Università degli Studi di Padova
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Specialistica in
INGEGNERIA INFORMATICA

Tesi di Laurea

Studio di uno Strumento di Ritrovamento dell'Informazione per il proprio *Desktop*

RELATORE: Prof. Massimo Melucci

LAUREANDO: Giovanni Melis

Anno Accademico 2009/2010

A mio fratello Marco

I problemi non possono essere risolti
allo stesso livello di conoscenza che li ha creati
Albert Einstein

La conoscenza è di due tipi:
o conosciamo un soggetto per nostro conto,
oppure conosciamo il posto
dove poter trovare informazioni al riguardo
Samuel Johnson

Indice

Sommario

Con l'aumento delle capacità dei dispositivi di storage e il largo utilizzo di Internet, i computer degli utenti sono diventati dei veri e propri archivi di documenti. Per cercare all'interno del *file system* dei calcolatori vengono solitamente utilizzati i cosiddetti *desktop search*, cioè veri e propri motori di ricerca per uso locale.

In questa tesi viene studiato uno strumento innovativo per la ricerca dei documenti nel *desktop* dell'utente. L'idea di base è quella di scansionare l'intera collezione di file memorizzata nell'hard-disk alla ricerca dei documenti di potenziale interesse, premiando in modo particolare quelli legati all'attività recente.

A questo scopo è stata sfruttata l'analogia fra l'attività recente e la cronologia web usualmente gestita dai browser. Lo studio di Firefox ha permesso di sviluppare l'applicazione *Awesome++* che permette di integrare la cronologia Internet con i documenti locali provenienti dal *desktop* dell'utente. Si è utilizzata la *awesome bar* di Firefox come interfaccia per proporre ai potenziali utilizzatori delle liste ibride di documenti locali e pagine web attinte dalla cronologia precedentemente arricchita.

Per valutare l'efficacia del sistema sviluppato, è stato creato un gruppo di valutazione avente il compito di valutare le liste di documenti proposti attraverso l'utilizzo di *Awesome++*; si è chiesto di esprimere giudizi di rilevanza su una configurazione con sole pagine web e sulla configurazione con liste ibride di documenti locali e web.

I risultati sperimentali ottenuti sono stati successivamente analizzati attraverso indici di sintesi statistici e con le misure di *Cumulated Gain* e loro derivate, proprie dell'*Information Retrieval*. Queste misure hanno evidenziato non solo la capacità del sistema sviluppato di rispondere alle esigenze degli utenti, ma anche una buona efficacia soprattutto sui topic caratteristici degli utenti stessi.

Il software *Awesome++* può essere considerato una base da cui partire per lo

sviluppo di uno strumento per utilizzo continuativo o professionale.

Capitolo 1

Introduzione

Il *Reperimento dell'Informazione* (in inglese *Information Retrieval*, abbreviato *IR*) è la disciplina che studia le attività di ricerca di informazione in collezioni di documenti. Il caso più noto di sistema di reperimento dell'informazione è quello dei motori di ricerca Web. Tuttavia il campo della ricerca dell'informazione è così vasto che sarebbe riduttivo pensare esclusivamente ai *search engine*.

In generale si può affermare che [2] il termine *Reperimento dell'Informazione* identifica tutte le attività utilizzate per scegliere, da una data collezione di documenti, quelli di interesse in relazione ad una specifica esigenza informativa di una persona. I documenti di interesse, sono i cosiddetti *documenti rilevanti*, cioè quelli in grado di colmare la mancanza di informazioni che ha portato l'utente a esprimere la propria esigenza informativa.

Fra le applicazioni più comuni delle tecniche di IR, possono essere elencati:

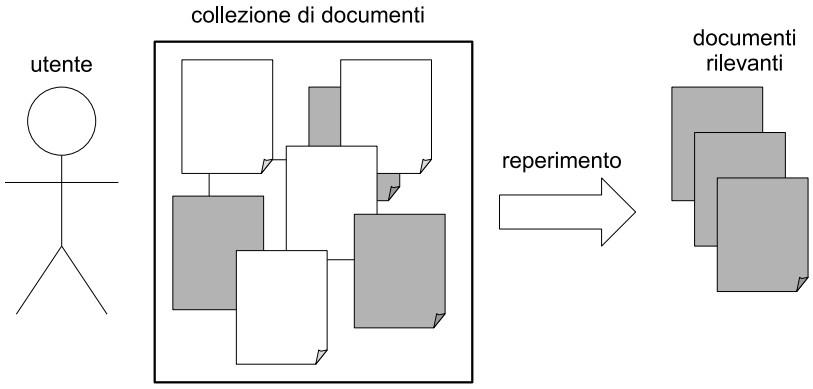

Figura 1.1: Obiettivo del Reperimento dell'Informazione

- I motori di ricerca Web
- I motori di ricerca *Desktop* (*desktop search*)
- Gli OPAC (*Online Public Access Catalogue*)
- L'Enterprise Search (spesso all'interno di una *Intranet*)
- I metamotori[1] (motori di ricerca che instradano verso altri motori)
- Il *Peer-to-Peer*
- Il Filtraggio
- La Classificazione / Categorizzazione[2]
- Il *Question Answering*

1.1 Motivazioni e obiettivi

Con la rapida evoluzione della tecnologia per lo *storage* dei dati, la quantità d'informazioni digitali registrabili in un PC casalingo o in una rete aziendale è notevolmente aumentata. Cercare un file in un *hard-disk* della capienza dell'ordine delle decine o delle centinaia di gigabyte può risultare un'operazione noiosa e dispendiosa in termini di tempo. Solo agli inizi di questo secolo le tecnologie originariamente sviluppate per l'ambiente Web sono state impiegate per la ricerca in locale. Ciò ha portato allo sviluppo di diversi motori di tipo *desktop*, per la ricerca dei file su un computer in modo abbastanza efficace ed economico.

Lo sviluppo e la diffusione di Internet hanno permesso di ampliare le fonti di informazioni a cui gli utenti riescono ad accedere e di fornire dei nuovi metodi di condivisione delle risorse. In pratica, un navigatore nel tempo scarica e memorizza

[1] Un interessante metamotore di recente creazione è *Ecosia*[13] che per eseguire le ricerche interroga i più noti Bing e Yahoo. I ricavi ottenuti dalle inserzioni pubblicitarie di Ecosia permettono ad un distaccamento del WWF in Brasile di salvare circa $2m^2$ di foresta pluviale per ogni ricerca effettuata.

[2] Un ottimo testo per l'approfondimento della Classificazione e la Categorizzazione, ma anche per alcune delle altre applicazioni elencate, è *Search Engines: Information Retrieval in Practice*[1].

una grande quantità di documenti di cui farà un uso spesso parziale o si dimenticherà dell'esistenza. Questa collezione di documenti presenta dei limiti intrinseci:

- I documenti di cui è composta sono dislocati in punti diversi nel *file system*, spesso in cartelle temporanee aventi nomi non espressivi del loro contenuto.
- I documenti sono sottoposti a un fenomeno di obsolescenza, specialmente se molto datati, in quanto il loro contenuto informativo può essere non aggiornato.

