Università degli Studi di Modena e Reggio Emilia

Facoltà di Ingegneria di Modena

Corso di Laurea in Ingegneria Informatica

Integrazione di WordNet Domains all'interno del sistema MOMIS

Relatore:
Chiar.mo Prof. Sonia Bergamaschi

Candidato:
Sawzar Rashid

Anno Accademico 2007/2008

PAROLE CHIAVE

WordNet

MOMIS

Domains

Disambiguazione

Lessicale

Integrazione dei domini

Sommario

Indice delle illustrazioni

Indice delle tabelle

Introduzione

Di fronte ad un cambiamento epocale, come l'esplosione di internet e del Web, noi ci troviamo ad affrontare problematiche, forse non del tutto nuove nella storia dell'umanità, trasportate in un contesto che ci costringe a nuove soluzioni.
Infatti, la diffusione delle reti di comunicazioni, Internet *in primis*, ha accentuato i problemi che attengono alla gestione delle informazioni. Oggi la fruizione di dati, grazie alla disponibilità di supporti e mezzi informatici, si è fatta molto più capillare e diffusa di quanto non fosse non più di venti anni fa.
Tuttavia, il contesto in cui ci si muove è quello di una quantità di sorgenti informative da cui si può attingere ma che sono al tempo stesso, anche molto eterogenee tra di loro.
Da un punto di vista più strettamente pratico ed economico, le aziende o gli operatori che in qualche maniera devono gestire flussi informativi, sono costretti a misurarsi con la frammentazione di queste informazioni tra una moltitudine di formati,supporti, metodi di accesso ecc...

Nell'ambito di una molteplicità di sorgenti di dati, nascono perciò i sistemi per l'Integrazione Intelligente delle Informazioni (I^3), che si propongono di ottenere in maniera automatica una selezione ragionata dei dati provenienti dalle varie sorgenti e forniscono una loro fusione intelligente. Tra questi sistemi I^3, MOMIS (**M**ediator Envir**o**nment for **M**ultiple **I**nformation **S**ource) è ideato per l'integrazione di sorgenti di dati testuali, strutturati e semi-strutturati.
Uno dei problemi fondamentali di tale sistema deriva dalla semantica dei dati; va da se che se ad esempio due o più persone (le sorgenti) che descrivono la stessa realtà in maniera differente oppure, nonostante condividano gli stessi concetti ontologici ma li denotano con nomi diversi, sollevano una questione di semantica nella loro rappresentazione.
Perciò è chiaro che l'integrazione dovrà partire dalla risoluzione delle differenze semantiche tra le diverse rappresentazioni dei dati; MOMIS utilizza a tale scopo, il metodo dell'annotazione semantica delle varie sorgenti.
L'annotazione di un termine comporta in maniera implicita un procedimento di disambiguazione del termine stesso. La disambiguazione è un processo grazie al quale si può

assegnare ad ogni termine analizzato un preciso significato e, all'interno del sistema MOMIS , questa operazione avviene con l'ausilio di un database lessicale chiamato WordNet.
WordNet è una risorsa lessicale che permette di assegnare ad ogni parola presente all'interno del suo vocabolario, uno o più significati. Tale proprietà può venire sfruttata in combinazione con algoritmi di disambiguazione di varia natura.
WordNet tuttavia, ha mostrato alcune lacune e limiti, che vedremo più avanti, che hanno portato a pensare a nuove soluzioni. Si è ritenuto utile perciò integrare una risorsa ulteriore a WordNet, vale a dire una estensione che desse qualcosa in più: il WordNet Domains.
WordNet Domains aggiunge alle informazioni sui termini ed i significati connessi a tali termini dell'ontologia di WordNet, quello di dominio; tale informazione permette una maggiore efficacia nell'operazione di disambiguazione.
In questa tesi si tesi si studierà l'estensione di WordNet Domains e ci si concentrerà in particolare su come poter integrare le gerarchie di dominio con WordNet all'interno del database "momiswn" creato all'interno del sistema MOMIS .
Il contenuto della tesi è così strutturato:

- Capitolo 1 **Il sistema MOMIS** in cui si parlerà dei sistemi I^3 con particolare riferimento alla struttura funzionamento di MOMIS.
- Capitolo 2 **WordNet e MOMIS** in cui si descriverà sommariamente WordNet e il suo ruolo in Momis.
- Capitolo 3 **WordNet Domains** nel quale di studierà più nel dettaglio questo pacchetto e la sua utilità nella disambiguazione dei termini.
- Capitolo 4 I**ntegrazione di WordNet Domains e WordNet in MOMIS** che è il nucleo del mio lavoro di tesi e descrive la realizzazione del processo di integrazione di WordNet Domains. In particolare si descriverà l'adattamento delle informazioni di WordNet Domains, rispetto alla strutture preesistenti in MOMIS.

Capitolo 1 Il sistema MOMIS

1.1 MOMIS

Un problema cui devono ormai far fronte numerose imprese ed organizzazioni, è quello della dispersione del loro patrimonio informativo. Si pensi ai numerosissimi metodi di immagazzinamento di informazioni presenti sul mercato o utilizzabili gratuitamente: DBMS, pagine HTML, pagine XML ecc…

Nel caso in cui un utente voglia reperire informazioni da sorgenti diverse, fatto che accade sempre più frequentemente ogni giorno, si trova di fronte a problemi di non facile soluzione: le sorgenti di conoscenza, infatti, sfrutteranno tecnologie differenti, difficilmente uniformabili, senza contare le possibili contraddizioni ed inconsistenze fra i dati ottenuti da diverse fonti. Per quanto concerne il problema dello sfruttamento di tecnologie differenti gli standard esistenti, (come l'ODBC, CORBA ecc…), risolvono parecchi problemi di comunicazione fra moduli diversi. Ciò che rimane irrisolta, è la questione della modellazione delle informazioni: i modelli dei dati, possono differenziarsi gli uno dagli altri, a tal punto da fornire ognuno una propria struttura logica di rappresentazione dei dati da immagazzinare. Tutto ciò crea un'eterogeneità semantica non risolvibile dagli attuali standard. Da quanto descritto in precedenza, si evincono le difficoltà che sorgono nel creare un sistema di integrazione e mediazione delle informazioni eterogenee che sia affidabile, flessibile, modulare (in modo da consentire il riuso delle diverse parti all'evolvere delle tecnologie), e capace di interagire con altri sistemi esistenti. Nel seguito si descriverà una proposta di ARPA (*Advanced Reserch Project Agency*) per un'architettura di integrazione di informazioni, flessibile e riusabile. L'approccio descritto dall'ARPA in [2], è stato seguito anche nel progetto MOMIS.

1.2 L'Integrazione Intelligente delle Informazioni

Come viene citato in [3], l'integrazione delle informazioni (I*2*) non cerca di collegare semplicemente alcune sorgenti, ma risultati opportunamente selezionati da esse. Lo scopo dell'integrazione dell'informazione è, quindi, quello di ottenere una selezione ragionata dei dati prelevati dalle varie sorgenti, e produrre una fusione intelligente ed una seguente sintesi degli stessi. Proprio a questo scopo, è stato sviluppato dall'ARPA, un progetto di ricerca atto a fornire un'architettura di riferimento, che realizzi l'integrazione di risorse eterogenee in maniera automatica; il nome di questo progetto è appunto I^3 (Integrazione Intelligente dell'Informazione). Secondo il programma I^3 è opportuno costruire architetture modulari, in grado di abbassare i costi di sviluppo e mantenimento, eseguite seguendo uno standard che ponga le basi dei servizi necessari all'integrazione.

1.2.1 L'architettura dei sistemi I^3

L'architettura del programma I^3 definita dall'ARPA, deriva la sua forma dal tentativo di far fronte ai problemi complessi come quelli citati nei paragrafi precedenti, e che vengono qui brevemente riproposti in un elenco quali l'eterogeneità delle sorgenti, la loro evoluzione, le loro dimensioni, le semantiche differenti da dedurre e, l'importante necessità di poter disporre di sistemi modulari e riusabili per contenere costi e tempi di sviluppo delle varie applicazioni e far fronte ai mutamenti tecnologici.

L'architettura del progetto I^3 si propone di evidenziare (separando in più moduli), i vari servizi che devono essere svolti ai fini dell'integrazione intelligente d'informazioni. I servizi evidenziati a questo proposito sono cinque:

- *Servizi di coordinamento:* svolgono i lavori di supporto, sia in fase di progettazione di nuove configurazioni, che a tempo di esecuzione delle richieste dell'utente. Questi servizi sono di alto livello e, oltre ad individuare quali sorgenti possono essere utili per soddisfare una data richiesta, presentano all'utente finale l'intero sistema, diviso fra i suoi vari moduli, come un blocco unico ed omogeneo. Grazie ai servizi di coordinamento, quindi, le divisioni interne di un sistema I^3 , sono trasparenti all'utente
- *Servizi di amministrazione:*

- *Servizi di integrazione e trasformazione semantica*: Hanno come input una o più sorgenti di dati tradotte dai servizi di *Wrapping,* e, come output, la "vista" integrata o trasformata di queste informazioni. Essi vengono indicati spesso come servizi di mediazione.
- *Servizi di wrapping*: fungono da interfaccia tra il sistema integratore e le singole sorgenti, in particolare, rendendo omogenee le informazioni. Si comportano come dei traduttori dai sistemi locali ai servizi di alto livello dell'integratore. Il loro obiettivo è, quindi, quello di standardizzare il processo di *wrapping* delle sorgenti, permettendo la creazione di una libreria di fonti accessibili.
- *Servizi ausiliari*: aumentano le funzionalità degli altri servizi e sono utilizzati prevalentemente dai moduli che agiscono direttamente sulle informazioni; essi vanno dai semplici servizi di monitoraggio del sistema, ai servizi di propagazione degli aggiornamenti e di ottimizzazione.

Fra tutti quelli elencati in precedenza, i servizi principali sono quelli di coordinamento il cui scopo è, appunto, quello di coordinare le operazioni attuate dai vari servizi sia in fase di progettazione dei vari link di integrazione fra le sorgenti, sia in fase di esecuzione (in tempo reale) su specifiche richieste dagli utenti.

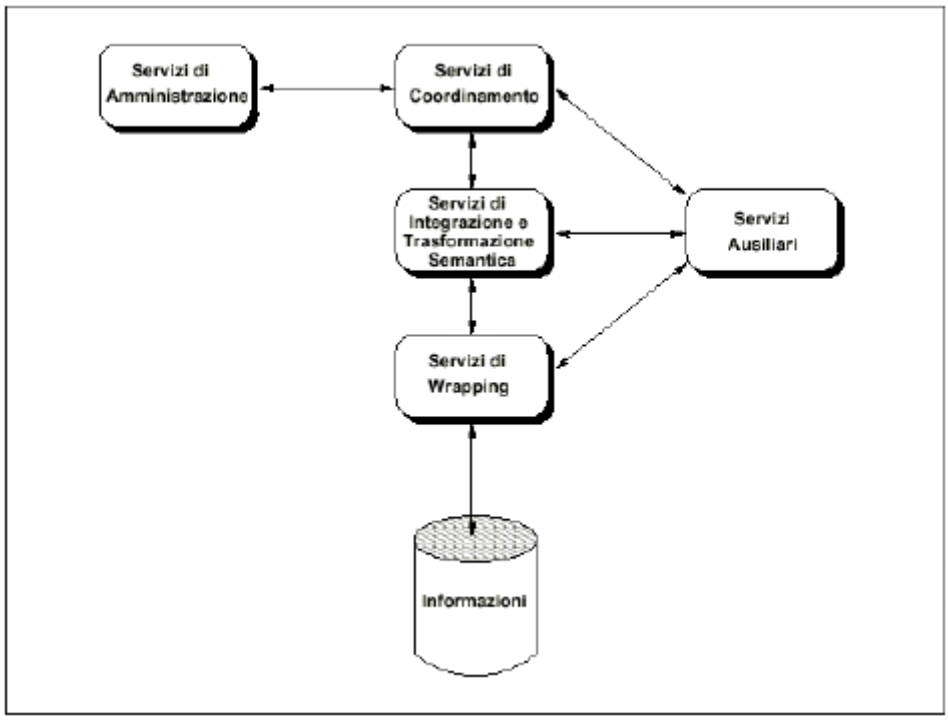


Figura 1.1: Diagramma dei servizi I [3]

1.2.2 Il mediatore

Il Mediatore è un modulo intermedio che si pone tra l'utente e le sorgenti d'informazione. Secondo al definizione di Wiederhold in [**3**] " un Mediatore è un modulo software che sfrutta la conoscenza su un certo livello superiore. Dovrebbe essere piccolo e semplice, così da poter essere amministrato da uno o al più, da pochi esperti."

I compiti del Mediatore sono:

- Assicurare un servizio stabile, anche quando cambiano le risorse;
- Amministrare e risolvere le eterogeneità delle diverse fonti;
- Integrare le informazioni ricavate da più risorse;
- Presentare all'utente le informazioni attraverso un modello scelto dall'utente stesso.

L’approccio architetturale adottato, è quello classico, che consta principalmente di tre livelli:

- *utente*: attraverso un’interfaccia grafica l’utente pone delle query su uno schema globale e riceve risposta, come se stesse interrogando un’unica sorgente d’informazioni.
- *mediatore*: il Mediatore gestisce l’interrogazione dell’utente, combinando, integrando ed, eventualmente, arricchendo i dati ricevuti dai wrapper, ma usando un modello (e quindi un linguaggio interrogatore) comune a tutte le fonti;
- *wrapper*: ogni wrapper gestisce una sorgente, ed ha una duplice funzione: da un lato converte le richieste del Mediatore in una forma comprensibile dalla sorgente, dall’altro traduce informazioni estratte dalla sorgente nel modulo usato dal mediatore.

L’approccio utilizzato in MOMIS, è caratterizzato dal fatto che il Mediatore deve conoscere, per ogni sorgente, lo schema concettuale (metadati); le informazioni semantiche sono codificate in questi schemi, deve essere disponibile un modello comune per descrivere le informazioni da condividere e i metadati, e infine, deve essere possibile un’integrazione (parziale o totale) delle sorgenti di dati. In questo modo il Mediatore può individuare i concetti comuni a più sorgenti e relazioni che li legano.

1.2.3 Problemi d'affrontare

Pur avendo a disposizione gli schemi concettuali delle varie sorgenti, non è certamente un compito facile individuare i concetti comuni ad essi, le relazioni che possono legarli, né tanto meno realizzare una loro coerente integrazione. Tralasciando le differenze dei sistemi fisici (alle quali dovrebbero pensare i moduli wrapper), i problemi che si sono dovuti risolvere, o con i quali occorre giungere a compromessi, sono (a livello di mediazione, ovvero di integrazione delle informazioni) essenzialmente tipo ontologico e semantico. I primi sono essenzialmente dovuti al fatto che sorgenti diverse possono riferirsi ad ontologie diverse ossia, domini di conoscenza e perciò una visione dei problemi diversi; i secondi si presentano qualora, pur ipotizzando che anche sorgenti diverse condividano un insieme di concetti comuni, niente ci assicura che diversi sistemi usino esattamente gli stessi vocaboli per rappresentare questi concetti, ne tanto meno le stesse strutture dati. Poiché le diverse strutture dati sono state progettate e modellate da persone differenti, è molto improbabile che queste persone condividano la stessa "concettualizzazione" del mondo esterno, ovvero non esiste nella realtà una semantica univoca cui chiunque possa riferirsi.

Come riportato in [**4**], la causa principale delle differenze semantiche, si può identificare nelle diverse concettualizzazioni del mondo esterno che le persone distinte possono avere, ma questa non è l'unica. Le differenze nei sistemi DBMS, possono portare all'uso di differenti modelli per la rappresentazione della porzione del mondo in questione; partendo così dalla stessa concettualizzazione, determinate relazioni fra concetti, avranno strutture diverse a seconda che siano state realizzate, ad esempio, attraverso un modello relazionale o un modello ad oggetti.

L'obiettivo dell'integratore, che ricordiamo è di fornire un accesso integrato ad un insieme di sorgenti, si traduce nel non facile compito di identificare i concetti comuni all'interno delle sorgenti, e risolvere le differenze semantiche che possono essere presenti. Possiamo classificare queste incoerenze semantiche in tre gruppi principali:

- *Eterogeneità tra le classi di oggetti*: benché due classi in due differenti sorgenti rappresentano lo stesso concetto nello stesso contesto, possono usare nomi diversi per

gli stessi attributi, per i metodi, oppure avere gli stessi attributi con domini di valori diversi;

- *Eterogeneità tra le strutture delle classi*: comprendono le differenze nei criteri di specializzazione, nelle strutture per realizzare un'aggregazione, ed anche le discrepanze schematiche;
- *Eterogeneità nelle istanze delle classi*: ad esempio l'uso di diverse unità di misura per i domini di un attributo, o la presenza/assenza di valori nulli.

