POLITECNICO DI MILANO
Corso di Laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Supporto alla navigazione nei sistemi di classificazione collaborativa: applicazione di tecniche semantiche alle folksonomie

Relatore: Prof. Marco Colombetti
Correlatore: Ing. Davide Eynard

Tesi di Laurea di:
David Laniado, matricola 666797

Anno Accademico 2005-2006

A mia nonna
Rachele

Sommario

I sistemi di classificazione collaborativa di risorse, o *folksonomie*, rappresentano un paradigma in crescente affermazione nel Web, grazie alla loro capacità di raccogliere l'intelligenza collettiva degli utenti e di dare buoni risultati anche in un dominio ampio, dinamico ed eterogeneo. La mancanza di gerarchia e di una semantica esplicita, a cui è soggetto questo modello di classificazione, pone però delle limitazioni.

L'obiettivo di questa tesi è quello di studiare le possibilità di migliorare l'interfaccia di navigazione di una folksonomia, integrando al suo interno informazioni semantiche definite a priori in un'ontologia.

Il sistema proposto si basa su WordNet per effettuare una mappatura semantica delle tag di del.icio.us, una delle folksonomie attualmente più affermate, e arricchire le possibilità di esplorazione di insiemi di parole chiave correlate.

L'architettura del sistema è basata su un paradigma client-server, dove il server si occupa di recuperare le informazioni, disambiguare le parole chiave in base al loro contesto e costruire un albero semantico per organizzare le tag secondo la gerarchia dei concetti di WordNet. L'albero delle tag correlate realizzato costituisce una struttura ibrida, basata sia sulle informazioni semantiche, definite a priori nell'ontologia, sia su dati empirici di correlazione fra le parole chiave, estratti dalla folksonomia.

Per rendere possibile l'utilizzo di questa struttura ibrida come supporto alla navigazione è stata adottata una soluzione basata sul principio della navigazione attiva: uno script può essere installato ed eseguito localmente nel browser dell'utente, per modificare dinamicamente il contenuto delle pagine visitate integrando le informazioni ricevute dal nuovo server nell'interfaccia di navigazione di del.icio.us.

L'analisi dell'interfaccia di navigazione integrata e lo studio del funzionamento del sistema attraverso alcuni esempi di uso mostrano che esso offre effettivamente delle possibilità di navigazione che non sono normalmente possibili in una folksonomia e che risolve almeno in parte alcuni dei principali

problemi a cui è soggetto questo tipo di sistemi.

I risultati ottenuti sono incoraggianti e mostrano che un'integrazione dei due approcci, quello gerarchico delle tassonomie e quello "democratico" delle folksonomie, può migliorare le possibilità di navigazione degli utenti.

Indice

Capitolo 1

Introduzione

Con l'abbattimento delle barriere alla pubblicazione di contenuti online, la diffusione dei *blog* come fenomeno di massa e la crescita impetuosa di internet, si pone il problema dell'organizzazione di questa mole di informazioni e in particolare di uno schema di classificazione che permetta di orientarsi nella ricerca e nell'esplorazione.

L'approccio più tradizionale al problema della classificazione, basato su tassonomie di concetti definite a priori, entra in crisi di fronte a un dominio potenzialmente illimitato, estremamente eterogeneo e in costante e rapida trasformazione. Stabilire a priori delle categorie, che possano comprendere tutte le risorse in una struttura rigida e gerarchica, e classificare le risorse man mano che vengono create diventa un'impresa sempre più ardua in questo contesto.

Per questo motivo si sta affermando nel Web un nuovo paradigma, quello delle *folksonomie*: sistemi per la classificazione collaborativa basati su *tag*, parole chiave che ciascun utente può associare liberamente alle risorse. Il grande vantaggio offerto da questo tipo di soluzione è che il lavoro di classificazione viene svolto dagli utenti stessi; non c'è bisogno di creare a priori una struttura rigida di categorie e vengono superati molti problemi che questo passaggio pone: la difficoltà di introdurre nuovi concetti, la distanza dal punto di vista degli utenti, la necessità di un'unica visione autoritativa valida per tutti.

In un sistema a tag le categorie sono scelte e create dagli utenti stessi e dunque rispecchiano meglio il loro vocabolario; i concetti più largamente condivisi sono portati ad emergere, mentre allo stesso tempo ogni idea anche originale e minoritaria può essere inclusa: è un sistema democratico.

Questo approccio, tuttavia, presenta diversi limiti: l'ambiguità delle pa-

role e la scarsa accuratezza di molti utenti portano a un basso livello di precisione, mentre l'eterogeneità del linguaggio e degli schemi mentali delle persone, in assenza di un vocabolario controllato, compromettono la possibilità di trovare le risorse cercate. Uno dei problemi maggiori è la mancanza di gerarchia, che rende lo spazio delle tag piatto limitando le possibilità di ricerca e di esplorazione organica in una folksonomia. L'assenza di relazioni semantiche fra le categorie, che possano guidare l'utente, è stata affrontata dalle principali applicazioni introducendo sistemi di suggerimenti di parole chiave correlate, scelte automaticamente in base a dati di cooccorrenza e tecniche di clustering. Questo tipo di soluzione tipicamente "bottom-up" migliora le possibilità di navigazione, ma da un lato lascia irrisolto il problema della mancanza di gerarchia e dall'altro, appoggiandosi soltanti sui dati empirici, non aggiunge nessuna informazione semantica esplicita.

1.1 Obiettivi e motivazioni

I due approcci fondamentali per presentare informazioni semantiche, quello gerarchico, top-down, delle tassonomie costruite a priori, e quello partecipativo, bottom-up, delle folksonomie, presentano entrambi delle limitazioni. Questo lavoro si colloca nello spazio fra i due approcci, come un esperimento di integrazione di alcuni aspetti di entrambi.

L'obiettivo è quello di migliorare le possibilità di esplorazione e di ricerca in una folksonomia, integrando nell'interfaccia di navigazione relazioni semantiche esplicite fra le tag, definite a priori. In particolare lo scopo è arricchire le possibilità di esplorazione di insiemi di tag correlate, organizzandole all'interno di una gerarchia di concetti, la cui struttura sia ricavata da un'ontologia; il risultato finale atteso è un'interfaccia ibrida, dove i dati rappresentati siano scelti in base a calcoli sulla cooccorrenza fra le tag, ma organizzati secondo un criterio semantico fissato.

Questa soluzione può permettere da una parte di sfruttare le preziose informazioni semantiche create dagli utenti di una folksonomia per associare le parole chiave alle risorse e per estrapolare dati sulla loro correlazione, dall'altra di appoggiarsi alla struttura coerente e organica di un'ontologia per aiutare la navigazione dell'utente.

Poiché l'integrazione proposta avviene a valle del processo di classificazione delle risorse da parte degli utenti, non comporta nessuno snaturamento della folksonomia, nessuna perdita in termini di flessibilità, di inclusività, di democraticità: arricchisce soltanto l'interfaccia di navigazione, aggiungendo una nuova funzione a uno strumento esistente.

Diversi dei principali problemi delle folksonomie possono essere in parte risolti con una soluzione di questo tipo: in primo luogo il problema dello spazio piatto costituito dalle tag, e dell'impossibilità di una visione organica del dominio; in secondo luogo il problema della mancanza di *recall*, che è legato al controllo dei sinonimi, poichè nella gerarchia di concetti i sinonimi si trovano generalmente vicini; infine anche il problema del *gaming* può essere in parte limitato dalla presenza di una struttura semantica predefinita, che confina ogni tag in una posizione e lascia più spazio per le altre.

1.2 Contributi originali

La prima idea originale di questo lavoro è quella da cui esso prende le mosse: integrare una gerarchia di concetti nell'interfaccia di navigazione di una folksonomia, effettuando la mappatura semantica delle tag su un'ontologia.

La realizzazione di questa idea ci ha posto di fronte a diversi problemi, dovuti alle diversità intrinseche fra una folksonomia e un'ontologia e alle conseguenti difficoltà di effettuare una mappatura fra i due domini; in particolare l'ontologia che abbiamo scelto di utilizzare è la gerarchia di sostantivi di WordNet, un lessico semantico che comprende buona parte delle parole della lingua inglese.

Il primo problema è quello delle parole non contenute nell'ontologia, che devono essere escluse dal processo e dal risultato finale; a un primo sguardo questo problema può sembrare compromettere la validità e l'utilità dell'intero sistema, poichè solo una piccola percentuale delle tag è contenuta nell'ontologia scelta. Abbiamo mostrato, con test effettuati su un campione consistente di dati, che la grande maggioranza delle tag più popolari sono contenute in WordNet e sono sostantivi, e in particolare che questo dato segue una distribuzione di tipo *power law*. Questo dimostra che le tag che si perdono sono quelle meno usate, che spesso hanno senso solo per l'utente che le utilizza, mentre la maggior parte delle tag di uso frequente e condiviso possono essere associate a concetti dell'ontologia.

Il secondo problema principale che si è posto, relativamente alla mappatura semantica, è quello dell'ambiguità delle tag: molte parole infatti possono avere più significati, anche molto distanti fra loro, mentre le relazioni semantiche di WordNet sono definite per unità di significato: per effettuare la mappatura abbiamo quindi affrontato il problema disambiguando ogni occorrenza di una tag rispetto alla risorsa a cui si riferisce. Per questo passaggio ci siamo serviti di una libreria di algoritmi che si basano su WordNet per calcolare misure di correlazione fra le parole; come contesto per la disambiguazione abbiamo usato le altre tag attribuite alla stessa risorsa.

Un altro problema particolare, relativo soprattutto alla presentazione dei dati all'utente, è la granularità molto sottile di WordNet e il numero di livelli della gerarchia troppo elevato; per risolverlo abbiamo sviluppato un algoritmo di compressione dell'albero, basato sull'eliminazione delle categorie di alto livello, concetti troppo generali per essere interessanti e utili, e i nodi intermedi non essenziali per la struttura dell'albero e non presenti come tag. Grazie a questo algoritmo l'albero risultante è più compatto, ha un numero di livelli e un fattore di ramificazione limitati e si presta meglio all'esplorazione.

Un aspetto originale di questo lavoro consiste nell'adozione del principio della *navigazione attiva*: l'utilizzo di un'estensione del browser che permette all'utente di eseguire codice JavaScript locale, modificando dinamicamente i contenuti delle pagine Web che sta visitando. Mentre tutto il processo di raccolta ed elaborazione dei dati è svolto dal server, lo script realizzato permette di richiedere ad esso l'albero semantico delle tag correlate a una certa parola chiave e di integrare dinamicamente il contenuto della pagina visualizzata dal browser con le nuove informazioni, appena le riceve. L'intero processo è trasparente all'utente, che deve solo installare lo script per l'estensione del browser; una volta che questo è installato, si attiverà automaticamente ogni volta che l'utente accede alla pagina di del.icio.us relativa a una tag. L'integrazione della nuova funzione nelle pagine di del.icio.us è un passaggio delicato che è stato affrontato in modo da garantire la massima coerenza, sia logica sia grafica, con l'interfaccia di navigazione della folksonomia.

Fra i problemi affrontati, infine, uno non convenzionale, più di basso livello, è quello del recupero dei dati dalla folksonomia: non è possibile infatti, con gli strumenti messi appositamente a disposizione per interfacciarsi in modo automatico con il sistema, accedere a una quantità sufficiente di informazioni; per questo motivo abbiamo realizzato un crawler, che è in grado di scandire le pagine HTML estraendo i dati necessari.

1.3 Struttura della tesi

Nel capitolo 2 abbiamo introdotto il contesto generale in cui si colloca questo lavoro, che è quello dei sistemi collaborativi e del Web2.0. Abbiamo poi descritto e confrontato fra loro i principali modelli di classificazione e le loro possibilità di applicazione nel Web; abbiamo così introdotto i sistemi basati su tassonomie e su *facet* e le folksonomie. Riguardo a queste ultime, che rappresentano il punto di partenza su cui si sviluppa questo lavoro, abbiamo analizzato lo stato dell'arte, illustrando i principali studi sul fenomeno e descrivendo alcune applicazioni significative.

Nel capitolo 3 abbiamo esposto gli obiettivi di questo lavoro, e abbiamo

definito i requisiti dell'applicazione e le principali scelte ambientali. Poichè l'obiettivo riguarda il miglioramento delle possibilità di esplorazione di una folksonomia, due sezioni particolari sono state dedicate agli aspetti più strettamente legati all'interfaccia con l'utente finale del sistema. Nel paragrafo 3.5 sono analizzate le possibilità di navigazione offerte dall'applicazione scelta come base per questo lavoro, del.icio.us, e le limitazioni che esse presentano; nel paragrafo 3.6 sono stati definiti i requisiti dell'interfaccia utente del nuovo sistema.

Gli aspetti relativi al progetto e alla realizzazione dell'applicazione sono trattati nel capitolo 4, in cui abbiamo illustrato l'architettura generale del sistema, le varie parti che lo costituiscono e l'interazione fra di esse. La struttura è basata su un paradigma client-server, dove il server è costituito da più elementi: il crawler che estrae i dati dall'interfaccia Web di del.icio.us, il modulo per la disambiguazione delle tag, la base di dati per memorizzare i dati elaborati da questi due moduli, il Web server che costruisce l'albero gerarchico delle tag, la base di dati di WordNet. Nel capitolo trovano spazio le descrizioni di ciascuno di questi elementi, delle relative scelte progettuali e degli aspetti più rilevanti riguardo alla realizzazione; particolare attenzione è stata dedicata agli algoritmi via via presentati e a osservazioni sulla complessità, soprattutto in merito a quelli eseguiti in tempo reale dal Web server, che abbiamo mostrato avere complessità lineare. L'ultimo paragrafo tratta il lato client, e descrive lo script che permette di presentare i dati elaborati dal server all'utente finale, integrandoli con l'interfaccia di navigazione Web di del.icio.us.

Nel capitolo 5 abbiamo illustrato i risultati qualitativi ottenuti. Nel primo paragrafo abbiamo descritto e mostrato attraverso delle immagini l'interfaccia del sistema realizzato, abbiamo eseguito e mostrato test sperimentando il funzionamento del sistema con diversi dati e confrontandolo con le funzioni offerte da altre applicazioni. Infine abbiamo discusso i risultati, in termini di prestazioni e di tempi di esecuzione, dei vari moduli dell'applicazione e in particolare del Web server.

Nel capitolo 6 infine abbiamo esposto le conclusioni e illustrato le linee di sviluppo e le possibili evoluzioni del sistema realizzato.

Capitolo 2

Sistemi per la classificazione collaborativa

2.1 Sistemi collaborativi

La crescita di internet sta creando possibilità prima impensabili di interazione e di collaborazione fra persone, anche lontane geograficamente. Non è sorprendente che siano stati proprio fra gli "addetti ai lavori", sviluppatori di software, i primi ad accorgersi della potenza della rete come strumento per la realizzazione di compiti anche complessi in modo collaborativo: lo sviluppo di Linux è stato forse il più grande progetto collaborativo della storia dell'uomo [Torvalds and Diamond, 2001].

Nel 1991 il giovane studente finlandese Linus Torvalds decise di rendere disponibile su internet il codice del rudimentale clone del kernel di Unix da lui sviluppato, invitando a provarlo e a collaborare per migliorarlo. Gli utenti del nuovo sistema erano trattati di fatto come co-sviluppatori, invitati a dare pareri e a proporre le proprie modifiche; attorno a Linus si formò una comunità virtuale di migliaia di utenti-sviluppatori che fecero crescere il sistema partecipando attivamente. Ogni settimana le nuove proposte e modifiche erano raccolte e sottoposte a critiche da parte di tutti; le migliori erano accettate in una sorta di processo di selezione naturale nella comunità. In pochi anni da un progetto portato avanti nel tempo libero, solo per passione, da volontari di tutto il mondo, era nato un sistema operativo all'altezza dei principali prodotti commerciali del momento.

L'innovazione, sociologica prima che tecnica, che aveva permesso questo

successo, fu studiata da Eric S Raymond [Raymond, 1997], che introdusse la celebre metafora della cattedrale e del bazaar:

- Nel modello della **cattedrale** tutto viene progettato nei dettagli da una persona o da un gruppo ristretto che dà le direttive; per gli altri si tratta solo di realizzare le singole parti del progetto, lavorando ciascuno per proprio conto, in quasi totale isolamento. L'intero processo si svolge a porte chiuse; all'esterno saranno visibili solo i risultati definitivi.

- Nel modello del **bazaar** invece l'ideazione è aperta a tutti e non si limita alla fase iniziale; ogni idea viene resa da subito disponibile agli altri per essere valutata, criticata ed eventualmente modificata e migliorata. Il risultato viene raggiunto dalla comunità attraverso una sorta di selezione naturale delle proposte migliori. Anche gli utenti sono importanti, tutti possono contribuire in quanto portatori di un'esperienza e di un punto di vista. La molteplicità dei punti di vista è proprio la forza del bazaar, dove una nuova idea brillante trova subito persone pronte a lavorarci sopra per integrarla nel progetto.

Il modello della cattedrale è quello tradizionale, dominante nelle aziende produttrici di software commerciale e anche nei primi progetti di software libero. Con lo sviluppo del kernel di Linux il mondo open source passa, secondo Raymond, al nuovo modello del bazaar, che è reso possibile dalla rete. Non si tratta solo dell'apertura del codice sorgente, ma anche di un nuovo modello sociale di cooperazione, fortemente legato all'etica hacker [Himanen, 2001]: la condivisione e la circolazione delle informazioni devono essere totali; le gerarchie tradizionali sono abbandonate per essere sostituite da gerarchie sociali implicite, basate sui meriti e sulle abilità dimostrate da ciascuno; la motivazione per il lavoro non è per forza legata al denaro, ma piuttosto al "divertimento", alla sfida intellettuale e al riconoscimento delle propria abilità nella "comunità dei pari".

É chiaro che il modello del bazaar non si limita all'ambito dello sviluppo di software, ma può trovare terreno fertile anche in altri contesti, in particolare quelli che hanno a che fare con l'organizzazione della conoscenza; alcuni esempi ormai affermati che mostrano questa possibilità sono Wikipedia [1], un'enciclopedia libera online, redatta collaborativamente dagli utenti stessi, o sistemi per la classificazione distribuita di risorse online, detti *folksonomie*, come Flickr [2] per le immagini e del.icio.us [3] per i link. Questo tipo di ap-

[1] http://wikipedia.org
[2] http://flickr.com
[3] http://del.icio.us

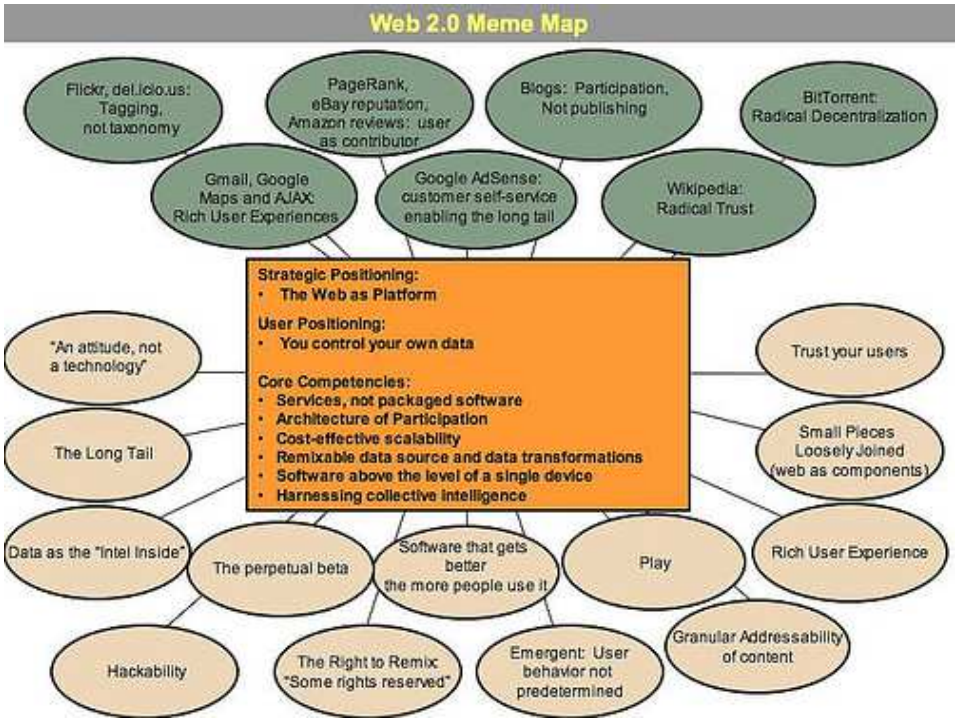

Figura 2.1: la figura mostra una mappa dei concetti che ruotano intorno all'idea di Web2.0; essa è il frutto di una sessione di brain storming svolta durante il FOO Camp, una conferenza alla O'Reilly Media

plicazioni Web, incentrate sulla partecipazione attiva da parte degli utenti, sono alla base di quello che viene chiamato Web2.0.

2.2 Web2.0

Il termine Web2.0 è stato coniato dalla O'Railly Media nel 2004 ed è oggi largamente utilizzato per indicare la trasformazione del Web a cui stiamo assistendo in questi anni.

Da quello che è stato definito retroattivamente Web1.0, un Web fatto per lo più di siti statici, è in atto una transizione verso un paradigma più dinamico e interattivo, con applicazioni basate fortemente sul contributo e sul ruolo attivo degli utenti. Questa transizione può essere illustrata con una serie di esempi, come il passaggio dai siti personali ai blog, dai CMS ai wiki, da server centralizzati al *peer to peer*, dalle tassonomie ai sistemi a tag (folksonomie), dalla *stickyness* dei siti che cercano di tenere incollati a

sé gli utenti alla *syndication* per rendere fruibili i contenuti attraverso canali differenti.

Dal punto di vista tecnologico, soluzioni come AJAX permettono una comunicazione più leggera e dinamica fra client e server, aumentando le possibilità di interazione dell'utente fino a rendere le applicazioni Web sempre più simili alle applicazioni tradizionali eseguite sul proprio computer, standard come XML e RSS favoriscono l'interoperabilità delle applicazioni e la disponibilità dei contenuti in modo indipendente dal canale e dalla presentazione.

La definizione più nota e più completa del concetto di Web2.0 è probabilmente quella che ha elaborato Tim O' Reilly nell'articolo [O'Reilly, 2005]. La prima immagine che O' Reilly introduce è quella Web come piattaforma.

Se il simbolo del Web1.0 era Netscape, quello del Web2.0 è Google. Google è nato come applicazione Web: non come un software con una licenza, da acquistare o distribuire, ma come un servizio che viene reso fruibile sul Web e viene pagato in qualche modo, diretto o indiretto, dagli utenti. La portabilità è massima, non c'è bisogno di installazione. Il servizio si colloca nello spazio del Web, fra il browser dell'utente e i server di destinazione; il suo valore è dato dalla capacità di organizzare le informazioni, ed è proporzionale alla quantità e alla dinamicità dei dati che riesce a gestire.

Una caratteristica del Web2.0 è la centralità dei dati; quello che accomuna tutte le più importanti applicazioni che si sono affermate nel Web, da Google a Amazon, da eBay a Napster, è la presenza di un database specializzato: tanto che per queste applicazioni è stato coniato il termine di *infoware* [O'Reilly, 1997].

Un altro aspetto fondamentale per le applicazioni Web della nuova generazione è la capacità di "imbrigliare l'intelligenza collettiva", secondo le parole di Tim O' Reilly, o di valorizzare la "saggezza della folla" [Surowiecki, 2004]. Questo è quello che fa Wikipedia raccogliendo contributi volontari appositamente inseriti dagli utenti, ma anche Google con l'algoritmo PageRank [Brin and Page, 1998], sfruttando l'"intelligenza" contenuta nei link che costituiscono la struttura del Web e della "blogosfera". É anche il principio che sta alla base dei sistemi collaborativi di filtraggio dello spam, che studiano grandi quantità di decisioni dei singoli utenti per "imparare" a riconoscere cosa è e cosa non è spam, o dei sistemi collaborativi per la classificazione di risorse, le folksonomie.

Una questione centrale è quella di come ottenere la partecipazione degli utenti. Questa è stata affrontata da Dan Bricklin, che ha studiato in par-

ticolare il successo di Napster [Bricklin, 2000]. Dan Bricklin osserva che ci sono fondamentalmente tre modi per riempire un database condiviso:

- organizzato manuale;
- organizzato meccanico;
- volontario manuale.

Per quanto riguarda i database del terzo tipo, come Napster, il punto centrale è come viene ottenuto il contributo degli utenti: in Napster condividere una canzone che si è appena scaricata non è un gesto che richieda alcuno sforzo aggiuntivo, né una sensibilità particolare; al contrario, è la scelta più comoda, quella a cui si viene portati in modo naturale dall'interfaccia utente, in quanto è la scelta di default. Questa caratteristica dell'interfaccia utente è proprio quella che ha permesso al database di Napster di raggiungere le sue dimensioni, e che ha contribuito in modo fondamentale al successo dell'applicazione.

Il numero degli utenti, che in molti contesti viene considerato un problema, in quanto essi consumano risorse, è invece la forza di un sistema come Napster: ogni utente in più, col suo comportamento, aggiunge valore al sistema. Questo è uno dei principi chiave del Web2.0, così come la capacità di ottenere in qualche modo il contributo degli utenti è una delle sfide per le nuove applicazioni Web.

2.3 Schemi di classificazione nel Web

La crescita impetuosa della quantità di informazioni disponibili sul Web, a cui stiamo assistendo in questi anni, rende sempre più centrale il problema della loro organizzazione e, in particolare, della classificazione.

Esistono due tipi di approcci fondamentali a questo problema:

- **enumerativo-gerarchico**: è un approccio di tipo top-down, basato su *tassonomie* di categorie via via più specifiche, fra le quali non è previsto generalmente overlap. Ogni elemento ha un'unica posizione all'interno della gerarchia.
- **analitico-sintetico**: è un approccio di tipo bottom-up, basato su *facet*, ovvero diversi aspetti in base ai quali uno stesso oggetto può essere descritto. Man mano che nuovi elementi vengono classificati, nuove categorie possono emergere e nuovi concetti possono essere integrati senza interferire con l'attività di classificazione precedente.

La soluzione più tradizionale è basata sul primo approccio: un numero ristretto di "esperti" stabilisce in modo coerente una tassonomia per il dominio, una struttura gerarchica di categorie; in seguito ogni elemento dovrà rientrare in una di queste categorie. Se vogliamo restare nella metafora di Raymond, si tratta del modello della cattedrale. Le *folksonomie* si basano sul secondo tipo di approccio e rappresentano abbastanza bene il modello del bazaar applicato al problema della classificazione; esse sono per molti aspetti più adatte delle tassonomie nell'ambito del Web, come apparirà chiaro nel corso di questo capitolo.

2.3.1 Tassonomie

L'approccio gerarchico alla classificazione è quello a cui siamo abituati, e quello che in molti contesti ci appare più naturale: esso sottintende un punto di vista unitario e coerente su ciò che deve essere classificato, e fra l'altro risponde all'esigenza fisica del mondo reale di collocare ogni oggetto in una sola posizione.

L'esempio più classico è quello delle biblioteche, dove sono stati sviluppati sistemi che assegnano ad ogni libro un codice univoco, e una sola categoria di appartenenza in una gerarchia. É molto facile, per un bibliotecario che conosca le convenzioni del sistema e la gerarchia di categorie, orientarsi e trovare la posizione esatta di un libro richiesto.

Questo modello ha il pregio di basarsi su un'organizzazione coerente degli oggetti e di fornire una visione organica del contesto; esso facilita il compito di trovare un oggetto preciso.

I risultati di questo approccio sono buoni in molti ambiti, ma non in tutti. I limiti della classificazione basata su tassonomie nell'ambito del Web, in particolare, sono stati evidenziati da Clay Shirky in [Shirky, 2005b], dove mostra come il modello gerarchico possa dare buoni risultati su domini che abbiano tendenzialmente queste caratteristiche:

- corpus relativamente limitato;
- categorie predefinite;
- entità stabili e ristrette;
- confini chiari.

Il problema è quello di avere un insieme di categorie stabili nel tempo, che contengano entità omogenee e che possano comprendere tutti gli elementi del dominio.

Accanto alle caratteristiche del dominio, è poi opportuno che si verifichino alcune condizioni legate alle persone coinvolte nel processo di classificazione:

- la presenza di catalogatori esperti;
- una fonte autorevole;
- utenti esperti.

Occorre infatti che le categorie, create una volta per tutte a priori, possano essere accettate e comprese a pieno dalle persone che dovranno utilizzarle per eseguire la classificazione, e da quelle che dovranno orientarsi per effettuare le ricerche.

Il Web costituisce un dominio che ha caratteristiche per lo più opposte a quelle fin qui elencate:

- il corpus è potenzialmente illimitato;
- il contesto è in continua e rapida trasformazione;
- gli utenti sono generalemente inesperti;
- non c'è un'autorità comunemente accettata.

Al crescere delle dimensioni del dominio, dell'eterogeneità dei contesti e dei punti di vista, l'approccio tradizionale e gerarchico incontra molte difficoltà e non si rivela efficace.

Un altro aspetto caratteristico del Web è stato sintetizzato sempre da Clay Shirky nella formula "there is no shelf": nel contesto della categorizzazione di risorse online "non c'è nessuno scaffale", ovvero le risorse non hanno un'esistenza fisica e non c'è la necessità di tenere ogni elemento in un unico luogo preciso. Questa è la tipica necessità di una biblioteca, dove un libro deve stare in un determinato scaffale, e in parte anche di un filesystem, in cui ogni file ha una collocazione in un'area di memoria. Nel Web questa necessità scompare: non ci sono scaffali, ci sono solo link; per questo le tassonomie si rivelano spesso troppo rigide in questo contesto.

Yahoo! rappresenta uno dei primi tentativi di classificare i siti Web e si basa sull'idea tradizionale di classificazione. L'idea dei due giovani fondatori di Yahoo! era quella di creare una tassonomia che potesse avere valore per tutto il Web, in modo che ogni sito potesse rientrare in una categoria definita. La vastità del Web, l'eterogeneità delle risorse, degli utenti e dei loro punti di vista, la rigidità intrinseca di una tassonomia hanno reso quella di Yahoo! un'impresa sempre più difficile da realizzare: mantenere una struttura adeguata, di categorie di valore generale si è rivelata una sfida molto

difficile. Non a caso, come motore di ricerca si è affermato invece Google, considerato il simbolo del Web2.0, che ha scelto di basarsi soltanto sui link fra le pagine Web.

