dei lemmi più rilevanti: le stringhe delle query, in particolare si è ricercato nei termini legati da AND, o legati da OR per identificare i sinonimi
- per la definizione dei concetti principali: libri, dizionari, enciclopedie legali e l'L.L.I. che contiene l'archivio storico del Linguaggio Legislativo Italiano.

Altro progetto altrettanto interessante e legato a Jur-WordNet è il LOIS (Lexical Ontologies for Legal Information Sharing, esso si pone all'interno del programma E-content e coinvolge strutture di diverse nazionalità.

Lo scopo è quello di sviluppare una base di dati multilingue costituita da wordnet giuridici in cinque lingue europee (inglese, tedesco, portoghese, ceco, italiano) collegate tra di loro attraverso la lingua inglese.

Esso è caratterizzato dai seguenti componenti:

- Una base di dati lessicali che contiene i concetti descritti dalla dottrina giuridica
- Una base di dati normativi che contiene i concetti definiti nelle direttive comunitarie e nelle leggi di attuazione delle direttive.

Per la presenza di questi ultimi si rende necessaria la presenza di due tipi di relazioni di equivalenza fra lemmi presenti in wordnet di lessici diversi:

- *Equivalenza semantica* (secondo l'accezione di EuroWordNet) è basata sulla descrizione astratta dei concetti e collega i synset della parte lessicale
- *Equivalenza normativa* stabilita da un criterio di appartenenza alle stesse fonti normative e collega i concetti della parte legislativa. Una relazione *implemented_as* lega i concetti presenti nelle direttive con i concetti definiti nella leggi di implementazione ed una *relazione di equivalenza* o *quasi equivalenza* tra i due concetti.

5.3.3 - Il Dominio Economico

Prendiamo ora in esame il progetto di integrazione di un WordNet contenente termini specifici del dominio economico e ItalWordNet, che abbiamo già introdotto nelle

precedenti sezioni. EconomicWordNet contiene circa 5000 lemmi distribuiti in 4700 synset come si può vedere in Tabella 5-3.

Quello che è interessante sottolineare di questo progetto è soprattutto il fatto che dallo stesso ha preso vita lo studio sul metodo plug-in per l'integrazione di WordNet generici e specialistici.

	EconomicWordNet
Synsets	4.687
Senses	5.313
Lemmas	5.130
Internal relation	9.372
Variants/synsets	1.13
Senses/lemma	1.03

Tabella 5-3 I numeri di EconomicWordNet

La procedura di integrazione è stata testata per questo database. Per prima cosa sono stati individuati 250 synset di base. Si è poi passati alla seconda fase con la scelta delle relazioni plug-in. In caso di gap si è provveduto a relazionare gli elementi con una relazione di tipo PLUG-HYPONYMY.

Il numero di relazioni plug-in instaurate è 269 divise come segue:

- PLUG-SYNONYMY : 92
- PLUG-NEAR-SYNONYMY : 36
- PLUG-HYPONYMY : 141

A titolo di esempio consideriamo la creazione di PLUG-HYPONYMY fra {attività}^{GWN} e {attività_di_intermediazione_finanziaria}^{SWN} che porta alla produzione di un nuovo synset {attività}^{PLUG} che avrà gli stessi iperonimi di {attività}^{GWN} , come iponimi gli stessi di {attività}^{GWN} e quelli di {attività_di_intermediazione_finanziaria}^{SWN}. Inoltre il nuovo synset {attività_di_intermediazione_finanziaria}^{PLUG} avrà {attività}^{PLUG} come iperonimo e come iponimi gli stessi di {attività_di_intermediazione_finanziaria}^{SWN}. Come ultima cosa verranno nascosti gli iperonimi di {attività_di_intermediazione_finanziaria}^{SWN}.

5.3.4 - Il dominio della Architettura

ArchiWordNet (ArchiWN) è un progetto che coinvolge l'ITC-irst di Trento e il Politecnico di Torino, ed ha come scopo quello di implementare una risorsa lessicale da utilizzare all'interno del SIS (Still Image Server), un archivio di architettura disponibile al Politecnico.

La risorsa lessicale è funzionale alla ricerca di immagini sul database del SIS, catalogate e caratterizzate da keywords.

Come si legge nella documentazione del progetto, esso implementa un dizionario bilingue Italiano/Inglese, utilizzando MultiWordNet come risorsa generica.

Possiamo suddividere il lavoro fatto in due parti: una prima parte che si è occupata di costruire la gerarchia lessicale estrapolando le informazioni da alcune risorse già esistenti come ad esempio l'AAT (Art and Architecture Treasures), che hanno un sistema gerarchico differente da quello adottato da WordNet ed una seconda che si occupa di integrare la risorsa lessicale appena costruita con MultiWordNet.

In Figura 5-3 è schematizzato il risultato della riorganizzazione della parola *metal* dal modelllo AAT al modello WordNet, dove con ISA si intende una relazione di iponimia/iperonimia.

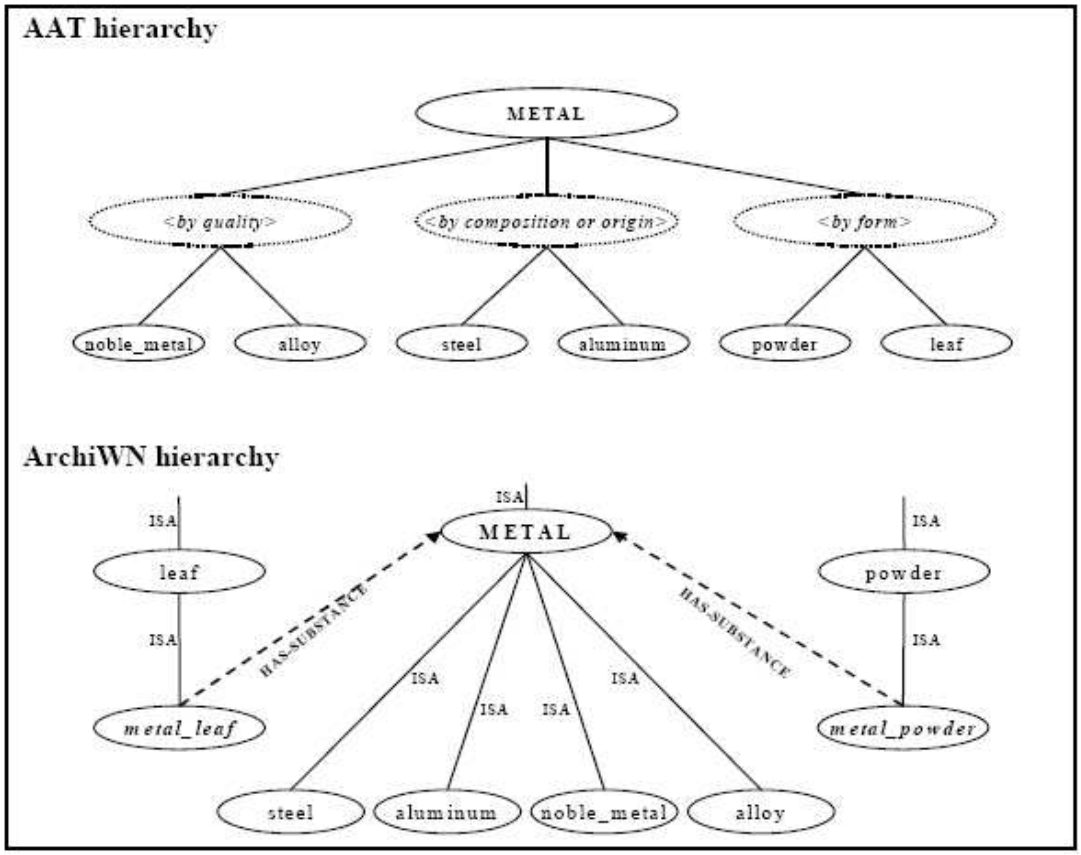


Figura 5-3 Riorganizzazione della gerarchia AAT secondo il modello WordNet

Per esigenze del dominio in questione sono state introdotte tre nuove relazioni:

