Al Prof. Massimo De Santo
che mi ha dato l'opportunità
di realizzare questo lavoro e
all'ing. Paolo Napoletano
che mi ha aiutato a svolgerlo.

Alla mia famiglia
per il sostegno durante
questo intenso percorso.

Indice

Introduzione

L'estrazione e la rappresentazione della semantica contenuta nel linguaggio sono tra i principali argomenti che da sempre animano le discussioni in psicologia cognitiva e intelligenza artificiale. Nell'ambito di queste comunità scientifiche il dibattito è riferito principalmente al problema di individuare il modo migliore per rivelare il significato che risiede in un qualsiasi atto comunicativo: *scrivere, leggere, parlare, ecc.* In alcuni rami applicativi dell'intelligenza artificiale così come nella Knowledge Engineering, i risultati di questi dibattiti sono stati utilizzati per introdurre nuovi linguaggi formali grazie ai quali è stato possibile sia rappresentare la semantica su un calcolatore che manipolarla per l'esecuzione di ragionamenti automatici. Questi linguaggi, tra cui XML (eXtensible Markup Language), RDF (Resource Description Framework), OWL (Ontology Web Language), sono stati la risposta tecnologica alla nascente visione del Web Semantico introdotta da Tim Berners-Lee, formalmente l'inventore del World Wide Web. Secondo questa nuova visione, il Web dovrebbe essere una rete altamente interconnessa di dati facilmente accessibile e comprensibile ad un qualsiasi calcolatore, un desktop o palmare, dove agenti software intelligenti sono in grado di risolvere richieste complesse dell'utente.

A tutt'oggi alcune importanti questioni legate alla nascita del Web Semantico non sono state risolte: questioni che si riferiscono principalmente al modo per rivelare e rappresentare la semantica stessa. Il problema fondamentale è che scoprire le intenzioni dell'autore, ad esempio contenute all'interno di un testo, può essere un processo molto complesso e soprattutto ambiguo per via del fatto che il *significato* stesso esiste indipendentemente dal linguaggio utilizzato e dal processo d'interpretazione. E' ben noto, infatti, che a causa delle imperfezioni del linguaggio umano i processi d'interpretazione ovvero codifica e decodifica del messaggio-significato potrebbero essere imperfetti e dunque

corrotti da rumore. La rivelazione della semantica del testo, dunque, non può essere fatta semplicemente associando un significato al testo attraverso linguaggi formali ma piuttosto deve essere realizzata utilizzando metodi in grado di considerare il fattore rumore intrinseco ai processi stessi e, di conseguenza nasce così l'esigenza di utilizzare ulteriori linguaggi - per forza di cose probabilistici - in grado di gestire al meglio tali processi rumorosi.

La Knowledge Engineering, dal canto suo, dovrebbe poi fornire semplicemente gli strumenti per facilitare l'interazione dell'utente con i dati, visto che i processi in gioco possono essere considerati situati e temporali in cui il testo stesso agisce come condizione a contorno e nel quale l'utente è, ex necessitate, il protagonista. Un processo d'interpretazione del significato dovrebbe così portare in conto i diversi livelli di rappresentazione della semantica, che partono dal testo - livello più basso - fino all'utente - livello più alto - usando un formalismo tale da consentire il trattamento dell'incertezza. Progettare modelli per la semantica in grado di non tralasciare tutti questi aspetti sembrerebbe l'unico modo per assegnare il significato ad un testo: significato che può essere rappresentato attraverso un linguaggio formale oltre che dal linguaggio naturale e che si presta facilmente ad essere manipolato da un agente artificiale anche e, come nel nostro caso, attraverso l'utilizzo di metodi automatici basati su ontologie.

In questo lavoro di tesi è stato così introdotto un sistema a livelli, come accennato in precedenza, per la manipolazione del significato secondo il quale l'estrazione della semantica può avvenire attraverso l'analisi delle relazioni tra i diversi tipi di elementi tra cui *parole*, *concetti* e *percetti*. Lo sforzo impiegato nella formalizzazione di questi aspetti ha dato luogo a due differenti tendenze. Una prima che si concentra principalmente sulla struttura delle relazioni associative tra parole del linguaggio naturale e sulle relazioni tra concetti e parole, che si definisce come la parte più superficiale della semantica (che possiamo estrarre direttamente dai testi), o *light semantics*. Una seconda che esalta strutture concettuali astratte, concentrandosi su relazioni tra concetti, relazioni tra concetti e percetti, relazioni tra percetti e azioni, e che si definisce come la componente più profonda della semantica (che si può estrarre ad esempio studiando l'utente ed i suoi comportamenti), o *deep semantics*. Una volta introdotto il modello generale, si è posta attenzione solo ad alcuni degli aspetti discussi precedentemente (si veda

Figura 1 Rappresentazione a livelli della semantica: il modello computazionale proposto in questo lavoro si riferisce alle relazioni incluse nella cornice rossa.

la figura 1) cosicché il cuore della nostra proposta è quindi la definizione di un tipo di conoscenza informale, indicata da noi come *informal Lightweight Ontology* (*iLO*)e che può essere desunta automaticamente da documenti.

Tale rappresentazione della conoscenza può essere ancora vista come un *Grafo di Termini* o *Grafo di Concetti*, composto cioè da nodi (i concetti stessi) e da collegamenti pesati tra essi (in grado di conservare e rappresentare le relazioni semantiche tra concetti) e dove ogni nodo-concetto può essere specificato attraverso un ulteriore grafo (in questo caso si parlerà di parole come nodi e legami/archi, ancora una volta pesati, tra parole). A questo punto, se per entrambi i grafi, i pesi dei diversi legami (tra concetti e tra parole) sono intesi attraverso una probabilità, è dunque possibile compiere inferenza e quindi apprendere una rappresentazione di un concetto e/o di un grafo di concetti attraverso tecniche probabilistiche[1].

Nell'ambito di questo lavoro è stato poi realizzato un ambiente sperimentale in grado di testare l'approccio proposto e verificarne la sua effettiva bontà quando impiegato in casi reali per l'interpretazione dell'intenzione utente. Specificatamente abbiamo realizzato così due scenari di interesse:

I) il primo sfrutta le potenzialità di categorizzazione concettuale delle *iLO* su grandi collezioni di dati testuali, per esempio repository di pagine web;

[1] All'uopo si è pensato di utilizzare una versione smoothed della Latent Dirichlet Allocation conosciuta in letteratura anche come Topic Model così come sarà ben descritto nel seguito di questo tesi.

II) il secondo più focalizzato in ambito di *User Satisfaction* sfrutta la stessa tecnica per recuperare da un repository di pagine web, contenuti più vicini alle intenzioni degli utenti quando questi effettuano queries di tipo informazionali.

Nel primo ambiente di testing, proprio per sollevarci dalla soggettività dei dati sperimentali, abbiamo previsto il confronto con un motore di ricerca text-based puramente sintattico molto diffuso in ambiente open source, *Lucene*, attraverso l'uso di indici di prestazione specifici (*Precision* e *Recall* tra gli altri) dei motori di ricerca su web. Allo stesso modo e per avere una ulteriore conferma sulla bontà del nostro sistema, all'interno della seconda fase di sperimentazione sono stati così condotti esperimenti in diversi contesti e, per ogni uno di essi, è stato richiesto ad alcuni gruppi di esseri umani di assegnare dei giudizi di rilevanza per il set di pagine web restituite sia da un motore di ricerca classico (nella fattispecie una versione personalizzata di Google - *Google Custom Engine*) e dal nostro motore di ricerca implementato con l'ausilio di *Lightweight Ontology*.

In entrambi i casi, i risultati ottenuti hanno confermato che la tecnica proposta aumenta sicuramente le prestazioni in termini di rilevanza e rispondenza alle vere intenzioni dell'utente e, poiché un'ontologia, almeno per come l'abbiamo intesa noi in questo lavoro, consiste di concetti e di collegamenti tra essi, una maggiore specializzazione di intenti, ovvero una rappresentazione coerente del significato, risulta molto utile per ridurre i problemi legati l'ambiguità intrinseca del linguaggio[2]. Questo lavoro di tesi è stato poi così organizzato: la prima parte, articolata in due capitoli, costituisce un inquadramento generale del problema ed in essa vengono introdotti i concetti chiave della semantica - *capitolo 1*, si espongono le difficoltà dei nei confronti di problemi quali la rappresentazione del significato e l'ambiguità del linguaggio e si esaminano le recenti tendenze dei diversi approcci, probabilistici e non, ad accostarsi a tali problematiche - *capitolo 2*. Con il *capitolo 3* si presenta invece la nostra proposta per una rappresentazione coerente e funzionale del significato stesso attraverso un *Grafo di Termini*: rappresentazione che vede poi nel *capitolo 4* una cospicua fase di sperimentazione in scenari reali e secondo esigenze pratiche derivate da comportamenti di diversi tipi di utenti.

[2]Si rimanda così il lettore all'ultimo capitolo di questo lavoro per il dettaglio sulle prestazione e per apprezzarne chiaramente il contributo innovativo.

Capitolo 1

Rappresentazione della semantica - problematiche generali

Sebbene la rappresentazione della conoscenza è uno dei concetti centrali e, in qualche modo, più familiari in AI, la questione più importante a questo proposito riguarda proprio: "*ma che cosa è la Knowledge Representation?*" Raramente c'è stata una risposta diretta: numerose riviste hanno fatto pressioni per uno o un altro tipo di rappresentazione, altre hanno spinto solo su alcune proprietà che una rappresentazione dovrebbe avere, altri ancora si sono concentrati in generale sulle proprietà più importanti per definire effettivamente il concetto di rappresentazione. La risposta può essere forse meglio compresa in termini dei cinque ruoli importanti e nettamente diversi che la rappresentazione stessa gioca, ognuno dei quali pone richieste distinte e a volte contraddittorie sulle diverse proprietà così come descritto da Davis et al. in [15]. La KR può essere vista allora come:

1. Un surrogato, ovvero un sostituto per l'oggetto in sé, utilizzato per consentire ad un utente di determinare conseguenze attraverso il "*pensare*" piuttosto che "*l'agire*": in altre parole direttamente collegato ai ragionamenti piuttosto che alle azioni;

2. Un set di scelte ontologiche o semplicemente una risposta alla domanda: in quali termini si deve pensare il mondo?

3. Una teoria frammentaria di "*ragionamento intelligente*", espresso a sua volta come combinazione di:

   - fondamenti della rappresentazione,
   - insieme delle conseguenze dovute al ragionamento stesso,
   - serie di inferenze che da esso scaturisce.

4. Un mezzo per una computazione pragmaticamente efficiente, ovvero l'ambiente di calcolo in cui il pensiero matura. Un contributo in questo senso è fornito dalle guide su come organizzare le informazioni in modo da rendere agevoli i ragionamenti e le inferenze;

5. Un mezzo di espressione umana, ovvero un linguaggio in cui affermiamo determinate cose sul mondo.

Sinteticamente possiamo quindi indicare come principale preoccupazione nel campo della Knowledge Representation, la comprensione e la descrizione delle ricchezze del mondo ovvero più semplicemente degli oggetti presenti nella vita di ogni essere umano. Siamo in grado di distinguere allora due tipi di approccio a tali problemi così come introdotto da Predoiu e Stuckenschmidt in [60]:

I) **Approccio classico** - basato su linguaggi di tipo logico (legato a dichiarazioni vere o false), ed implementato con modelli semantico-teorici che però presentano alcune lacune soprattutto nei mezzi per la rappresentare dell'incertezza: Resource Description Framework (RDF -RDF Schema), Ontology Web Language (OWL), Semantic Web Rule Language (SWRL), Description Logics (DL), agenti artificiali, ecc.

II) **Approccio probabilistico** - basato sia su estensioni probabilistica dei linguaggi per il Semantic Web tra cui: pRDF, BayesOWL, PrOWL, Probabilistic Logic, ecc. che su modelli - gioco forza di tipo probabilistico - come: Bayesian Networks (BNs), Bayesian Logic Programs (Stub), Independent Choice Logic (ICL), Multi-Entity Bayesian Networks (MEBNs), Probabilistic Datalog (pDatalog), ecc.

A questo punto, prima di intraprendere qualsiasi strada nella valutazione dei due approcci introdotti[1], è necessario tornare al problema

[1]Si rimanda il lettore al capitolo successivo per tale discussione.

iniziale per intendere fino in fondo cosa stiamo misurando e con quali mezzi, visto che rivelare le intenzioni dell'autore, ad esempio di un testo, può essere un processo molto complesso e soprattutto ambiguo per via del fatto che il significato stesso esiste indipendentemente dal linguaggio utilizzato e dal processo di interpretazione. Un processo di comunicazione attraverso il linguaggio, può essere riassunto come:

$$\textbf{significato} \rightarrow \textbf{codifica} \rightarrow \textbf{linguaggio} \rightarrow \textbf{decodifica} \rightarrow \textbf{significato}'$$

Nel caso in cui, ad esempio, il processo riguarda la lettura di un libro, gli attori coinvolti sono:

(a) autore/scrittore

(b) linguaggio

(c) lettore

In questo schema, all'origine del processo di comunicazione, c'è l'intenzione che risiede interamente nella mente dell'autore e in quello che egli desidera comunicare attraverso il testo. L'intenzione è dunque senza tempo, immutabile, pre-linguistica ed è codificata precisamente nella parte iniziale (a sinistra) dello schema: in altri termini, è interamente dipendente solo dall'atto dell'autore che la crea attraverso un suo processo e senza alcuna partecipazione da parte del lettore. L'autore traduce quindi tale creazione in un codice condiviso, il linguaggio, e lo invia aprendo una comunicazione al lettore. E' ben noto che, a causa delle imperfezioni casuali del linguaggio umano, tale processo di traduzione potrebbe essere imperfetto e dunque risulterebbe corrotto da rumore[2]. Una volta che l'intenzione codificata è trasmessa al lettore, inizia un processo di decodifica, anch'esso rumoroso, che conduce ad una ragionevole approssimazione dell'intenzione originale, così come era intesa dall'autore [65], [12]. E' possibile quindi affermare che il significato o "*intenzione*" non è mai interamente rappresentata da un codice, ma è sparsa lungo l'intera catena di codificatori attraverso il processo che Derrida [17] indica come neologismo differànce, cioè

[2]Solo in uno schema perfetto di traduzione si ha una riproduzione esatta del significato così come appare nella mente dell'autore.

"... processo che prende forma sul piano sintagmatico del testo" [20]. Tenere in conto gli aspetti chiave di tali processi di codifica e decodifica dell'intenzione è condizione necessaria per rivelare la semantica di un atto linguistico, ovvero descriverne la sua ontologia. Queste modalità risultano l'unico approccio corretto per assegnare il significato ad un testo: significato che può essere rappresentato attraverso un linguaggio formale, oltre che con il linguaggio naturale, e dunque idoneo ad essere manipolato da un agente artificiale che può, ad esempio, estrarlo utilizzando metodi automatici o semi-automatici. E' chiaro che la rivelazione della semantica di un testo non può essere realizzata associando semplicemente un significato al testo: la Knowledge Engineering dovrebbe quindi fornire gli strumenti utili a facilitare l'interazione dell'utente con i dati, ricordando che il processo di decodifica è praticamente un processo situato, temporale in cui il testo stesso agisce come condizione al contorno ma nel quale l'utente è, ex necessitate, il protagonista.

Alla luce di queste riflessioni, è possibile inquadrare il problema della rappresentazione della semantica, nello spirito di ciò che è stato discusso da T. L. Griffiths e colleghi in [70], come il problema di dedurre il legame, o meglio le relazioni, tra i diversi tipi di entità che sono proprie degli atti comunicativi: parole, concetti, percetti e azioni. Le relazioni che devono essere quindi prese in considerazione sono:

1. **Relazioni Parole – Concetti**: conoscenza del fatto che la parola cane si riferisce al concetto di cane, che la parola animale si riferisce al concetto di animale o che la parola tostapane si riferisce al concetto di tostapane;

2. **Relazioni Parole – Parole**: conoscenza del fatto che la parola cane si tende ad associarla o co-occorre ad altre parole come coda, ossa;

3. **Relazioni Concetti – Concetti**: conoscenza del fatto che i cani sono specie di animali, che i cani hanno la coda e che possono abbaiare, oppure che gli animali hanno un corpo e possono muoversi;

4. **Relazioni Concetti – Azioni**: conoscenza di come si addomestica un cane o come si utilizza un tostapane;

5. **Relazioni Concetti – Percetti**: conoscenza di quale è l'aspetto di un cane, di come un cane può essere distinto da un gatto.

Prendendo così spunto dalle discussioni introdotte in precedenza, questi aspetti non sono indipendenti tra loro ma influenzano il comportamento in diversi modi e sembrano essere ben descritti da rappresentazioni di tipo formale[3]. Come conseguenza di ciò, approcci differenti in grado di modellare la semantica tendono a porre il proprio focus solo su alcuni di essi fornendo così una prima macro-distinzione attraverso due filoni principali:

I) Il primo si concentra essenzialmente sulla struttura delle relazioni associative tra *parole* e *parole* del linguaggio naturale e su relazioni tra *concetti* e *parole* [45, 59, 22] così come indicato ai punti 1 e 2, e può essere percepito come la componente che rappresenta la parte più superficiale della semantica, o *light semantics* e che può essere rivelata semplicemente analizzando la struttura del testo;

II) Il secondo invece esalta strutture concettuali astratte, concentrandosi soprattutto su relazioni tra concetti, tra *concetti* e *percetti* e tra *percetti* e *azioni* [13]. Tale direzione è relazionata ai punti 3, 4 e 5, e può essere definita come la componente più profonda della semantica, o *deep semantics*, più slegata dal testo ma che coinvolge in prima persona l'autore stesso.

Seguendo tali distinzioni, si può allora notare che un agente artificiale potrebbe essere capace di rappresentare la semantica solo grazie all'interazione di entrambi gli aspetti di *light semantics* e *deep semantics* rispettivamente [12] ed inoltre appare evidente che il modo più naturale per trattare problemi di stimoli linguistici in presenza di incertezza - o rumore per come lo abbiamo indicato in precedenza - anche attraverso ragionamenti automatici risulta l'inferenza probabilistica . In questa direzione, è stato poi dimostrato che il linguaggio possiede una evidente struttura statistica e che potrebbe essere rivelata grazie all'utilizzo di modelli probabilistici del linguaggio e/o attraverso tecniche proprie del Machine Learning, della Statistica, dell'Information Retrival nonchè della Linguistica Computazionale[30]. Scendendo più in dettaglio, la descrizione di entrambi le relazioni *Parole–Parole* e *Parole–Concetti* che, seguendo ancora la nostra nomenclatura, indicheremo come afferenti alla *light semantics*, si può ulteriormente appoggiare ad una estensione di un modello computazionale, introdotto da Steyvers

[3]Si rimanda il lettore al capitolo successivo che inquadrerà il problema della rappresentazione formale o informale del significato.

e colleghi in [70] e noto in letteratura come *Topic Model*, dove si assume la dipendenza statistica tra le diverse occorrenze delle parole[4]. Per quel che concerne invece la *deep semantics*, essa è generalmente rappresentata in termini di sistemi astratti di proposizioni [13] ed i modelli presenti in letteratura per questa componente si concentrano principalmente nella comprensione dei fenomeni legati al comportamento umano come, ad esempio, l'evoluzione delle gerarchie concettuali che supportano la conoscenza proposizionale, del tempo di reazione da parte di soggetti adulti normali nell'analizzare proposizioni concettuali, del calo della reattività dell'analisi proposizionale dovuto all'età o dovuto alla presenza di danni al cervello, ecc.

Bisogna sottolineare però che mentre le relazioni *Concetti–Concetti* possono essere modellate utilizzando la *"prototype theory"* che gioca un ruolo centrale nella linguistica come risultato di un processo di mappatura dalle strutture fonologiche alla semantica [26], più interessante per noi risultano le relazioni *Concetti–Azioni* che possono essere rivelate usando la teoria della semantica emergente, nota come *emergent semantics*, mostrata da Santini e Grosky [64, 31]. Estrarre conoscenza - e quindi la semantica - sfruttando la percezione o, seguendo la nostra linea ragionamento, modellare le relazioni *Concetti–Percetti*, risulta un problema che può essere affrontato considerando la teoria computazionale di Marr [53]. In questo caso l'obiettivo si trasforma però nella comprensione di come un essere umano faccia uso della percezione stessa, per esempio visiva, nella codifica e rappresentazione del significato. Osservando ad esempio l'esplorazione di pagine web, l'essere umano impiega sia la visione che l'azione della mano - attraverso il movimento del mouse - e che pertanto potrebbe essere necessario lo studio dei meccanismi di coordinazione senso-motorio tra occhio e mano al fine di capire in che modo questi movimenti possono rivelare aspetti della *deep semantic*. Un ulteriore supporto in questa direzione potrebbe venire senz'altro dai vari studi nel campo della Computer Vision [3, 2]

[4]Altre caratteristiche fondamentali del Topic Model e che ne suggeriscono l'utilizzo per i nostri scopi sono la valutazione dei documenti come misture di *topic-argomenti* e la capacità, attraverso esso, di poterli generare: si rimanda il lettore ai successivi capitoli di questo lavoro di tesi e la consultazione del lavoro proposto in [70] per apprezzare l'utilizzo in queste accezione del Topic Model nonché per ulteriori spiegazioni in merito alla possibilità di generare documenti attraverso esso.

come anche dall'approccio proposto da Pylyshyn[5] [62] che sembrano essere molto adeguati ed indicati ai nostri scopi. A completamento di quanto detto sinora, studi precedenti hanno mostrato che un utente in fase di *"navigazione"* del web può rivelare sostanzialmente intenzioni logiche e che queste possono essere poi apprese ed utilizzate per il pre-caricamento di ulteriori pagine web relazionate [40]. Un modo allora per rilevare la semantica potrebbe quindi scaturire da un'analisi statistica del percorso di navigazione dell'utente: percorso che alla fine lo ha condotto, attraverso le diverse pagine, a quello che era alla base della sua ricerca e quindi del "*significato*" per come lo abbiamo introdotto anche in questo lavoro di tesi. Risulta così inevitabile associare il concetto di *tempo variante* e quindi di *dynamic semantics* a quello di *emergent semantics*, ed il discorso si complica ulteriormente.

In conclusione, provando a sintetizzare tutti gli spunti di ragionamento introdotti fin qui e proiettandoli in un contesto reale quale ad esempio l'intero World Wide Web, appare evidente che il significato latente sito in un oggetto-documento testuale - nella fattispecie in una pagina web - dovrà per forza di cose emergere dalla sinergia tra un/il contenuto ed un/il contesto e dove però la sua comprensione è tuttora al di là delle capacità di una qualsiasi tecnica di intelligenza artificiale anche e per via del fatto che la gran parte delle sue caratteristiche multimediali ne rendono l'estrazione e la rappresentazione del significato stesso impresa ancora più ardua. Una completa e piena rappresentazione della conoscenza può essere allora ottenuta attraverso la fusione di linguaggi dinamici (deterministici con rappresentazione ben strutturata) provenienti da un approccio generalmente di tipo classico con metodi/linguaggi probabilistici (basati chiaramente su inferenza statistica), provenienti da un approccio puramente probability-oriented in grado di affrontare i processi rumorosi per come sono stati descritti in precedenza. Dopo aver introdotto tutto lo scenario di riferimento e le difficoltà presenti attualmente in questi contesti, l'idea che è alla basta di questo lavoro di tesi parte per forza di cose dal livello più basso della conoscenza, o meglio, dalla rappresentazione della *light semantic* come base di supporto senza la quale i diversi tipi di ragionamento e/o inferenza risulterebbero impensabili.

[5]Secondo questo studio il metodo per situare la visione nel mondo può avvenire differenziando in tre modi in cui un agente artificiale che vuole interagire con l'esterno può rappresentare il suo mondo: *formalismo logico, rappresentazione con indessicali, internal world model.*

Capitolo 2

Linguaggi per la rappresentazione della Conoscenza

Per soddisfare una buona varietà di bisogni nell'ambito della modellizzazione dell'informazione, di sviluppo software ed integrazione, nonché nella gestione e nel riutilizzo della conoscenza, vari gruppi afferenti ai settori governativi ed accademici hanno sviluppato e distribuito diversi modelli condivisibili e riutilizzabili conosciuti come *ontologie*[1].

La definizione più comunemente citata di ontologia è quella che considera queste come *una specializzazione formale ed esplicita di una concettualizzazione condivisa* [32]. Per formale si intende che il significato è codificato in un "*linguaggio formale*" le cui proprietà sono ben comprese e fissate. Nella pratica, e nella maggior parte dei contesti applicativi, questo si traduce nell'utilizzo di linguaggi logic-based emersi naturalmente dalla comunità della Knowledge Representation e più nello specifico della Intelligenza Artificiale. *Formalità* è quindi un modo importante per provare ad eliminare l'ambiguità che è presente prevalentemente in linguaggi naturali ed in altre notazioni informali. In questo modo, grazie all'estensione della logica ai ragionamenti automatici, è possibile estrarre così nuova conoscenza e fare predizione sul mondo osservato.

[1] Gran parte del materiale di questa sezione è tratto da [8] e [49].

Le ontologie indicano quindi diversi tipi di oggetti in una determinata disciplina o contesto come ad esempio [*ala*, *oggetto fisico*, *fili*] che vengono rappresentati come classi (a volte chiamate concetti) ed in genere organizzati attraverso tassonomie di sottoclassi. Ogni classe è generalmente associata a varie proprietà (slot o roles) che descrivono le caratteristiche e gli attributi, nonché le varie restrizioni su di essi (a volte chiamati *facets* o regole di restrizione). In figura 2.1 si riporta un continuum di strutture, schemi e linguaggi, formali e non, per la rappresentazione della conoscenza in funzione della loro complessità. All'estremo sinistro troviamo le rappresentazioni più semplici che possono essere costituite dai soli termini, che in linea di principio sono associabili proprio alla *light semantics* secondo lo schema di rappresentazione a livelli introdotto nel capitolo precedente. Dall'altro lato, invece, troviamo quelle teorie estremamente rigorose tipiche dell'ingegneria della conoscenza e che portano al concetto più alto e complesso di ontologia formalizzata. Muovendoci lungo tale continuum, la quantità di significato specificato ed il grado di formalità aumentano riducendo così anche l'ambiguità del linguaggio ma imponendo grosse restrizioni concettuali e di rappresentazione.[2].

E' evidente altresì che il contributo dell'essere umano è sempre maggiore tanto più è complessa la rappresentazione, infatti non esistono metodi completamente automatici per la generazione di schemi così formali che offrono buone prestazioni. Ritorna così il concetto espresso nel capitolo precedente, secondo il quale la rivelazione della semantica non può essere realizzata associando semplicemente un significato al testo, ma l'utente, è ex-necessitate, il protagonista - *deep semantics*. Il suo contributo è quindi nella modellazione concettuale della catena di significati rappresentati da *Formal Ontologies*, ma che inevitabilmente porta ad una espressione soggettiva del mondo, che dovrebbe evolvere in un contesto collaborativo o almeno condiviso per essere universalmente riconosciuta come tale. D'altronde superato questo gap di modellazione, la potenza espressiva dei linguaggi formali consente di effettuare ragionamenti automatici per compiere inferenza.

In questo lavoro di tesi è stato sviluppato un modello per l'estrazione della parte di semantica, che noi abbiamo chiamato *light*, che è priva di specifiche formalizzazioni e che, se applicato a documenti testuali,

[2]In questo lavoro utilizzeremo il termine *ontologia* fissato sulla parte a sinistra di tale spettro così come varrà discusso dettagliatamente nella sezione 2.1.

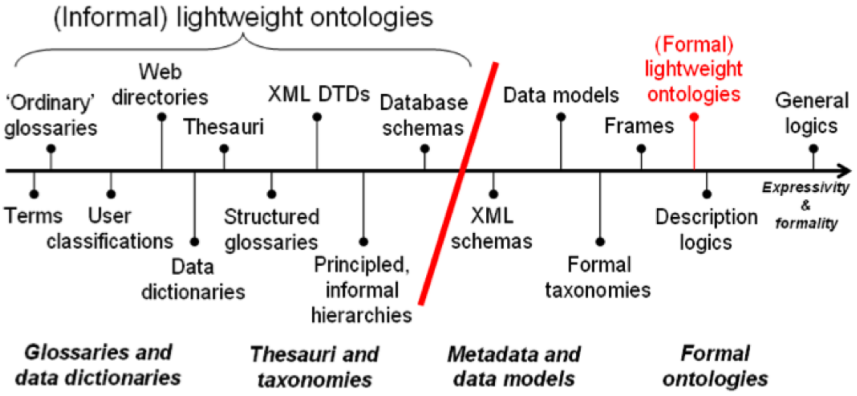


Figura 2.1 Gradi di formalizzazione delle informazioni.

è collocabile chiaramente più nell'ambito della Text Retrieval che del Web Semantico. In questo contesto, i modelli esistenti mostrano scarse prestazioni, ovvero bassa precisione/qualità nel recupero di documenti di testo che rispondono ad una specifica richiesta utente (*query*). In realtà, è ben documentato che la lunghezza tipica di una query in questi sistemi di information retrieval è piuttosto breve - di solito tra due e tre parole [42, 43] - che produce, proprio a causa della ambiguità del linguaggio (polisemia, ec), risultati/documenti recuperati irrilevanti data la query. Questa esigenza ha portato ad una intensa attività di ricerca al fine di individuare diverse soluzioni manuali, interattive e/o automatiche di *query expansion*, ovvero di aggiunta di conoscenza alla richiesta iniziale dell'utente [21, 52].

L'idea di fondo sulla quale si basano più o meno queste tecniche è che può essere sufficiente specificare meglio il *"significato/senso"* di ciò che l'utente stesso ha in mente quando ad esempio pone in essere una ricerca, o in altri termini, si prova a definir meglio *"il concetto principale"*(o insieme di concetti) a cui esso è interessato, così da estendere la query di partenza con una conoscenza a-priori.

In questo caso è allora possibile distinguere due tipi di conoscenza: una *esogena*, rappresentata tra le altre cose da ontologie, WordNet, ecc., e l'altra *endogena*, cioè quella che è possibile estrarre dalla sola elaborazione dei documenti [6, 58].

In questo lavoro ci siamo concentrati sulle tecniche che utilizzano conoscenza endogena, ancora una volta perché siamo interessati ai

modelli che considerano la parte *light* della semantica. Questi metodi considerano l'estrazione della semantica (*senso*) da un insieme di documenti (chiamato *training set* e da noi indicato di seguito con $\mathbb{S}$) in base esclusivamente allo studio dei diversi pattern che si presentano all'interno dei documenti stessi. Sono in grado cioè di selezionare una lista di termini - per esempio in base alla loro occorrenza in un insieme di documenti ben identificato - che può essere aggiunta alla query iniziale per fornire così una maggiore specializzazione dell'*intenzione utente* [21]. Senza soffermarsi su quale tra tutte risulta essere la miglior tecnica di query expansion, dimostreremo allora che, indipendentemente da come è stata estratta la lista di termini, le prestazioni di ricerca possono essere sicuramente migliorate se si usa come *vector of features* una rappresentazione più complessa invece di un semplice elenco di parole.