2.3.2 Faceted navigation

I facet possono essere visti come uno spazio cartesiano n-dimensionale, in cui il valore di ognuno di essi corrisponde alla posizione dell'oggetto da classificare nella dimensione corrispondente. Ogni oggetto viene quindi considerato sotto vari aspetti, che sono ortogonali fra loro.

Questo approccio consente di avere gerarchie, ma è più flessibile di quello basato su tassonomie; ogni facet può essere specializzato in categorie via via più specifiche, fra le quali ci può essere overlap.

Nell'ambito della classificazione libraria, i facet hanno trovato spazio in alcuni sistemi, come la Bliss bibliographic classification [Broughton, 2002].

Questo approccio ha avuto un certo successo recentemente nel Web, per creare interfacce di navigazione [Denton, 2003, English et al., 2002]. Esso permette una struttura più flessibile e più facilmente integrabile; è facile infatti aggiungere nuovi elementi: associandoli a uno o più facet, in un certo punto della rispettiva gerarchia.

Uno dei primi siti a realizzare caratteristiche di *facet browsing* è stato wine.com[4]: esso presenta un catalogo di vini, consultabile secondo vari criteri: vini bianchi, rossi o rosati, dolci o secchi, provenienti da una certa regione piuttosto che da un'altra e così via. Gli aspetti possono essere combinati fra loro in un'interfaccia che supporta questo tipo di navigazione. Un altro esempio di sito molto popolare è Epicurious[5], che raccoglie una grande quantità di ricette, accessibili attraverso un'interfaccia basata su facet.

Per la realizzazione di un sistema di navigazione di questo tipo sono necessari dei metadati, che definiscano dei facet associando ad essi delle gerarchie di categorie.

I metadati vengono generalmente creati da esperti; da qualche anno si sta creando interesse attorno ad alcuni studi per la creazione automatica di metadati per creare una struttura di facet [Dakka et al., 2005]. Un progetto particolarmente interessante che si sta muovendo in questa direzione, anche attraverso sistemi semiautomatici, è il progetto Flamenco[6] [English et al., 2002].

[4] http://www.wine.com/

[5] http://www.epicurious.com/

[6] La homepage del progetto si trova all'indirizzo http://flamenco.berkeley.edu/

Un esperimento interessante è stato realizzato, per esempio, per realizzare un'interfaccia per la ricerca di immagini [Yee et al., 2003].

2.3.3 Folksonomie

Una *folksonomia* è un sistema di classificazione collaborativo bottom-up, basato su etichette (*tag*). Il termine *folksonomy* è stato introdotto per la prima volta nel 2004 da Thomas Vander Wal e deriva dalla fusione delle parole *folk* e *taxonomy*. Accanto al termine folksonomia ne sono stati proposti altri, come *etnoclassificazione* [Merholz, 2004], che possono essere considerati più appropriati, in quanto non si tratta di un tipo di tassonomia ma piuttosto di un tipo di classificazione del tutto alternativo a quello basato su tassonomie. Tuttavia poichè esso è ormai ampiamente diffuso, utilizzeremo il termine folksonomia.

Gli utenti possono attribuire alle risorse una o più parole chiavi (tag) a propria scelta; non ci sono restrizioni sulle tag, qualsiasi sequenza di caratteri che abbia senso per un utente può essere utilizzata.

Il principio alla base di una folksonomia è quello secondo cui ogni utente agisce prima di tutto per la propria comodità personale, etichettando le risorse per poterle ritrovare più facilmente in seguito. Poichè le tag sono in genere disponibili pubblicamente, i gesti di ogni utente fanno crescere il sistema. "É fondamentalmente un modo per ricordare in pubblico", secondo le parole del fondatore di del.icio.us, Joshua Schachter.

Thomas Vander Wal ha introdotto la distinzione fra due tipi di folksonomie: *broad* e *narrow*[Vander Wal, 2005].

- Nelle folksonomie **broad** ogni oggetto può essere etichettato da tutti gli utenti, ciascuno secondo i propri modelli mentali, la propria lingua e il proprio vocabolario. La potenza di una folksonomia broad sta nella ricchezza portata dal numero degli utenti e dalla molteplicità dei punti di vista su uno stesso oggetto. La curva caratteristica che tende a emergere in un sistema di questo tipo mostra una distribuzione di tipo *power law*[Biddulph, 2004, Hyde, 2005]: alcune tag tendono ad affermarsi fortemente, in quanto sono le più viste e "copiate"; molte applicazioni incoraggiano questo meccanismo con un sistema di suggerimenti, che porta l'utente a scegliere per un sito le tag più utilizzate dagli altri. Così accade che poche tag sono utilizzate da un alto numero di utenti, mentre rimane una grande quantità di tag usate ciascuna da pochi utenti, spesso uno solo. Questo secondo fenomeno è stato de-

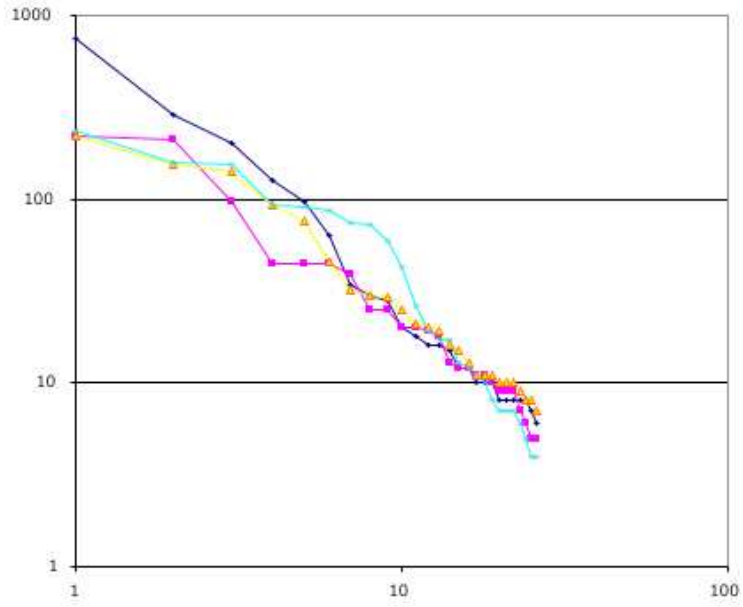

Figura 2.2: il grafico, pubblicato sul Web all'indirizzo http://enthusiasm.cozy.org/archives/2005/01/tagging-powerlaw/, mostra le tag attribuite a quattro siti popolari di del.icio.us. Ogni linea colorata rappresenta un sito, e ogni punto una parola chiave ad esso attribuita; in ordinata è rappresentato il numero di utenti che hanno utilizzato le tag, in scala logaritmica. Le tag sono ordinate sull'asse orizzontale in base alla frequenza. La curva mostra una tipica distribuzione power law.

nominato *long tail* [Anderson, 2004], la lunga coda delle parole chiave che spesso hanno un senso solo per chi le utilizza.

- Nelle folksonomie **narrow** ogni oggetto può essere etichettato solo da una o poche persone. In genere la persona o il gruppo che ha la possibilità di etichettare un oggetto è legata a questa risorsa in qualche modo particolare: tipicamente può essere chi l'ha pubblicata. In una folksonomia narrow le tag sono singole per natura, nel senso che possono essere associate una sola volta ad ogni risorsa: non c'è la power law.

Un aspetto importante, ben descritto in [Mathes, 2004], è quello del *browsing* contro il *finding*, e di *discovery* contro *searching*: le folksonomie privilegiano una modalità di navigazione basata sull'esplorazione piuttosto che sulla ricerca. L'utente spesso non ha un'idea precisa dell'oggetto che vuole trovare, ma può partire da una risorsa o da una parola chiave a cui è interes-

sato per trovare un proprio percorso all'interno dello spazio sterminato del Web. Le tag favoriscono la *serendipity*, mentre non sono adatte alla ricerca di una risorsa precisa (se non nell'ambito personale di un singolo utente che debba ritrovare ciò che ha classificato con una parola chiave, caso in cui si rivelano invece efficaci).

É generalmente possibile effettuare ricerche utilizzando più tag combinate insieme per restringere il campo, secondo un principio simile a quello su cui si basa la *faceted navigation*.

Ecco i principali vantaggi offerti dalle folksonomie:

- sono "democratiche", nel senso che non c'è un'autorità centrale che stabilisce le categorie, ma sono gli utenti stessi che le definiscono e i concetti maggiormente condivisi tenderanno ad emergere;
- rispecchiano maggiormente gli utenti: non essendoci la mediazione di catalogatori esperti, che devono "leggere loro nella mente" per decidere quali categorie siano più adatte, il vocabolario risulta più vicino agli utenti;
- sono flessibili: nuove categorie possono essere sempre introdotte;
- sono inclusive: poichè non c'è nessuna autorità né struttura top-down imposta, ogni punto di vista può trovare spazio in una folksonomia; esplorando la *long tail* si può accedere anche a idee originali e non-mainstream;

I principali problemi e le limitazioni invece possono essere sintetizzate nei seguenti punti:

- mancanza di precisione, dovuta all'ambiguità delle parole e all'assenza di standard condivisi per la definizione delle categorie;
- mancanza di gerarchia: le tag rappresentano uno spazio piatto e questo è un limite per le possibilità di ricerca e di esplorazione;
- nessun controllo dei sinonimi;
- scarsa "trovabilità", o mancanza di *recall*: a causa dell'eterogeneità delle parole scelte dagli utenti e dei loro punti di vista, e dell'assenza di vocabolario controllato, è difficile ottenere tutte le risorse correlate a un certo ambito;
- possibilità di inquinare il sistema per i propri scopi (*gaming*).

Nonostante i problemi e le limitazioni anche rilevanti a cui sono soggette, le folksonomie sono state definite da Shirky come una "mossa forzata", definizione largamente accettata in letteratura: in risposta alla diffusione dei blog e della possibilità di pubblicare contenuti in modo amatoriale sul Web, la "amatorializzazione della classificazione" dei contenuti diventa un passaggio necessario [Shirky, 2005a].

2.4 Folksonomie: stato dell'arte

É interessante osservare che le folksonomie non sono il frutto di una teoria, ma piuttosto è vero il contrario: esse sono nate dalla scelta implementativa di alcuni software per la condivisione di risorse online, che hanno introdotto la possibilità di associare liberamente delle parole chiave ai contenuti; l'interesse degli utenti per questo tipo di sistemi li ha portati a un rapido successo e ne ha favorito la diffusione.

Per questo motivo i primi studi sul fenomeno sono dovuti per lo più a blog o discussioni nei forum di "addetti ai lavori", come il noto post di Clay Shirky sul blog collettivo "Many to many" [Shirky, 2004], il riscontro e l'illustrazione del fenomeno della *power law* nelle folksonomie in [Biddulph, 2004, Hyde, 2005], la definizione delle folksonomie *broad* e *narrow* spiegata in [Vander Wal, 2005], la questione critica dell'*ecologia dei metadati* in assenza di vocabolari controllati, posta in [Rosenfeld, 2005], la questione delle diverse lingue che possono coesistere in una folksonomia [Dijck, 2005], la descrizione dettagliata del funzionamento di del.icio.us con uno *screencast* [Udell, 2005].

Uno dei primi articoli organici sull'argomento è uno studio di del.icio.us: [Mejias, 2004]; altre descrizioni complete del fenomeno delle folksonomie, che raccolgono tutto il materiale precedente in modo organico e con alcuni contributi originali, si trovano in [Mathes, 2004] e [Quintarelli, 2005].

Solo in una seconda fase, quando le folksonomie si sono affermate in vaste aree del Web, esse hanno iniziato ad attrarre anche l'attenzione di ambienti accademici e ad essere oggetto di pubblicazioni. Fra gli studi più specifici sulla dinamica e la semantica delle folksonomie, basati anche su dati quantitativi, segnaliamo [Golder and Huberman, 2005] e [Shepard et al., 2006]. Alcune pubblicazioni interessanti, riguardo la realizzazione di algoritmi di *ranking* per le folksonomie si trovano in [Szekely and Torres, 2005] e [Hotho et al., 2006].

La necessità di interoperabilità fra applicazioni e della definizione di standard condivisi per le folksonomie è oggetto di diversi studi. In particolare alcune proposte per spingere le folksonomie nella direzione del Web semantico [Berners-Lee et al., 2001], con la creazione di *folksologie*, ontologie di folksonomie, sono state formulate in [Gruber, 2005].

2.4.1 Applicazioni esistenti

Il successo dei primi sistemi di classificazione basati sul tagging ha prodotto una proliferazione nel Web di sistemi di questo tipo negli ambiti più svariati, dai bookmark agli articoli scientifici (come CiteULike[7]), dalle fotografie (come Flickr[8]) ai video (come YouTube[9]), dalle canzoni (come last.fm[10]) alle notizie (come Digg[11]).

Un buon successo hanno riscosso anche *reti sociali* basate su tag, come 43Things[12], che permette di condivere "le 43 cose che vorresti fare nella vita"; incredibilmente da questa improbabile idea è nato un sito che conta oggi più di 900.000 utenti registrati.

Le folksonomie non si prestano solo all'ambito del Web, ma anche all'utilizzo in intranet aziendali; la possibilità di adozione di folksonomie nell'intranet di IBM è stata studiata in [Gibson, 2004].

Nel seguito presenteremo alcune folksonomie particolarmente significative, la cui descrizione è utile ed esemplificativa del funzionamento di questo tipo di sistemi, delle possibilità che essi possono offrire e delle limitazioni a cui sono soggetti.

Del.icio.us

Del.icio.us, il più diffuso sistema di social bookmarking, è stato fondato nel 2003 da Joshua Schachter ed è considerato da molti la prima folksonomia; nel dicembre 2005 è stato comprato da Yahoo!. Ha superato nel settembre 2006 il milione di utenti registrati e nel febbraio 2007 il milione e mezzo[13] ed è in continua e sempre più rapida crescita.

É una folksonomia *broad*: ogni utente può attribuire delle tag a qualsiasi risorsa Web identificabile con un url.

La prima caratteristica di del.icio.us è quella di permettere di organizzare con delle tag i propri bookmark; alla comodità offerta dalla modalità flessibile di classificazione, che facilita molto il compito di ritrovare qualcosa che si è salvato, si aggiunge quella di avere i dati memorizzati su un server Web, e quindi accessibili da qualsiasi calcolatore connesso alla rete.

[7] http://www.citeulike.org/
[8] http://flickr.com/
[9] http://youtube.com/
[10] http://last.fm/
[11] http://digg.com/
[12] http://www.43things.com/
[13] il dato è riportato nel blog di del.icio.us:
http://blog.del.icio.us/blog/2007/02/overdue_new_yea.html

Una volta creato il proprio account, l'utente ha sostanzialmente un altro spazio di ricerca ed esplorazione, oltre a quello dei propri bookmark, che è quello creato da tutti gli altri utenti del sistema; questo è anche uno spazio di condivisione. É possibile mantenere privati i propri bookmark, o alcuni di essi, ma la scelta di default è quella di condividerli. L'utente, anche quando utilizza il sistema solo per la propria utilità personale di ritrovare i link utili in seguito, crea un valore per l'intera comunità.

É disponibile una funzione di ricerca testuale, che funziona come un motore di ricerca all'interno del dominio di del.icio.us, fra le tag, i titoli dei siti, le descrizioni e le note introdotte dagli utenti.

Del.icio.us ha introdotto i *bundle*, ovvero categorie generali che ogni utente può creare per organizzare in gruppi le proprie tag; i bundle sono personali: sono validi e visibili solo all'interno dell'account.

L'aspetto più sociale, che ha a che fare con le relazioni fra gli utenti, era inizialmente solo implicito: ognuno poteva scoprire degli utenti "interessanti", che avessero salvato particolari risorse o usato particolari tag, e memorizzare la pagina del relativo account per restare aggiornati sulle loro attività di tagging. Da aprile 2006 del.icio.us ha implementato una nuova caratteristica di *network*: inserendo altri utenti nel proprio network è possibile monitorare le loro tag attraverso il proprio account, senza bisogno di visitare le loro singole pagine su del.icio.us; inoltre è possibile segnalare dei link alle persone che sono parte del proprio network utilizzando una tag speciale.

Uno studio più completo dell'interfaccia di navigazione di del.icio.us è presentato nel paragrafo 3.5.

Scuttle

Scuttle è una sorta di "clone" *open source* del sistema precedentemente descritto; esso offre molte delle funzionalità di del.icio.us e ne supporta quasi tutte le API. Il codice del progetto è rilasciato sotto licenza GPL e può essere installato su qualsiasi server.

Anche se il server di Scuttle ha molti meno utenti di del.icio.us e non offre particolari funzionalità aggiuntive, il sistema è molto interessante in quanto diverse comunità in tutto il mondo stanno adottando questo software per offrire un servizio di social bookmarking su un proprio server. La compatibilità delle API rende utilizzabili con un server Scuttle molti degli strumenti progettati per interfacciarsi con del.icio.us.

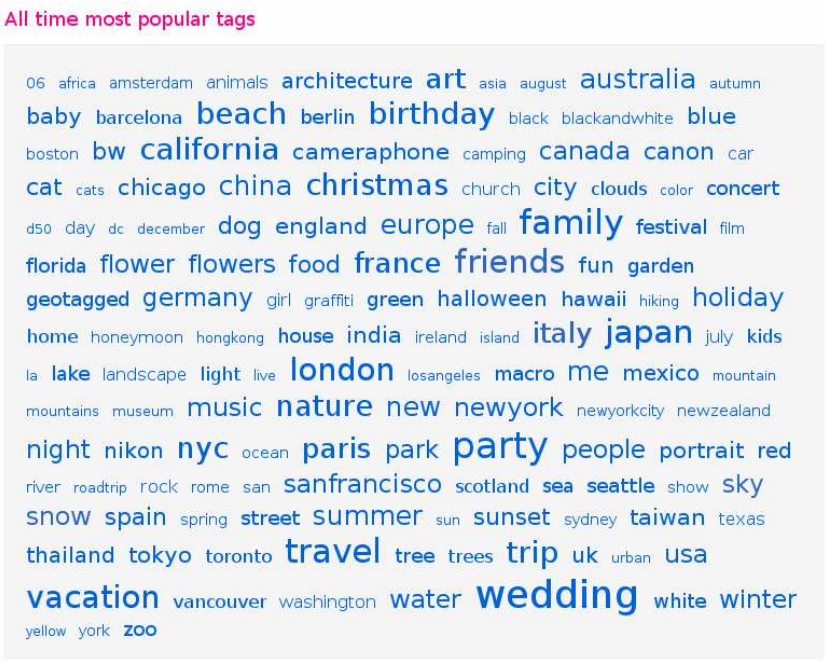

Figura 2.3: la figura mostra la tag cloud *delle parole chiave più popolari in Flickr*

Flickr

Flickr è il più popolare sito Web per la condivisione di fotografie online e una delle più vaste folksonomie esistenti. Lanciato all'inizio del 2004 dalla società canadese Ludicorp, è stato comprato da Yahoo! nel marzo 2005.

Flickr rappresenta un esempio di folksonomia *narrow*; inizialmente solo l'utente che pubblicava una fotografia poteva attribuirle delle tag, mentre per gli altri utenti era possibile solo aggiungere dei commenti. Più tardi è stata introdotta la possibilità di etichettare anche le fotografie di un altro utente, ma solo con il suo consenso.

Flickr è stato il primo sistema (almeno fra quelli di una certa notorietà) a introdurre le *tag cloud*, che hanno trovato poi ampia diffusione come standard nelle folksonomie. Si tratta di una tecnica di rappresentazione visiva di insiemi di parole chiavi (per esempio le tag più popolari, quelle più recenti, quelle di un particolare utente) in cui la dimensione del carattere di ogni tag è proporzionale alla sua frequenza di utilizzo. Generalmente possono essere disposte in ordine alfabetico o di popolarità.

Oltre alle tag, Flickr offre all'utente un altro sistema parallelo di classificazione: i *set*, gruppi di fotografie che cadono sotto uno stesso titolo (per esempio "il mio viaggio in Thailandia"). I set rispetto alle tag rappresentano

una forma di categorizzazione più vicina a quelle tradizionali; assomigliano alle cartelle di un file system, ma sono comunque più flessibili e prevedono overlapping: una foto può appartenere a uno, a nessuno, o a più set.

Da agosto 2005 Flickr ha reso disponibili due nuove funzioni: un sistema di *clustering* delle tag e un algoritmo di ranking delle risorse: la *interestingness*.

La prima caratteristica rappresenta un'innovazione molto interessante relativamente alla presentazione dei dati agli utenti, e alle possibilità di esplorazione di una folksonomia; ecco alcune scelte di Flickr che vale la pena di osservare:

- il clustering è basato sulle parole chiave e non sugli oggetti: non vengono creati cluster di fotografie, ma di tag;
- in particolare vengono "clusterizzate" le parole chiave correlate a una data tag, per definire in qualche modo dei sottoinsiemi di quest'ultima;
- i cluster sono creati unicamente in base a dati empirici sul modo in cui gli utenti utilizzano le tag, estratti dalla folksonomia stessa, senza nessun appoggio su dati semantici definiti a priori;
- il sistema non effettua nessun tentativo di attribuire un nome ai cluster: ogni cluster è descritto dall'insieme delle tag che lo costituiscono, e identificato dalle prime tre;
- non ci sono visualizzazioni grafiche astratte o sofisticate: i cluster sono rappresentati in modo simile alle tag cloud, come insiemi di parole chiave mostrate con corpo dei caratteri variabile a seconda dell'importanza;
- i cluster sono definiti dal sistema e sono fissati, non è possibile per l'utente variare il livello di granularità o cercare sotto-cluster.

Osservando il comportamento dell'algoritmo su diverse tag si può osservare che esso funziona bene in molti casi, quando riesce a cogliere la semantica implicita delle parola chiave grazie ai dati di correlazione e in questo modo a delimitare diversi ambiti semantici relativi a una tag; spesso accade invece che i dati rappresentati non abbiano un senso in relazione agli oggetti che dovrebbero descrivere. Appare evidente che in questi casi i dati di correlazione su cui l'algoritmo si basa non rispecchiano un modello reale o un criterio

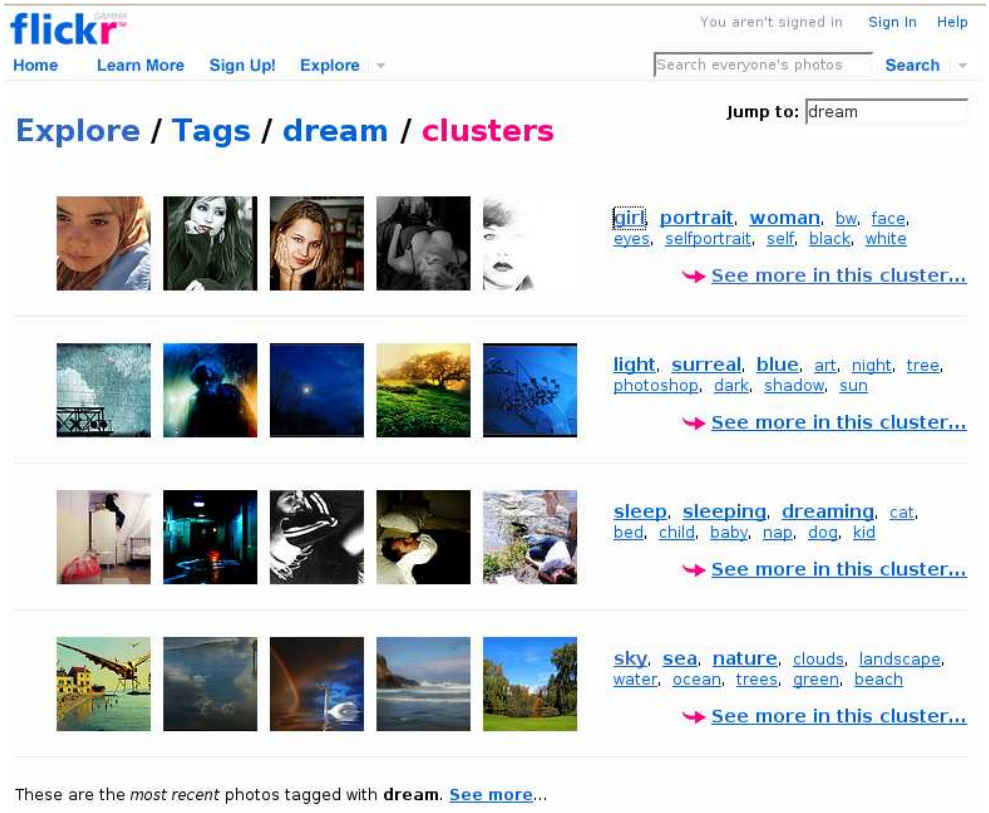

Figura 2.4: la figura mostra i cluster proposti da Flickr per la tag "dream".

sensato: rappresentano informazioni prive di un valore semantico. É il caso del risultato che abbiamo ottenuto per la tag "cat", dove i primi due cluster mostrati dal sistema sono formati dalle seguenti tag:

- (cats, kitty, kitten, pets, animals, sleep, tabby, sleeping, orange, white)
- (animal, pet, eyes, cute, black, feline, gato, bw, portrait, furry)

Si può osservare come tag che sono intuitivamente equivalenti, come "animal" e "animals" si trovino in cluster separati, mentre non si riesce a trovare un criterio che possa giustificare questa suddivisione, che nessun essere umano probabilmente avrebbe mai effettuato.

La seconda caratteristica aggiunta in Flickr è la *interestingness*, una classifica delle fotografie più interessanti, basata sul comportamento degli utenti. L'algoritmo, che può essere considerato simile al PageRank [Brin and Page, 1998] di Google, ma nel contesto differente di una folksonomia, non tiene solo conto del numero delle persone che hanno salvato una fotografia fra i preferiti o che l'hanno commentata, ma anche dell'identità di questi utenti e della loro relazione con la persona che ha pubblicato l'immagine; per esempio, un commento lasciato da un amico vale molto meno di un commento dato da una persona estranea. I dettagli dell'algoritmo sono tenuti nascosti per scoraggiare il *gaming*. La interestingness è anche oggetto di un brevetto, registrato da Yahoo!.

Tagzania

Tagzania [14] è una sorta di folksonomia geografica, dove le risorse che possono essere etichettate dagli utenti sono i luoghi su una mappa del pianeta. Il progetto è stato lanciato dalla società basca CodeSyntax nel 2005 e utilizza le API di Google Maps per gestire la visualizzazione delle mappe. Essa è solo uno dei tantissimi siti basati su questo principio; abbiamo deciso di illustrarla non tanto per l'interesse del sito in sè, ma proprio per la potenzialità che questo tipo di esperienza rappresenta.

Ogni "oggetto" possiede una posizione nella mappa, una descrizione, delle tag. Oltre alle possibilità di esplorazione usuali di una folksonomia ci sono quelle legate alla posizione: si possono visualizzare sulla mappa tutti i luoghi marcati con una determinata tag, piuttosto che tutti gli elementi che si trovano in una certa regione; nella pagina relativa a un oggetto sono segnalati gli utenti più vicini, le tag più vicine e gli oggetti più vicini, dove il concetto di "vicino" ha ovviamente un significato di vicinanza spaziale.

[14] http://www.tagzania.com/

Tool

Parallelamente alle caratteristiche che vengono via via implementate dai siti Web, nascono continuamente nuovi tool creati dagli utenti (proprio nell'accezione che viene data nel Web2.0 alla parola "utente", che implica un contributo attivo) per aggiungere funzionalità ai sistemi esistenti, per personalizzarli o per presentare i dati in altre forme.

Un elenco in costante aggiornamento dei tool che ruotano intorno a del.icio.us si può consultare sul Web[15]; ne illustreremo qui alcuni particolarmente significativi.

Cloudalicious[16] è un sito Web che permette di monitorare l'andamento nel tempo delle tag attribuite dagli utenti di del.icio.us a un sito Web. Dato un url, scelto dall'utente, come input, il sistema raccoglie da del.icio.us le informazioni relative alla storia di quel sito e le rappresenta in un grafico, come quello mostrato in figura 2.5. Il comportamento del sito è tanto semplice quanto efficace: i risultati mostrano informazioni che sono già presenti in del.icio.us e sono ricavabili, per ogni url, da una singola pagina, ma questo tipo di visualizzazione offre un valore aggiunto notevole e i risultati possono essere di grande interesse e utilità.

Fac.etio.us è un progetto sviluppato dalla società Sideran, con l'intento di presentare i bookmark di del.icio.us organizzati tramite facet. Su Web è disponibile una versione demo, che mostra quasi 200.000 link estratti da del.icio.us, organizzati secondo 12 criteri (facet).

Alcuni criteri corrispondono a caratteristiche generali dei bookmark come il dominio del sito Web, oppure la data di sottoscrizione o l'utente che lo ha etichettato (dati che sono ottenuti con feed RSS); altri facet sono ricavati da particolari tag: tramite il facet "place" è possibile visualizzare l'elenco di tutte le tag corrispondenti a nomi di nazioni o di città, mentre il facet "organization" raccoglie tutte le parole chiave corrispondenti a nomi di organizzazioni.

É possibile scegliere per esempio fra le attività (all'interno del facet "activity") la musica, e visualizzare tutti i link che contengono tale tag. Fra questi è possibile poi restringere il campo utilizzando altri criteri, come la data in cui sono stati sottoscritti o l'attributo (per esempio "cool" o "free").