- HAS FORM (n/n) {tympanum} HAS FORM {triangle, trigon, trilateral}
- HAS ROLE (n/n) {metal section} HAS ROLE {upright, vertical}
- HAS FUNCTION1 (n/v) {beam} HAS FUNCTION {to hold, to support, …}

La procedura di integrazione di ArchiWordNet con MultiWordNet segue, in maniera analoga a quanto visto per EuroWordNet, un approccio di tipo plug-in.

Sono state individuate, nel dominio, 13 aree semantiche che corrispondono ai nodi radice delle gerarchie lessicali di ArchiWordNet. Le aree semantiche individuate sono elencate nella prima colonna della Tabella 5-4.

ArchiWN hierarchies	**MultiWN Plug-in nodes** (lemma/sense number)	**Type of plug-in**
Architectural styles	architectural_style/1	Substitutive
Materials	material/1, substance/1	Substitutive
Construction products	building_material/1	Substitutive
Techniques	technique/1	Integrative
Tools	tool/1	Integrative
Components of buildings	structure/1, component/3, region/1	Hyponymic
Single buildings and building complexes Hyponymic	structure/ArchiWN building/1, building_complex/1	Hyponymic inverse
Physical properties	physical_property/1	Integrative
Conditions integrative	condition/1	Integrative
Disciplines	discipline/1	Integrative
People	person/1	Integrative
Documents	document/1	Integrative
Drawings and representations	drawing/2, representation/2	Integrative

Tabella 5-4 Integrazione delle gerarchie lessicali di ArchiWN con MultiWN

Sono stati formulati quattro tipi di procedure, in grado di rispondere alle esigenze dell'integrazione:

- *Substitutive plug-in.* Viene utilizzata quando la ArchiWN fornisce una gerarchia specialistica ben definita. Si procede così alla sostituzione del nodo della base di dati generica con quello della specialistica. Questa funzione, necessita di chiamare anche la procedura di eclipsing per nascondere i downward link del nodo sostituito.

- *Integrative plug-in*. Le due gerarchie vengono fuse assieme, il nodo della ArchiWN in questione sostituisce quello della MultiWN e gli iponimi coerenti al dominio della WordNet specialistica vengono inclusi attraverso una operazione di etichettatura.
- *Hyponymic plug-in*. Una gerarchia di ArchiWN si collega come iponimo ad un synset di MultiWN.
- *Inverse plug-in*. Un downward link di un nodo appartenente a MultiWN viene connesso ad ArchiWN come iponimo di un synset di ArchiWordNet. Questa procedura si utilizza principalmente quando i synset contenuti in MultiWordNet sono considerati rilevanti ai fini del dominino specialistico, ma si non si trovano nella posizione giusta.

Nella Tabella 5-4 Integrazione delle gerarchie lessicali di ArchiWN con MultiWN, per ogni gerarchia viene indicato il tipo di procedura plug-in utilizzata.

Capitolo 6 - Una Base di Dati di Termini Matematici

6.1 - Introduzione

Nell'ambito dell'attività di ricerca del gruppo Elaborazione del Linguaggio Naturale presso il Dipartimento di Matematica dell'Università di Padova nasce nell'anno accademico 2002-2003, grazie al lavoro di tesi dell'Ing. Croin, una base di dati il cui scopo è quello di costituire una collezione di termini matematici.

La volontà di creare una risorsa lessicale che caratterizzi il linguaggio matematico e non dia solamente delle semplici definizioni, portò a scegliere i principi che regolano WordNet come base per la costruzione del dizionario.

In particolare si scelse di seguire i principi di EuroWordNet, in modo da poter interfacciare questo con altri database, in particolare con ItalWordNet. La scelta è strategica nel senso che la filosofia abbracciata da EuroWordNet è quella di costruire database lessicali secondo le linee guida di WordNet ma in maniera autonoma.

La struttura è stata nel tempo modificata. Nell'ambito di questo lavoro di tesi si è cercato, in una prima parte, di documentare il programma così come era stato variato, completandolo di alcune parti e introducendo alcune ristrutturazioni della base di dati in modo da renderla più facilmente arricchibile di nuovi termini matematici.

La seconda parte del lavoro è basato invece sull'inserimento di una biografia per i nomi di matematici presenti nella banca dati e l'inserimento di figure e formule nella glossa dei termini matematici, laddove siano necessari per facilitarne la comprensione.

Nella terza ed ultima parte si è reso disponibile lo strumento on-line e si è eseguita la fase di test dello strumento, in maniera particolare si sono corrette incongruenze della banca dati e si è testata la parte che riguarda i link alle biografie dei matematici e alle formule e figure.

Nel presente capitolo affronteremo la prima parte, ovvero l'analisi delle procedure e delle strutture utilizzate dalla bancadati.

6.2 – I Source files

Il source file (nomi.txt) su cui si basa il database matematico è un file di testo contenente una serie di stringhe le quali sono essenzialmente composte di due campi:

Identificativo_lemma

Identificativo _argomento

Puntatori

Identificativo_glossa

L'identificativo è la stringa che rappresenta il lemma. I puntatori ai synset, argomento fondamentale nella costruzione della struttura, rispettano le direttive standard imposte dal WordNet con alcune accezioni. Nel WordNet matematico si è preferito indicare la relazione di sinonimia facendo seguire il lemma in questione dalla lettera S, e si è introdotta una nuova relazione R con il significato di "è in

relazione a". In realtà i casi di sinonimia sono molto pochi nell'ambito matematico. Inoltre non è stata implementata la ricerca dei plurali.

Il puntatore all'argomento identifica univocamente il concetto a cui il lemma fa riferimento. La chiave e la sua descrizione seguono uno schema autonomo basato sul principio di classificazione utilizzato per esempio in ambiente bibliotecario, ma non solo.

Uno schema di classificazione è una struttura informativa utilizzabile per la classificazione di documenti o di contenitori di conoscenze e esplicitato tramite una organizzazione gerarchica dei contenuti di una disciplina, di un settore di attività o dell'intero complesso dei saperi.

Così se consideriamo uno schema in più livelli, al primo livello si trovano classificate le categorie maggiori, per ogni categoria al livello successivo si trovano le categorizzazioni intermedie e così via secondo una struttura ad albero.