Tuttavia essa presenta anche aspetti positivi:

- I file di cui è composta sono stati copiati o scaricati dagli utenti e perciò sono il frutto di un'azione atta a colmare un'esigenza informativa intercorrente nel momento del loro salvataggio su disco.
- Molti dei documenti di cui è formata possono essere utili alle esigenze informative correnti, poiché sono frequenti i casi in cui l'utente sa di aver già letto e memorizzato un documento, magari in una fase di studio preliminare, e ha necessità di recuperarlo in un secondo tempo per un suo utilizzo concreto.
- È memorizzata nel *file system* in modo permanente e non ha bisogno di ausili esterni per la consultazione (ad esempio non è necessaria la connettività ad Internet).
- Differentemente dalle pagine web che cambiano e spariscono qualora un sito venga aggiornato o rimosso e di loro si ha al massimo a disposizione una *copia cache*, i documenti locali rimangono stabilmente a disposizione.

Il contenuto del *desktop* dovrebbe perciò essere rivalutato e considerato una preziosa risorsa, in grado di rispondere a molte delle esigenze informative dell'utente, senza ricorrere all'ausilio della Rete.

Sebbene la ricerca dell'informazione sul computer dell'utente sia quasi sempre associata ai *desktop search*, spesso ci si trova ad avere esigenze informative diverse da quelle colmabili con l'utilizzo di un motore di ricerca. Si analizzino le due seguenti situazioni:

S1 Un ricercatore sta eseguendo uno studio su un particolare argomento; potrebbe perciò trovare comoda un'applicazione che riesca a proporgli documenti inerenti ai propri interessi di ricerca recenti.

S2 Nella stesura di un articolo, un giornalista ha l'esigenza di recuperare un documento che sa di aver memorizzato in passato nel proprio disco, ma non si ricorda dove.

Entrambe queste situazioni non si adattano alle caratteristiche del *desktop search*. La prima [S1] perché un motore di ricerca non sa qual è l'attività recente dell'utente; la seconda [S2] perché un *desktop search* propone una lista di documenti rilevanti in base all'analisi del loro contenuto, mentre l'esigenza del giornalista era di semplice recupero di un particolare documento. La situazione S2 si configura in quello che Sergey Chernov et al.[3] definiscono come *Know-Item Retrieval Task*, cioè l'esigenza per l'utente di trovare uno specifico documento nel proprio *desktop*, non conoscendone la sua precisa posizione o il titolo esatto (percorso e nome del file).

Allo scopo di poter arrivare a gestire situazioni come quelle appena descritte, un settore dell'*Information Retrieval* si occupa del *Personal Information Management* (PIM), cioè lo studio delle attività che le persone eseguono per acquisire, organizzare, mantenere e recuperare informazioni come pagine-web, messaggi di posta elettronica o documenti personali. A questo scopo sono state proposte in letteratura diverse tecniche e modalità di registrazione dell'attività dell'utente (*logging*), da cui estrarre informazioni per migliorare gli strumenti di ricerca.

Sebbene numerosi *desktop search* siano stati sviluppati e proposti in letteratura e nel mercato ed essi sembrino offrire strumenti efficaci dal punto di vista del ritrovamento all'interno del *file system*, essi trascurano la modellazione dell'attività recente dell'utente. È in questo contesto che si colloca questo studio che ha come risultato lo sviluppo di uno strumento in grado di risolvere almeno parzialmente il problema proposto.

1.2 La soluzione proposta

Il problema nella sua visione più generale è quello di studiare un metodo per valorizzare e rendere utile il contenuto del *desktop* dell'utente, al fine di fare emergere i documenti potenzialmente interessanti per l'attività recente dell'utente. Cercare di risolvere un problema di così ampio respiro è difficile soprattutto perché è complesso modellare l'attività recente. Inoltre per sua natura la gestione del *file system* è strettamente connessa a funzioni interne al sistema operativo. Per eseguire alcune migliorie si dovrebbe perciò entrare nel delicato ambito della programmazione a basso livello del sistema operativo.

Dopo una fase di studio preliminare è stata riscontrata un'importante analogia fra attività recente e la *history* di un *web browser*. La cronologia di un browser infatti è una collezione di documenti strutturata e riportante riferimenti temporali di quando una visita di una certa pagina web è avvenuta. Evidenziando l'aspetto temporale anche per i documenti locali (tramite l'uso della data di ultima modifica dei file gestita dal sistema operativo) è possibile adattare i meccanismi di gestione della cronologia web anche ai documenti di tipo *desktop* e ottenere indicazioni sull'attività recente dell'utente.

Su questa linea guida, è stato sviluppato un sistema in grado di trovare e proporre all'utente documenti presenti nel proprio *desktop*. E' stato impostato ed eseguito un esperimento di valutazione, con l'obiettivo di misurare in modo sperimentale, attraverso giudizi di rilevanza dati da un gruppo di valutazione, le differenze in termini di efficacia di due configurazioni del sistema: una configurazione basata su una cronologia web uguale per tutti gli utenti e una configurazione con la cronologia web arricchita coi documenti locali presenti nel *desktop* di ciascun membro del gruppo di valutazione. I risultati sono stati complessivamente positivi e hanno mostrato una maggiore efficacia dello strumento sviluppato quando il sistema è stato interrogato con query relative all'attività recente e personale degli utenti. Buone prestazioni sono state riscontrate anche per query non proposte dagli utenti del gruppo di valutazione, ma di natura generale. Quando gli utenti hanno cercato di interrogare il sistema con query relative all'attività personale di altri membri del team di valuta-

zione, i risultati sono stati comprensibilmente negativi: ciò è la controprova che il sistema sviluppato sia basato sull'effettiva attività *personale* dell'utente.

1.3 Lavori precedenti e correlati

La letteratua dell'*Information Retrieval* è relativamente giovane e dedicata per la gran parte ai motori di ricerca e, più anticamente, alle biblioteche prima cartacee e successivamente digitali (*digital libraries*). C'è stato un grande studio soprattutto su tecniche di indicizzazione, sui modelli di reperimento e sui metodi di ordinamento dei documenti. L'innovazione significativa più recente è forse l'applicazione dell'algoritmo di ordinamento chiamato *Page-rank* che si basa sulla metafora del navigatore casuale (*random walk*) ed è stato brevettato da Sergey Brin e Larry Page, divenuti successivamente fondatori di Google®.

Come anticipato, il progetto sviluppato ha analogie con i *desktop search*. La letteratura che li riguarda è tuttavia relativamente recente e limitata. Nel 1999 Beaza-Yates & Ribero-Neto[4] stilarono semplicemente una prima lista dei motori di ricerca di tipo *desktop*. Successivamente Tom Noda & Shawn Helwig[5] effettuarono una prima valutazzone su dodici *desktop search* analizzandone anche alcuni aspetti qualitativi. Il lavoro significativo più recente è ad opera di Lu et al.[6] che nel 2007 eseguirono una valutazione delle performance su sette *desktop search* tra cui quelli di Yahoo®, Google® e Microsoft®. Nel 2006, Stefania Costache[7] propone un metodo per migliorare la ricerca su *desktop* al fine di poter applicare algoritmi di ordinamento quali *Page-rank*, basandosi su metadati semantici provenienti da diversi contesti (email, file e Web-cache).