1.3 L'architettura di MOMIS

MOMIS acronimo di ***M**ediator **E**nvir**O**ment for **M**ultiple **I**nformation **S**ources*, è il progetto di un sistema I^3, ideato per l'integrazione di sorgenti di dati testuali, strutturati e semi-strutturati. MOMIS nasce all'interno del progetto MURST 40% INTERDATA, come collaborazione fra le unità operative dell'Università di Milano e dell'Università di Modena e Reggio Emilia.

MOMIS è stato progettato per fornire un accesso integrato ad informazioni eterogenee, memorizzate sia all'interno di un *database* di tipo tradizionale (e.g. relazionali, *object-oriented*) o *file system* sia in sorgenti di tipo semi-strutturato come quelle descritte in XML.

Seguendo l'architettura di riferimento in [**2**] si possono distinguere i componenti disposti su tre livelli (figura 1.2):

- **Livello dati.** Qui si trovano i *Wrapper*. Posti al di sopra di ciascuna sorgente, sono i moduli che rappresentano l'interfaccia fra il Mediatore e le sorgenti di dati locali. La loro funzione è duplice:
  1. In fase d'integrazione forniscono la descrizione dell'informazione in essi contenute. Questa descrizione è fornita attraverso il linguaggio ODL_{I^3}.
  2. In fase di *query processing* traducono la query ricevuta dal Mediatore(espressa quindi nel linguaggio comune d'interrogazione $OQLI^3$ **,** definito a partire dal linguaggio OQL) in un'interrogazione comprensibile dalla sorgente stessa. Devono, inoltre, esportare i dati ricevuti come risposta all'interrogazione, presentandoli al mediatore, attraverso il modello comune di dati utilizzato dal

sistema.

- **Livello Mediatore.** Il Mediatore rappresenta il cuore del sistema ed è essenzialmente composto da due sotto moduli:
  - ***Global Schema Builder*** (GSB): è il modulo che integra gli schemi locali, il quale partendo dalle descrizioni delle sorgenti espresse, attraverso il linguaggio ODL$_{I^3}$ genera un unico schema globale da presentare all'utente. L'interfaccia grafica di GSB, cioè il *tool* d'ausilio al progettista, è *SI-Designer*.
  - ***Query Manager*** (QM)*:* è il modulo di gestione delle interrogazioni. In particolare, genera la *query* in linguaggio ODL$_{I^3}$ da inviare ai *wrapper,* partendo dalla singola *query* formulata dall'utente sullo schema globale. Servendosi delle tecniche di *Description Logics* di ODB-Tools, il QM genera automaticamente la traduzione della query sottomessa nelle corrispondenti sub-query da sottoporre ai wrapper (query e sotto-query sono espresse in linguaggio OQLI^3).
  - ***SI-Designer****: è la* GUI (Graphic User Interface) che guida l'utente attraverso le varie fasi dell'integrazione, dall'acquisizione delle sorgenti, fino alla messa a punto del Common Thesaurus. SI-Designer risulta a sua volta composto da quattro moduli tra cui:
    1. SIM (*Source Integrator Module*): estrae le relazioni inter-schema sulla base della struttura delle classi ODL$_{I^3}$ e delle sorgenti relazionali usando ODB-Tools. Inoltre effettua la "validazione semantica" delle relazioni e ne inferisce delle nuove sfruttando sempre ODB-Tools
    2. SLIM (S*ource Lexical Intergrator Module*): estrae le relazioni inter-schema tra nomi di attributi e classi ODL$_{I^3}$, sfruttando il database lessicale WordNet.
    3. TUNIM (*Tuning of the Mapping Table*): questo modulo gestisce la fase di creazione dello schema globale.

  La GUI di SI-Designer, è una sequenza di finestre, ognuna delle quali relativa ad una fase del processo d'integrazione, e mette a disposizione l'interfaccia per interagire con i moduli SIM, SLIM ed ARTEMIS.
- **Livello Utente.** Il progettista interagisce con il Global Schema Builder e crea la vista integrata delle sorgenti; l'utente formula le interrogazioni sullo schema globale,

passandole come input al Query Manager, che interrogherà le sorgenti e fornirà all'utente la risposta cercata.

Nella figura 1.2 compaiono inoltre, altri tre tool che accompagnano il Mediatore nella fase di integrazione e sono:

- *ODB-Tools Engine:* un tool basato sulle *Description Logics* [**5, 6**] che compie la validazione di schemi e l'ottimizzazione di query [**7, 8, 9**].
- *ARTEMIS-Tool Enviroment*: tool basato sulle tecniche di *clustering affinity-based* che compie l'analisi ed il clustering delle classi ODL I ^{3} [**10**].
- *WordNet*: un database lessicale ampiamente descritto nel capitolo 2.

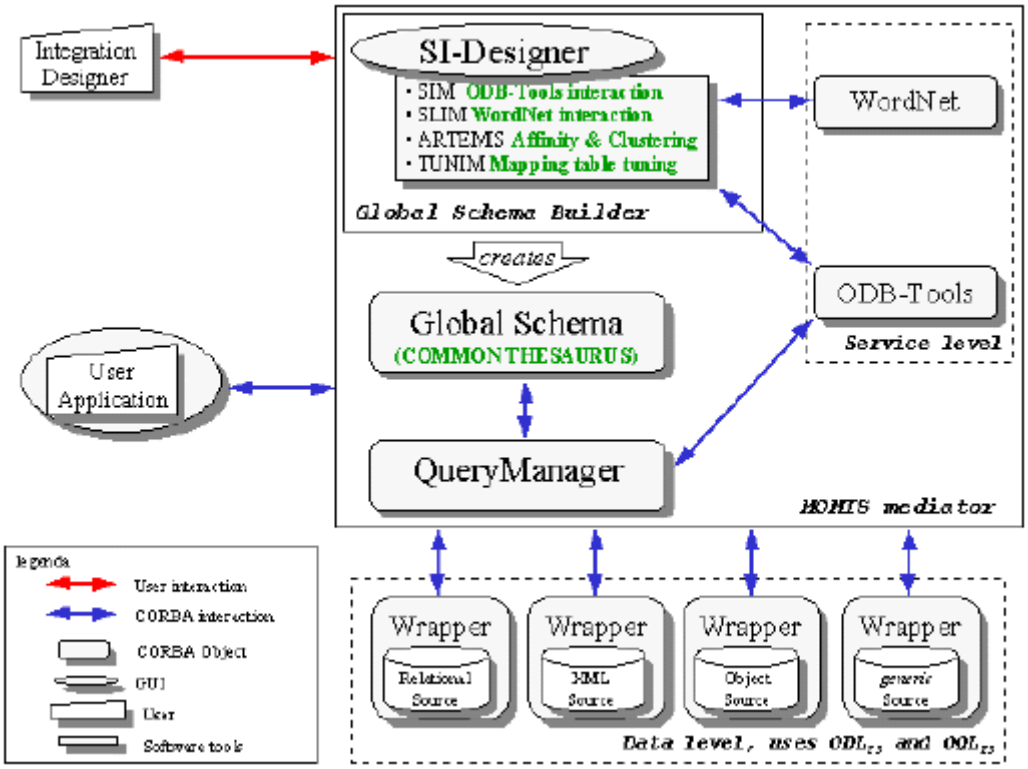


Figura 1.2: Architettura generale del sistema MOMIS

Lo scopo principale a cui ci si è proposti con MOMIS, è la realizzazione di un sistema di mediazione che, a differenza di molti altri progetti, contribuisca a realizzare, oltre alla fase di query processing, una reale integrazione delle sorgenti.

1.3.1 Il processo di Integrazione

L'integrazione delle sorgenti informative strutturate e semi-strutturate, è compiuta in modo semi-automatico, utilizzando degli schemi locali in linguaggio O D L $_{I^3}$, e combinando tecniche di *Description Logic* e di *clustering*. Come mostrato in figura 1.3, le attività compiute sono le seguenti:

1. *Generazione del Thesaurus Comune*, con il supporto di ODB-Tool e di WordNet. In questa fase è identificato un Thesaurus comune di relazioni terminologiche. Tali relazioni, esprimono la conoscenza inter-schema su sorgenti diverse e corrispondono alle asserzioni intenzionali utilizzate in [**11**]. Le relazioni terminologiche, sono derivate in modo semi-automatico a partire dalle descrizioni degli schemi in O D L $_{I^3}$, attraverso l'analisi strutturale (utilizzando ODB-Tools e le tecniche di *Description Logics*) e di contesto (attraverso l'uso di WordNet) delle classi coinvolte.
2. *Generazione dei cluster di classi ODL* I^3 con il supporto dell'ambiente ARTEMIS-Tool. Le relazioni terminologiche contenute nel Thesaurus, sono utilizzate per valutare il livello di affinità tra le classi O D L $_{I^3}$ in modo da identificare le informazioni che devono essere integrate a livello globale. A tal fine ARTEMIS, calcola i coefficienti che misurano il livello di affinità tra le classi, basandosi sia sui nomi delle stesse sia sugli attributi. Le classi con maggiore affinità sono raggruppate utilizzando tecniche di clustering [**12**].
3. *Costruzione dello Schema Globale*. I cluster delle classi ODL$_{I^3}$ affini, sono analizzati per costruire lo schema globale del Mediatore. Per ciascun cluster si definisce una classe globale che rappresenta tutte le classi locali riferite al cluster ed è caratterizzata dall'unione ragionata dei loro attributi e da una *mapping-table*. L'insieme delle classi globali definite, costituisce lo schema globale del Mediatore che sarà usato per porre le query alla sorgenti locali integrate.

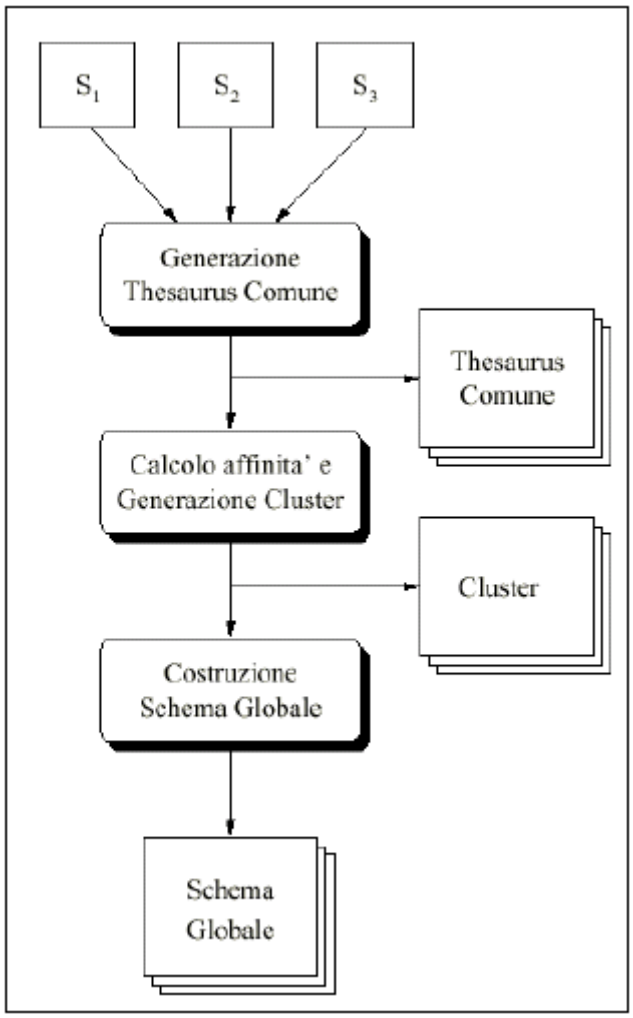


Figura 1.3: Fasi del processo d'integrazione

1.3.2 Query Processing e Ottimizzazione

Quando un'utente pone una query sullo schema globale, MOMIS la analizza e produce un insieme di sotto-query che saranno inviate a ciascuna sorgente informativa coinvolta. Il processo consiste di due attività principali:

- *Ottimizzazione Semantica.* L'ottimizzazione semantica è basata sull'inferenza logica a partire dalla conoscenza contenuta nei vincoli d'integrità dello schema globale. La stessa procedura di ottimizzazione semantica, si realizza in termini locali, su ogni

sotto-query tradotta dal Mediatore nella formulazione del piano d'accesso: in tal caso ci si basa sui vincoli d'integrità presenti sui singoli schemi locali.

- *Formulazione del piano d'accesso*. Il mediatore utilizza una "mappa" (generata nella costruzione dello schema globale) che definisce l'associazione tra le classi globali e le classi locali. La query globale è espressa in termini degli schemi locali, tenendo in considerazione anche l'eventuale conoscenza di regole inter-schema definite sull'estensioni delle classi locali.

Il Mediatore agisce sulla query, sfruttando la tecnica di ottimizzazione semantica da ODB-Tools, in modo da ridurre il costo del piano d'accesso, e, dopo aver ottenuto la query ottimizzata, genera l'insieme di sotto-query relative alle sorgenti coinvolte.

1.3.3 Il linguaggio ODL_{I^3}

Il linguaggio ODL (Object Definition Language) per la specifica di schemi ad oggetti proposto dal gruppo di standardizzazione ODGM-93 [**13**] è universalmente riconosciuto come standard. Le sue caratteristiche peculiari, al pari di altri linguaggi basati i sul paradigma ad oggetti, possono essere così riassunte:

- Definizioni di tipi-classe e tipi valore;
- Definizione fra intenzione ed estensione di una classe di oggetti.
- Definizione di attributi semplici e complessi.
- Definizione di attributi atomici e collezioni.
- Definizione di relazioni binarie con relazioni inverse.
- Dichiarazione delle segnature dei metodi.

Con l'estensione di ODL al linguaggio ODL_{I^3} sono stati raggiunti i seguenti obiettivi:

- Per ogni classe il wrapper, può indicare nome e tipo di sorgente di appartenenza.
- Per le classi appartenenti alle sorgenti relazionali, è possibile definire le chiavi candidate ed eventuali *foreign key.*
- Attraverso l'uso del costrutto "*union*" ogni classe può avere più strutture alternative, mentre il costrutto "*optional*" consente di indicare la natura opzionale di un attributo. Queste caratteristiche, sono in accordo con la strategia utilizzata per la descrizione di

dati semi-strutturati.

- Il linguaggio supporta la definizione di grandezze locali e di grandezze globali.
- Il linguaggio supporta la dichiarazione di regole di *mapping* fra grandezze locali e di grandezze globali.
- È data la possibilità di definire regole di integrità sia sugli schemi locali che globali;
- Il linguaggio supporta la definizione di relazioni terminologiche di sinonimia, ipernimia, iponimia, e associazione;
- Il linguaggio può essere automaticamente tradotto nella logica descrittiva OLCD usata da ODB-Tools, e quindi utilizzarne le capacità nei controlli di consistenza e nell'ottimizzazione semantica delle interrogazioni.

1.4 Disambiguazione

La disambiguazione del testo è un processo di assegnamento del significato ad un termine definito *target*. Nella maggior parte dei casi il problema si riduce essenzialmente nel selezionare il senso corretto tra un insieme dei sensi possibili estratti da un dizionario o da un database lessicale come è appunto WordNet.
Va detto per inciso che esiste un'ampia letteratura sugli algoritmi di disambiguazione che non saranno qui trattati se non quelli attinenti agli argomenti che andremo ad affrontare.
L'utilità di WordNet all'interno di MOMIS si è detto essere quello di consentire di determinare le relazioni inter-schema al fine di scoprire le affinità tra le classi appartenenti a differenti sorgenti di dati.
Avendo a che fare con risorse differenti, è necessario tradurre tutti gli attributi e le classi di tali sorgenti, in linguaggio comune. Tale obiettivo si realizza associando ad ogni termine il suo senso corretto corrispondente in WordNet ovvero operando un processo di disambiguazione di tali termini; d'ora in poi ci si riferirà a questo processo di *mapping*, come alla *fase di annotazione delle sorgenti di dati.*
Le relazioni fra i termini estratte da WordNet, sono poi sommate al Common Thesaurus.

1.5 Osservazioni

Va detto che WordNet si è dimostrata essere la conoscenza di base preferita per un gran numero di metodologie di disambiguazione. Tuttavia il suo utilizzo ne ha evidenziato alcune lacune, specialmente in quei casi in cui WordNet viene utilizzata come unica risorsa all'interno del processo di disambiguazione.

Essenzialmente tali lacune possono essere attribuite a vari aspetti di WordNet. Uno è rappresentato dall'elevato livello di granularità, nella distinzione fra i sensi dei termini. Per esempio, in WordNet esistono verbi ai quali vengono associati più di 40 synset differenti. E' facile immaginare, quindi, quanto complessa sia la disambiguazione di tali termini, specialmente in ambienti in cui il contesto informativo non fornisce sufficiente contributo per poter discernere fra tali sensi.

Un altro limite di WordNet risiede nella possibilità di individuare relazioni solo fra termini appartenenti alla stessa gerarchia sintattica. Ovvero la gerarchia di WordNet non da alcuna informazione riguardo ai legami tra synset appartenenti per esempio ad un verbo ed altri appartenenti ad un nome.

In realtà quest'ultima affermazione non è del tutto esatta in quanto dalla versione 2.0 in WordNet ha introdotto alcune relazioni di dominio tra synset e lemmi di categorie differenti che è tuttavia abbastanza scomodo da usare.