La scelta è stata quella di una classificazione per il linguaggio matematico a due livelli secondo quanto illustrato in Tabella 6-1.

1	Matematica generale e Biografie	11	Logica
		12	Teoria degli insiemi
2	Algebra (in generale)	21	Algebra (anelli, gruppi, strutture)
		22	Algebra (booleana, lineare)
		23	Algebra (equazioni e disequazioni algebriche, logaritmi, esponenziali, algebra elementare)
3	Aritmetica	31	Misure

		32	Teoria dei numeri (naturali, interi, razionali, irrazionali, trascendenti, reali, complessi, quaternioni)
4	Topologia		
5	Analisi	51	Analisi reale
		52	Analisi Complessa
		53	Analisi Funzionale
		54	Equazioni Differenziali
		55	Teoria delle funzioni
		56	Trasformazioni (Fourier, Laplace, ...)
		57	Teoria della misura
6	Geometria (in generale)	61	Geometria piana e solida, trigonometria, vettori, matrici
		62	Geometria algebrica
		63	Geometria differenziale
9	Matematica applicata (in generale)	91	Probabilita'
		92	Analisi Numerica
		93	Identificazione, stima
		94	Matematica applicata alla biologia, fisica,...
		95	Matematica discreta (alberi, teoria dei grafi,...)
		96	Ricerca operativa
		97	Statistica

		98	Teoria dei giochi
		99	Teoria dei sistemi

Tabella 6-1 Rappresentazione decimale degli argomenti

L'ultimo puntatore che si trova nella struttura è il puntatore alla glossa che descrive il lemma cercato. Inizialmente la glossa si trovava esplicitata fra parentesi nella riga del Synset. Per gli scopi didattici del dizionario matematico, e affinché esso sia di reale utilità non solo agli studenti, ma anche a neofiti della matematica, la glossa ricopre un ruolo molto importante.

Per questo motivo si è pensato di introdurre un file che contenesse appunto solamente le glosse, in modo da poterlo arricchire senza appesantire e rendere quasi illeggibile il file nomi.txt.

Scendiamo ora nel dettaglio e vediamo uno per uno i file che abbiamo introdotto.

6.2.1 - I Synset: nomi.txt

Come già introdotto, nella costruzione dei Synset vengono seguite le indicazioni date dal manuale d'uso di WordNet. In particolare la struttura riguardante i puntatori segue quella utilizzata per la rappresentazione dei synset relativi alla categoria dei nomi.

Ogni riga di nomi.txt è quindi strutturata come segue:

{lemma [identificativo_argomento] [f1,puntatore],[f2,puntatore],…, [fi,puntatore] ([identificativo_glossa])}

dove le parentesi quadre indicano che i campi fra esse racchiuse sono opzionali.

Vediamo il valore che possono assumere le variabili presenti nella riga:

- *lemma* è la forma di parola a cui il synset rappresentato dalla riga si riferisce e ne è la chiave primaria
- *Identificativo_argomento* è dato dalla chiave dell'argomento al quale il lemma si riferisce; esso può assumere i valori indicati nelle colonne 1 e 3 della Tabella 6-1
- *le coppie del tipo fi, puntatore* sono le forme di parola (*fi)* e il simbolo (*puntatore*) che le mette in relazione con lemma, puntatore è uno dei simboli indicati in capitolo 2
- *identificativo_glossa* è il lemma stesso; il file glossa.txt contiene una riga relativa al lemma e ne rappresenta la chiave di ricerca in quest'ultimo file.

Come si può notare una riga in forma minima deve contenere almeno il lemma e le due parentesi *( ).*

6.2.2 - Gli Argomenti: Concetti.txt

Concetti.txt è il file in cui sono contenuti gli argomenti secondo la convezione di classificazione scelta e precedentemente introdotta e la classificazione Dewey rappresentata in Tabella 6-1.

Ogni riga del file concetti.txt contiene i seguenti campi:

identificativo_argomento argomento

Scendendo nel dettaglio:

- *identificativo_argomento* è un codice decimale e assume i valori rappresentati nella prima e terza colonna della Tabella 6-1, esso è anche la chiave primaria
- *argomento* è una stringa che può assumere uno dei valori elencati nella seconda e quarta colonna della Tabella 6-1.

6.2.3 - La glossa: glossa.txt

glossa.txt è l'ultimo file di informazioni del dizionario. Ogni riga assume la forma:

identificativo_glossa glossa

dove:

- *identificativo_glossa* è il lemma stesso ed è la chiave primaria
- *glossa* è una stringa contenente sia semplice testo che parti scritte in linguaggio HTML

Entreremo più avanti nel dettaglio di ciò che contiene il campo glossa, quando parleremo delle biografie dei matematici e della possibilità di introduzione di immagini.

6.3 - La struttura Dati

6.3.1 - La classe FormaCorr

In WordNet la peculiarità è quella di mettere in relazione diverse forme di parola fra loro correlate da specifiche relazioni semantiche. Come abbiamo visto, la struttura dei SourceFile è fatta in modo che ogni forma di parola correlata si trovi nella riga stessa e separata da una virgola e da un simbolo che ne identifica la relazione.
FormaCorr è una struttura che potremmo chiamare “Puntatore”. La sua struttura può essere schematizzata come in figura.

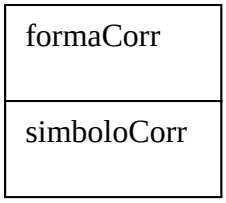


Figura 6-1 FormaCorr

FormaCorr è una classe che contiene la forma correlata e il simbolo che la rappresenta. In essa è incluso anche un costruttore che assegna alle due variabili appena elencate i valori passati come argomento del costruttore stesso. Le altre due funzioni getFormaCorr e getSimboloCorr restituiscono l’una il valore della stringa formaCorr e l’altra il valore del simbolo simboloCorr.
La definizione delle funzioni della classe è contenuta in FormaCorr.Cpp.

```
// FormaCorr.h
```

```
#include <string>
using namespace std;
class FormaCorr {
private:
      string formaCorr;
      string simboloCorr;
public:
      FormaCorr(string forma, string simbolo);
      string getFormaCorr();
      string getSimboloCorr();
}; // FormaCorr
```

6.3.2 - I simboli

I simboli utilizzati nella classe FormaCorr vengono definiti nel file Simboli.h.

Le righe dello stesso non sono altro che delle direttive *define* nelle quali vengono elencati i possibili simboli che esprimono le relazioni con la forma relativa alla riga di nomi.txt analizzata:

```
// Simboli.h

#define SIMB_ANTONIMO "!"
#define SIMB_IPERONIMO "@"
#define SIMB_IPONIMO "~"
#define SIMB_MERONIMO "%p"
#define SIMB_OLONIMO "#p"
#define SIMB_RELAZIONE "R"
#define SIMB_SINONIMO "S"
```

Come si può notare, a differenza di WordNet, qui è stato introdotto sia un simbolo per indicare la sinonimia “S” (in WordNet il sinonimo non ha un simbolo) che un ulteriore simbolo per la Relazione “R”.