Una volta chiarito il contesto teorico e sperimentale in cui si colloca questo studio, ci preme precisare che proporremo un metodo per l'estrazione automatica di un *vector of features*, chiamato di seguito mixed *Graph of Terms*[3] o *informal Lightweight Ontologies*, da un insieme di documenti $\mathbb{S}$ attraverso un metodo *term extraction* basato su una tecnica supervisionata di *term clustering* [66], pesata attraverso il *Probabilistic Topic Model* [70] che è una versione smooth della *Latent Dirichlet Allocation* [7]. Il grafo gerarchico risultante, come illustrato in figura 2.2, sarà così composto da un sottografo diretto ed uno a-diretto: il livello più basso, vale a dire il livello delle parole o *word level* si otterrà raggruppando i termini a due a due con un alto grado di "*legame semantico*" producendo così diverse gruppi di *parole* connesse ai loro centroidi (directed edges). Per quel che riguarda invece il secondo livello, o *livello concettuale*, anche esso si può ottenere attraverso la stessa procedura deducendo quindi il legame semantico tra i centroidi dei vari gruppi ovvero, come sarà spiegato meglio nel capitolo successivo, tra *concetti* (si parla in questo caso di undirected edges)[4].

[3] Grafo di termini o concetti: di seguito in questo lavoro si cercherà di risolvere l'ambiguità tra questi due sostantivi

[4] L'idea generale è supportato da lavori precedenti [5, 56] che hanno confermato il potenziale dei metodi di clustering supervisionato per l'estrazione del termine anche in caso di query expansion [9, 48].

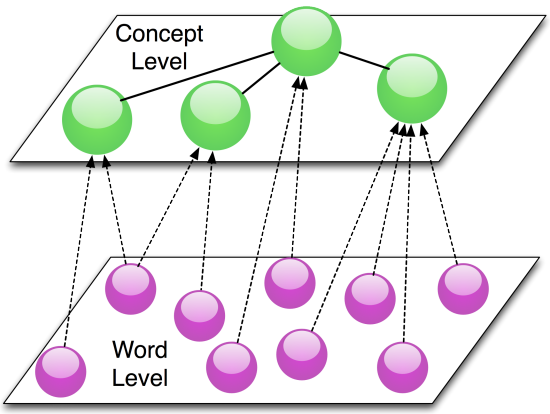


Figura 2.2 Grafo gerarchico a due livelli.

2.1 Un approccio probabilistico per la costruzione di informal Lightweight Ontologies

In letteratura sono stati proposti diversi metodi manuali piuttosto che semi-automatici o automatici [50, 11], per la costruzione di ontologie. Tra gli approcci semi-automatici e automatici presenti, possiamo poi distinguere quelli basati su tecniche di Machine Learning e quelli basati su teorie proprie dell'Intelligenza Artificiale (IA). Tuttavia, la maggior parte dei metodi esistenti, si basa sul concetto di ontologia secondo ciò che è comunemente riconosciuto in informatica e introdotto precedentemente, ovvero un grafo orientato i cui nodi indicano i *concetti* e gli archi le *relazioni* tra di essi [32]. La comunità dell'IA ritiene inoltre che l'ossatura di una ontologia è più formalmente una tassonomia in cui i rapporti sono esclusivamente del tipo *è-un*, mentre la struttura del grafo restante fornisce poi solo informazioni ausiliarie relative al dominio modellato (in tal caso può comprendere le relazioni come *parte-di*, *si trova in*, *è-padre-di*, ecc. [33]).

Qualunque sia però la definizione di ontologia, possiamo affermare che i sistemi che hanno l'obbiettivo di organizzare la conoscenza (ad esempio sistemi di classificazione, thesauri, ecc.) dovrebbero essere considerati fondamentalmente come sistemi che lavorano su *concetti* e sulle loro *relazioni semantiche*. Pertanto, quando vengono mostrate le ontologie come grafi, i loro nodi sono spesso indicati con i corri-

spondenti *nomi-concetto* del linguaggio naturale. Anche se oggi non c'è un consenso completo su cosa si intende per *concetto* [1], generalmente si rappresenta esso lessicalmente attraverso un termine, alimentando erroneamente la convinzione che un *"concetto"* ed un *"termine"* rappresentino la stessa informazione.

Infatti, un termine è un segno fonico e/o grafico - una parola, un gruppo di parole, una parola composta o una locuzione, una forma abbreviata - oppure un simbolo che permette di esprimere un concetto speciale relativo a oggetti concreti o astratti (per esempio <periscopio>, <motore a scoppio>, <computer>,ecc.) definibili in modo univoco all'interno di una determinata disciplina. Il rapporto tra termine *"designazione di un oggetto mediante un'unità linguistica*[5]*"*, concetto *"unità di pensiero costituita per astrazione sulla base delle proprietà comuni ad un insieme di oggetti*[6]*"*ed oggetto *"elemento della realtà che può essere percepito o immaginato*[7]*"* è rappresentabile attraverso il cosiddetto triangolo semiotico[8] di figura 2.3.

Semplificando al massimo, la mente umana raggruppa gli oggetti (concreti e astratti) in base alle proprietà che condividono e assegna loro un'immagine mentale, il concetto, che a sua volta viene rappresentato da un segno (parola, simbolo, icona, ecc.), nel nostro caso il termine. Nel triangolo semiotico il collegamento tra il segno e l'oggetto viene espresso da una linea tratteggiata proprio perché le parole non denotano direttamente l'oggetto ma sono una convenzione, un'etichetta, come ben dimostra l'esempio del fiore. L'ovvia conclusione è che nel lavoro terminologico è molto importante un approccio orientato al concetto, specialmente in ambito multilingue. Chiaramente un termine può riferirsi a concetti generali <duomo> oppure a concetti individuali <duomo di Milano>. I concetti comprendono le caratteristiche, cioè le qualità peculiari di singoli oggetti concreti o astratti ben determinati <benzina diesel>, <fondo di investimento> oppure di intere classi di oggetti <macchine fotografiche>. L'insieme delle caratteristiche serve a determinare e a comprendere un concetto, e permette di collocarlo in un sistema concettuale, vale a dire in *"un insieme strutturato di con-*

[5]ISO 1087: Terminologia - Vocabolario. Geneva-1992, 2000, 5.3.1.3
[6]ISO 1087: Terminologia - Vocabolario. Geneva-1992, 2000, 3.1
[7]ISO 1087: Terminologia - Vocabolario. Geneva-1992, 2000, 2.1
[8]il segno è un sistema, composto da un segnale, una referenza e un referente, che rinvia ad un contenuto. La semiotica studia proprio la capacità del segno di dare la possibilità a chi interpreta di comprenderne il contenuto

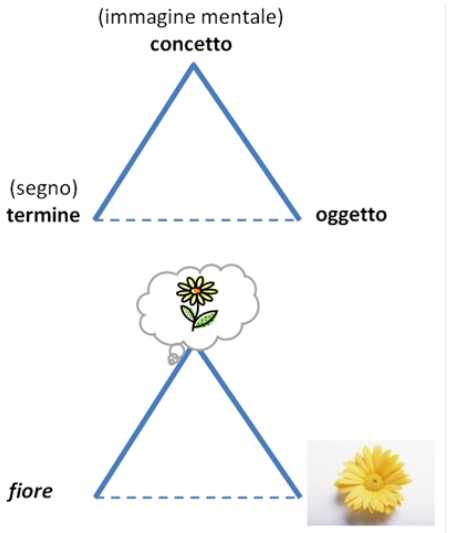


Figura 2.3 Triangolo semiotico.

cetti costruito sulla base delle relazioni stabilite tra questi concetti e nel quale ogni concetto è determinato dalla posizione occupata in questo insieme[9] *"*.
Partendo da queste considerazioni sugli aspetti principali della definizione della conoscenza nonché dei concetti, sorgono quindi alcune domande:

1. *Come dovrebbe essere formalmente la definizione dei concetti e delle loro relazioni in modo da essere universalmente condivisa ed accettata?*

2. *Quanto deve essere l'intervento umano per costruire una conoscenza universalmente condivisa?*

La formalità di una ontologia può essere intesa in due modi diversi, ovvero il grado e l'espressività del linguaggio usato per descriverlo piuttosto che la qualità delle fonti di informazioni utilizzate per costruire tale conoscenza. Sulla base di questa ulteriori considerazione possiamo ritornare sulla figura 2.1 che descrive il continuum di *"tipi di ontologie"* (presente in [72]), partendo banalmente da *termini* e *web directory* (ontologia chiamata anche *lightweight*), e continuando fino a teorie logiche

[9]ISO 1087: Terminologia - Vocabolario. Geneva-1992, 2000, 3.10

rigorosamente formalizzate. La maggior parte delle indicazioni convergono comunque sul fatto che una ontologia dovrebbe essere comunque definita in un linguaggio formale, che in pratica significa un linguaggio basato sulla logica adatta ad un ragionamento automatico[10], e dovrebbe essere costruita attraverso una conoscenza formale, cioè pensando al coinvolgimento di diversi esperti. Tuttavia, l'informalità di una ontologia può essere legata alla natura delle informazioni che alimentano la definizione della conoscenza: è stata estratta l'informazione in modo automatico o semi-automatico da fonti non strutturate? Se è così, allora questa informazione contribuisce a formare una specie di conoscenza informale. Di conseguenza possiamo dire che una ontologia è ben determinata se si definiscono due aspetti:

1. la formalità, così come l'espressività, del linguaggio utilizzato per descrivere e rappresentare la conoscenza (che risponde alla prima domanda vista in precedenza);

2. la definizione della formalità delle fonti di informazione da cui viene estratta la conoscenza (che risponde alla seconda domanda).

A nostro parere, la definizione più ampiamente condivisa di ontologia, che considera la formalità per entrambi questi aspetti, non sembra essere adatta a descrivere la conoscenza informale che possiamo, ad esempio, dedurre automaticamente da documenti testuali. Al contrario, la definizione di una sorta di *informal Lightweight Ontology* sembra fornire una struttura più flessibile e facilmente applicabile ad una descrizione informale della conoscenza all'interno di uno specifico contesto [29, 35, 51]. La nostra idea è quindi di introdurre una rappresentazione chiamata *mixed Graph of Terms (mGT)* (si veda la figura 2.2), e un metodo in grado di apprendere automaticamente tale rappresentazione da documenti di testo: si dimostrerà quindi che un mGT è una sorta di rappresentazione informale e poiché estratta/appresa automaticamente da documenti, può essere considerata come una *informal Lightweight Ontology*[11], ovvero rappresentazione leggera della conoscenza.

[10] Si noti che in questo l'affermazione non vuol dire che un tale linguaggio non può essere probabilistico (vedere quello che è stato dimostrato in [44])

[11] Si fa presente che nel seguito della trattazione, potrà capitare di riferirci alle *informal Lightweight Ontology* usando impropriamente la sola parola *ontologia* ma da questo punto in poi appare chiara la differenza.

2.2 Linguaggi formali e probabilistici per la rappresentazione della conoscenza

E' necessario sottolineare nuovamente il ruolo determinate dell'informazione di tipo probabilistica all'interno della KR anche alla luce di tutti i ragionamenti effettuati sul rumore sempre presente in tutti i processi comunicativi (così come riportato nella parte introduttiva di questo lavoro) e risulta semplice ed intuitivo indicare come contesto-prova proprio il Web e nella fattispecie quello Semantico. Una forte motivazione in questo senso, scaturisce dalla consapevolezza che le tecnologie per il Semantic Web potrebbero trarre grande beneficio da una più stretta integrazione con tecniche proprie del machine learning e tecniche di reperimento delle informazioni[12]. Le criticità di questi approcci nascono però da valutazioni soggettive di eventi e quindi risulta chiaro che in ambienti vasti ed aperti come la rete non si cercherà di utilizzare valutazioni soggettive di probabilità ma di prevedere meccanismi statistici di rappresentazione e di reperimenti delle informazioni disponibili sul web stesso. Esistono quindi alcune problematiche a carattere generale che si riscoprono in questo settore e che danno poi spunto ad una valutazione pragmatica di tale aspetti:

I *Rappresentazione delle informazioni intrinsecamente nascoste*: non tutta l'informazione che deve essere portata in conto nel Semantic Web è data in termini di dichiarazioni ben definite - l'assenza di informazioni statistiche in grado di fornire spunti per valutazioni sui dati condivisi in rete quali possono essere i diversi legami tra la percentuale di persone di una certa età ed una certa malattia cronica ne sono un esempio pratico. Ci sono poi molte altre situazioni in cui l'uso di questi dati può essere manipolato anche per migliorare il comportamento di agenti intelligenti come potrebbe essere un sistema di suggerimenti in grado di fornire all'utente solamente determinate informazioni in base alla propria di età, stato sociale, ecc.

Il *Ontology Learning*: la creazione manuale di ontologie è stata identificata come uno dei principali ostacoli per il Web Semantico. Al fine di superare questo problema, la letteratura sta valutando diversi metodi di apprendimento automatico da testi anche se

[12]Tecniche per lo più basate chiaramente su modelli probabilistici.

l'approccio attualmente più gettonato è una combinazione di Natural Language Processing e tecniche di text mining [49] - compiti tipici in questo caso sono comunque l'individuazione di sinonimi e dei rapporti classe-sottoclasse utilizzando spesso anche tecniche specifiche di clustering. In entrambi i campi, il risultato del processo di estrazione può essere interpretato in termini di giudizio probabilistico nella correttezza dei legami appresi.

III *Classificazione di documenti*: può essere visto come un particolare caso di Ontology Learning chiamato nello specifico Ontology Population. Una parte importante delle informazioni sul web è presente in termini di documenti (pagine Web, documenti PDF, ecc.) ed un modo comune di collegare tali documenti alla conoscenza codificata attraverso ontologie è quello di assegnare i singoli documenti ad uno o più topic che ne rappresentano il contenuto. Per questo problema sono state adoperate diverse tecniche di machine learning [67] tra cui la più usata è incentrata su classificatori di tipo naive Bayes in grado di stimare la probabilità di appartenenza di un documento ad un argomento sulla base del verificarsi o meno di condizioni legate alle occorrenze di termini in documenti campione.

IV *Ontology Matching*: fonti diverse utilizzano spesso diverse ontologie per organizzare le proprie informazioni anche se riferite poi agli stessi argomenti. Al fine di accedere agevolmente ai dati su queste diverse fonti, devono essere determinate le varie corrispondenze semantiche tra le rispettive classi di ontologie e poi codificati nel successivo mapping. Bisogna dire che proprio di recente sono state proposte una serie di indicazioni utili a determinare automaticamente i mapping di questo tipo [23]. Anche per questa problematica i maggiori successi si ottengono utilizzano tecniche di machine learning in grado di calcolare la probabilità che due classi rappresentino le stesse informazioni.

V *Uso di Ontology Mapping per l'integrazione delle informazioni*: generalmente l'utilizzo di mappature per l'estrazione delle informazioni è costituito da una fase di pre-processing durante la quale tutte le associazioni che hanno un valore di "fiducia" superiore ad una determinata soglia sono considerate deterministicamente vere mentre tutte le altre deterministicamente false. Tuttavia,

vi sono prove che questo tipo di soluzione è soggetta ad errori soprattutto quando i mapping sono composti da diverse tipologie di ontologie.

Per quel che concerne l'approccio di tipo probabilistico, avendo capito che per rappresentare l'incertezza contenuta intrinsecamente nel processo rumoroso della comunicazione l'unico framework realistico può essere solo di questo tipo, di seguito verranno evidenziati e caratterizzati (anche se in modo sintetico) i vari modelli ed estensioni dei linguaggi presenti in letteratura in grado, almeno su carta, di rispondere ai nostri problemi mettendo così in luce di volta in volta le criticità che ne rendono impercorribile la strada. Per far ciò e per valutare in modo eterogeneo i vari sistemi, che risultano generalmente e chiaramente molto diversi tra loro, sarà utile introdurre una sorta di parametri di confronto quanto più oggettivi possibili: si cercherà in questo modo di mettere in luce soprattutto il rapporto costo-prestazione di ognuno di essi per indicare lo sforzo da sostenere rispetto al beneficio dei risultati attesi. Tali parametri possono essere quindi:

- Espressività ed efficienza (E);
- Motivazione ed inferenze – sistemi di ragionamento (M);
- Applicabilità per l'integrazione delle informazioni (A);

Nella trattazione che segue si ometteranno i dettagli dei vari linguaggi classici del web per non appesantire ulteriormente la lettura: si confida cioè sulla ormai consolidata conoscenza comune in riferimento a tale settore.

2.2.1 Estensione del RDF

RDF può essere considerato il linguaggio più utilizzato nell'ambito del Semantic Web in quanto fornisce la base sintattica per gli altri linguaggi "operativi". Una prova del successo di tale standard è l'enorme quantità di software scritto per l'elaborazione dei dati di questo tipo. Allo stesso modo, anche in termini di popolarità, sono stati proposti alcuni approcci come combinazione di probabilità e strutture RDF portando in conto e concentrandosi più su questioni di rappresentazione pura ovvero di una "nuova" logica probabilistica.

Rappresentazione dell'informazione probabilistica in RDF

A titolo di esempio viene riportata la proposta di Fukushige [25] come una miscelanza di questa doppia anima statica-dinamica e la successiva creazione di un vocabolario RDF per una rappresentazione dei dati attraverso reti bayesiane.

(E): il vocabolario adottato può essere considerato sostanzialmente come un formato di interscambio sintattico ed in grado di rappresentare solo reti di tipo bayesiano: anche in questo caso, come accade in modo specifico per le reti bayesiane, non è possibile rappresentare descrizioni probabilistiche cicliche: grave problema soprattutto in ambienti aperti e non strutturati quale il web.

(M): ad oggi non è stato implementato alcun tipo di ragionatore ma tuttavia, dopo aver realizzato un opportuno parser per questo vocabolario, in linea di principio potrebbe funzionare qualsiasi strumento di inferenza che lavora su reti Bayesiane.

(A): il vocabolario può essere utilizzato per rappresentare i mapping tra ontologie in modo simile a quanto fatto con i Bayesian Description Logic Programs [61]. Un enorme svantaggio in questo caso è che le reti bayesiane non sono adeguatamente interlacciate con i meta-livelli RDF: il vocabolario per la rappresentazione di reti Bayesiane usa RDF solo per la sintassi senza alcun legame con il modello logico. Pertanto, le ontologie RDF, e per analogia anche quelle generate con OWL (Web Ontology Language), non possono essere adeguatamente integrate con quelle espresse attraverso questo procedimento.

pRDF

A differenza del precedente formalismo che mira solo a fornire un vocabolario per la rappresentazione di reti Bayesiane, *pRDF* è una estensione probabilistica formale del *RDF* con una logica propria al suo interno [71].

(E): è una estensione probabilistica di un sottoinsieme *RDF* basato su una coppia (S, I) con S schema definito come set finito di quadruple che estendono lo schema classico *RDF* o triplette non probabilistiche - formate attraverso legami deterministici tra due oggetti ad esempio $[o1, sottoclasse/range/dominio/rdf_s, o2]$ - e I istanza. Per entrambi i domini dello schema S ed I rispettivamente, vale la ristrettezza nella scelta dei valori da assegnare, ovvero possono essere selezionati ben

pochi punti nello spazio delle proprietà *RDF* per tali oggetti. Inoltre, le istanze *pRDF* devono essere acicliche e quindi si prestano ad essere realizzate solo in ambienti piccoli e limitati: ancora una volta il Web non rientra in questo contesto.

(M): il modello semantico in questo caso è basato su una distribuzione $t - normale$ e le query per le istanze *pRDF* sono solamente di tipo atomico, ovvero non possono essere gestite congiunzioni o legami. Una query è definita da una quadrupla (i, p, S, P) dove i può essere una istanza, p una proprietà, S un set di istanze i collegate attraverso p e P una distribuzione di probabilità per questo sequel di assiomi e dove, al massimo, solo uno degli elementi della quadrupla può essere una variabile. Nessun sistema informatico è purtroppo però ancora in grado di rispondere per schemi di tipo *pRDF* a query strutturate in questo modo, mentre esistono alcuni motori inferenziali (anche se solo prototipi) in grado di lavorare su istanze *pRDF*.

(A): questo formalismo può essere utilizzato per l'integrazione di informazioni con le varie mappature: è possibile risalire ai legami, e quindi alle rispettive analogie, sia tra classi che tra istanze afferenti a due ontologie diverse. Tuttavia, l'incertezza allegata ad ogni mappatura può essere utilizzata anche per ragionamenti intelligenti e/o per inferenza. A causa della struttura limitata del *RDF*, non solo le mappature, ma anche e soprattutto le ontologie *RDF* risultano avere una espressività molto limitata.

2.2.2 Estensione del OWL

Vista ormai la grossa diffusione del *Web Ontology Language*, del tutto naturale sono una serie di proposte ad estensione del linguaggio stesso, in grado di applicare nozioni legate alla probabilità nel Semantic Web come meccanismo centrale di rappresentazione di conoscenze complesse. Osservando quindi una serie di proposte esistenti in questo contesto, è possibile distinguere sostanzialmente due approcci radicalmente diversi per combinare *OWL* con informazioni probabilistiche:

- Il primo approccio implementa un accoppiamento *light* tra *OWL* e modelli probabilistici. In particolare, si propone quindi di utilizzare *OWL* esclusivamente come linguaggio per connettere i modelli probabilistici. Un esempio di questa visione è il lavoro di Yang e Calmet che propongono una ontologia minima *OWL* per

la rappresentazione di variabili casuali e le dipendenze tra le variabili casuali con la corrispondente probabilità condizionate [75]. Attraverso questo procedimento, un utente può esprimere modelli probabilistici che corrispondono a reti Bayesiane come istanze di ontologie *OntoBayes*. La codifica del modello in *OWL* rende possibile un collegamento esplicito tra le variabili casuali ed elementi di un'ontologia, anche se si notano carenze di integrazione almeno sul piano formale. Un'altra soluzione afferente allo stessa tipologia di approccio, chiamato *Pr-OWL*, è proposta invece da Costa e Laskey ed usa le ontologie *OWL* per descrivere modelli probabilistici con la logica del primo ordine [14]. Più in particolare, le ontologie *OWL* modellano reti Bayesiane Multi-Entity [47] basate su distribuzioni di probabilità del primo ordine in maniera modulare. Simile a *OntoBayes*, non vi è alcuna integrazione formale dei due paradigmi di rappresentazione ed *OWL* è usato come meta-livello solo per codificare la struttura generale delle reti Bayesiane Multi-Entity.

- Il secondo approccio mira in realtà ad arricchire le ontologie *OWL* con le informazioni probabilistiche per sviluppare ragionamenti basati sulle incertezze presenti al proprio interno. Secondo questa visione, gli approcci che si rivolgono direttamente ad *OWL* come un linguaggio ontologico sono meno ambiziosi rispetto alla combinazione semantica-logico e probabilistico dei *Description Logic*. Un esempio è il lavoro di Holi and Hyvönen [37] che descrive un framework per rappresentare l'incertezza semplicemente nelle gerarchie di classificazione utilizzando reti Bayesiane. Un altro utilizzo in questa direzione, chiamato *BayesOWL*, è suggerito da Ding et al. [18] che considerano operatori booleani disgiunzione ed equivalenza delle classi *OWL* e presentano la loro idea per la costruzione di una rete bayesiana partendo da espressioni di classi legate da questi costrutti.

Pr-OWL

(E): è sostanzialmente una ontologia *OWL* che descrive reti Bayesiane Multi-Entity - *MEBNs*. Più nel dettaglio, *OWL* è principalmente utilizzato come base per un plugin di *Protégé* in grado di modellare e rappresentare *MEBNs*, ovvero gestire i vari collegamenti tra ontologie

all'interno di diversi domini. D'altra parte, le *MEBNs* possono essere tradotte in reti Bayesiane e quindi, estendendo il concetto, *Pr-OWL* potrebbe essere utilizzato semplicemente per collegare ontologie *OWL* alle reti bayesiane attraverso il formalismo delle *MEBNs*. La valutazione nei termini dell'espressività del *Pr-OWL* si riduce quindi verso il basso per l'analisi della espressività fornita dal *MEBNs* strettamente connesso al modello reale in grado di rappresentare l'incertezza in questi tipi di approccio. Secondo gli autori, le *MEBNs* sono in grado di rappresentare e sviluppare un ragionamento probabilistico con frasi espresse in logica del primo ordine definendo solo alcune restrizioni per l'uso ad esempio dei quantificatori. Le *MEBNs* sono anche in grado di specificare le variabili casuali che rappresentano i vari concetti ed organizzarli nei cosiddetti frammenti che descrivono un certo aspetto del mondo. Ogni frammento, codificato come parte di una rete bayesiana, definisce la distribuzione congiunta sulle variabili aleatorie in termini di probabilità condizionate: formule logiche sono modellate da frammenti speciali che codificano la semantica degli operatori booleani, dei quantificatori e delle diverse istanze. E' comunque molto difficile stabilire se le *MEBNs* sono sufficientemente espressive e capaci di catturare le informazioni probabilistiche su ontologie *OWL*. In linea di principio quindi dovrebbe essere possibile tradurre ogni ontologia *OWL* in logica del primo ordine e di assegnare delle probabilità al modello risultante attraverso l'associazione ad una *MEBNs*. Finora comunque non è stato verificato se le restrizioni su l'uso dei quantificatori nelle *MEBNs* si ripercuotono e limitano la rappresentazione delle ontologie stesse. In generale, il linguaggio deve risultare comunque molto espressivo per rappresentare il mapping tra termini provenienti da diverse ontologie: deve permettere cioè di combinare termini di ontologie diverse utilizzando i semplici operatori logici così come la probabilità condizionata di uno termine dato l'altro. È meno chiaro se questo tipo di mapping può essere integrato in modo "semanticamente coerente" con definizioni che vanno al di là di un semplice riferimento alle parti delle ontologie.

(M): alcuni tipi di ragionamento all'interno delle *MEBNs* vengono impostati attraverso la costruzione di una rete Bayesiana da istanze dei frammenti. All'interno di ogni frammento, viene quindi creata una sottorete, giocoforza di ulteriori frammenti, che include le variabili casuali e le probabilità condizionate per tutti gli oggetti in ingresso sulla base del modello specificato: si fa notare che gli effettivi valori della

probabilità condizionata dipendono fortemente dal numero di oggetti forniti in input. Il problema di fondo delle *MEBNs* per quanto concerne l'efficienza è però senza dubbio la complessità del linguaggio logico supportato. In particolare, questo ha un impatto notevole sulla dimensione della rete specifica creata e come essa rappresenti le informazioni probabilistiche su tutte le istanze contemporaneamente, invece di rivalutare più volte una rete standard. Tuttavia, nelle applicazioni di Information Retrival spesso si assume che i vari oggetti sono indipendenti l'uno dall'altro e non devono essere trattati in parallelo. Anche se la creazione di tipo bottom-up di questa rete assicura che venga costruita solo la parte che realmente è necessaria per rispondere alla query, è possibile ritrovare ancora dimensioni infinite del problema. Sarebbe auspicabile e ovviamente interessante individuare solo sottoinsiemi limitati di *MEBNs* che corrispondono a più frammenti trattabili attraverso la logica del primo ordine.

(A): come accade con il formalismo di Fukushige [25] indicato precedentemente, il vocabolario può essere usato per rappresentare il mapping tra ontologie nel formalismo *MEBNs*. Tuttavia considerando che *Pr-OWL* non riesce a fornire una corretta integrazione del formalismo *MEBNs* e la logica base di *OWL* sui meta-livelli, anche in questo caso le ontologie *OWL* non possono essere integrate propriamente con la mapping espressa attraverso il suddetto vocabolario. Più in particolare, visto che non risulta formalizzata la connessione tra una dichiarazione in *Pr-OWL* e una dichiarazione in *OWL*, non è chiaro in che modo sia possibile eseguire l'integrazione di ontologie che contengono dichiarazioni con entrambi i formalismi.

BayesOWL

(E): può essere visto in qualche modo come estensione del lavoro di Holi and Hyvönen [37] e riguarda sostanzialmente un approccio probabilistico per la rappresentazione delle informazioni, attraverso ontologie *OWL*, in riferimento all'appartenenza o meno ad una classe. In entrambi i lavori risulta possibile la rappresentazione delle sovrapposizioni tra classi in termini di probabilità condizionata nella forma $P(C|D)$, dove C e D sono i nomi delle classi e P riguarda evidentemente la probabilità che una istanza membro di D sia anche afferente a C. Tra le caratteristiche principali di *BayesOWL*, oltre a supportare le gerarchie, è possibile ritrovare definizioni di classi come:

- Equivalenza: $C(x) \leftrightarrow D(x)$
- Complementarietà: $C(x) \leftrightarrow \neg D(x)$
- Disgiunzione: $C(x) \rightarrow \neg D(x)$
- Intersezione: $C(x) \leftrightarrow D(x) \wedge E(x)$
- Unione: $C(x) \leftrightarrow D(x) \vee E(x)$

BayesOWL è in realtà un'estensione probabilistica della logica proposizionale piuttosto che dei più espressivi *Description Logic* (si veda di seguito, DL). Questa restrizione risulta abbastanza forte ed implica l'impossibilità a rappresentare i dati sulle diverse relazioni eccezion fatta per i legame di sussunzione: l'applicabilità di questa soluzione è così limitata a scenari in cui l'interesse riguarda solo la classificazione degli oggetti e non le relazione che intercorrono tra di essi. Come diretta conseguenza si può affermare che tale approccio non è adatto a sostenere ragionamenti su informazioni strutturate ovvero motori inferenziali legati direttamente ai dati che, nell'ampio contesto del web semantico ade esempio, svolgono sicuramente un ruolo determinante.

(M): i task di ragionamento associati alle *BayesOWL* sono forniti in termini di qualificazione di un oggetto e per determinarne le probabilità di appartenenza a tutte le classi dell'ontologia di riferimento. Come in *Pr-OWL*, le probabilità condizionate dei vari nodi della rete sono predefinite all'interno di una tabella e vengono utilizzate per rappresentare i diversi operatori booleani. I valori presenti nelle tabelle vengono estratti usando un metodo di fitting proporzionale-iterativo, ovvero una tecnica statistica particolare che seleziona una distribuzione di probabilità che meglio si adatta alle stesse probabilità condizionate presenti all'interno della rete. Si fa notare che questo approccio risulta abbastanza diverso dagli altri visto che l'inferenza non è guidata da una query specifica ma può essere molto vantaggiosa se distribuita poi sulle diverse query legate ai vari aspetti del modello stesso. Di contro è necessario valutare la complessità generata inutilmente quando ad esempio si è interessati solo ad aspetti molto specifici, come può essere il metodo per calcolare le varie probabilità, che risultano non avere legami diretti espliciti con le variabili in gioco. Nonostante ciò, l'uso delle reti Bayesiane presenti in molti lavori risulta essere relativamente efficiente soprattutto e chiaramente nel caso di reasoner probabilistici.