Alcuni studi sono stati eseguiti sulla gestione delle informazioni personali (PIM). È il caso del lavoro di Sergey Chernov et al.[3] che è basato sull'uso dei *log* dell'attività dell'utente al fine di costruire un dataset arricchito. Anche Paul-Alextandru Chirita & Wolfgang Nejdl[8] hanno usato l'analisi comportamentale degli utenti per migliorare gli algoritmi di ordinamento di applicazioni di ricerca di tipo *desktop*. L'idea di base è quella di generare collegamenti fra risorse in modo da ricostruire quei *link* che sono alla base degli algoritmi di *random-walk*. Un altro lavoro di Paul-

Alexandru Chirita et al.[9] usa invece il tempo di lettura (*reading time*) dell'utente per ottenere *feedback impliciti* in grado di identificare il contesto dell'attività recente dell'utente, al fine di migliorare la ricerca in ambiente *desktop*.

Di utile lettura è stato il *paper* di Sergey Chernov et al.[10] che spiega molto bene come costruire un buon banco di prova per eseguire test sperimentali quando si vogliono valutare sistemi di *Personal Information Management* o comunque in ambiente di ricerca locale.

1.4 Composizione della relazione

La relazione si sviluppa in 7 capitoli:

- Capitolo 2 - *Analisi della* Awesome bar *di Mozilla Firefox*: studio di funzionamento della barra degli indirizzi di Mozilla Firefox che, dalla versione 3.0 del browser, suggerisce pagine Web presenti nella cronologia Web;

- Capitolo 3 - *Sviluppo dell'applicazione* Awesome++: spiega l'ideazione e lo sviluppo dello strumento che permette di integrare la cronologia web generata da Mozilla Firefox con documenti locali di provenienza *desktop*;

- Capitolo 4 - *Fase di valutazione*: descrive l'organizzazione e la conduzione della fase di valutazione del sistema di reperimento sviluppato con l'ausilio del tool *Awesome++*;

- Capitolo 5 - *Risultati sperimentali della valutazione*: vengono riportati i risultati delle sessioni di valutazione eseguite dal gruppo di utenti, con l'uso di tabelle, grafici e indici di sintesi;

- Capitolo 6 - *Analisi dei risultati*: è una discussione critica dei risultati sperimentali ottenuti;

- Capitolo 7 - *Conclusioni e sviluppi futuri*: fornisce una valutazione dello strumento sviluppato e suggerisce linee guida per lo sviluppo futuro di applicazioni analoghe;

Seguono tre appendici:

A. Risultati sperimentali ottenuti dagli utenti (in formato tabellare)

B. Istruzioni di configurazione del sistema

C. Guida per eseguire la valutazione

Capitolo 2

Analisi della *Awesome bar* di Mozilla Firefox

Volendo arrivare allo sviluppo di uno strumento in grado di tener conto della storia dell'utente, si è subito pensato al parallelo tra l'attività recente nel proprio *desktop* e la cronologia[1] web di un browser. Esattamente come un browser registra le visite delle pagine web organizzandole in ordine temporale, anche il sistema operativo di un dispositivo (anche non collegato in Rete) automaticamente memorizza le date di ultima modifica e di ultimo accesso dei file contenuti nel proprio *file system*. L'idea è includere nella cronologia di un browser le risorse provenienti dal dispositivo. Poiché Firefox[14], ideato e sviluppato all'interno della *Mozilla Foundation*, ha la possibilità di essere esteso e personalizzato, esso è il naturale candidato a fornire gli strumenti necessari allo scopo. Non solo: è anche multipiattaforma, quindi indipendente dal sistema operativo; è *open source*[2], cioè lasciato dai propri autori aperto allo studio e alle modifiche di altri programmatori.

Dalla versione 3.0 di Firefox, quella che tutti conoscono come la barra degli indirizzi (o *URL-bar*) ha acquisito funzionalità di ricerca che hanno portato gli autori

[1] La cronologia è diversa dalla *web cache*. La cache infatti è una semplice copia su disco delle pagine web visitate recentemente ed ha lo scopo di velocizzare il caricamento delle pagine sul browser o di permettere la cosiddetta *navigazione non in linea*. La cronologia invece è l'insieme di pagine web visitate di recente, associate a parametri temporali e di utilizzo.

[2] La licenza di utilizzo di tipo open source non cede i diritti d'autore. Il nome e il logo di Firefox sono marchi registrati. La legge vigente per i marchi registrati impone di accompagnarli sempre dal simbolo corrispondente. Per questioni di pesantezza di notazione, poiché il nome Firefox viene usato molto in questa relazione, si è scelto di non riportare il simbolo di marchio registrato accanto al nome. Tuttavia la dicitura *Firefox* è sempre da intendersi come *Firefox*®.

a chiamarla *Awesome bar*, per le sue capacità di suggerimento di pagine web. Essa infatti ha un comportamento di questo tipo: non appena l'utente comincia a digitare una parola o una porzione di un indirizzo nella barra, un algoritmo di *pattern matching* cerca corrispondenze (anche parziali) fra quanto digitato dall'utente e l'insieme delle stringhe costituenti il Titolo[3] e l'URL delle pagine presenti nella cronologia del browser. Sotto la barra viene proposto immediatamente un menu contenente diverse *entry* ciascuna formata da due campi corrispondenti a pagine della *history*: Titolo e URL. L'utente può quindi scegliere di cliccare su una delle pagine proposte. Questa caratteristica si rivela utile, in quanto gli utenti possono accedere in modo veloce a pagine o siti web che sono stati visitati di recente.

La *awesome bar* è proprio atta a risolvere il problema del *Know-Item Retrieval Task*, nel caso particolare in cui gli *item* sono pagine web visitate. Il suo utilizzo più comune è infatti quello di ricercare una pagina web specifica, che l'utente sa di aver già visitato ma di cui non si ricorda bene l'indirizzo o il titolo. Inoltre avendo come collezione di documenti di lavoro la cronologia del browser, di fatto lavora con l'attività web recente[4].

L'idea di base per lo sviluppo concreto dell'applicazione di ritrovamento di documenti locali è di estendere l'uso della *awesome bar* anche alla ricerca di docu-

Figura 2.1: Screenshot dell'interfaccia di Firefox: sono evidenziate la *awesome bar* e la *search bar*

[3]La stringa di caratteri racchiusa all'interno del tag TITLE della pagina HTML

[4]Nel caso dell'attività web, gli sviluppatori di Firefox considerano *recente* quella degli ultimi tre mesi. Ciò è quanto si evince dall'algoritmo di calcolo del parametro *frecency* che verrà illustrato in seguito.