L'idea è molto interessante in quanto le tag si prestano bene per la fa-

[15] all'indirizzo http://www.econsultant.com/delicious-by-function/

[16] http://cloudalicio.us/

[16] http://demo.siderean.com/facetious/facetious.jsp

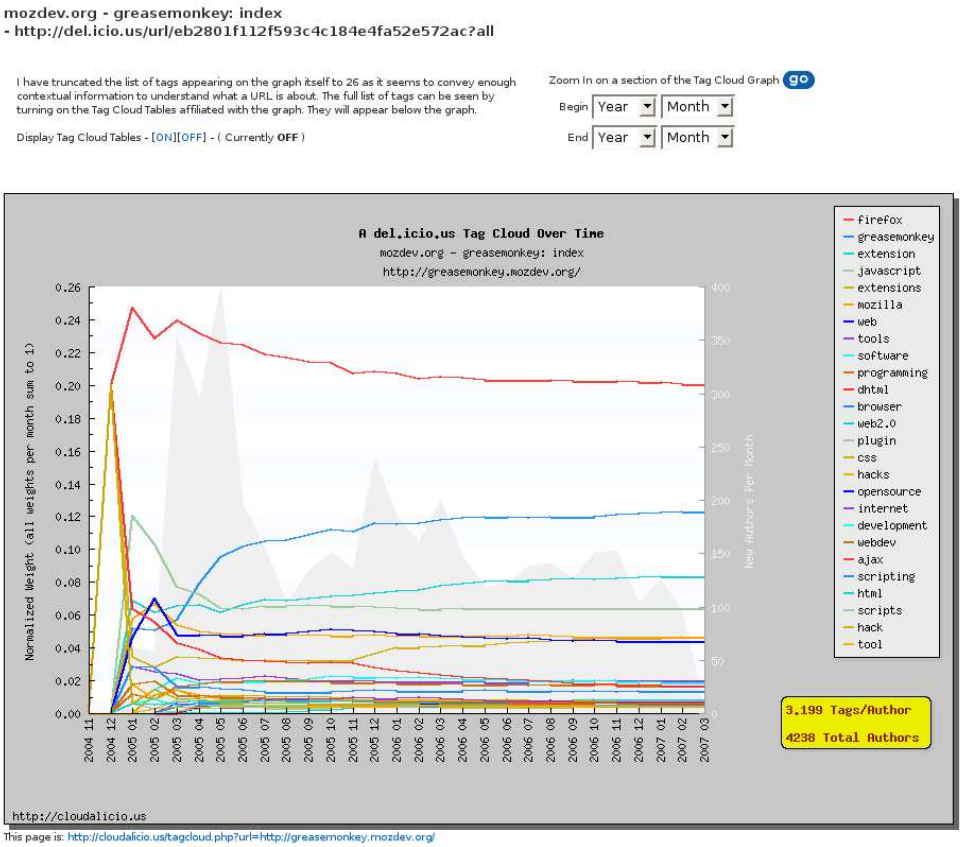

Figura 2.5: Il grafico mostra l'andamento nel tempo delle tag attribuite all'homepage del sito del progetto Greasemonkey. Si può osservare che la "power law" è rispettata: come avviene per la maggior parte dei siti, un ridotto numero ridotto di tag sono usate dalla grande maggioranza degli utenti, mentre ci sono moltissime tag il cui utilizzo è di un ordine di grandezza inferiore; nel grafico sono mostrate per altro solo le 26 tag più utilizzate: la "lunga coda" è stata tagliata a un certo punto.

Figura 2.6: La home page del

ceted navigation per costituzione: si basano sullo stesso tipo di approccio al problema della classificazione.

Una limitazione di questo esperimento è dovuta al fatto che il tool riconosce solo le tag che siano state esplicitamente associate a un determinato facet; c'è bisogno di un lavoro che sia svolto a priori per stabilire relazioni fra le tag e i facet. Tuttavia basandosi su vocabolari come Dublin Core e SKOS è possibile ottenere automaticamente una buona quantità di dati utili.

La limitazione più grossa del sistema a nostro avviso è probabimente un'altra: l'assenza di una gerarchia. I facet su cui si basa il sistema non sono articolati ma sono "piatti" e questo porta a diversi problemi nell'organizzazione delle tag. In particolare si pone un problema di scalabilità: come si può osservare dall'immagine 2.6, per alcuni facet sono disponibili centinaia di categorie, ovvero di tag associate, ma l'interfaccia per ovvi motivi ne mostra solo qualche decina per volta, ordinati per popolarità. Per vedere gli altri occorre scorrere spesso attraverso diverse schermate.

Gre.gario.us è un sito attualmente non attivo, di cui è disponibile online una demo[17]. L'idea di base è quella di trovare nuovi "amici" di un utente, cercando altre persone che abbiano salvato delle risorse in comune con lui.

Similicio.us[18] è un servizio Web che, dato un url, permette di trovare siti simili basandosi sulle tag attribuite dagli utenti di del.icio.us. Il sistema non è fruibile solo attraverso l'interfaccia Web: è stata sviluppata un'estensione per Firefox, grazie alla quale si può creare un bottone sul proprio browser, che permette di cercare siti simili a quello correntemente visualizzato.

Un'esperimento particolarmente interessante, infine, è quello denominato sid.vicio.us[19], che si propone di integrare del.icio.us con delle ontologie definite fra le tag. Il sistema si basa su un'ontologia OWL, permette a un utente di definire relazioni semantiche fra le proprie tag e propone un'interfaccia nella quale è possibile accedere ai dati del proprio account di del.icio.us ed effettuare query logiche sui propri bookmark.

[17] http://demos.semsym.com/gregarious/
[18] http://similicio.us/
[19] http://alteree.hardcore.lt/rdql/sidvicious2.php

Capitolo 3

Definizione dei requisiti

3.1 Obiettivi

L'obiettivo che ci siamo posti con questo lavoro è quello di arricchire le possibilità di navigazione nell'ambito di una folksonomia mediante tecniche semantiche.

Le principali folksonomie esistenti mostrano solo informazioni sulle tag e relazioni fra di esse basate su dati statistici, di frequenza e di cooccorrenza; per questo motivo esse presentano alcune limitazioni. Diversi dei problemi riportati in letteratura, legati alle possibilità di esplorazione e di ricerca in una folksonomia, sono in buona misura riconducibili alla mancanza di relazioni semantiche esplicite fra le parole chiave e in particolare all'assenza di gerarchia [Quintarelli, 2005]; le tag infatti costituiscono uno spazio di ricerca piatto, che non offre una visione organica e coerente del dominio. Per questo motivo non è sempre facile trovare un elemento specifico, è impossibile recuperare tutte le risorse legate a un certo ambito (perchè ogni concetto può trovarsi "sparso" fra più tag, sinonimi e modi diversi di definirlo, e l'esistenza di molte tag può essere ignorata o non considerata dall'utente), è difficile orientarsi nell'esplorazione.

Scopo del presente lavoro è mostrare come sia possibile superare almeno in parte queste limitazioni grazie all'utilizzo di un'ontologia come supporto per la navigazione.

Abbiamo ritenuto fondamentale non intervenire nello stadio della classificazione delle risorse da parte dei singoli utenti e lasciare invece che esso resti un processo essenzialmente spontaneo, in cui l'utente è libero di scegliere le

parole chiave che preferisce, senza alcun tipo di restrizione e senza alcuna richiesta di sforzi aggiuntivi. Questa scelta è fondamentale per preservare le caratteristiche di flessibilità, adattabilità e inclusività delle folksonomie. Non vogliamo neanche che sia richiesto all'utente nessuno sforzo aggiuntivo per definire una semantica delle tag o fra le tag che utilizza.

La nostra ipotesi di progetto è quella di introdurre la semantica a posteriori, sul materiale già categorizzato, mappando le tag utilizzate dagli utenti su un'ontologia già esistente.

Una volta che questa *mappatura semantica* sia stata realizzata, è possibile beneficiare di alcuni dei vantaggi offerti da un'ontologia, restando nel contesto di una folksonomia; in altre parole, accedere a informazioni costruite in modo libero e spontaneo dagli utenti, avendo a disposizione anche relazioni semantiche fra queste informazioni. I benefici potrebbero essere vari, non ultima la possibilità di effettuare *reasoning* sulle tag e sulle risorse; in questo lavoro tuttavia il nostro interesse è limitato al miglioramento delle possibilità di esplorazione e di ricerca.

In particolare vogliamo utilizzare la mappatura sull'ontologia per rendere possibile un'esplorazione organica di gruppi di tag correlate. Il risultato che ci aspettiamo è una struttura di navigazione *ibrida*, determinata sia da dati di correlazione sia da dati semantici.

3.2 Problematiche

Uno dei passaggi più delicati da affrontare è quello di attribuire un valore semantico a una tag utilizzata da un utente, trovando una corrispondenza fra tag ed elementi dell'ontologia; questo comporta alcune difficoltà.

Il primo problema è quello della polisemia [Golder and Huberman, 2005]: una stessa parola può avere più significati, e quindi una stessa tag può essere interpretata in diversi modi; al contrario gli elementi di un'ontologia devono essere privi di ambiguità. Per effettuare la mappatura occorre dunque risolvere le ambiguità e associare ad ogni parola chiave una precisa collocazione nell'ontologia. Pensiamo a una risorsa a cui sia associata la tag "turkey", parola che in inglese significa sia "Turchia" sia "tacchino": per mappare la parola chiave su un'ontologia occorre stabilire a quale dei due significati si riferisce la tag, data la risorsa. Nel caso si tratti del secondo significato, "tacchino", potrebbe essere opportuno anche distinguere se l'ambito sia quello della zoologia o della gastronomia.

Il secondo problema è quello della rigidità intrinseca di un'ontologia, a fronte della flessibilità e dell'apertura di una folksonomia: ogni sequenza di

caratteri può costituire una tag, e nuove tag possono continuamente essere create dagli utenti secondo qualsiasi criterio. Ci sono problemi legati alle diverse lingue usate, a neologismi, abbreviazioni, espressioni gergali, codici, errori di ortografia, uso di caratteri speciali. Appare chiaro come sia praticamente impossibile pensare di poter avere una mappatura completa delle tag su un'ontologia già esistente: alcune tag dovranno in ogni caso essere lasciate fuori.

Si pone poi il problema di stabilire in quale "spazio" eseguire la mappatura: se per l'intera folksonomia o per delle sue aree. La prima ipotesi è difficilmente praticabile, per la quantità di dati e di tag che ci si troverebbe a dover esaminare, anche in una folksonomia di medie o piccole dimensioni; essa porterebbe più facilmente a ricadere nei problemi tipici legati all'applicazione di un'ontologia in un contesto così vasto ed eterogeneo, descritti in [Shirky, 2005b], dovuti alla rigidità e alla monoliticità intrinseca di questo schema di classificazione.

La seconda ipotesi, quella di scegliere sottospazi della folksonomia, offre possibilità differenti a seconda del tipo di aree di azione scelte. Si possono considerare le tag usate da un utente, o dalla sua rete di amici, seguendo quindi un criterio "sociale", oppure ci si può basare su gruppi di tag correlate. Quest'ultima ipotesi è particolarmente interessante in quanto la correlazione fra tag è già sostanzialmente un fatto semantico, anche quando viene definita soltanto in base a dati empirici di cooccorrenza.

Per quanto riguarda più specificamente l'aspetto dell'interazione con l'utente, si pongono alcune problematiche fondamentali.

Occorre in primo luogo che le relazioni semantiche rappresentate abbiano un senso e un'utilità immediati per l'utente; per questo è particolarmente delicata la scelta dell'ontologia su cui basare la mappatura e delle relazioni semantiche da rappresentare.

Inoltre è necessario che le informazioni siano presentate all'utente in modo tale da offrire una reale possibilità di navigazione. Per questo sono possibili due alternative: realizzare una interfaccia di navigazione nuova a sé stante, oppure integrare le informazioni semantiche con l'interfaccia di un'applicazione esistente. Abbiamo scelto di concentrarci sulla seconda possibilità in quanto ci sembra decisamente più significativa l'opportunità di mostrare dei miglioramenti che possano essere ottenuti su un sistema maturo.

3.3 Requisiti

Per lo sviluppo dell'applicazione, abbiamo individuato i seguenti requisiti fondamentali di carattere generale che devono essere soddisfatti.

- **Portabilità**: il software deve essere nella misura possibile indipendente dalla piattaforma.
  Questa caratteristica è importante soprattutto riguardo agli utenti finali: deve essere garantita la possibilità di utilizzo del sistema alla fascia più ampia possibile di utenti del Web. In particolare l'applicazione deve:
  - essere fruibile con un browser, in modo indipendente dalla piattaforma del sistema dell'utente;
  - gravare il meno possibile sul sistema dell'utente, in termini di utilizzo di risorse computazionali;
  - non richiedere un elevato scambio di dati con l'esterno per il sistema dell'utente finale.
- **Modularità** e **trasparenza**: il software deve essere sviluppato in modo tale che le singole parti siano per quanto possibile indipendenti, possano essere facilmente modificate e integrate e offrano interfacce ben definite.
- **Scalabilità**: la capacità di lavorare bene anche con corpus vasti ed eterogenei è proprio uno dei punti di forza delle folksonomie; per questo è importante che l'applicazione sia in grado di elaborare efficacemente anche quantità consistenti di dati.
  In particolare, anche al crescere delle dimensioni dell'input, essa deve:
  - produrre risultati utili;
  - rispondere in tempi accettabili.

  Entrambe queste condizioni devono essere verificate con particolare attenzione alle esigenze dell'utente dell'interfaccia Web.
- **Usabilità**: l'applicazione deve migliorare effettivamente le possibilità di navigazione nella folksonomia. In particolare le relazioni semantiche aggiunte devono:
  - costituire informazioni immediatamente comprensibili e utili per l'utente;

- essere integrate in modo coerente con il sistema esistente e con la sua interfaccia di navigazione.

Il primo requisito si traduce fondamentalmente nella necessità di presentare una struttura gerarchica che:

- rappresenti concetti condivisi, che possano avere un senso per l'utente;
- sia compatta, ovvero mostri solo le informazioni essenziali;
- sia bilanciata, ovvero non sia né troppo piatta ne troppo profonda [Blei et al., 2004]. Ciascuna di queste due caratteristiche costituirebbe un impedimento per la navigazione; infatti una gerarchia molto profonda, che richieda di attraversare troppi livelli diventa un impedimento, mentre un eccessivo appiattimento dell'albero può portare a perdere i vantaggi di organizzazione delle informazioni offerti dalla gerarchia.

Poiché si tratta di un'applicazione il cui scopo ultimo è quello di migliorare le possibilità di navigazione per l'utente, i requisiti relativi all'interfaccia utente sono fondamentali; essi sono trattati in modo più specifico e dettagliato nell'apposito paragrafo 3.6.

3.4 Scelte ambientali

A partire dagli obiettivi che stanno alla base di questo lavoro, e dei requisiti formulati nella sezione precedente, verranno di seguito illustrate e discusse le principali scelte ambientali affrontate.

3.4.1 Struttura client-server

Dal punto di vista della struttura del sistema, per garantire la trasparenza e la portabilità è utile ricorrere a un paradigma client-server.

Al server competono tutti i compiti relativi al recupero, all'organizzazione e all'elaborazione delle informazioni; al client solo quelli di visualizzazione. In questo modo sono delegate al server tutte le operazioni più pesanti in termini di sforzo computazionale e occupazione di memoria. Il client invece può essere mantenuto il più leggero possibile: questa scelta è importante per garantire la massima portabilità rispetto all'utente finale e per rendere il servizio accessibile a un'ampia fascia di utenti.

L'uso di un formato standard per lo scambio di dati fra client e server è una condizione importante per garantire l'interoperabilità e per rendere

eventualmente fruibili anche secondo altre modalità le informazioni elaborate; in particolare è conveniente utilizzare il protocollo HTTP per gestire la comunicazione, mentre i risultati delle elaborazioni svolte dal server devono essere disponibili in un formato standard, come XML.

Per garantire la persistenza e la disponibilità delle informazioni raccolte e di alcune elaborazioni parziali in modo efficiente è opportuno ricorrere ad una base di dati.

3.4.2 Scelta dell'ontologia: WordNet

Avendo scelto di basarsi su un ontologia data, era possibile crearne una ad hoc oppure utilizzarne una esistente. La prima soluzione sarebbe stata praticabile, nell'ambito di questo lavoro di tesi, solo scegliendo un dominio ristretto, mentre a noi interessa l'esplorazione di vaste aree del Web.

Proprio per la necessità di coprire un dominio il più ampio e generale possibile, WordNet è particolarmente adatto al nostro scopo.

WordNet è un lessico semantico della lingua inglese, sviluppato a partire dal 1985 all'università di Princeton e rilasciato con una licenza libera. Oggi contiene oltre 150.000 parole, organizzate in quattro gerarchie distinte (nomi, verbi, aggettivi, avverbi) e diversi tipi di relazioni semantiche e lessicali fra queste.

WordNet rappresenta un'opportunità interessante per diversi motivi:

- contiene la maggior parte delle parole della lingua inglese;
- in particolare la tassonomia basata sulla relazione di iperonimia-iponimia comprende la maggior parte dei sostantivi della lingua inglese organizzati in un'unica gerarchia, e anche un certo numero di nomi propri, seppure ancora in quantità ridotta (parole come Goofy, Bob Dylan, Milano, Linux, sono riconosciute da WordNet e sono integrate nel resto dell'ontologia generalmente mediante relazioni di *instanceof*, ovvero sono considerate istanze di qualche concetto: Bob Dylan è un'istanza di cantante, che è iponimo di artista e di persona, Linux è una istanza di sistema operativo e così via);
- è stato sviluppato secondo principi di psicolinguistica, per cui le relazioni semantiche, e in particolare quella di iperonimia/iponimia, sono state strutturate con una attenzione particolare al modo in cui la mente umana interpreta le parole e il linguaggio, in accordo con i risultati più importanti ottenuti nel settore [Fellbaum, 1998];

- è disponibile con licenza libera ed è accessibile attraverso interfacce API con vari linguaggi di programmazione;
- la possibilità di integrazione di altre wordnet in lingue differenti offre una prospettiva molto interessante. Ci sono diversi esperimenti che vanno in questa direzione[1], uno particolarmente interessante ha come base la lingua italiana [Pianta et al., 2002].

Se i primi due punti sono essenziali per garantire l'applicabilità dell'algoritmo su larga scala e su insiemi eterogenei di dati, organizzando una grande quantità di parole in un'unica ontologia, fondamentale è anche il terzo, che porta nella direzione della massima generalità e validità dei risultati. Abbiamo discusso nel paragrafo 2.3.1 come proprio la rigidità e la "pretesa" di dare un'unica visione monolitica e immutabile delle cose rendano le tassonomie particolarmente inadatte in un contesto eterogeneo e dinamico come il Web [Shirky, 2005b]; diversamente da molte ontologie, legate per costituzione a contesti o a esigenze specifiche e soggette a molti bias, WordNet costituisce un tentativo di rappresentare categorie di validità generale, che rispecchino le strutture fondamentali del linguaggio. Anche la comprensibilità immediata e intuitiva delle relazioni semantiche rappresentate è perseguita, attraverso il tentativo di cogliere l'essenza degli schemi secondo cui la mente umana classifica gli oggetti e i concetti, e di distinguere le associazioni mentali principali da quelle secondarie.

A fronte di queste caratteristiche, la scelta di WordNet comporta alcuni svantaggi, principalmente i seguenti due:

- l'ampiezza e l'accuratezza del lessico, che contiene anche parole poco comuni e distingue fra diversi significati e sfumature di una stessa parola, rappresenta certamente una ricchezza, ma può costituire anche un problema: la granularità troppo sottile infatti può in molti casi appesantire inutilmente l'esplorazione dell'ontologia per un utente umano; in particolare, spesso la profondità dell'albero risulta eccessiva: per esempio per arrivare dalla radice dei sostantivi ("entity") alla parola "cat", nel significato dell'animale domestico, bisogna scendere 11 livelli nella gerarchia;
- WordNet è molto valido per la sua ampiezza, ma non ha una copertura completa di domini specifici.

[1] The Global WordNet Association è un'associazione che si occupa di coordinare e supportare questo tipo di progetti in tutto il mondo e di proporre degli standard condivisi. Essa possiede un sito Web: http://www.globalwordnet.org/

Per quanto riguarda il primo punto, l'utilizzo di WordNet richiede di adottare delle strategie per ridurre il livello di granularità [Mihalcea and Moldovan, 2001], rendendo più leggera l'esplorazione.

Per quanto riguarda il secondo, può essere opportuno integrare altre ontologie relative ad ambiti specifici.

Devono poi essere tenute in considerazione alcune limitazioni che la scelta dell'ontologia basata sulla relazione di iperonimia-iponimia di WordNet impone:

- considerare solo le tag che corrispondono a sostantivi della lingua inglese (compresi alcuni nomi propri);
- trascurare le relazioni semantiche di tipo diverso dalla iperonimia/iponimia.

Per quanto riguarda il primo punto si può osservare, da una parte, che le tag non corrispondenti ad alcuna parola della lingua inglese sarebbero difficilmente riconoscibili, anche utilizzando una qualsiasi altra ontologia, almeno se restringiamo il campo alle ontologie già esistenti a priori; dall'altra parte che la grande maggioranza delle tag che possono essere riconosciute come parole sono sostantivi. Uno studio più completo di questo problema, corredato da dati specifici, è affrontato nella sezione 4.3.1.

Per quanto riguarda il secondo punto, esso è una scelta obbligata nel contesto di questo lavoro. Lo stesso WordNet contiene molte altre relazioni, sia semantiche, come la meronimia, sia lessicali, come l'antonimia. Ciascuna di queste relazioni però è definita solo per un numero ristretto di synset o di parole, e nessuna è utilizzabile su larga scala. La relazione di iperonimia/iponimia al contrario comprende in modo coerente tutti i sostantivi e ha proprio quelle caratteristiche di organicità e di gerarchia che possono essere più utili per completare una folksonomia.

L'algoritmo Castanet: uso di WordNet per creare una gerarchia di facet

Un'esperienza particolarmente interessante, riguardo la possibilità di utilizzo di WordNet nell'ambito della classificazione di risorse, viene sviluppata da qualche anno all'Università di Berkley, nell'ambito del progetto Flamenco per la creazione di interfacce di faceted navigation [2].

Castanet in particolare è l'algoritmo che viene utilizzato per creare una gerarchia di facet in modo semiautomatico basandosi su WordNet [Stoica et al., 2006].

[2] L'homepage del progetto è http://flamenco.berkeley.edu/

Esso lavora su oggetti appartenenti a un ambito definito e associati a brevi descrizioni testuali, come ricette o titoli di articoli biomedici. A partire dalle descrizioni testuali sceglie delle parole rappresentative delle risorse e costruisce un albero semantico contenente queste parole, basato sulla gerarchia della relazione di iperonimia di WordNet.

L'algoritmo è semiautomatico, perché prevede un passaggio di supervisione umana per scegliere quali rami dell'albero creato dall'algoritmo siano adatti ad essere dei facet e per effettuare eventuali aggiustamenti.

I problemi principali riscontrati, indicati dagli autori dell'algoritmo, sono la polisemia e la granularità sottile di WordNet. Per risolvere il primo problema l'algoritmo esegue una disambiguazione basata sui cammini di iperonimia e sulla conoscenza del dominio comune di appartenenza degli oggetti; il problema della granularità viene affrontato eseguendo una compressione dell'albero e rimuovendo le categorie di alto livello.

L'algoritmo mostra di funzionare bene e i risultati sono incoraggianti. Le interfacce di navigazione ottenute con Castanet in una serie di contesti offrono buone caratteristiche di usabilità [Stoica et al., 2006].

3.4.3 Scelta del linguaggio di programmazione: Perl

Come linguaggio di programmazione per realizzare il server abbiamo scelto Perl. Esso è un linguaggio interpretato, molto diffuso per la realizzazione di applicazioni Web, e offre alcune caratteristiche che lo rendono particolarmente adatto per le nostre esigenze:

- è multipiattaforma;
- offre un notevole supporto alla manipolazione di stringhe e al *pattern matching*;
- offre la disponibilità di librerie specializzate di supporto per esplorare e processare pagine Web;
- offre librerie per l'interfaccia con WordNet;
- è software libero.

Per quanto riguarda il secondo e il terzo punto, che rappresentano due aspetti fondamentali per la realizzazione del server, e in particolare del crawler, Perl presenta caratteristiche particolarmente avanzate.

Per quanto riguarda l'integrazione con WordNet, oltre alle librerie per l'accesso al database del lessico, fornite anche da altri linguaggi, ne sono

state sviluppate alcune che mettono a disposizione strumenti, algoritmi e metriche per elaborare i dati e le relazioni fra le parole.

Uno svantaggio di Perl è quello delle prestazioni in termini di efficienza, leggermente inferiori a quelle offerte da altri linguaggi compilati.

3.4.4 Navigazione attiva: Firefox e Greasemonkey

Sul lato client, abbiamo evidenziato la necessità di avere un'applicazione più leggera possibile e di garantire la massima portabilità. Allo stesso tempo, abbiamo posto il problema dell'integrazione dei nuovi contenuti elaborati con quelli normalmente forniti tramite l'interfaccia Web da una folksonomia esistente.

Una soluzione pratica a questo problema è quella di eseguire uno script locale nel browser, per modificare dinamicamente le pagine Web visualizzate, aggiungendo le informazioni elaborate dal nuovo server.

JavaScript è il linguaggio di scripting più comunemente utilizzato per modificare dinamicamente i contenuti visualizzati in una pagina Web e costituisce uno standard largamente affermato. Gli script però sono generalmente associati a una pagina Web, inseriti nel codice HTML o in un file indicato nella pagina da chi l'ha creata; di norma non è possibile, per un utente, eseguire un proprio script su una pagina visitata.

Per rendere questo possibile ci sono fondamentalmente due ipotesi: Opera è l'unico browser che supporta direttamente la possibilità di eseguire uno script JavaScript locale su una pagina Web visitata; in alternativa sono disponibili delle estensioni per altri browser che permettono di ottenere gli stessi risultati.

Firefox è un browser open source, multipiattaforma, conforme agli standard del Web, che sta conoscendo una rapida diffusione, avendo conquistato in pochi anni una fetta di mercato superiore al 12% del mercato, come mostrato nel grafico 3.1. In particolare, Firefox sta conoscendo una buona affermazione nell'ambito del Web2.0, grazie alle caratteristiche di portabilità, flessibilità ed estensibilità.

Greasemonkey è un'estensione di Firefox che, una volta installata nel browser, permette di eseguire script locali per modificare dinamicamente le pagine Web visitate. Greasemonkey è disponibile anche per altri browser; in particolare associato a Firefox costituisce la soluzione più diffusa e meglio supportata per eseguire script locali.

Il fatto di eseguire uno script locale, in aggiunta a quelli eventualmente definiti nella pagina Web, pone alcuni problemi e alcune restrizioni tecniche,

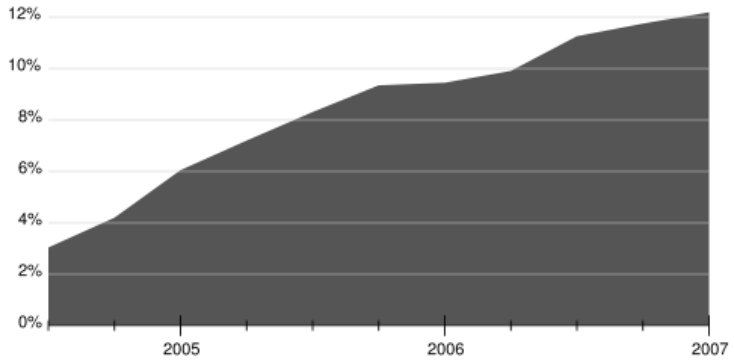

Figura 3.1: Il grafico è stato pubblicato all'indirizzo http://en.wikipedia.org/wiki/Image:Mozilla-firefox-usage-data.svg e mostra l'andamento dello share *di uso di Firefox negli ultimi anni. I dati sono ottenuti calcolando la media da tre fonti globali: TheCounter (http://www.thecounter.com/), OneStat (http://www.onestat.com), and NetApplications (http://www.netapplications.com/)*

relative alle variabili globali e alle funzioni che è possibile utilizzare; tuttavia Greasemonkey fornisce alcune funzioni speciali per aggirare questi problemi e con uno script locale si può in definitiva realizzare pressoché tutto ciò che può essere ottenuto con uno script definito sul lato del server.

L'utilizzo di Greasemonkey è particolarmente interessante per il principio su cui si basa, che è quello della *navigazione attiva*. Nel Web attuale l'utente tende spesso a essere relegato a un ruolo passivo, costretto all'interno delle strutture navigazionali spesso rigide create e imposte da chi ha progettato un'applicazione Web; Greasemonkey valorizza la centralità dell'utente come soggetto attivo, dandogli la possibilità di scegliere e di definire anche i propri modi di accedere alle informazioni sul Web e di visualizzarle.

La direzione di fondo è quella del Web2.0, che valorizza il ruolo attivo dell'utente; in particolare un aspetto fondamentale del progetto consiste nella possibilità di condividere il codice: un repository raccoglie tutti gli script degli utenti che abbiano deciso di pubblicare il codice JavaScript da loro realizzato. Gli script disponibili attualmente sono oltre 5000 e sono organizzati mediante tag attribuite dall'autore, in una folksonomia narrow[3].

[3]Il repository degli script per Greasemonkey è accessibile all'indirizzo: http://userscripts.org/

3.4.5 Scelta della folksonomia: del.icio.us

La prima scelta relativa alla folksonomia riguarda il contesto. In questo lavoro, come abbiamo evidenziato più volte, ci interessa confrontarci con lo spazio più generale del Web; il tipo di risorsa più generale e più diffuso come oggetto di una folksonomia sono i link, i bookmark. Un'applicazione di social bookmarking, che non si collochi in un dominio specializzato, rappresenta senza dubbio il contesto più ampio e generale; le applicazioni di questo tipo inoltre, al contrario di altre basate sulla classificazione di risorse più specifiche, sono folksonomie broad e offrono una dinamica molto più ricca, come osservato nel paragrafo 2.3.3 [Vander Wal, 2005].

La scelta più naturale, dovendosi integrare con un sistema di social bookmarking esistente, avrebbe potuto essere quella di un'applicazione *open source*, come Scuttle o de.lirio.us. Questo aspetto tuttavia non costituisce un requisito indispensabile: il nostro obiettivo non è quello di intervenire modificando direttamente un'applicazione esistente. Nell'ambito del Web2.0, come mostrato nel paragrafo 2.2, ci sono molti modi per integrare fra loro applicazioni diverse, anche senza bisogno di condividere il codice.

Nel nostro caso non è tanto cruciale la disponibilità del codice sorgente dell'applicazione, e neanche delle API, che generalmente permettono la gestione dell'account di un singolo utente, ma in primo luogo la disponibilità dei dati: il nostro sistema infatti ha bisogno di una quantità di informazioni consistente come base per le proprie elaborazioni. Il secondo aspetto importante riguarda l'interfaccia di navigazione, per l'integrazione con il sistema delle nuove funzionalità che vogliamo rendere disponibili.

Dal punto di vista della reperibilità dei dati, i sistemi di social bookmarking esistenti possono essere considerati equivalenti, in quanto per ottenere una quantità interessante di informazioni non sono in ogni caso sufficienti i meccanismi standard come API o feed RSS, ma è necessario scandire le pagine HTML. Alcuni sistemi non consentono un'esplorazione del tutto completa della folksonomia neanche tramite l'interfaccia Web, tuttavia questo tipo di restrizione riguarda solo casi di grandi moli di dati e non costituisce per noi una limitazione.