Un'altra caratteristica speciale di *BayesOWL* è quella di permettere e prevedere nella procedura di inferenza [57] il mapping probabilistico fra diverse ontologie: mapping rappresentati in termini di dichiarazioni di probabilità condizionata e che includono concetti provenienti da diverse ontologie. Come per molti standard, le probabilità condizionate possono essere apprese attraverso i molteplici metodi statistici ben noti in letteratura. In sintesi comunque l'approccio è ben adattato per le applicazioni che utilizzano classificazioni piuttosto semplici di elementi di informazione, quali ad esempio documenti, che sono classificati secondo una topic hierarchy: non appena le applicazioni richiedono maggiori informazioni strutturali come i metadati caratteristici del documento, l'approccio raggiunge i limiti in termini di capacità, o meglio di incapacità, nel rappresentare i dati sulle diverse relazioni.

(A): questo formalismo prevede un'integrazione tra reti di tipo bayesiano e *OWL* e quindi può essere usato sia per esprimere il mapping tra ontologie *OWL* che integrare le informazioni distribuite sulle varie ontologie. Un grave svantaggio però di questo approccio è legato al solo supporto delle definizioni di classi, visto che né il mapping né le ontologie possono contenerne le istanze. Inoltre, l'espressività a livello di schema è molto basso e quindi solo un sottoinsieme molto limitato di *OWL* può essere utilizzato per esprimere le ontologie oggetto del mapping.

2.2.3 Estensione dei Description Logic

Esistono una serie di approcci in letteratura per estendere i *Description Logics* con le informazioni di tipo probabilistico. Uno dei primi a proporre un riferimento alla nozione di sussunzione (o condizionamento) per la ALC logic è stato Heinsohn [34] seguito pochi anni dopo da Jaeger [41] che ha posto invece attenzione su alcuni problemi generali connessi all'estensione di *T-Boxes* e *A-Boxes* con probabilità oggettive e soggettive: il focus del suo lavoro ha riguardato sostanzialmente un metodo generale per compiere inferenza da intervalli di probabilità legati ai vari assiomi dei *DL*. Più di recente, sulla falsariga tracciata da Jaeger, Giugno and Lukasiewicz hanno proposto una estensione probabilistica per la *SHOQ logic* [28] introdotta come *P-SHOQ(D)*. Uno dei principali vantaggi di questo tipo approccio è sicuramente il trattamento integrato dei concetti e delle istanze utilizzati per modellare l'incertezza sulle relazioni direttamente connesso alle nozioni lessicogra-

fiche probabilistiche scaturenti da motori di inferenza classici. Oltre a questo aspetto importante, è comunque da evidenziare che il motore di inferenza utilizzato in questo caso, a differenza delle soluzioni passate, è decidibili grazie soprattutto alla co-operazione di tecniche standard in ambito *SHOQ(D)* e tecniche di programmazione lineare. Un modo alternativo di combinare *DL* con informazioni probabilistiche è stato proposto anche da Koller et al. [46]. A differenza degli approcci indicati sopra, l'approccio *P-Classic* non è basato su intervalli di probabilità bensì su di una specifica distribuzione di probabilità espressa attraverso una rete bayesiana i cui nodi corrispondono alle espressioni dei concetti nella Classic logic. Le reti bayesiane sono anche usate per connettere logiche meno espressive come ad esempio le *TDL* [76].

P-SHOQ

(E): è fondato sulla *SHOQ(D) logic* e risulta anche molto vicino alla logica che fornisce la semantica in *OWL* con la sola differenza del'uso di regole inverse: in particolare, il linguaggio supporta nello stesso modo anche tipi di dati contemplati in *OWL*. Per la gestione delle conoscenze probabilistiche, la loro rappresentazione si ha nella forma $(C|D)[l,u]$ in cui C e D sono le espressioni dei concetti come in *SHOQ(D)* e invece 1 e u sono rispettivamente la massima e la minima probabilità che un'istanza di D sia anche istanza di C. Con questo schema generale possono essere rappresentati i diversi tipi di conoscenza come:

- La probabilità che C sia subsunto da $D \rightarrow P(C(x)|D(x))$
- La probabilità che un particolare individuo o sia un membro del concetto $C \rightarrow P(C(o))$
- La probabilità che un individuo o sia legato ad una istanza del concetto $C \rightarrow P(R(o,x)|C(x))$
- La probabilità che due individui o ed o' siano relazionati $\rightarrow P(R(o,o'))$

Dal punto di vista della rappresentazione, *P-SHOQ(D)* offre molte possibilità di supportare e risolvere i task visti in precedenza, come ad esempio la sovrapposizione del mapping dei concetti tra diverse ontologie, sfruttando la forma $P(i : C(x)|j : D(x))$ dove C e D sono i concetti espressi nelle ontologie i e j rispettivamente. Per quel che

concerne invece il task di apprendimento delle ontologie, il linguaggio è sufficientemente espressivo per acquisire le informazioni tipiche di un processo di apprendimento, come ad esempio il concetto di gerarchia. Tuttavia, la mancanza di relazioni inverse nel linguaggio non permette di rappresentare le diverse restrizioni di dominio.

(M): i sistemi di ragionamento ed inferenza in *P-SHOQ* si basano principalmente su una funzione μ che mappa ogni istanza del dominio Δ su un numero in $[0, 1]$ tale che la somma dei valori di questa funzione per tutti gli elementi in Δ sia pari ad 1 cosicché la probabilità $P(C)$ del concetto C è quindi definita come la somma di tutti i valori μ delle istanze di C. Sulla base di questa ipotesi sono stati quindi definiti un certo numero di task di ragionamento che possono essere sviluppati utilizzando procedure inferenziali appropriate. Al livello più semplice, le funzioni supportate dal linguaggio sono in grado di determinare se una base di conoscenza è consistente o meno ed utile per calcolare l'upper ed il lower bound della probabilità condizionata $P(C(x)|D(x)) \in [l, u]$ anche se questi bounds sono valutati con logica indipendente: le differenti scelte sono specificate dalle possibili relazioni semantiche indicate tra una qualsiasi coppia di concetti. Sulla base di questo metodo generale per il calcolo di limiti superiori e inferiori, possono essere definite una serie di compiti per l'inferenza che generalizzano i ragionamenti standard fatti per i *Description Logics*. In particolare, l'approccio appena indicato supporta i seguenti compiti:

- *Concept satisfiability*: decidere in particolare se $P(\exists x : C(x)) \in [0, 0]$
- *Concept Subsumption*: dati due concetti C e D, si valuta in 1 ed u la probabilità $P(C(x)|D(x)) \in [l, u]$ data la base di conoscenza;
- *Concept Membership*: dati una istanza o ed un concetto C, si valuta in 1 ed u la probabilità $P(C(o)) \in [l, u]$ data la base di conoscenza;
- *Role Membership*: date due istanze o ed o' e la relazione R, si valuta in 1 ed u la probabilità $P(R(o, o')) \in [l, u]$ data la base di conoscenza;

Questi compiti costituiscono una base adeguata per supportare attività quali ad esempio il recupero probabilistico dei dati attraverso le diverse ontologie. In particolare, possiamo formulare query come concetti di

SHOQ(D) e calcolare la probabilità che alcune istanze sono afferenti alle *"query-concetto"*. Un potenziale problema di questo approccio rispetto al recupero dei dati è però l'uso delle probabilità come base per la classificazione: esso si fonda sostanzialmente su intervalli di probabilità piuttosto che su probabilità esatte e quindi non esiste un ordine totale dei risultati che dovrebbero essere utilizzati. Un altro potenziale problema potrebbe essere la complessità computazionale in quanto non descritto in dettaglio in letteratura: appare però evidente che il ragionamento in *SHOQ(D)* è sensibilmente intrattabile.

(A): i *P-SHOQ(D)* possono essere utilizzati per esprimere e rappresentare tutte le mappature citate in precedenza ma tuttavia le ontologie non sono pensate a contenere e gestire le regole inverse. Inoltre, il linguaggio *RDF* con le rispettive ontologie, la cui semantica non può essere descritta solo attraverso il paradigma dei *Description Logics*, non può di fatto essere integrata vista la non copertura del *P-SHOQ(D)* rispetto alla *Logic Programming* necessaria per descrivere propriamente la semantica *RDF*.

P-Classic

(E): è una estensione probabilistica dei *DL CLASSIC* - diversa da *SHOQ*, le *DL-CLASSIC* è stata progettata per l'efficienza dei motori di inferenza piuttosto che per la forza espressiva. In particolare, *CLASSIC* contiene solo pochi operatori del tipo congiunzione, negazione su concetti primitivi, quantificatori universali, role fillers e range di esistenza (limitazione numerica). Come risultato di tali posizioni, possono essere calcolate in tempo polinomiale valutazioni strutturali basate sul confronto dei concetti, riferite soprattutto a proprietà di sussunzione. Tali proprietà vengono poi integrate attraverso l'estensione probabilistica *P-CLASSIC* e quindi, con utilizzo reti bayesiane, è possibile anche rappresentare le probabilità delle relazioni tra i concetti atomici. Con questo approccio è possibile evidenziare verosimilmente le incertezze espresse soprattutto nel mapping e all'interno dei risultati dell'apprendimento. Le altre caratteristiche del linguaggio indicano che può essere utile anche per rappresentare l'apprendimento dal punto di vista delle ontologie.

(M): il ragionamento nei *P-CLASSIC* è orientato al calcolo della probabilità dell'espressione di un concetto complesso come distribuzione di probabilità congiunta delle classi *"atomiche"* e delle caratte-

ristiche delle varie relazioni. L'algoritmo di inferenza quindi prende in input il concetto e una base di conoscenza, di tipo chiaramente *P-CLASSIC*, e restituisce la probabilità del concetto complesso. Questa probabilità è costruita attraverso un procedimento di tipo bottom-up per la rete bayesiana che poi, in genere, viene utilizzata per dedurre la probabilità che un oggetto arbitrario sia membro di una classe-concetto piuttosto che di un'altra. Questo approccio può essere facilmente impiegato per il recupero di dati ed il fatto che esso si basi su probabilità esatte, piuttosto che intervalli, significa che la probabilità stessa definisce una funzione di ranking naturale per le risposte idonee alle varie query. Si ricorda comunque che il vantaggio principale dell'estensione *P-CLASSIC* sta nel ragionamento relativamente efficiente rispetto ad altri formalismi visto che entrambi gli aspetti, il formalismo logico e quello probabilistico, sono stati scelti proprio con questa idea di base: l'algoritmo per la costruzione della rete bayesiana della classe è definito quindi come una diretta estensione dell'algoritmo di sussunzione della *P-CLASSIC* che è noto per essere di tipo polinomiale. Un ulteriore problema è l'aggiunta dalla necessità di valutare la rete: anche se questo caso è noto in letteratura per avere una complessità esponenziale, tale assunzione è vera solo nel numero massimo dei parents di un nodo ed inoltre il riutilizzo dei risultati per espressioni precalcolate migliora effettivamente il tempo necessario per calcolare la probabilità stessa.

(A): quando è stato sviluppato il *P-CLASSIC*, non si pensava come linguaggio per l'integrazione delle informazioni ma si è inteso esprimere e *"ragionare"* sul grado di sovrapposizione tra i concetti di un'ontologia. In linea di principio *P-CLASSIC* lavora con formalizzazioni probabilistiche delle cosiddette *p-classi*, ognuna delle quali descrive una classe di individui. Ad eccezione dell'espressività di una distribuzione di probabilità per i *role fillers*, le espressioni probabilistiche formalizzano in questo approccio solo i concetti e, come breve valutazione conclusiva, la possibilità di esprimere solo questa distribuzione di probabilità ne limita la capacità di integrazione con altri sistemi.

2.2.4 Problemi rimasti irrisolti

Si conclude questa ultima sezione tirando le fila delle varie soluzioni presenti in letteratura e riportate all'interno di questo capitolo che hanno evidenziato sostanzialmente l'esistenza di due diversi approcci

basati su estensioni probabilistiche a supporto dei vari linguaggi per il Semantic Web ma anche della conoscenza in generale:

I) Il primo approccio riguarda il legame piuttosto "*light*" esistente tra i suddetti linguaggi e i modelli probabilistici: in questi casi i primi vengono usati solo sintatticamente come vocabolario per lo scambio di basi di conoscenza indicate all'interno del modello probabilistico. Le estensioni di questo tipo ritenute abbastanza significative possono essere sintetizzate con tre articoli/soluzioni:

- Fukushige in [25] che propone un vocabolario per codificare *Reti Bayesiane* con RDF;
- Yang and Calmet in [75] che propongono un vocabolario per codificare *Reti Bayesiane* con OWL;
- Laskey and Costa in [47] che propongono un vocabolario per codificare *Reti Bayesiane Multi-Entity* attraverso OWL.

Questi approcci risultano piuttosto insoddisfacenti giacché non considerano la semantica intrinseca del linguaggio ma piuttosto si focalizzano solo su particolari modelli probabilistici, nella fattispecie reti Bayesiane o Multi-Entity, ed attraverso esse provano a fornire un Web Semantico basato solo sul formato di scambio sintattico per i modelli stessi. Così, dei cinque settori/aree connessi al Semantic Web introdotti in precedenza ed all'interno dei quali è necessaria inevitabilmente una valutazione dell'incertezza, vengono in qualche modo soddisfatti solo i bisogni dell'area legata alla *rappresentazione delle informazioni intrinsecamente nascoste*. Bisognerebbe poi anche discutere del fatto che i modelli probabilistici riescano o meno a trarre vantaggio nell'utilizzare solo un vocabolario legato al linguaggio del Semantic Web senza alcuna integrazione formale, ma si eviterà di farlo per non dilungarci troppo e perdere così la linea principale del nostro ragionamento. Si fa notare inoltre che non è stato ancora realizzato alcun motore inferenziale per questi vocabolari, ovvero non esistono ancora strumenti in grado di analizzare i termini estratti dal "*linguaggio del Web*" definito per il particolare modello probabilistico e consegnarli di seguito ad un motore in grado di trattare con essi anche se qualcosa si sta muovendo nell'ambito del *PR-OWL*.

II) Il secondo tipo prevede invece una più stretta integrazione a livello formale tra il linguaggio, o un sottoinsieme di esso, ed un modello probabilistico: in questo scenario rientrano chiaramente anche i formalismi che integrano la programmazione logica con i modelli probabilistici stessi. Estensioni di questo tipo rispettano naturalmente i requisiti per la rappresentazione di informazioni statistiche ed inoltre, poiché l'integrazione è molto più rigorosa sul piano formale, sono anche più appropriate per l'*Ontology Matching* e *Ontology Learning* attraverso metodi bayesiani di machine learning. Anche nell'ambito del *Ontology population* e/o classificazione di documenti, un approccio interessante proposto da Straccia and Troncy [69] utilizza un metodo di apprendimento probabilistico per il mapping tra ontologie *OWL* e *pDatalog rules*: in questo caso si parla di implementazione attraverso un framework chiamato *oMAP* dove le regole *pDatalog* possono essere rappresentate anche con $pOWLLite\text{-}/pOWLLite_{EQ}$. Per completezza espositiva bisogna però ricordare e rimarcare - se a questo punto ce ne fosse ancora bisogno - che la rigida formalità risulta poco efficiente dal punto di vista della natura delle fonti e quindi della conoscenza endogena, ovvero la domanda che nasce spontanea è che si può pensare di estrarre tout-court solo con queste tecniche le varie rappresentazioni della conoscenza e/o del significato solo da semplici testi presenti sul Web?

Da questa piccola analisi è anche possibile sottolineare che nessuna delle precedenti estensioni menzionate sono in grado di affrontare le rappresentazioni di tipo ciclico: si ritiene questo un inconveniente oggettivamente grave vista la natura aperta e libera di molti contesti reali e anche dello stesso Web Semantico così come della conoscenza in generale. Se le varie rappresentazioni - ontologie, i logic programs e mapping, ecc. - sono considerate nel loro insieme, le descrizioni di tipo ciclico risultano molto probabili solo in piccoli mondi giocattolo cosicché da questa considerazione è necessario partire per arrivare ad una soluzione diversa del problema. Ed ancora, strumenti di ragionamento e/o motori inferenziali in generale non sono previsti per i diversi linguaggi ma solo per i relativi formalismi logici: quando ci sono strumenti di ragionamento, sono specializzati per il singolo linguaggio e ne sostengono solo una parte di esso così come nel caso di *pRDF*.

Dopo tali analisi e considerazioni, nel prossimo capitolo proveremo a dare allora una precisa formalizzazione della nostra linea di ragionamento nonché della nostra soluzione in grado di rappresentare il significato cercando di mettere in luce la potenza di tale idea strettamente connessa alla natura informale delle fonti.

Capitolo 3

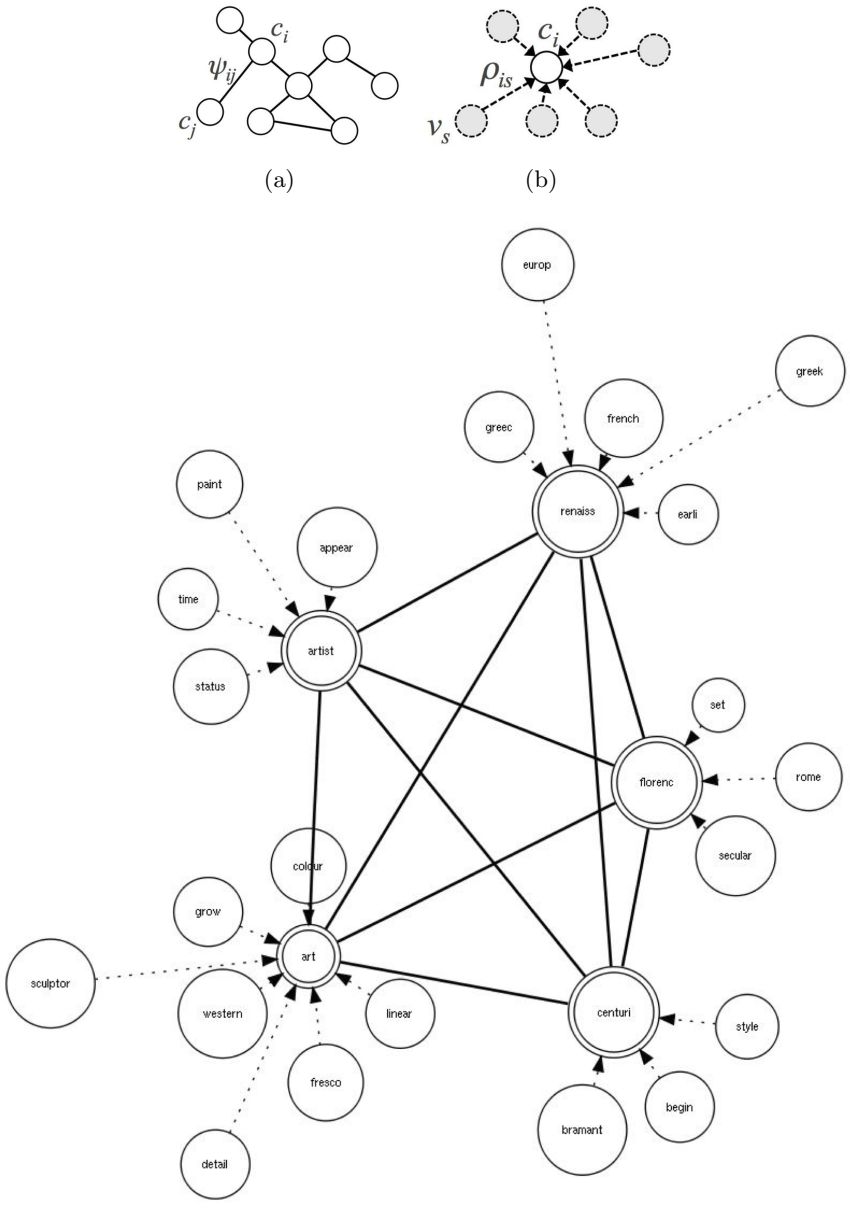
Un modello probabilistico per la rappresentazione della Semantica

Dopo aver individuato tutta una serie di problemi nell'ambito dell'estrazione e della rappresentazione del significato, nonché le carenze delle varie soluzioni formali, dei linguaggi e dei modelli probabilistici attualmente presenti in letteratura, proveremo in questo capitolo a modellare formalmente e quindi sfruttare l'idea del *Grafo di Termini* (*informal Lightweight Ontologies*). Una simile rappresentazione della conoscenza può essere allora definita come una struttura composta da nodi-concetti ed archi ponderati tra essi (in piedi per le relazioni semantiche tra concetti) e dove ogni concetto può essere definito attraverso un ulteriore grafo di tipo gerarchico, aciclico e formato da parole che meglio lo specializzano.

3.1 Il grafo di *Concetti*

Proviamo a definire quindi il *Grafo di Concetti*[1] come una tripla $\mathcal{G}_{\mathcal{C}} = \langle N, E, C \rangle$ dove N indica un set finito di nodi, E un set di archi pesati su N attraverso i pesi ψ_{ij} tali che $\langle N, E \rangle$ sia un grafo a-diretto (si veda la figura 3.1(a)), e C definisce un set di concetti, in modo che per

[1] La definizione corretta per oggetti di questo tipo dovrebbe essere *Grafo di Termini* ma sfruttando le precisazioni introdotta nel capitolo precedente, si compie qui un abuso di nomenclatura tra "*termine*" e *concetto*"



(a) (b) (c)

Figura 3.1 Rappresentazione grafica della conoscenza - 3.1(a) Grafico di Concetti: il peso ψ_{ij} rappresenta la probabilità che due concetti sono semanticamente correlate. 3.1(b) Rappresentazione grafica di un concetto: il peso ρ_{in} rappresenta la probabilità che una parola è semanticamente correlate ad un concetto. 3.1(c) Grafico dei Concetti che rappresenta una informal Lightweight Ontology estratta da una serie di documenti sul tema *"Arte Rinascimentale"*.

ogni nodo $n_i \in N$ esiste uno ed un solo concetto $c_i \in C$. Il peso ψ_{ij} può essere considerato come il grado di correlazione semantica tra due concetti c_i *è-relazionato*$_{\psi_{ij}}$*-a* c_j e può essere calcolato come la probabilità $\psi_{ij} = P(c_i, c_j)$. Dal momento che le relazioni tra i nodi possono essere solo del tipo *è-relazionato-a* allora questa rappresentazione può essere considerata al più una concettualizzazione di tipo *light* che noi indicheremo di seguito con $\mathcal{O}$. La probabilità di $\mathcal{O}$ dato il parametro τ[2] può essere quindi scritta come la probabilità congiunta tra tutti i concetti. Assumiamo sia possibile scrivere la probabilità congiunta come la fattorizzazzione di:

$$P(\mathcal{O}|\tau) = P(c_1, \cdots, c_H|\tau) = \frac{1}{Z} \prod_{(i,j)\in E_\tau} \psi_{ij} \tag{3.1}$$

dove H è il numero di concetti, $Z = \sum_{\mathcal{O}} \prod_{(i,j)\in E_\tau} \psi_{ij}$ è una costante di normalizzazione e τ può essere utilizzato per modulare il numero di archi nel grafo. Ogni concetto c_i, a sua volta, può essere definito come un grafo gerarchico di parole v_s ed un ulteriore set di links pesati da un altro parametro (o più propriamente funzione potenziale) indicato di seguito come ρ_{is} (si veda la figura 3.1(b)). Il peso ρ_{is} può misurare allora fino a che punto quella stessa parola è legata ad un concetto o, in altri termini, quanto abbiamo bisogno di una parola per specificare il concetto stesso. Tale peso può nuovamente essere considerato come una probabilità condizionata $\rho_{is} = P(c_i|v_s)$ e quindi, analogamente a quanto fatto per $P(\mathcal{O}|\tau)$, la probabilità di un concetto, indicato come c_i dato un parametro μ, è definita attraverso i fattoriρ_{is}

$$P(c_i|\{v_1, \cdots, v_{V_\mu}\}) = \frac{1}{Z_C} \prod_{s\in S_\mu} \rho_{is} \tag{3.2}$$

dove, in questo caso, $Z_C = \sum_C \prod_{s\in S_\mu} \rho_{is}$ è una costante di normalizzazione e V_μ il numero di parole che definiscono il concetto strettamente legate al parametro di modulazione μ. Per valutare allora le espressioni 3.1 e 3.2 risulta indispensabile calcolare entrambi i fattori ψ_{ij} e ρ_{is} rispettivamente. Nella sezione successiva si mostrerà allora come tali rappresentazioni grafiche possono essere apprese direttamente dai documenti di tipo testo facendo uso di una specifica tecnica probabilistica nota come latent Dirichlet Allocation(LDA) [7]

[2]Verrà chiarita durante il prosieguo della trattazione la funzionalità del parametro τ legato sostanzialmente al numero di archi coinvolti

3.2 Apprendimento dei *Concetti* e delle *Relazioni Semantiche* attraverso un metodo probabilistico

Proviamo a definire formalmente il nostro spazio di lavoro per evitare ambiguità e incongruenze. Possiamo dire allora che un concetto è lessicamente identificato da una parola specifica di un dato vocabolario ed è rappresentato a sua volta attraverso un insieme di parole connesse (si veda la figura 3.1(b)): una *parola* può essere quindi vista come un oggetto di un vocabolario indicizzato $\{1, \cdots, V\}$ e rappresentata in forma vettoriale normalizzata cosicché la v-esima istanza è descritta da un V-ettore w tale che $w^p = 1$ e $w^q = 0$ con $p \neq q$. Un documento è quindi una sequenza di L parole descritta da $\mathbf{w} = (w_1, w_2, \cdots, w_L)$, dove w_n risulta l'n-esima parola nella sequenza ed un corpus , seguendo la stessa logica, è così una collezione di M documenti descritto da $\mathcal{D} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_M\}$.

Individuando ora come obbiettivo principale l' estrazione automatica del significato da un documento o da una serie di documenti, che nel nostro caso sarà rappresentata da un grafo di concetti, appare ovvio stabilire quando una parola denota propriamente un concetto e/o contribuisce a definire una parte di esso. A questo scopo, il nostro metodo considera in prima istanza ogni parola come un possibile concetto e calcola il suo grado di associazione con le parole rimanenti, ovvero la probabilità condizionata di un concetto $P(c_i|\{v_1, \cdots, v_V\}_{\neq i})$ con $i \in \{1, \cdots, V\}$ tenendo anche presente che ogni parola del vocabolario può essere un potenziale concetto. Sarebbe sufficiente quindi calcolare la probabilità per ogni concetto al fine di determinare chi tra essi è meglio rappresentato da un insieme di parole o, in altri termini, quali sono i concetti più probabili. In questo modo, possiamo definire effettivamente quali parole del corpus rappresentano i concetti e quindi calcolare le loro dipendenze statistica, ψ_{ij}, per ottenere il grafo di concetti. Poiché ogni concetto è rappresentato attraverso una parola, il calcolo di $\rho_{is} = P(c_i|v_s) = P(v_i|v_s)$, dove il concetto c_i è lessicamente identificato da v_i e $\psi_{ij} = P(c_i, c_j) = P(v_i, v_j)$ dove i concetti c_i e c_j sono lessicamente identificati da v_i and v_j rispettivamente, può essere visto come un problema di *word association* e quindi risolto attraverso un metodo probabilistico.

Dato lo scenario così delineato e considerando la spazio dei parame-

tri in gioco, dei numerosi metodi probabilistici a disposizione, la nostra scelta è ricaduta sull'utilizzo del *Topic Model* descritto da *Griffiths* e *Blei* in [70, 7]. La teoria introdotta in questi lavori afferma soprattutto che la rappresentazione semantica all'interno della quale sono considerati i significati delle parole come termini di una serie di argomenti (topics) probabilistici z_n, è possibile solo se viene fissata l'indipendenza tra le singole parole w_n o, come indicato nelo specifico lavoro, viene imposta l'ipotesi del *bags of words*[3].

Supponiamo ora di indicare con $P(z)$ la distribuzione di probabilità sui topics z in un particolare documento e $P(w|z)$ come la probabilità condizionata sulla parola w dato il topic z, sarà allora possibile sfruttare la natura del *Topic Model* come metodo generativo e quindi ogni parola w_n in un documento risulta scelta dalla distribuzione $P(w|z)$ dopo aver campionando il topic dalla distribuzione $P(z)$.

Possiamo allo stesso modo definire $P(z_n = k)$ come la probabilità che il k-esimo topic sia stato campionato per la n-esima parola in modo da evidenziare quali topic sono rilevanti per un particolare documento e $P(w_n|z_n = k)$ come la probabilità della parola w_n condizionata al topic k, capace di indicare le parole significative per il topic fissato. Il modello quindi consente di specificare anche la distribuzione per ogni parola v_i del vocabolario all'interno del corpus di documenti come:

$$\begin{aligned} P(v_i) &= \sum_{d=1}^{D} P(w_n^d = v_i|\mathbf{w}_d)P(\mathbf{w}_d) \\ &= \sum_{d=1}^{D}\sum_{k=1}^{T} P(w_n^d = v_i|z_n^d = k, \mathbf{w}_d)P(z_n^d = k|\mathbf{w}_d)P(\mathbf{w}_d) \end{aligned} \quad (3.3)$$

dove T è il numero di topics ed n si riferisce all' n-esima parola all'interno del documento[4].

Ritornando al nostro problema di *word association*, attraverso il *Topic Model*, è possibile quindi ricondursi ad un problema di semplice

[3]Tale ipotesi indica che un documento può essere considerato come un vettore di features in cui ogni elemento evidenzia la presenza (o assenza) di una parola: side effect di questa ipotesi è però la perdita delle informazioni relative alla posizione della parola stessa all'interno del documento.