6.3.3 - *La Classe Synset*

Synset è una classe le cui variabili private sono una stringa forma_di_parola, una stringa idArg che identifica l'argomento, una lista di puntatori FormaCorr e una stringa denominata glossa. Esistono varie funzioni sia public che private; ne illustreremo le principali.

Le loro definizioni sono contenute nel file Synset.cpp.

```
// Synset.h

#include<list>
#include "FormaCorr.h"

class Synset {
private:
    string forma_di_parola;
    string idArg;
    list<FormaCorr> p;
    string glossa;
public:
    Synset();
    Synset(string, int);
    string getForma();
    string getIdArgom();
    list<FormaCorr> getFormaCorr();
    string getGlossa();
private:
    void insert_forma(string);
    void insertIdArgom(string);
    void insert_FormaCorr(string, string);
    void insert_glossa(string);
    void visualizza_Synset();
    string estraiFormaGlossa(int, string);
    string estraiFormaCorrSimbolo(string, string::size_type);
    static void erroreCar(int, char);
    static string estrai(string, char, int, string::size_type* i);
```

```
    static void errore(int, string);
}; // Synset
```

I due costruttori , il primo privo di argomento e il secondo con argomenti una stringa e un intero, utilizzano delle funzioni private della classe e quindi per il momento le tralasceremo e inizieremo da queste altre funzioni.

Iniziamo dal seguente gruppo:

```
void insert_forma(string);
void insertIdArgom(string);
void insert_FormaCorr(string, string);
void insert_glossa(string);
```

Esse si occupano di inserire nel Synset rispettivamente: la forma di parola, l'identificatore dell'argomento, il puntatore FormaCorr e il puntatore alla glossa. FormaCorr viene inserita nella lista tramite una funzione di push.

In maniera del tutto analoga, le funzioni:

```
string getForma();
string getIdArgom();
list<FormaCorr> getFormaCorr();
string getGlossa();
```

ritornano i valori delle variabili interessate contenute nel Synset. La funzione getFormaCorr restituisce la lista dei puntatori.

Vediamo ora le due funzioni fondamentali della classe (quelle appunto usate dal costruttore):

```
string estraiFormaGlossa(int, string);
string estraiFormaCorrSimbolo(string, string::size_type);
```

La prima delle due funzioni analizza la riga in ingresso e inizia a memorizzare nelle variabili della classe Synset la forma, il puntatore all'argomento e quello alla glossa. Mentre in uscita restituisce la stringa che contiene la parte di riga contenente i simboli e le forme correlate.

La seconda funzione, invece, con in ingresso appunto la stringa contenente i simboli e le forme correlate provvede a riconoscere all'interno della stringa stessa il simbolo e la forma correlata, e inserisce ciò che ha estratto nella lista *p* di tipi *FormaCorr* definita nella parte *private* della classe che stiamo esaminando.

Come abbiamo già detto le due funzioni appena esaminate sono fondamentali in quanto utilizzate nel costruttore della classeSynset.

```
Synset::Synset(string riga, int nLinea) {
    string::size_type iSpaz;
    riga = estraiFormaGlossa(nLinea+1, riga);
    while ((iSpaz = riga.find_first_of(' ')) != string::npos)
        riga = estraiFormaCorrSimbolo(riga, iSpaz);
    riga = estraiFormaCorrSimbolo(riga, iSpaz);
} // Synset
```

La creazione della classe avviene attraverso l'inizializzazione delle sue variabili private. Si può notare che nel *ciclo while* viene iterata la funzione `estraiFormaCorrSimbolo(riga, iSpaz)` per ogni coppia (forma, simbolo) trovata nella stringa del file nomi.txt.

6.3.4 - La Classe ListaSynset

Nella classe Listasynset viene implementata la struttura a lista atta a contenere i synset.

Si utilizza la struttura di tipo *list* per definire una variabile privata lista_Synset. La struttura *list* fa parte di un gruppo di oggetti che in C++ viene chiamato STL (Standard Template Unit). Essi implementano dei contenitori che, in maniera indipendente dai tipi delle variabili che andranno a contenere, mettono a disposizione una serie di funzioni di manipolazione, inserimento ed estrazione. Fondamentale nelle STL è l'uso degli iteratori, generalizzazione del concetto di puntatore ad un elemento della sequenza.

ListaSynset è una classe che contiene una variabile privata lista_Synset, implementata da una *list* i cui elementi sono di tipo Synset.

```
class ListaSynset {
private:
      list<Synset> lista_Synset;
public:
      void put(Synset s);
      Synset get(string chiave);
}; // ListaSynset
```

6.3.5 - *La Classe Argomento*

La classe Argomento consiste di due variabili di tipo private che sono rispettivamente l'identificativo e la descrizione dell'argomento.

```
// Argomento.h

#include <list>
#include <string>

using namespace std;

class Argomento {
private:
    string idArg;
    string descrizione;
public:
    Argomento();
    Argomento(string, int);
    string getId();
    string getDescrizione();
private:
    void setId(string);
    void setDescrizione(string);
    static void erroreCar(int, char);
    static string estrai(string, char, int, string::size_type* i);
    static void errore(int, string);
}; // Argomento
```

6.3.6 - *La Classe ListaArgomento*

La classe contiene una variabile di tipo private che è una list di elementi di tipo Argomento.

```
// ListaArgomento.h

#include "Argomento.h"
```

```
// Lista di Argomento.
class ListaArgomento {
private:
      list<Argomento> listArgomento;
public:
      void put(Argomento s);
    string getDescrizione(string);
}; // ListaArgomento
```

6.3.7 - La classe Definizione

La classe che andiamo a descrivere è stata introdotta nell'ambito di questo lavoro di tesi per permettere la separazione del file dei nomi da quello delle glosse. Per evitare incongruenze la struttura è molto simile a quella della classe Argomento. La variabile FormaID è la parola di cui si dà la definizione, quest'ultima invece è memorizzata in Glossa.

```
#include <list>
#include <string>

using namespace std;

// Definizione.
// Contiene l'id. e la descrizione della definizione.
class Definizione {
private:
    string FormaID;
    string Glossa;
public:
    Definizione();
    Definizione(string, int);
```

```
    string getFormaID();
    string getGlossa();
private:
    void setFormaID(string);
    void setGlossa(string);
    static void erroreCar(int, char);
    static string estrai(string, char, int, string::size_type* i);
    static void errore(int, string);
}; // Definizione
```

6.3.8 - La classe ListaDefinizione

La classe ListaDefinizione, anch'essa inserita per la suddivisione della glossa, si occupa di creare una struttura di tipo list che contenga le glosse in maniera analoga a quanto visto per ListaArgomento e ListaSynset.

```
#include "Definizione.h"

// Lista di Definizione.
class ListaDefinizione {
private:
      list<Definizione> listDefinizione;
public:
      void put(Definizione s);
        string getDefinizione(string);
}; // ListaDefinizione
```

6.3.9 - *La classe Db*

Db è il vero e proprio DataBase su cui si basa il Dizionario Matematico.

```
class Db {
private:
      ListaSynset tabella[DIM];
      ListaArgomento tabArgom[DIM_ARGOM];
public:
    Db(char* nomeFileNomi, char* nomeFileArgom); //costruttore con
argomenti file Nomi e Argomento
    Synset getSynset(string);
    string estraiArgomento(Synset&);
    void outFormeCorr(Synset, int, string);
private:
    void insert_Synset(Synset);
    void insertArgomento(Argomento);
    static int fhash(string);
};
```

Le tabelle hash

Una tabella hash è una vettore di N elementi contenitore chiamati “bucket”.