Inoltre tali relazioni, anche quando sussistono tra istanze di termini appartenenti alla medesima categoria sintattica, risultano spesso insufficienti per delineare in maniera completa un processo di disambiguazione basato, appunto, sulle relazioni fra i termini.

Capitolo 2 WordNet e MOMIS

Sviluppato presso l'università di Princeton [19] sotto la direzione del professore Gorge A.Miller il database lessicale WordNet [**1**] è disponibile gratuitamente presso il sito http://www.cogsci.princeton.edu/wn, la licenza d'uso consente l'utilizzo gratuito anche a fini commerciali ed al di fuori della ricerca, a condizione che siano citati gli autori ed il sito ufficiale del progetto.

WordNet è un sistema lessicale di riferimento il cui disegno è ispirato alle teorie psico-linguistiche contemporanee, sulla memoria lessicale umana. I termini infatti, sono organizzati tenendo conto delle loro affinità di significato. WordNet comprende quattro categorie sintattiche: nomi, verbi, aggettivi ed avverbi. Ogni categoria è composta da diversi insiemi di sinonimi; ad ognuno di questi insiemi è associato un unico significato, condiviso da tutti i termini ad esso associati.

Ovviamente un termine, può possedere più di un significato ed essere, quindi, presente in molti di questi insiemi, ed anche in più di una categoria sintattica.

Nella terminologia utilizzato all'interno del progetto WordNet, un insieme di vocaboli che condivide il medesimo significato, prende il nome di *synset*. Un'altro elemento rilevante, che contraddistingue WordNet da un semplice dizionario di vocaboli, è la presenza di relazioni fra i synset. Vi sono diverse tipologie di relazioni che possono collegare due synset, come, ad esempio, l'iperonimia e l'iponimia, tramite cui si è in grado di creare, all'interno dell'intera categoria sintattica, gerarchie di significato.

2.1 La terminologia di WordNet

Vediamo ora più dettagliatamente la terminologia usata per indicare gli elementi caratteristici di WordNet. Tali termini verranno utilizzati in questo capitolo, che tratta di WordNet e della sua struttura, e nei capitoli successivi.

- ***Categoria sintattica:*** sono le grandi categorie in cui sono suddivisi i termini (ed anche

i file in cui sono contenuti) di WordNet. Le categorie sintattiche trattate sono quattro: nomi, verbi, avverbi ed aggettivi.

- ***Lemma:*** è la parola/termine a cui vengono associati uno o più significati. A volte un lemma può essere costituito da due o più parole, ed, in tal caso, i singoli termini (detti composti) sono uniti dal carattere *underscore* (_).
- ***Synset:*** un synset rappresenta un significato che viene associato ad un insieme di lemmi appartenenti alla stessa categoria sintattica. In pratica risulta corretto affermare, che ad un synset corrispondono più lemmi. Un synset, infatti, può essere rappresentato, oltre che dalla sua glossa, anche dall'insieme dei suoi lemmi.
- ***Glossa:*** rappresenta la descrizione a parole di un significa specifico; ogni synset, oltre a contenere un insieme di sinonimi, possiede anche una glossa.
- ***Relazione Semantica:*** si tratta di una relazione di WordNet, che lega due synset appartenenti alla stessa categoria sintattica; i diversi tipi di relazioni semantiche verranno trattate di seguito.
- ***Relazione lessicale:*** è una relazione che collega due lemmi appartenenti a due synset distinti (ma sempre appartenenti alla stessa categoria sintattica); i diversi tipi di relazioni lessicali verranno trattate di seguito.

2.1.1 La matrice lessicale

Alla base del progetto del *WordNet* c'è la Matrice Lessicale. La necessità di creare una matrice lessicale nasce dall'osservazione che una parola è un'associazione convenzionale tra il suo significato e il modo in cui viene letta o scritta. Per ridurre l'ambiguità derivata dal termine parola useremo:

- "*word meaning*" o *significato*, se ci riferiamo al concetto lessicale o significato
- "*word form*" o *lemma o termine*, se ci riferiamo al modo in cui viene letta o scritta.

Quest'associazione è di molti a molti, e dà luogo alle seguenti proprietà:

- ***Sinonimia:*** proprietà per cui lo stesso significato è esprimibile, tramite l'uso di due o più parole distinte.
- ***Polisemia:*** proprietà per cui ad una stessa parola, sono associati due o più significati distinti. Tale parole sono ambigue dal punto di vista del significato, e vengono dette *polisemiche*, per distinguerle da quelle che viceversa possiedono un solo significato (quindi non ambigue), e vengono dette *monosemiche.*

	W_1	W_2	W_3	W_4	W_5
M_1	$E_{1,1}$				
M_2		$E_{2,2}$			
M_3		$E_{3,2}$	$E_{3,3}$		
M_4					
M_5				$E_{5,4}$	
M_6					$E_{6,6}$

Figura 2.1: La matrice lessicale

Le relazioni fra forma della parola e significato, possono trovare rappresentazione in quella che viene chiamata, *matrice lessicale.* In tale matrice, le righe rappresentano i significati che è possibile attribuire ad una parola, mentre le colonne, rappresentano i diversi lemmi. In pratica, volendo leggere la matrice lessicale tramite la terminologia di WordNet, ad ogni riga è associato un synset, e ad ogni colonna una lemma. Ogni elemento non nullo che comparare all'interno della matrice, implica che il particolare lemma o termine, situato in quella riga, può essere usato per rappresentare lo specifico significato associato a quella colonna. Se all'interno di una colonna sono contenuti più elementi, si ha un caso di polisemia (il termine associato alla colonna può essere

utilizzato per esprimere più di un concetto); se, al contrario, due o più elementi compaiono sulla stessa riga, si è in presenza di una caso di sinonimia (il significato o synset di tale colonna, può essere espresso tramite più parole distinte).

Il concetto di matrice lessicale, viene espresso nel database di WordNet, tramite la separazione fra lemmi e synset (mantenendo cioè separati, termini e significati).

Un synset viene espresso nei file usati in WordNet, tramite l'insieme dei termini che sono ad esso associati. Tuttavia, nella maggioranza dei casi, un insieme di parole di questo tipo, non è sufficiente a descrivere un significato (si pensi al caso in cui si stia trattando un significato particolare che può essere descritto da una sola parola), così viene associata a ciascun synset, anche una descrizione del significato tramite la glossa.

2.2 Le relazioni

WordNet presenta al suo interno, due grandi gruppi di relazioni che si differenziano seconda del tipo di operatori a cui sono applicate. Si hanno così relazioni semantiche quando gli operandi sono synset, viceversa, si hanno relazioni lessicali nel caso in cui gli operandi siano lemmi. Fino alla versione WordNet 2.0 non esistevano relazioni tra un lemma ed un synset o fra operandi appartenenti a differenti categorie sintattiche (ad esempio fra un nome ed un verbo). All'interno dei file originali di WordNet tutte le relazioni (eccezion fatta per la relazione di sinonimia), sono rappresentate tramite puntatori e tramite caratteri speciali, che indicano il tipo di relazione specificata. Nei prossimi paragrafi saranno descritte le principali relazioni semantiche e lessicali presenti in WordNet.

2.2.1 Relazioni semantiche

Le relazioni di tipo semantico, coinvolgono sempre due concetti, due significati (due synset), non semplicemente due lemmi.

Iponimia e ipernimia

Le relazioni di *iponimia* e *ipernimia* (che rappresenta la relazione inversa), possono essere considerate l'equivalente delle gerarchie di specializzazione/generalizzazione per database relazionali o per l'ereditarietà dei modelli ad oggetti. Una relazione semantica di questo tipo, è valida solamente per le categorie sintattiche dei nomi e dei verbi (ma per i verbi si parla di toponimia). Una relazione di iponimia, lega un concetto (nel nostro caso un synset) ad uno più generale, quello che può essere ritenuto una sua generalizzazione. Trattandosi di un database di lingua inglese è lecito dire che, un synset *X* è un iponimo di un synset *Y,* se è corretta l'affermazione "*X is a kind of Y*". Per quanto riguarda la relazione opposta, quella di ipernimia, essa lega un concetto ad uno più particolare, più specializzato. In pratica si può affermare che un synset *X* rappresenta un ipernimo di un synset *Y,* se *Y* presenta tutte le caratteristiche di *X* più, almeno, una sua caratteristica particolare ed aggiuntiva. Le relazioni di iponimia ed ipernimia, sono le relazioni più numerose presenti all'interno del database

lessicale, di WordNet. Un semplice esempio, per comprendere meglio queste importanti relazioni, potrebbe essere: ABETE è in iponimo di ALBERO, ALBERO è a sua volta un iponimo di VEGETALE. Sono, altresì, verificate le razioni opposte: VEGETALE è un ipernimo di ALBERO, ALBERO è un ipernimo di ABETE. La relazione di iponimia (assieme a quella di ipernimia), può essere utilizzata per formare una gerarchia di specializzazione fra i synset di WordNet.

Meronimia e olonimia

Anche la relazione di *meronimia* lega fra loro due concetti, o synset, e anche in questo caso si è in presenza di una relazione inversa indicata come *olonimia* . Un concetto *X* è detto meronimo di un concetto*Y,* se è lecito per una madrelingua inglese, pronunciare la frase "*X is a part of Y*". Anche la relazione di meronimia, come quella di iponimia, può essere sfruttata per costruire una gerarchia sui synset di WordNet, in cui uno risulta essere una parte dell'altro. Le relazioni di meronimia e olonimia, vengono formulate sulla categoria sintattica dei nomi. Un esempio potrebbe essere rappresentato dai concetti MURA e FONDAMENTA come meronimi di COSTRUZIONE.

Implicazione

La relazione di *implicazione* è posta fra due verbi. Tale relazione può essere ritenuta simile a quella di meronimia posta sui nomi. Questa relazione è verificata se è vera la seguente proposizione: un verbo *X* implica un verbo *Y* se *X* non può verificarsi a meno che non si sia verificato (o non si stia verificando) Y. L'implicazione non è solamente una relazione semantica, ma è possibile avere anche implicazioni lessicali fra verbi (fra singoli termini). Per comprende meglio questo concetto si consideri i verbi DORMIRE e SOGNARE: in base alla definizione precedentemente, SOGNARE risulta implicare DORMIRE (*to dream entails to sleep*), infatti non è possibile sognare senza dormire. L'implicazione lessicale è una relazione univoca: se un verbo *X* implica un verbo *Y* non può essere vero anche il contrario.

Relazione causale

La relazione causale è simile alla relazione di implicazione ma senza inclusione temporale. Un esempio potrebbe essere FORZARE che implica AGIRE.

Raggruppamento di verbi

Questa relazione viene utilizzata per produrre raggruppamenti nella categoria sintattica dei verbi. In un gruppo formato in tale maniera, i synset hanno tutti un significato semantico molto simile. Un esempio di raggruppamento di verbi è dato da: *mistake, confuse, counfound, confuse, mix_up, confuse, blur, oscure.*

Similarità

La relazione di *similarità* è utilizzata solamente nell'ambito della categoria sintattica riguardante gli aggettivi. Molti synset di questa categoria sono raggruppati in coppie legate da una relazione di antinomia (si pensi, ad esempio, a synset trattanti i concetti di PESANTE e LEGGERO, in netta contrapposizione semantica fra di loro); tali synset, vengono chiamati synset principali (o *head synset*). A questi synset principali sono collegati per similarità, dei synset satelliti, che condividono indirettamente le relazioni di antinomia insieme al significato principale a cui sono legati. Ricapitolando, un aggettivo descrittivo, può avere una relazione di antinomia diretta (si tratta quindi di un synset principale), oppure una indiretta tramite l'ausilio di una relazione di similarità (synset satellite).

Attributo

La relazione di attributo rappresenta il legame che intercorre fra un aggettivo ed un nome di cui esprime il valore. Gli aggettivi in grado di descrivere il valore di un attributo, sono gli aggettivi descrittivi. Per fare un esempio basta pensare ad una frase come: *questa persona è alta.* L'aggettivo descrittivo alta, indica il valore dell'attributo *altezza* riferito a persona. Aggettivi quali *alta* o e *basso,* sono, quindi, legati al nome altezza. WordNet contiene puntatori fra gli aggettivi descrittivi ed i synset, appartenenti alla categoria sintattica dei nomi, che rappresentano gli attributi con cui conferiscono il valore.

Coordinazione

La *coordinazione* non è un tipo di relazione base, ma si potrebbe definire derivata. Due synset

sono detti coordinati se possiedono lo stesso ipernimo, se, cioè, risultano essere la specializzazione del medesimo concetto.

2.2.2 Relazioni lessicali

Le relazioni lessicali, diversamente da quelle semantiche, coinvolgono sempre due lemmi non due synset.

Sinonimia

La sinonimia, anche se rappresenta una relazione lessicale, non è espressa formalmente come le altre relazioni di WordNet: non esiste alcun puntatore che colleghi un termine al suo sinonimo. La relazione è espressa, invece, tramite l'appartenenza, da parte dei due vocaboli sinonimi, allo stesso synset.

Due possibili definizioni di sinonimia sono :

- Due termini sono sinonimi se la sostituzione di uno per l'altro non cambia mai il valore della frase in cui è fatta la sostituzione. (Leibniz)

- Due termini sono sinonimi, all'interno di un contesto linguistico C, se la sostituzione di un termine con l'altro, all'interno di C, non varia il valore della frase (definizione relativa ad un contesto).

La seconda definizione è decisamente più permissiva rispetto alla prima: esistono pochi termini considerati sinonimi nel senso descritto da Leibniz. Infatti, è estremamente difficile trovare due parole da poter interscambiare in ogni genere di contesto. Il database lessicale di WordNet, comunque, adotta, per stabilire la relazione di sinonimia, la seconda definizione: due lemmi sono sinonimi solo all'interno di uno stesso contesto, e di un certo synset. Anche tramite la seconda definizione di sinonimia, appare chiaro che due termini appartenenti a categorie sintattiche differenti, non potranno in nessun caso essere sinonimi. Proprio per questa ragione WordNet è stato diviso nelle categorie sintattiche di nomi, avverbi, verbi e aggettivi.

Antinomia

L*'antinomia* è una relazione lessicale fra due lemmi. Due termini legati da una relazione di antinomia sono l'uno il contrario dell'altro. Non è sempre corretta, comunque, l'affermazione che non X è antonimo di X. Si pensi, ad esempio, ai termini *ricco* e *povero*: se un individuo non è ricco, non è necessariamente detto che sia povero. Non è corretto considerare l'antinomia come una relazione semantica, quindi fra synset; per esempio i synset {*rise*, *ascend}* e {*fall, descend*}, pur essendo concettualmente opposti, non rappresentano degli antinomi. Una relazione di antinomia, invece, è presente fra i termini *rise* e *fall*, e *descend* e *ascend*.

Relazione di pertinenza

La relazione di *pertinenza* concerne gli aggettivi relazionali. Un aggettivo relazionale, svolge un ruolo che può essere riassunto in una espressione come: *associato con,* oppure *pertinente a* o semplicemente *di* in relazione ad un nome. L'aspetto di un aggettivo relazionale, risulta molto simile a quello del nome cui è legato, leggermente modificato. Si pensi all'espressione *accuratezza mentale*, l'aggettivo relazionale *mentale*, è associato al nome *mente*, tramite una relazione, appunto, di pertinenza.

Vedi anche

La relazione detta *vedi anche*, è una relazione lessicale e lega singoli lemmi di synset differenti. I motivi di tale relazione possono essere molto differenti fra loro.

Relazione partecipiale

Questa relazione lega fra loro gli avverbi o gli aggettivi, detti participiali, rispettivamente ai nomi o ai verbi da cui derivano. Come esempio si può pensare all'aggettivo *bruciato,* derivante dal verbo *bruciare* (all'interno di WordNet esiste quindi una relazione partecipiale fra i lemmi *burned* e *burn,* appartenenti alle categorie sintattiche, rispettivamente di aggettivi e verbi).

Derivato da

Alcuni aggettivi derivano da antichi nomi Greci o Latini. Questa affermazione risulta essere

vera, sia per la lingua italiana, che per quella inglese (idioma su cui è costruito WordNet). L'aggettivo relazionale *verbale*, deriva dal nome neutro latino *verbum ,* mentre *lessicale* deriva dal corrispondente nome greco. La relazione *derivato da* lega gli aggettivi ai nomi stranieri da cui derivano.