[4]Si fa notare che in ogni documento è possibile avere più di un indice che si riferisce alla parola v_i: in questo caso si preferirà usare n per non appesantire la notazione.

previsione e quindi la probabilità condizionata risultante può essere ottenuta semplicemente sommando i vari contributi su tutti i documenti e su tutti i topics[5]:

$$\begin{aligned} P(v_i|v_j) &= \sum_{d=1}^{D} P(w_n^d = v_i, \mathbf{w}_d | w_{n+1}^d = v_j) \\ &\propto \sum_{d=1}^{D}\sum_{k=1}^{T} P(w_n^d = v_i | z = k, \mathbf{w}_d) \times ... \\ &\times \quad P(w_{n+1}^d = v_j | z = k, \mathbf{w}_d) P(z = k | \mathbf{w}_d) \end{aligned} \tag{3.4}$$

dove $P(w_n^d = v_i | w_{n+1}^d = v_j, z = k, \mathbf{w}_d) = P(w_n^d = v_i | z = k, \mathbf{w}_d)$ vera per le ipotesi di *bags of words* ed *exchangeability* presenti nel *Topic Model*. Con queste premesse ed assunzioni, si potrebbero allora utilizzare diverse tecniche statistiche in grado di apprendere su grandi repository di documenti per ottenere una procedura di tipo unsupervised.

La nostra scelta, come già accennato in precedenza, è però ricaduta su una versione smoothed del modello generativo introdotto da Blei in [7] chiamata latent Dirichlet Allocation (*LDA*) che fa uso del *Gibbs sampling* e di distribuzione multinomiale[6]. I risultati così ottenuti eseguendo l'algoritmo LDA su una serie di documenti sono sintetizzati attraverso due matrici:

1. matrice parola-topics Φ contenente la probabilità che una parola v_i è assegnata al topic j, $P(w_n^d = v_i | z_n^d = k, \mathbf{w_d})$;

2. matrice topics-documenti Θ contenente la probabilità che un topic j sia assegnato ad alcune parole all'interno di un documento, $P(z = k|\mathbf{w}_d)$.

Proveremo allora a descrivere la procedura formale che ci permette di sfruttare queste due matrici nell'ambito della rappresentazione della semantica, o meglio per come abbiamo definito noi la questione, all'interno della *light semantics*.

[5] E' opportuno riferirsi a A.1 in appendice di questo lavoro per la computazione matematica completa della probabilità.

[6] Ovvero una distribuzione di probabilità standard estratta da distribuzione di *Dirichlet* [27].

3.3 Apprendimento dei *Concetti* e delle *relazioni* del Grafo

Dato un preciso set di documenti, l'intera procedura che estrae automaticamente un *Grafo di Concetti* (*GC*) si compone principalmente di due fasi: una per l'individuazione dei concetti dal vocabolario (*apprendimento* delle relazioni parole-concetto) ed un'altra per il calcolo delle relazioni tra concetti (*graph learning*). In entrambi le fasi è però necessaria un'opportuna scelta di alcuni parametri che possono essere poi impostati sia manualmente che automaticamente a seconda delle diverse applicazioni che usufruiranno di tali basi di conoscenza. Dopo una prima analisi e descrizione anche formale dei parametri necessari, vedremo come sviluppare una successiva procedura di ottimizzazione multi-obiettivo per determinarne i valori: l'implementazione di tale procedura potrà basarsi sia sul ben noto algoritmo *Random Hill-Climbing* [24] per una versione completamente automatica nonché su una impostazione di tipo manuale sensibilmente più rapida ed esaustiva ma chiaramente che coinvolge l'essere umano in alcune decisioni.

Prendiamo quindi in considerazione un documento dal corpus (la procedura vale ancora se si sceglie più di un documento) con le rispettive V parole/concetto contenute in esso e attraverso la matrice Φ valutiamo l'Eq. 3.2, ovvero la probabilità $P(c_i|\{v_1, \cdots, v_V\}_{\neq i})$. Analizzando la distribuzione di probabilità dei concetti così ottenuti possiamo individuare una soglia e quindi filtrare quelli che risultano meno probabili: in termini di modello, tale filtraggio corrisponde in pratica alle parole che non rappresentano evidentemente un concetto. Un procedura analoga potrebbe essere la selezione diretta del numero di concetti H nell'insieme di quelli plausibili in modo da considerare H come variabile, ovvero un parametro che assume un valore fissato all'interno di un range plausibile. Una volta che il numero di concetti è stata fissato, il passo successivo consiste nell'individuare per ogni concetto c_i una soglia μ_i in grado di selezionare il numero di relazioni *concetti-parole* attraverso la probabilità $P(c_i|\{v_1, \cdots, v_V\}_{\neq i})$, $\forall i$. Lo step finale è la valutazione della probabilità $P(\mathcal{O}|\tau)$, Eq. 3.1, che in qualche modo rappresenta le relazioni *concetto-concetto*, o più formalmente, il valore τ in grado di determinare ψ_{ij} maggiormente probabile. Sintetizzando i risultati della nostra analisi, sarà quindi indispensabile indicare un valore per il parametro H che fissa il numero di concetti, un valore τ e

gli H valori di μ_i ,$\forall i \in [1, \cdots, H]$ evidenziando una cardinalità $H + 2$ dello spazio dei parametri coinvolti.

Al variare di tali parametri si osserverà chiaramente una modulazione della rispettiva ontologia $\mathcal{O}_t$[7] creando quindi diverse estrazioni e quindi diverse rappresentazioni dello stesso insieme di documenti. Per completare la nostra modellazione, è necessario quindi definire un ulteriore criterio, che sarà ovviamente in qualche modo legato al nostro scopo "applicativo", in grado di selezionare l'ontologia migliore: potremmo essere interessati, ad esempio, a quelle ontologie che meglio rappresentano individualmente ciascun documento del repository posto in ingresso al sistema.

3.3.1 Procedura di ottimizzazione multi-obbiettivo

Come abbiamo mostrato nel paragrafo precedente, siamo in grado di ottenere un ontologia $\mathcal{O}_t$ per ogni set di parametri, $\mathbf{\Lambda}_t = (H, \tau, \mu_1, \cdots, \mu_H)_t$. Se si potesse allora definire una funzione che misura la qualità delle singole ontologie $\mathcal{O}_t$, allora potremmo definire la serie di parametri per i quali i risultati dell'intero sistema risultino il migliore possibile. Un modo per dire che una ontologia è la migliore possibile per quella serie di documenti è sicuramente quello di dimostrare che essa produce lo score massimo raggiungibile per ciascuno dei documenti quando essa stessa è utilizzata come base di conoscenza per l'esecuzione di query sul set contenente solo i documenti che l'hanno generata.

A tale scopo supponiamo di utilizzare un corpus di M documenti indicati da $\mathcal{D} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_M\}$, e sia $\mathcal{O}_t$ l'ontologia t-esima possibile costruita da quel repository con il set di parametri $\mathbf{\Lambda}_t$. Ogni ontologia può essere quindi rappresentata attraverso la sintassi booleana di *Lucene*, così come si mostrerà nel dettaglio in sezione 4.1.1, che corrisponde, in alternativa, a selezionare due vettori, uno che rappresenta tutte le coppie $\mathbf{q}_t = \{q_1, \cdots, q_U\}_t$ e l'altro che rappresenta i singoli fattori di relazione tra di esse, considerati secondo il boost implementato in *Lucene* stesso $\mathcal{B}_{\mathbf{q}_t} = \{\mathcal{B}_{q_1}, \cdots, \mathcal{B}_{q_U}\}_t$. Eseguendo una query di ricerca in *Lucene* sullo stesso repository $\mathcal{D}$ che si appoggia all'ontologia $\mathcal{O}_t$, otteniamo quindi uno score per ogni documento $\mathbf{w}_i$ indicato come $\mathbf{S}_t = \{\mathcal{S}(\mathbf{q}_t, \mathbf{w}_1), \cdots, \mathcal{S}(\mathbf{q}_t, \mathbf{w}_M)\}_t$ attraverso l' Eq. 4.1, dove chiaramente ognuno di essi dipende dal set di parametri $\mathbf{\Lambda}_t$. Un

[7] Con la variabile t si indicano i diversi set di valori assunti dai singoli parametri in questa fase.

criterio allora per calcolare il miglior set $\mathbf{\Lambda}$ potrebbe essere quello di massimizzare il valore dello score per ogni documento, il che significa cercare implicitamente l'ontologia che meglio descrive e rappresenta l'intero repository. Va però sottolineato che una tale procedure di massimizzazione deve ottimizzare allo stesso tempo tutti gli M elementi di $\mathbf{S}_t$ creando così problemi di convergenza e di elevata complessità computazionale. In alternativa a questa ipotesi e per ridurre il numero degli oggetti da ottimizzare, si potrebbe contemporaneamente massimizzare il valore medio degli score e minimizzare la loro deviazione standard in modo da semplificare e trasformare il problema multi-obiettivi in un problema a due variabili. Possiamo ancora riformulare quest'ultimo problema attraverso una combinazione lineare dei suoi obiettivi, ottenendo così una singola funzione, indicata con *Fitness* ($\mathcal{F}$), che dipende direttamente da $\mathbf{\Lambda}_t$ in termini di:

$$\mathcal{F}(\mathbf{\Lambda}_t) = E_m\left[\mathcal{S}(\mathbf{q}_t, \mathbf{w}_m)\right] - \sigma_m\left[\mathcal{S}(\mathbf{q}_t, \mathbf{w}_m)\right], \tag{3.5}$$

dove E_m è il valor medio di tutti gli elementi di $\mathbf{S}_t$ e σ_m la standard deviation. Riassumendo possiamo quindi scrivere:

$$\mathbf{\Lambda}^* = \underset{t}{\operatorname{argmax}}\{\mathcal{F}(\mathbf{\Lambda}_t)\} \tag{3.6}$$

Proviamo a semplificare ulteriormente la nostra procedura considerando che la funzione di *Fitness* ($\mathcal{F}$) indicata nell' Eq. 3.5 dipende da $H + 2$ parametri e quindi lo spazio delle possibili soluzioni potrebbe crescere esponenzialmente. Per limitare tale spazio si può pensare di ridurre il numero di parametri in gioco, ad esempio considerando i parametri μ_i costanti, ovvero $\mu_i = \mu$, $\forall i \in [1, \cdots, H]$ in modo da passare da $H + 2$ parametri a 3 parametri forti del fatto che questa riduzione risulta invariante rispetto al numero di concetti.

Procedura Evolutiva di selezione dei parametri - RMHC

Il metodo di ottimizzazione che abbiamo poi scelto è noto all'interno della comunità scientifica di Computazione Evolutiva come *Random Mutation Hill-Climbing* (RMHC) [24] ed esegue in pratica una procedura di ricerca attraverso metodo Monte Carlo a zero-temperatura generando nuove soluzioni come variazioni della migliore individuata (estensione del *Random Search*). La scelta è ricaduta sul RMHC soprattutto perchè tale metodo ha evidenziato le sue ottime capacità in

molti problemi di ottimizzazione di funzioni-costo NP-hard, deceptive e neutrali come illustrato in [24, 16, 55, 19] in quanto lavora in accordo al seguente algoritmo 3.2.:

---

select a starting point by generating a random solution;
set such a solution to the *best-evaluated*;
while a pre-defined number of fitness evaluations has not been performed **do**
  generate a new candidate solution as variation of the current best.

  if mutation leads to an equal or higher fitness **then**
    set *best-evaluated* to the resulting solution;
  end if
end while

Figura 3.2 RMHC optimisation procedure

Soluzione approssimata per la selezione dei parametri

Proviamo a questo punto a compiere ulteriori ragionamenti per capire effettivamente quali possono essere altre riduzioni efficaci per lo spazio dei parametri in gioco. Per evitare quindi di dover creare e gestire grafi troppo piccoli o troppo grandi (chiaramente con dimensioni medie si ottimizza sia il processo di quering in termini di una discreta quantità di informazioni di supporto che di manipolazione del grafo stesso), ci metteremo nell'ipotesi che il numero H di concetti può variare solo da un minimo di 5 ad un massimo di 20[8] e, considerando che si tratterà poi solo di un numero intero, abbiamo così fissato il numero di concetti possibili per H pari a 15. Considerando allora che ψ_{ij} e ρ_{is} sono probabilità ed i rispettivi valori reali $\tau \in [0, 1]$ e $\mu_i \in [0, 1]$, abbiamo pensato di impostare uno step di analisi pari all'1% dell'intero intervallo $[0, 1]$ cosicché risultino 100 possibili valori per τ e 100 for each μ_i.

Risulta che con il numero di parametri/valori in gioco risulteranno quindi $100 \times 100 \times H \times 15$ possibili valori di Λ ovvero $750.000@_{H=5}$

[8] Ci siamo accorti in modo empirico che eccedere verso valori alti o eccessivamente bassi rispetto al numero dei concetti è poco producente sia per la mole di dati che poi devono essere manipolati - evidentemente caso *big graph* - sia per il basso contenuto ontologico - caso *very small graph* - da utilizzare effettivamente nelle *query expantion* o in altri casi applicativi di interesse.

e $3.000.000@_{H=20}$. Per limitare tale spazio, si può effettivamente ridimensionare il numero dei valori considerando come già detto $\mu_i = \mu$, $\forall i \in [1, \cdots, H]$ e ottenere così 150.000 possibili valori di Λ, indipendentemente dagli H concetti, In questa ottica abbiamo poi ridotto anche il relativo spazio residuo di soluzione applicando il metodo di clustering *K-means* a tutti i valori ψ_{ij} e ρ_{is}. Seguendo questo approccio e scegliendo, ad esempio, 10 classi di valori per τ e μ, si ottiene che lo spazio di possibili Λ è ridotto a $10 \times 10 \times 15$, ovvero 1.500. Come conseguenza di questo ragionamento, la soluzione ottimale può essere ottenuta esattamente dopo l'esplorazione di tutto lo spazio delle soluzioni a differenza del metodo evolutivo descritto in precedenza.

Interpretazione del grafo

E' inoltre molto importante chiarire che il *Grafo di termini* non è da considerarsi come una matrice di co-occorrenza anche perché in letteratura, così come indicato da Efthimiadis e Manning [21, 52], è stato già dimostrato che tecniche basate su tali matrici sono estremamente limitate in fase di query expantion. In realtà, il nucleo del grafo è quindi la probabilità $P(v_i, v_j)$ calcolata attraverso il *probabilistic Topic Model* e nello specifico con il problema della *word association*: si ricorda allora che nel *Topic Model* tale problema è visto come un problema di previsione ovvero dato che si è presentata una forma, quali nuove parole si potrebbero verificare nello stesso contesto? Ciò significa che il modello non tiene in conto solo del fatto che due parole si verificano nello stesso documento, ma che si verificano nello stesso documento quando è assegnato un *topic* specifico al documento stesso[9].

Come risultato di queste considerazioni, si evidenzia quindi che la nostra rappresentazione è sicuramente molto più vicina ad una idea di *Grafo* che di matrice e, sfruttando tutto ciò che è stato descritto fin qui, proveremo che tale rappresentazione risulterà molto efficiente quando la si usa come rappresentazione del significato contenuto in un insieme di documenti testuali.

[9]Infatti la probabilità $P(v_i|v_j)$ è il risultato della somma su tutti i topic.

Capitolo 4

Sperimentazione

Per valutare l'efficacia del nostro approccio, abbiamo sviluppato un applicativo web based in Java e Java Server Pages, $iSoS_{lite}$, che include una versione personalizzata delle API open source di Apache Lucene[1], un motore di full text retrieval. Tale applicativo è composto da tre parti principali: un *Web Crawler*, un *Indexer* ed un *Searcher* illustrate nelle prossime sezioni e che rendono possibile la fase di convalida del nostro modello in funzione di due problematiche ritenute in letteratura molto importanti:

1. **Text Retrieval** - la possibilità di recuperare la maggior parte delle informazioni rilevanti (documenti di testo) all'interno di un repository di grandi dimensioni.

2. **User Satisfaction** - quanto un utente rimane soddisfatto dalle pagine restituite da un motore di ricerca a valle della propria richiesta di informazione.

In entrambi gli scenari di sperimentazione apparirà evidente quindi la necessità di utilizzare una rappresentazione dell'informazione capace di conservare bene il significato per come l'utente lo ha immaginato e codificato nel proprio messaggio[2] e quindi la bontà effettiva del nostro modello a sfruttare la *light semantics* in tali contesti.

[1] http://lucene.apache.org/

[2] Proprio come il caso dello *scrittore* nel capitolo 1 di questo lavoro.

4.1 Incorporare il nostro modello in un motore di ricerca: Web Crawling e Indexing

Ogni motore di ricerca web memorizza informazioni su pagine web recuperate da un Web Crawler, ovvero da un programma che segue in sostanza tutti i collegamenti che trova navigando il web. A causa delle limitazioni hardware dei nostri sistemi, non abbiamo implementato un vero e proprio sistema di scansione autonomo ma un ambiente relativamente più piccolo: la fase di scansione è stata quindi effettuata inviando una query specifica al famoso motore di ricerca web di Google[3], ed estraendo gli URL dai risultati recuperati. Da questi URL il nostro software scarica le corrispondenti pagine web che vengono memorizzate in cartelle specifiche e, infine, indicizzate.

Tuttavia, ci sono diversi approcci per l'indicizzazione delle pagine web come ad esempio va ricordato che alcuni motori non indicizzano per intero le parole ma solo i loro lemmi. Il processo di stemming, quindi riduce le parole alla loro forma-radice partendo dalla considerazione che *word* aventi la stessa radice portano un analogo contenuto informativo. Una ulteriore manipolazione del repository prima della memorizzazione vera e propria dell'*index* arriva dal filtraggio attraverso *stop-word* per evitare l'indicizzazione di parole comuni come le preposizioni, congiunzioni e articoli che spesso non portano alcuna informazione aggiuntiva.

Entrando un po' più nel dettaglio della nostra applicazione, è stato sviluppato a questo livello un analizzatore personalizzato incluso in *Lucene* che permette l'indicizzazione sia delle parole che dei loro lemmi senza alcun filtraggio di *stop-word*. Abbiamo considerato poi la possibilità di inserire le nostre *iLO* nel processo di ricerca, *iLO* composte sostanzialmente dai soli lemmi per ottimizzare la ricerca senza penalizzare la precisione originale della query. Un fase preliminare di analisi dei documenti[4] è stata necessaria però per riconoscere i vari oggetti presenti dei testi come *tag*, *metadati* e *contenuti informativi* e quindi selezionare sola la parte di essi che contiene l'informazione. E' necessario anche rimarcare che la fase di indicizzazione di un repository può aiutare a decidere quali documenti rispondono ad una determina-

[3] www.google.com

[4] Questo passaggio è spesso indicato come fase di *parsing*.

ta query ma non si pone come scopo la classificazione del repository stesso.

4.1.1 Searching and Scoring

Il cuore di un motore di ricerca, come anche della nostra applicazione, è la sua capacità di assegnare un grado o un ordine ai documenti che rispondono ad una determinata query. Questo può essere fatto chiaramente attraverso la computazione di uno score e/o attraverso specifiche politiche di *ranking*. Diverse operazioni di Information Retrieval (incluso lo score di documenti restituiti ad una query, classificazione e/o clustering) sono spesso basate sullo *vector space model* dove i documenti sono rappresentati come vettori in uno spazio vettoriale comune [52].

Secondo questo modello, ogni documento $\mathbf{w}$ può essere considerato quindi un vettore $\vec{V}(\mathbf{w})$ contenente una componente per ogni parola (o termine) del vocabolario: il valore specifico di ogni componente viene così calcolato con la *term frequency-inverse document frequency* (*tf-idf*). Nello specifico, la *tf-idf* assegna al termine w un peso all'interno di un documento $\mathbf{w}$ dato da $tf\text{-}idf_{w,\mathbf{w}} = tf_{w,\mathbf{w}} \times idf_w$, dove $tf_{w,\mathbf{w}}$ è legato alla frequenza del singolo termine[5]e idf_w(*inverse document frequency*) è definita a sua volta da $idf_w = 1 + \log\left(\frac{M}{df_w}\right)$con M numero totale dei documenti e df_w numero di documenti contenenti il termine w[6]. Seguendo ancora questo modello anche per quel che riguarda le queries, è possibile rappresentarle attraverso vettori $\vec{V}(\mathbf{q})$ garantendo così la possibilità di calcolare lo score relativo per un documento $\mathbf{w}$ in base ad una misura di *cosine similarity*:

$$score(\mathbf{q}, \mathbf{w}) = \frac{\vec{V}(\mathbf{q}) \cdot \vec{V}(\mathbf{w})}{|\vec{V}(\mathbf{q})| \cdot |\vec{V}(\mathbf{w})|}$$

dove il denominatore rappresenta il prodotto tra le distanze euclidee ed è in grado di compensare l'effetto della lunghezza dei documenti. Il

[5]Definisce il numero di volte che il termine w appare nel documento corrente $\mathbf{w}$: un documento che ha più occorrenze del termine riceverà chiaramente uno score più alto.

[6]In questo modo si garantisce che i termini che si verificano raramente nel documento, contribuiranno poco nel calcolo effettivo dello *score*

meccanismo vero e proprio di score, utilizzato da Lucene e derivato direttamente dalla misura di *cosine similarity*, è però legato formalmente alla seguente funzione:

$$\mathcal{S}(\mathbf{q}, \mathbf{w}) = \mathcal{C}(\mathbf{q}) \cdot \mathcal{N}_{\mathbf{q}}(\mathbf{q}) \cdot \sum_{q \in \mathbf{q}} (tf_{q,\mathbf{w}} \cdot idf_q^2 \cdot \mathcal{B}_q \cdot \mathcal{N}(q, \mathbf{q})) \tag{4.1}$$

dove, oltre ai termini $tf_{q,\mathbf{w}}$ e idf_q illustrati in precedenza, possiamo indicare con $\mathcal{C}(\mathbf{q})$ il fattore di score basato su quanti dei termini della query si sono ritrovati all'interno del documento specificato: in altre parole, un documento che contiene più di termini della query riceverà un punteggio superiore a quello che ne conterrà di meno.

$\mathcal{N}_{\mathbf{q}}(\mathbf{q})$ e $\mathcal{N}(q, \mathbf{q})$ sono invece fattori di normalizzazione usati per rendere lo score comparabile tra le diverse queries. Si fa notare allora che tali fattori non influiscono sul ranking del singolo documento dal momento che tutti i documenti classificati sono moltiplicati per lo stesso numero.

$\mathcal{B}_q$ è il fattore di boost per la ricerca del termine q nella query $\mathbf{q}$ così come verra descritto meglio nel seguito di questo lavoro. Possiamo dire che intuitivamente attraverso questo fattore sarà possibile assegnare un peso maggiore ad un termine specifico di una query affidando quindi ad esso una maggiore importanza quando si verifica all'interno di un documento. E' da notare inoltre che Lucene incorpora algoritmi di ricerca e quindi di scoring molto efficienti basati sia su questo tipo di modello che sul modello booleano. Per effettuare allora le ricerca con l'ausilio delle *iLOs*, abbiamo così personalizzato il meccanismo di esecuzione delle queries stesse: difatto le nostre ontologie sono rappresentate come coppie di parole correlate attraverso un valore reale (ψ_{ij} e/o ρ_{is} che indicano rispettivamente i legami tra concetti e tra concetti e parole così come descritto nelle sezioni 3.1 e 3.2 e sintetizzati come *Relation factors*) e abbiamo usato il meccanismo di *terms boost* implementato già all'interno di Lucene per estendere la query originale con il contributo dell'*iLO*. La tabella 4.1.1 mostra un esempio di rappresentazione dell'ontologia per il topic *Serbatoi*: utilizzando quindi una ontologia sìfatta nel processo di ricerca, il nostro sistema pone come query la seguente stringa:

```
((tank AND roof)^4.0) OR ((tank AND larg)^2.0)...
```

che significa la ricerca delle coppie di parole *tank* AND *roof* con un boost factor di 4.0 OR la coppia di parole *tank* AND *larg* con un boost

Conceptual Level		
Concept i	**Concept** j	**Relation Factor** (ψ_{ij})
tank	roof	4,0
tank	water	3,37246
tank	storag	3,33194
...	...	...
liquid	type	3,43828
liquid	pressur	3,07028
...	...	...

Word Level		
Concept i	**Word** s	**Relation Factor** (ρ_{is})
tank	larg	2,0
tank	construct	1,6
...	...	...
liquid	maker	1,11673
liquid	fix	1
...	...	...

Tabella 4.1 Un esempio di una *iLO* del topic *Serbatoi* estratta da un piccolo repository di documenti di testo.

factor di 2.0 e così via. Il valore di default del boost factor è 1 ed è assegnato origianariamente alla query di partenza effettuata dall'utente. Un modo per rendere più efficace la partecipazione dei termini del grafo è allora quello di definire per essi un fattore di boost superiore ad 1. Inoltre, a causa del fatto che noi consideriamo le relazioni tra concetti più importanti delle relazioni tra concetti e parole (questa è una conseguenza della struttura gerarchica del grafico da noi introdotto) abbiamo spostato l'intervallo originale di valori di ψ_{ij} e ρ_{is} da $[0, 1]$ in $[3, 4]$ e $[1, 2]$ rispettivamente.

4.2 Prima Fase di sperimentazione e convalida del sistema

Sono stati selezionati 9 scenari di ricerca per convalidare i risultati proposti dal motore *iSoS* e sono state effettuate altrettante query sul repository estratto dal sito http://www.thomasnet.com[7]: il numero di documenti in formato *html* presi in considerazione è stato di 154.243 per un totale di 3.0 GB e la lingua di riferimento è l'inglese. Le query inoltre sono state rafforzate dall'utilizzo di *informal Lightweight Ontology* opportunamente create per immagazzinare la conoscenza rispettivamente dei nove scenari.

[7]Noto come "big green books" e come "Thomas Registry", ThomasNet è una directory multi-volume di informazioni su prodotti industriali di circa 650.000 distributori, produttori e società di servizi distribuite all'interno di circa 67.000 categorie industriali.

1) *Lubrificanti*	4) *Generatori*	7) *Valvole*
2) *Pompe*	5) *Trasformatori*	8) *Cavi LAN*
3) *Adesivi*	6) *Inverter*	9) *Serbatoi*

Tabella 4.2 Topic selezionati per la sperimentazione

La scelta di questi specifici scenari è stata effettuata attraverso l'analisi del repository di partenza ed in particolare sfruttando l'afferenza delle diverse pagine a categorie: in fase di *crawling* del repository stesso, sono state *taggate* tutte le pagine in base al percorso del *Web directory*[8] e si è ottenuto per ognuna di esse una label in grado di discriminarne il contenuto. Uno scenario considerato come valido garantisce così la presenza di almeno 100 pagine/label. I topic che quindi rientrano in questa procedura sono riportati in tabella 4.2. È stato necessario così introdurre un altro motore di ricerca per confrontare le prestazioni e capire se e come il nostro sistema risponde correttamente alle richieste dell'utente. La scelta per il confronto è così ricaduta su uno dei più performanti motori di ricerca open source presenti oggi in rete, *Lucene*. Per avere una misura in qualche modo "oggettiva" dei risultati proposti dai due motori di ricerca, è stato predisposto uno schema *xml*, come indicato nell'esempio sottostante, per ben codificare le intenzioni utente legate alle query ed evitare quindi casi di ambiguità, per questo motivo si fa riferimento a questo tipo di query come "faceted". Ogni subtopic deve soddisfare query di tipo informazionale.

[8] Una web directory non è né un motore di ricerca né archivia i siti attraverso tag, bensì li presenta attraverso categorie e sottocategorie tematiche, ovvero un elenco di siti web suddivisi in maniera gerarchica

```
<topic number="3" type="faceted" >
  <query> adhesive</query>
     <description>
         I am looking for information on adhesive.
     </description>
         <subtopic number="1" type="inf" >
            I am looking for web pages containing datasheets
            of several adhesive types
         </subtopic>
         <subtopic number="2" type="inf" >
            I am looking for descriptions of adhesives as products
         </subtopic>
         <subtopic number="3" type="inf" >
            I am looking for list of adhesive categories and their
            description
         </subtopic>
         <subtopic number="4" type="inf" >
            I am looking for information on adhesive usages
         </subtopic>
</topic>
```

In questo modo è anche possibile evitare di coinvolgere necessariamente grossi numeri di valutatori in quanto il concetto ricercato è ben spiegato dalle informazioni ausiliari inserite in questo formato. La struttura di questo xml, come anche di tutta la fase di sperimentazione, è stata predisposta come indicato in *TREC*[9] e le valutazioni finali sono strettamente legate ad indicatori standard di prestazione presenti in letteratura quali *Precision* e *Recall*: il primo è definito come la frazione di documenti recuperati che sono rilevanti, il secondo come la frazione dei documenti pertinenti che vengono recuperati. In particolare:

$$precision = \frac{|\{documenti\ rilevanti\} \cap \{documenti\ restituiti\}|}{\{documenti\ restituiti\}}$$

[9]Il Text Retrieval Conference (TREC), co-sponsorizzato dal National Institute of Standards and Technology (NIST) degli Stati Uniti e del Dipartimento della Difesa, è stato avviato nel 1992 nell'ambito del TIPSTER Text program. Il suo scopo è quello di sostenere la ricerca all'interno della comunità dell' information retrieval fornendo le infrastrutture necessarie per la valutazione su larga scala delle metodologie di recupero del testo.

Query	♯ di termini	♯ di coppie
Lubrificanti	54	69
Pompe	63	70
Adesivi	45	67
Generatori	58	68
Trasformatori	67	82
Inverter	62	84
Valvole	47	66
Cavi LAN	69	85
Serbatoi	51	66
Average Size	**57**	**73**

Tabella 4.3 Sintesi del numero di forme e numero di coppie estratte per i diversi topic/query

$$recall = \frac{|\{documenti\ rilevanti\} \cap \{documenti\ restituiti\}|}{\{documenti\ rilevanti\}}$$

4.2.1 Costruzione della Conoscenza a-priori

Per ogni scenario, un gruppo di 3 persone ha individuato un set di documenti per estrarre la conoscenza a-priori. Per effettuare quindi le ricerche rispetto ai nove scenari e quindi creare le *iLOs* da aggiungere alle query, sono stati selezionati da Internet tre documenti ritenuti in qualche modo significativi, che corrispondono in numero a circa l'1% della corrispondente serie di documenti per ogni topic (quelli considerati come insieme per l'apprendimento) e da essi sono stati estratti automaticamente i concetti e le parole attraverso *iSoS*[10]. La struttura delle diverse *iLOs* ottenute per ogni topic, ovvero il numero di termini e di coppie presenti nelle singole ontologie, è riportato nella tabella 4.3 dove la dimensione media della lista di termini e del numero di coppie risulta essere pari a 57 e 73 rispettivamente ottenute grazie alla procedura di apprendimento illustrata in sez. 3.3.1 impostando il valore ottimo di concetti $H \in [5, 15]$.