Il vantaggio di utilizzare una tabella hash per contenere gli elementi del dizionario è sicuramente dato dal costo lineare delle operazioni fatte su questi ultimi indipendentemente dal numero di elementi memorizzati.

L’inserimento di un elemento x, del quale conosciamo la natura, avviene tramite una funzione hash h(x) che restituisce un valore intero compreso fra 0 e N-1. Tale valore identifica l’indice del bucket nel quale inserire x. La funzione hash deve rispettare alcuni requisiti fra i quali quello di mescolare adeguatamente gli elementi in modo che il numero di elementi contenuti in ogni bucket tenda ad essere uguale. Se questo non avviene le operazioni eseguite sugli elementi della struttura tenderebbero ad essere lineari e ciò porterebbe ad una minore efficienza. Un altro fattore importante per l’efficienza della tabella hash è la scelta del numero di bucket, ovvero N. Esso

va scelto in base al numero di elementi da memorizzare e va scelto abbastanza grande in modo che il numero di elementi per bucket sia limitato.

La classe DB contiene due tabelle hash: una è un vettore di liste di Synset denominato appunto tabella, l'altra è un vettore di liste di argomenti chiamata tabArgom. La loro dimensione è pari rispettivamente ad una costante DIM e ad una costante DIM_ARGOM definite entrambe nel file Db.h e fissate a 19.

La funzione hash

La funzione hash utilizzata per entrambe le tabelle ha come parametro d'ingresso una stringa (l'identificativo del Synset e l'identificativo dell'argomento). La funzione più utilizzata per un ingresso di tipo string consiste nel calcolare la somma del valore di ogni singolo carattere appartenente alla stringa e di fornire in uscita l'intero ottenuto come resto della divisione fra questo e il numero di bucket della struttura. Dovendo ottenere in uscita l'indice dell'elemento del vettore in cui memorizzare (o cercare) la lista da inserire nelle tabelle, la funzione appena descritta ci garantisce in uscita un intero compreso fra 0 e N-1, con N dimensione del vettore. Inoltre visto che l'identificatore delle liste da inserire nelle tabelle è variabile si ha che la funzione garantisce, per valori equamente distribuiti, codici hash diversi. Di seguito riportiamo il listato relativo alla funzione hash della classe DB.

```
int Db::fhash(string str) {
    int valore=0; //valore inizialmente pari a zero
    //somma i valori dei singoli caratteri
    for (int i=0;i<str.size();i++)
        valore+= (int)str[i];
    return valore%DIM;
```

```
//restituisce il resto della divisione intera fra valore
//accumulato e la dimensione del vettore
} // fhash
```

Nelle tabelle appena introdotte vengono quindi memorizzati i synset estrapolati dai file nomi.txt e concetti.txt secondo la funzione hash descritta. La costruzione del db viene fatta leggendo riga per riga i source file, elaborandone le righe e memorizzandole in strutture di tipo ListaSynset e ListaArgomento già viste in precedenza.

La ricorsione

Ciò che si richiede dalla ricerca di una forma di parola nel DataBase è di fornire una rappresentazione delle relazioni contenute nel synset. Molte delle relazioni che regolano WordNet hanno caratteristiche tali da creare delle gerarchie. Ovvero ricercare ad esempio gli iponimi della parola significa, attraverso i synset, ricostruire la gerarchia data dalla relazione fino all'ultimo livello. Per questo motivo, si fa uso del metodo ricorsivo. Le relazioni che necessitano della ricorsione sono iponimia, iperonimia, meronimia, omonimia. Per le altre relazioni, la ricorsione non si rende necessaria, in quanto si tratta di relazione non gerarchiche. La funzione di ricerca è:

```
void Db::outFormeCorr(Synset synAnal, int i, string simbolo)
```

dove la lista degli argomenti contiene il synset da analizzare, un intero che indica o meno la necessità di eseguire la ricorsione (valori maggiori o uguali a zero dell'intero *i* indicano che la relazione è di tipo gerarchico) ed il simbolo relativo alla relazione

in esame. La funzione stessa si occupa anche della stampa a video del risultato con le indentazioni necessarie ad una corretta presentazione dei risultati.

6.4 – Il programma grind

In ultima analisi, prendiamo in considerazione il programma grind. Esso si occupa di ricevere in ingresso la forma di parola da ricercare, di creare la base di dati ed infine di lanciare le ricerche di glossa, argomento e relazioni semantiche. Il risultato presentato a video sarà il risultato di questa ricerca.

Capitolo 7 - Inserimento delle biografie dei Matematici e risultati.

7.1 - Introduzione

In questo capitolo parleremo dell'integrazione delle biografie dei Matematici, di formule e Immagini. Accenneremo alla fase di test. Vedremo come vengono presentati i risultati tramite la pagina Internet messa a disposizione dal Dipartimento di Matematica.

Infine illustreremo quali potranno essere gli sviluppi futuri di questo lavoro di tesi.

7.2 - La fase di test sul file nomi.txt

La scrittura del file nomi.txt, è un'operazione molto delicata. Un simbolo erroneamente inserito al posto sbagliato può portare in fase di ricerca ad un ciclo infinito nella ricorsione.

Il primo passo fatto nell'operazione di ristrutturazione della banca dati è stato quello di controllare che tutti gli elementi inseriti in DB fossero coerenti: ciò significa

individuare quali synset fossero mal specificati. Alcuni di loro infatti producevano un errore che mandava in loop l'intero processo di ricerca.

Ciò che si è potuto notare in questa fase è che la struttura del file nomi, contenendo nella stessa riga 3 campi, alle volte anche molto lunghi, porta a generare questo tipo di errori.

La presenza di un errore in una riga, oltre a produrre un loop quando si esegue una ricerca dell'elemento da essa rappresentato, può generare errori anche quando si esegue una ricerca che contenga l'elemento stesso in un punto qualsiasi della gerarchia generata dalle relazioni semantiche.

Per rendere più leggibile e modificabile il file dei nomi si è pensato di separare la parte che elenca le relazioni semantiche da quella che contiene la glossa, introducendo un nuovo file: glossa.txt.

7.3 - Le biografie dei Matematici

Oltre ad un controllo sui dati, ciò che ci si è proposto di fare è arricchire la banca dati con l'inserimento di un nuovo elemento. Presso *la School of Mathematics and Statistics* dell'*University of St.Andrews (Scotland)* è stato implementato un sito che contiene una raccolta di biografie (in lingua inglese) piuttosto complete di Matematici: ***The MacTutor History of Mathematics archive***.

L'indirizzo è il seguente : http://www-history.mcs.st-andrews.ac.uk/.