2.3 WordNet e MOMIS

In MOMIS si adotta, come risorsa lessicale e semantica d'informazione, il database WordNet come già detto. La scelta di utilizzare WordNet come risorsa lessicale, va ricercata nel fatto che quest'ultimo è un database lessicale molto conosciuto, tra i più completi e professionali, ed è liberamente disponibile.
Per poter essere utilizzato all'interno del sistema MOMIS, è stato analizzato, e il suo contenuto informativo è stato riportato all'interno di un database relazionale indicato con *momiswn*. Tale database contiene al suo interno un insieme di tabelle che consento di accedere velocemente alle informazioni di WordNet.
Tali tabelle contengono le seguenti informazioni:

- *wn_synset*: contiene essenzialmente le glosse dei vari synset; tali glosse sono associate ai rispettivi synset attraverso i campi *byte_offset* e *syntactic_category* che consentono di identificare un determinato synset in base alla notazione utilizzata in WN.
- *wn_relationship_type:* contiene i tipi di relazioni previsti nella versione di WN.
- *wn_relationship:* contiene tutte le relazioni fra i synset di WN.
- *wn_lemma_synset:* associa ciascun lemma, ai relativi possibili synset.
- *wn_lemma:* contiene tutti i lemmi presenti in WN e per ciascuno indica la categoria sintattica di appartenenza.

Rimandiamo per una descrizione più approfondita della struttura di WordNet in *momiswn* al capitolo 4.

L’utilità di WordNet all'interno di MOMIS si è detto essere quello di consentire di determinare le relazioni inter-schema al fine di scoprire le affinità tra le classi appartenenti a differenti sorgenti di dati.

Inizialmente la fase di annotazione dello schema, veniva eseguita dal *designer* (utente che interagisce con l’interfaccia MOMIS-WordNet) che essenzialmente, doveva scegliere manualmente il senso corretto di WordNet per ogni elemento dello schema. Nel far ciò, si dovrà, ovviamente, considerare il particolare contesto dettato dagli elementi dello schema da integrare.
La fase di scelta di uno o più significati da attribuire ad ogni termine, viene eseguita in due passi:

- *Scelta del termine*: durante questa prima fase il *WordNet morphologic processor* viene in aiuto al designer realizzando lo *stem* dei termini originali e ottenendo la *word form*. Successivamente la *word form* viene automaticamente cercata in WordNet. Nel caso in cui non sia presente, il designer può inserirla manualmente, senza però che modificare il database.
- *Scelta del significato*: il designer può scegliere di associare a ciascun elemento, zero, uno o più synset.

Una delle limitazioni principali di tale fase di annotazione, oltre ad essere completamente manuale, era quella di non consentire al designer di modificare il database inserendo nuovi lemmi, synset o relazioni.
Nel 2002 Veronica Guidetti in [**15**], integra all’interno di WordNet il componente *WordNet Editor,* il quale essenzialmente rappresenta una GUI, che sfrutta una libreria Java e rende possibile l’estensione del database di WordNet all’interno di MOMIS. Con “estensione di WordNet”, in questo caso, ci riferisce alla possibilità e alla necessità di poter sfruttare nuovi concetti non presenti in WordNet. WordNet infatti, pur essendo popolare e ampiamente utilizzato, presenta alcune limitazioni . In generale, con il termine “estensione di WordNet”, si intenderà un processo di arricchimento e/o completamento, delle informazioni, sia semantiche che strutturali, contenute in WordNet.

Con WordNet Editor, si fornisce la possibilità al designer di poter estendere WordNet allo scopo di colmare le sue lacune. Attraverso tale editor, è possibile, infatti, inserire nuovi termini, concetti, o relazioni fra synset nuovi o già esistenti. Per consentire ciò, il database lessicale di MOMIS, è stato modificato aggiungendo la tabella WN_EXTENDER, la quale tiene traccia delle informazioni riguardanti le modifiche effettuate sul database originale.

Capitolo 3 WordNet Domains

Una risorsa indipendente, utilizzabile in svariate applicazioni umane o automatiche, è rappresentata dalle gerarchie di dominio (*domain hierachies*).
La nozione di dominio è connessa a termini come *argomento principale, soggetto, soggetto di dominio, categoria ecc...*, termini a volte usati in maniera intercambiabile, a volte differenziati per significato.
In questa tesi, per dominio (o *domain*), intendiamo individuare un'area della conoscenza che sia, in qualche modo, riconosciuta come unitaria.
Un dominio può essere caratterizzato dal nome di una disciplina, nell'ambito della quale una certa area di conoscenza si è sviluppata (es: Chimica), oppure dal nome di uno specifico oggetto che è caratteristico di quest'area di conoscenza (es: cibo).
Benché gli oggetti della conoscenza e le discipline che li studiano siano palesemente correlate, la relazione tra questi due modi di vedere i domini è talvolta poco chiara e ambigua, e può pertanto essere fonte di ambiguità anche per l'esatta definizione del dominio stesso.
Un'altra interessante dualità, quando si parla di domini, la constatiamo per il fatto che la conoscenza si manifesta sia attraverso le singole parole, sia attraverso i testi.
Così la nozione di dominio, può essere applicata sia nello studio delle parole, dove rappresenta l'area di conoscenza alla quale un certo elemento lessicale appartiene, sia nello studio di interi brani scritti, dove rappresenta il 'filo conduttore' o argomento principale o il tema principale, che dir si voglia.

3.1 Le gerarchie di dominio

Per loro natura i domini possono essere organizzati in gerarchie basate su relazioni di specificità o particolarizzazione. Ad esempio possiamo certamente riconoscere che TENNIS è un dominio più specifico di SPORT o che ARTE è più generale di MUSICA .
Le gerarchie di dominio possono essere utilmente integrate con altre risorse linguistiche e profittevolmente usate nell'ambito dell'elaborazione del linguaggio naturale (NLP) come i Word Sense Disambiguation, Text Categorization,e Information Retrieval.

Come esempio di utilizzo delle gerarchie di dominio nel campo della lessicografia multilingua, citiamo l'ontologia Euro WordNet Domains, grazie alla quale i concetti interlinguistici (cioè comuni alle varie lingue) possono essere assegnati.

Grandi gerarchie di dominio sono inoltre disponibili in rete, principalmente allo scopo di classificare i documenti (ad esempio nei motori di ricerca di Google o Yahoo!).

Un'applicazione a larga scala di una gerarchia di dominio è rappresentata appunto, da WordNet Domains (Magnini e Cavaglia 2000).

WordNet Domains è una risorsa lessicale sviluppata da ITC-irst [0] (Istituto Trentino di Cultura- centro per la ricerca scientifica e tecnologica), nella quale ciascun synset di WordNet è annotato con una etichetta di dominio, selezionata da una gerarchia creata proprio per questo scopo.

Grazie alla sua caratteristica di indipendenza linguistica, la WordNet Domains Hierarchy (WDH) trova applicazione nell'ambito del framework Multi WordNet , un database lessicale multilingua (sviluppato sempre da ITC-irst), grazie al quale è possibile allineare WordNet in lingua italiana con WordNet in lingua inglese.

Un'altra importante applicazione è quella della creazione di corpus letterari, in cui i domini vengono utilizzati come criterio discriminante per la scelta dei testi da inserirvi.

3.1.1 La struttura

La prima versione di WDH era composta da 164 etichette di dominio, selezionate a partire dai codici di soggetto [illegible] dizionari correnti, e dai codici di soggetto contenuti nella classificazione decimale di Dewey (DDC- Decimal Dewey Calssification), la quale rappresenta un metodo di classificazione della conoscenza generale, ed è la tassonomia di gran lunga più usata per la catalogazione bibliografica [21].

I codici soggetto sono particolari codici che vengono assegnati alle categorie dell'ontologia e,nel caso della DDC, ad ogni elemento della tassonomia.

Le etichette di dominio (*domain labels*) erano organizzate in cinque alberi principali che avevano una profondità massima pari a 4.

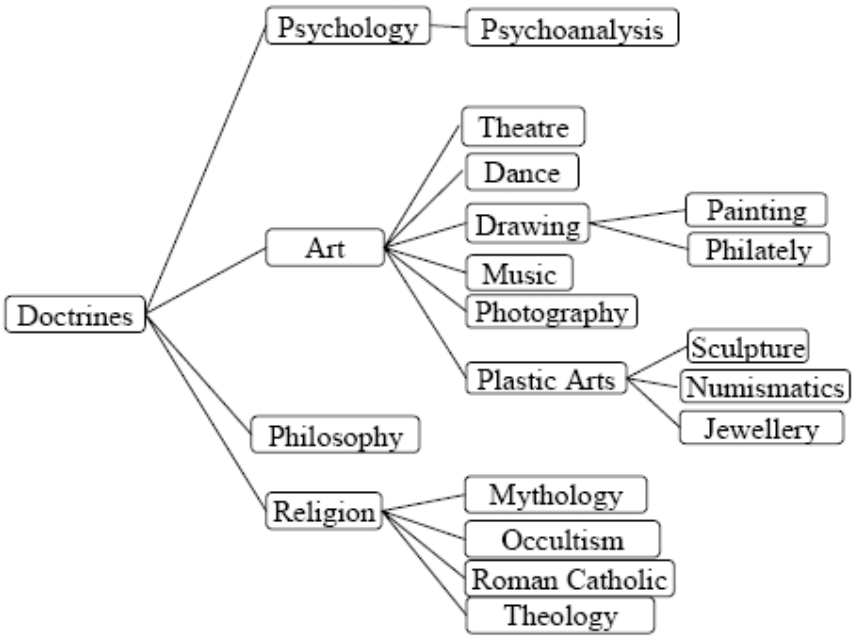


Figura 3.1: Frammento della gerarchia originale di WordNet Domains

Le etichette di dominio erano inizialmente concepite per essere orientate all'applicazione, cioè sono state integrate in WordNet con lo scopo preciso di permettere la categorizzazione dei sensi delle parole e fornire utili informazioni durante il processo di disambiguazione.

Il secondo livello di WDH è costituito dai domini di base, che include domini come ART, SPORT, RELIGION e HISTORY, mentre nel terzo livello si trova un grado di maggiore specializzazione dei domini (es. DRAWING, PAINTING, TENNIS, VOLLEYBALL e ARCHEOLOGY). Nell'ambito del NLP, l'insieme dei domini di base ha già mostrato di possedere un livello di astrazione e granularità adatti.

Benché la prima versione di WDH trovasse applicazione in svariati campi, essa presentava diversi problemi.

Prima di tutto le etichette di dominio non avevano una semantica definita. Il contenuto dei domini poteva venire suggerito dal significato lessicale delle loro etichette, ma non vi era alcuna esplicita indicazione di come dovessero essere interpretate.

Inoltre, non era molto chiaro se i domini di base possedessero o meno certi requisiti di copertura e bilanciamento della conoscenza.

In effetti, si era assunto per essi, che avessero un livello misurabile di granularità e, contemporaneamente, coprissero tutta la conoscenza umana. Tuttavia, non sempre tale assunzione veniva poi verificata..

Ad esempio, VETERINARY era posto allo stesso livello di ECONOMY, benché i due domini non possedessero lo stesso livello di granularità. Inoltre, non tutti i settori della conoscenza umana erano rappresentati (ad esempio, il dominio HOME).

Si è allora revisionata WDH, facendo ampio uso della DDC, in modo da poter eliminare i problemi appena descritti.

Occorre allora dare una veloce spiegazione di cosa sia e com'è organizzata la Dewey Decimal Classificazione.

3.1.2 Excursus: la Decimal Dewey Classification (DDC)

La DDC è una tassonomia ampiamente usata in ambito di classificazione bibliografica, in quanto fornisce un sistema logico di organizzazione di qualunque argomento di conoscenza, attraverso una ben definita gerarchia di codici di soggetto, la semantica di ciascun codice soggetto è determinata da un codice numerico, una breve descrizione associata ad esso, ed una relazione gerarchica con gli altri codici di soggetto.

La DDC non è usata solo per classificare insiemi di libri, ma anche per catalogare risorse internet, ed è stata concepita per adattarsi all'espansione della conoscenza umana.

Inoltre, essa è organizzata per discipline (o campi di studio) e ciò comporta che un soggetto può comparire in più di una disciplina, a seconda dell'aspetto del tema discusso.

La gerarchia della DDC, consente ad un argomento di essere definito come una parte di quello più generalizzato sopra di esso, e ciò determina il senso di una classe e la sua relazione con le altre.

A livello più generale, la DDC è formata da dieci classi principali (*main classes*) tra di loro mutuamente esclusive, che prese insieme coprono l'intera gamma delle conoscenze umane.

Ciascuna classe principale è suddivisa in dieci divisioni (dette *Hundred Divisions* o *Second Summary*) e ciascuna di esse in dieci sezioni (*Thousand Sections* o *Third Summary*).

Ciascuna categoria della DDC è rappresentata mediante un codice numerico come mostrato nell'esempio sottostante.

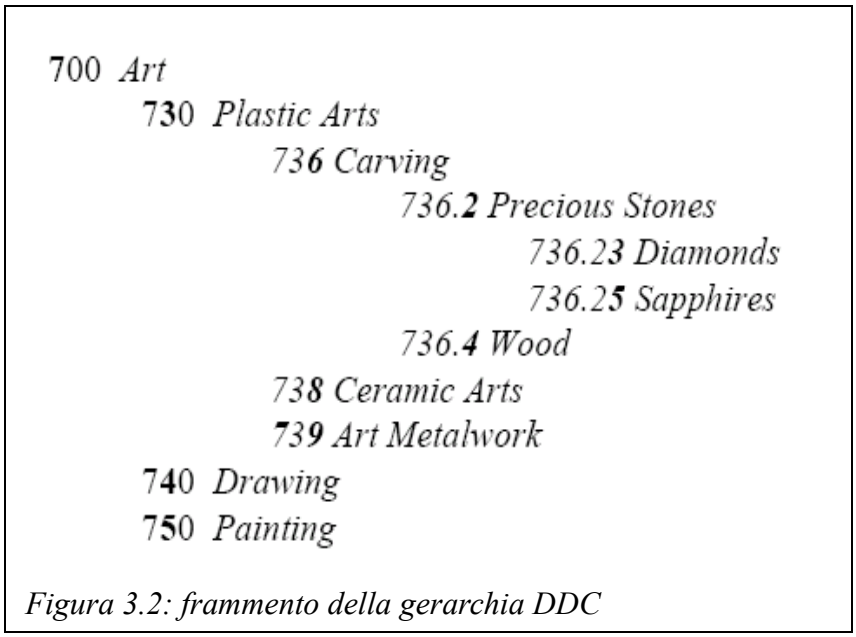


Figura 3.2: frammento della gerarchia DDC

La prima cifra del codice indica la classe principale (**7**00 è usato per la gerarchia che fa capo ad *Arts*), la seconda per le divisioni (7**3**0 per *Plastic arts,* 7**4**0 per *Drawing* ecc..) e la terza per le sezioni

La DDC può essere suddivisa ulteriormente in sottoclassi aggiungendo il punto decimale ed ulteriori cifre a seconda del livello di specificità richiesto.

3.1.3 DDC e WDH

La revisione di WDH utilizzando la DDC ha cercato di ottemperare a quattro requisiti fondamentali:

- Ciascuna etichetta di WDH deve avere una esplicita semantica ed essere identificata senza ambiguità.
- L'interpretazione di ciascuna etichetta WDH non deve sovrapporsi a quelle delle altre (disgiunzione).

- Tutto la conoscenza umana deve essere rappresentata dai basic Domains (copertura).

- I basic Domains devono avere un livello di granularità bilanciato.

Per dare ad un dominio una chiara semantica si è associato a ciascun dominio uno o più codici DDC.

Molti domini corrispondono uno ad uno ad un codice DDC; gli altri sono stati risistemati all'interno della gerarchia o sono stati rinominati in modo da conformarsi meglio a quella DDC. Ne risulta così una gerarchia di domini nella quale la quasi totalità delle etichette corrisponde un significato esplicitato nella codifica DDC.

Inoltre, dal momento che i codici DDC sono tra loro, semanticamente disgiunti, tale proprietà si riflette anche sui domini WDH; esiste un'eccezione per i domini ANTHROPOLOGY e SOCIOLOGY che nei domain sono distinti mentre la classificazione DDC non separa in maniera netta;in effetti WDH contiene due distinti domini ANTHROPOLOGY e SOCIOLOGY che tuttavia, si sovrappongono parzialmente dal momento che entrambi vengono assegnati i medesimi codici DDC tra 301:307 si veda [21].

La copertura della conoscenza si basa sull'assunzione che la codifica DDC in oltre un secolo di storia (fu per la prima volta formalizzata da Melvil Dewey nel 1876) abbia incluso in se tutta la conoscenza umana; nel nostro caso basta verificare che non ci siano codici di tale codifica che non siano associati ad un dominio WDH.

Ovviamente, ciò che si deve verificare è che i basic Domains coprano tutta la gamma di codici DDC, ed in effetti per ottenere ciò sono stati aggiunti a WDH alcuni domini nuovi rispetto alla prima versione (PARANORMAL, HOME, HEALTH che sono di base e FINANCE, GRAPHIC ARTS).

Un particolare molto importante da notare è quello dell'esistenza di un dominio FACTOTUM

che è generico e al quale sono associati tutti quei significati (parliamo di synset di WordNet) che non combaciano in maniera soddisfacente con nessun altro dominio della gerarchia.