Abbiamo di seguito calcolato il numero medio di concetti per ogni struttura ottenendo $H = 9$ che significa che ogni concetto è stato specializzato approssimativamente da $57/9 \approx 6$ parole appartenenti al *word level*. Una ulteriore misura della qualità dei documenti scel-

[10]In appendice sono riportati sia le tabelle delle *ILO* memorizzate su MySql che la loro forma grafica

Lubrificanti	**Pompe**	**Adesivi**
1.txt 0.3601954	1.txt 0.39506367	1.txt 0.47583854
3.txt 0.33609098	2.txt 0.30361077	3.txt 0.3257996
2.txt 0.30370623	3.txt 0.27889237	2.txt 0.29266663
Generatori	**Trasformatori**	**Inverter**
3.txt 0.49393117	1.txt 0.4579826	2.txt 0.35039437
2.txt 0.34441447	3.txt 0.37482366	3.txt 0.34016997
1.txt 0.34012255	2.txt 0.27974448	1.txt 0.3167223
Valvole	**Cavi LAN**	**Serbatoi**
1.txt 0.45499673	2.txt 0.40091187	3.txt 0.3392135
2.txt 0.38837972	1.txt 0.28216907	1.txt 0.30538452
3.txt 0.3883196	3.txt 0.2487683	2.txt 0.2820832

Tabella 4.4 Funzione di Score valutata sui repository corrispondenti alla Conoscenza a-priori

ti[11], della distanza tra di essi (utilizzando la funzione di *Score* 4.1 come metrica) e della corrispondenza "*ontologia creata/peso dei singoli documenti*" è indicata di seguito attraverso la tabella 4.4 ed, in modo più intuitivo, con la figura 4.1. I valori presenti in tabella 4.4 e figura 4.1 indicano la funzione di *Score* calcolata con le diverse ontologie sul solo repository della conoscenza degli esperti[12]. Data l'ampiezza delle informazioni presenti in rete oggi rispetto ai diversi topic proposti, la scelta di tali documenti è stata fatta portando in conto sia il contenuto puramente semantico, ovvero considerando una valutazione da parte di esseri umani in riferimento alle query *xml*, sia cercando di ridurre al minimo la distanza (considerando come metrica nuovamente la funzione di *Score*) tra i singoli documenti all'interno dei topic estratti singolarmente. Ad onor del vero bisogna però dire che con una scelta un po' più accurata di tali documenti – in numero, in dimensione ed in contenuto - i risultati potrebbero altresì essere ancora più performanti in quanto sicuramente potrebbero non essere presenti in essi tutte le informazioni importanti per il contesto e quindi non sfruttare al massimo il fattore discriminante del sistema.

[11] In appendice sono riportati gli URL da cui sono stati scaricati i documenti per formare il repository della conoscenza: data la dinamicità della rete Internet, i documenti attuali potrebbero non corrispondere agli effettivi documenti da noi utilizzati, e per questo motivo una copia di questi è da noi custodita.

[12] Repository composto con i documenti dai quali è stata estratta precedentemente l'ontologia stessa

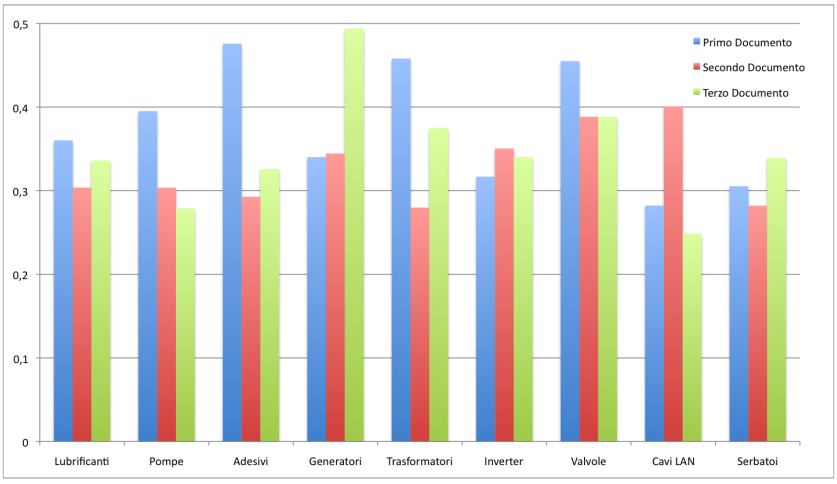


Figura 4.1 Confronto della funzione Score nei diversi Topic per la Conoscenza a-priori.

L'insieme dei concetti e delle parole così ottenute è stato poi utilizzato sia in *iSoS*, sfruttando chiaramente anche la potenza dei link tra *concetti* e *concetti-parole* (ovvero i rispettivi pesi ottenuti con l'approccio probabilistico indicato in precedenza) che all'interno di *Lucene* per effettuare una ricerca di tipo puramente sintattica. L'elenco delle parole coinvolte nei rispettivi topic è mostrato di seguito in tabella 4.5 e 4.6:

4.2.2 Fase di judgments e risultati sintetici

Durante questa fase sono stati valutati da 9 esseri umani i primi 100 risultati proposti da entrambi i motori di ricerca per i rispettivi nove scenari di testing indicati seguendo formalmente gli schemi xml che riportano le intenzioni sulle query. Dato quindi che il numero delle valutazioni per ogni topic ed il numero di topic stesso è relativamente basso, gli esseri umani hanno potuto valutare tutti i risultati ottenuti anche in contrasto al *Minimum Test Collection method* [10].

In tabella 4.7 viene mostrata la statistica dei risultati calcolata dopo le valutazioni degli esseri umani normalizzata al massimo e il minimo ottenuti indipendentemente dal motore e dal topic. Tale valutazione è basata su logica a tre livelli: *rilevante*, *molto rilevante* e *non rilevante* ed ha quindi indicato la qualità dei risultati per entrambi i motori

Query	Concept/Word
Lubrificanti	addit, agent, antioxid, antiwear, api, appli, aromat, automot,base, bear, characterist, classif, classifi, composit, corros, distil,engin, ep, finish, fluid,good, grade, greas, group, heat, high, higher, hydraul, hydrodynam, includ, inhibitor, load, lubric, make, manufactur, meet, naphthen, oil, oxid, perform, previous, purpos, refer, societi, solid, stock, synthet, tabl, temperatur, test, thick, viscos, water, wear
Pompe	air, call, centrifug, check, compon, convers, curv, cylind, discharg, displac, doubl, due, electr, end, ensur, figur, fluid, function, handl, head, higher, improv, increas, industri, kinet, larg, liquid, lower, mechan, mix, oper, option, peristalt, pipe, point, posit, practic, pressur, pump, radial, ram, raw, reduc, repair, resist, rotat, servic, shaft, shape, side,speed, stationari, suction, suppli, total, trap,turbin, type, typic, util, valv, vane, water
Adesivi	addit, adhes, appli, applic, bond, case, cohes, common, consist, cure, depend, design, dri, failur, forc, form, fractur, glass, glue, hot, improv, joint, label, locat, made, materi, onlin, paper, part, pressur, raw, reaction, reactiv, remain, resin, resist, sealant, sensit, servic, solvent, strength, structur, surfac, test, water
Generatori	acceler, air, applic, batteri, circuit, compon, condit, connect, convers, convert, design, develop, direct, due, effect, effici, electr, energi, engin, equip, exceed, factor, field, frequenc, generat, grid, induct, inher, lag, lead, limit, low, machin, magnet, motor, oper, output, page, paragraph, perform, phase, port, power, practic, process, proport, reduc, relat, result, rotor, show, stator,synchron, system, total, uniti, vehicl, wind
Trasformatori	ac, air, angular, applic, area, case, chang, circuit, coil, common, complet, compon, conductor, configur, connect, constant, construct, contact, core, cost, current, design, direct, eddi, effect, end, field, fill, form, ground,heat, import, induc, insid, insul, lamin, larger, lead, liquid, lower,occur, ondari, open, oper, phase, power, practic, pressur, primari, produc, rate, ratio, refer, reson, secondari, section, side, sine, singl,tap, transform, turn, type, vari, voltag, wind, work

Tabella 4.5 Word List per la ricerca in Lucene

Query	Concept/Word
Inverter	aa, ac, accident, batteri, bias, circuit,circuitri, control, current,cycl, d1,damag, dc, dcdcconv, deliv, develop, diod, effici,fair, flow, flyback, frequenc, harmon, high, iinn, increas,induct, input,isol, lead, limit, low, main, maintain, network, oper, outlet, output,process, produc,protect, provid, q1, refer, result, sens, singl, small,smaller, sourc, suppli, switch, time, transfer, turn, vari, vconvert,viinn, voltag, voouutt, waveform, width
Valvole	abbrevi, actuat, automat, back, ball, brass, case, check, close, common, condit, control,design, devic, diaphragm, differ, drop, fail, flow, general,includ, inlet, intern, linear, liquid, low, manual, mean, mechan, open, oper, order, outlet, pipe, piston, plug, port, process, rotari, seal, seat, slurri, societi, spring, system, vacuum, valv
Cavi LAN	10, 40, 1000base-t, 100base-tx, 802.3, articl, blue, bnc, cabl, carri, case, categori, center, coaxial, colour, communic, conductor, connect, connector, copper, cross, descript, drain, electr, end, ethernet, fast, fiber, fig, foil, gbit, ground, hub, ieee, inform, instal, lan, length, light, male, manchest, maximum, mbps, multi, note, pair, pair, pc, plastic, radio, requir, run, scheme, segment, short, show, signal, solid, spec, specif, speed, standard, stp, thick, twist, unshield, vendor, wire, wireless
Serbatoi	base, bottom, build, call, capabl, coat, concret, construct, contain, excess, fill, fix, flash, flow, foundat, frequent, fuel, full,gas, general, high, hose, inch, intern, larg, line, liquid, low, mainten, maker, milk, miscibl, pipe, pressur, protect, protect, refineri, regular, requir, reservoir, riser, roof, standard, steel, storag, suppli, surfac, tank, time, type, water

Tabella 4.6 Word List per la ricerca in Lucene

run&T	Doc Ril	Score Max	Score Min	Δ Score	E(Score)	σ(Score)
iSoS 1	55	0.83	0.08	0.75	0.1786	0.0167
Lux 1	35	0.35	0.06	0.29	0.0975	0.0023
iSoS 2	62	0.42	0.09	0.33	0.1427	0.0032
Lux 2	37	0.22	0.09	0.13	0.1208	0.0006
iSoS 3	67	0.80	0.17	0.63	0.3515	0.0510
Lux 3	44	0.29	0.12	0.17	0.1681	0.0021
iSoS 4	48	0.90	0.07	0.83	0.1424	0.0103
Lux 4	61	0.46	0.09	0.37	0.1389	0.0023
iSoS 5	36	1.00	0.06	0.94	0.1947	0.0443
Lux 5	31	0.39	0.11	0.28	0.1483	0.0026
iSoS 6	33	0.65	0.13	0.52	0.2534	0.0165
Lux 6	35	0.19	0.03	0.16	0.0629	0.0011
iSoS 7	76	0.86	0.14	0.72	0.2179	0.0157
Lux 7	57	0.36	0.10	0.26	0.1428	0.0019
iSoS 8	24	0.76	0.00	0.76	0.0432	0.0083
Lux 8	27	0.28	0.02	0.26	0.0388	0.0013
iSoS 9	43	0.83	0.16	0.67	0.2688	0.0135
Lux 9	16	0.18	0.09	0.09	0.1095	0.0002

Tabella 4.7 Distribuzione dello Score con *iSoS* e *Lucene* per i diversi Topic

fornendone così la loro distribuzione. La colonna che indica il numero dei documenti rilevanti - *Doc Ril* - è ottenuta sommando le valutazioni "*rilevante*" e "*molto rilevante*". Si può notare quindi che su tutti i topic analizzati, il motore *iSoS* raggiunge un valore di *Score* più alto rispetto a *Lucene*, indice di una maggiore uniformità dei risultati proposti con i documenti/query ricercate.

Riteniamo molto importante riportare all'interno della tabella riassuntiva 4.8 le valutazioni effettuate dagli utenti per i diversi topic in grado di far apprezzare ulteriormente la qualità dei nostri risultati. Partendo dai dati e dalle valutazioni così ottenute, è possibile calcolare, come proposto altresì in letteratura (*vedi bibliografia*), i diversi indici di prestazione che si basano sostanzialmente su variazioni di *Precision* e *Recall* standard. Prima di scendere nel dettaglio delle prestazioni dei due sistemi analizzati, si riporta in tabella 4.9 solo a titolo di esempio, il formato dei risultati valutati da esseri umani - così come proposto in *TREC* - per evidenziare le potenzialità della funzione *Score* in grado, tra le altre cose, di far apprezzare le diverse sfumature dei singoli record

	Lubrificanti		**Pompe**		**Adesivi**	
	iSoS	Lucene	iSoS	Lucene	iSoS	Lucene
Rilevante	30	28	43	19	58	38
Molto Rilevante	25	7	19	18	9	6
Non Rilevante	45	65	38	63	33	56
	Generatori		**Trasformatori**		**Inverter**	
	iSoS	Lucene	iSoS	Lucene	iSoS	Lucene
Rilevante	33	48	12	18	27	29
Molto Rilevante	15	13	24	13	6	6
Non Rilevante	52	39	64	69	67	65
	Valvole		**Cavi LAN**		**Serbatoi**	
	iSoS	Lucene	iSoS	Lucene	iSoS	Lucene
Rilevante	54	29	16	19	26	12
Molto Rilevante	22	28	8	8	17	4
Non Rilevante	24	43	76	73	57	84

Tabella 4.8 Sintesi delle valutazione degli utenti per i diversi topic

o, detto in altre parole, quanto due documenti sono vicini tra loro. Si coglie l'occasione per sottolineare che attraverso un uso opportuno di questa funzione si potrebbe implementare anche un sistema automatico di classificazione dei documenti capace di fare un *clustering* di un intero repository in base alla conoscenza fornita dalle ontologie di dominio *iLO*. Nelle tabelle 4.10 e 4.11 sono mostrati rispettivamente gli indici di prestazione calcolati secondo il formato TREC per i singoli topic separatamente e come insieme attraverso il valor medio. I valori indicati in esse sono legati ad indicatori di precisione e richiamo presenti in letteratura, come esplicitato dalle formule 4.2,4.3,4.4,4.5,4.6,4.7,4.8.

$$eAP = \frac{1}{ER}\sum_{i=1}^{k}\frac{x_i}{i} + \sum_{j>i}\frac{x_i x_j}{j} \tag{4.2}$$

$$ePrec@k = eP@k = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{4.3}$$

$$ERprec = \frac{1}{ER}\sum_{i=1}^{ER} x_i \tag{4.4}$$

$$ER = \sum_{i=1}^{n} x_i \tag{4.5}$$

Topic	Uso Futuro	Documento	Rank	Score Normalizzata	Processo
2	Q0	20447	1	1,0000000	iSoS
2	Q0	11605	2	0,8975487	iSoS
2	Q0	27900	3	0,8580963	iSoS
2	Q0	125702	4	0,7580287	iSoS
2	Q0	108742	5	0,6571187	iSoS
2	Q0	131769	6	0,6093244	iSoS
2	Q0	15241	7	0,5894318	iSoS
2	Q0	52347	8	0,5818480	iSoS
2	Q0	7521	9	0,5510380	iSoS

Tabella 4.9 Formato Trec per i risultati di entrambi i motori

$$DCG_x = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2 (1 + i)} \tag{4.6}$$

$$nDCG_x = \frac{DCG_x}{IDCG_x} \tag{4.7}$$

$$CG_x = \sum_{i=1}^{n} rel_i \tag{4.8}$$

Dove eAP indica la precisione media su di un topic, x_i e x_j rappresentano degli indicatori booleani di rilevanza, k è la cardinalità dell'insieme preso in considerazione ($k = 100$), ER sottoinsieme di documenti rilevanti[13], $eMAP$ la media di eAP sui diversi topic, $eP@k$ la precisione calcolata con k livello di analisi (k è il numero di documenti ordinati da prendere in considerazione - Esempio: $eP5$ è la precisione calcolata considerando i primi 5 risultati dei motori di ricerca). In tabella 4.12 vengono mostrate altre due misure standard di prestazione che portano in conto la qualità dei risultati correlata alla posizione in cui essi si sono presentati.

In particolare il *Cumulative Gain* indica quanti e con che livello di rilevanza si sono verificati – Esempio: $iSoS@_{T=1} => 25(Moltorilevanti)*2+30(Rilevanti)*1 = 80$. Per poter confrontare i due sistemi di ricerca

[13]Il valore di ER dipende dall'intersezione dell'insieme dei risultati di entrambi i motori di ricerca: corrisponde alla somma del numero dei risultati rilevanti e molto rilevanti presenti in $R_{iSoS} \cup R_{Lucene} - R_{iSoS} \cap R_{Lucene}$

run-T	eR	eAP	eR prec	eP5	eP10	eP15	eP20	eP30	eP100
iSoS 1	64	0.5939	0.7031	1.00	0.7778	0.7143	0.7368	0.5862	0.5455
Lux 1	64	0.3298	0.4062	0.75	0.6667	0.6429	0.7368	0.6552	0.3535
iSoS 2	76	0.5605	0.5921	1.00	1.0000	0.8571	0.7368	0.6897	0.6263
Lux 2	76	0.2542	0.3947	0.75	0.6667	0.7143	0.6316	0.5517	0.3737
iSoS 3	75	0.7402	0.7200	1.00	1.0000	1.0000	1.0000	0.7931	0.6667
Lux 3	75	0.3658	0.4400	0.50	0.7778	0.8571	0.8947	0.6207	0.4444
iSoS 4	73	0.5009	0.5890	1.00	0.6667	0.7857	0.8421	0.8621	0.4848
Lux 4	73	0.6826	0.6575	0.75	0.8889	0.9286	0.9474	0.8276	0.6162
iSoS 5	49	0.4840	0.4694	1.00	0.8889	0.9286	0.8421	0.5517	0.3636
Lux 5	49	0.2991	0.4286	1.00	0.5556	0.3571	0.3684	0.3793	0.3131
iSoS 6	38	0.5903	0.6053	0.75	0.7778	0.7857	0.8421	0.7241	0.3333
Lux 6	38	0.6740	0.6579	0.75	0.8889	0.9286	0.8947	0.7241	0.3535
iSoS 7	99	0.6210	0.7576	1.00	0.8889	0.7857	0.8421	0.8276	0.7576
Lux 7	99	0.3960	0.5657	1.00	0.6667	0.6429	0.6316	0.6207	0.5657
iSoS 8	28	0.3182	0.3214	0.50	0.5556	0.4286	0.3158	0.3448	0.2424
Lux 8	28	0.4648	0.3929	1.00	0.5556	0.5714	0.4737	0.3793	0.2727
iSoS 9	45	0.7348	0.6667	1.00	1.0000	0.9286	0.8947	0.7931	0.4343
Lux 9	45	0.1456	0.1556	0.75	0.5556	0.4286	0.3684	0.2414	0.1616

Tabella 4.10 Indici di prestazioni sui diversi topic per entrambi i motori di ricerca

run	eMAP	eRprec	eP5	eP10	eP15	eP20	eP30	eP100
iSoS	0.5715	0.6027	0.9167	0.8395	0.8016	0.7836	0.6858	0.4949
Lux	0.4013	0.4555	0.8056	0.6914	0.6746	0.6608	0.5556	0.3838

Tabella 4.11 Valori medi degli indici di prestazioni sui diversi topic per entrambi i motori di ricerca

è necessario calcolare il **nDCG** sfruttando l'*Ideal DCG* ottenuto considerando il ranking ottimo (ideale) tra i risultati proposti dai singoli motori: dato il grado di rilevanza 2 associato al "*Molto rilevante*", 1 associato al "*Rilevante*" e 0 al "*Non rilevante*", bisogna quindi calcolare il IDCG posizionando ai posti più alti i documenti più rilevanti. È opportuno sottolineare però che questa misura soffre della qualità massima dei risultati, ovvero si fa notare che il *DCG* e *IDCG* di iSoS sono sensibilmente più alti di quelli mostrati in *Lucene* in tutti i contesti e quindi si deve necessariamente concludere che se è vero che le prestazioni sono confrontabili attraverso questi indicatori, in *iSoS* i documenti

run-T	Ril	CG	Discounted CG	Ideal DCG	nDCG
iSoS 1	55	80	25.5356	30.0302	0.8503
Lux 1	35	42	15.5023	17.4208	0.8899
iSoS 2	62	81	24.25712	28.5772	0.8488
Lux 2	37	55	17.0527	23.6897	0.7198
iSoS 3	67	76	18.5679	24.2882	0.7644
Lux 3	44	50	12.0480	18.4353	0.6535
iSoS 4	48	63	19.3302	24.2669	0.7965
Lux 4	61	74	21.3520	25.4954	0.8374
iSoS 5	36	60	23.7140	26.1747	0.9060
Lux 5	31	44	16.3302	20.0723	0.8136
iSoS 6	33	39	10.6976	16.3657	0.6536
Lux 6	35	41	11.0685	16.7541	0.6606
iSoS 7	76	98	25.4047	32.2047	0.7888
Lux 7	57	85	23.7480	31.6206	0.7510
iSoS 8	24	32	11.8173	15.8259	0.7467
Lux 8	27	35	12.3693	16.4570	0.7516
iSoS 9	43	60	20.9774	24.3355	0.8620
Lux 9	16	20	8.7752	11.2292	0.7814

Tabella 4.12 Cumulative Gain (CG), Discounted Cumulative Gain (DCG), Normalized Discounted Cumulative Gain (nDCG)

rilevanti sono numericamente superiori.

Di seguito si riporta una rappresentazione grafica della tabella 4.13 per evidenziare le differenze tra i due sistemi in termini di *nDCG*: è possibile così notare che mediamente (l'area sottesa dalle curve) il valore legato al nostro *iSoS*, quando positivo, supera di molto l'analogo in *Lucene*, cosa che non succede quando *Lucene* è "migliore" di *iSoS* (parte negativa del grafico).

4.2.3 Prime Conclusioni

Come si può chiaramente apprezzare dalle tabelle introdotte in precedenza e quindi dai diversi indicatori calcolati su specifici scenari di interesse, i risultati del nostro motore appaiono molto buoni anche alla luce del fatto che praticamente sono state usate le stesse parole per effettuare le query su entrambi i motori. Questo aspetto non è del tutto scontato in quando stiamo effettuando delle query anche su Lucene con termini estratti dalla nostra procedura automatica. Il punto di forza

Topic	iSoS-Lux
1	-0,039543144
2	0,128995487
3	0,110953647
4	-0,040917151
5	0,092422839
6	-0,006986965
7	0,037824427
8	-0,004912888
9	0,080539143

Tabella 4.13 Differenze tra nDCG per iSoS e Lucene

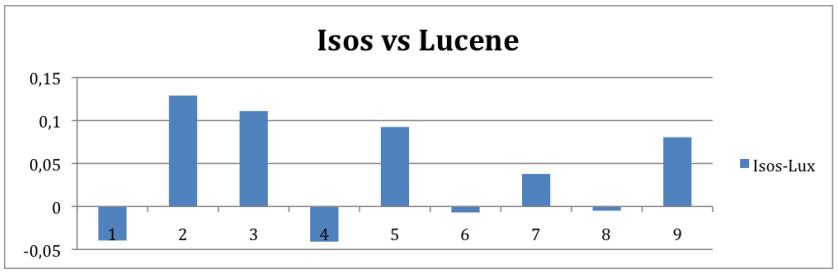


Figura 4.2 Differenze per la qualità dei risultati tra iSoS e Lucene - nDCG a confronto.

del nostro sistema, che si mette qui in risalto, è proprio il boost che si può rilevare introducendo (e quindi pesando) un legame tra concetti: lo stesso legame che si può "apprendere" automaticamente da una serie di documenti che formano la conoscenza intrinseca di base del nostro sistema e che fornisce una sorta di coscienza attiva avanzata nel processo di ricerca.

4.3 Seconda Fase di sperimentazione: la User Satisfaction

Per valutare le prestazioni di un motore di ricerca possono essere prese in considerazione diverse misure standard come il tempo di risposta alle query, la copertura del database, l'index refreshing, la disponibilità di pagine web, lo user effort, l'efficacia recupero, ecc. [4] [52]. Poiché uno degli obbiettivi di questo lavoro di tesi è di evidenziare i miglio-

ramenti nel processo di ricerca in termini di efficienza nel recupero dei documenti dovuti all'introduzione delle ontologie, ci si concentrarà di seguito solo sugli aspetti relativi alla rilevanza dei primi documenti recuperati. In generale, durante la fase di testing di un motore di ricerca, come avviene per qualsiasi altro artefatto, può essere sicuramente utile confrontarsi con diversi altri motori per evidenziare i punti di forza e di debolezza del sistema in prova. Con questa linea di ragionamento, una prima valutazione è stata condotta confrontando il comportamento di $iSoS_{lite}$ con quella del Custom Google Search Engine (CSE)[14]. Ritornando allora alla valutazione dell'efficacia di recupero, i criteri più comunemente utilizzati sono in questo caso la *precisione* e *recall* [52] così come sono stati definiti in precedenza.

Gli enormi domini e la continua evoluzione dei sistemi web rende però impossibile calcolare il valore vero di recall che richiederebbe la conoscenza del numero totale di elementi rilevanti nella raccolta e quindi molto spesso il calcolo risulta approssimato o eventualmente omesso [36]. Per questa ragione si utilizzerà di seguito una recall leggermente modificata così come introdotta da Vaughan in un precedente lavoro [73] che si basa su scala continua in accordo a giudizi forniti da esseri umani [38] che possono essere visti come riferimento ideale per la qualità dei risultati e può fornire un termine di paragone migliore tra i diversi sistemi sotto test.

4.3.1 Scelta dei Topic e delle rispettive query

Quando sono impiegati giudizi umani per valutare la rilevanza di un risultato di un motore di ricerca, una grande varietà di fattori possono influenzare la fase di sperimentazione soprattuto a causa della soggettività del concetto stesso di rilevanza. Precedenti studi, [54] fra tutti, hanno sottolineato che il giudizio di rilevanza può essere eseguito solo dalle persone che hanno originariamente una specifica necessità di informazioni e, di conseguenza, sia l'argomento di interesse che la query di ricerca dovrebbe coinvolgere solo le persone che effettueranno poi le valutazioni. Per questo motivo abbiamo pensato di interpellare 55 persone prima nella fase di definizione dei topic (diverse necessità di informazioni) e poi le stesse nella fase di valutazione dei risultati: ab-

[14] I motivi per utilizzare Google CSE al posto del Google standard sono relative alle limitate capacita di crawling di $iSoS_{lite}$: abbiamo dovuto assicurarci che entrambi i motori svolgessero ricerche sul medesimo corpus di pagine web.

biamo così formato cinque gruppi di undici persone e ciascun gruppo ha definito in lingua italiana uno specifico argomento con la rispettiva query. Abbiamo così definito:

1. Topic: *Arte Rinascimentale.*
   Query: *Arte Rinascimentale* (AR).
2. Topic: *Evoluzione della lingua italiana.*
   Query: *Evoluzione della lingua italiana* (ELI).
3. Topic: *Storia del teatro napoletano.*
   Query: *Storia del teatro napoletano* (STN).
4. Topic: *Storia dell'opera italiana.*
   Query: *Storia dell'opera italiana* (OPI).
5. Topic: *Origini della mozzarella.*
   Query: *Origini della mozzarella di bufala* (OMB).

Per ogni argomento, il motore di ricerca $iSoS_{lite}$ ha scaricato ed indicizzato un migliaio di pagine dal Web e quindi i loro URL sono stati poi utilizzati per programmare contemporaneamente il motore di ricerca personalizzato di Google, permettendo così ad entrambi di effettuare ricerche sul medesimo corpus di documenti.

4.3.2 Costruzione delle ontologie e recupero di pagine web

Allo scopo di alimentare il nostro *ontology builder* con la conoscenza informale contenuta nei diversi testi e quindi produrre le ontologie appropriate per l'interrogazione sui topic individuati, abbiamo coinvolto cinque persone molto qualificate su questi temi e abbiamo chiesto a ciascuno di loro di fornire un insieme di documenti (un numero tra 5 e 10 su ognuno dei 5 argomenti) che meglio potesse rispondere alle richieste degli utenti nel caso delle query da effettuare. Applicando quindi la procedura di apprendimento che abbiamo descritto nel paragrafo 3.2 per ogni set di documenti fornitoci, abbiamo ottenuto una *iLO* per ogni topic e abbiamo effettuato le query su entrambi i motori $iSoS_{lite}$ e *Google CSE*. Per valutare meglio il contributo delle nostre ontologie abbiamo anche effettuato le diverse ricerche su $iSoS_{lite}$ inserendo

		Rank for AR		
BCf	**URLs**	**HB**	**iSoS**	**Google**
1	www.artistiinrete.it	1	**3**	**5**
0,98	www.bilanciozero.net	2	**2**	> 10
0,93	it.encarta.msn.com	3	**5**	**3**
0,90	www.firenze-online.com	4	**1**	> 10
0,88	it.wikipedia.org	5	**4**	**1**
0,72	www.arte.go.it	6	**8**	> 10
0,66	digilander.libero.it	7	**9**	> 10
0,62	www.visibilmente.it	8	**6**	**7**
0,58	www.salviani.it	9	> 10	**4**
0,57	www.arte-argomenti.org	10	**7**	> 10
User Satisfaction			86,1%	65,8%

Tabella 4.14 Arte Rinascimentale

		Rank for ELI		
BCf	**URLs**	**HB**	**iSoS**	**Google**
1	blogs.dotnethell.it	1	**3**	> 10
1	it.wikipedia.org	2	**1**	**1**
0,93	www.letteratour.it	3	**2**	> 10
0,89	www.nonsoloscuola.net	4	**4**	> 10
0,80	idigilander.libero.it	5	**7**	> 10
0,75	xoomer.virgilio.it	6	**6**	> 10
0,67	www.etx.it	7	**8**	> 10
0,62	www.regione.emilia-romagna.it	8	> 10	**10**
0,51	www.tesionline.com	9	> 10	**6**
0,46	www.tesionline.it	10	> 10	**3**
User Satisfaction			83,3%	47,7%

Tabella 4.15 Evoluzione della lingua italiana

semplicemente solo i termini estratti dal builder[15]. Per ogni query, abbiamo allora selezionato/ottenuto 30 pagine corrispondenti alle prime 10 pagine recuperate da ogni motore e, come conseguenza di tale procedura di fusione, il numero di pagine per AR, OPI, ELI, STN e OMB è stato rispettivamente 17, 19, 20, 19 e 17. Ci riferiremo allora a questi set di risultati come *resulting repositories*.

[15]In letteratura si parla in questo caso di query expansion ovvero inserimenti di termini aggiuntivi alla query originale per specializzarne meglio il senso: nel nostro caso abbiamo volutamente tralasciato i rapporti tra termini - peso dei link.