La completezza di questa base di dati offre l'opportunità di ampliare il WordNet matematico con informazioni biografiche freeware.

Il link è stato inserito nella glossa, in quanto si tratta di una descrizione del lemma: per ogni riga rappresentante un nome di un Matematico si avrà l'indirizzo della pagina della biografia del Matematico.

Facciamo un esempio con Abel, matematico Norvegese vissuto fra il 1802 e il 1829.

La pagina che si apre ricercando Abel nell'elenco alfabetico presente nel sito si può vedere in Figura 7-1, essa ha un indirizzo che si può individuare dalla barra di navigazione.

Nelle righe del file glossa.txt relative ad ogni matematico abbiamo inserito il collegamento alle pagine del sito *The MacTutor History of Mathematics archive* utilizzando codice HTML.

Per ogni Matematico l'indirizzo è del tipo:

```
http://www-history.mcs.st-
andrews.ac.uk/Mathematicians/Nome_del_Matematico.html
```

dove al posto di *Nome_del_Matematico.html* va inserito il nome della pagina html relativa.

La riga relativa al lemma Abel è quindi:

```
Abel <A target="_blank" title="Biografia Completa" HREF="http://www-
history.mcs.st-andrews.ac.uk/Mathematicians/Abel.html">Niels Henrik
Abel 1802-1829 </A>, norvegese
```

Niels Henrik Abel

1802 - 1829

Click the picture above
to see five larger pictures

In 1824 **Abel** proved the impossibility of solving algebraically the general equation of the fifth degree.

Full MacTutor biography [Version for printing]

List of References (14 books/articles)

Some Quotations (6)

A Poster of Niels Abel

Mathematicians born in the same country

Show birthplace location

Honours awarded to Niels Abel
(Click below for those honoured in this way)

Lunar features **Crater Abel**

Paris street names **Rue Abel** (12th Arrondissement)

Other Web sites

1. Encyclopaedia Britannica
2. NNDB
3. MathWorld
4. Oslo Norway
5. Norfolk Va

Previous (Chronologically) Next Main Index

Previous (Alphabetically) Next Biographies index

The URL of this page is:
http://www-history.mcs.st-andrews.ac.uk/Mathematicians/Abel.html

Figura 7-1 Biografia Abel

7.4 - Ricerca e presentazione dei risultati

Il programma, scritto in C++, e compilato tramite compilatore in linea, è stato sviluppato per essere utilizzato lanciandolo da linea di comando con la chiamata:

./grind Forma_di_Parola

Per rendere il database consultabile, è stata creata una pagina HTML , che tramite il metodo POST manda in esecuzione il programma grind e fornisce una nuova pagina con il risultato della ricerca.

L'indirizzo della pagina di ricerca è:

http://www.math.unipd.it/~laurap/grupponlp/wordnet.html

Se lanciamo la ricerca per il nome Abel il risultato ottenuto è quello in Figura 7-2; come si può vedere compaiono l'Argomento, la Definizione con il link alla pagina della biografia e di seguito gli Iponimi, Iperonimi, Meronimi, Omonimi, Sinonimi, Antonimi e Relazionati.

Figura 7-2 Ricerca della forma di parola *Abel*

Vediamo ora invece l'output della ricerca della parola *operazione*: essa instaura relazioni diverse con lemmi presenti nella banca dati.

La gerarchia che si identifica esaminando un downward link dato dagli iponimi si trova in Figura 7-3.

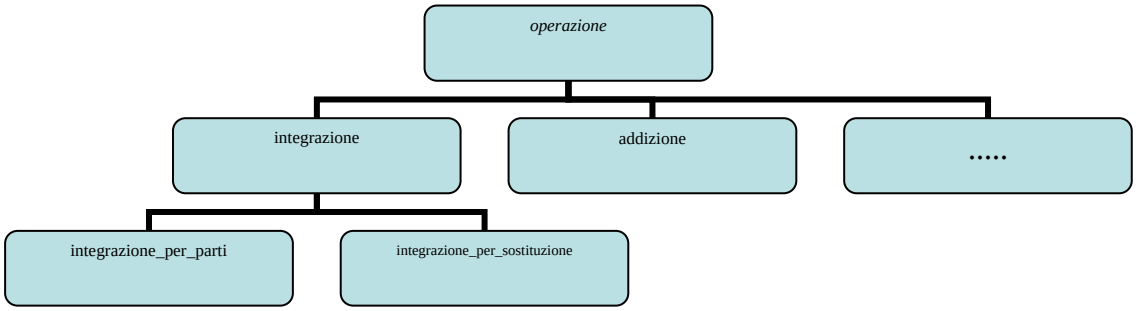


Figura 7-3 Parte della struttura gerarchica data dagli iponimi di *operazione*

Se si lancia la ricerca di *operazione* si ottiene la pagina di Figura 0-4, come si può vedere, sotto la voce IPONIMI troviamo integrazione, iponimo di *operazione*, e subito sotto *integrazione_per_parti* e *integrazione_per_sostituzione*, a loro volta iponimi di *integrazione*.

Parola: operazione

Risultato:

```
ARGOMENTO: Algebra (in generale)
DEFINIZIONE
regola o complesso di regole che associa a n elementi dati in un certo ordine un altro elemento detto risultato dell'operazione
IPONIMI DI operazione
    operazione_esterna
    operazione_interna
    addizione
    sottrazione
    moltiplicazione
    divisione
    estrazione_di_radice
    elevamento_a_potenza
    logaritmo
        logaritmo_decimale
        logaritmo_di_Briggs
        logaritmo_naturale
        logaritmo_nel_campo_complesso
        logaritmo_neperiano
        modulo_di_un_logaritmo
    derivazione
    integrazione
        integrazione_per_parti
        integrazione_per_sostituzione
    prodotto_esterno
    prodotto_scalare
    prodotto_vettoriale
IPERONIMI DI operazione
 Nessun elemento trovato
MERONIMI DI operazione
    risultato
OLONIMI DI operazione
 Nessun elemento trovato
SINONIMI DI operazione
 Nessun elemento trovato
ANTONIMI DI operazione
 Nessun elemento trovato
RELAZIONATI CON operazione
    operatore
    operazione_inversa
```

Figura 0-4 Risultato della ricerca per *operazione*

Se fra gli iponimi di *operazione* scegliamo il lemma *addizione* e ne lanciamo la ricerca del database matematico, Figura 7-5, ritroviamo la parola *operazione* come iponimo e come meronimi *addendo* e *somma*, mentre come antinomo *sottrazione*, operazione inversa dell'addizione.