Accanto alla possibilità di ottenere i dati dal sistema è fondamentale anche la quantità e qualità dei dati disponibili. Alcuni sistemi contengono per lo più tag in lingue diverse dall'inglese; altri, come de.lirio.us, mostrano segni evidenti di massicci interventi di *gaming*: operare su quelle tag e su risorse che sono in buona parte spam sarebbe poco significativo; altri, come Scuttle, presentano una quantità modesta di dati. L'interesse principale di

un sistema open source come Scuttle risiede nel fatto che un programma in grado di interagire col server potrebbe integrarsi anche con tutti gli altri sistemi basati sullo stesso software, senza nessun costo aggiuntivo. Per questo motivo questo sistema è stato preso in considerazione molto seriamente, però la mancanza di una quantità rilevante di dati, sia sul server centrale sia su altri che utilizzano lo stesso software, ci ha portati a scartare questa ipotesi.

Per quantità di dati spicca, fra tutti i sistemi di social bookmarking, del.icio.us: avendo superato in febbraio 2007 il milione e mezzo di utenti e continuando a crescere a ritmi sempre più rapidi, esso ha una dimensione di qualche ordine di grandezza superiore a tutti i sistemi concorrenti. L'alto numero di utenti è la migliore garanzia per la significatività dei dati: esso comporta un maggiore numero dei punti di vista su singole risorse, una maggiore varietà nell'uso delle tag, una maggiore quantità di risorse classificate, una maggiore difficoltà di inquinare sensibilmente il sistema con tecniche di gaming. Specularmente, l'eventuale quantità eccessiva di dati, per esempio relativi a un determinato url o a una determinata tag, non costituisce in ogni caso un problema, in quanto è sempre possibile restringere il campo secondo un criterio opportuno (scegliendo ad esempio i dati più recenti).

A questi vantaggi di del.icio.us si aggiungono quelli dovuti al fatto di essere un sistema maturo, in cui trovano applicazione la maggior parte delle caratteristiche presenti in altre folksonomie concorrenti, e intorno al quale viene sviluppata anche una varietà di strumenti esterni che lo arricchiscono, lo integrano, e ne interpretano i dati. In particolare l'interfaccia Web è particolarmente usabile e intuitiva; essa costituisce in qualche modo uno standard per le folksonomie di social bookmarking.

Per tutti questi motivi abbiamo scelto del.icio.us come folksonomia di base per questo lavoro.

3.5 Analisi della struttura navigazionale di del.icio.us

La figura mostra lo schema logico della struttura navigazionale del sito del.icio.us, realizzato nella forma di un *interactive dialogue model* (IDM) [Bolchini and Paolini, 2006]. Questo tipo di schema è stato studiato per supportare il design di un'interfaccia di navigazione e si basa sulla metafora di un dialogo fra l'utente e l'applicazione. Esso permette di descrivere un'applicazione Web con vari livelli di astrazione, concettuale, logico o di presentazione.

Noi ci siamo serviti in particolare dello schema IDM logico, che fornisce una notazione compatta e intuitiva per descrivere l'organizzazione dei conte-

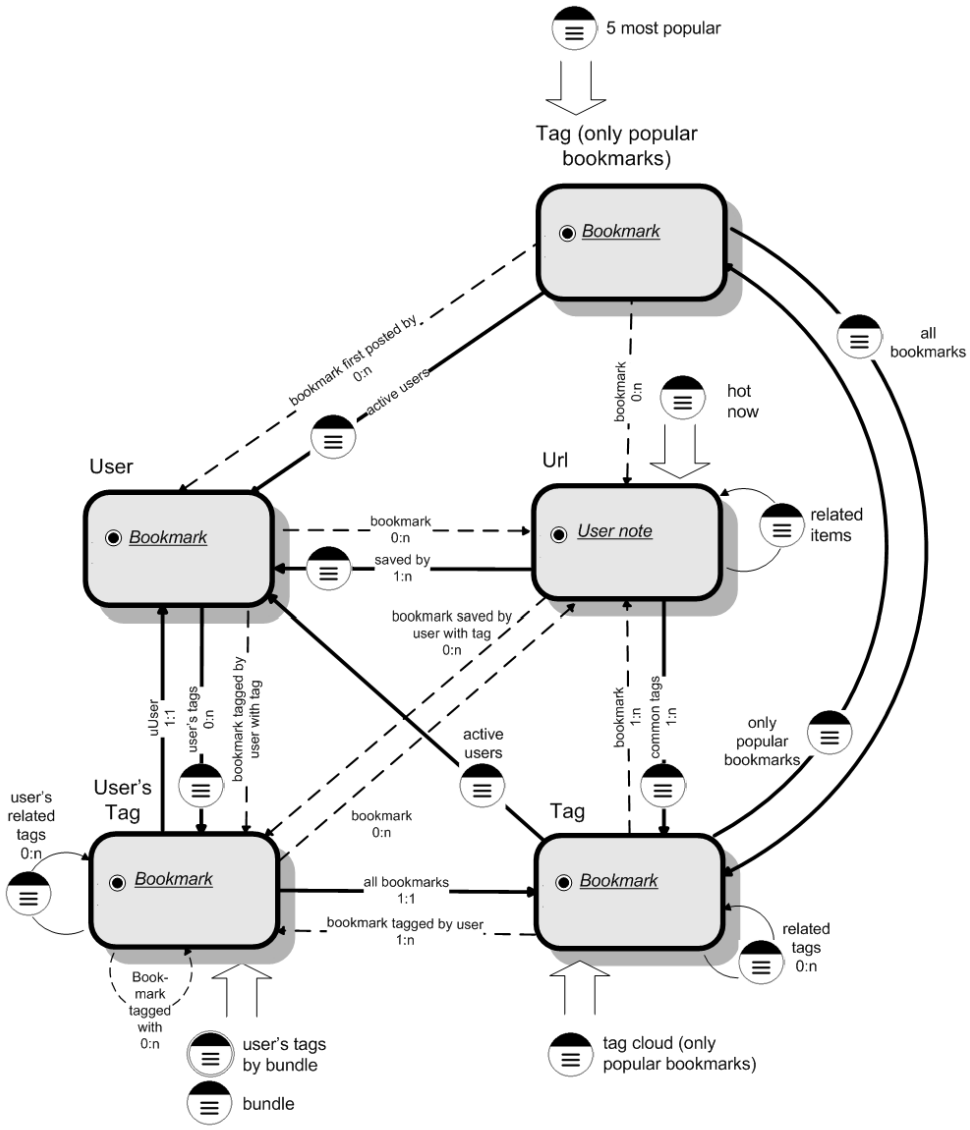

Figura 3.2: lo schema IDM logico di navigazione in del.icio.us

nuti, la struttura logica di un sito e delle relazioni fra le pagine, le strategie di navigazione che sono rese possibili. Nel caso di un sito fortemente interconnesso come del.icio.us questo tipo di notazione è di grande aiuto per descriverne aspetti altrimenti difficilmente rappresentabili.

I rettangoli sono i *kind of topic*, unità di contenuto fondamentali. In del.icio.us essi corrispondono bene ai "tipi di url": tutte le pagine con url del tipo http://del.icio.us/tag/«tag» rappresentano delle istanze del kind of topic "tag", quelle del tipo http://del.icio.us/«utente» rappresentano delle istanze di "user" e così via. Ogni istanza di kind of topic non si traduce in una singola pagina, ma in una collezione di oggetti presentati in più pagine; la strategia di navigazione fra le pagine di una stessa collezione, in del.icio.us, è gestita con un meccanismo di tipo sequenziale: sono mostrati solo gli oggetti più recenti ed è possibile visualizzare gli altri solo scorrendo le pagine precedenti, ad una ad una; specificando il parametro "page" nella richiesta HTTP è però possibile accedere direttamente a una pagina, in qualsiasi posizione.

Nello schema abbiamo raffigurato la struttura essenziale del sito; fra le cose che abbiamo tralasciato di rappresentare c'è un altro kind of topic, che raccoglie i link più recenti per ogni tag: esso è analogo a quello che raccoglie i bookmark più popolari. Non abbiamo rappresentato le caratteristiche di network, disponibili per gli utenti che abbiano effettuato il log in nell'applicazione.

Le unità di contenuto, all'interno dei topic, sono i *dialog act*. Il grafico mostra una caratteristica particolare, che dipende dalla struttura del sito di del.icio.us, in cui i link sono più importanti dei contenuti: ogni kind of topic comprende un solo tipo di contenuto. Questo coincide quasi sempre con un elenco di bookmark, tranne nel caso del kind of topic "url", che contiene un elenco di commenti da parte degli utenti.

Le pagine sono fortemente interconnesse, mentre sono pochi i *punti di accesso* alle informazioni (mostrati cone delle frecce larghe):

- i bookmark più popolari del momento, una *hotlist* accessibile dalla homepage;
- le 5 tag più popolari del momento (link dalla homepage alle pagine di bookmark popolari per quelle tag);
- accessibile dall'homepage la *tag cloud* delle tag più popolari di del.icio.us (con link alle pagine delle tag);

- tutte le tag di un utente, organizzate in *bundle*.

La scarsità dei punti di accesso alle informazioni è compensata in buona parte dalla possibilità di accedere alle pagine del sito costruendo gli url manualmente e dall'alto livello di interconnessione fra le pagine.

Per quando riguarda le *relazioni rilevanti* (o *change of subject*), che rendono possibile il passaggio da un kind of topic a un altro, le abbiamo distinte in due tipi: "strutturate" o "casuali".

Le relazioni che abbiamo definito strutturate sono quelle che legano uno o più kind of topic ad uno di partenza, in modo stabile. Essi generalmente corrispondono a un link o a un elenco di link, accessibili da una zona precisa di una pagina. Per esempio, la lista degli "active users" è presente nella barra laterale in ogni pagina relativa a una tag e mostra gli utenti che utilizzano di più quella tag, indipendentemente dai particolari bookmark che la pagina contiene.

Le altre relazioni sono "casuali", nel senso che sono legate ai particolari elementi visualizzati nella pagina: per esempio, dalla pagina relativa a una tag, la relazione diretta da un sito all'utente che lo ha salvato è visibile solo per i siti presenti nella pagina visualizzata, ovvero, per esempio, solo per le ultime 10 risorse etichettate con quella tag.

Con frecce piene abbiamo raffigurato le relazioni strutturate, con frecce tratteggiate le relazioni "casuali". Quelle che ci interessano di più in questa analisi sono quelle del primo tipo, che forniscono strumenti più organici di esplorazione.

Come mostrato in figura 3.3 ci sono strumenti che permettono di esplorare le tag di un utente in modo organico: in primo luogo i *bundle* permettono ad un utente di raggruppare le proprie tag secondo criteri personali, una sorta di "clustering semantico" che l'utente stesso può effettuare manualmente. Grazie ai bundle si può avere subito una visione d'insieme delle aree semantiche di interesse di un utente. Questo è uno strumento importante, anche se attualmente non molti utenti di del.icio.us lo utilizzano.

In secondo luogo la relazione *related tags* nell'ambito di un particolare utente mostra tutte le tag correlate, ovvero tutte quelle che quell'utente ha utilizzato per almeno un sito insieme alla tag corrente, ordinate per numero di cooccorrenze con questa.

Nello spazio delle tag di tutti gli utenti insieme invece le possibilità di navigazione ed esplorazione sono decisamente più ristrette.

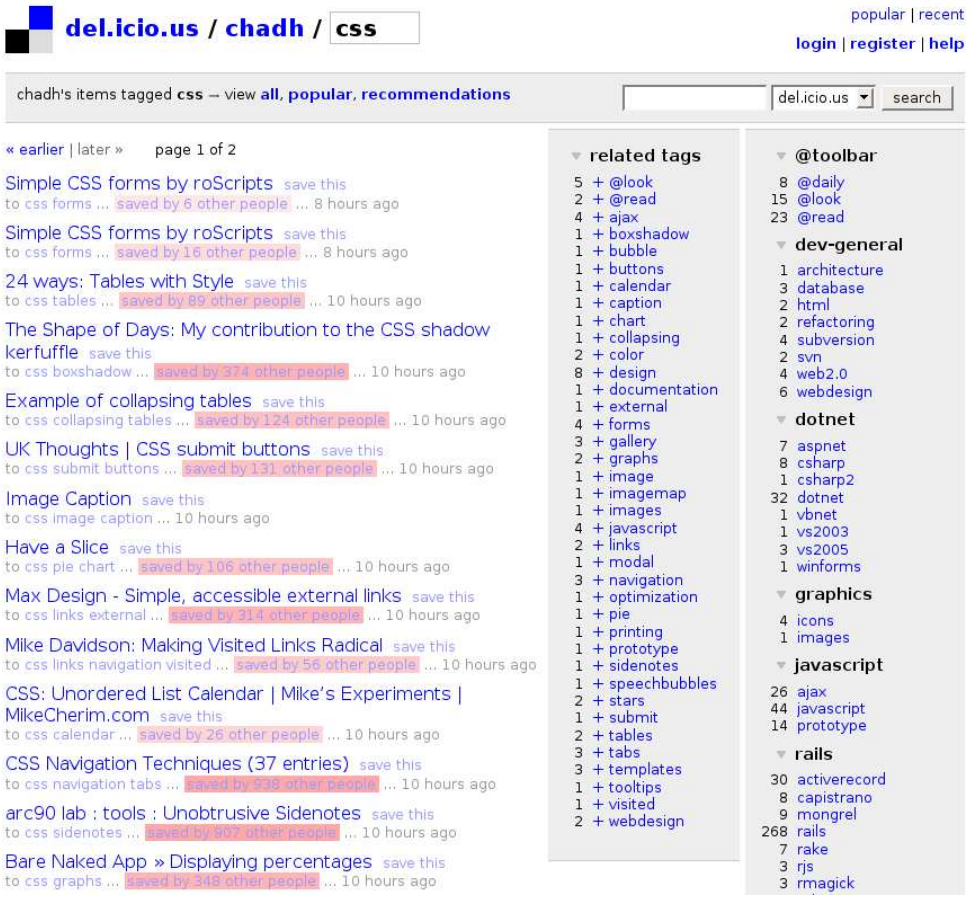

Figura 3.3: la figura mostra la pagina di del.icio.us di un utente relativa alla tag "css". Essa permette di osservare, nella barra laterale più esterna, i bundle *creati dall'utente per organizzare tutte le sue tag in modo coerente. La barra laterale più interna mostra invece le tag correlate a "css"; si osservi che sono mostrate tutte le tag correlate, nello spazio delle risorse dell'utente, e per ognuna di esse è indicato il numero di risorse che costituiscono l'intersezione con la parola chiave corrente.*

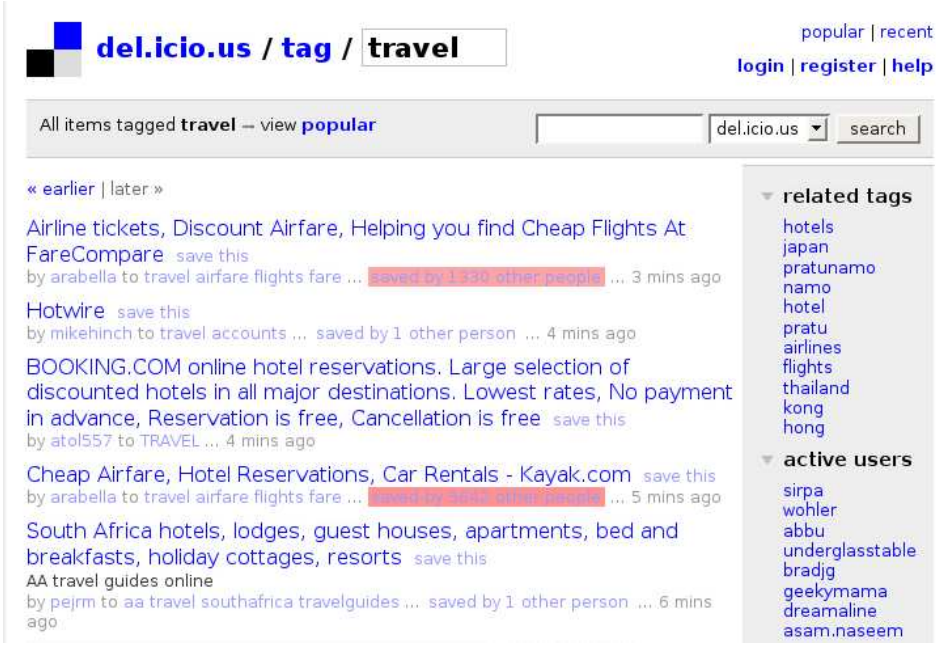

Figura 3.4: la figura mostra la pagina di del.icio.us relativa alla tag "travel", con la barra laterale delle tag correlate.

Non esistono bundle e l'unico punto di accesso all'insieme delle tag è la pagina della *tag cloud*, che mostra solo le parole chiave più popolari: decisamente poche, rispetto al numero di tag, anche molto popolari, presenti in del.icio.us.

Le uniche relazioni strutturate sono *related tags* e *active users*; entrambe non sono presenti nelle pagine di tutte le tag, ma solo di quelle che superano un certo livello di popolarità. Esse mostrano i primi 20 utenti più attivi, relativamente a quella parola chiave, e le 11 tag maggiormente correlate.

La relazione *related tags* presenta molti limiti; potenzialmente molto utile per permettere l'esplorazione della folksonomia, in molti casi non si rivela tale nei fatti. Spesso fra le related tags appaiono tag generiche molto usate e onnipresenti, come "blog", "nyc" (che sta per "New York City") o "toread" (che indica genericamente cose "da leggere"), che pur non avendo magari nessun nesso particolare con la tag in oggetto risultano fra le prime per numero di cooccorrenze. É chiaro però che se dalla pagina di una parola chiave non molto popolare si segue il link alla pagina della tag "blog", "nyc" o "toread", difficilmente si troveranno in queste pagine delle risorse correlate alla tag di partenza.

La scelta di lasciare povera e minimale la navigazione fra le tag correlate è determinata probabilmente in buona parte da motivi di efficienza, essen-

doci grandi moli di dati da elaborare, ma anche dalla difficoltà di mostrare una lunga lista di tag senza organizzarle in qualche modo, in assenza di qualsiasi forma di gerarchia. A nostro avviso questa scelta, che è comune alle principali folksonomie esistenti, ne costituisce uno dei limiti maggiori, in quanto riduce le possibilità di esplorazione di una relazione potenzialmente di grande interesse.

Fra le principali folksonomie esistenti fa eccezione Flickr, che ha trovato nel clustering una possibile soluzione a questo problema. Si può osservare che in una folksonomia narrow il numero medio di tag assegnate a ogni risorsa è molto inferiore rispetto a quello di una folksonomia broad; questo fattore ha sicuramente favorito Flickr, rendendo computazionalmente molto più leggera l'implementazione della funzione.

Flickr, come osservato nel paragrafo 2.4.1, si basa unicamente su calcoli di dati di correlazione per effettuare il clustering. Questa è forse la soluzione più tipica e più naturale, nell'ambito di una folksonomia, per arricchire le possibilità di navigazione fra le tag e fra tag correlate; essa si basa sull'assunzione che la semantica possa essere interamente ricavata dai dati del sistema e può portare risultati utili, ma incorre nelle limitazioni e nei problemi che sono legati intrinsecamente alla mancanza di una semantica esplicita.

3.6 Requisiti dell'interfaccia utente

L'analisi delle possibilità di navigazione offerte dall'interfaccia Web di del.icio.us ha evidenziato come la relazione delle tag correlate ne costituisca un punto debole. Potenzialmente molto interessante per l'esplorazione della folksonomia, questa relazione si rivela spesso, di fatto, di scarsa utilità nella sua forma attuale.

Per questo motivo abbiamo scelto di intervenire su questa relazione per arricchirla con informazioni semantiche esplicite, nei modi descritti precedentemente in questo capitolo. La mappatura semantica sull'ontologia sarà effettuata, volta per volta, per insiemi di tag correlate a una parola chiave data.

Per definire i requisiti relativi agli aspetti di interfaccia con l'utente, ci siamo serviti di AWARE (Analysis of Web Application Requirements), un metodo orientato agli obiettivi per l'analisi dei requisiti nell'ambito delle applicazioni Web [Bolchini and Mainetti, 2004]. AWARE permette di partire da obiettivi di alto livello e procedere per passi successivi di raffinamento fino a formulare dei requisiti specifici per l'interfaccia.

Figura 3.5: il diagramma AWARE per la definizione dei requisiti specifici dell'interfaccia utente dell'applicazione.

L'unica categoria di *stakeholder* che abbiamo deciso di rappresentare è quella generale degli utenti, non essendoci particolari distinzioni fra tipologie di utenti degne di essere rappresentate, e non essendoci altri attori importanti sulla scena (per definizione si tratta di un sistema costruito dagli utenti, per gli utenti).

La necessità di una gerarchia per supportare l'esplorazione è un'assunzione, discussa in precedenza, che costituisce il punto di partenza di questo lavoro; a partire da questa sono definiti i requisiti più specifici dell'interfaccia, elaborati per soddisfare le esigenze dell'utente e per offrire le diverse possibilità di interazione che possono essere necessarie.

Ecco l'elenco dei requisiti specifici, raccolti secondo gli aspetti di design che riguardano.

Requisiti di **contenuto**; devono essere presentate le seguenti informazioni:

- elenco completo delle tag correlate;
- numero di elementi corrispondenti ad ogni tag correlata;
- definizione di ogni concetto della gerarchia.

Requisiti di **accesso** ai contenuti:

- accesso alle tag correlate mediante una gerarchia semantica.

Requisiti di **navigazione**:

- link alle pagine delle tag correlate;
- link alle pagine delle intersezioni della tag oggetto con le tag correlate.

Requisiti di **presentazione**:

- i rami più interessanti, maggiormente correlati, mostrati in posizione privilegiata.

Requisiti di **interazione**:

- possibilità di espandere e nascondere aree della gerarchia.

Questi requisiti non sono importanti solo per lo sviluppo dell'interfaccia vera e propria con l'utente finale, ma in tutti i passi che portano alla sua realizzazione; i requisiti di contenuto, in particolare, determinano i dati che devono essere presentati nell'albero e dunque anche la scelta delle informazioni da raccogliere e organizzare e delle strutture dati da utilizzare, e la definizione degli algoritmi che devono essere realizzati.

Capitolo 4

Progetto e realizzazione

Dopo aver definito l'obiettivo e i requisiti e delineato le principali scelte ambientali e progettuali, in questo capitolo saranno illustrati tutti gli aspetti relativi al progetto e alla realizzazione dell'applicazione e delle sue singole parti.

Partiremo nel prossimo paragrafo dalla definizione della struttura e dell'architettura di alto livello del sistema, per evidenziare poi le interazioni fra i vari moduli che lo costituiscono e le interfacce tramite cui comunicano. In seguito sono trattate nello specifico le questioni principali affrontate: quelle più generali poste dall'uso di WordNet nel paragrafo 4.3, quelle relative al reperimento dei dati dalla folksonomia e alla realizzazione del crawler nel paragrafo 4.4, quelle legate alla risoluzione del problema della polisemia delle parole nel paragrafo 4.5, quelle relative alla costruzione dell'albero semantico delle tag nel paragrafo 4.6; l'ultimo paragrafo è dedicato invece alla realizzazione dello script per Greasemonkey e dell'interfaccia con l'utente finale del sistema.

4.1 Architettura del sistema

La struttura del sistema è basata su un paradigma client-server. Abbiamo realizzato un nuovo server Web, che estrae dei dati da del.icio.us e li elabora per fornire informazioni aggiuntive: data una tag, ricevuta come parametro, costruisce l'albero semantico delle tag correlate, basato sulla gerarchia degli iperonimi di WordNet. Sul lato client abbiamo creato uno script JavaScript da eseguire nel browser dell'utente per permettere l'integrazione dei dati aggiuntivi nelle pagine Web ottenute da del.icio.us.

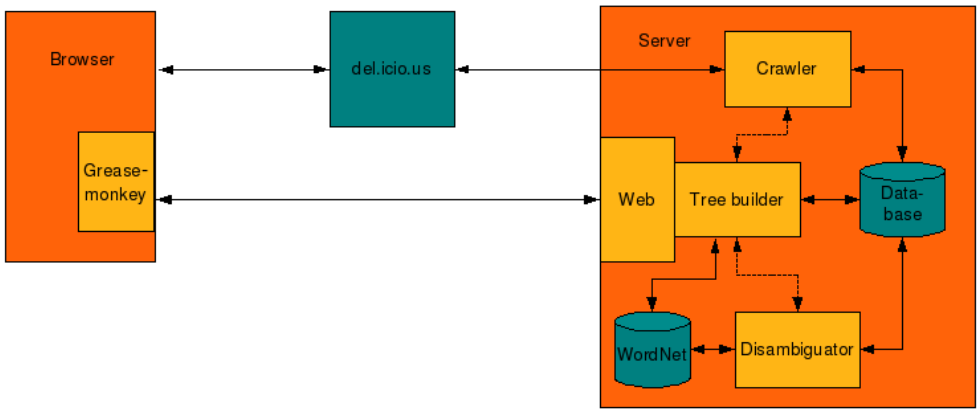

Figura 4.1: lo schema mostra l'architettura generale del sistema.

Il server è composto da tre moduli indipendenti, che svolgono tre funzioni differenti; essi comunicano sostanzialmente tramite il database:

- il **crawler** estrae i dati dal sito di del.icio.us e li salva sul database;
- il **disambiguator** elabora i dati ottenuti dal crawler e arricchisce il database con i valori semantici delle tag;
- il **Web server** utilizza i dati per costruire l'albero semantico delle tag.

In alcuni casi il Web server può attivare gli altri due moduli in seguito alla richiesta di un utente, quando nel database non siano già disponibili i dati necessari a soddisfare la richiesta.

Il Web server costituisce l'interfaccia con l'esterno; esso comunica con il client mediante il protocollo HTTP. Nelle richieste deve essere specificata una tag; in risposta il server restituisce l'albero semantico delle tag correlate a quella data, in formato XML o HTML.

Oltre alla tag, nella richiesta HTTP possono essere specificate una serie di altre opzioni relative all'albero:

- il formato dell'output desiderato (di default è HTML, può essere richiesto XML);
- il numero n_sites di siti che devono essere considerati nella costruzione dell'albero: verranno utilizzati solo gli n_sites siti più recenti etichettati con la tag data (se il parametro non è specificato vengono utilizzati tutti i siti disponibili nel database esplorati relativamente alla tag richiesta);

- il numero massimo *n_tags* di tag che devono essere considerate, per ogni sito, nella costruzione dell'albero: verranno utilizzate le *n_tags* tag più popolari per quel sito (di default questo parametro non è impostato e vengono considerate tutte le tag con le quali il sito è stato salvato almeno una volta);
- il parametro *nocompress*, una variabile booleana che determina se l'albero deve essere compresso oppure no (di default essa è falsa, ovvero la compressione dell'albero viene eseguita);
- il parametro *explore*, una variabile booleana che determina il comportamento del server nel caso il database non contenga i dati necessari per elaborare la richiesta: se il parametro ha valore positivo, il server attiverà gli altri due moduli per ottenere i dati necessari e soddisfare la richiesta.
- il parametro *HTML_page*, una variabile booleana che determina se il risultato debba essere inserito in una pagina HTML completa (di default essa è falsa, ovvero viene restituita solo la struttura HTML dell'albero).

Il significato di questi parametri risulterà più chiaro nei paragrafi successivi.

Il formato XML è uno standard consolidato per lo scambio di dati fra applicazioni e garantisce il disaccoppiamento totale fra contenuti e presentazione. Accanto ad esso abbiamo previsto anche la possibilità di utilizzare un output HTML, per la maggiore comodità pratica di integrazione all'interno di una pagina Web. L'output in formato HTML contiene solo tag di "struttura" che permettono di definire la gerarchia dell'albero e gli attributi di ogni nodo. Tutto ciò che ha a che fare con la presentazione delle informazioni è assente.

Sul lato client, un'estensione del browser permette di eseguire uno script che arricchisce le pagine di del.icio.us con i dati ricevuti dal Web server. I contenuti vengono integrati nelle pagine di del.icio.us creando una barra laterale aggiuntiva, all'interno della quale è possibile visitare l'albero in modo interattivo.

4.2 Funzionamento

Lo schema 4.2 mostra l'interazione fra le varie componenti del sistema attraverso la sequenza standard degli eventi che portano dalla richiesta dell'utente

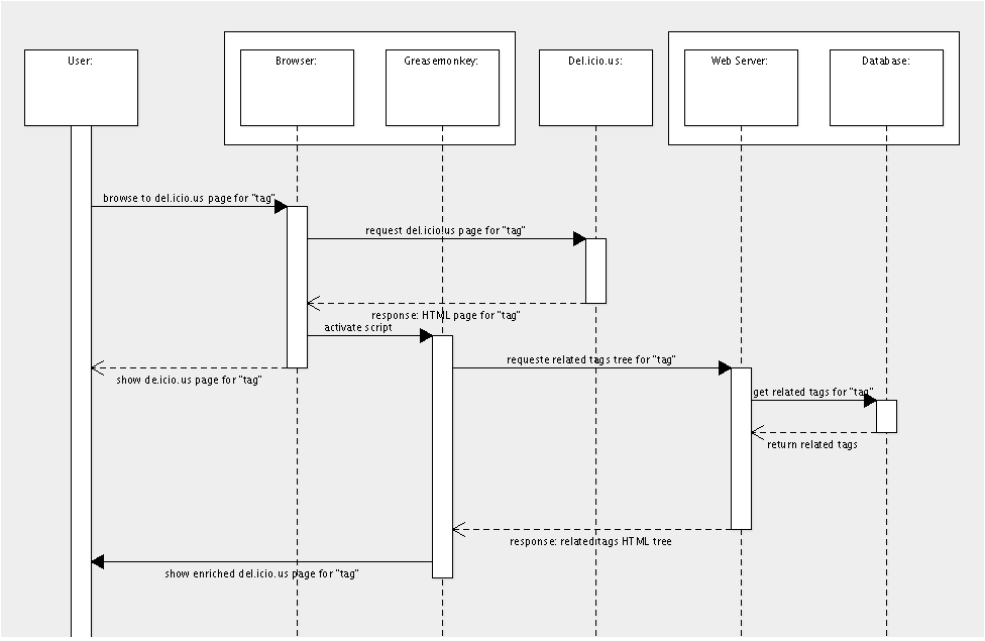

Figura 4.2: sequence diagram UML che rappresenta le interazioni fra le varie componenti del sistema.

alla visualizzazione del risultato completo nel browser:

1. l'utente digita nel browser l'indirizzo della pagina di del.icio.us relativa a una tag;
2. il browser richiede la pagina al server del.icio.us;
3. il server del.icio.us risponde;
4. il browser attiva lo script Greasemonkey;
5. lo script inoltra una richiesta al Web server specificando la tag (e altri eventuali parametri);
6. la pagina di del.icio.us intanto viene caricata normalmente dal browser e mostrata all'utente;
7. il Web server Perl estrae dal database i dati relativi alla tag richiesta;
8. il Web server costruisce l'albero semantico delle tag correlate e lo restituisce in formato HTML allo script Greasemonkey;
9. lo script modifica dinamicamente la pagina del.icio.us e crea una barra laterale aggiuntiva;

10. la pagina completa viene mostrata all'utente.