		Rank for OPI		
BCf	**URLs**	**HB**	**iSoS**	**Google**
1	it.wikipedia.org	1	**4**	**10**
0,91	www.jazzplayer.it	2	**2**	> 10
0,83	musicallround.forumcommunity.net	3	**3**	> 10
0,82	www.sonorika.com	4	**5**	> 10
0,75	www.sapere.it	5	**6**	> 10
0,65	www.bookonline.it	6	**1**	> 10
0,62	www.ilpaesedeibambinichesorridono.it	7	**10**	> 10
0,55	www.gremus.it	8	> 10	**1**
0,51	www.bulgaria-italia.com	9	**8**	> 10
0,49	www.rodoni.ch	10	**9**	> 10
User Satisfaction			79,9%	36,1%

Tabella 4.16 Storia dell'opera italiana

		Rank for STN		
BCf	**URLs**	**HB**	**iSoS**	**Google**
1	www.sottoilvesuvio.it	1	**1**	**7**
1	www.blackwikipedia.org	2	**3**	> 10
0,90	it.wikipedia.org	3	**2**	**1**
0,90	xoomer.virgilio.it	4	**5**	**5**
0,89	www.webalice.it	5	**6**	> 10
0,88	www.laboriosi.it	6	**8**	> 10
0,59	www.denaro.it	7	> 10	**4**
0,56	www.gttempo.it	8	**7**	> 10
0,47	www.teatroantico.org	9	**10**	> 10
0,47	azzurrocomenapoli.myblog.it	10	**4**	> 10
User Satisfaction			85,7%	63,4%

Tabella 4.17 Storia del teatro napoletano

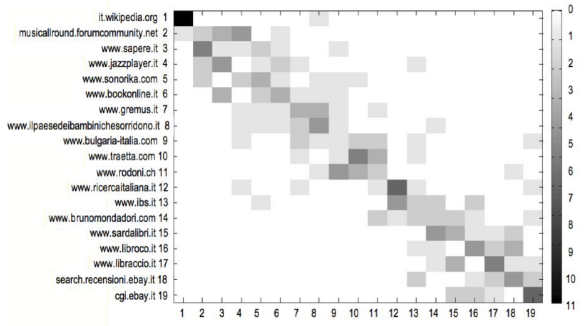
		Rank for OMB		
BCf	**URLs**	**HB**	**iSoS**	**Google**
1	magazine.paginemediche.it	1	**6**	> 10
0,88	www.caseificioesposito.it	2	**1**	> 10
0,78	www.agricultura.it	3	**2**	> 10
0,74	it.wikipedia.org	4	**3**	**2**
0,73	www.mozzarelladibufala.org	5	> 10	**1**
0,72	www.ciboviaggiando.it	6	**9**	**6**
0,62	www.sito.regione.campania.it	7	**8**	**5**
0,59	www.aversalenostreradici.com	8	**7**	> 10
0,55	www.tenutadoria.it	9	**7**	**7**
0,46	www.bortonevivai.it	10	**5**	**8**
User Satisfaction			77,8%	51,4%

Tabella 4.18 Origini della mozzarella di bufala

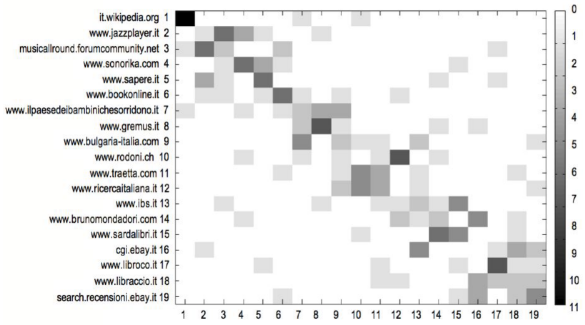
4.3.3 Ranking umano delle pagine Web

I soggetti coinvolti nella nostra sperimentazione sono stati 55 selezionati tra laureati in varie discipline - da Ingegneria (informatica, elettronica, gestionale) a Lettere ed Economia - ed individuati all'interno di un corso universitario post-laurea. Come accennato in precedenza, sono stati suddivisi quindi in cinque gruppi da undici persone ed ognuno di essi ha effettuato una valutazione sul *resulting repositories* collegato ad uno dei cinque quesiti. In particolare, ogni soggetto ha assegnato un grado di apprezzamento ad ogni pagina web visitata in base alla propria valutazione soggettiva dei criteri di rilevanza producendo così una lista ordinata di risultati. In seguito, ogni persona ha incontrato gli altri appartenenti allo stesso gruppo e ha discusso la propria classifica ed i propri criteri al fine di migliorare la qualità del proprio ranking e ridurre così gli effetti soggettivi-individuali presenti nelle proprie valutazioni. Risultato di questa ulteriore discussione è stato un nuovo elenco revisionato di pagine web per ogni persona.

Per ottenere un'aggregazione consistente dei dati, è risultato opportuno decidere su di un comportamento medio delle persone. Un approccio previsto in letteratura in grado di risolvere questo problema è senz'altro il metodo di Borda Count [63] applicato agli elenchi dei ri-



(a)



(b)

Figura 4.3 Risultati espressi attraverso il Borda Count prima della discussione (unbiased) 4.3(a) e dopo la discussione (biased) 4.3(b) rispetto al topic *OPI*.

sultati sia prima che dopo la discussione tra le persone. Le figure 4.3(a) e 4.3(b) mostrano quindi una rappresentazione grafica dei risultati di tale metodo riferito al topic OPI pre e post discussione. Tale rappresentazione fornisce un modo semplice di indagine per quel che riguarda il grado di accordo dei diversi soggetti per ogni risultato: graficamente possiamo dire che il colore nero indica il massimo della coerenza (ovvero gli undici soggetti valutatori concordano sul ranking delle singole pagine) mentre il bianco ne rappresenta il minimo (nessuno ha votato per quella posizione).

Analizzando quindi i risultati riportati nella figura 4.3(b), è possibile notare che le caselle più scure sono allineate lungo la diagonale del grafico e quindi il Borda Count in questo caso suggerisce il ranking finale effettivo. Bisogna comunque sottolineare la nostra scelta sul numero di pagine da prendere in considerazione per la valutazione

finale dei risultati. Coerentemente a ciò che viene illustrato in letteratura [68], abbiamo quindi deciso di concentrare la nostra attenzione solo sulle prime 10 pagine anche perché generalmente gli utenti visitano solo queste tra i risultati recuperati dai motori di ricerca. Al fine di rendere esplicito il consenso delle persone per ogni gruppo, calcoliamo il fattore di Borda Count (BCF) per ogni query q come $f_i^q = \frac{n_i}{n_1}$, dove n_i è il numero di persone che ha deciso di assegnare tale pagina alla posizione *i-esima*. Le tabelle 4.14, 4.15, 4.16, 4.17 e 4.18 mostrano i primi 10 risultati classificati dalle persone per ogni query evidenziando la diverse posizione di $iSoS_{lite}$ e *Google CSE* per ogni singola pagina.[16].

4.3.4 Valutazione dei risultati di ricerca attraverso Precision e Recall

Possiamo affermare allora che la *Precision* è una misura sempre presente nei problemi formali di recupero delle informazioni. Come descritto in precedenza, essa è definita come la frazione di documenti recuperati che sono contemporaneamente anche rilevanti e che quindi dipendono direttamente da come viene fatto il giudizio stesso di rilevanza. Molto spesso per questo scopo vengono utilizzati giudizi di tipo binario *on-off*[17], ovvero un documento è rilevante o non rilevante per il topic fissato[74] senza alcuna possibilità di sfumatura. Diversi studi hanno mostrato anche l'utilizzo di giudizi di rilevanza a più livelli piuttosto che binari ma in tutte queste valutazioni discrete si soffre la possibilità di conferire lo stesso punteggio a documenti diversi e quindi risultano poco utili quando il campione è un lista di risultati ordinati. In questo lavoro di tesi è opportuno risottolineare che verranno utilizzate solo alcune varianti di *Precision* e *Recall* che si basano sostanzialmente sui risultati classificati dagli esseri umani. In particolare per questa prima sperimentazione, considereremo come rilevanti solo i primi 10 risultati proposti dagli esseri umani in modo che risulterà comodo evitare il calcolo, nell'ambito dell'intero corpus, del valore reale dei documenti

[16] Gli URL completi delle pagine sono riportati in appendice A.2 e si ricorda che tali pagine sono state scaricate nel dicembre 2009. Per le tabelle 4.14,4.15,4.16,4.17,4.18 si indicherà con BC il fattore di Borda Count e HB l'Human Behavior

[17] Nella sezione successiva si mostrerà che anche in esperimenti e competition prestigiose - TREC fra tutte - viene implementata una procedura di giudizio di questo tipo

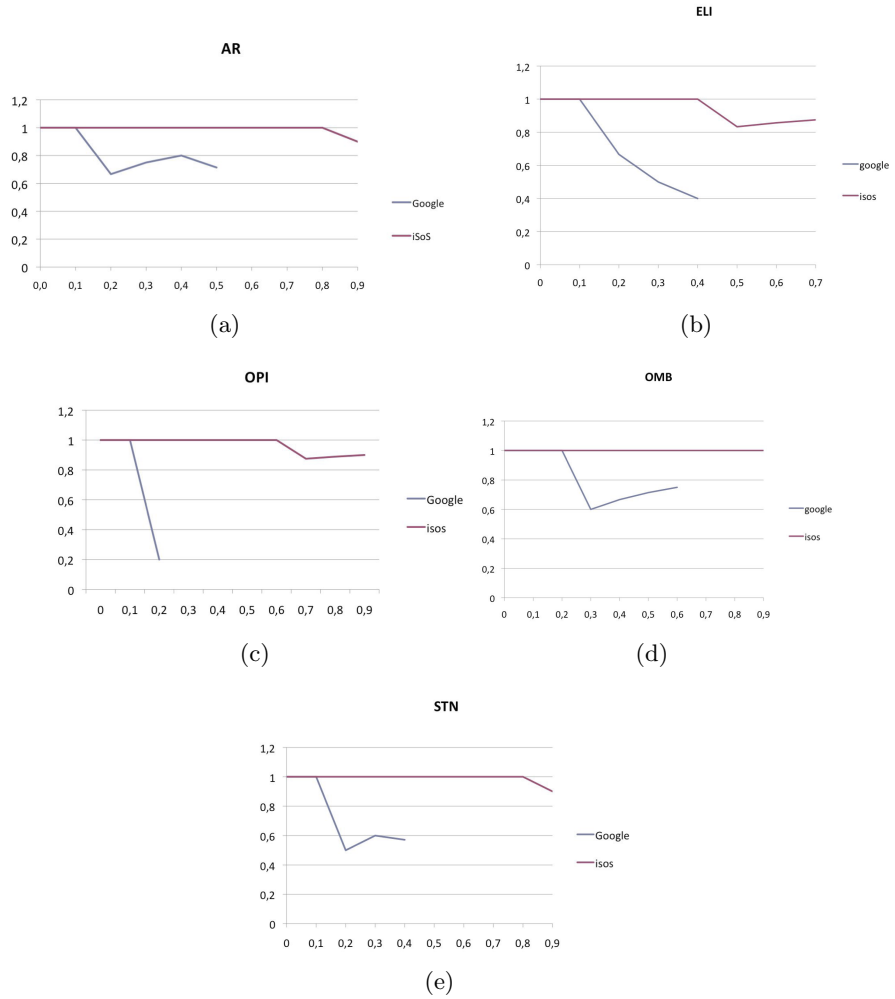


Figura 4.4 Precision - Recall per le diverse queries: 4.4(a) AR, 4.4(b) ELI, 4.4(c) OPI, 4.4(d) OMB, 4.4(e) STN

rilevanti data chiaramente la complessità stessa della valutazione: si deve considerare come nostro obiettivo il confrontare $iSoS_{lite}$ con *Google CSE* avendo come riferimento il comportamento/gradimento umano. Per ogni set di risultati sono mostrati i valori di *Precision* e *Recall* attraverso una curva interpolata cosicché la precisione interpolata p_{interp} ad un livello certo di recall r è definita come la più alta precisione individuata per ogni livello di recall $r' \geq r$, [52]:

$$p_{interp}(r) = \max_{r' \geq r} p(r')$$

Il modo tradizionale di rappresentare una tale curva è la *11-point interpolated average precision*, ovvero per ogni information need, la curva viene valutata con 11 livelli di recall r' pari a $0.0, 0.1, 0.2, ..., 1.0$: per ogni livello di recall abbiamo quindi calcolato la media aritmetica della precisione interpolata a quel determinato livello di recall in tutto il corpus. Le figure 4.4(a), 4.4(b), 4.4(c), 4.4(d) e 4.4(e) mostrano i grafici interpolati di *Precision - Recall* per le query AR, ELI, OPI, OMB e STN rispettivamente. Si può notare che per tutti i casi esaminati $iSoS_{lite}$ mostra un comportamento migliore di *Google CSE*, con OPI caso migliore. Tuttavia, come detto in precedenza, tali misure di *Precision* e *Recall* si basano su giudizi di rilevanza binari e quindi non possiamo ritenerli sufficientemente accurati ed oggettivi per effettuare un buon confronto: nella sezione successiva proponiamo quindi una metodologia alternativa per confrontare i due sistemi, $iSoS_{lite}$ con *Google CSE* sempre basata su comportamento/giudizio umano.

4.3.5 Valutazione della User Satisfaction

Una valutazione significativamente migliore delle performance può essere allora eseguita confrontando i risultati proposti da $iSoS_{lite}$ e quelli di *Google CSE* con il ranking umano che noi consideriamo come comportamento di riferimento. Per riuscire in questo confronto è conveniente calcolare ed aggregare diverse misure di pertinenza partendo dal fatto che per ogni query possiamo sfruttare i giudizi di rilevanza compresi tra 0 e 1, almeno per i primi 10 risultati ottenuti grazie al fattore di Borda Count f_i^q (si veda la colonna BCf in tabella 4.14). Si definisce quindi la *User Satisfaction* come semplice posizione media ponderata dei risultati, in cui per posizione vogliamo chiaramente intendere il posto dove il risultato si è presentato: primo, secondo e così via per la determinata query, mentre il peso considerato sarà il fattore $1/position$. Se indichiamo i posizionamenti del sistema in prova come pos_1, pos_2, pos_3, ecc.[39] allora possiamo scrivere:

$$US(q) = \frac{\sum_{n=1}^{10} \text{pos}_n / n}{\sum_{n=1}^{10} 1/n} \tag{4.9}$$

			Rank for AR	
BCf	**URLs**	**HB**	**iSoS**	**iSoS$_{qe}$**
1	www.artistiinrete.it	1	**3**	**5**
0,98	www.bilanciozero.net	2	**2**	**7**
0,93	it.encarta.msn.com	3	**5**	**3**
0,90	www.firenze-online.com	4	**1**	**8**
0,88	it.wikipedia.org	5	**4**	**4**
0,72	www.arte.go.it	6	**8**	**10**
0,66	digilander.libero.it	7	**9**	> 10
0,62	www.visibilmente.it	8	**6**	> 10
0,58	www.salviani.it	9	> 10	> 10
0,57	www.arte-argomenti.org	10	**7**	> 10
User Satisfaction			86,1%	71,6%

Tabella 4.19 Confronto tra $iSoS_{lite}$ con $iSoS_{lite}qe$

dove consideriamo $\text{pos}_n = f_i^q$ se il sistema pone l' $i$esimo risultato in posizione n.

In base a questa formulazione, la tabella 4.14 fornisce ad esempio le percentuali di $US(q)$ per la query AR eseguita da $iSoS_{lite}$ e da *Google CSE* utilizzando i numeri in grassetto per indicare i risultati di ricerca appartenenti al set di documenti considerati rilevanti per esseri umani. Questa rappresentazione ci permette allora di fare un confronto visivo rapido tra i diversi comportamenti e dedurre quindi che il sistema che presenta il numero più elevato di valori in grassettoè probabilmente più vicino ad un comportamento umano ideale. Chiaramente questo tipo di analisi è stata condotta anche per le altre queries così come presentato nelle tabelle 4.15, 4.16, 4.17 e 4.18 dove si può osservare che $iSoS_{lite}$ esibisce un comportamento mediamente migliore per tutti i casi con OPI picco migliore. Anche se questi risultati sembrano simili a quelli di *Precision* e *Recall*, va detto che questo metodo aiuta ad evidenziare anche le debolezze di $iSoS_{lite}$ come può essere visto, per esempio, negli ultimi tre risultati della query ELI (tabella 4.15).

4.3.6 Considerazioni sulle Query Expansion

Si potrebbe sostenere che l'introduzione delle iLO agisce come niente più di una *query expansion*, il che significa che più parole-chiave vengono aggiunte al processo di ricerca. Per chiarire questo aspetto, abbiamo fatto un confronto tra il comportamento di $iSoS_{lite}$ con l'uso di iLO e il comportamento di $iSoS_{lite}$ con la semplice aggiunta dei termini

dell'ontologia[18]. I risultati così ottenuti sono mostrati in tabella 4.19 dove si può notare che l'uso di ontologie produce un migliore ranking coerentemente con l'idea che una coppia di parole porta in sé aspetti di semantica maggiori rispetto al singolo termine.

4.4 Conclusioni seconda fase di esperimenti

In questo seconda fase di sperimentazione abbiamo mostrato come le prestazioni di un classico motore di ricerca web, in termini di qualità di risultati recuperati eseguendo informational queries, può essere sensibilmente migliorata attraverso l'utilizzo di una tecnica di ricerca innovativa basata su *informal Lightweight Ontology*. Più nel dettaglio, il metodo proposto è in grado di recuperare pagine più vicini alle intenzioni degli utenti e quindi migliorare il livello complessivo di soddisfazione degli stessi. La logica dietro al nostro metodo risponde all'idea che certamente l'intenzione di un utente alla ricerca di informazioni su un determinato argomento, per esempio*Arte Rinascimentale*, può essere ben conservata in alcuni documenti testuali (pagine web incluse) e che quindi se avessimo quei documenti, potremmo pensare di estrarre la conoscenza da essi e rappresentarla successivamente con una ontologia [19]. Passo finale di questa idea è chiaramente utilizzare tale conoscenza per specificare meglio le queries sullo specifico argomento. Per fare tutto ciò, il cuore della nostra proposta è sicuramente la definizione di un tipo di conoscenza informale, da noi chiamata *informal Lightweight Ontology*, che può essere dedotta automaticamente da documenti testuali.

Una simile rappresentazione della conoscenza può essere definita meglio come un Grafo di Concetti, ovvero una struttura composta da nodi (i concetti stessi) e link ponderati tra essi (in piedi per le relazioni semantiche tra concetti). Ogni concetto può anche essere definito con un ulteriore grafo di tipo gerarchico, aciclico e formato da parole che specializzano il concetto stesso. I pesi di entrambi i grafi possono essere allora interpretati e visti attraverso delle probabilità ovvero appresi at-

[18] Non siamo riusciti a fare un confronto in questi termini con *Google CSE* perché il motore non ha restituito alcun risultato con una tale query expansion.

[19] Il termine ontologia, come è stato spiegato sin dall'inizio di questo lavoro è usato in modo improprio o comunque in un'accezione più vicina al contesto filosofico che scientifico.

traverso una tecnica di tipo probabilistica: nel nostro sistema abbiamo usato una versione smoothed della Latent Dirichlet allocation, meglio noto come il Topic Model. La tecnica proposta è stata quindi sviluppata come applicazione web-based attraverso Java e Java Server Pages ed include una versione personalizzata di API Apache Lucene che sono di tipo open source.

E' stato prima convalidata attraverso un confronto delle performance (*Precision* e *Recall* su tutte) con una versione personalizzata di un motore di ricerca web (*Google*) e poi è stato effettuato un confronto sulla base del giudizio umano al fine di valutare $iSoS_{lite}$ su scala continua. Gli esperimenti sono stati condotti in diversi contesti e, per ogni contesto, abbiamo chiesto a diversi gruppi di persone di assegnare giudizi di rilevanza per l'insieme di pagine web raccolte dai risultati ottenuti da entrambi i motori di ricerca. Abbiamo concluso che l'efficacia del nostro metodo è dovuto soprattutto al fatto che la rappresentazione della volontà dell'utente e quindi del significato per come lo abbiamo inteso e spiegato all'inizio di questo lavoro, non è solo impostata attraverso la query ma anche considerando ontologie di tipo informali. Tali ontologie sono praticamente concetti e collegamenti pesati tra di essi in grado di specializzare meglio l'intenzione: approccio utile a ridurre problemi inerenti l'ambiguità del linguaggio così come introdotto e discusso nel Capitolo 1.

Capitolo 5

Conclusioni e sviluppi futuri

Materiali testuali sono fonti estremamente preziose di informazioni per le quali è necessaria una buona e corretta analisi per evitare interpretazioni soggettive dei contenuti. Come è noto, qualsiasi processo di scrittura può essere pensato come un processo di comunicazione in cui l'attore principale, vale a dire lo scrittore, codifica le sue intenzioni attraverso il linguaggio. Il linguaggio allora può essere considerato come un codice che esprime quello che definiamo *significato* al lettore che ne realizza poi il processo di decodifica. A causa delle imperfezioni accidentali del linguaggio stesso, ambiguità in primis, sia il processo di codifica che di decodifica risultano corrotti da *rumore*. In questa ottica, alcune comunità scientifiche, Semantic Web e Knowledge Engineering tra tutte, hanno introdotto diversi strumenti e linguaggi per la descrizione semantica e provare quindi ad evitare le ambiguità insite in questi tipi di comunicazione.

I risultati di questi dibattiti sono stati utilizzati per introdurre nuovi linguaggi formali grazie ai quali e' stato possibile sia rappresentare la semantica stessa su un calcolatore che manipolarla per l'esecuzione di ragionamenti automatici. Seguendo queste discussioni, sono stati altresì introdotti linguaggi specifici, RDF (Resource Description Framework), OWL (Ontology Web Language), ecc., per sostenere il creatore dei documenti (scrittore) a descrivere di proprio pugno le relazioni semantiche tra concetti/parole, vale a dire ad esempio attraverso l'aggiunta di dati su dati-documenti, ovvero generazione di *metadati*. Durante tali pro-

cessi di creazione, tutti gli elementi *rumorosi* vengono così evitati con l'impiego di una conoscenza condivisa, ovvero riconosciuta dagli attori della comunicazione stessa, e rappresentata attraverso strutture formali e non, quali ad esempio possono essere le ontologie. L'analisi testuale (TA) che fa uso poi di tecniche statistiche assicura anche l'esplorazione sistematica della struttura del testo (dimensioni, occorrenze, ecc) e contemporaneamente concede la possibilità, in qualsiasi momento, di tornare al testo originale e consentirne una successiva interpretazione.

In questo lavoro di tesi si è proposta una nuova tecnica basata su un modello probabilistico del linguaggio noto in letteratura come *Topic Model* in grado di analizzare documenti di testo e inferire una rappresentazione specifica del significato contenuto in esso. La nostra rappresentazione è quindi un Grafo di termini, indicato come *informal Lightweight Ontology*, per estrarre il senso/conoscenza da diversi documenti testuali ed utilizzare tale significato/grafo in uno scenario reale di Text Retrieval. Una simile rappresentazione può essere definita meglio allora come un Grafo di concetti, ovvero una struttura composta da nodi (i concetti stessi) e link ponderati tra essi (in piedi per le relazioni semantiche tra concetti). Ogni concetto può anche essere definito con un ulteriore grafo di tipo gerarchico, aciclico e formato da parole che specializzano il concetto stesso. I pesi di entrambi i grafi possono allora essere interpretati come delle probabilità ovvero appresi attraverso una tecnica di tipo probabilistica: nel nostro sistema abbiamo usato una versione smoothed della Latent Dirichlet allocation, meglio noto come il Topic Model.

Questa rappresentazione è estratta automaticamente facendo uso della sola conoscenza di tipo *endogena* (ovvero contenuta direttamente e solamente nei testi in esame), in quanto risulta molto difficile e costoso richiedere l'intervento di esseri umani per creare metadati e/o conoscenza esogena.

Al fine di validare l'approccio proposto sono stati analizzati due scenari reali di text retrieval. In un primo momento è stata utilizzata la rappresentazione grafica per derivare automaticamente l'estensione delle queries digitate da utenti in un sistema full text search. In questo caso, infatti, è noto che le prestazioni, misurate in termini di numero di documenti rilevanti restituiti, possono essere sensibilmente aumentate se si usa una tecnica di query expansion per meglio specificare (e quindi disambiguare) l'intenzione dell'utente. In un contesto interattivo, gli

utenti forniscono al sistema i documenti che ritengono rilevanti per lo scopo prefissato, e da questi viene estratto il grafo di termini. Se la rappresentazione estratta è in grado di immagazzinare l'intenzione dell'utente, allora è verosimile che il numero di documenti restituiti, grazie a formulazione di una nuova query, sarà più elevato.

In un secondo momento, abbiamo misurato la soddisfazione degli utenti di un sistema di text retieval quando si utilizza come tecnica di query expansion quella descritta in precedenza. In questo caso non conta solamente l'attinenza o meno dei risultati con l'oggetto della *query* ma quanto questi sono soddisfacenti e legati all'intenzione sita ancora nella mente dell'utente.

Entrambi gli scenari sperimentali sono stati creati grazie ad un applicativo web based scritto in Java e Java Server Pages, $iSoS_{lite}$, che include tra le altre cose, una versione personalizzata delle API open source di un motore di full text search, *Lucene*. Per il primo caso, sono stati selezionati 9 scenari di ricerca e sono state effettuate altrettante query su di un web-repository estratto dal sito http://ww.thomasnet.com. Il sistema è stato confrontato con uno che utilizza come query expansion una rappresentazione più semplice, formata dalla sola lista di parole del grafo, mostrando migliori prestazioni indipendentemente dal contesto.

Nella seconda fase di sperimentazione abbiamo mostrato invece come le prestazioni di un classico motore di ricerca web, in termini di qualità di risultati recuperati relativi ad informational queries, può essere sensibilmente migliorata attraverso l'utilizzo della tecnica di ricerca innovativa basata su *informal Lightweight Ontology*. Più nel dettaglio, il metodo proposto è in grado di recuperare pagine più vicini alle intenzioni degli utenti e quindi migliorare il livello complessivo di soddisfazione degli stessi. E' stato quindi prima convalidato attraverso un confronto standard delle performance (*Precision* e *Recall* su tutte) con una versione personalizzata di un famoso motore di ricerca web (*Google*) e poi è stato effettuato un confronto sulla base del giudizio umano per valutare $iSoS_{lite}$ su scala continua. Gli esperimenti sono stati così condotti in diversi contesti e, per ognuno di essi abbiamo richiesto a diversi gruppi di persone di assegnare giudizi di rilevanza per l'insieme di pagine web raccolte dai risultati ottenuti da entrambi i motori di ricerca.

Grazie alle misure di prestazione del metodo proposto nei diversi scenari reali abbiamo concluso che una rappresentazione basata su un

grafo gerarchico di termini è in grado, meglio di quanto non sia possibile ottenere con una semplice lista di parole, di specificare l'intenzione dell'utente e quindi può essere impiegata per estendere la query originale e limitare quei problemi di ambiguità intrinseca del linguaggio responsabili delle basse prestazioni di sistemi full text search.

Le buone prestazioni dell'approccio discusso in questo lavoro suggeriscono che il grafo possiede una proprietà discriminativa che può essere impiegata per la categorizzazione di grandi collezioni di dati in un contesto supervisionato. In questo caso, avendo un piccolo insieme di documenti etichettati secondo uno specifico insieme di categorie, allora potrebbe essere possibile estrarre delle rappresentazioni basate su *iLO* da ogni singolo set di documenti per ogni categoria, e usarle per assegnare un grado di appartenenza ad ogni documento contenuto in un grande repository non etichettato.

Inoltre, potrebbe essere interessante integrare sistemi di questo tipo con altri basati su conoscenza esogena, per esempio WordNet, per evitare la ridondanza di quei termini, presenti nel grafo, che hanno la stessa valenza semantica, o meglio trasportano lo stesso significato. Questo servirebbe anche per dedurre ulteriore conoscenza latente, quella che per esempio emergerebbe dalla co-presenza di un piccolo insieme di termini che completa la descrizione di un concetto incluso in WordNet in una porzione del grafo.