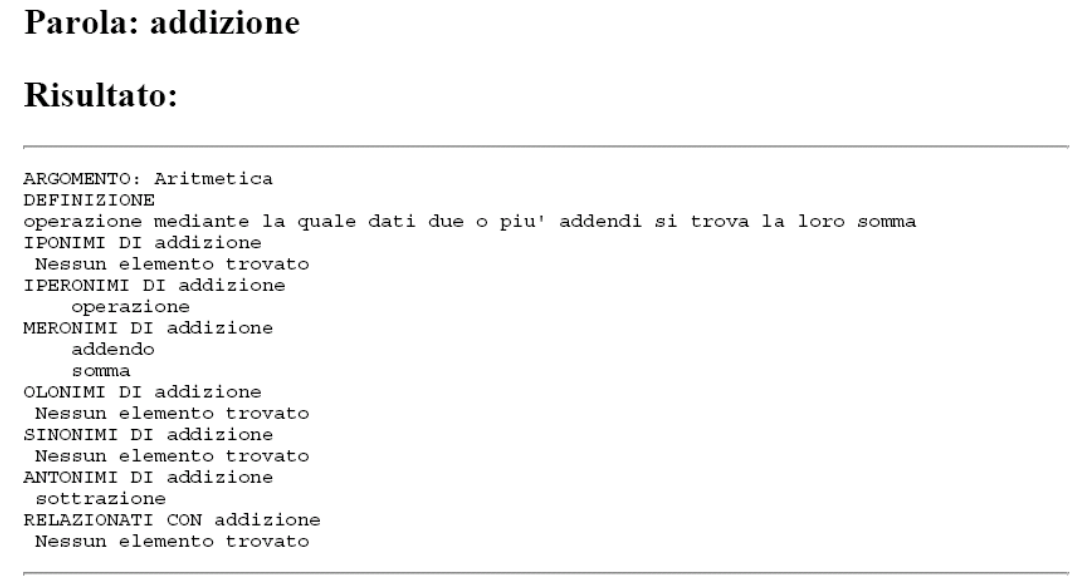


Figura 7-5 Risultato della ricerca per *addizione*

Passiamo infine ad illustrare l'ultima introduzione fatta nel file glossa.txt: per alcune forme di parola, si rende necessaria la presenza di un'immagine che possa far risultare un grafico, una formula o una forma più chiari di una semplice spiegazione a parole. Prendiamo ad esempio il lemma matrice, la sua glossa originaria era:

"tabella rettangolare formata da nxm elementi, spesso numeri, disposti in n righe ed m colonne"

Per un utente "esperto", la definizione è abbastanza chiara, mentre per un utente che si appresta ad affrontare per la prima volta la forma matriciale, la definizione non lo aiuta affatto ad immaginarne la rappresentazione.

Per questo motivo si è pensato di utilizzare HTML anche per inserire, oltre a quanto visto per i Matematici, delle immagini. Esse saranno mantenute in cartelle distinte a seconda dell'elemento che si va a rappresentare: formule e immagini.

Parola: matrice

Risultato:

ARGOMENTO: Geometria piana e solida, trigonometria, vettori, matrici
DEFINIZIONE

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

tabella rettangolare formata da nxm elementi, spesso numeri, disposti in n righe ed m colonne
IPONIMI DI matrice
    matrice_diagonale
    matrice_hermitiana
    matrice_identica
    matrice_jacobiana
    matrice_ortogonale
    matrice_quadrata
    matrice_simmetrica
    matrice_singolare
    matrice_trasposta
    matrice_triangolare
IPERONIMI DI matrice
    tabella
MERONIMI DI matrice
    digonale
    colonna
    riga
    minore
OLONIMI DI matrice
 Nessun elemento trovato
SINONIMI DI matrice
 Nessun elemento trovato
ANTONIMI DI matrice
 Nessun elemento trovato
RELAZIONATI CON matrice
    determinante

Figura 7-6 Risultato della ricerca per *matrice*

Lanciando la ricerca della forma di parola *matrice*, Figura 7-6, otteniamo direttamente nella pagina del risultato un'immagine che illustra la forma di una matrice, mentre per il lemma *equazione_di_Legendre*, Figura 7-7, abbiamo in riferimento alla vecchia glossa in formato testo l'inserimento di un collegamento ipertestuale che apre una pagina con la formula dell'equazione di Legendre scritta in maniera più leggibile:

$$\frac{d}{dx}\left[(1-x^2)\frac{d}{dx}P(x)\right] + n(n+1)P(x) = 0$$

contenuta nel file equazione_di_legendre.png

Parola: equazione_di_Legendre

Risultato:

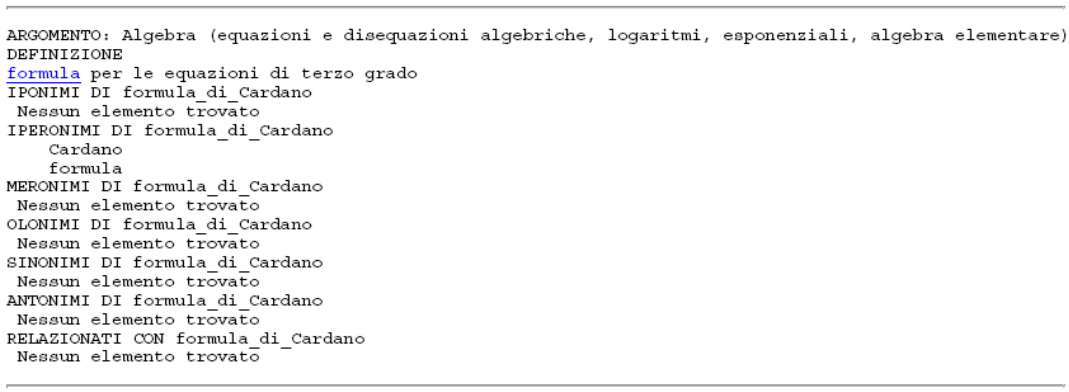
```
ARGOMENTO: Equazioni differenziali
DEFINIZIONE
equazione differenziale della forma [1-x^2]y''-2xy'+n[n+1]y=0
IPONIMI DI equazione_di_Legendre
 Nessun elemento trovato
IPERONIMI DI equazione_di_Legendre
    equazione
    Legendre
MERONIMI DI equazione_di_Legendre
 Nessun elemento trovato
OLONIMI DI equazione_di_Legendre
 Nessun elemento trovato
SINONIMI DI equazione_di_Legendre
 Nessun elemento trovato
ANTONIMI DI equazione_di_Legendre
 Nessun elemento trovato
RELAZIONATI CON equazione_di_Legendre
 Nessun elemento trovato
```

Figura 7-7 Risultato della ricerca per *equazione_di_Legendre*

Ovviamente possiamo collegare alla glossa anche documenti di tipo pdf, come abbiamo fatto per esempio con il lemma formula_di_Cardano per il quale cliccando sul link contenuto nella glossa, Figura 7-8, si apre il file formula_di_Cardano.pdf contenente quanto illustrato in Figura 7-9.

Parola: formula_di_Cardano

Risultato:



```
ARGOMENTO: Algebra (equazioni e disequazioni algebriche, logaritmi, esponenziali, algebra elementare)
DEFINIZIONE
formula per le equazioni di terzo grado
IPONIMI DI formula_di_Cardano
 Nessun elemento trovato
IPERONIMI DI formula_di_Cardano
    Cardano
    formula
MERONIMI DI formula_di_Cardano
 Nessun elemento trovato
OLONIMI DI formula_di_Cardano
 Nessun elemento trovato
SINONIMI DI formula_di_Cardano
 Nessun elemento trovato
ANTONIMI DI formula_di_Cardano
 Nessun elemento trovato
RELAZIONATI CON formula_di_Cardano
 Nessun elemento trovato
```

Figura 7-8 Risultato della ricerca per *formula_di_Cardano*

Formula di Cardano per l'equazione del tipo $x^3 + px = q$, a cui può ridursi ogni equazione del tipo generico $ax^3 + bx^2 + cx + d = 0$:

$$x = \sqrt[3]{\frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} - \sqrt[3]{-\frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}$$

Figura 7-9 Formula_di_Cardano

I link nella glossa vengono specificati attraverso il Tag HREF.