Il requisito di bilanciamento è inteso ad assicurare che tutti i basic domain abbiano livelli di granularità simili.
La definizione di una misura di granularità per i domini è un problema non banale; gli approcci possono essere diversi, ad esempio si può considerare il numero di pubblicazioni per un dato dominio, oppure il numero di sottocodici nella DDC o la rilevanza sociale che un determinato argomento può avere. Posto che nessuno di questi approcci sia soddisfacente, alla fine si è deciso di considerare ancora una volta la classificazione DDC: ovvero si suppone che tutte quelle etichette che si trovano allo stesso livello gerarchico, cioè alla stessa profondità rispetto la radice dell'albero, abbiamo la stessa granularità (o almeno granularità molto vicine, comparabili),
Di conseguenza i domini che sono mappati nella DDC allo stesso livello, vengono considerati come aventi granularità affini.
La soluzione non è ottimale ma può essere considerata soddisfacente.

3.2 Disambiguazione con i domini

Per quel che riguarda ,il ruolo dell'informazione di dominio nell'ambito della disambiguazione del testo, una buona analisi viene fatta da Magnini e Strapparava in [16] . L'ipotesi è che le etichette di dominio (come medicina, sport ecc…), forniscano un potente strumento per stabilire delle relazioni semantiche tra i termini, le quali possono successivamente essere utilizzate per disambiguare il testo.

In particolare essi assumono che i domini costituiscono una fondamentale proprietà semantica sulla quale si basa la coerenza del testo;cioè i sensi delle parole che compaiono all'interno di una porzione di testo, tendono a massimizzare l'appartenenza ad uno stesso dominio.

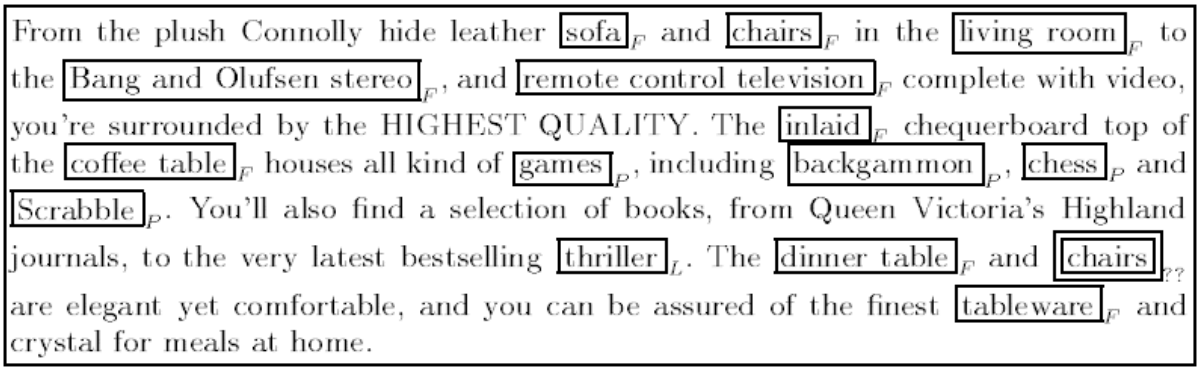


Figura 3.3: Informazioni di dominio in un testo campione

La figura 3.3, mostra un esempio estratto *dall'English lexical Sample task* durante il Senseval-2. Il termine target è la seconda occorrenza di "*chairs*". Si suppone che per la maggior parte delle parole all'interno dell'esempio, e per ogni senso ad esse associato, sia disponibile in WordNet un'etichetta di dominio, e che tali parole siano già disambiguate all'interno del testo. Si può notare come diverse parole come "*sofà", "living room", "dinner table"*, siano associate al dominio FURNITURE; altre poche parole come "*games", "chess"* e "*backgammon*" sono invece associate al dominio PLAY, mentre solo un termine è associato al domino LETTERATURA. Allo scopo di disambiguare il termine "*chair*", sembra naturale dover considerare che il dominio prevalente nel testo sia FORNITURE.
Ciò porta a disambiguare il termine con il senso più strettamente legato a tale dominio.

I WordNet Domains (WND), fornendo un'annotazione di dominio per ciascun synset di WordNet, rende possibile il meccanismo di disambiguazione precedentemente dedotto, nel quale si
ha bisogno di poter usufruire di una risorsa lessicale che sia in grado di associare, ad ogni senso di ogni termine, il dominio corrispondente. Inoltre, attraverso questa risorsa, è possibile effettuare alcune analisi del testo che consentono di individuare ad esempio, il dominio

prevalente di un testo o di una sua porzione, allo scopo di determinare come questo influisca sulla determinazione dei sensi delle parole in esso contenute.

Inoltre, è possibile calcolare una misura di coerenza del testo, sulle basi dell'ipotesi di *one domain of discourse*, in opposizione a quella di *one sense of discorse*. Un risultato rilevante di tale analisi, è che un numero piuttosto limitato di termini contribuiscono a determinare il dominio prevalente di una porzione di testo. Tali parole rappresentano i "centroidi" del processo di disambiguazione basato sull'etichette di dominio[18].

L'approccio descritto prevede, in sostanza

- Calcolo del dominio prevalente nel testo
- Confronto con i domini associati ai sensi del lemma L da disambiguare
- Scelta del senso che massimizza le similarità

3.2.1 Le utili proprietà dei domini

Un dominio può includere synset appartenenti a differenti categorie sintattiche: per esempio, il dominio MEDICINE raggruppa insieme sensi di nomi come *doctor#1* e *hospital#1*, e di verbi come *operate#7*.

Inoltre, un dominio può includere sensi appartenenti a differenti sottogerarchie di WordNet: per esempio, il dominio SPORT contiene sensi come *atlete#1* derivante da *life_form#1*, *game_equipment#1* derivante da *Physical_object#1, sport#1* derivante da *act#2,* e *playing_field#1* derivante da *location#1*.

Infine, i domini possono raggruppare sensi di una stessa parola all'interno di differenti *cluster* tematici, i quali hanno l'importante effetto di ridurre il livello di ambiguità quando si sta disambiguando attraverso il dominio.

Sense	Synset and Gloss	Domains	Semcor
#1	depository financial institution, bank, banking concern, banking company (a financial institution...)	ECONOMY	20
#2	bank (sloping land...)	GEOGRAPHY, GEOLOGY	14
#3	bank (a supply or stock held in reserve...)	ECONOMY	-
#4	bank, bank building (a building...)	ARCHITECTURE, ECONOMY	-
#5	bank (an arrangement of similar objects...)	FACTOTUM	1
#6	savings bank, coin bank, money box, bank (a container...)	ECONOMY	-
#7	bank (a long ridge or pile...)	GEOGRAPHY, GEOLOGY	2
#8	bank (the funds held by a gambling house...)	ECONOMY, PLAY	
#9	bank, cant, camber (a slope in the turn of a road...)	ARCHITECTURE	-
#10	bank (a flight maneuver...)	TRANSPORT	-

Figura 3.4: Synset associati al lemma 'bank'

La tabella mostra un esempio: la parola "*bank"* possiede dieci sensi differenti all'interno di WordNet, tre di questi possono essere raggruppati sotto il dominio di ECONOMY, mentre

altri due appartengono ai domini GEOGRAPHY e GEOLOGY.

È possibile individuare tre ruoli differenti che una parola può assumere all'interno di un testo (la stessa parola può giocare ruoli differenti a seconda del testo in cui appare):

- *Text Related Domain words* (TRD): rappresentano parole che possiedono almeno un senso che contribuisce a determinare il dominio dell'intero testo; per esempio, il termine *bank* all'interno di un testo riguardante l'ECONOMY, è probabilmente un TRD.
- *Text Unrelated Domain words* (TUD): rappresentano parole che possiedono sensi appartenenti a domini specifici, ma che non contribuiscono a determinare il dominio predominante del testo; per esempio, il termine "*church*" all'interno di un testo, riguardante l'ECONOMY probabilmente non riguarderà l'argomento principale del testo.
- *Text Unrelated Generic words* (TUG): rappresentano parole che non portano con se alcuna informazione di dominio; tali synset sono etichettati con *factotum* e rappresentano la maggior parte dei sensi annotati. Un esempio è rappresentato dal verbo *to be*.

Domain	#Syn	Domain	#Syn	Domain	#Syn
Factotum	36820	Biology	21281	Earth	4637
Psychology	3405	Architecture	3394	Medicine	3271
Economy	3039	Alimentation	2998	Administration	2975
Chemistry	2472	Transport	2443	Art	2365
Physics	2225	Sport	2105	Religion	2055
Linguistics	1771	Military	1491	Law	1340
History	1264	Industry	1103	Politics	1033
Play	1009	Anthropology	963	Fashion	937
Mathematics	861	Literature	822	Engineering	746
Sociology	679	Commerce	637	Pedagogy	612
Publishing	532	Tourism	511	Computer_Science	509
Telecommunication	493	Astronomy	477	Philosophy	381
Agriculture	334	Sexuality	272	Body_Care	185
Artisanship	149	Archaeology	141	Veterinary	92
Astrology	90				

Figura 3.5: Distribuzione dei synset WordNet tra i domini scelti della gerarchia DDC

Allo scopo di fornire una stima quantitativa della distribuzione delle tre tipologie differenti di termini, è stato effettuato un esperimento sul corpus SemCor, utilizzando WordNet Domains come sorgente di annotazioni di domini. In questo esperimento, Magnini e Strapparava, hanno considerato solo 42 domini disgiunti, escludendo *factotum* (per esempio, si è usato solamente il dominio SPORT al posto di domini come VOLLEYBALL, BASKETBALL ecc…).
Questo insieme, consente un buon livello di astrazione, senza una perdita rilevante d'informazione e, in più, evita il problema dell'applicazione di tecniche di apprendimento a domini non rappresentati abbastanza all'interno dei testi disponibili.

Per ogni testo di SemCor, è assegnato un punteggio a ciascuno di questi 42 domini, in base alla loro frequenza fra i sensi delle parole dei testi. I tre domini con punteggio più alto sono stati considerati come domini prevalenti all'interno del testo. Successivamente, ogni parola del testo è stata assegnata ad una delle tre tipologie di parole precedentemente descritte, in base al fatto che:

- Almeno un dominio tra quelli ai quali il termine è associato, coincida con uno

dei tre domini rilevanti (parole TRD-*Text Related Domain*).

- La maggior parte dei sensi del termine siano associati ad un dominio specifico ma nessuno di questi corrisponda ad uno dei tre domini individuati (parole TUD-*Text Unrelated Domain*).
- La maggior parte dei sensi associati ai termini, siano etichettati come *factotum* e nessuno dei sensi rimanenti appartenga ad uno dei tre domini principali (parole TUG-*Text Unrelated Generic*).

Ogni gruppo di parole, viene analizzato ulteriormente in base alla categoria sintattica di appartenenza, e viene calcolata la media dei termini polisemici in base a WordNet.
I risultati dell'esperimento sono riportati in figura 3.6:

Word class	Nouns	Verbs	Adjectives	Adverbs	All
TRD words	18732 (34.5%)	2416 (8.7%)	1982 (9.6%)	436 (3.7%)	21%
Polysemy	3.90	9.55	4.17	1.62	4.46
TUD words	13768 (25.3%)	2224 (8.1%)	815 (3.9%)	300 (2.5%)	15%
Polysemy	4.02	7.88	4.32	1.62	4.49
TUG words	21902 (40.2%)	22933 (83.2%)	17987 (86.5%)	11131 (93.8%)	64%
Polysemy	5.03	10.89	4.55	2.78	6.39

Figura 3.6: Tabella delle distribuzioni delle parole tra le tipologie individuate in Semcor

Si nota che solo il 21% dei termini contribuisce a determinare i domini prevalente (e tra questi circa l'80% sono i nomi). I termini TUG, come ci si aspettava, sono sia i più frequenti (circa il 64%), sia quelli più polisemici. A tale risultato contribuiscono principalmente i verbi con l'83%, i quali si caratterizzano per essere altamente polisemici e, quindi, non danno un contributo sostanziale alla determinazione dei domini.

L'ipotesi di O*ne Sense per Discorse* (OSD), spiega come nell'uso multiplo di un termine, si tenda ad avere sempre anche il medesimo senso, all'interno di un discorso ben scritto.
Seguendo lo stesso ragionamento, l' ipotesi il o*ne domain per discourse* (ODD), porta ad

assumere che, l'uso multiplo di un termine all'interno di una porzione coerente di testo, tende a mostrare lo stesso dominio.

Dimostrando l'assunzione di ODD, si rafforzerebbe l'ipotesi alla base dell'uso di WordNet Domain nell'ambito della disambiguazione del testo, ovvero che il dominio prevalente di un testo, rappresenta una caratteristica importante nella determinazione del senso corretto dei termini stessi.

Per dimostrare la validità di tale approccio, è stato eseguito un test, utilizzando sempre ovviamente WordNet Domains come sorgente di informazione di dominio. Secondo Krovetz in [], per invalidare l'ipotesi di OSD, è sufficiente che almeno un termine all'interno dello stesso testo, non rispetti tale assunzione. Seguendo questa osservazione, per effettuare il test, è stato estratto da SemCor un insieme di 23,877 parole ambigue con occorrenze multiple all'interno dello stesso documento, e si è contato il numero di termini annotati con sensi differenti.

Successivamente per ciascuno dei vari sensi dei termini così individuati si è determinato il dominio associato da WordNet Domains. La differenza tra OSD e ODD, si palesa quando si consideri ad esempio il termine *bank* il quale compare tre volte all'interno di un testo e con tre sensi differenti (*bank#1, bank#3, bank#8*). In questo caso viene dimostrata l'inconsistenza dell'ipotesi di OSD, ma rimane consistente quella di ODD poiché tutte le tre occorrenze del termine sono etichettate sotto lo stesso dominio, ECONOMY.

One Sense per Discourse vs. One Domain per Discourse

Pos	Cases [a]	Exceptions to OSD [b]	Exceptions to ODD [c]
All	23877	7469 (31%)	2466 (10%)
Nouns	10291	2403 (23%)	1142 (11%)
Verbs	6658	3154 (47%)	916 (13%)
Adjectives	4495	1100 (24%)	391 (9%)
Adverbs	2336	790 (34%)	12 (1%) [d]

Figura 3.7: One Sense per Discourse vs. One Domain per Discourse

I risultati del test sono riportati in tabella. Essi mostrano che l'ipotesi di ODD è verificata insieme a quella che all'interno di un testo esista solo un numero limitato di domini rilevanti.
Le poche eccezioni al ODD sono probabilmente dovute a variazioni di dominio all'interno dei lunghi testi di SemCor,(mediamente sono composti da 2000 termini per testo); ciò fa sì infatti, che alcune parole possono appartenere a domini differenti, in differenti porzioni dello stesso testo.
La figura 3.8, ottenuta dopo aver disambiguato i termini in base ai loro domini possibili, mostra come la rilevanza dei due domini, PEDAGOGY e SPORT, vari all'interno del medesimo testo.

Il concetto di dominio predominante, ha quindi senso se applicato all'interno di una porzione di testo, piuttosto che rispetto all'intero testo. Supponiamo, per esempio, di dover disambiguare il termine *acrobatics*. Considerando solo la porzione di testo rappresentata dai termini intorno alla parola target, *acrobatics* viene correttamente disambiguato assegnandogli il senso associato al dominio SPORT.

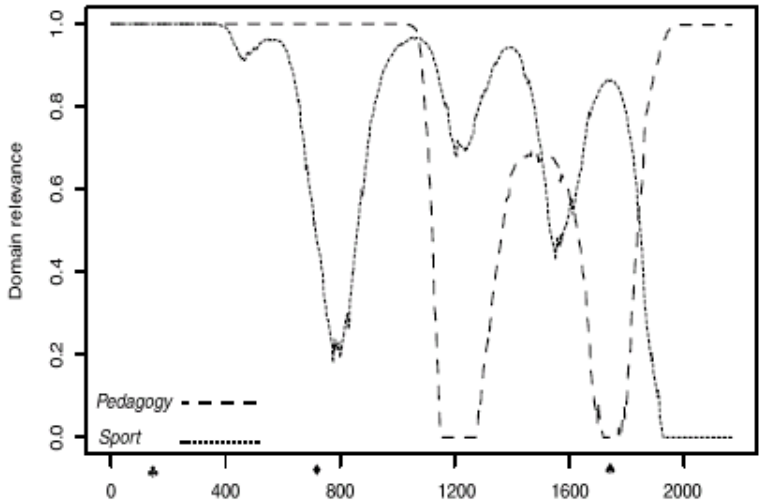


Figura 3.8: Variazioni di dominio all'interno del testo br-e24 del corpus SemCor

3.2.2 Domains Driver Disambiguation

Sulla base delle affermazioni precedenti Magnini e Strapparava in [16], propongono un algoritmo di disambiguazione del testo chiamato Domain Driver Disambiguation (DDD). L'idea di base è che il processo di disambiguazione di un termine all'interno del suo contesto, è principalmente un processo di comparazione tra il dominio del contesto e i domini associati ai sensi del termine.
L'algoritmo richiede, in input, porzioni di testo, e fornisce una struttura per integrare l'informazione di dominio acquisita tramite testi annotati.