Il tempo di esecuzione del programma sul server per la creazione dell'albero (punto 8) è lineare nella dimensione dell'input, ovvero nel numero delle tag correlate. Questo valore a sua volta dipende dai parametri n_sites e n_tags, più precisamente è proporzionale al loro prodotto. Le osservazioni sulla complessità degli algoritmi per la creazione dell'albero semantico e sui tempi di risposta del Web server sono trattate nel paragrafo 4.6.4.

Il tempo di risposta del server è variabile, a seconda dei parametri e della dimensione dell'input, nell'ordine di grandezza dei secondi; i risultati sono discussi nel paragrafo 5.4.

In ogni caso è importante notare che dal punto di vista dell'utente, fino a quando i dati aggiuntivi non sono disponibili, la pagina di del.icio.us viene caricata e visualizzata normalmente. Quando i risultati delle elaborazioni del server sono disponibili, essi vengono integrati "al volo" dallo script, nella pagina che l'utente sta visualizzando. I tempi di attesa dunque riguardano solo le informazioni aggiuntive, mentre il caricamento della pagina non viene ritardato per niente dal nuovo servizio.

Dopo l'ultimo passaggio, l'utente può continuare a interagire con lo script di Greasemonkey, per espandere l'albero. Lo script, che risiede nel browser dell'utente, ormai ha tutte le informazioni necessarie: le interazioni successive sono immediate.

Qualora il server non trovi le informazioni necessarie a soddisfare la richiesta, a seconda del valore del parametro *explore* esso potrà attivare gli altri due moduli; dovrà quindi attendere che essi abbiano finito di elaborare i dati, poi potrà estrarre le informazioni dal database e proseguire come mostrato nello schema 4.2. I tempi di risposta in questo caso diventano più lunghi.

Nei paragrafi successivi saranno illustrate le problematiche principali affrontate nella progettazione e nella realizzazione dei componenti del sistema e le soluzioni algoritmiche e implementative adottate.

4.3 L'uso di Wordnet

Come abbiamo osservato nel paragrafo 3.2, uno dei nodi centrali da affrontare in questo lavoro riguarda la mappatura semantica delle tag e i problemi che questo processo pone: si tratta di conciliare la struttura rigida e gerarchica di un'ontologia con quella flessibile e "democratica" di una folksonomia. In

riferimento alla scelta dell'ontologia, WordNet, e in particolare della sola relazione di iperonimia/iponimia definita per i sostantivi, abbiamo accennato ad alcune problematiche specifiche nella sezione 3.4.2.

In questo paragrafo affronteremo in modo più completo queste problematiche e le relative scelte progettuali che abbiamo adottato.

4.3.1 Il riconoscimento delle tag

Il primo ostacolo incontrato, per effettuare la mappatura semantica delle parole chiave sull'ontologia definita dalla relazione di iperonimia/iponimia di WordNet, è quello di riconoscere le tag per poterle associare a qualche elemento dell'ontologia.

In particolare, i problemi principali relativi a come trattare le tag sono tre:

- il primo riguarda le parole che non sono riconosciute da WordNet;
- il secondo è quello delle parole che non sono sostantivi, e di quelle che hanno significato sia come sostantivi sia come altro;
- il terzo è quello di come trattare le forme flesse e le differenze fra caratteri maiuscoli e minuscoli.

Il primo problema, a meno di espandere in qualche modo l'ontologia, ha una soluzione ovvia e obbligata, che è quella di trascurare le parole che non vengono riconosciute.

Questo fatto è per certi versi meno grave di come può apparire a prima vista, se si studiano i dati relativi alla frequenza delle tag che sono e che non sono riconosciute da WordNet: essi infatti mostrano un tipico andamento di tipo *power law*.

Gli esperimenti che presenteremo sono stati svolti su un campione significativo estratto da del.icio.us, costituito dai dati completi relativi a 30.000 utenti, per un totale di oltre 480.000 tag usate, tre milioni e mezzo di risorse etichettate e 20 milioni di relazioni ternarie di tagging ("user-url-tag"); i risultati sono disponibili sul Web [Eynard, 2007b].

Prima di confrontare una tag con il lessico, ne è stato eseguito lo stemming; questo permette di riconoscere diverse forme di parole, in particolare i plurali, che rappresentano una buona percentuale delle tag.

Effettuando le analisi sul dataset è emerso come primo dato generale che solo poco più dell'8% delle tag totali sono parole che appartengono a WordNet; questo dato non deve stupire più di tanto, se si considera che le

parole totali contenute in WordNet sono circa 150.000, meno di un terzo delle diverse tag contenute nel campione.

Il dato può sembrare scoraggiante, ma se prendiamo in considerazione le 140 tag più popolari, otteniamo che 114 sono riconosciute come appartenenti al lessico: la percentuale risultante è dell'81.4%. Il dato è notevole, e lo è a maggior ragione se si considera che il dominio di del.icio.us è sbilanciato verso un ambito specifico, quello del software, di cui WordNet non ha una copertura completa, e dove c'è, molto più che in altri settori, un proliferare di parole tecniche e neologismi, anche molto usati, che non appartegono propriamente alla lingua inglese: si pensi a parole come "Firefox" o "Web2.0", che sono fra le più popolari in assoluto in del.icio.us.

Se prendiamo le prime 1000 tag più popolari, la percentuale di quelle riconosciute da WordNet è del 76.4%; se consideriamo le mille tag successive, la percentuale scende al 67.1%. Proseguendo la percentuale continua a scendere rapidamente secondo una tipica distribuzione *power law*, come mostra la figura 4.3. Il grafico è stato ottenuto suddividendo le tag in gruppi di mille, in base alla loro frequenza; in ascissa sono rappresentati i gruppi di tag, in ordine decrescente di popolarità, mentre in ordinata è rappresentato, per ciascuno di essi, il numero delle parole che sono contenute in WordNet. La *long tail* si assesta su un valore compreso fra il 3% e il 5% di tag riconosciute dal lessico per quelle meno utilizzate.

Un secondo dato incoraggiante si ottiene ragionando sulle singole risorse e calcolando, per ciascuna di esse, la percentuale media delle tag appartenenti a WordNet, fra tutte quelle ad essa attribuite: il valore medio risultante è del 57%; se si escludono dal calcolo i siti etichettati con una sola tag, il cui apporto alla media è meno interessante, si ottiene un valore del 64%: quest'ultimo dato è probabilmente dovuto anche al fatto che molte tag singole usate dagli utenti per risorse poco popolari (si tratta infatti di risorse a cui è stata attribuita una sola tag in tutto, spesso da un singolo utente) sono dei codici o delle sequenze di caratteri arbitrarie che hanno senso solo per l'utente, per ritrovare facilmente quel sito.

Questi risultati mostrano che la restrizione di utilizzare solo le parole presenti in WordNet non è grave come potrebbe apparire; ci sono anche delle tag che non sono presenti nel lessico e che possono essere significative per molti utenti, ma queste sono una quantità limitata: per la grande maggioranza, infatti, le tag non riconosciute da WordNet sono quelle che si collocano nella long tail, e che molto spesso hanno un significato comprensibile solo all'utente che le ha utilizzate; per questo motivo non costituiscono una perdita di vitale importanza, tale da compromettere l'utilità di un sistema basato sulla mappatura delle tag su WordNet.

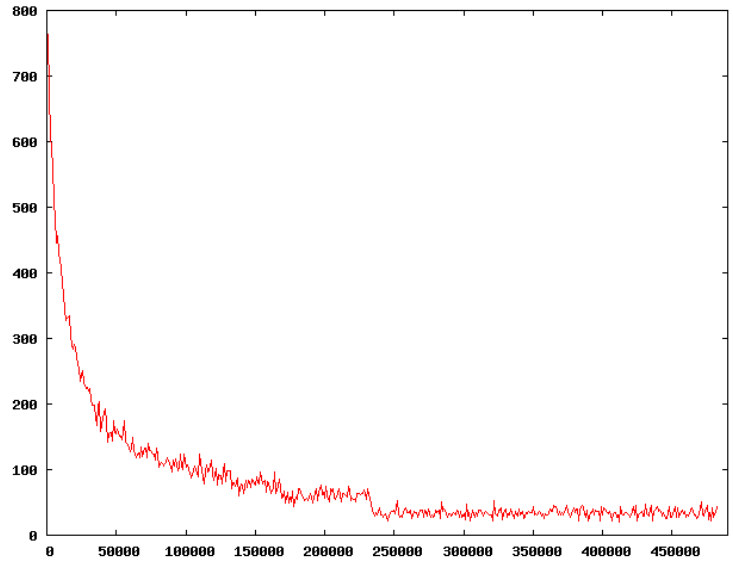

Figura 4.3: il grafico rappresenta la probabilità che una tag sia riconosciuta da WordNet, in funzione della sua frequenza: sull'asse orizzontale sono disposte le tag, raggruppate in sezioni di 1000 e ordinate per popolarità, in ordinata è rappresentato il numero delle tag riconosciute da WordNet per ogni segmento. I dati sono relativi a un campione di 30.000 utenti di del.icio.us, che comprende circa 480.000 tag differenti.

Un aspetto importante da considerare è che fra le tag non riconosciute devono essere comprese anche tutte quelle in lingue diverse dall'inglese, sempre più numerose man mano che del.icio.us si afferma anche fuori dai paesi anglofoni; risolvendo questo problema con l'integrazione del lessico con altre wordnet locali in lingue diverse, ci aspettiamo non solo che aumenti la percentuale delle tag totali riconosciute, ma anche che questa percentuale aumenti in modo particolare fra le tag che superano una certa soglia di popolarità, ovvero che stanno fuori dalla long tail. É naturale infatti supporre che, arricchendo il bacino delle tag riconosciute dall'ontologia con altre parole largamente condivise all'interno di comunità di utenti che parlano la stessa lingua, il fenomeno della power law tenda ad accentuarsi.

Passiamo ad affrontare il secondo problema, quello posto dall'utilizzo della sola relazione di iperonimia/iponimia, che è definita solo sui sostantivi di WordNet, e dunque delle parole che non sono sostantivi. Abbiamo già osservato che i sostantivi costituiscono la maggior parte delle tag riconosciute da WordNet. Questo accade essenzialmente per due motivi. Il primo è che i sostantivi sono più numerosi: delle circa 150.000 parole di WordNet il 75%

sono sostantivi[1]. Il secondo motivo è che i sostantivi si prestano meglio delle altre parole ad essere delle tag.

Quest'ultima tesi è confermata dai dati: se effettuiamo i test sullo stesso campione di 480.000 tag di del.icio.us e calcoliamo la probabilità che queste siano sostantivi di WordNet in base alla frequenza, osserviamo che la distribuzione power law è accentuata. Delle 140 tag più popolari, il 76.4% sono sostantivi (ovvero il 94% di quelle riconosciute come parole di WordNet), delle prime 1000, lo sono il 71% (ovvero il 92.8% di quelle riconosciute in WordNet), delle prime 10.000 il 49.2% (ovvero il 92% di quelle riconosciute in WordNet). In totale, sono riconosciute come sostantivi il 6.85%, ovvero l'85% di quelle riconosciute in WordNet.

In tutti i casi si osserva come la percentuale di sostantivi fra le parole riconosciute da WordNet sia piuttosto alta, e sia considerevolmente più alta per le tag più frequenti. Nei calcoli abbiamo considerato come sostantivi tutte le parole che hanno almeno un significato in WordNet come sostantivi.

Considerando automaticamente sostantivi tutte le parole che hanno almeno un significato come nomi si possono commettere delle imprecisioni, quando queste hanno anche valore come aggettivi o verbi. Per esempio, la tag "cook", utilizzata da alcuni utenti per indicare il verbo "cucinare", può essere interpretata come sostantivo solo con il significato di "cuoco". Questa interpretazione in molti casi non è precisa. D'altro canto si può osservare che un numero molto più alto di utenti utilizza invece la tag "cooking", un verbo sostantivato che nel senso comune si presta meglio ad essere una tag. Essa viene riconosciuta correttamente da WordNet come sostantivo, con il significato di "atto di cucinare".

Per quanto riguarda le parole che hanno almeno un senso come sostantivi, abbiamo scelto di includerle in ogni caso nel database e di attribuire loro poi uno di questi significati con l'algoritmo di disambiguazione, anche se questo introduce un elemento di imprecisione; questa scelta permette di includere delle parole che altrimenti verrebbero scartate, dove il significato come nome spesso ricalca quello della parola come verbo o aggettivo, essendo la radice la stessa. Si pensi a un caso tipico come la parola "jump", che ha in WordNet 13 possibili significati come verbo e 6 come nome. I significati, come nome e come verbo, si collocano nelle stesse aree semantiche e sono molto vicini fra loro. Se la parola venisse individuata come verbo nel significato più generale di "saltare" e scartata, si perderebbe la parola; forzandone l'interpretazione come nome, invece, essa può essere associata al significato generale di "salto",

[1] I dati sono disponibili sul sito di WordNet all'indirizzo http://wordnet.princeton.edu/man/wnstats.7WN

mantenendo un'indicazione corretta sull'ambito semantico di appartenenza.

Per quanto riguarda il terzo problema, quello delle forme flesse, in del.icio.us l'unica restrizione riguardo alle tag consentite è quella dei caratteri che è possibile utilizzare (per esempio una tag non può contenere spazi); per il resto non è previsto nessun controllo e ogni utente è libero di usare qualsiasi sequenza di caratteri consentiti. Anche se in alcuni contesti tendono ad affermarsi degli standard impliciti, la disomogeneità delle tag è un aspetto costante con cui bisogna confrontarsi. In particolare i problemi più comuni sono quelli del singolare e del plurale (e più in generale delle parole in forma flessa), e delle maiuscole e minuscole.

Questi due casi sono trattati in modo diverso in del.icio.us; infatti le differenze fra maiuscole e minuscole sono tollerate dal sistema, ma non determinanti ai fini dell'identificazione di una tag: l'uso della tag "Italy" e "italy" è indifferente per il sistema, anche se nella pagina dell'utente la tag utilizzata risulterà con l'iniziale maiuscola o minuscola a seconda della sua scelta. Per quanto riguarda invece le forme flesse di una parola, come singolare e plurale, esse non vengono identificate dal sistema e due forme diverse costituiscono tag a tutti gli effetti distinte.

Sarebbe stato possibile mantenere, sia nel database sia nell'albero finale, le differenti forme flesse di una parola, ed eventualmente anche le differenze fra maiuscole e minuscole, considerando che queste disomogeneità in alcuni casi possono costituire un elemento di ricchezza in una folksonomia. Abbiamo invece scelto di operare una semplificazione uniformando le parole prima di inserirle nel database, per diversi motivi.

Innanzitutto vi è la considerazione di carattere generale che queste differenze di forma di una parola sono per lo più irrilevanti, e in ogni caso la mancanza di precisione, che è una caratteristica propria delle folksonomie, rende inaffidabili questo tipo di dati; le tag infatti sono scelte dagli utenti in modo per lo più approssimativo, soprattutto rispetto ad aspetti di dettaglio.

Per quanto riguarda il risultato finale, poi, è importante che l'albero sia il più compatto possibile per essere fruibile dall'utente; la presenza nell'albero di più nodi corrispondenti a forme differenti della stessa parola ne appesantirebbe eccessivamente la struttura e ne renderebbe più difficile l'esplorazione.

Infine c'è un motivo di coerenza dei dati nel database, importanti in particolare per valutare i coefficienti di correlazione fra le tag e per la disambiguazione: è utile che il peso di ogni tag e il synset di appartenenza siano valutati unitariamente per tutte le varianti di forma della parola.

4.3.2 La polisemia

In WordNet una stessa parola può appartenere a più *synset*, gruppi di sinonimi che rappresentano le unità di significato nell'ontologia. La maggior parte delle relazioni semantiche definite in WordNet, compresa quella di iperonimia-iponimia, sono definite fra synset e non fra parole. Dunque per potersi avvalere di queste relazioni non si possono utilizzare le tag così come sono, ma per ciascuna di esse deve essere specificato il synset di appartenenza.

Nell'algoritmo Castanet [Stoica et al., 2006], che abbiamo introdotto nel paragrafo 3.4.2, il senso di una parola fra i possibili synset viene stabilito una volta per tutte e il risultato ha valore per tutte le occorrenze di quella parola all'interno dell'albero.

Questa scelta è resa possibile in primo luogo dal fatto che Castanet lavora in contesti limitati, come le ricette di cucina o gli articoli biomedici; esso si basa sull'assunzione che ci sia un ambito semantico privilegiato. Se una parola ha più significati e uno di questi è direttamente correlato all'ambito semantico principale, quest'ultimo significato verrà scelto. Sulla base di queste assunzioni, e delle parole che appartengono a un solo synset, viene costruito lo scheletro, la struttura base dell'albero; le altre parole sono inserite cercando il percorso più breve che le leghi allo scheletro mediante relazioni di iperonimia.

Uno dei vantaggi principali di questo algoritmo per la disambiguazione è che esso porta a un albero compatto; il costo maggiore è la semplificazione che esso comporta, che può risultare eccessiva e generare grosse imprecisioni.

Questa soluzione infatti può dimostrarsi approssimativa in molti casi: si osservi per esempio che, anche restando nell'ambito ristretto delle ricette, la parola "turkey" potrebbe riferirsi a un tacchino alla messicana, ma anche a una ricetta turca. L'algoritmo Castanet propenderebbe probabilmente in ogni caso per la prima ipotesi, trovando una correlazione diretta fra "tacchino" e "cucina", anche se le parole circostanti facessero riferimento a una "ricetta vegetariana mediorientale".

La soluzione di Castanet diventa nel nostro caso tanto approssimativa da essere sostanzialmente impraticabile. Nella nostra folksonomia infatti il contesto è l'intero Web; si potrebbe considerare come contesto la tag principale oggetto dell'esplorazione, ma questa assunzione non sarebbe corretta: infatti la tag principale definisce soltanto lo spazio di esplorazione, che non necessariamente coincide con un ambito semantico.

La via che abbiamo deciso di praticare è quella di attribuire a ogni tag un

significato in relazione al sito a cui si riferisce. In questo modo la stessa parola potrà appartenere a *synset* differenti, anche nell'ambito dell'esplorazione della stessa tag principale; la tag avrà invece un unico significato rispetto ad ogni sito: questa è una semplificazione che ci è sembrata del tutto accettabile, poiché ogni risorsa rappresenta realmente un ambito semantico.

4.3.3 Ereditarietà multipla

L'uso dell'ontologia basata sulle relazioni di iperonimia/iponimia di WordNet per la costruzione dell'albero semantico pone un altro problema accanto a quello della polisemia: essa comprende la possibilità, pur non frequente, per un synset, di avere più iperonimi. In altre parole, la struttura dell'ontologia è quella di un albero che contiene dei cicli, mentre noi vogliamo ottenere un albero aciclico; una gerarchia più rigida infatti risponde meglio all'esigenza di rappresentare visivamente l'albero per permettere una esplorazione organica, in accordo coi requisiti dell'interfaccia utente definiti nel paragrafo 3.6.

Castanet risolve questo problema insieme a quello della polisemia, in una volta sola; l'algoritmo infatti esegue una disambiguazione dei cammini di iperonimia e non del senso delle parole. Per ogni parola, Castanet cerca fra tutte le catene di iperonimi possibili quella che è più vicina all'ambito semantico generale di riferimento e che meglio si integra con l'albero già esistente; la catena di iperonimi viene scelta indipendentemente dal senso della parola a cui corrisponde.

Noi abbiamo deciso di scegliere invece il senso della parola in base al contesto e, una volta stabilito il synset di appartenenza, cercare un cammino di iperonimi per la parola. Il problema della molteplicità dei cammini di iperonimia è in buona parte risolto, una volta stabilito il synset di appartenenza, ma come abbiamo osservato ci sono anche casi, pur non molto frequenti, in cui un synset ha più di un genitore nell'albero.

Questo problema è concettualmente diverso da quello della polisemia: infatti quando dobbiamo attribuire un senso a una parola le diverse alternative sono tutte possibili a priori in contesti differenti, mentre in un determinato contesto una sola di queste è quella corretta. Nel caso della molteplicità dei cammini di iperonimia, invece, possiamo osservare che essi sono tutti sempre corretti, ma semplicemente trattano aspetti diversi del problema, proprio come i facet.

Per esempio, il synset definito dai sinonimi "person, individual, someone, somebody, mortal, human, soul" ha come iperonimi sia il synset compren-

dente "organism, being", sia quello identificato da "causal agent, cause, causal agency": una persona può essere considerata sia un organismo vivente sia un'entità che può causare degli eventi. Entrambe queste interpretazioni sono corrette, in generale.

In alcuni contesti un iperonimo può essere più pertinente degli altri; sarebbe possibile dunque scegliere sempre l'iperonimo più pertinente, a seconda dei casi, basandosi sul contesto della parola che sta alla base della catena: una soluzione analoga a quella adottata per determinare il senso di una tag.

Questa soluzione ci è sembrata però complicare eccessivamente il problema senza portare benefici indispensabili e abbiamo scelto di considerare invece sempre, per ogni synset, il primo iperonimo proposto da WordNet. Questa scelta è motivata da alcune osservazioni teoriche e altre di carattere pratico.

Prima di tutto, come discusso precedentemente, ogni relazione di iperonimia rappresenta comunque un'informazione corretta; in particolare quello presentato per primo da WordNet è sempre l'iperonimo che viene considerato il più rilevante; ha senso che ogni parola nella gerarchia sia associata sempre all'iperonimo principale: in questo modo l'albero risulta più compatto.

Occorre poi ricordare che la presenza di più di un iperonimo fra cui scegliere non rappresenta la norma, ma un insieme abbastanza limitato di casi, localizzati per lo più nei nodi più in alto della gerarchia di WordNet; questi nodi corrispondono in buona parte alle categorie di alto livello, che nel nostro albero vengono eliminate.

Inoltre la scelta del primo iperonimo, in modo deterministico, in ogni caso di indecisione, consente di garantire che sia percorso sempre lo stesso cammino per tutti i sinonimi; i sinonimi infatti appartengono allo stesso synset e la relazione di iperonimia è definita per i synset, non per le singole parole.

Da un punto di vista pratico, infine, abbiamo considerato che mentre la disambiguazione delle tag può essere eseguita una volta per tutte in una fase precedente alla richiesta dell'utente, la ricerca degli iperonimi è un passaggio che deve essere svolto nella fase di costruzione dell'albero, e deve essere ripetuto per ogni tag: esso potrebbe appesantire eccessivamente l'algoritmo aumentando i tempi di attesa dell'utente.

4.4 L'estrazione dei dati

Per come è strutturato il sistema, il problema che ci troviamo di fronte, riguardo all'estrazione dei dati, è il seguente: data una parola chiave, trovare tutte le tag ad essa correlate.

Non esiste una definizione univoca di quando due tag siano correlate; noi abbiamo considerato correlate due tag se esiste almeno una risorsa che è stata etichettata con entrambe, anche da utenti diversi.

Le informazioni che occorrono sono dunque:

- quali siti sono stati etichettati con la tag che vogliamo esplorare;
- quali altre tag sono state utilizzate per etichettare questi siti.

Le API di del.icio.us rendono disponibili solo alcuni dati, per esempio permettono a un utente di accedere ai propri bookmark, previa autenticazione, ma non permettono di accedere a tutto il database. I feed RSS invece, che mostrano i dati in formato XML, si limitano ai dati più recenti: per esempio gli ultimi 30 siti etichettati con una certa tag, gli ultimi 30 siti salvati da un certo utente o gli ultimi 30 utenti che hanno salvato un certo sito. In ogni caso si tratta di quantità molto ridotte di informazioni, che possono essere utili per esempio per monitorare nel tempo una tag; nonostante la comodità di avere i dati in formato XML, questa soluzione non può essere adottata nel nostro caso.

Per raccogliere tutti i siti etichettati con una certa tag, o anche solo una quantità superiore a 30 elementi, è necessario collegarsi tramite la normale interfaccia Web del sistema ed estrarre i dati dal codice HTML. Questo rappresenta una debolezza del programma, nel senso che il server potrebbe cambiare la struttura delle pagine o qualche particolare del codice HTML; in tal caso il parser andrebbe riadattato. Per ottenere una quantità interessante di dati però questa è l'unica soluzione praticabile. Anche attraverso l'interfaccia Web c'è qualche limite alle informazioni che possono essere estratte, ma si tratta di ordini di grandezza differenti.

Abbiamo mostrato nello schema 3.2 la struttura del sito del.icio.us; ora analizzeremo più in dettaglio le pagine che contengono i dati che ci interessano.

La pagina relativa a una tag contiene un elenco di tutte le volte che essa è stata utilizzata: per ogni elemento della lista dunque sono specificati:

- l'url del sito;
- l'utente che l'ha salvato;
- le tag che ha utilizzato (fra queste c'è sempre la tag oggetto della pagina);

- la data;
- la pagina del.icio.us relativa all'url, in caso esso sia stato salvato da almeno un altro utente;

La struttura di queste pagine è simile a quella delle pagine relative a un utente, che consistono in un elenco di bookmark. Il massimo di elementi che possono essere visualizzati in ogni pagina è 100: vengono mostrati i 100 bookmark più recenti. É possibile richiedere le pagine con i bookmark precedenti, ma mentre per quanto riguarda le pagine di un utente è possibile risalire fino al primo sito che ha salvato, per le tag non è così: si può richiedere solo fino alla centesima pagina. Questo vuol dire che è possibile visualizzare, in un certo momento, solo gli ultimi 10000 siti etichettati con una certa tag.

La pagina relativa a un url contiene l'elenco degli eventuali commenti degli utenti su quel sito; in una spalla laterale della pagina è presente l'elenco di tutti gli utenti che l'hanno salvato. Gli utenti sono mostrati in ordine cronologico, e l'elenco è suddiviso per mesi. Per ogni utente sono specificate le tag che ha utilizzato; non è mostrata la data precisa. Può essere visualizzata la lista completa degli utenti.

L'unico modo per avere tutti i dati relativi a una tag sarebbe quello di esplorare l'intero del.icio.us tramite le pagine degli utenti, o tramite quelle degli url; questa soluzione sarebbe decisamente sproporzionata rispetto allo scopo di questo lavoro, per il quale 10000 siti per ogni tag sono un numero più che sufficiente. Monitorando una tag nel tempo è possibile raccogliere quantità superiori di dati.

Al Politecnico di Milano è stato sviluppato uno scraper distribuito per del.icio.us; il software, rilasciato con licenza libera, può essere installato ed eseguito da più client, che concorrono a riempire un database centralizzato [Eynard, 2007a]. Lo scraper raccoglie i dati a partire dalle pagine degli utenti; in questo modo si potrebbe arrivare a raccogliere tutti i dati del sistema e dunque avere a disposizione, per ogni tag, più dei 10000 siti accessibili tramite la relativa pagina, potenzialmente anche tutti.

Il numero di siti da considerare per l'esplorazione di una tag è un parametro del programma, che può essere specificato; nell'implementazione attuale il valore massimo è 10000, per i limiti imposti da del.icio.us, ma può essere aumentato monitorando una tag nel tempo. Il valore di default è 500.

Per ogni url esplorato invece abbiamo deciso di considerare tutti gli utenti che l'hanno salvato e di conteggiare tutte le tag da essi attribuite; in questo modo gli url oggetto di analisi possono essere descritti nel modo più completo

possibile. Se si vuole imporre un limite al numero di tag da considerare per ogni sito, questo come vedremo sarà reso possibile con dei parametri, relativi agli algoritmi che utilizzano questi dati; in ogni caso i dati memorizzati nel database sono completi.

Avendo raccolto tutte queste informazioni, sarà possibile ridurre il numero di tag più avanti nell'algoritmo secondo diversi criteri: scegliendo di considerare, per ogni sito, soltanto le n tag più utilizzate, oppure solo quelle scelte almeno da un certo numero di utenti.

4.4.1 Il crawler

Ora che abbiamo affrontato tutte le principali problematiche generali relative all'estrazione dei dati, è possibile descrivere il funzionamento del crawler.

Il crawler richiede al server la pagina relativa alla tag che deve essere esplorata e scandisce il codice HTML estraendo, per ogni elemento, l'url, l'utente e le tag; per ogni tag la tripletta "url-utente-tag" viene inserita nel database. Una volta scandita l'intera pagina, se il parametro n_sites è superiore a 100 il crawler richiede la pagina con i 100 elementi successivi, e così via.

Per ogni bookmark incontrato, il crawler verifica se esso è stato salvato da almeno un'altra persona; in caso positivo richiede la pagina relativa al bookmark e la scandisce, memorizzando tutte le triplette "sito-utente-tag".

In molte di queste triplette potrebbe non comparire la tag che è oggetto dell'esplorazione, ma la correlazione è data dal fatto che il sito è stato etichettato con essa almeno una volta.

Un fatto da osservare è che l'esplorazione procede per url: i dati relativi ad un sito sono salvati in modo completo e indipendente dalla tag oggetto di esplorazione nell'esecuzione corrente. Se un'esecuzione successiva dell'algoritmo di esplorazione, relativa a una tag differente, incontra un sito già esplorato, non c'è bisogno di richiedere la pagina del'url e di scandirla, se non per aggiornare i dati con gli eventi più recenti non ancora presenti nel database.

Un crawler rappresenta una modalità di interazione con il sito Web diversa da quella umana, per la quale il sito è progettato. Per questo per realizzare un crawler è opportuno avere alcune accortezze. Per non creare problemi al server sommergendolo con molte richieste automatiche ravvicinate è opportuno assicurarsi che passi un certo intervallo di tempo fra le richieste delle pagine. Questo fra l'altro evita spiacevoli inconvenienti come

quello dell'identificazione da parte del server e del blocco temporaneo delle richieste provenienti dall'indirizzo IP del crawler.