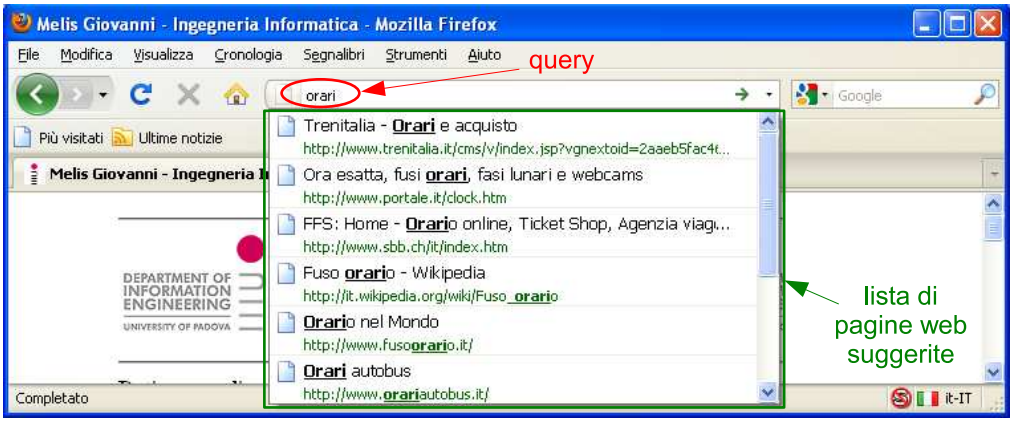

Figura 2.2: Visualizzazione della funzione di suggerimento della *awesome bar*

menti presenti nel *desktop* dell'utente. Per fare ciò è necessario studiare i dettagli di funzionamento della barra, descritti nelle sezioni successive.

2.1 Funzionamento della *Awesome bar*

Per capire come la *aweseme bar* riesca a produrre la lista di documenti da suggerire all'utente, è necessario studiare la documentazione per gli sviluppatori pubblicata nel *Mozilla Developer Center*[16]. Il suo funzionamento è basato su un *database SQLite*[5] chiamato *places.sqlite* che è stato introdotto dalla versione 3.0 del browser per gestire cronologia, preferiti (intesi come *bookmark*), sessioni e altre funzionalità di supporto. Quando l'utente digita del testo nella barra degli indirizzi, Firefox automaticamente interroga il database dove è memorizzata la cronologia alla ricerca di corrispondenze fra quanto digitato e i record presenti nella tabella *moz_places*. L'ordine con cui le *entry* della *history* vengono presentate nel menu segue l'ordinamento decrescente del campo *frecency* della stessa tabella. *Frecency*[20] è un valore assegnato ad ogni pagina presente nella cronologia che sintetizza la frequenza di visita di quella pagina e la distanza temporale dell'ultima visita. Tutto questo risulterà più chiaro dopo aver analizzato la struttura della tabella *moz_places* in dettaglio. A questo livello è sufficiente dire che le pagine della *history* proposte dalla *awesome bar* sono presentate in ordine decrescente di *frecency*.

[5] SQLite è una libreria multipiattaforma scritta in linguaggio C che implementa un motore SQL per database[17]

id	url	title	rev_host	visit_co...	hidden	typed	favicon_id	frecency	last_visit_date
766	http://www.lyricsmode.com/lyrics/a/andrea_b...	Andrea Bocelli Lyr...	/illecob_aerdn...	1	0	0		1000	12653926028750...
767	http://www.lyricsmode.com/lyrics/z/zucchero/	Zucchero Lyrics, L...	/orehccuz/z/s...	1	0	0		233	12653926033750...
768	http://classicalmusic.about.com/od/opera/qt/...	Nessun Dorma Ly...	/mth.txetamro...	1	0	0		100	12653926038590...
769	http://www.lyricstime.com/-lyrics.html	Pink Martini - Ninn...	/mth.sciryl-/m...	1	0	0		54	12653926043430...
770	http://www.elyrics.net/song/e/eros-ramazzot...	EROS RAMAZZOT...	/mth.sciryl-itto...	1	0	0		34	12653926049060...
771	http://www.elyrics.net/song/t/tiziano-ferro-ly...	TIZIANO FERRO ...	/mth.sciryl-orr...	1	0	0		23	12653926053590...
772	http://www.youtube.com/watch%3Fv%3Do7...	YouTube - Fratelli...	/wa1u-ymp7o...	1	0	0		17	12653926058750...
773	http://www.lyricsmania.com/gianna_nannini_l...	Gianna Nannini Ly...	/mth.sciryl_ini...	1	0	0		13	12653926063750...
774	http://www.testimania.com/	Testi Mania.com -...	/moc.ainamits...	1	0	0		10	12653926068280...
775	http://www.lyrics.it/	Lyrics :: Happy F...	/ti.sciryl.www...	1	0	0		8	12653926072500...
776	http://www.metrolyrics.com/Andrea-Bocelli-ly...	ANDREA BOCELLI...	/mth.sciryl-illec...	1	0	0		7	12653926077810...
777	http://lyrics-keeper.com/en/eros-ramazzotti/p...	Eros Ramazzotti -...	/mth.em-noc-a...	1	0	0		5	12653926083750...

Figura 2.3: Visione di alcuni record della tabella *moz_places*

2.2 Gestione della cronologia

Per capire come Firefox gestisce la propria cronologia, è necessario dare uno sguardo alla tabella *moz_places* su cui vengono salvati i dati delle pagine web visitate di recente. In linguaggio *SQL* (*Structured Query Language*), la tabella ha la seguente struttura:

```
CREATE TABLE moz_places (id INTEGER PRIMARY KEY, url
LONGVARCHAR, title LONGVARCHAR, rev_host LONGVARCHAR,
visit_count INTEGER DEFAULT 0, hidden INTEGER DEFAULT 0
NOT NULL, typed INTEGER DEFAULT 0 NOT NULL, favicon_id
INTEGER, frecency INTEGER DEFAULT -1 NOT NULL,
last_visit_date INTEGER)
```

Significato dei campi della tabella *moz_places*:

- *id*: la chiave primaria della tabella, è un intero;
- *url*: stringa dove viene memorizzato l'URL della pagina;
- *title*: stringa dove viene memorizzato il titolo della pagina web (corrispondente a quanto racchiuso nel tag TITLE delle pagine HTML);
- *rev_host*: contiene il *reverse*[6] della stringa contenuta campo *url*;
- *visit_count*: intero che conta il numero di volte che la pagina è stata visitata;
- *hidden*: flag che segnala se l'URL non è stato navigato di proposito dall'utente, ma attraverso *frame* interni ad altre pagine o tramite chiamate *Javascript*;

[6]Data una stringa x, reverse(x), spesso scritto anche x^R, è dato dalla stringa x scritta al contrario. Ad esempio, se $x = abcd$, $x^R = dcba$.

- *typed*: flag che segnala se l'URL è stato digitato dall'utente;
- *favicon_id*: è l'ID della *favicon*[7] associata alla pagina;
- *frecency*: è un intero che corrisponde al valore di *frecency*[8] calcolato per la pagina;
- *last_visit_date*: è il campo in cui viene memorizzata la data di ultima visita della pagina, in formato *timestamp*

Com'è possibile dedurre dalla struttura della tabella *moz_places*, Firefox registra una serie di attributi per ogni pagina web che è stata aperta nel browser. Attraverso l'ispezione del database in cui è memorizzata la tabella, ad esempio attraverso l'estensione *SQLite Manager*[19], è possibile verificare che a parità di *matching*, nell'uso della *awesome bar* vengono proposte le voci di cronologia in ordine decrescente rispetto al campo *frecency*.