Si è perciò individuata la struttura dati deputata a raccogliere le informazioni di dominio in un vettore chiamato vettore di dominio, e la sua lunghezza è pari al numero di domini considerati; tale approccio è stato sperimentato considerando solo, 42 domini e quindi un vettore egualmente lungo.
I vettori di dominio consentono di fondere e gestire con facilità le informazioni riguardanti i sensi dei termini, per porzioni di testo. In particolare si sono utilizzati due tipi di vettore:

- **Vettore di Testo**: rappresenta la rilevanza di una porzione di testo calcolata rispetto ad ogni dominio considerato.
- **Vettore di Sensi**: rappresenta la rilevanza di un senso di un dato termine, calcolata rispetto ad ogni dominio considerato.

Per disambiguare l'occorrenza di un termine *w* all'interno di una porzione di testo *T,* devono essere calcolati sia il vettore di testo per la porzione di testo *T* intorno al termine *w,* sia tutti i vettori per i sensi *s1, s2, s3… sk,* di *w* . Successivamente il sistema sceglierà il senso il cui

vettore, ha massima similarità con il vettore della porzione di testo *T. T* può essere rappresentato come una lista di coppie <*lemma, POS*> ottenute tramite un *tagger*. I lemmi sono indicizzati in base alla loro posizione nel testo. D'ora in poi la notazione *Tp,* sarà utilizzata per riferirsi ad una parola collocata in posizione *p* all'interno del testo *T.*

Magnini e Strapparava, rappresentano la rilevanza di un dominio rispetto ad un testo, con un numero reale positivo compreso fra [0, 1]. Dato un dominio, esso avrà rilevanza pari ad 1 per un testo, se rappresenta l'argomento principale,mentre avrà rilevanza 0, se non è relazionato in nessun modo al testo. Per esempio, il testo il cui argomento è "*September 11th attack on the Twin Towers*" può avere una rilevanza pari ad 1, rispetto ai domini POLITICS o MILITARY, e rilevanza 0 per domini come SPORT.

L'algoritmo di calcolo dei domini rilevanti per una data porzione di testo, intorno ad una parola in posizione Tp, prima di tutto identifica la sotto sequenza delle parole da *T(p-c)* a *T(p+c),* dove *2c* è la dimensione del contesto fornita all'algoritmo come parametro. Tale approccio è stato testato su SemCor. La sperimentazione ha dimostrato come le prestazioni dell'algoritmo diminuiscano quando il valore di *2c* supera 50.

Il secondo passo consiste nel raccogliere tutte le annotazioni di dominio corrispondenti ai vari synset dei termini, e calcolare la frequenza di ogni dominio, all'interno di questo insieme.

Tuttavia, per Strapparava e Magnini, la frequenza di un dominio in un testo non implica necessariamente la sua rilevanza all'interno del testo stesso. Per esempio, può accadere che POLITICS sia il dominio più frequente all'interno di un articolo giornalistico, perfino se in realtà l'argomento dell'articolo corrisponda al dominio VETERINARY perché le parole legate a VETERINARY, sono meno frequenti dei termini legati al dominio POLITICS. La loro ipotesi è che un dominio è rilevante per un testo, se la sua frequenza nel testo è significativamente più alta rispetto alla frequenza che tale dominio ha in testi con cui non è in relazione.

Per stimare la relazione fra frequenza e rilevanza, Magnini e Strapparava, assumono che all'interno di un generico corpus bilanciato, il numero di testi rilevanti per un certo dominio *D*, è distribuito egualmente. In fase di sperimentazione, ciò significa dover determinare la deviazione standard per ogni dominio di WordNet Domains all'interno del LOB Corpus [17], considerandolo come generico corpus bilanciato per l'Inglese. La rilevanza di dominio è valutata utilizzando teoremi riguardanti la distribuzione normale: se la frequenza di un

dominio *D*, calcolata nel testo *T*, è significativamente più alta della frequenza di *D* nel corpus (cioè eccede più del doppio la deviazione standard), allora *D* è rilevante rispetto a *T*.
Per esempio, supponiamo di voler valutare la rilevanza di ECONOMY nella frase " *Today I draw money from my bank*". L'algoritmo individuerà tutti i domini associati da WND ad ogni senso di ogni parola. Il nome *bank* ha 5 occorrenze associate al dominio ECONOMY su 10, il nome *money* ne ha 3 su 3 e il verbo *draw* ha un'occorrenza sola associata ad ECONOMY su un totale di 33. Da qui la frequenza totale di ECONOMY è 1.53. Supponiamo che la frequenza di ECONOMY nel corpus LOB sia di 0.2, e che la deviazione standard sia 0.1. Tale valore rappresenta la distribuzione della frequenza di ECONOMY in testi non correlati. Di conseguenza, ECONOMY non sarà da considerarsi come dominio prevalente,in testi in cui la sua frequenza è compresa nel *range* [0, 0.4]; viceversa sarà considerato rilevante, se avrà una frequenza significativamente più alta, come nel caso in esempio .

Dunque un vettore di testo è un vettore di dominio, estratto da una porzione di testo. Dato un insieme di domini *D=[D1, D2, ..., Dn],* un testo *T* e una posizione *p*, il vettore di testo *Tp* è il vettore n-dimensionale, il cui componente i rappresenta la rilevanza del dominio i-esimo per *T* alla posizione *p*.
Dato un contesto, intuitivamente, *Tp* rappresenta i domini rilevanti per un punto *p* del testo. I vettori di testo calcolati su differenti posizioni dello stesso testo, possono essere diversi, e per lo stesso testo, possono esistere più domini rilevanti.

Un vettore di senso, è un vettore di dominio ottenuto a partire da un synset. Esso fornisce due informazioni importanti: la sua lunghezza, rappresentante la frequenza di occorrenze dei sensi, e le sue direzioni, rappresentanti il vettore "significato" dei testi dove il senso appare generalmente.
La via più naturale per costruire i vettori dei sensi, è l'applicazione di tecniche supervisionate a dei dati di *training.* Tuttavia, in questi casi i dati di training non sono quasi mai disponibili.
Una via alternativa, consiste nell'ottenere i vettori dei sensi sfruttando le informazioni contenute in WordNet Domains. Questa possibilità, la quale è stata applicata durante gli esperimenti del Senseval-2, rende l'approccio di Magnini e Strapparava, molto flessibile.

Nel caso in cui siano disponibili i dati di *training*, il vettore dei sensi è costruito attraverso la somma dei vettori di testo, considerando la direzione del vettore significato dei testi all'interno del quale compaiono tipicamente i synset. Questo metodo è, inoltre, particolarmente efficace per sensi generici (cioè classificati come factotum), i quali in genere compaiono in vari tipi di testo e producono vettori senza una dimensione dominante. Tuttavia, dati dei testi generici, spesso questi presentano pochi domini prelevanti, ed è necessario un elevato numero di dati di *training,* per produrre vettori di senso generico.
Nel caso in cui non si abbiano a disposizione i dati di *training*, il vettore dei sensi deve essere costruito utilizzando WordNet Domains e SemCor. In questi casi, il vettore dei sensi ha un 1 nelle rispettive posizioni di appartenenza di dominio di WordNet Domains e 0 altrimenti. La sua lunghezza è proporzionale alla frequenza del senso in SemCor. Per esempio, il vettore di *bank#1* nell'esempio in figura 3.4, sarà ((ECONOMY 20)(ARCHITECTURE 0)…(SPORT 0)). Se il senso viene annotato con *factotum*, il suo vettore dei sensi ha la direzione di un vettore di 1 per ogni componente.
Il processo di disambiguazione di un termine Tp, consiste in un semplice confronto tra, il vettore di testo *Tp* ed i vettori di tutti i sensi di tale parola. Allo scopo di prendere in considerazione sia le direzioni (cioè il dominio) sia la lunghezza (ovvero la frequenza), dei vettori di senso, viene calcolato il prodotto vettoriale tra *Tp* e ogni vettore di senso. Il risultato è un lista ordinata di sensi di *Tp* e la selezione finale si basa sul confronto rispetto ad una soglia fissata.

Da qui derivano tre possibili output:

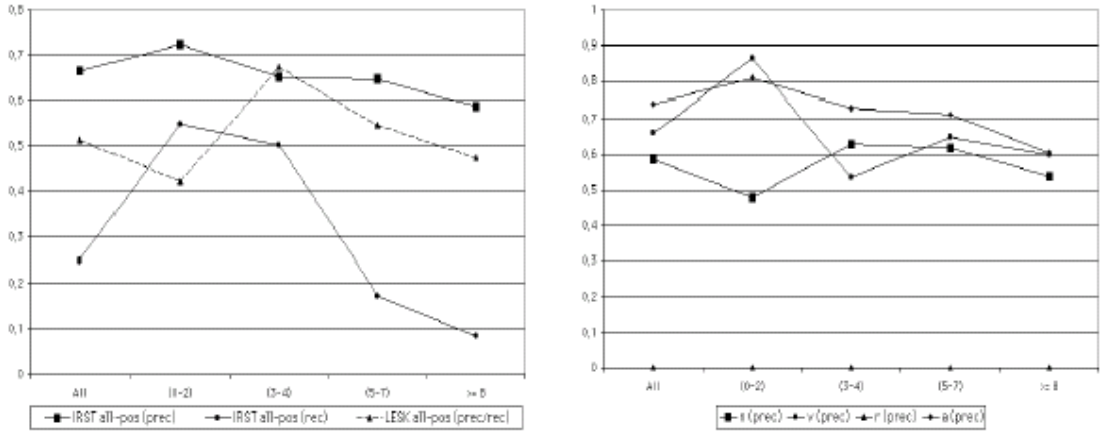
- Se il *match* di un senso eccede in maniera significativa la soglia fissata, il senso viene selezionato come più probabile.
- Se si raggiunge più di un buon *match* (ovvero si hanno due sensi appartenenti allo stesso dominio rilevante), la strategia prevede semplicemente di non assegnare alcun senso al termine, come se non vi fosse nessun altra informazione disponibile.

- Nel caso in cui non si individui alcun *match*, come può accadere per esempio, con termini altamente polisemici, non si seleziona alcun senso.

Supponiamo per esempio di voler disambiguare la parola *bank*, all'interno della frase "*Today* I *have draw money from my bank*" rispetto ai sensi *s1: bank#1* ed *s2*: *bank#2*. Tale situazione è riportata in figura 3.9, dove sia il vettore dei sensi sia il vettore del testo, sono rappresentati per una sotto insieme di domini. Il prodotto vettoriale tra *T8* ed *s1* da come risultato 1.7356, mentre quello tra *T8* ed *s2* da 0.06185. Di conseguenza si seleziona *bank#1*.

	SPORT	MEDICINE	ECONOMY	GEOGRAPHY
$\vec{s}_1$ $(Bank\#1)$	0.02	0.08	1.73	0.04
$\vec{s}_2$ $(Bank\#2)$	0.005	0.03	0.04	0.69
$\vec{T}_8$	0.2	0.005	1	0.03

Figura 3.9: Risultati tra bank#1 e bank#2



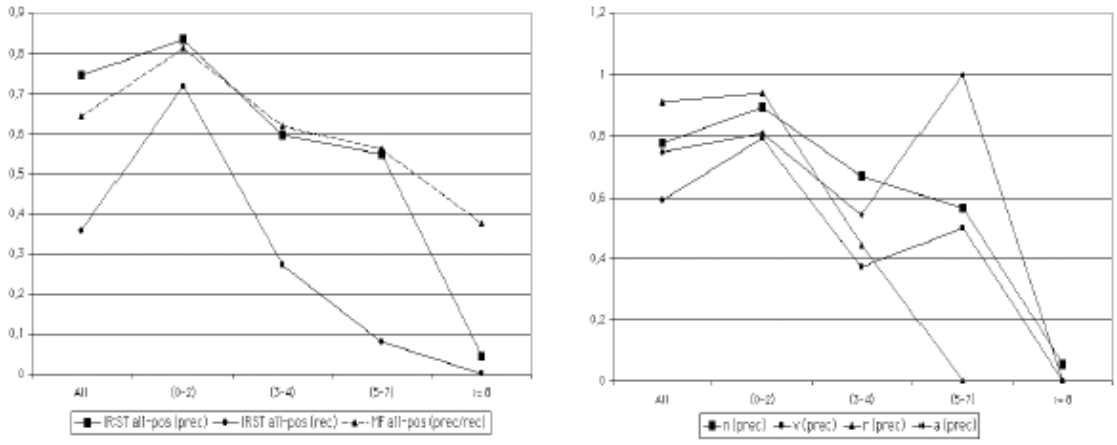


Figura 3.10: Prestazioni dell'algoritmo DDD

Nei grafici in figura 3.10 sono riportati i risultati ottenuti da tale metodo di disambiguazione, durante il Senseval-2. Tali risultati vanno interpretati considerando che, oltre all'informazione di dominio, non si è utilizzata nessun altra risorsa ne sintattica ne semantica. I risultati sono valutati in termini di Recall e Precision così determinate: P= annotazioni corrette/(annotazioni sbagliate + annotazioni corrette), R = (annotazioni corrette / (annotazioni sbagliate + annotazioni corrette + annotazioni non eseguite).

$$P = \frac{annotazioni\ corrette}{annotazioni\ sbagliate + annotazioni\ corrette}$$

$$R = \frac{annotazioni\ corrette}{annotazioni\ sbagliate + annotazioni\ corrette + annotazioni\ non\ eseguite}$$

Capitolo 4 Integrazione di Wordnet Domains e WordNet in Momis

Si è già detto dell'importanza di poter integrare WordNet Domains e WordNet . Il package di WordNet Domains, rilasciato liberamente sul sito di ITC-irst http://wndomains.itc.it/download.html è arrivato, al momento della stesura di questa tesi, alla versione 3.2 che è anche la versione da noi adottata per l'implementazione dell'utilità di integrazione: essa è stata rilasciata nel febbraio 2007 ed è basata sulla versione di WordNet 2.0.

In realtà la versione più recente di WordNet è la 2.1 per sistemi Windows e la 3.0 per Unix, Linux, Solaris ecc… ma WordNet Domains non è ancora aggiornata a queste versioni sicché dalla 3.0 in poi WordNet Domains ha compatibilità solo con WordNet 2.0 (vedi file *version-history.txt* del pacchetto *wn-domains-3.2*).

Occorre, per capire come procedere, rivedere brevemente com'è strutturato il package di WordNet e come viene poi implementato all'interno del database *momiswn* utilizzato in Momis per la disambiguazione dei termini.

4.1 Il database *momiswn*

WordNet è distribuito in modalità *freeware* e comprende, al suo interno, un'applicazione per eseguire interrogazioni e cercare i significati di parole in lingua inglese, file di documentazione e svariati file di testo che raccolgono il vero e proprio dizionario dell'ontologia lessicale.

In particolare per ogni categoria sintattica (cioè nomi, verbi, aggettivi, avverbi) esiste:

- un file indice (*idx*) che contiene su ciascuna delle sue righe, un lemma di tale categoria e una lista di puntatori *(byte offset*) a tutti i synset di cui tale lemma fa parte (più svariate altre informazioni quali il numero d'ordine che ha il lemma in ciascuno dei synset associati ed il numero di synset associati);

- un file data (*data*) nel quale ciascuna riga è associata ad un synset e contiene l'offset in byte di tale riga (e quindi identifica univocamente, per quella categoria il synset) e altre informazioni che riguardano le relazioni lessicali e semantiche che coinvolgono il synset o i suoi lemmi.

Quindi, da questi file si è in grado di estrarre tutto ciò che è relativo ai lemmi, synset, relazioni lessicali e semantiche.

Tutto questo contenuto informativo è riversato in un normale database relazionale che, nell'ambito del sistema Momis, viene chiamato come detto *momiswn* ed è utilizzato durante la fase di disambiguazione dei termini.