Le richieste al server sono intervallate dalle operazioni di parsing e di lettura e scrittura del database, ma queste operazioni sono relativamente brevi e i tempi possono non essere sufficienti; per questo il crawler attende sempre un certo numero di secondi prima di richiedere al server una nuova pagina. Il tempo di attesa prima di ogni richiesta è un parametro del programma; di default vale due secondi.

4.4.2 La base di dati

La base di dati contiene essenzialmente due categorie di dati: quelli inseriti dal crawler e quelli inseriti dal modulo per la disambiguazione.

Per quanto riguarda i primi, essi sono strutturati come relazioni ternarie "utente-risorsa-tag". Questa struttura ricalca quella sottostante al sito di del.icio.us; da queste relazioni è possibile inferire tutti i dati di correlazione.

In accordo con le scelte discusse nel paragrafo 3.4.2, prima di essere inserite nel database, le tag sono riportate in un formato standard. Ogni tag prima di essere memorizzata deve essere processata nei seguenti passaggi:

- uniformazione di caratteri (tutto minuscolo);
- stemming.

L'algoritmo con cui viene eseguito lo stemming è lo stesso che viene usato internamente da WordNet; questo garantisce la coerenza delle tag contenute nel database con WordNet. L'algoritmo di stemming è disponibile nella libreria Perl WordNet::Similarity.

Se una parola non viene riconosciuta come appartenente a WordNet con almeno un significato come sostantivo, essa non viene inclusa nella base di dati, in quanto non potrebbe essere utile in nessuno dei passaggi successivi.

I dati elaborati dal modulo per la disambiguazione sono memorizzati in una tabella apposita, strutturata in base a una relazione binaria "tag-sito", come sarà illustrato nel prossimo paragrafo.

La base di dati è stata realizzata con MySQL, un DBMS disponibile con licenza libera, che offre buone caratteristiche di portabilità.

4.5 La disambiguazione delle tag

Per effettuare la disambiguazione delle tag, relativamente al sito a cui sono state attribuite, era possibile basarsi su vari elementi legati al sito: il titolo, il contenuto, i commenti degli utenti di del.icio.us, le altre tag. La nostra scelta è ricaduta su questa ultima possibilità per diverse ragioni.

I commenti degli utenti contengono informazioni spesso poco interessanti se non fuorvianti per estrapolare un ambito semantico. Essi possono riguardare esperienze o giudizi personali, o aspetti del tutto marginali di un sito, per esempio aspetti tecnici, come accade spesso, che nulla hanno a che vedere con il contenuto. Le parole contenute nei commenti non sono dunque adatte ad essere considerate rappresentative di un sito. Inoltre esse costituiscono un dato fortemente disomogeneo: molti siti non possiedono neanche un commento da parte degli utenti, mentre altri ne sono colmi; proprio la presenza di un commento su un sito può indurre facilmente altri utenti a esprimere il loro punto di vista.

I contenuti di un sito rappresentano certo il dato più completo relativamente al sito stesso, ma una loro analisi pone diverse difficoltà, che rendono questa soluzione sostanzialmente impraticabile: molte pagine comprendono testi estesi, e dunque una quantità di parole onerosa da analizzare per estrapolare la semantica; altre possono contenere soltanto contenuti multimediali difficilmente interpretabili in modo automatico. Inoltre ci possono essere pubblicità, commenti, link, e una quantità di dati disomogenei all'interno di ogni singola pagina; orientarsi in questa giungla è un'impresa titanica, non certo alla portata di questo lavoro di tesi.

Il titolo, ovvero il campo *title* di una pagina HTML, è un metadato inserito da chi ha realizzato il sito: esso costituisce in un certo senso un minuscolo elemento di Web semantico nel Web attuale e si trova in una posizione facilmente accessibile nel codice HTML della pagina. Per questo potrebbe sembrare ideale per lo scopo. Tuttavia anche questo è un dato soggetto a troppe variabili per potere essere considerato affidabile.

Spesso il titolo ha la forma di una mission, di un gioco di parole, di una frase tesa ad attirare l'attenzione o a colpire l'immaginario degli utenti; frequentemente si serve della metafora. La metafora è nemico acerrimo di qualsiasi applicazione di elaborazione del linguaggio naturale: essa ha a che fare con la ricchezza del linguaggio, con la varietà dei significati che gli esseri umani riescono ad attribuire alle parole, seguendo percorsi semantici tortuosi e imprevedibili. Si tratta di un processo tipico della mente umana, quando procede per associazione, come nei sogni, e non per induzione logica: per questo la semantica di un testo basato su una metafora è difficilissima da

interpretare per un'applicazione automatica.

In molti casi inoltre il titolo è uno solo per tutte le pagine all'interno dello stesso sito o dello stesso dominio; questo capita spesso, purtroppo, anche per siti vasti, sia per quantità sia per varietà delle informazioni contenute. Se tutte le pagine del sito di una rivista online hanno lo stesso titolo, con una descrizione generica della rivista, questo titolo non dà nessuna informazione utile sui contenuti delle pagine relative ai singoli articoli. Infine c'è il problema della lingua: non tutti i siti hanno un titolo in inglese.

Oltre ai titoli inseriti come metadati all'interno delle pagine HTML, ci sono i titoli attribuiti dagli utenti nel salvare i bookmark. Si potrebbero allora analizzare questi titoli, ma questa soluzione ricade negli stessi problemi della precedente: infatti la grande maggioranza degli utenti di del.icio.us ha la tendenza a lasciare come titolo di un sito quello suggerito dal sistema come default, ovvero ancora il campo *title* della pagina HTML. Questo avviene anche perché il modo principale che gli utenti hanno per attribuire un valore semantico a un sito è un altro: quello di "taggarlo".

Nelle tag si manifesta nel modo più completo e compatto la semantica che gli utenti vogliono attribuire a una risorsa: esse rappresentano la fonte più naturale, nell'ambito di una folksonomia, anche per un compito di disambiguazione.

Il problema della disomogeneità dei dati esiste anche per le tag, soprattutto nel caso di una folksonomia broad, come del.icio.us: i siti più popolari avranno un numero maggiore di tag; tuttavia questa disomogeneità è limitata. Infatti bisogna considerare che anche un sito salvato da una sola persona possiede generalmente più di una tag, mentre al crescere degli utenti tendono ad affermarsi fortemente alcune parole chiave condivise: è la *power law*. Il numero di tag per un singolo sito può comunque risultare eccessivo, soprattutto in caso di siti particolarmente popolari, per via dell'altra faccia del problema: la *long tail*, per cui nonostante l'emergere di tag largamente condivise molti utenti seguiranno in ogni caso propri schemi mentali differenti; questo problema può però essere facilmente aggirato, tagliando in un certo punto la "lunga coda".

Le tag si dimostrano particolarmente adatte per un compito di disambiguazione basato su confronti fra parole. Per analizzare un testo (che sia un titolo, un commento, o il contenuto di una pagina) occorre prima di tutto un passaggio preliminare, che consiste nell'estrarre le parole più significative, scartando per esempio gli articoli e le congiunzioni; questo passaggio è molto delicato, perché non è sempre facile scegliere quali parole siano più adatte a rappresentare l'ambito semantico di una frase. Con le tag questo passaggio

non è necessario, perché i dati più interessanti sono già a disposizione nella forma più comoda: un insieme di parole chiave. Una buona parte del lavoro è già stata compiuto in qualche modo dagli utenti.

4.5.1 La libreria Perl SenseRelate

Per la disambiguazione ci siamo serviti delle librerie Perl WordNet::Similarity [Pedersen et al., 2004] e WordNet::SenseRelate [Patwardhan et al., 2005], che mettono a disposizione rispettivamente diverse metriche basate su WordNet per calcolare la "distanza semantica" fra coppie di parole e diversi algoritmi per la disambiguazione.

L'algoritmo AllWord riceve in ingresso un insieme di parole e restituisce per ciascuna di esse il synset di WordNet a cui quella istanza della parola, in quel contesto, appartiene. L'algoritmo è stato progettato per un testo di qualsiasi dimensione e dunque i confronti vengono eseguiti, per ogni parola, con quelle circostanti, all'interno di una certa finestra di dimensione fissata. É possibile specificare diverse opzioni, fra cui il tipo di metrica per i confronti fra le parole e la dimensione della finestra in cui effettuare i confronti. É anche possibile fissare la *part of speech* di una parola da disambiguare, ovvero restringere il campo dei risultati, per quella parola, ai sostantivi, ai verbi, agli aggettivi o agli avverbi.

L'algoritmo TargetWord è sostanzialmente analogo al precedente ma è fatto per disambiguare una singola parola dato il contesto in cui si trova.

Le metriche per calcolare la "distanza semantica" fra due synset di WordNet sono fornite nel pacchetto Similarity, all'interno di SenseRelate, e sono suddivise in due categorie:

- metriche di **somiglianza**;
- metriche di **correlazione**.

Tra le metriche di somiglianza, la più semplice si basa sul conteggio dei passi che separano due synset all'interno dell'ontologia; altre metriche più complesse tengono conto della profondità nella tassonomia dei due synset da confrontare e dell'antenato comune che si trova più in basso nella gerarchia.

Sempre per calcolare la somiglianza fra due parole, ci sono altre metriche che si basano anche sull'*information content* di un synset, che rappresenta la sua specificità; esso è calcolato come l'inverso della frequenza dell'intero sottoalbero sottostante, ovvero della frequenza di tutti i synset sottostanti nella gerarchia.

In ogni caso si tratta in qualche modo di misure di distanza all'interno dell'ontologia, basate su questo assunto: più due parole si trovano "vicine" nell'ontologia più sono "simili".

Il secondo tipo di metriche è basato invece sul concetto più ampio di correlazione semantica ed è più interessante ai fini della disambiguazione. Due parole molto dissimili fra loro, che si trovano in rami diversi e lontani dell'ontologia, possono essere però fortemente correlate da un punto di vista semantico, e questo è un dato che può essere molto utile per individuare un contesto semantico.

L'algoritmo di Lesk [Lesk, 1986] utilizza una metrica basata sulle definizioni delle parole che devono essere confrontate: la correlazione fra due parole è calcolata in base al numero di parole che cooccorrono nelle loro definizioni. Con la libreria WordNet::Similarity è fornita l'implementazione di una versione riadattata dell'algoritmo, che utilizza le definizioni dei synset di WordNet [Banerjee and Pedersen, 2002].

Sullo stesso principio si basa l'algoritmo dei *context vectors*, che sostanzialmente estende il calcolo delle cooccorrenze alle definizioni delle parole che compongono a loro volta le definizioni delle parole da confrontare.

Infine l'algoritmo di Hirst & St-Onge si basa su catene lessicali, ovvero come alcune metriche di somiglianza cerca il percorso più breve fra due parole nell'ontologia, ma al contrario di queste non si limita alla relazione di iperonimia e considera anche altre relazioni presenti in WordNet, come meronimia, antonimia e pertinenza. L'algoritmo tiene conto anche del numero di "cambi di direzione" nel percorso all'interno dell'ontologia e assegna un punteggio ad ogni percorso.

4.5.2 Algoritmo per la disambiguazione

L'algoritmo che abbiamo sviluppato procede nel seguente modo:

per ogni sito:

1. raccoglie tutte le tag che sono state utilizzate per descriverlo, ordinate in base al numero di utenti che le hanno scelte per quel sito;
2. prende le prime k tag più scelte e le disambigua fra loro utilizzando l'algoritmo AllWord;
3. confronta ciascuna delle prime $n-k$ tag rimanenti con le k tag disambiguate, con l'algoritmo TargetWord.

Le funzioni AllWord e TargetWord sono sempre invocate specificando, per ogni parola, che si tratta di un sostantivo. In questo modo gli eventuali altri significati possibili, relativi a part of speech diverse, come verbi o aggettivi, sono scartati in partenza.

La dimensione k della "base" per la disambiguazione e il numero n di tag che devono essere prese in considerazione per ogni sito sono parametri del programma. Il valore di default di k è 5, quello di n è infinito, nel senso che vengono considerate tutte le tag, anche quelle meno usate. Anche la metrica da utilizzare negli algoritmi AllWord e TargetWord è un parametro del programma; di default viene usato *adapted lesk*.

Se una tag non è stata disambiguata, perché la funzione AllWord o TargetWord non ne è stata in grado, l'algoritmo assegna alla parola il senso corrispondente al synset più probabile, ovvero al primo; in WordNet infatti i numeri dei synset sono attribuiti alle parole in ordine di frequenza, secondo analisi statistiche effettuate su corpus annotati.

I risultati della disambiguazione vengono man mano salvati nel database, nella apposita tabella *disambiguated_tags*. Alla fine del processo di disambiguazione relativo a un sito il database contiene, per ogni tag che gli è stata attribuita, una riga della tabella con le seguenti informazioni:

- il codice identificativo del sito;
- la tag;
- il numero di utenti che l'hanno utilizzata per quel sito;
- il synset di WordNet corrispondente.

Il terzo dato non è strettamente necessario, in quanto potrebbe essere ricavato dal database, ma è riportato nella tabella per comodità: esso evita di eseguire query complesse ogni volta che deve essere estratto un dato, visto che le tag devono essere estratte in ordine di frequenza.

Questo algoritmo porta diversi vantaggi in termini sia di efficacia che di efficienza.

In primo luogo infatti esso garantisce che solo le tag più utilizzate siano determinanti nel processo di disambiguazione e così limita i danni potenziali portati da tag scelte a sproposito: è meno sensibile al rumore.

In secondo luogo l'algoritmo AllWord ha complessità esponenziale nel numero di parole, dunque utilizzarlo in una volta sola per tutte le tag attribuite ad un sito comporterebbe una crescita incontrollata del costo computazionale; costo che potrebbe essere altissimo nel caso di siti salvati da migliaia

di utenti con decine di tag differenti. Utilizzare la funzione facendo scorrere una finestra di dimensione ridotta, come previsto dall'algoritmo, invece non avrebbe senso poiché in questo caso non si tratta di un testo e tutte le tag sono riferite allo stesso oggetto.

La soluzione che abbiamo adottato invece permette di beneficiare dell'accuratezza dell'algoritmo AllWord per le tag più importanti, e di sfruttare il risultato per disambiguare le altre tag.

Di fatto, l'algoritmo TargetWord nella sua implementazione attuale non risponde a tutte le specifiche: alcuni parametri possono essere impostati ma restano di fatto ignorati e alcune opzioni non sono realmente utilizzabili. L'autore promette però che tutte le funzionalità verranno via via implementate. Ora è possibile dare in ingresso all'algoritmo tag già disambiguate, ma questo dato non viene considerato; per questo motivo l'esecuzione del secondo passo dell'algoritmo richiede un certo tempo (nell'ordine di grandezza di qualche secondo per ogni sito). Poiché questa operazione può dover essere ripetuta molte volte per ogni sito, essa è il collo di bottiglia del processo di disambiguazione e dell'algoritmo complessivo, dato che la disambiguazione è a sua volta il collo di bottiglia dell'intero sistema.

Siamo fiduciosi che questo problema verrà risolto quanto prima dall'autore della libreria SenseRelate; in ogni caso esso non costituisce una limitazione grave per il sistema, dato che il passaggio della disambiguazione è eseguito sul server in una fase precedente e dunque i tempi di esecuzione non rappresentano tempi di attesa per l'utente finale.

In un sistema maturo, la disambiguazione potrebbe essere eseguita di tanto in tanto man mano che il crawler raccoglie nuovi dati relativi ai siti, in modo da aggiornare il database con dati pronti per una fruizione rapida.

4.6 L'albero

L'albero è la struttura dati fondamentale che memorizza la gerarchia all'interno della quale vengono organizzate le tag. Di fatto si tratta di un sottoinsieme dell'ontologia di iperonimia/iponimia dei sostantivi di WordNet, scelto in modo da contenere tutte le tag correlate a quella data e di essere agevolmente esplorabile.

Ogni nodo dell'albero rappresenta una parola di Wordnet in un particolare significato e contiene due attributi: il "peso" del nodo (attributo *own*) e il "peso" del sottoalbero sottostante (attributo *branch*).

Il nome identificativo di ogni nodo è costituito dalla parola seguita dal codice identificativo del synset WordNet a cui appartiene in quella partico-

lare accezione, nella forma standard di WordNet: "word#p#n" dove "word" indica la parola, "p" la *part of speech* (nel nostro albero essa vale "n" per ogni nodo, ovvero specifica che si tratta di sostantivi) e il numero finale rappresenta il *synset*. Per esempio, il nome del nodo "turkey#n#2" indica che si tratta della parola "turkey", nella sua seconda accezione di sostantivo, definita anche come "Republic of Turkey".

L'attributo *own* di un nodo è calcolato come il numero di siti etichettati con quella tag in quella particolare accezione; l'attributo *branch* invece è dato dalla somma degli attributi own di tutti i nodi sottostanti nell'albero, compreso il nodo stesso. Non tutti i nodi interni dell'albero corrispondono necessariamente a delle tag di del.icio.us, alcuni sono presenti solo in quanto iperonimi di qualche tag: il peso proprio di questi nodi è zero. Il peso proprio di un nodo foglia invece deve sempre essere positivo, così come il anche il peso di ogni sottoalbero.

L'attributo *own* è utile soprattutto nella fase di compressione dell'albero, in quanto indica quali nodi devono essere salvaguardati in quanto corrispondono a delle tag. L'attributo *branch* invece è utile per la fase di ordinamento dei rami. Il valore dell'attributo *branch* potrebbe essere ricavato calcolandolo volta per volta in base al valore dei nodi sottostanti; questa soluzione però richiederebbe un maggiore sforzo computazionale.

Poiché la profondità massima dell'albero di WordNet è limitata ad un valore ridotto (compreso fra 15 e 20 nodi), il numero di nodi del nostro albero può essere considerato funzione lineare del numero di tag. Questo dato sarà importante nel seguito per calcolare la complessità dei vari passaggi dell'algoritmo.

Per l'implementazione dell'albero ci siamo basati sulla libreria Perl Tree::DAG-Node, che mette a disposizione diverse funzioni per la gestione di alberi generici (la sigla DAG significa *Directed Acyclic Graph*).

4.6.1 Costruzione dell'albero

Algoritmo

Per ogni valore differente di tag disambiguata:

1. sia w il numero di occorrenze di quella tag corrispondenti a quel determinato synset;

2. crea un nuovo "ramo" contenente tutta la catena degli iperonimi della parola, fino alla radice dell'albero dei concetti di WordNet;

3. imposta a w il valore dell'attributo *branch* di ogni nodo del nuovo ramo e dell'attributo *own* del nodo foglia, a zero tutti gli altri;

4. partendo dalla radice e procedendo verso il basso, confronta ogni nodo del nuovo ramo con l'albero già esistente:

   (a) finché esiste un cammino corrispondente, seguilo ed esegui il *merge* del nuovo ramo, incrementando il valore dell'attributo *branch* di ogni nodo attraversato;

   (b) se incontri il nodo che ha un nome corrispondente alla tag da cui è stato creato il nuovo ramo, di questo nodo incrementa anche il valore *own* di w unità;

   (c) se invece da un certo nodo in poi non esiste più un cammino, aggiungi la parte rimanente del nuovo ramo come sottoramo dell'ultimo nodo corrispondente incontrato.

Per la costruzione della catena degli iperonimi al passo 2 si pone il problema delle possibili ereditarietà multiple; come discusso nel paragrafo 4.3 abbiamo deciso di considerare per ogni synset solo il primo iperonimo proposto da WordNet.

L'algoritmo illustrato ha il pregio di sfruttare al massimo i benefici offerti da una struttura dati ad albero. Si osservi che se per aggiungere le tag all'albero fossimo partiti dal basso, cercando fra tutti i nodi già esistenti un possibile genitore, avremmo dovuto eseguire a ogni passo un numero di confronti proporzionale al numero di nodi dell'albero e la complessità sarebbe risultata $O(n^2)$ (dove n, la dimensione dell'input, corrisponde al numero di tag disambiguate distinte).

Complessità

Vogliamo dimostrare che invece l'algoritmo di costruzione dell'albero che abbiamo descritto ha complessità lineare nel numero di tag disambiguate n. Per farlo introduciamo due assunzioni:

- la profondità dell'albero è limitata a un numero di livelli d;
- il fattore di ramificazione dell'albero è limitato a b figli per ogni nodo;

Entrambe le assunzioni sono ragionevoli in accordo con la struttura di WordNet, struttura che costituisce l'ossatura dell'albero e pone un limite intrinseco alle sue possibili dimensioni. I valori massimi di b e d in WordNet

sono nell'ordine di grandezza di 15-20, i valori medi più bassi. Il valore di b nel nostro albero, in particolare, è mediamente molto più basso.

Introdotte queste assunzioni si può osservare che per ogni distinta tag disambiguata devono essere eseguite solo operazioni di complessità costante. Infatti, in riferimento ai passi più pesanti dell'algoritmo abbiamo che:

- il passo 2 dell'algoritmo consiste in un massimo di d operazioni;
- il passo 4(a) deve essere ripetuto al peggio per tutti i nodi che costituiscono un cammino dalla radice a una foglia, quindi d volte; per ogni nodo deve essere effettuato un numero di confronti uguale al massimo al numero di figli b. In tutto quindi il numero totale di confronti che devono essere effettuati è $O(b * d)$.

Per gli altri passi dell'algoritmo non occorre dimostrazione poiché rappresentano in maniera evidente operazioni costanti.

Assumendo un valore limitato di b e d abbiamo dimostrato che la complessità dell'algoritmo è lineare nel numero totale n di tag disambiguate in input.

4.6.2 Compressione dell'albero

La compressione dell'albero è un passaggio essenziale per renderlo "esplorabile" da un utente umano. Come abbiamo osservato nella formulazione dei requisiti relativi all'usabilità dell'applicazione (paragrafo 3.3) la gerarchia deve risultare compatta e bilanciata [Blei et al., 2004], per facilitarne l'esplorazione. Abbiamo anche osservato descrivendo WordNet (nei paragrafi 3.4.2 e 4.3) che esso ha una granularità molto sottile, tanto che molte parole di uso comune si trovano a livelli di profondità superiore a 10 nella gerarchia; questo costituisce un grave appesantimento della navigazione.

Avendo a che fare con un sottoinsieme dell'albero della gerarchia di WordNet, dove ci interessa soprattutto rappresentare alcuni elementi, quelli coincidenti con le tag o con concetti utili per la loro organizzazione, la compressione dell'albero è un'operazione opportuna e necessaria per eliminare gli elementi superflui e rendere più visibili quelli importanti.

Questa operazione rappresenta un passaggio molto delicato e deve essere svolta in modo tale da ridurre la granularità di WordNet per rendere più leggera la navigazione, senza però cancellare i nodi essenziali per la struttura e senza appiattire eccessivamente la gerarchia.

Algoritmo

L'algoritmo di compressione si basa su due principi fondamentali:

- la rimozione delle categorie di alto livello di WordNet, che esprimono concetti troppo generali per essere di utilità per l'utente e nascondono i nodi sottostanti con alto contenuto informativo ;
- l'eliminazione di nodi intermedi non strettamente necessari per la struttura dell'albero e non corrispondenti a categorie utilizzate dagli utenti.

La compressione viene svolta in un'unica visita dell'albero, nella quale sono cancellati (ovvero rimpiazzati dai figli) tutti i nodi che soddisfano alcune condizioni:

- la parola corrisponda a una categoria di alto livello di WordNet, oppure
- il nodo abbia peso proprio nullo e
  - non abbia nessun fratello, oppure
  - abbia meno di k figli.

L'algoritmo esegue una visita in ampiezza dell'albero e testa le condizioni, volta per volta, sui figli del nodo visitato; in questo modo è possibile percorrere l'intero albero senza che si creino problemi quando un nodo viene cancellato. In una lista vengono salvati i figli del nodo corrente; quando uno di questi viene cancellato, viene anche eliminato dalla coda e sostituito dai propri figli. Quando tutti i figli del nodo corrente sono stati esaminati l'algoritmo viene eseguito ricorsivamente su tutti i nodi che si trovano nella coda.

Il valore di k è un parametro del programma, che può essere specificato; di default esso vale 2, ovvero vengono salvaguardati i nodi che hanno più di un figlio.

Per verificare se una parola corrisponde a una categoria di alto livello, essa viene confrontata con una lista, redatta manualmente una volta per tutte. La lista è un particolare abbastanza delicato, da una parte perché stabilisce una volta per tutte quali parole verranno sempre escluse dall'albero, dall'altra parte perché le parole più in alto nella gerarchia di WordNet che non vengono incluse nella lista hanno una buona probabilità di costituire il primo livello dell'albero finale presentato agli utenti. Nell'implementazione attuale la lista contiene 33 parole.

Le condizioni per la cancellazione di un nodo sono fatte in modo da ridurre quanto più possibile la profondità dell'albero, lasciando solo i nodi essenziali per l'esplorazione: i nodi che che hanno un solo figlio, o che sono figli unici, non sono indispensabili per la struttura dell'albero.

La seconda assunzione su cui si basano queste regole è che, quando una parola ha almeno una occorrenza come tag in questo contesto, è più facile che essa rappresenti un concetto di qualche interesse: quindi può essere utile, a maggior ragione, se si trova come nodo interno dell'albero.

Al contrario, le parole che non hanno neanche una occorrenza come tag sono spesso concetti dell'ontologia che hanno scarsa rilevanza nel contesto attuale; dunque se essi non sono indispensabili per la struttura dell'albero possono essere cancellati.

La scelta di procedere dall'alto verso il basso nella compressione, infine, porta a privilegiare i nodi che si trovano più in basso, ovvero i concetti più specifici, quelli con un contenuto informativo maggiore.

Queste scelte portano a qualche incoerenza e asimmetria nella struttura dell'albero, come sarà mostrato nel capitolo 5. Anche nodi con contenuto informativo molto specifico infatti possono "risalire" nell'albero, qualora tutti gli iperonimi siano eliminati nella compressione. Questo fenomeno però può riguardare solo nodi isolati, visto che le politiche di compressione non permettono di eliminare gli antenati qualora ci siano delle ramificazioni; senza che quel ramo venisse compresso, l'utente si troverebbe a dover discendere una gerarchia per incontrare un solo nodo foglia interessante. Per questo motivo questa soluzione risulta sensata, anche se può portare delle asimmetrie: è la soluzione che permette la massima efficacia, riducendo il numero di nodi inutili.

É proprio una caratteristica desiderata dell'algoritmo quella di "far salire" i nodi il più possibile, senza però appiattire troppo l'albero. Le condizioni di cancellazione dei nodi costituiscono un compromesso fra queste due esigenze: quella di ridurre la granularità dell'albero, e in particolare di ridurre al minimo il numero di livelli della gerarchia che devono essere percorsi per raggiungere un termine specifico, e di non appiattirlo eccessivamente. Entrambe le caratteristiche, un ramo troppo profondo o troppo piatto, sono da evitare per non appesantire o impoverire le possibilità di navigazione.

L'appiattimento è determinato dal parametro k, che di default abbiamo stabilito di lasciare al valore minimo di 2 (2 è da considerarsi il minimo valore non banale: con $k = 1$ infatti l'algoritmo non eliminerebbe nessun nodo, visto che gli unici nodi con meno di un figlio sono le foglie, che per come è strutturato l'albero hanno sempre valore dell'attributo $own >= 1$). Questa scelta preserva tutte le ramificazioni che costituiscono la struttura dell'albero ed evita i rischi di appiattimento eccessivo che renderebbe più difficile la navigazione; non c'è invece un limite fissato alla profondità: sottoalberi molto popolati e ramificati possono teoricamente raggiungere anche 6 o 7 livelli di profondità. Per avere un albero meno profondo è possibile scegliere un valore

maggiore per il parametro k, sacrificando parte della struttura dell'albero.

Complessità

L'algoritmo ha complessità lineare nel numero di nodi dell'albero, e quindi nel numero di tag in input, come l'algoritmo di costruzione dell'albero. Esso infatti consiste in una visita di tutti i nodi interni dell'albero non ancora compresso, dove per ciascuno di essi vengono testate le condizioni sui figli.

Detto b^d il numero di nodi, b^{d-1} è il numero di nodi interni. Il numero di operazioni che devono essere svolte è $O(b^{d-1} * b) = O(b^d) = O(n)$.

4.6.3 Ordinamento dei rami

L'ordinamento dei rami dell'albero è l'ultima operazione prima della stampa del risultato. La scelta di eseguire questa operazione è determinata dall'esigenza di mostrare prima i rami più pesanti, che si suppone siano i più interessanti, in accordo con uno dei requisiti dell'interfaccia utente definiti nel paragrafo 3.6.

Algoritmo

L'ordinamento viene eseguito sull'albero già compresso, con un algoritmo ricorsivo di visita in profondità dell'albero. Quando un nodo viene visitato, i suoi figli vengono riordinati in base al valore decrescente dell'attributo *branch*. Il peso di ogni singolo nodo non viene considerato in questo procedimento. Questa è una scelta naturale poiché parliamo di ordinamento dei rami e non dei nodi; un ordinamento dei rami, a partire dalla radice, basato sul peso dei singoli nodi, non avrebbe senso.

L'albero ordinato è in un certo senso un albero "ibrido": la struttura è determinata solo dalle relazioni semantiche fra le tag rappresentate, ma l'ordine dei rami dipende dai dati di frequenza delle tag.

Complessità

La complessità dell'ordinamento potrebbe risultare a una prima analisi essere $n * \log n$, dove n è il numero di nodi dell'albero.

Osserviamo però che non si tratta di ordinare l'intero albero confrontando fra loro tutti gli elementi, ma solo di ordinare fra loro i figli di ogni singolo nodo interno dell'albero: il numero totale dei figli dei nodi interni coincide con il numero dei nodi dell'albero (per precisione meno una unità, corrispondente alla radice) ma il numero di confronti che devono essere eseguiti fra

di essi è molto inferiore a quello che risulterebbe nel caso in cui si volessero ordinare tutti gli elementi, come se fossero in un array per esempio.

Infatti ogni nodo deve essere confrontato, al più, con il fattore massimo di ramificazione (ovvero con il numero massimo di figli che può avere un nodo). Il caso peggiore è quello di avere un albero costituito da un solo nodo interno, la radice, padre di tutti gli altri nodi, che sono delle foglie. Poiché la struttura dell'albero non pone limitazioni precise sul numero massimo di figli che può avere ogni nodo, il caso peggiore è teoricamente possibile. Tuttavia, se assumiamo che l'albero abbia un fattore di ramificazione massimo di un numero fissato b, l'algoritmo può essere visto come equivalente all'operazione di ordinare b elementi, ripetuta i volte, dove i sia il numero dei nodi interni dell'albero. Con questa assunzione, che sembra del tutto ragionevole in base alla struttura di WordNet e dell'algoritmo di compressione, anche l'ordinamento dei rami dell'albero risulta essere un'operazione lineare nel numero di tag disambiguate.