Si è più volte parlato dell'attività di *pattern matching*. Essa consiste nel confrontare la stringa digitata dall'utente con i campi *URL* e *Title* di tutti i record presenti nella tabella *moz_places*. Per eseguire quest'operazione, Firefox utilizza gli strumenti di interrogazione forniti da *SQLite* e concretamente esegue delle operazioni di SELECT con operatore LIKE, proprie del linguaggio SQL.

La tabella *moz_places* viene aggiornata non solo all'apertura del browser, ma anche quando è in modalità *idle*, cioè quando l'utente non lo sta utilizzando direttamente. Durante queste fasi, alcuni record di cronologia vengono cancellati (ad esempio quelli con data di ultimo accesso antecedente al limite consentito dal periodo temporale di permanenza in cronologia impostanto dall'utente), mentre altri vengono aggiornati (ad esempio il valore di *frecency* che è correlato al tempo).

[7] Una favicon, contrazione inglese di favorite icon, è una piccola immagine associata alla pagina web e che ne rappresenta il logo o il contenuto. In Firefox viene visualizzata a sinistra nella barra degli indirizzi o accanto al nome delle pagine nei Preferiti.

[8] La definizione del parametro *frecency* viene data nella sezione *2.3 Studio del parametro* frecency.

2.3 Studio del parametro *frecency*

L'ordinamento dei documenti proposti dalla *awesome bar* (cioè quelli che presentano un match con la stringa digitata dall'utente) è stilato sulla base di un parametro chiamato *frecency* [20]. Come suggerisce il nome, questa misura sintetizza una miscela fra frequenza (in inglese: *frequency*) e freschezza temporale (in inglese: *recency*[9]). *Frecency* assume valori interi positivi in un range teoricamente non limitato superiormente. Più alto è il valore di *frecency*, più in alto il documento corrispondente comparirà nella lista suggerita dalla *awesome bar*. Vengono perciò premiate le pagine della cronologia che vengono visitate più spesso e che sono state visitate più di recente.

Per calcolare *frecency*, Firefox fa riferimento ai dati contenuti nel database *SQLite*, la cui tabella *moz_places* è stata descritta in dettaglio nella sezione precedente. Segue una descizione ad alto livello del processo di calcolo di *frecency* [21]:

```
FOR ciascuna delle 10 visite più recenti:
   determinaBonus();
   determinaPeso();
   calcolaPunteggioVisita();
END FOR
calcolaFrecency();
```

Procedura *determinaBonus*

Assegna un bonus sulla base della tipologia di visita, secondo la casistica seguente:

- 120 se la visita è avvenuta attraverso un collegamento ipertestuale (un *link*)
- 200 se la visita è avvenuta attraverso digitazione dell'URL da parte dell'utente nella barra degli indirizzi
- 140 se la visita è avvenuta attraverso l'utilizzo dei segnalibri (preferiti)
- 0 se la visita è di altra natura

[9] Un sistema che propone l'utilizzo di *recency* come parametro per l'ordinamento di documenti in ambito di web search è stato studiato recentemente da Dong et al. per Yahoo![11].

Procedura *determinaPeso*

Assegna un peso alla visita in base a quando la visita è avvenuta, secondo la casistica seguente (viene selezionato il peso più basso):

- 100 se la visita è avvenuta negli ultimi 4 giorni
- 70 se la visita è avvenuta negli ultimi 14 giorni
- 50 se la visita è avvenuta negli ultimi 31 giorni
- 30 se la visita è avvenuta negli ultimi 90 giorni
- 10 se la visita è avvenuta oltre 90 giorni prima

Funzione *calcolaPunteggioVisita*

Calcola un punteggio complessivo per la visita in esame. Indicando con

- *bonus* il valore selezionato nella procedura *determinaBonus()*;
- *weight* il valore di peso selezionato nella *procedura determinaPeso()*;

$$VisitScore = \frac{bonus}{100}\, weight$$

Funzione *calcolaFrecency*

Dopo aver calcolato gli score per le ultime 10 visite, si passa all'effettivo calcolo del valore di *frecency*. Innanzi tutto si precalcola la somma dei punteggi delle 10 visite, ottenendo il valore *SumOfScores*. La formula da usare è la seguente:

$$Frecency = TotalVisitCount\, \frac{SumOfScores}{NumberOfSampledVisits}$$

È bene precisare ulteriormente il significato dei tre fattori che la compongono:

- *TotalVisitCount* è il numero complessivo di visite ricevute dalla pagina in oggetto (per definizione è maggiore o uguale a *NumberOfSampleVisits*);

- *NumberOfSampledVisits* è l'effettivo numero di visite recenti su cui è stato eseguito il calcolo di *frecency*; è un intero che varia da 1 a 10, in quanto come già detto il calcolo di *frecency* è limitato alle 10 visite più recenti. È chiaro che se una pagina è stata visitata meno di dieci volte, questo valore sarà inferiore a 10;

- *SumOfScores* è la somma dei punteggi (scores) calcolati per le *NumberOfSampledVisits* visite più recenti;

La formula sopra riportata mostra come il valore di *frecency* aumenti all'aumentare della frequenza di visita della pagina, riassunta dal fattore *TotalVisitCount*. Il fattore *SumOfScores* tiene invece conto, al di là dei bonus relativi alla tipologia di visita, dell'aspetto temporale (*recency*), ossia di quando le visite sono avvenute. La divisione per *NumberOfSampledVisits* serve ad eseguire una media degli score sul numero di visite effettive sulle quali è esteso il calcolo di *frecency*.

Esempio 1

Si supponga di voler calcolare il valore di *frecency* per una pagina presente nei segnalibri e visitata complessivamente 4 volte, di cui una volta ieri, un'altra una settimana fa (arrivandoci cliccando da un *link*) e le rimanenti due volte più di 90 giorni fa.

- (Visita di ieri) Weight: 100, Bonus: 140 (bookmark).

$$Score = 100 * 140/100 = 140$$

- (Visita di una settimana fa) Weight: 70, Bonus: 120 (link).

$$Score = 70 * 120/100 = 84$$

- (Visita di più di 90 giorni fa) Weight: 10, Bonus: 140 (bookmark).

$$Score = 10 * 140/100 = 14$$

- (Visita di più di 90 giorni fa) Weight: 10, Bonus: 140 (bookmark).

$$Score = 10 * 140/100 = 14$$

$$SumOfScores = 140 + 84 + 14 + 14 = 252$$

$$Frecency = TotalVisitCount\ \frac{SumOfScores}{NumberOfSampledVisits} = 4\ \frac{252}{4} = 252$$

Esempio 2

Si supponga di voler calcolare il valore di *frecency* per una pagina visitata molto spesso e recentemente. La pagina è stata visitata complessivamente 50 volte, di cui le ultime 10 sempre attraverso i segnalibri negli ultimi 2 giorni.