Lo schema E/R di *momiswn* è la seguente:

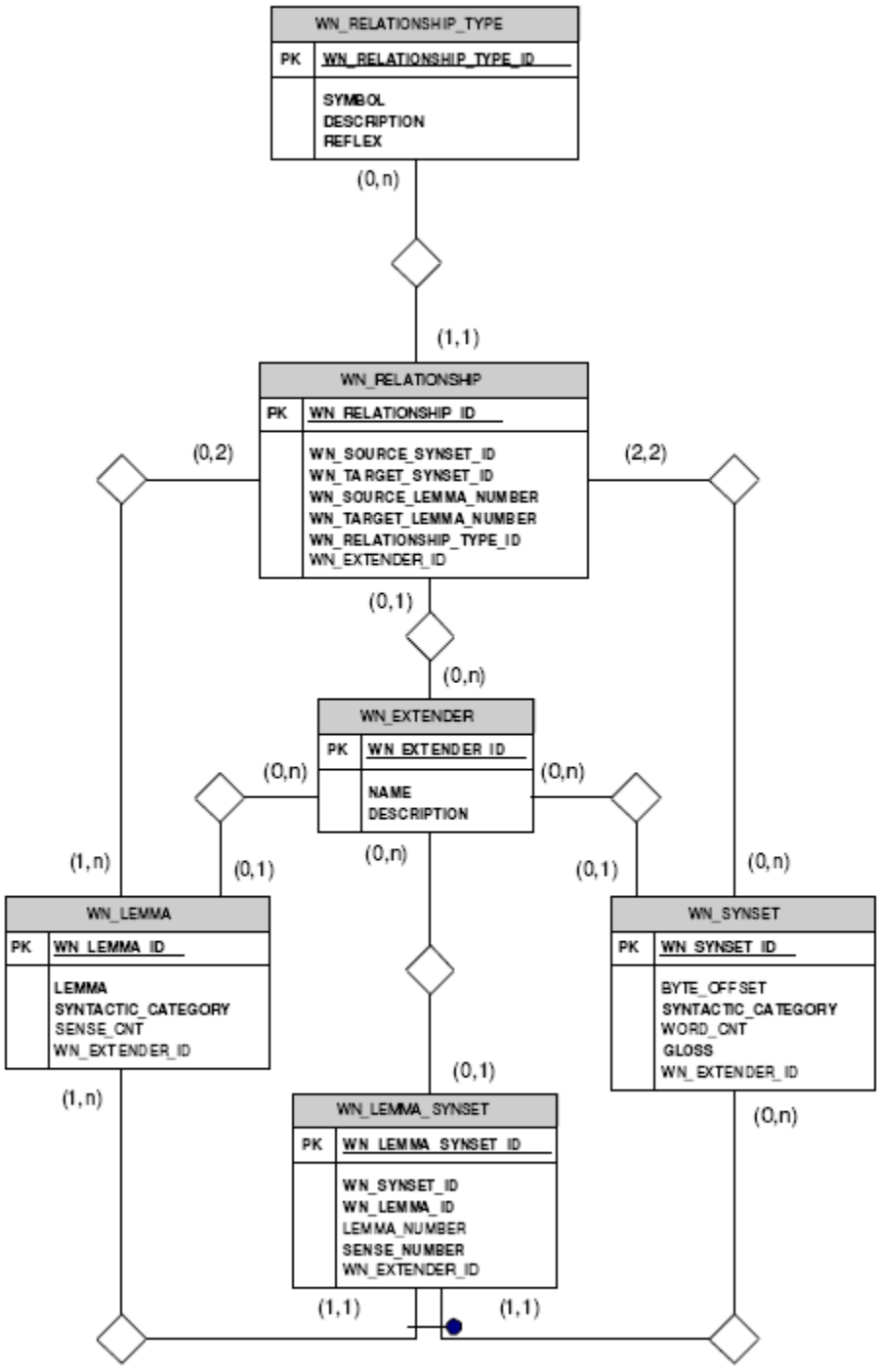


Figura 4.1: Schema E/R del DB "momiswn"

che nel modello relazionale si traduce nelle tabelle seguenti:

- *wn_synset* contiene tutti i synset di WordNet, ovverosia tutti i significati (o glosse) delle parole; è identificato da una chiave primaria auto incrementata ma anche dalla coppia (*byte_offset,categoria_sintattica*);
- *wn_lemma* contiene tutte le parole (cioè i lemmi);
- *wn_lemma_synset* contiene tutte le associazioni tra synset e lemmi;
- *wn_relationship* contiene tutte le relazioni semantiche (tra synset) e lessicali (tra lemmi di synset diversi) insieme all'indicazione del tipo di relazione tramite *foreign key* alla tabella successiva;
- *wn_relationship_type* contiene i tipi di relazione lessicali e semantiche riconosciuti ed usati nell'ontologia;
- *wn_extender* contiene le estensioni , ovvero un riferimento a tutto ciò che viene aggiunto a WordNet originale che è esso stessa riconosciuto come la prima delle estensioni (l'extender con codice 'WN' per la precisione)

ed una tabella chiamata *wn_reverse_index* utilizzata come supporto alle ricerche di glosse per similitudine, e che non è presa in considerazione per i nostri scopi.

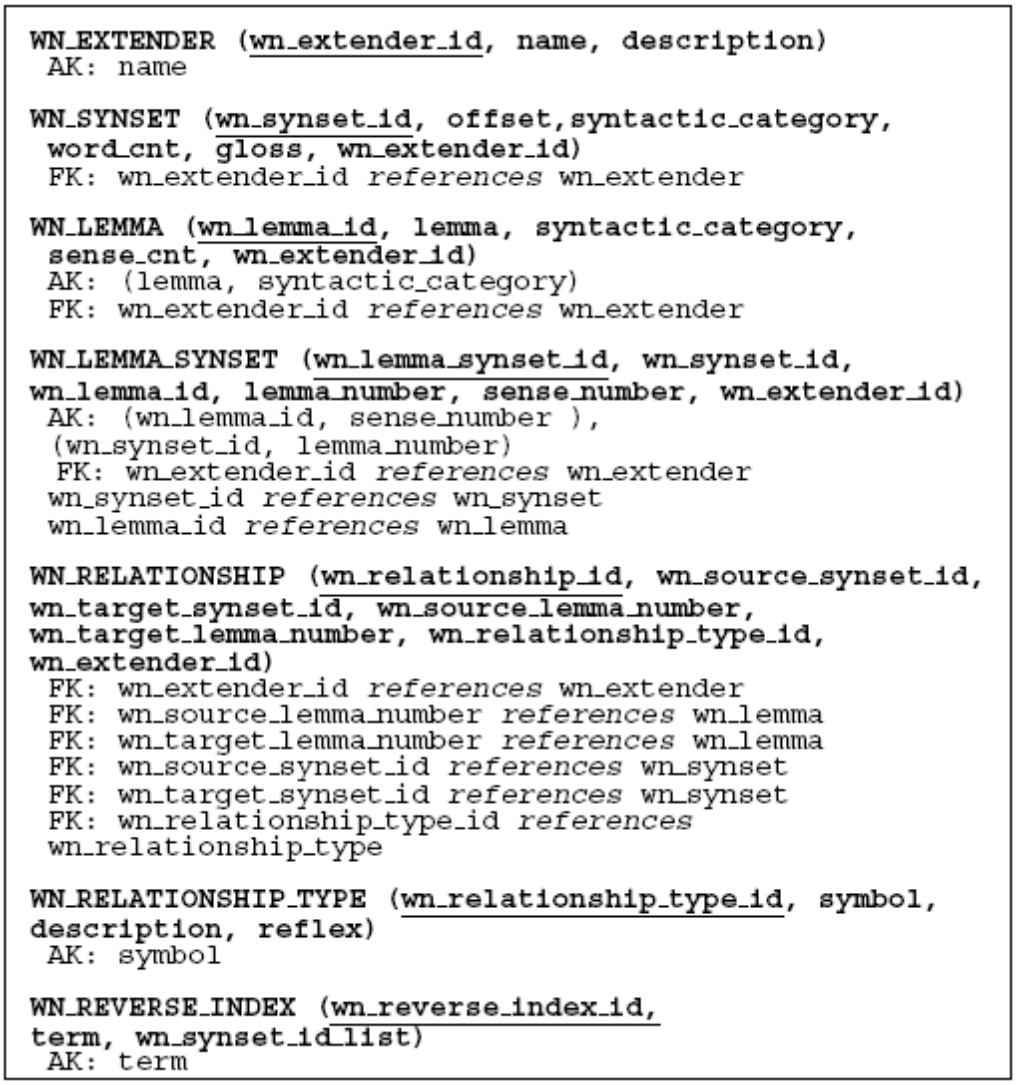


Figura 4.2: Relazionale di WordNet in momiswn

Non è certamente inutile ricordare qui lo schema relazionale di *momiswn*, dal momento che dovremo utilizzare proprio questo nell'integrare WordNet Domains .

4.2 Analisi del processo di integrazione

4.2.1 La struttura delle informazioni in WordNet Domains

WordNet Domains viene, come precedentemente accennato, rilasciato gratuitamente da ITC-irst [0]. Il pacchetto comprende i file con le informazioni sulle gerarchie dei domini nelle varie versioni precedenti, la loro corrispondenza con i codici DDC (Dewey Decimal Classification) [20] e [21] quali vengono tarati. Ci sono anche uno storico con tutte le versioni rilasciate fino all'attuale, le versioni di WordNet con cui sono compatibili i file contenuti nel package e le modifiche rispetto le versioni anteriori, nonché i *wn-affect* che definiscono una gerarchia di domini per così dire "affettivi" (cioè domini che comprendono tutto ciò che attiene alla sfera dell'affettività, dei sentimenti, delle emozioni ecc..) che però esulano dall'ambito di nostro interesse.

Infine, ciò che è più rilevante per noi, i file (di testo) che contengono tutte le associazioni che legano i synset di WordNet con i domini di WordNet Domains.

In particolare, la versione 3.2 di WordNet Domains fornisce due file:

- wn-domains-2.0 che contiene la versione compatibile con WordNet 1.6;
- wn-domains-3.2 che è quella compatibile con la versione 2.0 di WordNet e quindi quella da noi utilizzata.

Il file è suddiviso in righe, ciascuna delle quali fa riferimento ad un particolare synset, e riporta una o più etichette di dominio cui quel synset dovrebbe essere associato. Nello specifico, ciascuna linea del file ha la forma

byte_offset-categoria_sintattica dominio_1 dominio_2 ... dominio_n

dove *byte_offset* e *categoria_sintattica*, identificano palesemente un synset di WordNet, mentre *dominio_1 .. dominio_n* indicano i nomi dei suoi domini.

Es:

...

00005598-n factotum

00006026-n biology person

00012748-n animal biology

…

Infatti, ciascun synset può essere associato ad uno o più domini sebbene, nella pratica, non si va oltre 3 o 4 in un numero limitato di casi e per la stragrande maggioranza un solo dominio.

Numero di domini di associazione	Numero di synset
1	91586
2	22199
3	1484
4	145
5	7
6	2
7	1

Tabella 1: Numerosità delle classi di synset sulla base del numero di domini associati

Facciamo notare che il *byte_offset* ha il formato di 8 cifre decimali che è anche quello adottato da WordNet e anche il carattere che indica la categoria sintattica rispecchia la notazione di WordNet (-n nome; -v verbo; -a aggettivo; -s aggettivo satellite; -r avverbio).

Un ulteriore osservazione è quella che il file risulta ordinato per categoria sintattica e, a parità di categoria, per byte offset.

4.2.2 Requisiti per l'integrazione

L'integrazione dei domini di WordNet Domains deve essere fatta attenendosi a dei requisiti di semplicità dell'importazione dei dati (in termini di procedure di importazione) e non invasività sulla struttura del database *momiswn*; ovvero si richiede che le nuove informazioni vengano aggiunte ed integrate senza andare a modificare lo schema del database esistente e, di riflesso, le modalità di accesso e modifica allo stesso.

Inoltre, si è ritenuto opportuno, insieme alle relazioni che legano i synset ai domini, aggiungere le informazioni di struttura gerarchica di WordNet Domains, cioè le relazioni tra un dominio e l'altro; ciò può rivelarsi utile in tutti quegli algoritmi di disambiguazione che possano sfruttare questa caratteristica.

Bisogna osservare tuttavia che in WordNet Domains, tutte le indicazioni sulla gerarchia tra i domini sono contenute in file pdf assieme ad altre informazioni di versione e perciò non sono direttamente utilizzabili. Si è perciò deciso di creare un file di testo ad hoc che tenesse conto della struttura ad albero dei domini.

Se si immaginano tutti i domini come i nodi di un albero, possiamo rappresentare tale albero come l'insieme di tutte le coppie (nodo_padre, nodo_figlio) che vi si trovano. Quindi il file riporterà in ciascuna riga la coppia (dominio_padre, dominio_figlio) nel formato sotto riportato

dominio_padre dominio_figlio

Il file è stato creato manualmente facendo riferimento alla documentazione fornita menzionata sopra. Il numero di domini totale non è alto (ci sono 168 etichette di dominio nella versione utilizzata) sicché si è potuto procedere in tale maniera abbastanza facilmente.

4.2.3 Una prima soluzione

Si è pensato, in un primo momento, che la maniera più naturale di integrare WordNet Domains con la preesistente ontologia lessicale di WordNet, fosse quella di modellare un'entità *wn_domain* e che fosse sufficiente collegarla in associazione n-n con l'entità *wn_synset*; si aggiungeva poi un'altra associazione 1-n della nuova entità con se stessa per rappresentare la relazione *figlio-di* o *is-a*.

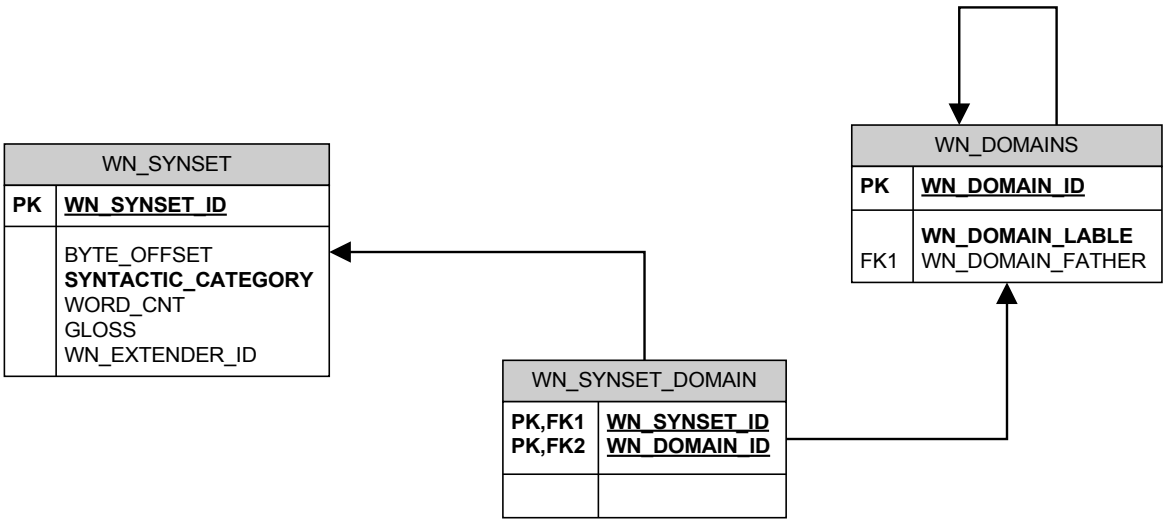


Figura 4.3: Schema della prima soluzione esaminata

Da cui si ricava il seguente schema relazionale:

WN_DOMAINS(WN_DOMAIN_ID,WN_DOMAIN_LABEL,WN_DOMAIN_FATHER)

AK: WN_DOMAIN_LABEL

FK: WN_DOMAIN_FATHER references WN_DOMAINS

WN_SYNSET_DOMAIN(WN_SYNSET_ID,WN_DOMAIN_ID)

FK:WN_SYNSET_ID references WN_SYNSET

FK:WN_DOMAIN_ID references WN_DOMAINS

La tabella *wn_domains* contiene tutti i domini di WordNet Domains e inoltre la *foreign key* verso se stessa, che rappresenta l'associazione *is-a* e lega un dominio con il proprio genitore; *wn_synset_domain* invece è la tabella di associazione tra la precedente (con i domini) e

wn_synset di *momiswn*.

4.2.4 Limiti

Questa soluzione è abbastanza semplice ed ha il vantaggio di separare logicamente e fisicamente (poiché entità e relazioni nuove) ciò che riguarda WordNet dalle informazioni di dominio sulle quali si può predicare ed operare separatamente. Inoltre, il database originale non viene modificato nella sua struttura, così come ci si era proposti.

Tuttavia, dopo una più attenta valutazione, questo tipo di approccio mostra qualche limite che ne complica la realizzazione.

Infatti, il problema principale di questa soluzione è il costo in termini di procedure per creare, accedere ed operare sulle nuove relazioni introdotte, nel senso che occorre definire e creare, a livello di programmazione in Momis, i metodi (essenzialmente codice Java) per accedere e modificare i record delle nuove relazioni di dominio, oltre a quelli di interrogazione di tali tabelle da sole e in combinazione con quelle di WordNet già presenti. Senza contare il fatto che creare nuove relazioni è di per se una operazione non banale in Momis.

In altre parole occorre metter mano abbastanza pesantemente dentro al codice esistente, cosa che si preferirebbe evitare se non necessario.

4.2.5 Breve excursus sugli strumenti di base di Momis..

Il collegamento tra Momis e il DBMS sottostante (nel nostro caso MySql) è molto stretto in quanto le interazioni avvengono attraverso oggetti creati a questo scopo.

Infatti momis utilizza un tool chiamato Torque, sviluppato inizialmente come parte del Jakarta project di Apache, ma poi diventato autonomo; tra le altre cose si occupa della connessione al database che si intende usare e della creazione di oggetti Peer collegati direttamente alle tabelle e capaci di interagire in vario modo con esse gestendo una serie di connessioni Jdbc in maniera trasparente.

Quindi avviando Torque otteniamo la generazione di oggetti e perciò classi Java che verranno usate per modellare il database momiswn, consentendo poi di creare, cancellare o selezionare oggetti che rappresentano record delle tabelle del database stesso.