Infatti, detti d la profondità massima dell'albero e b il fattore di ramificazione massimo, abbiamo che:

- il numero dei nodi totali dell'albero è $n <= b^d$,
- il numero di nodi interni dell'albero è $i <= b^{d-1}$.

L'operazione di ordinare b elementi ha complessità $b \log b$ e deve essere eseguita i volte. Dunque la complessità dell'algoritmo è:

$$O(b^{d-1} b \log b) = O(b^d \log b) = O(b^d) = O(n),$$

ovvero è lineare nel numero di nodi dell'albero.

Questo risultato è particolarmente importante perché garantisce che la complessità di tutte le operazioni eseguite dal Web server sia lineare rispetto alla dimensione dell'input.

L'output

Una volta che l'albero Perl è stato costruito e compresso, e che i suoi rami sono stati ordinati, esso viene stampato, con una procedura ricorsiva *depth-first*, in formato XML o HTML.

Per ogni nodo sono stampate le seguenti informazioni:

- l'identificativo del nodo, ovvero la parola e il synset di appartenenza;
- il valore dell'attributo *own*.

L'albero HTML inoltre contiene anche, per ogni nodo:

- la definizione di WordNet del *synset* a cui appartiene la tag, nel campo *title* del nodo;
- un link alla pagina di del.icio.us relativa alla tag;
- un link alla pagina di del.icio.us relativa all'intersezione della tag con la tag principale.

Questi dati sono quelli necessari per soddisfare i requisiti dell'interfaccia utente definiti nel paragrafo 3.6, e in particolare i requisiti di contenuto e di navigazione.

Se il parametro $HTML\_page$ ha valore falso, solo la struttura dell'albero viene stampata; questa è la scelta di default. Se invece il parametro è impostato con valore vero (esso può essere specificato nella richiesta HTTP), il Web server inserisce la struttura dell'albero in una pagina HTML autonoma (minimale).

In questo modo è resa possibile attraverso un qualsiasi browser la visualizzazione dell'albero intero in una pagina a sé stante, completo di link alle pagine di del.icio.us.

La stampa del risultato consiste in una serie di operazioni di complessità costante, ripetute per ogni nodo dell'albero: la complessità è dunque lineare nel numero di nodi, che a sua volta è lineare nel numero di tag disambiguate.

4.6.4 Osservazioni generali sulla complessità

Nelle sezioni relative agli algoritmi che concorrono alla realizzazione dell'albero abbiamo dimostrato come ciascuno di essi abbia complessità al più lineare nel numero di tag disambiguate distinte n; possiamo dunque concludere che questa è la complessità dell'intero procedimento di creazione dell'albero delle tag. Questo dato è particolarmente importante perché determina i tempi di risposta del Web server.

Il valore di n, che rappresenta la dimensione dell'input, ha un limite superiore che dipende da due parametri dell'applicazione:

- il numero massimo n_sites di siti, relativi alla tag principale, che devono essere considerati;
- il numero massimo n_tags di tag che devono essere considerate per ogni sito;

In relazione a questi questi due parametri e al valore $n\_synsets$, che rappresenta il numero massimo di diversi *synset* per ogni tag, la dimensione massima dell'input è $n\_sites * n\_tags * n\_synsets$. Quest'ultimo numero tuttavia è trascurabile in quanto è una costante e ha un valore ridotto (nel caso peggiore un sostantivo può appartenere a circa 10 *synset*, ma la maggior parte dei sostantivi appartengono a uno solo *synset*, e la media è di 1.23[2]).

Possiamo dunque affermare che la dimensione massima dell'input dipende dal prodotto $n\_sites * n\_tags$. Sappiamo che il primo parametro può valere al massimo 10000, per il limite imposto da del.icio.us nella consultazione delle pagine relative a una tag, mentre per il secondo non c'è un limite a priori, se esso non è specificato. I valori di default del programma sono 500 per il primo parametro e 20 per il secondo.

4.7 Lo script Greasemonkey

Lo script JavaScript per Greasemonkey è fatto per integrare le informazioni ottenute dal Web server nella pagina di del.icio.us relativa a una tag, creando un menù espandibile attraverso il quale possa essere visitato l'albero delle tag correlate. Esso è stato progettato e realizzato in accordo coi requisiti dell'interfaccia di navigazione definiti nel paragrafo 3.6.

Lo script viene attivato quando l'utente richiede al browser una pagina dalla forma http://del.icio.us/tag/«tag»; esso per prima cosa estrae la tag dall'url della pagina richiesta e richiede a sua volta al Web server Perl l'albero semantico delle tag correlate, in formato HTML. Per effettuare la richiesta HTTP si serve della funzione predefinita di Greasemonkey GM_xmlhttpRequest.

Gli altri parametri con cui viene invocato il Web server sono quelli di default, ma possono essere cambiati facilmente modificando le prime righe del codice JavaScript. Questo passaggio deve essere eseguito manualmente perché Greasemonkey attualmente non possiede un'interfaccia grafica per gestire le opzioni degli script.

Lo script agisce modificando dinamicamente la pagina Web; questo significa anche che essa può essere comunque caricata normalmente dal browser. Mentre lo script è in attesa della risposta da parte del server perl la pa-

[2]Le statistiche sulla polisemia delle parole sono disponibili sul sito di WordNet all'indirizzo http://wordnet.princeton.edu/man/wnstats.7WN

gina, se è già stata trasmessa dal server di del.icio.us, viene visualizzata normalmente, soltanto ancora priva delle nuove informazioni.

Quando il Web server risponde, lo script crea una barra laterale nella pagina di del.icio.us e inserisce al suo interno l'albero HTML. La barra viene creata come un nuovo elemento HTML con identificativo di tipo "sidebar", sfruttando la definizione delle barre laterali di del.icio.us. In questo modo alla nuova barra laterale viene associato automaticamente il relativo foglio di stile standard predefinito di del.icio.us.

Questo garantisce la massima coerenza grafica del nuovo elemento; un restyling grafico del sito del.icio.us verrebbe supportato in modo automatico dallo script, che non richiederebbe di essere aggiornato. Esso infatti non interviene su singoli aspetti relativi alla grafica, ma solo sulla struttura della pagina, sfruttando il disaccoppiamento fra dati di struttura (HTML) e grafici (CSS) su cui si basa il sito e senza interferire con questi ultimi.

Lo stile della barra laterale viene solo leggermente modificato localmente, per permettere la creazione di un menù gerarchico espandibile. Fondamentalmente vengono ridefiniti alcuni parametri relativi ai margini, che vengono ampliati, e vengono impostate le caratteristiche essenziali di stile per i nuovi tipi di elementi che devono essere definiti.

Il menù espandibile viene realizzato modificando il codice HTML relativo ai nodi interni dell'albero, per aggiungere nuovi campi, necessari per la gestione dell'interazione. In particolare a ogni nodo interno sono aggiunti due elementi come link fittizi, visualizzati come "[+]" e "[-]"; il click del mouse su questi elementi viene associato alle funzioni locali che permettono rispettivamente di espandere o di collassare i singoli nodi, ovvero di mostrarne o nasconderne i figli. Per ogni nodo viene mostrato solo l'elemento "[+]" se esso può essere espanso, solo "[-]" nel caso contrario. L'elenco dei nodi e del loro stato attuale è memorizzato in un array locale.

Gli unici elementi che vengono mostrati da subito nella barra sono i nodi di primo livello dell'albero. Questa scelta è stata determinata dall'esigenza di non appesantire eccessivamente la pagina; gli altri nodi possono essere visitati man mano che l'utente espande i rami dell'albero che gli interessano.

Poiché l'albero restituito dal server è già ordinato in base all'attributo *branch* dei rami, i nodi più "pesanti" nella gerarchia, quelli che contengono come iponimi le tag maggiormente correlate, si troveranno più in alto nella barra; questo vale anche per l'ordinamento dei sottorami al di sotto di ogni nodo, a tutti i livelli dell'albero.

I campi *title* dei vari elementi HTML creati sono riempiti in modo tale

che quando l'utente passa col mouse sopra un elemento, il browser mostri in un *tooltip* le informazioni relative ad esso e alle possibilità di navigazione che offre. Per ogni parola corrispondente a un nodo dell'albero, il browser mostrerà la definizione di WordNet del synset a cui essa appartiene; per i nodi che contano almeno una risorsa correlata, verrà mostrata anche la destinazione del link alla pagina dell'intersezione fra quella tag e la parola chiave principale.

L'intera nuova barra laterale può essere nascosta con un click su un apposito pulsante, che attiva la relativa funzione JavaScript.

Abbiamo realizzato anche uno script JavaScript che esegue le stesse operazioni di creazione del menù espandibile sulla pagina HTML contenente solo l'albero. In questo modo l'albero delle tag correlate può essere esplorato anche in modo interattivo in una pagina a sé stante, restituita dal Web server, al di fuori dell'interfaccia Web di del.icio.us.

Questa soluzione non rappresenta la modalità principale di navigazione per cui il sistema è stato progettato, ma abbiamo scelto di prevederla comunque in quanto essa non necessita dell'installazione di Greasemonkey e costituisce una possibilità di fruizione dell'albero per un'utenza più ampia: in questo modo infatti è sufficiente un qualsiasi browser che supporti codice JavasScript.

Capitolo 5

Risultati

In questo capitolo riporteremo e discuteremo i risultati che abbiamo ottenuto, analizzando alcuni aspetti del sistema realizzato e sperimentandolo in diversi modi.

Poiché si tratta di un'applicazione rivolta agli utenti e finalizzata a migliorare le possibilità di navigazione, l'unico modo efficace di presentare dei risultati quantitativi sarebbe quello di sottoporre il sistema a un certo numero di utenti per testarne l'utilità, confrontandolo eventualmente con altre interfacce. Trattandosi di un progetto che presenta caratteristiche originali, anche per il tipo di navigazione proposta, e di un'applicazione che non può considerarsi matura, abbiamo ritenuto che possa essere sufficiente mostrare dei risultati qualitativi, che illustrino attraverso alcuni esempi il funzionamento del sistema e le possibilità di interazione che esso offre, i punti di forza e le limitazioni a cui è soggetto.

5.1 L'interfaccia utente

In figura 5.1 è mostrata la pagina che appare all'utente quando, dopo aver installato lo script per Greasemonkey, visita la pagina di del.icio.us relativa alla tag "pasta".

Aprendo la pagina sono visibili solo i nodi di primo livello della gerarchia: espandendoli l'utente può visitare le varie aree semantiche; essi costituiscono in qualche modo dei facet. Questi nodi, le categorie che appaiono per prime alla vista dell'utente, sono ordinati per importanza; l'utente è così orientato nella visita dell'albero da due criteri: quello dell'importanza dei rami che va a esplorare e quello dell'ambito semantico a cui appartengono.

Figura 5.1: la pagina di del.icio.us relativa alla tag "pasta". La barra laterale più interna contiene l'albero semantico delle tag correlate, elaborate dal Web server che abbiamo realizzato. L'albero è basato su 300 siti correlati.

Figura 5.2: l'immagine raffigura il tooltip che compare passando con il mouse sopra il nome di un nodo dell'albero e che mostra la definizione della parola (in questo caso è mostrata la definizione dei "tortellini").

Il primo nodo è spesso quello che contiene nel proprio sottoalbero la tag oggetto dell'esplorazione: essa infatti, comparendo per definizione in tutti i siti esplorati, è quella col peso più alto; inoltre spesso tag molto correlate rappresentano parole "simili", che si trovano nello stesso ramo. Questo è il caso che si osserva in figura 5.1 rispetto alla tag pasta: il primo ramo contiene un'ontologia del cibo, nel quale il ramo a sua volta più importante è quello che comprende i vari tipi di pasta; accanto si trovano altri tipi di cibi e di ingredienti.

Ogni nodo dell'albero consente all'utente diverse possibilità di interazione, che soddisfano i requisiti specificati nel paragrafo 3.6:

- se è un nodo interno si possono espandere o nascondere i rami sottostanti, a seconda della situazione, cliccando rispettivamente sull'elemento '[+]' o '[-]' (oltre al simbolo intuitivo e convenzionale del "+" o del "-", anche un tooltip indica l'azione associata, se si passa col mouse sopra l'elemento);
- passando col cursore sopra il nome del nodo, viene mostrata in un *tooltip* la definizione di WordNet della parola, in quel particolare significato, come mostrato in figura 5.2: questo è utile sia nel caso di concetti sconosciuti all'utente presenti nella gerarchia, sia per aiutare a orientarsi nella navigazione (comprendere il significato specifico con cui è inteso un termine può essere importante per decidere quando vale la pena di esplorare un nodo e quando no);
- ogni nodo è anche un link alla pagina di del.icio.us della tag relativa;

Figura 5.3: l'immagine mostra il tooltip che compare passando col mouse sopra il numero fra parentesi, di fianco al nome del nodo ("broccoli"), e che indica la destinazione del link raggiungibile cliccando su questo elemento: la pagina dell'intersezione fra la parola chiave principale corrente esplorata nella pagina ("pasta") e quella del nodo. Il numero fra parentesi indica il numero di risorse correlate a entrambe le tag.

- per ogni nodo a cui sia associata almeno una risorsa correlata, è presente un link alla pagina dell'intersezione con la tag principale, come mostrato in figura 5.3: il link mostra il numero di risorse correlate nell'insieme analizzato. La presenza di questo link è fondamentale per garantire la possibilità di raffinare la ricerca scegliendo un ambito specifico.

Le informazioni su quali nodi sono espansi e quali no è persistente: se si nasconde il ramo sottostante a un nodo e poi lo si riespande, i sottolivelli compariranno eventualmente già espansi, a loro volta, se lo erano nel momento in cui il ramo è stato nascosto.

La presenza della definizione di WordNet può essere particolarmente utile nel caso di parole con più significati. Come si vede in figura 5.1, la parola "turkey", che abbiamo spesso usato come esempio di polisemia, compare come iponimo di "poultry", insieme a "chicken". Non compare invece in questo albero come iponimo di "country": tutte le occorrenze, dato il contesto, sono state interpretate come corrispondenti al senso di "tacchino".

La presenza di link per accedere all'intersezione delle tag rappresentate con la parola chiave principale, oggetto dell'esplorazione, è fondamentale per permettere il raffinamento della ricerca in una sottoarea dello spazio definito dalla parola chiave principale. Il numero riportato fra parentesi, che dà il nome al collegamento, mostra la dimensione di questa sottoarea nei dati di supporto; la dimensione ovviamente può essere molto maggiore date le dimensioni di del.icio.us rispetto ai dati considerati, ma il calcolo delle cooccorrenze fra gli ultimi n siti etichettati con la parola richiesta è una "unità di misura" coerente, che può essere indicativa. Essa può essere più

o meno rappresentativa a seconda del valore di n (coincidente con il parametro n_sites), della dimensione della parola chiave principale e della sua eterogeneità e variabilità nel tempo; in ogni caso i dati più recenti sono generalmente i più interessanti, e sono anche quelli che sono mostrati per primi dall'interfaccia di del.icio.us, dunque sono quelli che corrispondono effettivamente alle risorse verso cui sono diretti i link (di default, del.icio.us mostra solo i primi 10 siti in ogni pagina).

Dal punto di vista grafico si può osservare la coerenza del nuovo elemento, che si integra perfettamente nella pagina e nello stile di del.icio.us.

Il disaccoppiamento fra presentazione e contenuti permette anche di cambiare l'aspetto grafico della pagina senza che il nuovo elemento crei problemi. In figura 5.4 è mostrata una pagina di del.icio.us dopo che oltre al nostro script per Greasemonkey ne è stato installato anche un altro, che agisce parallelamente[1]: esso applica al sito di del.icio.us un nuovo foglio di stile, con una grafica "autunnale"; come si vede, la barra si integra perfettamente anche nel nuovo stile.

É possibile nascondere la nuova barra laterale, cliccando sull'apposito pulsante. In questo caso la pagina appare pulita, quasi come se lo script non fosse installato, come si vede in figura 5.5: è solo visibile il pulsante col quale la barra può essere nuovamente mostrata.

5.2 Esempi di uso

Abbiamo testato il comportamento del sistema con varie parole chiave di partenza, differenti per ambito, specificità e popolarità come tag, e con diversi valori dei parametri, in particolare quelli che stabiliscono la dimensione dell'input.

L'esempio mostrato in figura 5.1 e utilizzato per illustrare l'interfaccia di navigazione è stato realizzato in base all'esplorazione di 300 siti e con un valore del parametro n_tags non specificato (mantenendo cioè la "lunga coda").

Come si vede, la nuova barra laterale permette per esempio, nel primo ramo dell'albero, di visualizzare vari tipi di pasta, che sono degli iponimi della tag principale, nel primo ramo dell'albero. É importante osservare come la nuova interfaccia mostri tutti i tipi di pasta che sono presenti nella

[1] Lo script è disponibile nel repository degli user script di Greasemonkey alla pagina http://userscripts.org/scripts/show/1755

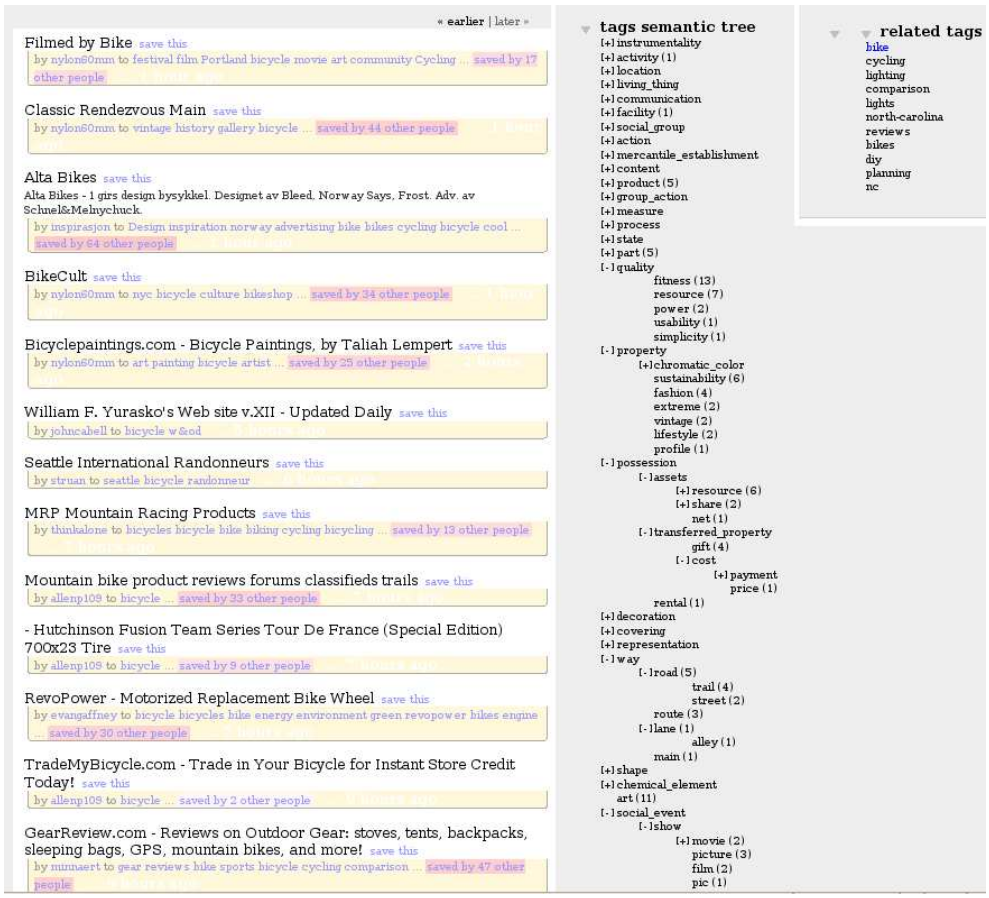

Figura 5.4: la pagina di del.icio.us relativa alla tag "bicycle", quando oltre allo script per la visualizzazione dell'albero semantico delle tag correlate ne è installato anche uno che definisce un nuovo foglio di stile per del.icio.us

Figura 5.5: la pagina di del.icio.us relativa alla tag "guitar", quando la barra laterale aggiuntiva è stata nascosta dall'utente.

folksonomia, o almeno nell'insieme di risorse considerate, in questo caso i 300 siti più recenti salvati. Essi sono mostrati in ordine di importanza, ovvero in base al numero di risorse correlate a ciascuno. Se confrontiamo questo risultato coi suggerimenti di del.icio.us, osserviamo che solo un tipo di pasta è presente, il più diffuso nell'intera folksonomia. Per gli altri non c'è spazio. Un utente che non sappia dell'esistenza di un tipo di pasta chiamato "tortellini" non ne troverà traccia, a meno che non si imbatta casualmente in un sito etichettato con quella parola, nella pagina relativa alla tag "pasta".

Nella nuova barra laterale invece l'utente non solo può visualizzare ogni tipo di pasta a cui siano correlate delle risorse nella folksonomia, individuandolo intuitivamente come sottoclasse della parola chiave principale, ma può anche scoprirne la natura, grazie al *tooltip* che mostra la definizione di WordNet della parola e ovviamente può saltare alla pagina relativa a quella tag.

In questo caso non è particolarmente interessante la possibilità di andare alla pagina relativa a entrambe le tag combinate, poprio perché si tratta di un iponimo stretto della tag principale. É interessante invece la possibilità di visitare le pagine dell'intersezione con le tag corrispondenti ad altri cibi, come mostrato in figura 5.3, per esempio che possano essere cucinati insieme. Non a caso il ramo delle verdure compare per secondo, subito dopo quello della pasta, nel sottoalbero del cibo. Le verdure sono a loro volta ordinate per importanza: le prime sono il ramo del peperoncino e i broccoli, che evidentemente compaiono molto spesso insieme alla pasta.

Si osservi che sono presenti sia la parola "chili" sia "chilli", sinonimi, ed entrambi iponimi di "hot pepper". Questo rappresenta un importante aiuto per affrontare il problema dell'assenza di controllo dei sinonimi, tipico di una folksonomia. L'utente che sia interessato alle ricette piccanti di pasta è indirizzato sia alla pagina che combina la tag "pasta" con "chili", sia a quella che la combina con "chilli"; se vuole avere una visione completa del sottospazio determinato da questi due concetti può visitare entrambe le pagine. In assenza di un vocabolario controllato, la gerarchia pone in qualche modo un rimedio.

Abbiamo osservato che spesso il primo nodo è quello corrispondente al ramo che contiene la parola chiave principale. Non è il caso del risultato ottenuto per la tag "vacation" con 400 siti: il primo nodo è "location". Espandendo questo ramo si ottiene il risultato mostrato in figura 5.6 (dove per motivi di spazio non sono stati espansi i nodi più corposi, come "country", "town" o "american state").

Tramite questa gerarchia è possibile esplorare tutte le tag corrispondenti

Figura 5.6: la pagina di del.icio.us relativa alla tag "vacation" dove è stata espansa l'ontologia geografica, sotto il nodo "location". L'albero è stato costruito coi dati relativi a 400 siti esplorati.

Figura 5.7: la pagina di del.icio.us per la tag "recipe"; nella barra laterale aggiuntiva sono visibili due nodi "food", che corrispondono a due diverse accezioni del termine, e sotto i quali si sviluppano due rami differenti della gerarchia. L'albero è stato costruito coi dati relativi a 500 siti esplorati

a località o aree geografiche che sono state utilizzate per un sito etichettato anche con "vacation". La gerarchia mostra anche tag utilizzate da un solo utente, come "everest": permette di esplorare la "lunga coda" dei posti marcati da altri utenti, almeno per quanto riguarda i dati più recenti.

Per la costruzione dell'albero non è stato fissato un valore per il parametro di n_tags.

La figura 5.7 mostra il risultato ottenuto con la tag "recipe", con 500 siti esplorati e il valore del parametro n_tag per tagliare la *lunga coda* non specificato. Si può osservare che ai primi posti nella gerarchia risultante si trovano due nodi, entrambi etichettati "food". Passando col mouse sopra ciascuno dei due il browser mostra le relative definizioni:

- *any solid substance (as opposed to liquid) that is used as a source of nourishment; food and drink.*
- *any substance that can be metabolized by an organism to give energy and build tissue;*

La differenza fra i due rami appare evidente soprattutto osservando i nodi sottostanti: nel sottoalbero corrispondente al primo significato di "food", quello più concreto, si trovano parole come "bread" e "muesli"; sotto al secondo parole più astratte, come "dish" e "meal", con le relative sottogerarchie. L'utente medio comprende immediatamente il criterio su cui basa la suddivisione dei due rami dopo che li ha espansi di un livello e sa così in quale dei due orientarsi per proseguire l'esplorazione.

La granularità sottile di WordNet, che distingue fra due significati diversi di "food" si rivela molto utile in questo caso per definire due rami essenziali di una gerarchia; se essi fossero mischiati insieme i concetti sarebbero tutti mostrati in uno spazio più piatto e sarebbe meno agevole la ricerca.

La parola "food", come tag, è stata utilizzata 180 volte in un contesto in cui è stata riconosciuta appartenere al primo significato, 7 volte al secondo. In questo caso la differenza non è così rilevante, perchè i due nodi sono risultati fratelli nella gerarchia e i link sono diretti alle stesse pagine.

Una caratteristica della struttura "ibrida" rappresentata nella barra laterale aggiuntiva è la asimmetria, dovuta al processo di compressione: nell'albero si trovano talvolta affiancate categorie di livello diverso: è il caso di "restaurant" che compare come nodo di primo livello nell'albero relativo alla tag "recipe" mostrato in figura 5.7, anche se si trova più in basso nella pagina, rispetto all'area mostrata nell'immagine. La tag "restaurant" in questo caso costituisce un nodo di primo livello e allo stesso tempo un nodo foglia, e si trova si trova di fianco a concetti più generali, come "food".

Questo fatto è una conseguenza del funzionamento dell'algoritmo di compressione. Esso infatti è stato progettato per eliminare tutti i concetti che non siano essenziali per la struttura dell'albero.

Nel caso in questione, evidentemente, tutta la catena degli iperonimi di "restaurant" è stata eliminata nel processo di compressione dell'albero, perchè non c'erano altre tag mappate in questo ramo della gerarchia di WordNet. Sarebbe stato inutile riportare per esempio l'iperonimo "building", non essendoci altri tipi di edifici nell'albero.

Anche se da un certo punto di vista può rappresentare un elemento di incoerenza e un impoverimento (si perde l'informazione esplicita che un "restaurant" è un tipo di "building"), questa struttura flessibile risulta il modo

più efficace per mostrare tutte le tag dell'insieme analizzato facendole rientrare nell'albero, rendendo allo stesso tempo quest'ultimo il più compatto e leggero possibile.

Per testare il sistema con la dimensione massima dell'input, $n\_sites = 10.000$, abbiamo usato la tag blog, che è una delle più popolari in del.icio.us (per la precisione, è seconda solo a "design", dato curioso, legato alla "deformazione professionale" di molti utenti, molto attivi, della folksonomia) e anche una delle più generali, poichè ci sono blog che trattano qualsiasi argomento: questo esperimento somiglia quasi a una mappatura dell'intero spazio delle tag su WordNet. Per la compressione dell'albero abbiamo impostato il parametro n_tags a un valore di 20, per limitare gli effetti della *long tail* su una quantità già elevata di dati.

Un problema che questo esempio evidenzia è quello del numero elevato di nodi di primo livello risultanti, quasi un centinaio, numero che pregiudica la comodità di uso dell'interfaccia: questo risultato era prevedibile ed è dovuto al fatto che l'algoritmo di compressione cancella le categorie di alto livello per mostrare in alto i concetti più interessanti, ovvero con un contenuto più specifico; l'algoritmo di compressione è stato calibrato per insiemi di dati di dimensioni inferiori, o meno eterogenei. Questo esempio rappresenta in qualche modo un caso limite, dove l'applicazione non risulta avere una scalabilità adeguata a presentare tutti i contenuti in una forma organica e cade nel problema di presentare uno spazio troppo piatto.

Il fatto che i rami siano ordinati per importanza però dà comunque un valore significativo al risultato: trascurando i rami che vengono mostrati in fondo alla pagina, la lunga coda, e concentrandosi sui primi, è possibile esplorare le aree semantiche principali correlate alla tag. In figura 5.8 è possibile vedere i nodi più importanti di primo livello risultanti e, espansa in parte, la gerarchia dei contenuti e delle discipline scientifiche.

Il fatto che una tag non compaia in WordNet non pregiudica la possibilità di esplorarla: in figura 5.9 è mostrato il risultato ottenuto con la tag "Greasemonkey", non presente in WordNet.

Questo test permette anche di sperimentare l'applicazione nell'ambito specifico del software, quello più presente in del.icio.us e non ben coperto da WordNet, dal momento che molti termini sono di introduzione recente (o almeno, il loro significato relativo a questo ambito).