Calcolo dello score per le ultime 10 visite. Weight: 100 (perché avvenute entro i 4 giorni precedenti), Bonus: 140 (bookmark).

$$Score = 100 * 140/100 = 140$$

$$SumOfScores = 10 * 140 = 1400$$

Calcolo di *frecency*:

$$Frecency = TotalVisitCount\ \frac{SumOfScores}{NumberOfSampledVisits} = 50\ \frac{1400}{10} = 7000$$

Capitolo 3

Sviluppo dell'applicazione *Awesome++*

Dopo aver studiato come Firefox implementa la propria *Awesome bar*, è facile intuire le analogie fra la ricerca nella cronologia web e quella in una ipotetica cronologia su documenti locali. Per ritrovare e proporre all'utente documenti presenti nel *desktop* dell'utente, si può usare la *awesome bar*, a patto di riuscire a inserire nella cronologia i documenti locali. È necessario anche adattare i parametri usati nello *storage* e nella ricerca delle pagine web alle peculiarità di documenti non web.

3.1 Scelte progettuali

L'utilizzo della *awesome bar* di Firefox come strumento di ricerca e di ordinamento dei documenti impone di trattare i documenti locali alla stregua delle pagine web. Per questo motivo a fini di indicizzazione e di ricerca, non viene letto il contenuto dei file locali, ma ci si limita a considerare il *path* (cioè il percorso all'interno del *file system*) e il *filename* (il nome del documento in esame), compreso di estensione. Sarà proprio su *path* e *filename* che verrà eseguito il *pattern matching*, analogamente a quello che Firefox esegue sulla cronologia web per i campi *URL* e *Title*.

I documenti locali si possono classificare in base alle estensioni dei file che ne individuano la tipologia. Vista la natura sperimentale del progetto e il suo *testing* in ambito accademico, si è deciso di limitarsi alle tipologie di documenti più diffuse. Sono stati presi in considerazione documenti di *Word* e di *Power Point* del pac-

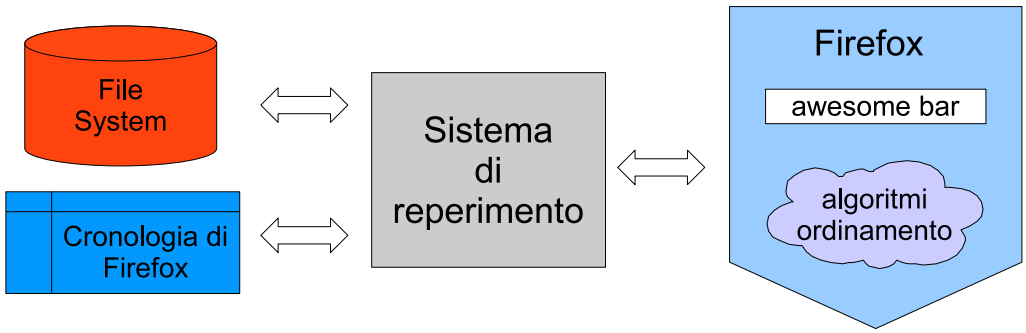

Figura 3.1: Schema di progetto del nuovo sistema di reperimento

chetto Office di casa *Microsoft*® e degli analoghi *Writer* e *Impress* di *OpenOffice*, di *Sun Microsystems*®. Inoltre sono stati inclusi, proprio per la loro diffusione, i formati PDF (*Portable Document Format*) e *PostScript* ideati da *Adobe Systems*®. Inizialmente era stato prevista anche l'elaborazione dei file di testo semplice o in formato *Rich Text Format*, poi esclusa perché poco utilizzata dagli utenti ma largamente utilizzata per produrre i cosiddetti *readme* o le guide di numerosi software. Il loro impiego avrebbe aumentato di molto il numero di file da indicizzare, senza che essi si rivelassero davvero utili, causando *overhead* nelle fasi di elaborazione.

Volendo valutare il ritrovamento di documenti nel proprio *desktop*, si dovranno confrontare configurazioni del sistema che differiscono per le collezioni di documenti da usare come input. Non si è voluti intervenire perciò sugli algoritmi e le tecniche di ordinamento che Firefox usa per ricercare nella cronologia e per proporre nella sua *awesome bar*, poiché esse non sono variabili da valutare. Di conseguenza, nel progetto si utilizza Firefox così com'è per tutto ciò che riguarda l'ordinamento e la presentazione dei risultati, andando invece ad eseguire opportune operazioni di integrazione, formattazione e adattamento dei dati relativi ai documenti locali in modo tale che il browser li gestisca in modo del tutto analogo a quelli delle pagine web presenti in cronologia.

3.2 Problemi affrontati

Nella fase di sviluppo e di adattamento delle procedure di elaborazione dei file locali, si sono affrontati alcuni sottoproblemi che si è ritenuto opportuno descrivere

nelle sottosezioni seguenti.

3.2.1 Gestione e analisi del database

Per integrare la cronologia web con dati dei documenti *desktop* è necessario accedere in scrittura al database *places.sqlite* memorizzato nel profilo[1] di Firefox. Il database è usato nel normale funzionamento del browser, perciò risulta *locked* per qualsiasi tentativo di accesso da parte di applicazioni esterne. Il comportamento di bloccaggio di Firefox è del tutto normale e serve per garantire integrità e consistenza dei dati interni al database stesso. Per ovviare a questo problema, è necessario che la fase di integrazione della cronologia con documenti locali avvenga quando il browser non è in funzione. In questo modo il database risulterà *unlocked* e quindi accedibile per operazioni di scrittura. Questa scelta si rivela necessaria ma non limitante, in quanto per il funzionamento dell'applicazione *Awesome++* e per gli obiettivi di progetto non è necessario che la fase di indicizzazione avvenga in tempo reale. Il *tool* è infatti stato progettato e sviluppato a scopo accademico e non a scopi industriali o professionali. Il caso limite in cui un utilizzatore provi a usare lo strumento per cercare un documento locale appena creato, con browser attivo, porta ad un semplice non reperimento del nuovo documento, in quanto la fase di indicizzazione è da eseguirsi manualmente prima dell'avvio di Firefox.

Il problema successivo è creare i nuovi record da inserire nella tabella *moz_places* del database affinché Firefox li tratti esattamente come se fossero genuini e alla stessa stregua dei record relativi alla storia di navigazione dell'utente. Per perseguire questo obiettivo è sufficiente costruire dei record che abbiano la stessa struttura di quelli della tabella *moz_places*, riempiti però con i dati dei file locali. Il record di un documento locale varia rispetto a quello di una pagina web (si faccia riferimento al paragrafo *3.2 Gestione della cronologia*) per il contenuto dei seguenti campi:

[1]Pochi sono a conoscenza della possibilità di creare più profili in Firefox. La loro ideazione era pensata originariamente per permettere a diversi utenti (usanti lo stesso computer) di usare il browser con le proprie impostazioni preferite e i propri bookmark. Firefox usa implicatamente un profilo, chiamato profilo di default, in cui memorizza alcuni settaggi, i Preferiti, la cronologia e le sessioni di utilizzo del browser.

- *url*: l'URL della pagina è il path compreso di *filename*, con un prefisso di tipo *file://* (ad esempio `file://C:/documenti/prova.doc`);
- *title*: è riempito dalla stringa corrispondente al filename, compreso di estensione del documento (ad esempio `prova.doc`);
- *rev_host*: contiene la stringa `'.'` che è il valore di default dato da Firefox ai documenti locali visualizzati nel browser;
- *frecency*: è il valore di *frecency* calcolato per i documenti locali;
- *last_visit_date*: viene memorizzata la data di ultima modifica del documento locale, in formato *timestamp*;

La discussione dei valori da assegnare ai campi *frecency* e *last_visit_date* è lasciata ai due paragrafi seguenti.