Gli oggetti generati sono di quattro classi differenti per ciascuna tabella dello schema. A titolo di esempio, dalla tabella wn_synset verranno generate le quattro classi

1. WnSynset
2. WnSynsetPeer
3. BaseWnSynset
4. BaseWnSynsetPeer

la prima e la seconda sono sottoclassi rispettivamente della terza e quarta; le due classi di base (quelle che iniziano con Base) contengono tutta la logica generata da Torque in termini di metodi di accesso al singolo record e alla tabella per interrogazioni e modifiche. WnSynset e WnSynsetPeer aggiungono la logica di business, ovvero quella che serve all'applicazione che deve operare sui dati.

In particolare WnSynset rappresenta proprio un oggetto riga della tabella e possiamo tramite esso andare a manipolare i campi e salvarlo con il metodo save(); questa operazione equivarrà ad andare a modificare il record corrispondente nella tabella di riferimento.

Le classi Peer invece hanno funzionalità di wrapper verso le corrispondenti tabelle e forniscono metodi statici per interagire con interrogazioni, inserimenti cancellazioni e

aggiornamenti.

Questi sono sostanzialmente gli strumenti che andremo ad adoperare.

L'abbozzo di soluzione vista sopra comporta il dover riscrivere lo schema del database,ed andare a rigenerare tutti gli oggetti base associati in precedenza più quelli nuovi e riscrivere le loro sottoclassi.

4.3 Soluzione realizzata

Si è perciò optato per una seconda soluzione che, sulla linea di quanto proposto da Serena Sorrentino[18] utilizzasse la struttura già esistente di *momiswn* che già contiene l'ontologia di WordNet, senza modificare niente del suo schema.

Innanzitutto, affrontiamo il problema di come rappresentare le informazioni di dominio. Un dominio è essenzialmente un'etichetta (che rimanda ad un ben preciso ambito della conoscenza), ed è diversa da tutte le altre etichette di dominio nella gerarchia; ben si adatta quindi ad essere rappresentata come un lemma, che all'interno di una categoria sintattica, è univoco. Si può quindi vedere il dominio come un lemma di una categoria "speciale".

Tuttavia, i lemmi all'interno dello schema E/R di *momiswn*, non possono essere utilizzati direttamente per esprimere associazioni e quindi relazioni di gerarchia, di appartenenza ecc... ma devono essere collegati a dei synset che invece lo possono.

Si andranno quindi a creare delle coppie "lemma-synset" legate tra loro da un'associazione uno a uno, per rappresentare i domini; poi si creeranno le relazioni tra synset di dominio e synset WordNet di tipo "Domain of synset" e "Member of this domain", una l'inversa dell'altra, già definite in *momiswn*.

Per descrivere la gerarchia dei domini andremo a creare due nuovi tipi di relazione tra loro simmetriche e cioè:

- *WordNet Domains domain of synset* che esprime la relazione padre-figlio nella gerarchia e sarà indicato con il simbolo ";WordNetDomain";
- *WordNet Domains Member of this domain* cioè la relazione inversa figlio-padre e viene indicata con il simbolo "-WordNetDomain".

Questi nuovi tipi saranno utilizzati per creare poi le relazioni tra domini e domini .

4.3.1 Dettagli implementativi

Il primo passo è stato quello di creare una costante chiamata "SC_DOMAIN" che sarà usata come categoria sintattica che indica un dominio, cioè nei lemmi e nei synset, e non una categoria classica di WordNet.

Poi, riutilizzando un metodo del *loader*, la classe che carica WordNet in *momiswn*,[1] creato in Momis, si è caricata una nuova estensione nella tabella *wn_extender*; si è cercato il record con i campi

WN_EXTENDER_ID	NAME	DESCRIPTION
3	wnd	WordNet Domain Extender

Figura 4.4: Valori inseriti nei campi dell'extender per WordNet Domains

Questa estensione verrà referenziata nelle altre tabelle come FK (Foreign Key) ogni volta che un lemma, un synset o una relazione vedono coinvolto un dominio.

Il terzo passo consiste nel creare i nuovi tipi di relazione di gerarchia inserendo due record nella tabella *wn_relationship_type*; come già premesso gli altri due tipi di relazione che utilizzeremo per legare synset con domini, erano già presenti in tabella.

[1] La classe *loader* WordNetLoader implementa, utilizzando le classi associate alle tabelle, tutti i metodi per popolare il database *momiswn* a partire dai file forniti con WordNet .

WN_RELATIO...	SYMBOL	DESCRIPTION	REFLEX
14	;c	Domain of synset - TOPIC (nouns,verbs,adjectives,adverbs)	1
15	-c	Member of this domain - TOPIC (nouns)	1
16	;r	Domain of synset - REGION (nouns,verbs,adjectives,adver...	1
17	-r	Member of this domain - REGION (nouns)	1
18	;u	Domain of synset - USAGE (nouns,verbs,adjectives,adverbs)	1
19	-u	Member of this domain - USAGE (nouns)	1
20	*	Entailment (verbs)	0
21	>	Cause (verbs)	0
22	^	Also see (verbs,adjectives)	0
23	$	Verb Group (verbs)	0
24	&	Similar to (adjectives)	1
25	<	Participle of verb(adjectives)	0
26	\	Pertainym (pertains to noun,adjectives)|Derived from adjecti...	0
27	;WordnetDomain	WordNetDomain Domain of synset - TOPIC	1
28	-WordNetDomain	WordNetDomain Member of this domain - TOPIC	1

Figura 4.5: Una porzione dei record di wn_relationship_type

Il campo “REFLEX” se settato a 1 indica che la relazione è simmetrica, altrimenti è a 0.

A questo punto si deve importare il vero e proprio contenuto informativo di WordNet Domains andandolo ad estrarre dai file del pacchetto.

Dal momento che il file delle gerarchie è utilizzato per creare relazioni ma ci è utile anche per poter scorrere rapidamente tutti i domini ed essendo il numero totale di domini non elevato, si è deciso di creare una nuova classe *WordNetDomainLoaderHierarchy* che implementa una struttura dati (una lista) su cui va a caricare dal file tutte le coppie di domini (padre,figlio), ed è in grado di metterli a disposizione insieme o separatamente.

Istanziando questa classe dunque e iterando su ciascun elemento figlio, creiamo per ogni dominio gli oggetti *WnLemma*, *WnSynset* ed il collegamento tra loro *WnLemmaSynset*: ciò corrisponde a creare una entry nelle tabelle *wn_lemma, wn_synset* e *wn_lemma_synset* con i valori dei campi specificati di seguito, supponendo *l'extender_id* uguale a 3

Tabella	Campo	valore
wn_lemma	LEMMA	dominio
	SYNTACTIC_CATEGORY	SC_DOMAIN
	SENSE_CNT	1
	WN_EXTENDER_ID	3
wn_synset	BYTE_OFFSET	0
	SYNTACTIC_CATEGORY	SC_DOMAIIN
	WORD_CNT	1
	GLOSS	dominio
	WN_EXTENDER_ID	3
wn_lemma_synset	WN_SYNSET_ID	<Id del synset>
	WN_LEMMA_ID	<Id del lemma>
	LEMMA_NUMBER	1
	SENSE_NUMBER	1
	WN_EXTENDER_ID	3

Tabella 2: Valori da assegnare ai campi per importare i domini

Utilizzando gli oggetti associati alle tabelle del database che vengono create in Momis, e quello di classe *WordNetDomainLoaderHierarchy* appena creato, si recuperano gli identificativi dei synset di ogni coppia e si creano le due relazioni di gerarchia.

Si noti che tali relazioni hanno i campi “source_lemma_number” e “target_lemma_number” a 0, ad indicare una relazione tra synset, che nell'ambito di WordNet indica una relazione semantica.

Infine, si devono caricare le relazioni che legano i synset ai domini: anche qui si creano due oggetti *WnRelationship* associati alla tabella *wn_relationship* che rappresentano i due versi della relazione e vengono salvati nei corrispettivi record di tabella.

Per fare questo però si deve andare ad aprire in lettura il file “wn-domain-3.2” e per ciascuna riga

- estrarre l'identificativo del synset corrispondente al *byte_offset* ed alla categoria

sintattica letta tramite l'oggetto Peer corrispondente;

- per ciascun dominio presente sulla linea si creano le due relazioni con il synset trovato sopra;
- si salvano tali relazioni

e l'applicazione termina in quanto ha raggiunto il suo scopo.

La classe d'importazione dei domini, una volta lanciata, impiega dai 30 ai 40 min per caricare tutto;

un pò meno se lanciata manualmente dal *prompt* del sistema operativo.

Tutto è stato implementato impiegando i metodi di base già definiti in Momis con particolare riferimento agli oggetti di collegamento alla base di dati messi a disposizione.

Tutto ciò che si è fatto è stato creare tre nuove classi Java, una che effettuasse il caricamento utilizzando gli strumenti disponibili, le altre due di supporto e funzionali alla prima.

WordNetDomainLoader che è la classe principale, il *loader* per i file di WordNet Domains in Momis; essa viene lanciata con la seguente sintassi

WordNetLoader configFile wnDbDir wnDomain_file gerarchie_file

dando come parametri di ingresso

- *configFile* il nome assoluto del file di configurazione che contiene tra le altre cose i parametri di connessione a *momiswn* tramite il particolare DBMS usato (MySql) ed usato per l'inizializzazione di Torque;
- *wnDbDir* il percorso della cartella con i file di wordnetdomains e della gerarchia
- *wnDomain_file* il nome del file contenente WordNet Domains
- *gerarchie_file* il nome del file delle gerarchie.

Questa classe carica, con l'ausilio delle classi successive e di quelle messe a disposizione già viste sopra, le informazioni da integrare in ciascuna delle tabelle interessate.

WordNetDomainLoaderHierarchy una classe per importare la gerarchia in una struttura dati in memoria

WordNetDomainLoaderRecord che è una classe ausiliaria che parsa ciascuna riga del file dei domini ne rende disponibile i campi

4.3.2 Considerazioni

Abbiamo, in tal modo, adempiuto a quanto ci eravamo prefissi in partenza e cioè mantenere la struttura del database originario senza andare a modificare il codice che era già scritto per operare su quella struttura ma anzi utilizzandolo per fare l'integrazione.

Un ulteriore osservazione che ritengo utile fare è che, al di la dell'efficienza del codice scritto per l'importazione (che può essere limato), l'informazione di dominio all'interno dello schema di database precedente mantiene una sua coerenza ed autonomia rispetto a WordNet, visto che si possono utilizzare i valori di categoria sintattica o di *extender* per distinguerli.

Tuttavia, utilizzando questo meccanismo si possono in futuro prevedere integrazioni di domini ristretti a determinati ambiti e campi applicativi più specifici di WordNet Domains in maniera abbastanza semplice, all'interno della stessa struttura informativa di *momiswn*, che assumerebbe una valenza un pò più ampia rispetto ad una ontologia lessicale.

Conclusioni

In questa tesi è stata descritta la soluzione adottata per risolvere il problema dell' integrazione delle informazioni di dominio contenute in WordNet Domains con quelle lessicali contenute in WordNet nell'ambito di MOMIS.
Questo perché WordNet da solo, ha mostrato alcune lacune che si ripercuotevano sulle capacità di disambiguazione del sistema. Con i domini si riescono a superare tali limiti e costituiscono, integrati in *momiswn*, un migliore strumento di lavoro. In particolare poi, le proprietà di bilanciamento , copertura e granularità fanno di WordNet Domains un'ontologia di domini particolarmente adatta.
Il processo ha richiesto alcune semplici modifiche al codice consistenti nell'introduzione di una nuova classe *loader* e altre due classi di supporto.
L'esigenza preminente era quella della semplicità e non invasività sul sistema esistente, sia per quel che riguarda il database *momiswn* sia per quanto riguarda il codice sovrastante.
A mio avviso si è riusciti a soddisfare la richiesta.
Anzi si è addirittura utilizzato quanto già c'era per introdurre qualcosa di nuovo e raggiungere il nostro obiettivo.

Personalmente questo lavoro mi ha sicuramente giovato, in termini di esperienza e arricchimento personale consentendomi di affrontare tematiche a me nuove e stimolanti, dandomi nel contempo l'opportunità di fornire un utile servizio al progetto MOMIS .

Un appunto che si potrebbe fare riguarda il fatto che si è dovuto creare *ex-novo* il file delle gerarchie di WordNet Domains che è una delle fonti informative per una corretta integrazione. Da questo punto di vista si potrebbe richiedere all'istituto ITC-irst di fornire un supporto fruibile da cui estrarre la vera e propria struttura gerarchica dei domini.

Allargando un po’ il campo si potrebbe pensare, in futuro, all'integrazione di altre estensioni di dominio per così dire 'specialistiche', cioè ristrette a certi ambiti applicativi. In questa prospettiva si potrebbe anche rendere l'implementazione, descritta nella presente tesi, più generale e versatile in modo da poter affrontare questo problema.

Riferimenti

[0] Sito Web di ITC-irst : per il download del package WordNet Domain http://wndomains.itc.it/download.html

[1] Gorge A. Miller: *WordNet: a lexical database for english.* Communications for the ACM,38(11): 39-41, 1995.

[2] R. Hull and R. King et al. *Arpa I3 reference architecture.* 1995. Reperibile presso: http://www.isse.gmu.edu/I3_Arch/index.html/.

[3] Gio Wiederhold et al. *Intergrating artificial intelligence and database technology.* Journal of Intelligent Integration System, 2/3 Giugno 1996.

[4] F.Saltor and E. Rodriguez .*On intelligent access to heterogeneous information*. In Proceeding of the 4th KRDB Workshop, Atene, Grecia, Agosto 1997.

[5] D. Beneventano, S. Bergamaschi, S.Lodi, e C. Sartori. *Consistency checking in complex object database schemata with integrity constraints*. Technical Report 103, CIOC , Bologna, Italia , 1994.

[6] S. Bergamaschi e B.Nebel. *Acquisition and validation of complex object database schemata supporting multiple inheritance.* Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks and Complex Problem Solving Technologies, 4:185-203, 1994.

[7] D.Beneventano, S.Bergamaschi, C.Sartori e M.Vincini. *ODB-tools: a description logics based tool for schema validation and semantic query ottimization in object oriented*

databases. In Proc. Of Int. Conference of the Italian Association for Artificial Intelligence (AI*IA, 97), Roma, 1997.

[8] D.Beneventano, S.Bergamaschi, C.Sartori, e M.Vincini. *Odb-qoptimizer: a tool for semantic query optimization in oodb*. In Proc. Of Int. Conf. On Data Engineering ICDE'97, Birningham, UK, April 1997.

[9] D.Beneventano, S.Bergamaschi, C.Sartori, e M.Vincini. *A description logics based tools for schema validation and semantic query optimization in oodb*. In Proc. Of Int. Conf. On Data Engineering ICDE'97, Birningham, UK, April 1997.

[10] S.Castano e V.De Antonellis. *Deriving global conceptual views from multiplie information sources*. In preProc. Of ER'97 Preconference Symposium on Conceptual Modeling, Historical Perspective and future Directions, 1997.

[11] T. Catarci e M. Lenzerini. *Rapresenting and using interschema knowledge in cooperative information systems.* Journal of Intelligent and Cooperative Information Systems, 2(4)375-398, 1993.

[12] B. Everitt. *Computer-Aided Database Design: the DATAID Project*. Heinemann Educational Books Ltd, Social Science Research Council, 1974.

[13] R.G.G. Cattel, editor. *The object Database Standard: ODMG93*. Morgan Kaufman Publisher, San Francisco, CA, 1997.

[14] R Benassi, S.Bergamaschi, A.Fergnani e D.Miselli. *Extending a Lexicon Ontology for Intelligent Information Integration.*

[15] V.Guidetti. *Intelligent information integration system: Extending a lexicon ontology.* Master thesis in Computer Science, Università di Modena e Reggio Emilia, 2006.

[16] B.Magnini, S.Strapparava et. al.. *The role of domain Information in Word Sense Disambiguation.* Natural Language Engineering, 25 luglio 2002.

[17]A.Gliozzo, C.Strapparava, I. Dagan. *Unsupervised e Supervised Exploitation of Semantic Doamins in Lexical Disambiguation.* 2002.

[18] S.Sorrentino. *Metodi di disambiguazione del testo ed estensioni di WordNet nel sistema Momis.* Master thesis in Computer Science, Università di Modena e Reggio Emilia, 2002.

[19] Sito web del Princeton WordNet Home Page: http://wordnet.princeton.edu

[20] A.Diekema. Dewey Decimal Classification, 1998.

[21] L. Bentivogli e al.. *Revising the WordNet Domains Hierarchy: semantics, covarage e balancing.* Corpus Linguistics 2003, Conference, Lancaster United Kingdom.

Ringraziamenti:

Vorrei ringraziare la Prof. Sonia Bergamaschi e il suo staff, in particolare l'Ing. Serena Sorrentino e l'Ing. Alberto Corni per l'aiuto e la pazienza.

Ringrazio la mia famiglia che aspettava questo momento e Carol, Meryem e Selma che ormai mi sopportano da anni e Ron per le dritte.

Tutti gli altri della BSI e X-Rum non riesco a citarli tutti.. grazie.