I risultati sono interessanti: il ramo più importante, nell'albero, è quello di "written communication", sotto al quale si colloca la parola "software", insieme alle altre che si vedono nella figura. La parola "documentation",

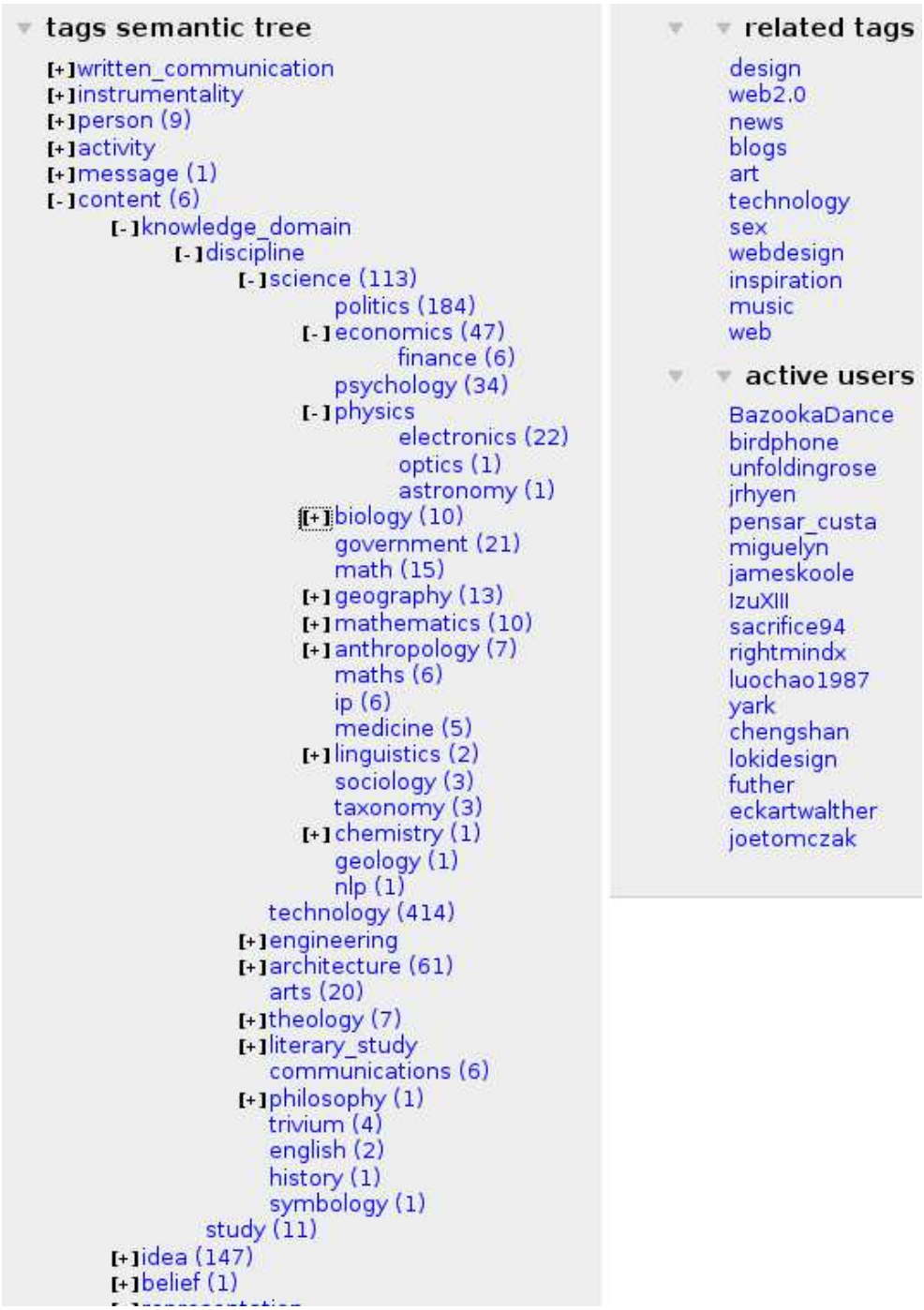

Figura 5.8: le due barre laterali nella pagina di del.icio.us relativa alla tag "blog". L'albero è stato costruito coi dati relativi a 10.000 siti esplorati

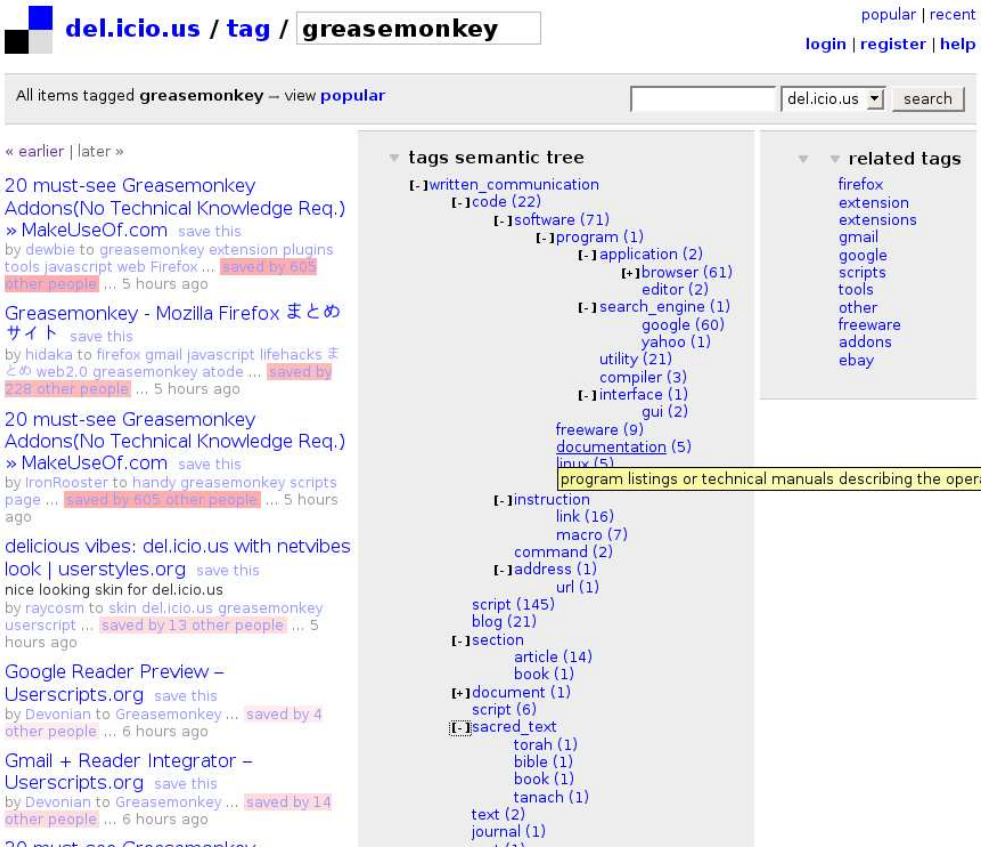

Figura 5.9: la pagina di del.icio.us relativa alla tag "greasemonkey"; le tag maggiormente correlate non sono presenti nella barra laterale aggiuntiva perchè non sono parole riconosciute da WordNet.

che per WordNet ha tre significati, di cui quello in ambito software non è il primo, viene interpretata come sempre appartenente a questa categoria. La parola "script", invece, non è conosciuta da WordNet nel suo significato nel campo dell'informatica, ed è dunque interpretato nell'accezione di "copione".

Ci sono però diverse osservazioni da fare: in primo luogo il numero di siti esplorati è molto più alto di quello dei siti effettivamente utilizzabili. Esplorando oltre 2000 siti il crawler ne ha ottenuti solo poco più di 200 associabili ad almeno una parola di WordNet. Se usando come parola chiave di partenza una parola contenuta in WordNet c'è la certezza che ogni sito esplorato abbia almeno una tag contenuta nel lessico (la parola chiave stessa), per come è strutturato il programma, in questo caso invece non è così. Per esempio, tutti i siti etichettati solo con la tag Greasemonkey, o anche con "Greasemonkey", "Firefox" e "JavaScript" vengono scartati dal crawler che non li inserisce nella base di dati.

Abbiamo un dato positivo: gli elementi riconosciuti da WordNet come appartenenti all'ambito del software sono organizzati in una forma che risulta coerente ed efficace, come si vede nella figura.

Se confrontiamo le parole contenute nelle due barre laterali, osserviamo però che delle 11 tag suggerite da del.icio.us nessuna è presente anche nel ramo "softawre" dell'albero: parole come "gmail" non sono state riconosciute, mentre parole come "extension" o "script" sono state interpretate con altri significati.

Questa situazione non pregiudica le possibilità di ottenere risultati migliori: le parole più largamente utilizzate in ambito informatico verrano certamente integrate in una prossima versione di WordNet. Per esempio, WordNet riconosce già la sigla "ie" come browser per "Internet Explorer", e anche "Netscape", ma non conosce "Firefox", che rappresenta un prodotto più recente. Per quanto riguarda parole come "script" ed "extension", se WordNet le contenesse anche nella loro accezione in campo informatico esse sarebbero probabilmente state disambiguate correttamente.

Una soluzione efficace che potrebbe contrastare in buona misura il problema sarebbe quella di integrare un'ontologia di dominio con il relativo ramo di WordNet. Vista la rapidità con cui nuove parole e nuove accezioni di parole esistenti sono create in questo ambito, però, alcune parole resterebbero comunque escluse o non riconosciute correttamente.

In figura 5.10 è mostrato il risultato ottenuto per la tag "health" esplorando 1000 siti e senza specificare il parametro n_tags.

A uno sguardo attento le related tag suggerite da del.icio.us si rivelano essere tipici contenuti di spam, il probabile risultato di azioni di gaming.

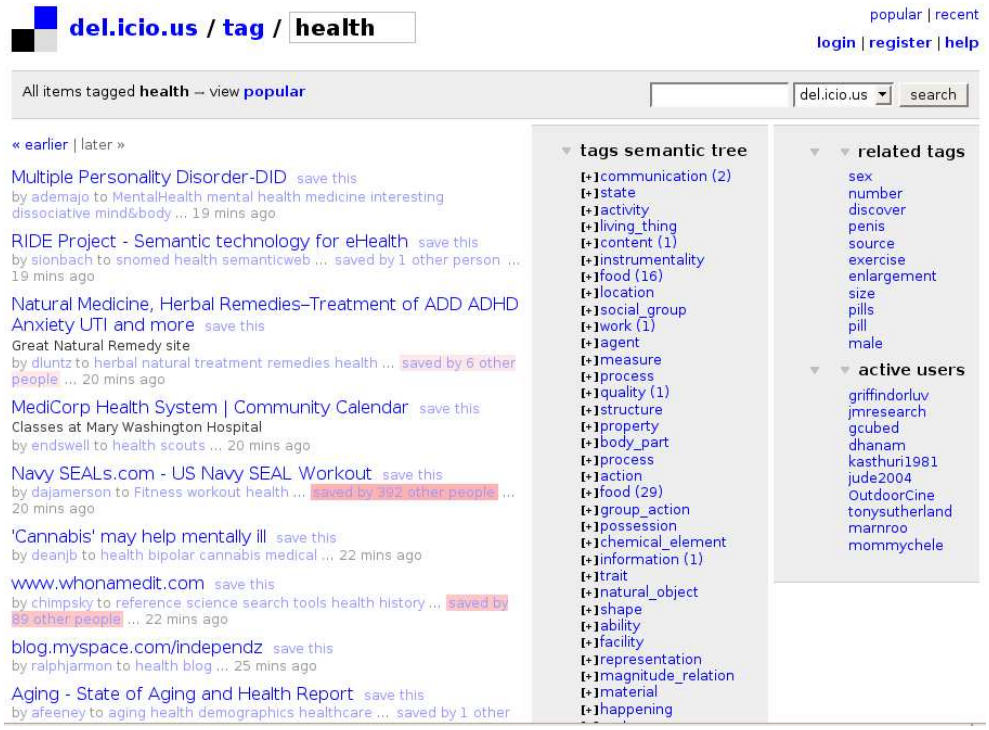

Figura 5.10: la pagina di del.icio.us relativa alla tag "health"; se si osservano le due barre laterali, si nota che quella delle related tags suggerite da del.icio.us è inquinata dallo spam. L'albero è stato costruito coi dati relativi a 1000 siti esplorati

Queste tag non sono assenti dall'albero presentato nella nuova barra laterale, ma in questa sono circoscritte nelle loro rispettive aree semantiche, dunque non riescono a "guastare tutto" come nelle related tags di del.icio.us.

5.3 Confronti con altri sistemi

Abbiamo osservato come i nodi di primo livello dell'albero costituiscano qualcosa di analogo a un insieme di facet; un confronto con applicazioni Web basate su facet, come Epicourious, porta sicuramente ad emergere una maggiore accuratezza di queste ultime: questo fatto non deve meravigliare, se si considera che questo tipo di sistemi si colloca in contesti limitati, e si basa su metadati creati ad hoc. Lo stesso si può dire anche per quanto riguarda le interfacce ottenute con gli algoritmi semiautomatici del progetto Flamenco [Stoica and Hearst, 2004]: questi algoritmi sono fatti per lavorare in contesti specifici e prevedono un intervento di supervisione umana, per filtrare i risultati e produrre dei facet che abbiano valore per un determinato insieme di risorse.

Al contrario, il sistema che abbiamo realizzato offre risultati più grezzi e di qualità spesso inferiore, ma che comunque mostrano di poter costituire un supporto importante alla navigazione; la minore accuratezza del risultato è il prezzo naturale da pagare per una soluzione che offra le seguenti caratteristiche:

- non sia legata a un contesto limitato, ma abbia applicabilità generale;
- si basi su un algoritmo del tutto automatico;
- lavori in tempo reale.

Per quanto riguarda il primo punto, la validità generale dei risultati è garantita, almeno in linea di massima, dall'uso di WordNet, che propone categorie generali basate sulle strutture essenziali del linguaggio. I criteri di classificazione proposti non sono sempre quelli più adatti, come sarebbe impossibile garantire con un'ontologia definita a priori, ma hanno caratteristiche di generalità e di immediatezza che li rendono validi e utili in molti casi.

Riguardo al terzo punto, l'applicazione realizzata si è dimostrata in grado di rispondere in tempi trascurabili anche per grandi dimensioni dell'input, ammesso che abbia a disposizione alcuni dati preelaborati; questa problematica è discussa più specificamente nel paragrafo 5.4.

Una volta che queste tre condizioni siano soddisfatte, questo tipo di soluzione ha il vantaggio di poter essere definito in modo generale e applicato

volta per volta in contesti differenti, pur senza garantire in tutti i casi la qualità del risultato.

Considerazioni simili valgono nel confronto di questa soluzione con fac.etio.us, il tool descritto nel paragrafo 2.4.1, che si colloca nello stesso contesto, poichè permette di navigare fra i siti di del.icio.us organizzandoli in facet. Fac.etio.us presenta il vantaggio di mostrare categorie di alto livello di buona qualità e che si rivelano spesso utili; queste categorie però sono fissate una volta per tutte, per l'intero spazio di del.icio.us. L'interfaccia qui proposta presenta al contrario categorie di alto livello che non sempre sono di comprensione così immediata, però sono differenti in base all'ambito semantico scelto dall'utente per la navigazione, per coprire tale ambito in modo migliore.

La seconda differenza è la mancanza di gerarchia che caratterizza fac.etio.us, dove le tag più importanti nei singoli facet emergono più facilmente, ma tutte le altre non trovano spazio, annegate in categorie molto vaste e piatte. Il sistema che abbiamo proposto invece rischia talvolta di nascondere parole chiave rilevanti nei livelli bassi di una gerarchia, ma garantisce la possibilità di esplorare organicamente tutto lo spazio orientandosi secondo un criterio semantico; inoltre le tag più rilevanti, anche se "nascoste" in livelli bassi dell'albero, tendono comunque a fare emergere il ramo in cui si trovano.

5.4 Tempi di esecuzione

I tempi di risposta del Web server, una volta che i dati necessari siano già disponibili nel database, sono nell'ordine di grandezza di qualche secondo; al crescere della dimensione dei dati di input i tempi di risposta aumentano in modo lineare.

Come abbiamo mostrato nel capitolo precedente e in particolare nel paragrafo 4.6.4, infatti, la complessità degli algoritmi del Web server è lineare nel numero di tag correlate, che a sua volta è proporzionale al prodotto dei parametri n_sites e n_tags (che rappresentano rispettivamente il massimo numero di siti che devono essere considerati per trovare le tag correlate e il massimo numero di tag che devono essere considerate relativamente a ogni sito).

É però importante l'osservazione che, al crescere del primo parametro, l'aumento del numero di tag distinte è mediamente più lento che lineare, perchè le tag usate tendono a ripetersi. Questo è vero soprattutto quando la tag esplorata si colloca in un ambito semantico abbastanza definito e limitato.

É rilevante a questo proposito anche la politica adottata rispetto alla *long tail*, ovvero di fatto il valore del secondo parametro, che permette a un certo punto di "tagliare la coda". Limitando questo parametro, ci aspettiamo che a maggior ragione valga quanto detto precedentemente, ovvero che anche al crescere della quantità di siti esplorati il numero totale di tag aumenti in misura ridotta, se si taglia la "lunga coda".

Non abbiamo eseguito test specifici in proposito, ma i risultati ottenuti sembrano confermare queste ipotesi e mostrano tempi di risposta di pochi secondi anche nel caso del valore massimo dell'input, con valore del parametro n_tags fissato.

La dimensione dell'input ha invece un impatto lineare senza attenuazioni per il crawler; esso infatti deve in ogni caso scandire tutte le pagine necessarie per procurare la quantità di dati richiesta. La popolarità dei siti incontrati è un dato aleatorio, che ne influenza i tempi, dato che per ogni url incontrato deve essere scandita l'intera pagina degli utenti che l'hanno salvato.

Risparmi in termini di efficienza per il crawler sono possibili quando esso incontra una risorse già scandita a partire da un'altra parola chiave. In questo caso, se i dati già presenti nel database sono aggiornati, esso non ha bisogno di scandire nuovamente la pagina relativa all'url. Questa ottimizzazione può risultare fondamentale quando si applichi l'algoritmo per un certo numero di tag, soprattutto se correlate fra loro.

Capitolo 6

Conclusioni e sviluppi futuri

6.1 Conclusioni

In questo lavoro abbiamo introdotto i sistemi di classificazione collaborativa di risorse basati su tag, o folksonomie, studiando il contesto in cui si sono sviluppati e affermati, analizzandone le caratteristiche e descrivendo alcune applicazioni esistenti. Abbiamo in particolare evidenziato i punti deboli di questo modello di classificazione, e ne abbiamo individuato nella mancanza di gerarchia una delle limitazioni principali.

Abbiamo presentato una proposta per affrontare questo problema effettuando la mappatura delle tag su un'ontologia, in modo automatico, per ottenere un'interfaccia di navigazione più ricca, dove le informazioni siano organizzate secondo un criterio semantico condiviso.

Abbiamo quindi presentato il sistema realizzato, che si integra nell'interfaccia Web di del.icio.us, una folksonomia particolarmente popolare e rappresentativa, e utilizza WordNet, un lessico semantico della lingua inglese, per effettuare la mappatura delle tag e introdurre una gerarchia.

Abbiamo illustrato i principali problemi che si sono presentati e le soluzioni che abbiamo adottato per risolverli: il recupero dei dati con un crawler, la disambiguazione delle tag necessaria per associarle a concetti dell'ontologia, la creazione di un albero semantico basato sulla relazione di iperonimia di WordNet e la compressione della gerarchia per renderla esplorabile; la realizzazione di uno script, da eseguire nel browser dell'utente per modificare dinamicamente il contenuto delle pagine integrando le informazioni elaborate dal server.

Infine abbiamo illustrato i risultati qualitativi ottenuti, mostrando il fun-

zionamento del sistema in diverse situazioni. L'analisi di alcuni esempi di uso ha mostrato che l'interfaccia integrata offre all'utente delle strategie di navigazione che possono rivelarsi molto utili in diversi contesti, per trovare informazioni e per orientarsi nello spazio delle tag, e che in una folksonomia "piatta" sarebbero impossibili; la gerarchia introdotta mostra inoltre di affrontare almeno in parte diversi dei principali problemi indicati in letteratura come i punti deboli delle folksonomie: l'ambiguità, il controllo dei sinonimi, il basso livello di *recall*, il *gaming*.

Il sistema realizzato non può dirsi maturo e presenta dei limiti, emersi soprattutto in contesti semantici eterogenei con quantità elevate di dati; i risultati ottenuti sono però complessivamente incoraggianti e ci portano a concludere che questo tipo di soluzione, che mescola aspetti dei due diversi approcci, top-down e bottom-up, al problema della classificazione, può costituire una risposta utile ad alcune delle principali limitazioni delle folksonomie, e una strada interessante per arricchirne le possibilità di navigazione.

6.2 Sviluppi futuri

Il primo aspetto particolarmente delicato su cui si potrebbero ottenere dei miglioramenti significativi del sistema in termini di qualità dei risultati è l'algoritmo di compressione dell'albero: nell'implementazione attuale esso presenta un elemento di rigidità, nel cancellare alcuni concetti, ritenuti troppo generali, ogni volta che essi compaiono; potrebbe essere utile un algoritmo più flessibile, che a seconda delle dimensioni dell'input e delle situazioni possa decidere per esempio di includere anche alcune delle categorie generali, se necessario per limitare il fattore di ramificazione dell'albero al primo livello.

Per quanto riguarda l'aspetto delle prestazioni in termini di efficienza, abbiamo mostrato come esse siano buone, anche con quantità consistenti di dati di input, per il modulo che si occupa della costruzione dell'albero; i tempi di esecuzione del crawler e del modulo per la disambiguazione delle tag, invece, costituiscono un ostacolo alla possibilità di fornire il servizio in tempo reale anche per dati che non siano stati almeno in buona parte preprocessati e la cui dimensione non sia molto ridotta. I tempi necessari per la disambiguazione delle tag potrebbero essere migliorati considerevolmente implementando un algoritmo più efficiente, anche a partire da quelli attualmente usati; un miglioramento notevole si potrebbe ottenere nel processo di recupero dei dati, appoggiandosi al database descritto in [Eynard, 2007a]. Avendo a disposizione un database sufficientemente aggiornato, non sareb-

be più necessario ricorrere al crawler, se non per raccogliere i dati più recenti.

Per quanto riguarda le restrizioni sulle tag che non sono riconosciute e non possono essere utilizzate dal sistema, ci sono due ipotesi che offrono prospettive interessanti: l'integrazione di WordNet con ontologie di dominio per settori semantici specifici e con wordnet locali per offrire uno strumento multilingua.

Una possibilità interessante in cui il sistema potrebbe trovare facilmente applicazione è quella di estrarre dati da più folksonomie contemporaneamente. Rimanendo nell'ambito del social bookmarking, i dati potrebbero essere estratti da diversi sistemi in parallelo e utilizzati insieme, mescolati, nel processo di disambiguazione e nella costruzione dell'albero. Gli attributi del peso proprio di ogni nodo potrebbero essere differenziati in base alle folksonomie di provenienza dei bookmark; in questo modo rimarrebbe nell'albero una traccia utile per reindirizzare l'utente verso sistemi diversi. L'albero risultante potrebbe essere integrato con diverse folksonomie analogamente al sistema attuale, oppure essere consultato su un sito a sé stante, con link dai nodi dell'albero diretti alle pagine dei diversi sistemi.

Facendo un passo ulteriore rispetto al punto precedente, si potrebbero integrare nel sistema anche folksonomie basate su altri tipi di risorse. Le tag infatti hanno valore generale: con la stessa parola possano essere annotati una fotografia, un video, una notizia, un luogo su una mappa, un articolo di un blog. I dati relativi a risorse differenti non verrebbero mischiati nella fase di disambiguazione, qualora si tratti di risorse di tipo differente, ma solo nella fase successiva di costruzione dell'albero.

Infine osserviamo che il problema dell'ambiguità delle tag è stato affrontato solo in parte; infatti esse vengono disambiguate rispetto alle risorse a cui si riferiscono, ma questo dato viene utilizzato solo per effettuare la mappatura delle tag, e non viene sfruttato per distinguere le risorse: una volta che la disambiguazione è stata effettuata, si potrebbe prevedere la possibilità di filtrare i contenuti delle pagine di del.icio.us in base a questi dati.

Nel nostro classico esempio relativo alla parola "turkey", la pagina relativa a questa tag potrebbe essere modificata dinamicamente da un opportuno script di Greasemonkey, che permetta di visualizzare, a scelta, solo le risorse identificate come relative ad uno dei significati della parola, avendo ricevuto queste informazioni dal server.

Bibliografia

[Anderson, 2004] Anderson, C. (2004). The long tail. http://www.wired.com/wired/archive/12.10/tail.html.

[Banerjee and Pedersen, 2002] Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 136–145, London, UK. Springer-Verlag.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5).

[Biddulph, 2004] Biddulph, M. (2004). Introducing del.icio.us. http://www.xml.com/lpt/a/2004/11/10/delicious.html.

[Blei et al., 2004] Blei, D. M., Griffiths, T. L., Jordan, M. I., and Tenenbaum, J. B. (2004). Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*.

[Bolchini and Mainetti, 2004] Bolchini, D. and Mainetti, L. (2004). Wp7.1: Requirements engineering methodology for multichannel and multimodal interactive applications.

[Bolchini and Paolini, 2006] Bolchini, D. and Paolini, P. (2006). Interactive dialogue model: a design technique for multichannel applications. *IEEE Transactions on Multimedia*, 8(3):529–541.

[Bricklin, 2000] Bricklin, D. (2000). The cornucopia of the commons: How to get volunteer labor. http://www.bricklin.com/cornucopia.htm.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

[Broughton, 2002] Broughton, V. (2002). Faceted classification as a basis for knowledge organization in a digital environment: the bliss bibliographic classification as a model for vocabulary management and the creation of multidimensional knowledge structures. *New Rev. Hypermedia Multimedia*, 7(1):67–102.

[Dakka et al., 2005] Dakka, W., Ipeirotis, P. G., and Wood, K. R. (2005). Automatic construction of multifaceted browsing interfaces. In Herzog, O., Schek, H.-J., Fuhr, N., Chowdhury, A., and Teiken, W., editors, *CIKM*, pages 768–775. ACM.

[Denton, 2003] Denton, W. (2003). How to make a faceted classification and put it on the web-web-howto.html. http://www.miskatonic.org/library/facet-web-howto.html.

[Dijck, 2005] Dijck, P. V. (2005). Emergent i18n effects in folksonomies. http://poorbuthappy.com/ease/archives/2005/01/15/2419/multilingual-folksonomies.

[English et al., 2002] English, J., Hearst, M., Sinha, R., Swearingen, K., and Yee, K.-P. (2002). Hierarchical faceted metadata in site search interfaces. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 628–639. ACM Press.

[Eynard, 2007a] Eynard, D. (2007a). Perl hacks: del.icio.us scraper. http://davide.eynard.it/?p=28.

[Eynard, 2007b] Eynard, D. (2007b). Some del.icio.us stats. http://davide.eynard.it/?p=33.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet – An Electronic Lexical Database*. MIT Press.

[Gibson, 2004] Gibson, B. (2004). Ibm's intranet and folksonomy. http://thecommunityengine.com/home/archives/2005/03/ibms_intranet_a.html.

[Golder and Huberman, 2005] Golder, S. and Huberman, B. A. (2005). The structure of collaborative tagging systems.

[Gruber, 2005] Gruber, T. (2005). Ontology of folksonomy: A mash-up of apples and oranges. http://tomgruber.org/writing/ontology-of-folksonomy.htm.

[Himanen, 2001] Himanen, P. (2001). *The Hacker Ethic and the Spirit of the Information Age*. Random House Inc., New York, NY, USA. Epilogue By-Manuel Castells and Prologue By-Linus Torvalds.

[Hotho et al., 2006] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006). Trend detection in folksonomies. In Avrithis, Y. S., Kompatsiaris, Y., Staab, S., and O'Connor, N. E., editors, *Proc. First International Conference on Semantics And Digital Media Technology (SAMT)*, volume 4306 of *LNCS*, pages 56–70, Heidelberg. Springer.

[Hyde, 2005] Hyde, B. (2005). Tagging powerlaw. http://enthusiasm.cozy.org/archives/2005/01/tagging-powerlaw/.

[Lesk, 1986] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, NY, USA. ACM Press.

[Mathes, 2004] Mathes, A. (2004). Folksonomies – cooperative classification and communication through shared metadata. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[Mejias, 2004] Mejias, U. A. (2004). A del.icio.us study.

[Merholz, 2004] Merholz, P. (2004). Ethnoclassification and vernacular vocabularies. http://www.peterme.com/archives/000387.html.

[Mihalcea and Moldovan, 2001] Mihalcea, R. and Moldovan, D. I. (2001). Ez.wordnet: Principles for automatic generation of a coarse grained wordnet. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 454–458. AAAI Press.

[O'Reilly, 1997] O'Reilly, T. (1997). Hardware, software, and infoware. *Commun. ACM*, 40(2):33–34.

[O'Reilly, 2005] O'Reilly, T. (2005). What is web 2.0. design patterns and business models for the next generation of software.

[Patwardhan et al., 2005] Patwardhan, S., Pedersen, T., and Banerjee, S. (2005). SenseRelate::TargetWord - A Generalized Framework for Word Sense Disambiguation. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 73–76, Ann Arbor, MI.

[Pedersen et al., 2004] Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet: : Similarity - measuring the relatedness of concepts. In *AAAI*, pages 1024–1025.

[Pianta et al., 2002] Pianta, E., Bentivogli, L., and Girardi, C. (2002). Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet, Mysore, India*.

[Quintarelli, 2005] Quintarelli, E. (2005). Folksonomies: power to the people.

[Raymond, 1997] Raymond, E. S. (1997). *The Cathedral and the Bazaar*. O'Reilly.

[Rosenfeld, 2005] Rosenfeld, L. (2005). Folksonomies? how about metadata ecologies?

[Shepard et al., 2006] Shepard, H., Halpin, H., and Robu, V. (2006). The dynamics and semantics of collaborative tagging.

[Shirky, 2004] Shirky, C. (2004). Folksonomy. http://many.corante.com/archives/2004/08/25/folksonomy.php.

[Shirky, 2005a] Shirky, C. (2005a). Folksonomies are a forced move. http://many.corante.com/archives/2005/01/22/folksonomies_are_a_forced_move_a_response_to_liz.php.

[Shirky, 2005b] Shirky, C. (2005b). Shirky: Ontology is overrated – categories, links, and tags. http://shirky.com/writings/ontology_overrated.html.

[Stoica et al., 2006] Stoica, E., Hearst, M., and Richardson, M. (2006). Automating creation of hierarchical faceted metadata structures.

[Stoica and Hearst, 2004] Stoica, E. and Hearst, M. A. (2004). Nearly-automated metadata hierarchy creation. http://flamenco.berkeley.edu/papers/hlt-naacl04.pdf.

[Surowiecki, 2004] Surowiecki, J. (2004). *The wisdom of crowds*. Doubleday.

[Szekely and Torres, 2005] Szekely, B. and Torres, E. (2005). Ranking bookmarks and bistros: Intelligent community and folksonomy development.

[Torvalds and Diamond, 2001] Torvalds, L. and Diamond, D. (2001). *Just for Fun: The Story of an Accidental Revolutionary*. HarperBusiness, New York City.

[Udell, 2005] Udell, J. (2005). del.icio.us: the screencast. http://weblog.infoworld.com/udell/2005/03/14.html.

[Vander Wal, 2005] Vander Wal, T. (2005). Explaining and showing broad and narrow folksonomies. http://www.personalinfocloud.com/2005/02/explaining_and_.html.

[Yee et al., 2003] Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. (2003). Faceted metadata for image search and browsing. In *CHI '03: Proceedings of the conference on Human factors in computing systems*, pages 401–408. ACM Press.