7.5 - Considerazione finali e sviluppi futuri

Tramite questa tesi, sono stati introdotte integrazioni al wordnet matematico che ne rendessero più semplice l'estensione, intesa come incremento delle forme di parole in esso contenute. Con la divisione in due parti del file nomi si auspica di aver dato un aiuto a chi si proporrà di inserire nuovi lemmi, riducendo la possibilità di commettere errori. Inoltre si è indicato un cammino per l'introduzione di formule e figure che rendano la base di dati consultabile in maniera proficua, chiara e più esaustiva possibile.

Anche lo scopo dell'introduzione delle biografie dei Matematici si inserisce in questo obiettivo, oltre a quello di mettere in relazione risorse diverse, sviluppate in contesti diversi.

Ciò che ora, rimane da completare potrebbe essere la parte di collegamento con un wordnet generico multilingue.

Nel capitoli precedenti si è parlato di due tipi di approcci al multilinguismo e di come essi abbiano portato a due soluzioni diverse: EuroWordNet e MultiWordNet.

Successivamente abbiamo dato esempio di alcune esperienze di costruzione di wordnet specialistici e di come essi siano stati integrati con risorse generiche. Si potrebbe quindi pensare di collegare il presente database ad EuroWordNet, in particolare a ItalWordNet. Esso è il modello che consente più libertà rispetto alla costruzione dei wordnet specialistici data la presenza di un collegamento indipendente dalle basi di dati ad esso collegate (la struttura ILI).

Per ultimo vogliamo citare anche la possibilità di inserire la categoria lessicale degli aggettivi. Di questi esiste già un primo file sorgente agg.txt, anche se formalmente non ancora molto preciso.

Bibliografia

[1] WordNet: An Electronic Lexical Database (1990), MIT Press

[2] WordNet Reference Manual

[3] **Bentivogli Luisa, Bocco Andrea, Pianta Emanuele.** *ArchiWordNet: Integrating WordNet with Domain-Specific Knowledge.* In Proceedings of the Second Global WordNet Conference, Brno, Czech Republic, January 20-23, 2004, pp. 39-46.

[4] **Croin Stefano.** *Sviluppo di un dizionario matematico nella struttura di WordNet*. Tesi di Laurea Università degli Studi di Padova, Facoltà di Ingegneria. A.A. 2002-2003

[5] **Giunta A., Minnaja C., Paccagnella L.G.** *Extending the Italian WordNet with the Specialized Language of the Mathematical Domain.* Grundlagenstudien aus Kibernetik und Geisteswissenschaft, vol.46, pp. 3-12, ISSN: 0723 – 4899 , 3 , 2005.

[6] **Lippman, Stanley B.e Lajoie, Josèe.** *C++ Corso di programmazione (terza edizione)* (2000), Addison - Wesley.

[7] **Magnini Bernardo, Speranza Manuela.** *Integrating Generic and Specialized WordNets.* In *Proceedings of Recent Advances in Natural*

Language Processing, RANLP-2001, Tzigov Chark, Bulgaria, 2001, pp. 149-153.

[8] **Magnini Bernardo, Speranza Manuela.** *Merging Global and Specialized Linguistic Ontologies*. In Proceedings of the Workshop Ontolex-2002 Ontologies and Lexical Knowledge Bases, LREC-2002, pp. 43-48.

[9] **Magnini Bernardo, Strapparava Carlo.** *Costruzione di una base di conoscenza lessicale per l'italiano basata su WordNet*. XXVII Congresso Internazionale di Studi della Societa' di Linguistica Italiana "Linguaggio e Cognizione",Palermo, 27-29 October 1994, Bulzoni, Roma, 1997.

[10] **Marinelli R., Roventini A., Enea A.** *Building a Maritime Domain Lexicon: a Few Considerations on the Database Structure and the Semantic Coding*. LREC 2004: Fourth International Conference on Language Resources and Evaluation

[11] **Miller, G.A. e Beckwith, R. e Fellbaum, C. e Gross, D. e Miller, K.** *WordNet: an on-line lexical database (1990),* International Journal of Lexicography (special issue), 3 (4), pp. 235-312.

[12] **Miller, G.A. e Beckwith, R. e Fellbaum, C. e Gross, D. e Miller, K.** *Introduction to WordNet: an on-line lexical database (1993),* http://www.cogsci.princeton.edu/ wn/.

[13] **Morato Jorge, Marzal Miguel Angel, Lloréns Juan, Moreiro José.** *WordNet Applications.* Sojka et al., 2004, pp. 270-278.

[14] **Powell Thomas A. .** *HTLM la Reference*. McGraw-Hill

[15] **Pianta Emanuele, Bentivogli Luisa, Girardi Christian.** *MultiWordNet: developing an aligned multilingual database*. Proceedings of the First

International Conference on Global WordNet, Mysore, India, January 21-25, 2002.

[16] **Rasi Roberto.** *Ontologie Lessicali Multilingua: MultiWordNet ed EuroWordNet*. Tesi di Laurea Università degli Studi di Modena e Reggio Emilia, Facoltà di Ingegneria. A.A. 2002-2003

[17] **Roventini Adriana and Marinelli Rita.** *Extending the Italian WordNet with the Specialized Language of the Maritime Domain*. In Sojka et al., pp. 193-198.

[18] **Sagri Maria Teresa, Tiscornia Daniela (Institute for Theory and Techniques for Legal Information, CNR, Firenze, Italy), Bertagna Francesca (Istituto di Linguistica Computazionale, CNR, Pisa, Italy).** *Jur-WordNet* In Proceedings of the Second Global WordNet Conference, Brno, Czech Republic, January 20-23, 2004, pp. 305-310

[19] **Vossen, P.** *EuroWordNet General Document*. Piek Vossen (ed.) , 2007

[20] *Tassonomie.* http://www.ce.unipr.it/people/bianchi/Teaching/IntelligenzaArtificiale/WebSemantico_Ontologie/Tassonomie.pdf

[21] EuroWordNetBuilding http://www.illc.uva.nl/EuroWordNet/

[22] GlobalWordNet http://www.globalwordnet.org/

[23] Istituto di Linguistica Computazionale http://www.ilc.cnr.it/indexflash.html

[24] ItalWordNet http://tcc.itc.it/research/textec/topics/multiwordnet/

[25] Programmazione.it

http://www.programmazione.it/

[26] WordNet

http://www.cogsci.princeton.edu/~wn/

[27] Lexical Ontologies for legal Information Sharing (LOIS)

http://www.loisproject.org/

Indice delle figure

Indice delle tabelle