3.2.2 Data di modifica o data di ultimo accesso

Sapendo che *frecency* tiene conto dell'aspetto temporale delle visite alle pagine web, è cruciale stabilire se per un documento locale ha più significato la data di ultimo accesso al documento o la data di ultima modifica dello stesso. Lo sviluppo di un piccolo *tool* software ad hoc ha mostrato che la data di ultimo accesso di un documento risulta poco significativa. In molti casi la data di ultimo accesso non corrisponde all'effettivo ultimo istante di utilizzo del file da parte dell'utente: le funzioni fornite dal sistema operativo per l'esplorazione visuale delle directory modificano la data di ultimo accesso ai file contenuti. Questo si traduce in un'ulteriore anomalia: quasi tutti i documenti contenuti in una directory hanno la stessa data di ultimo accesso. L'appiattimento delle date di ultimo accesso per i file interni alla stessa cartella falserebbe la componente temporale del calcolo di *frecency*. La classificazione temporale eseguita nella procedura *determinaPeso* porterebbe ad avere pesi identici per i documenti interni alla medesima directory, ma questo peso sarebbe causato da operazioni legate al sistema operativo e non all'attività effettiva dell'utente. Questi motivi ed altri più squisitamente legati all'adattamento di *frecency* per i documenti locali (si veda paragrafo successivo) hanno portato alla decisione

di usare la data di ultima modifica dei documenti, a discapito di quella di ultima visita.

3.2.3 Calcolo di *frecency* per i documenti locali

Come già detto nel paragrafo *2.3 Studio del parametro* frecency, quella di *frecency* è una misura che tiene conto sia della frequenza di visita della pagina web, sia della freschezza temporale con cui le visite sono avvenute. Nel tentativo di adattare questa definizione di *frecency* ai documenti locali, è evidente che la componente di frequenza non può essere derivata dal *file system*, a meno di costruire un analizzatore delle attività dell'utente; ciò implicherebbe lo studio e l'implementazione di un software in grado di installarsi nel sistema operativo per ricavare i *log* dell'utente. Poiché ciò esula lo scopo di questo progetto, si è dovuto procedere all'adattamento di *frecency* usando la sola componente temporale.

La frequenza di visita di una pagina web dà una misura della sua utilizzazione. Una pagina web però non è modificabile dall'utente (è accessibile solo in lettura[2]) e da qui nasce la necessità di ricorrere alla frequenza di visita. Viceversa un documento locale ha la possibilità di essere acceduto sia in lettura che in scrittura, ottenendo una preziosa informazione sull'attività dell'utente. Per i documenti *desktop* perciò si sfrutta l'aspetto temporale per avere un buon indice di utilizzazione. La modifica del file infatti avviene quando l'utente concretamente esegue azioni di scrittura del documento (creazione, copia, *download*, salvataggio, etc.). L'uso di questo valore come unica variabile per calcolare *frecency* fornisce perciò indicazioni implicite sull'attività dell'utente.

Il calcolo di *frecency* per i documenti locali può essere notevolmente semplificato, avendo a disposizione il solo valore di data di ultima modifica del file. Si adotta una procedura del tutto simile a *determinaPeso*, con limiti temporali estesi (quasi sempre raddoppiati) per tenere conto che si tratta di tempistiche di modifica (non di ultimo accesso) e che si fa riferimento a documenti presenti nel *file system* aventi tempi di persistenza nel sistema molto più lunghi.

[2]Ciò è vero in generale, eccezion fatta per i siti web in cui gli utenti contribuiscono alla creazione dei contenuti (un esempio su tutti: Wikipedia).

Procedura per il calcolo di *frecency* per i documenti locali

Assegna uno score temporale, secondo la casistica seguente (viene selezionato lo score più basso):

- 100 se la visita è avvenuta negli ultimi 14 giorni
- 70 se la visita è avvenuta negli ultimi 31 giorni
- 50 se la visita è avvenuta negli ultimi 90 giorni
- 30 se la visita è avvenuta negli ultimi 180 giorni
- 10 se la visita è avvenuta oltre 180 giorni prima

I nuovi limiti temporali sono stati ricavati attraverso numerose prove pratiche del nuovo algoritmo su una collezione di documenti locali di test. Essi hanno mostrato di essere la migliore combinazione di intervalli temporali atta a garantire un equilibrio fra il concetto di attività recente e la necessità di far risalire i documenti locali nelle prime posizioni della lista di documenti suggeriti dalla *awesome bar*. I valori degli *score* sono invece esattamente gli stessi utilizzati da Firefox nell'implementazione dell'algoritmo di calcolo di *frecency*. Lo score temporale deve poi diventare un valore di *frecency*. A questo fine lo si adatta al range di valori di *frecency* già presenti nella cronologia. In questo modo i valori di *frecency* per i documenti locali risultano confrontabili con quelli delle pagine web già presenti nella cronologia. Assumendo che *score* sia il valore selezionato dalla procedura precedente e che $MaxFrecency$ sia il massimo valore di *frecency* calcolato sulla cronologia web, il valore di *frecency* per un documento locale viene calcolato usando la seguente formula:

$$LocalFrecency = \frac{score \cdot MaxFrecency}{100}$$

La costante 100 serve a rapportare il valore *score* all'ampiezza dell'intervallo su cui *score* stesso varia. Si noti che il calcolo di *frecency* non è macchinoso quanto quello per una pagina web, in quanto per i documenti *desktop* viene meno il concetto di visita e di pesatura delle ultime dieci visite precedenti.

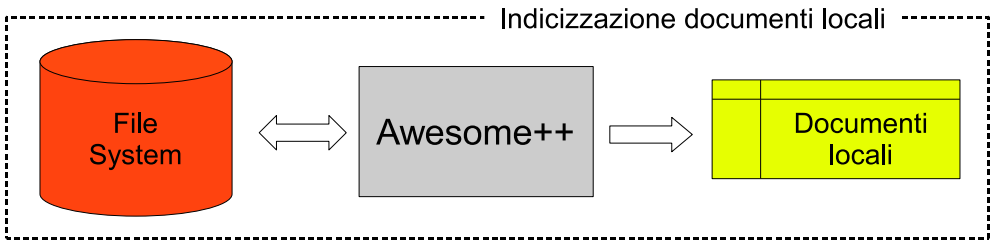

Figura 3.2: Schema concettuale per la fase di scansione del *file system*

Le scelte eseguite in questa fase di modellazione di *frecency* sono cruciali, poiché modificano l'ordinamento dei documenti proposti dalla *awesome bar*. Al tempo stesso questo aspetto è pure quello più difficile e dove numerose migliorie e varianti possono essere apportate.