Appendice A

A.1 Conditional Probability computation

La probabilità condizionata può essere semplificata utilizzando il teorema di Bayes, le assunziondi di *bags of words* ed *exchangeability* così come indicato da Blei et al. e T. L. Griffiths [[7] , [70]]. L'*exchangeability* afferma che le parole sono generate da topic attraverso una distribuzione condizionata fissa e che tali topic sono infinitamente scambiabili all'interno di un documento. Come conseguenza ogni parola w_n di un documento è condizionatamente indipendente dato un topic z_n do ogni altra parola w_{n+1} o, in termini di probabilità possiamo scrivere che

$$P(w_n|w_{n+1}, z_n) = P(w_n|, z_n)$$

$$\begin{aligned}
P(v_i|v_j) &= \sum_{d=1}^{D} P(w_n^d = v_i, \mathbf{w}_d | w_{n+1}^d = v_j) \\
&= \sum_{d=1}^{D} \sum_{k_1=1}^{T} \sum_{k_2=1}^{T} P(w_n^d = v_i, z_n^d = k_1, z_{n+1}^d = k_2, \mathbf{w}_d | w_{n+1}^d = v_j) \\
&= \sum_{d=1}^{D} \sum_{k=1}^{T} P(w_n^d = v_i, z = k, \mathbf{w}_d | w_{n+1}^d = v_j) \\
&= \sum_{d=1}^{D} \frac{P(\mathbf{w}_d)}{P(w_{n+1}^d = v_j)} \sum_{k=1}^{T} P(w_n^d = v_i | w_{n+1}^d = v_j, z = k, \mathbf{w}_d) * ... \\
&... \quad * P(w_{n+1}^d = v_j | z = k, \mathbf{w}_d) P(z = k | \mathbf{w}_d) \\
&= \sum_{d=1}^{D} \frac{P(\mathbf{w}_d)}{P(w_{n+1}^d = v_j)} \sum_{k=1}^{T} P(w_n^d = v_i | z = k, \mathbf{w}_d) * ... \\
&... \quad * P(w_{n+1}^d = v_j | z = k, \mathbf{w}_d) P(z = k | \mathbf{w}_d) \\
&\propto \sum_{d=1}^{D} \sum_{k=1}^{T} P(w_n^d = v_i | z = k, \mathbf{w}_d) * ... \\
&... \quad * P(w_{n+1}^d = v_j | z = k, \mathbf{w}_d) P(z = k | \mathbf{w}_d) \qquad \text{(A.1)}
\end{aligned}$$

A.2 Complete list of URLs undertaken

A.2.1 Table 4.14

1. www.artistiinrete.it/STORIA_ARTE/RINASCIMENTO/rinascimento.htm
2. www.bilanciozero.net/rinascimento/homepage/arte/storia_dell'arte_it.htm
3. it.encarta.msn.com/encyclopedia_761554529/Rinascimento/_(arte).html
4. www.firenze-online.com/artisti-toscani/
5. it.wikipedia.org/wiki/Arte_del_Rinascimento
6. www.arte.go.it/materiali/arte_004.htm
7. digilander.libero.it/mogent/cm/arteB/index.htm
8. www.visibilmente.it/02arts/story/2/rinascimento2.html

9. http://www.salviani.it/vitelli/storia/art_rin.htm
10. www.arte-argomenti.org/saggi/interventi/artista.htm

A.2.2 Table 4.15

1. blogs.dotnethell.it/artblog/10.-Claudio-Marazzini-Breve-storia-della-lingua-italiana-sintesi__8846.aspx
2. it.wikipedia.org/wiki/Evoluzione_della_lingua_italiana_parlata
3. www.letteratour.it/critica/b00gramma01.htm
4. www.nonsoloscuola.net/piccoli_manuali/stlingua.html
5. digilander.libero.it/letteratura/lette_ita/origini/origini.html
6. xoomer.virgilio.it/r.crosio/latino_diff.htm
7. www.etx.it/curiosita/manzoni.htm
8. www.regione.emilia-romagna.it/urp/semplificazione/ragioni/evoluzione.htm
9. www.tesionline.com/intl/thesis.jsp?idt=2500
10. www.tesionline.it/default/tesi.asp?idt=2500

A.2.3 Table 4.16

1. it.wikipedia.org/wiki/Opera_italiana
2. www.jazzplayer.it/lirica.php
3. musicallround.forumcommunity.net/?t=8924885
4. www.sonorika.com/v2/musica/genere/opera/
5. www.sapere.it/tca/MainApp?srvc=vr&url=/4/5219_1
6. www.bookonline.it/Locale_Arena_pag1.html
7. www.ilpaesedeibambinichesorridono.it/opera_italiana.htm
8. http://www.gremus.it/?q=microcosmo_3
9. www.bulgaria-italia.com/bg/forums/topic.asp?TOPIC_ID=2691
10. www.rodoni.ch/busoni/turandotandreafranco.html

A.2.4 Table 4.17

1. www.sottoilvesuvio.it/teatro.html
2. www.blackwikipedia.org/it/wiki/Teatro_napoletano
3. it.wikipedia.org/wiki/Teatro_napoletano
4. xoomer.virgilio.it/golfo37/attori.htm
5. www.webalice.it/galletta.vincenzo/TEATRO.html
6. www.laboriosi.it/index.php?option=com_content&task=view&id=312&Itemid=243
7. http://www.denaro.it/VisArticolo.aspx?IdArt=555567&KeyW=PARTENOPEI
8. www.gttempo.it/Autori007.htm
9. www.teatroantico.org/mostre/defilippo.html
10. azzurrocomenapoli.myblog.it/teatro-napoletano/

A.2.5 Table 4.18

1. magazine.paginemediche.it/it/366/dossier/dietologia/detail_90328_la-mozzarella-la-regina-della-cucina-mediterranea.aspx?c1=25&c2=0
2. www.caseificioesposito.it/storia_mozzarella.html
3. www.agricultura.it/dettagli_azienda.php?ID=96
4. it.wikipedia.org/wiki/Mozzarella_di_Bufala_Campana
5. www.mozzarelladibufala.org/mozzarella.htm
6. www.ciboviaggiando.it/genitin.asp?lan=I&iti=100010
7. www.sito.regione.campania.it/AGRICOLTURA/Tipici/mozzarel-new.html
8. www.aversalenostreradici.com/30.2Mozzarella.htm
9. www.tenutadoria.it/la-nostra-storia.htm
10. www.bortonevivai.it/news/mozzarella-di-bufala-campana.asp

Appendice B

B.0.6 Link-documenti utilizzati per la costruzione della conoscenza a-priori

Lubrificanti

1.txt en.wikipedia.org/wiki/Lubricant

2.txt www.maintenanceworld.com/Articles/johnsonr/BestPractice3.pdf

3.txt 140.194.76.129/publications/eng-manuals/em1110-2-1424/c-13.pdf

Pompe

1.txt en.wikipedia.org/wiki/Pump

2.txt www.retscreen.net/fichier.php/908/Chapter%20Pumps%20and%20... Pumping%20Systems.pdf

3.txt www.nesc.wvu.edu/pdf/dw/publications/ontap/2009_tb/... pumps_DWFSOM56.pdf

Adesivi

1.txt en.wikipedia.org/wiki/Adhesive

2.txt www.ascouncil.org/news/adhesives/

3.txt www.speedace.info/composites/adhesives.htm

Generatori

1.txt en.wikipedia.org/wiki/Electrical_generator

2.txt www.eolss.net/ebooks/Sample%20Chapters/C05/E639A05-03.pdf

3.txt www.bestelectricmachine.com/file_download/FAQ.pdf

Trasformatori

1.txt en.wikipedia.org/wiki/Transformer
2.txt www.lgedrein.org/archive_file/books/ces/... Fundamentals_Electric_Motors_Transformers.pdf
3.txt www.wbdg.org/ccb/ARMYCOE/COETM/tm_5_686.pdf

Inverter

1.txt ecee.colorado.edu/∼rwe/papers/Encyc.pdf
2.txt www.jaycar.com.au/images_uploaded/dcdcconv.pdf
3.txt www.jaycar.com.au/images_uploaded/inverter.pdf

Valvole

1.txt en.wikipedia.org/wiki/Valve
2.txt www.maintenanceworld.com/Articles/valvemag/whatis.html
3.txt www.documentation.emersonprocess.com/groups/public/documents/book/

Cavi LAN

1.txt en.wikipedia.org/wiki/Ethernet_physical_layer
2.txt www.zytrax.com/tech/layer_1/cables/tech_lan.htm
3.txt members.tripod.com/barhoush_2/cabling.htm

Serbatoi

1.txt en.wikipedia.org/wiki/Storage_tank
2.txt www.buckeyef.com/images/PDF/FOAMPDF/fixed.pdf
3.txt www.mrwa.com/OPStorage.pdf

B.0.7 Struttura e peso dei legami Concetto-Concetto e Concetto-Parola: Schema DB

Lubrificanti

Word 1	Word 2	RF
base	high	3.03869
base	fluid	4
lubric	base	3.38684
high	viscos	3
base	oxid	3.09608
lubric	high	3.02514
high	oxid	3.3673
lubric	addit	3.20467
lubric	oil	3.31675
fluid	oxid	3.04688
oil	viscos	3.97357
lubric	oxid	3.44146
temperatur	oxid	1.06085
thick	oil	1.06815
basestock	fluid	1.06637
perform	addit	1.00392
solid	high	1.03883
manufactur	lubric	1.10863
test	lubric	1.07765
test	oil	1
water	high	1.03693
api	oxid	1.01335
finish	base	1
includ	viscos	1.18318
bear	oxid	1.04458
inhibitor	fluid	1.01261
fluid	base	1.06477
corros	oxid	1.03589
wear	oxid	1.04092
higher	oxid	1.01148
higher	high	1.02958
tabl	oil	1.14405
hydraul	addit	1.03163
load	oil	1.22517
synthet	oxid	1.00297
hydrodynam	fluid	1.01946

Tabella B.1: Struttura DB per Lubrificanti.

Word 1	Word 2	RF
naphthen	lubric	1.26966
classifi	viscos	1.13541
heat	oxid	1.01804
characterist	viscos	1.12425
engin	high	1.00202
refer	viscos	1.22495
heat	high	1.00455
good	lubric	2
automot	oil	1.01858
grade	addit	1.27141
composit	high	1
previous	viscos	1.22101
naphthen	viscos	1.24775
base	fluid	1.03056
classif	addit	1.09295
higher	addit	1.07886
ep	lubric	1.41982
antioxid	lubric	1.16247
antiwear	lubric	1.71548
basestock	base	1.00315
societi	oil	1.18522
appli	oil	1.16711
greas	viscos	1.4214
includ	oil	1.12888
purpos	oil	2
group	base	1.0246
aromat	fluid	1.07037
agent	addit	1
meet	lubric	1.36726
make	addit	1.09243
distil	lubric	1
finish	fluid	1.07735
refer	lubric	1.98064

Tabella B.1: Struttura DB per Lubrificanti.

Pompe

Word 1	Word 2	RF
fluid	displac	3.00176
oper	type	3.15478
type	centrifug	3.02758
fluid	water	3.00917
oper	pressur	4
pressur	suction	3.03025
head	suction	3
pump	displac	3.54304
pump	head	3.25454
type	pressur	3.05349
head	pressur	3.05867
point	fluid	1.07934
electr	pump	2
typic	pump	1.18408
point	centrifug	1.10348
ensur	pressur	1.02928
shaft	suction	1.03673
pressur	oper	1.29435
air	fluid	1.07912
check	water	1.28579
cylind	fluid	1.05462
shape	type	1.01392
turbin	water	1.46866
radial	fluid	1.11659
ram	fluid	1.01274
peristalt	fluid	1.02634
mix	fluid	1.00909
discharg	type	1.00508
doubl	centrifug	1.03808
pipe	oper	1.01569
higher	pump	1.43218
convers	type	1.0426
mechan	displac	1.0222
improv	suction	1.02198
oper	pressur	1.29347
curv	head	1.05138
reduc	suction	1.05084
handl	fluid	1

Tabella B.2: Struttura DB per Pompe.

Word 1	Word 2	RF
call	water	1.57917
total	pump	1
trap	type	1.00081
vane	centrifug	1.03008
compon	head	1.00724
kinet	pump	1.06135
speed	suction	1.02592
industri	head	1.04009
servic	water	1.34319
larg	centrifug	1.0686
lower	centrifug	1.01014
due	type	1
resist	head	1.06346
option	suction	1
util	pressur	1.0331
end	head	1.01104
raw	water	1.33581
suppli	water	1.50118
curv	suction	1.02133
figur	head	1.08688
increas	pressur	1.10436
stationari	head	1
util	oper	1.00211
rotat	pressur	1.08998
practic	water	1.34992
function	centrifug	1.02159
rotat	oper	1
repair	type	1.00121
side	displac	1.01647
posit	displac	1.06581
liquid	pressur	1.10038
valv	centrifug	1.08025

Tabella B.2: Struttura DB per Pompe.

Adesivi

Word 1	Word 2	RF
surfac	applic	3
bond	failur	3.09858
bond	glue	3.07624
adhes	glue	3.37005
glue	surfac	3.0065
adhes	design	4
adhes	failur	3.19058
adhes	materi	3.10616
bond	materi	3.61114
adhes	bond	3.03068
adhes	surfac	3.36022
adhes	applic	3.04645
reactiv	surfac	1.00542
design	surfac	1.00097
pressur	surfac	1.02876
test	surfac	1.09599
cure	surfac	1.04397
test	glue	1.12563
appli	failur	1.06279
form	applic	1.01749
fractur	applic	1.05684
improv	materi	1.18693
appli	surfac	1.00983
glass	materi	1.05587
depend	adhes	1.44358
consist	glue	1.03749
consist	materi	1.25606
materi	bond	1.1965
forc	bond	1
water	design	1.00563
part	glue	1.01289
resist	applic	1.01733
structur	failur	1.07851
resin	materi	1.05171
common	failur	1.09526
applic	surfac	1.05752
label	surfac	1
bond	materi	1.26561
joint	bond	1.06202

Tabella B.3: Struttura DB per Adesivi.

Word 1	**Word 2**	**RF**
case	glue	1.03616
addit	glue	1.02668
hot	failur	1.11485
adhes	design	1.07759
sealant	applic	1.11496
reaction	failur	1.12188
surfac	applic	1.04065
onlin	materi	1.07681
failur	bond	1.00104
design	adhes	2
paper	materi	1.05588
part	adhes	1.08202
pressur	bond	1.08913
servic	materi	1.15021
remain	applic	1.05517
reactiv	glue	1.10486
cohes	materi	1.0793
raw	glue	1
sensit	materi	1.2542
made	glue	1.01069
resist	glue	1.00433
dri	adhes	1.22498
solvent	adhes	1.0622
pressur	design	1.10511
locat	adhes	1
strength	glue	1.05069
sealant	surfac	1.10187
cure	applic	1.00387

Tabella B.3: Struttura DB per Adesivi.

Generatori

Word 1	Word 2	RF
synchron	field	3.00729
field	rotor	3.21
synchron	wind	3.20834
synchron	machin	3.09186
synchron	rotor	3.00546
machin	wind	3
generat	magnet	4
machin	power	3.08827
synchron	power	3.11697
machin	field	3.00867
electr	motor	3.03636
field	motor	3.02061
generat	rotor	3.2407
oper	motor	1.31968
engin	field	1
equip	magnet	1.28224
applic	power	1.71568
lead	machin	1
practic	motor	1.14482
low	generat	1
process	electr	1.20264
air	electr	2
vehicl	magnet	1.53112
connect	power	1.19564
output	field	1.05163
develop	power	1.24459
effect	motor	1.37528
limit	motor	1.16597
limit	power	1.31682
exceed	motor	1.12935
compon	wind	1.21495
factor	synchron	1.54427
proport	electr	1.13851
effici	electr	1.34625
acceler	machin	1.1923
total	wind	1.14711
vehicl	generat	1.92495
result	rotor	1.35277
design	magnet	1.19395

Tabella B.4: Struttura DB per Generatori.

Word 1	Word 2	RF
magnet	generat	2
convert	wind	1.09797
lag	synchron	1.75089
system	rotor	1.11192
frequenc	rotor	1.08556
stator	wind	1.05039
perform	power	1.22132
reduc	wind	1.09145
port	wind	1.05807
page	wind	1.14801
equip	generat	1.10257
oper	rotor	1.38606
show	machin	1.0967
grid	motor	1.6585
convers	electr	1.37954
inher	wind	1.09474
condit	rotor	1.07307
direct	generat	1.05865
circuit	power	1.6992
relat	machin	1.00224
generat	magnet	1.27574
energi	rotor	1.16342
batteri	magnet	1.42032
uniti	machin	1.0482
paragraph	electr	1.24199
vehicl	power	1.26028
induct	rotor	1.14783
phase	machin	1.82996
due	power	1.38118

Tabella B.4: Struttura DB per Generatori.

Trasformatori

Word 1	Word 2	RF
secondari	primari	3.02832
oper	connect	3.03109
transform	connect	3.26751
transform	secondari	3.88176
current	connect	3.00847
wind	power	3.09935
transform	oper	3.16497
core	wind	3.80997
current	core	3.31379
core	primari	3.21485
wind	primari	4
transform	voltag	3
connect	oper	1.00377
ac	voltag	1.35751
phase	current	1.03139
type	connect	1.05065
chang	connect	1.00269
sine	core	1.06646
insul	connect	1.01473
turn	primari	1.31762
section	power	1
larger	transform	1.12826
side	secondari	1.08205
construct	oper	1.00683
constant	connect	1.03044
common	core	1.03974
area	voltag	1.30662
coil	oper	1.01833
practic	wind	1.16108
turn	voltag	1.29452
wind	core	1.13796
end	core	1.0494
tap	current	1
core	wind	1.22134
angular	secondari	1.00159
cost	voltag	1.63746
wind	primari	1.32344
configur	voltag	1.2408
insid	connect	1.02436

Tabella B.5: Struttura DB per Trasformatori.

Word 1	**Word 2**	**RF**
circuit	core	1.31089
contact	secondari	1.19823
heat	oper	1.03743
primari	wind	1.65094
applic	primari	1.09557
open	transform	1
occur	secondari	1
form	secondari	1.09881
induc	power	1.02594
air	voltag	1.70279
fill	current	1.00837
secondari	transform	1.17103
lead	voltag	1.45884
complet	secondari	1.01021
vari	primari	1.07967
reson	current	1.02079
lamin	wind	1.32518
ratio	wind	1.90142
case	connect	1.01184
ratio	core	1.03776
eddi	power	1.0536
effect	wind	1.18136
direct	transform	1.04264
pressur	core	1.21346
produc	voltag	1.22548
work	core	1.07007
conductor	connect	1.01463
ground	secondari	1.07842
liquid	transform	1.08708
ondari	secondari	1.00113
turn	wind	1.85889
lower	oper	1
fill	power	1.03593
compon	transform	1.1225
rate	connect	1.04252
import	secondari	1.18207
ratio	primari	1.25938
design	core	1.09202
lamin	primari	1.08144
oper	connect	1.00521
refer	transform	1.02086

Tabella B.5: Struttura DB per Trasformatori.

Word 1	Word 2	RF
singl	transform	2
field	current	1.12589

Tabella B.5: Struttura DB per Trasformatori.

Inverter

Word 1	**Word 2**	**RF**
voltag	high	3.48083
voltag	output	3.62624
dc	switch	3.28258
switch	frequenc	3.0039
voltag	oper	3.21929
voltag	flow	3.00257
output	turn	3.00035
output	flow	3.17547
flow	oper	3.01574
turn	flow	3.0661
switch	ac	3.03165
high	oper	3.09134
high	flow	3.01817
voltag	switch	3.24721
dc	output	3
voltag	dc	3.02837
flow	frequenc	3.07325
dc	flow	3.12329
high	output	3.03309
dc	oper	3.13354
output	ac	4
flow	ac	3.14869
output	oper	3.56177
dc	ac	3.06759
turn	oper	3.74591
ac	oper	3.05636
iinn	flow	1.00543
voouutt	turn	1.29376
fair	oper	1.08561
main	oper	1.13462
current	flow	1.02222
result	ac	1.25899
singl	flow	1
effici	dc	1.00686
oper	turn	1.10075
isol	voltag	2
small	frequenc	1.16164
width	switch	1.18495
damag	frequenc	1.07676

Tabella B.6: Struttura DB per Inverter.

Word 1	Word 2	RF
effici	output	1.15119
induct	dc	1.00999
develop	high	1
provid	oper	1.14185
low	switch	1.22701
small	switch	1.39206
deliv	high	1.00199
batteri	ac	1.209
maintain	voltag	1
increas	frequenc	1.11492
input	ac	1.17439
transfer	voltag	1.34785
protect	frequenc	1.10159
process	ac	1.15814
limit	switch	1.18686
dcdcconv	turn	1.19415
suppli	output	1.22144
network	switch	1.24267
refer	flow	1.01191
bias	dc	1.05481
flyback	voltag	1.41876
outlet	frequenc	1.12505
q1	ac	1.23538
circuitri	high	1.04773
sens	ac	1.18328
q1	flow	1.00292
produc	output	1.20147
d1	voltag	1.77295
vconvert	turn	1.17024
control	frequenc	1.1289
diod	dc	2
smaller	output	1.17573
waveform	output	1.18301
cycl	voltag	1.74549
turn	oper	1.11479
sourc	voltag	1.77111
time	dc	1
lead	frequenc	1.13437
aa	high	1.06592
viinn	turn	1.11698
harmon	switch	1.27426

Tabella B.6: Struttura DB per Inverter.

Word 1	**Word 2**	**RF**
circuit	output	1.21774
vari	voltag	1.31478
accident	frequenc	1.08416

Tabella B.6: Struttura DB per Inverter.

Valvole

Word 1	Word 2	RF
flow	seat	3.09589
valv	system	3.00407
control	system	3.13045
flow	control	4
control	actuat	3.02903
flow	plug	3.53562
oper	seat	3.34177
open	system	3.104
control	open	3.14397
oper	ball	3.35996
open	plug	3.01798
valv	ball	3.059
valv	seat	3.20104
seat	plug	3.04756
oper	system	3
control	plug	3.17353
oper	plug	3.0338
differ	flow	1.16756
brass	valv	1.26155
mechan	flow	1.03415
liquid	valv	1.25968
common	ball	1.06596
diaphragm	control	1.22682
linear	actuat	1.0489
check	valv	2
back	ball	1.10619
piston	system	1.00073
case	ball	1.09155
vacuum	valv	1.08512
automat	ball	1.04367
societi	actuat	1.00577
order	flow	1.02687
seat	oper	1.02293
condit	flow	1.1234
plug	flow	1
seal	flow	1.35867
spring	seat	1.27446
slurri	valv	1.01826
fail	open	1.00671

Tabella B.7: Struttura DB per Valvole.

Word 1	Word 2	RF
design	plug	1.0011
manual	valv	1.06133
open	system	1
ball	oper	1.09029
includ	open	1
drop	ball	1.14324
diaphragm	actuat	1.45714
process	open	1.06486
intern	control	1.05769
oper	ball	1.09794
design	actuat	1
common	oper	1.01155
mean	actuat	1.3035
inlet	seat	1.29384
automat	plug	1.00814
pipe	valv	1
general	actuat	1.23858
back	oper	1.06859
port	seat	1.37439
abbrevi	flow	1.01582
piston	open	1.02109
low	ball	1.08374
close	control	1.03821
outlet	seat	1.16699
spring	plug	1
rotari	control	2
devic	actuat	1.17725

Tabella B.7: Struttura DB per Valvole.

Cavi LAN

Word 1	Word 2	RF
pair	connect	3.38385
connect	end	3.04275
connector	twist	3.00493
cabl	wire	3.09205
cabl	pair	3.53394
pair	wire	3
ethernet	signal	3.01795
connector	signal	3.06022
standard	end	3.14274
ethernet	connector	3.56924
pair	connector	3.1574
cabl	connect	3.39231
pair	signal	3.12472
ethernet	standard	4
100base-tx	ethernet	1.27899
coaxial	pair	1
10	signal	1.13829
scheme	end	1.07021
electr	connector	1.20244
multi	cabl	1.19952
unshield	twist	1.17983
inform	connector	1.07595
solid	cabl	1.38627
hub	pair	1.10546
fast	ethernet	1.19841
40	ethernet	1.08796
ieee	ethernet	1.68691
fiber	signal	1.31055
hub	connect	1.02
conductor	wire	1.0398
maximum	connector	1.03066
plastic	standard	1.04582
segment	twist	1.19553
light	twist	1.57977
run	connect	1.01098
gbit	ethernet	1.18166
speed	end	1.05128
lan	cabl	1
connect	pair	1.11945

Tabella B.8: Struttura DB per Cavi LAN.

Word 1	**Word 2**	**RF**
radio	twist	1.17573
ground	cabl	1.2513
speed	ethernet	1.49653
drain	wire	1.00768
802.3	cabl	2
length	signal	1.043
descript	standard	1.02441
mbps	twist	1.27672
pc	pair	1.12588
center	twist	2
thick	standard	1.02191
blue	connect	1.00684
communic	twist	1.17523
note	end	1.00485
copper	signal	1.07533
show	wire	1.01678
requir	standard	1.12768
cross	pair	1.3588
note	wire	1.32956
pc	end	1.07476
male	wire	1.02415
stp	connector	1
manchest	ethernet	1.20883
colour	wire	1.0363
short	ethernet	1.25306
length	pair	1.04884
instal	connector	1.23643
specif	connector	1.0425
speed	standard	1.30032
wireless	twist	1.2109
foil	end	1
end	standard	1.00394
1000base-t	standard	1
vendor	standard	1.00141
run	end	1.02876
ethernet	standard	1.13156
categori	connector	1.23338
spec	pair	1.03251
standard	ethernet	1.41455
pair	connect	1
fig	standard	1.01585

Tabella B.8: Struttura DB per Cavi LAN.

Word 1	Word 2	RF
bnc	twist	1.19095
articl	ethernet	1.1769
carri	standard	1.01872
case	wire	1.01681
spec	end	1.00034

Tabella B.8: Struttura DB per Cavi LAN.

Serbatoi

Word 1	Word 2	RF
protect	roof	3.11342
tank	liquid	3.13853
pressur	protect	3.18248
storag	water	3
storag	pressur	3.13408
tank	water	3.37246
tank	requir	3.00775
tank	construct	3.88201
pressur	type	3.01753
liquid	type	3.43828
liquid	pressur	3.07028
tank	pressur	3.2875
tank	storag	3.33194
type	protect	3.04272
tank	roof	4
tank	protect	3.15123
protect	requir	3.01563
liquid	protect	3.0777
low	construct	1.30163
suppli	protect	1.04498
build	construct	1.16158
frequent	water	1.35073
contain	requir	1.03436
pipe	protect	1.24216
inch	pressur	1.01996
general	construct	1.21333
flow	storag	1.16865
base	storag	1
fuel	roof	1.70785
fix	type	1.37652
larg	construct	1.47372
fix	liquid	1
time	requir	1
high	protect	1.03494
concret	water	1.38367
reservoir	roof	1.93949
construct	tank	1
capabl	storag	1.42194
steel	pressur	1.06845

Tabella B.9: Struttura DB per Serbatoi.

Word 1	Word 2	RF
regular	water	1.3908
contain	pressur	1.03399
fill	construct	1.23174
standard	storag	2
full	protect	1
flash	type	1.56955
milk	roof	1.48009
riser	storag	1.37695
maker	liquid	1.11673
foundat	water	1.81879
high	pressur	1.30984
miscibl	storag	1.69999
coat	construct	1.14296
high	storag	1.10384
surfac	requir	1.14406
larg	tank	2
type	liquid	1.21123
bottom	requir	1.18658
hose	liquid	1.06024
line	pressur	1.09122
refineri	roof	1.5813
liquid	type	1.19912
gas	roof	1.90823
intern	roof	1.5457
excess	storag	1.16241
mainten	water	1.36383
call	roof	1.5196

Tabella B.9: Struttura DB per Serbatoi.

B.0.8 Struttura e peso dei legami Concetto-Concetto e Concetto-Parola: Rappresentazione grafica con Informal Lightweight Ontology

Lubrificanti

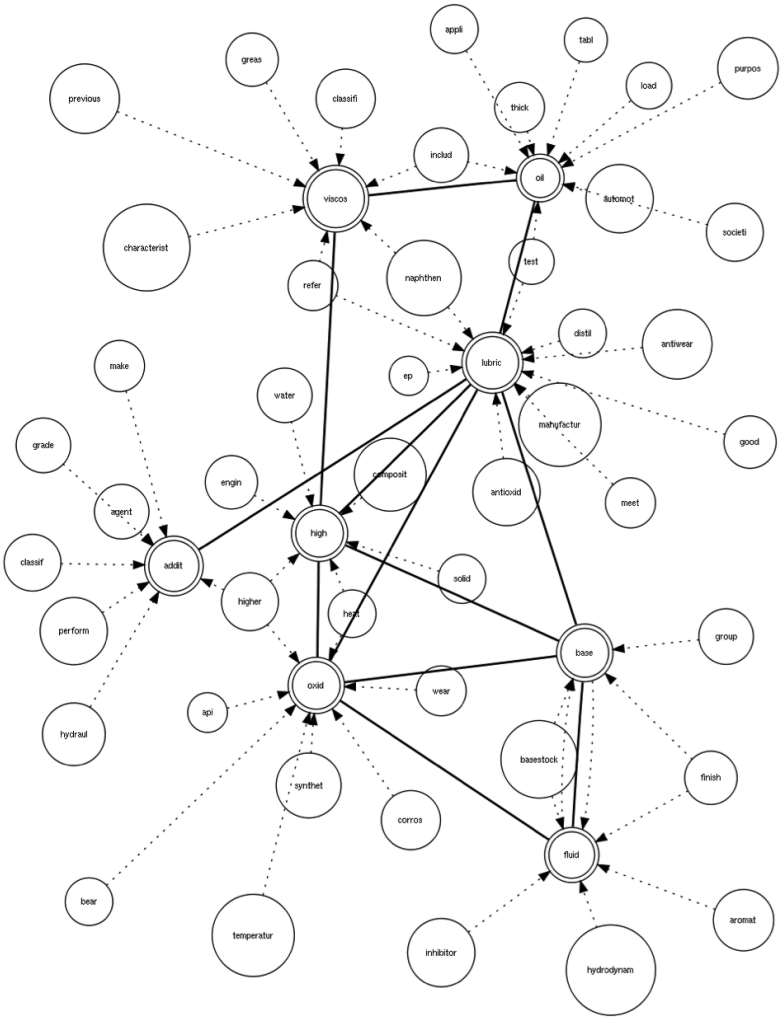


Figura B.1 ILO Lubrificanti

Pompe

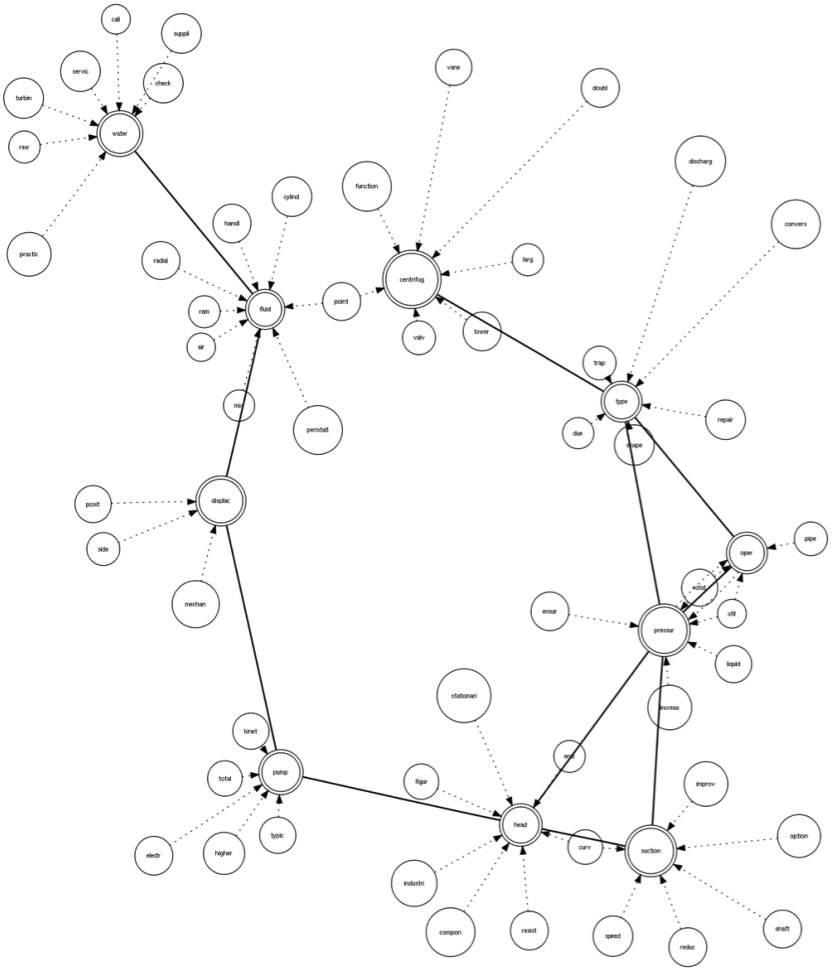


Figura B.2 ILO Pompe

Adesivi

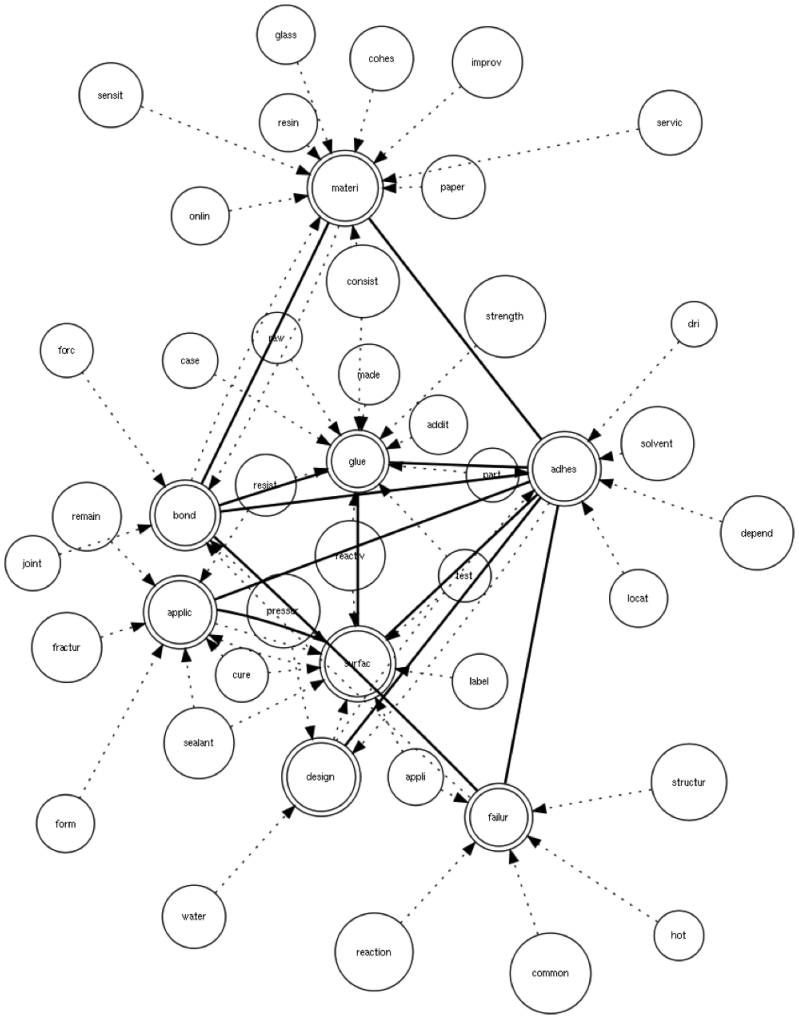


Figura B.3 ILO Adesivi

Generatori

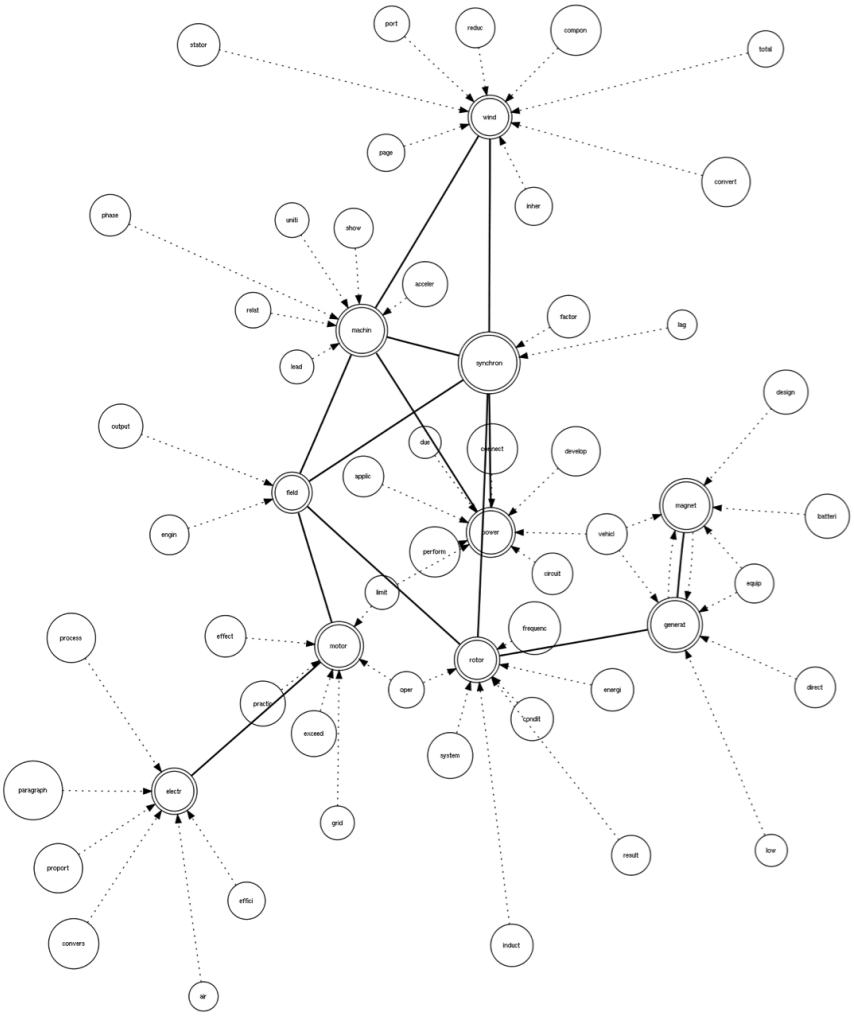


Figura B.4 ILO Generatori

Trasformatori

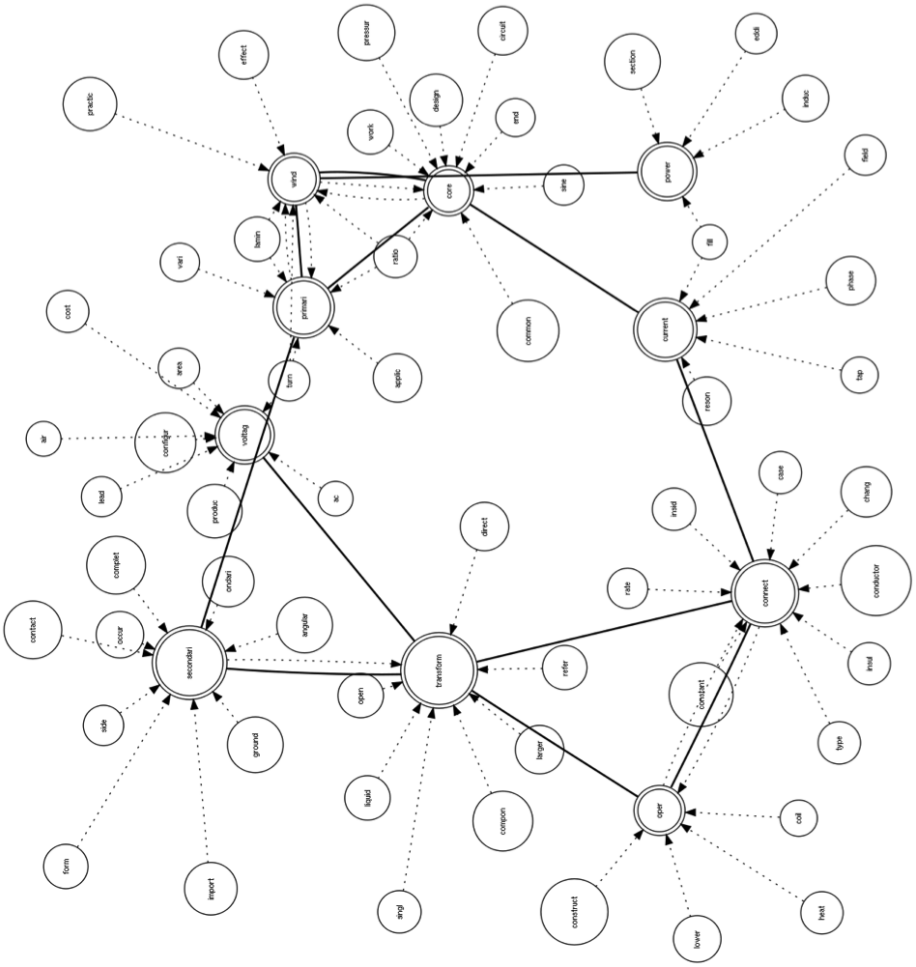


Figura B.5 ILO Trasformatori

Inverter

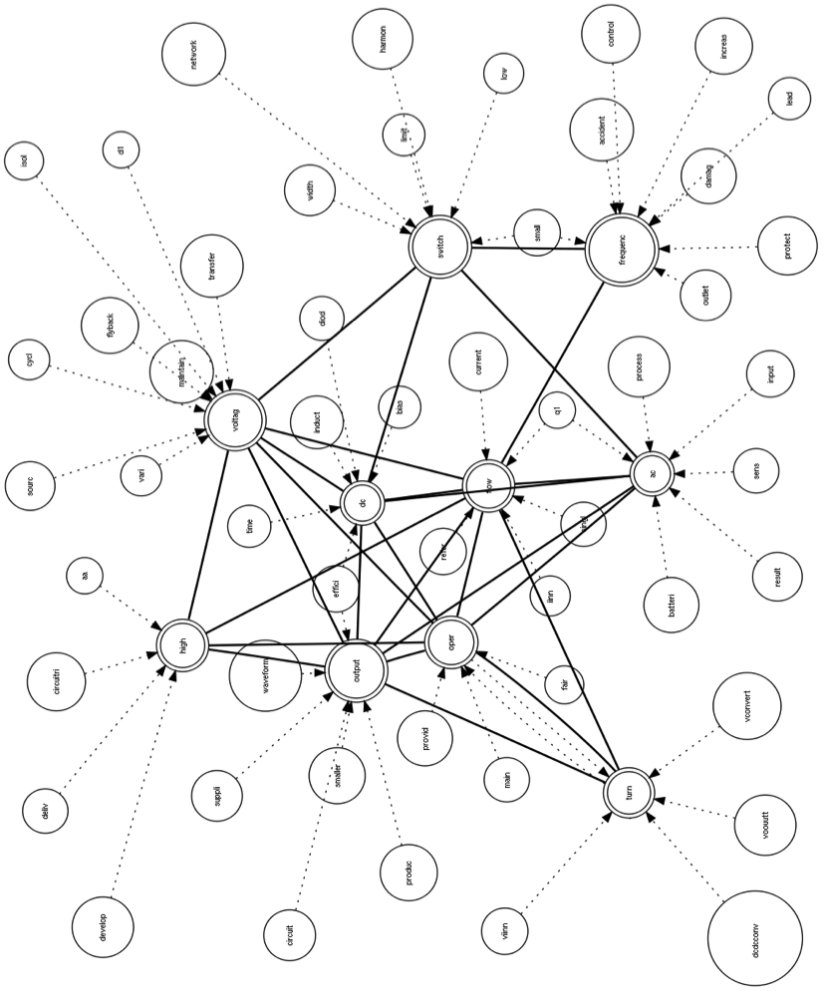


Figura B.6 ILO Inverter

Valvole

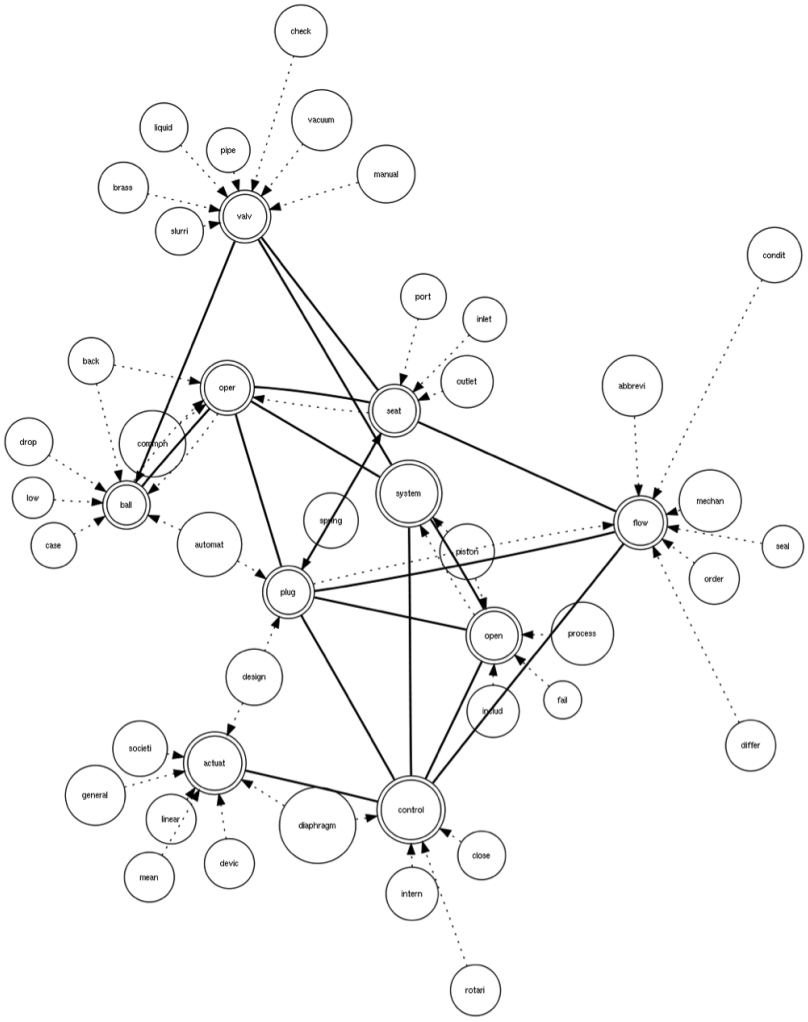


Figura B.7 ILO Valvole

Cavi LAN

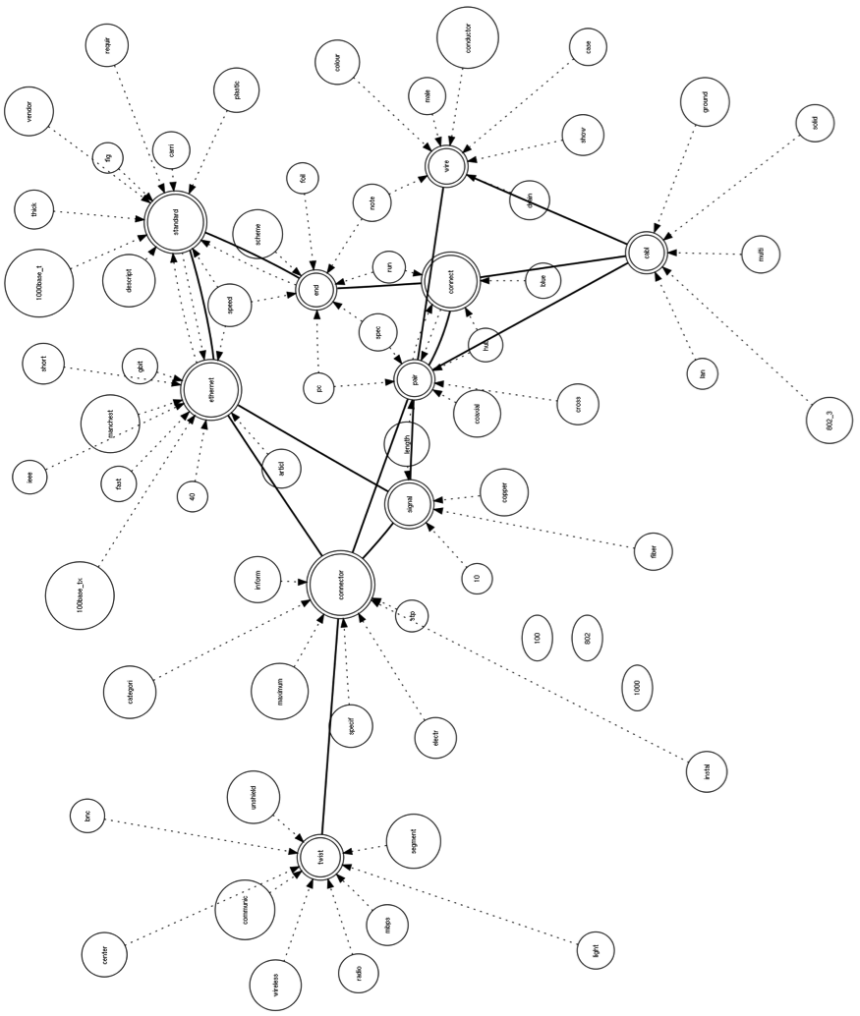


Figura B.8 ILO Cavi LAN

Serbatoi

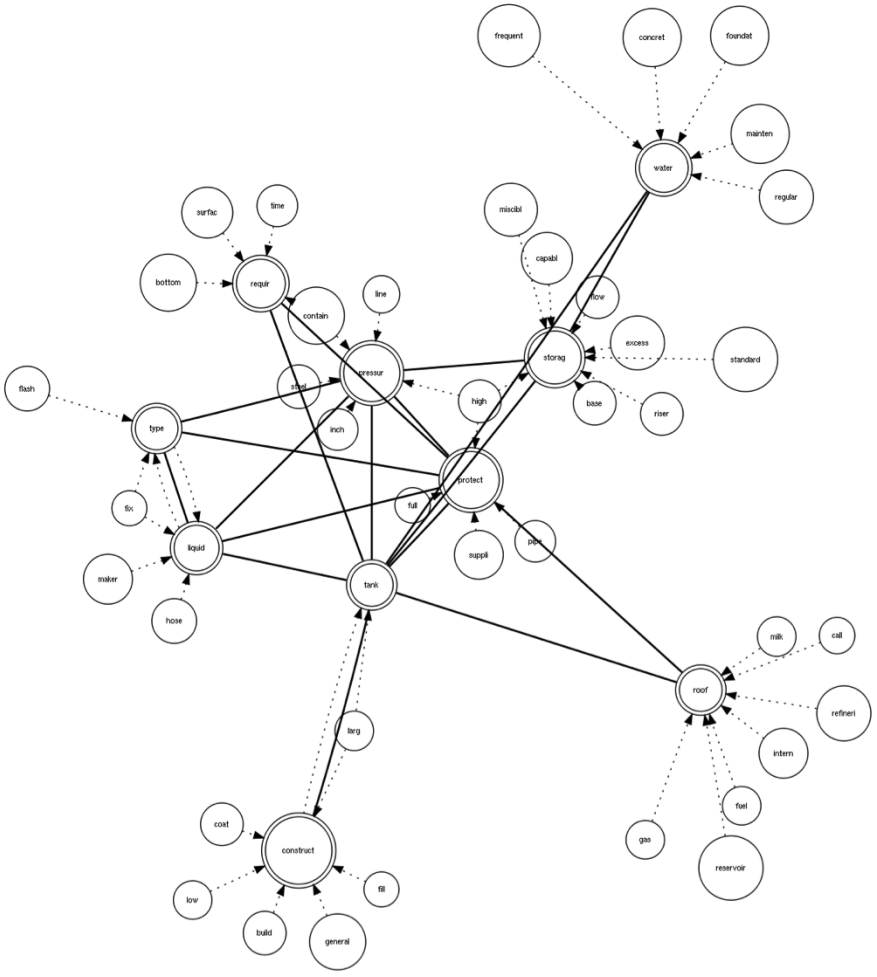


Figura B.9 ILO Serbatoi

Bibliografia

[1] Concept theory. *Journal of the American Society for Information Science and Technology*, 60(1519 - 1536), 2009.

[2] D.H. Ballard. *An Introduction to Natural Computation*. The MIT Press, Cambridge, MA, 1997.

[3] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice Hall, New York, N.Y., 1982.

[4] Judit Bar-Ilan. Methods for measuring search engine performance over time. *Journal of the American Society for Information Science and Technology*, 53(308–319), 2004.

[5] P. Berkhin. A survey of clustering data mining techniques. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data*, pages 25–71. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-28349-2.

[6] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing & Management*, 43(4):866 – 886, 2007. ISSN 0306-4573. doi: DOI:10.1016/j.ipm.2006.09.003.

[7] D. M Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(993–1022), 2003.

[8] J.M. Bradshaw, G. Boy, E. Durfee, M. Gruninger, H. Hexmoor, N. Suri, M. Tambe, M. Uschold, and J. Vitek. Semantic integration. In *Software Agents for the Warfighter*. ITAC Consortium Report. Cambridge, MA: AAAI Press/The MIT Press, 2004.

[9] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.

[10] Ben Carterette, James Allan, and Ramesh Sitaraman. Minimal test collections for retrieval evaluation. In *29th International ACM SIGIR Conference on Research and development in information retrieval*, 2008.

[11] Philipp Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, 2006.

[12] F. Colace, M. De Santo, and P. Napoletano. A note on methodology for designing ontology management systems. In *AAAI Spring Symposium*, 2008.

[13] A. M. Collins and M. R. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, (8): 240–247, 1969.

[14] Paulo C. G. Costa and Kathryn B. Laskey. Pr-owl: A framework for probabilistic ontologies. In *Proceeding of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, pages 237–249, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press. ISBN 1-58603-685-8. URL `http://portal.acm.org/citation.cfm?id=1566079.1566107`.

[15] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.

[16] Ivan De Falco, Antonio Della Cioppa, Domenico Maisto, Umberto Scafuri, and Ernesto Tarantino. Extremal optimization dynamics in neutral landscapes: The royal road case. In *Artificial Evolution*, pages 1–12, 2009.

[17] Jaques Derrida. De la grammatologie. *Paris:Minuit*, 1997.

[18] Zhongli Ding, Yun Peng, and Rong Pan. BayesOWL: Uncertainty Modeling in Semantic Web Ontologies. *Soft Computing in Ontologies and Semantic Web*, pages 3–29, 2006. URL `http://dx.doi.org/10.1007/3-540-33473-4_1`.

[19] D. Duvivier, Philippe Preux, and El-Ghazali Talbi. Climbing up np-hard hills. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 574–583, London, UK, 1996. Springer-Verlag. ISBN 3-540-61723-X.

[20] Umberto Eco. A theory of semiotics. *Bloomington:Undiana University Press.*, 1979.

[21] Efthimis N. Efthimiadis. Query expansion. In Martha E. Williams, editor, *Annual Review of Information Systems and Technology*, pages 121–187. 1996.

[22] K. A. Ericsson and W. Kintsch. Long-term working memory. *Psychological Review.*, 102:211–245, 1995.

[23] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007. ISBN 3-540-49611-4.

[24] Stephanie Forrest and Melanie Mitchell. Relative building-block fitness and the building-block hypothesis. In Darrell L. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 109–126. Morgan Kaufmann, San Mateo, CA, 1993.

[25] Yoshio Fukushige. Representing probabilistic relations in rdf. In Paulo Cesar G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey, and Michael Pool, editors, *ISWC-URSW*, pages 106–107, 2005. URL `http://dblp.uni-trier.de/db/conf/semweb/ursw2005.html#Fukushige05`.

[26] Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2004.

[27] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. New York: Chapman & Hall, 1995.

[28] Rosalba Giugno and Thomas Lukasiewicz. P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the

semantic web. In *Proceedings of the European Conference on Logics in Artificial Intelligence*, JELIA '02, pages 86–97, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44190-5. URL `http://portal.acm.org/citation.cfm?id=646333.756800`.

[29] F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Towards a theory of formal classification. AAAI Press, 2005.

[30] T. L. Griffiths, M. Steyvers, and S. Dennis. Probabilistic inference in human semantic memory. *Trends in Cognitive Science*, (10): 327–334, 2006.

[31] W. I. Grosky, D. V. Sreenath, and F. Fotouhi. Emergent semantics and the multimedia semantic web. In *SIGMOD Record*, volume 31, pages 54–58, 2002.

[32] T. R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis*, 5, 1993.

[33] N. Guarino. *Formal Ontology in Information Systems*. IOS Press, Cambridge, MA, USA, 1998.

[34] J. Heinsohn. A hybrid approach for modeling uncertainty in terminological logics. In R. Kruse and P. Siegel, editors, *Symbolic and Quantitative Approaches to Uncertainty: Proc. of the European Conference ECSQAU*, pages 198–205. Springer, Berlin, Heidelberg, 1991.

[35] M. Hepp and J. de Bruijn. Gentax: A generic methodology for deriving owl and rdf-s ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In *4th European Semantic Web Conference*. LNCS Springer, 2007.

[36] Marilyn Rosenthal Heting Chu. Search engines for the world wide web: a comparative study and evaluation methodology. In *In Proceedings of the 59th annual meeting of the American Society for Information Science*, pages 127–135, 1996.

[37] Markus Holi and Eero Hyvönen. *Modeling Uncertainty in Semantic Web Taxonomies*. Eurooppa, 2006. ISBN 3540334726. URL `http://www.seco.tkk.fi/publications/2006/holi-hyvonen-modeling-uncertainty-in-2006.pdf`.

[38] Amanda Spink Howard Greisdorf. Median measure: an approach to ir systems evaluation. *Information Processing and Management*, 37(6)(843–857), 2001.

[39] Scott B. Huffman and Michael Hochster. How well does result relevance predict session satisfaction? In *SIGIR*, pages 567–574, 2007.

[40] T. I. Ibrahim and C.-Z. Xu. Neural net based pre-fetching to tolerate www latency. In *20th International Conference on Distributed Computing Systems*, 2000.

[41] Manfred Jaeger. Probabilistic reasoning in terminological logics. In *KR*, pages 305–316, 1994.

[42] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207–227, 2000. ISSN 0306-4573.

[43] Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management*, 44(3):1251 – 1266, 2008. ISSN 0306-4573.

[44] E. T. Jaynes. *Probability Theory - The Logic of Science*. Cambridge Press, 2003.

[45] W. Kintsch. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95:163–182, 1988.

[46] Daphne Koller, Alon Levy, and Avi Pfeffer. P-classic: A tractable probabilistic description logic. In *In Proceedings of AAAI-97*, pages 390–397, 1997.

[47] Kathryn B. Laskey and Paulo C. G. Da Costa. Of starships and klingons: Bayesian logic for 23rd century. In *Proc. UAI-05*, pages 346–353, 2005.

[48] Chia-Jung Lee, Yi-Chun Lin, Ruey-Cheng Chen, and Pu-Jen Cheng. Selecting effective terms for query formulation. In Gary

Lee, Dawei Song, Chin-Yew Lin, Akiko Aizawa, Kazuko Kuriyama, Masaharu Yoshioka, and Tetsuya Sakai, editors, *Information Retrieval Technology*, volume 5839 of *Lecture Notes in Computer Science*, pages 168–180. Springer Berlin / Heidelberg, 2009.

[49] A. Maedche and S. Staab. Ontology learning. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 173–190. Springer, 2004. ISBN 3-540-40834-7.

[50] Alexander Maedche. *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, 2002. ISBN 0262133601.

[51] B. Magnini, L. Serafini, and M. Speranza. Making explicit the hidden semantics of hierarchical classifications. In *AI*IA 2003: Advances in Artificial Intelligence*. LNCS Springer, 2003.

[52] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008. ISBN 978-0-521-86571-5. URL `http://nlp.stanford.edu/IR-book/information-retrieval-book.html`.

[53] D. Marr. *Vision*. Freeman, S. Francisco,CA, 1982.

[54] Praveen Pathak Michael Gordon. Finding information on the world wide web: the retrieval effectiveness of search engines. *Information Processing and Management*, 35(141–180), 1999.

[55] Melanie Mitchell and John H. Holland. When will a genetic algorithm outperform hill climbing? In *Proceedings of the 5th International Conference on Genetic Algorithms*, page 647, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1-55860-299-2.

[56] Slonim Noam and Tishby Naftali. The power of word clusters for text classification. In *In 23rd European Colloquium on Information Retrieval Research*, 2001.

[57] Rong Pan, Zhongli Ding, Yang Yu, and Yun Peng. A bayesian network approach to ontology mapping. In *In: Proceedings ISWC 2005*, pages 563–577. Springer, 2005.

[58] Scott Piao, Brian Rea, John McNaught, and Sophia Ananiadou. Improving full text search with text mining tools. In Helmut Horacek, Elisabeth Métais, Rafael Muñoz, and Magdalena Wolska, editors, *Natural Language Processing and Information Systems*, volume 5723 of *Lecture Notes in Computer Science*, pages 301–302. Springer Berlin / Heidelberg, 2010.

[59] M. C. Potter. Very short term conceptual memory. *Memory & Cognition*, (21):156–161, 1993.

[60] L. Predoiu and H. Stuckenschmidt. Probabilistic models for the semantic web – a survey. *The Semantic Web for Knowledge and Data Management: Technologies and Practices*, 2008.

[61] Livia Predoiu. Information integration with bayesian description logic programs. In *Proceedings of the Workshop on Information Integration on the Web (IIWeb 2006), in conjunction with WWW2006*, Edinburgh, Scotland, 5 2006.

[62] Z.W. Pylyshyn. Situating vision in the world. *Trends in Cognitive Sciences*, 4(5):197–207, 2000.

[63] Donald G. Saari. *Chaotic Elections! A Mathematician Looks at Voting*. American Mathematical Society, Providence, 2001.

[64] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data engineering*, 13(3):337–51, 2001.

[65] Simone Santini. Summa contra ontologiam. *Current Trends in Database Technology*, 2006.

[66] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002. ISSN 0360-0300.

[67] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/505282.505283. URL `http://doi.acm.org/10.1145/505282.505283`.

[68] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *ACM SIGIR Forum*, 40(677–691), 1999.

[69] Umberto Straccia and Raphaël Troncy. Towards distributed information retrieval in the semantic web: Query reformulation using the omap framework. In York Sure and John Domingue, editors, *ESWC*, volume 4011 of *Lecture Notes in Computer Science*, pages 378–392. Springer, 2006. ISBN 3-540-34544-2.

[70] J. B. Tenenbaum T. L. Griffiths, M. Steyvers. Topics in semantic representation. *Psychological Review*, 114(2):211–244, 2007.

[71] Octavian Udrea, V. S. Subrahmanian, and Zoran Majkic. Probabilistic rdf. In *IRI*, pages 172–177. IEEE Systems, Man, and Cybernetics Society, 2006.

[72] M. Uschold and M. Gruninger. Ontologies and semantics for seamless connectivity. In *SIGMOD Rec.*, 2004.

[73] Liwen Vaughan. New measurements for search engine evaluation. *Information Processing and Management*, 40(677–691), 2004.

[74] Ellen M. Voorhees. Overview of trec 2003. In *In Proceedings of the 12th Text Retrieval Conference*, pages 1–13, 2003.

[75] Yi Yang and Jacques Calmet. Ontobayes: An ontology-driven uncertainty model. *Computational Intelligence for Modelling, Control and Automation, International Conference on*, 1:457–463, 2005. doi: http://doi.ieeecomputersociety.org/10.1109/CIMCA.2005.24.

[76] Philipp M. Yelland. An alternative combination of bayesian networks and description logics. In *KR*, pages 225–234, 2000.