3.3 Cenni implementativi

L'applicazione *Awesome++* è stata scritta e sviluppata in linguaggio *Java* per la sua natura di linguaggio multipiattaforma, per le caratteristiche di robustezza e la possibilità di usare le API[18] per la gestione dei *database SQLite*. Le sue funzioni principali sono le seguenti:

- indicizzare il *file system* dell'utente allo scopo di individuare i documenti da inserire in cronologia
- integrare la cronologia web con i documenti locali selezionati con la procedura precedente

Indicizzare il *file system*

Questa fase consiste nell'esplorazione automatizzata del *file system*, alla ricerca dei documenti locali con cui successivamente integrare la cronologia. A questo scopo è stata sviluppata una procedura ricorsiva che a partire da una directory o da un drive del *file system*, esplora in profondità tutte le subdirectory contenute. La visita dell'albero delle sottocartelle avviene in modalità *depth-first*.

Alcune procedure sono state sviluppate per l'elaborazione di *path* e *filename*, in modo da ottenere il solo nome del file o la sola estensione. Ad esempio la procedura

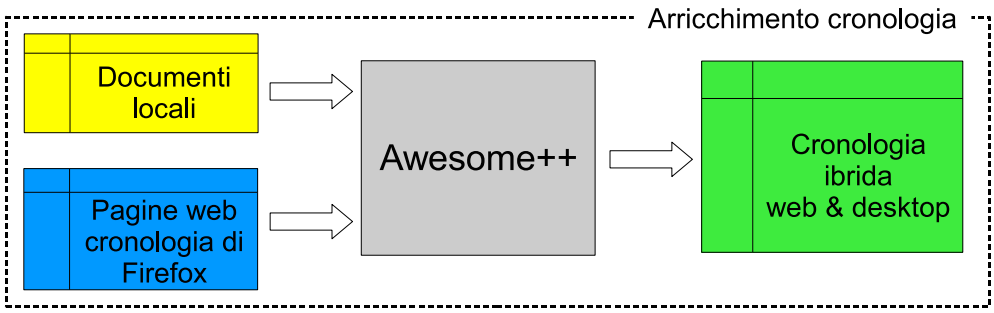

Figura 3.3: Schema concettuale per la fase di integrazione della cronologia web con i documenti locali, al fine di ottenere una cronologia ibrida

di selezione dei documenti usa l'estensione dei file per capire se il file in esame corrisponde a una delle tipologie di documenti da indicizzare.

Integrare la cronologia

La cronologia memorizzata nel database *places.sqlite* viene integrata con l'elenco dei documenti selezionati nella procedura di scansione del *file system*. È stato necessario usare le API di *Java* per la gestione delle basi di dati di tipo *SQLite*.

Per questioni di *locking* del *database*, questa procedura deve essere eseguita con Firefox chiuso. I record da inserire in cronologia vengono costruiti uno alla volta a partire dall'indice costruito nella fase precedente. In particolare il valore di *frecency* di ciascun documento locale viene calcolato in questa fase, con una funzione ad hoc avente come parametro di input la data di ultima modifica del file. In mancanza di una procedura di *loading* di più record nel database, i record devono essere inseriti uno alla volta. La fase di inserimento dei record attraverso ripetute istruzioni SQL di INSERT è quella più dispendiosa in termini di tempo.

Al fine di rimuovere eventuali record di documenti locali già presenti nel database ad opera di sessioni precedenti completate o parzialmente interrotte e anche per questioni di *debugging*, è stata creata anche una procedura di pulizia della cronologia per le sole voci riguardanti documenti locali. L'ispezione della base di dati è stata possibile grazie all'estensione per Firefox chiamata *SQLite Manager*[19].

Capitolo 4

Fase di valutazione

Lo sviluppo del tool *Awesome++* per l'integrazione della cronologia web con documenti presenti nel *desktop* dell'utente permette di ottenere un sistema di reperimento che ha come collezione di documenti un insieme ibrido[1] di pagine web e di documenti locali. Una volta che l'applicazione è stata sviluppata, ed averne verificato il corretto funzionamento, è possibile ideare un esperimento per valutare le prestazioni del nuovo sistema di reperimento dell'informazione.

Quando ci si accinge ad eseguire una valutazione nell'ambito dell'*Information Retrieval* solitamente, oltre al sistema stesso, ci si munisce di una collezione sperimentale costituita da una collezione di documenti, corredata da giudizi di rilevanza e da un insieme di query. Alcune collezioni sperimentali sono disponibili, altre vengono proposte da istituzioni come *TREC*[22] (*Text REtrieval Conference*). Tuttavia al di là delle possibilità di ottenere collezioni sperimentali costruite professionalmente, per questo progetto sarebbe difficile trovare collezioni adatte a modellare e valutare lo specifico *task* in esame. Un'ulteriore difficoltà sarebbe quella di riuscire a interfacciare il software sviluppato e Firefox con collezioni sperimentali ad esempio fornite in formato *TREC*.

Questo porta alla necessità di impostare un esperimento ad hoc, attraverso la formazione di un gruppo di valutazione che sia in grado sia di dare giudizi di rilevanza sia di valutare i documenti proposti dalla *awesome bar*. Verrà successivamente sotto-

[1] Va precisato che, per com'è stato costruito il sistema, la collezione è da ritenersi ibrida anche perché i documenti locali sono concretamente presenti nella collezione, mentre le pagine web sono rappresentate esclusivamente dai record presenti nella cronologia del browser (non fisicamente dalle pagine HTML).

lineata l'esigenza di usare una collezione di documenti uguale, almeno inizialmente, per tutti i membri del gruppo di valutazione.

4.1 Descrizione dell'esperimento

Per valutare le prestazioni del sistema sviluppato in questo progetto, sono state messe a punto due configurazioni. Nell'analisi dei risultati si procederà poi al confronto fra le due, in modo da evidenziare le differenze e i miglioramenti prestazionali ottenuti.

Configurazione con cronologia base

La configurazione iniziale serve a creare una base di confronto sul quale successivamente misurare le variazioni prestazionali. Nella fattispecie, gli utenti del gruppo di valutazione sono chiamati a valutare attraverso giudizi di rilevanza i documenti proposti dalla *awesome bar* di Firefox quando la collezione di documenti è una cronologia web uguale per tutti gli utenti.

In questo modo non solo si metteranno tutti gli utenti nelle stesse condizioni iniziali, ma si otterrà la costruzione di quei giudizi di rilevanza necessari per formare la collezione sperimentale necessaria per la valutazione.

Configurazione con cronologia arricchita coi documenti locali

Questa configurazione del sistema prevede l'utilizzo del tool *Awesome++* per l'integrazione della cronologia base (la stessa usata nella configurazione con cronologia base) con i documenti locali presenti nel *desktop* dell'utente e opportunamente indicizzati dal tool.

I membri del gruppo di valutazione sono chiamati a dare giudizi di rilevanza sui documenti proposti dalla *awesome bar* di Firefox, segnalando opportunamente i documenti locali presenti nella lista.

4.2 Costruzione della collezione sperimentale

Le due configurazioni del sistema si caratterizzano per la differenza delle collezioni di documenti utilizzate:

- la *cronologia base* è una cronologia web uguale per tutti gli utenti
- la *cronologia arricchita* è la cronologia base integrata con i documenti locali presenti nel *desktop* di ciascun partecipante al gruppo di valutazione; ciò implica che la cronologia arricchita sia diversa per ciascun utente per quanto riguarda i documenti di tipo *desktop*

Raccolta delle query

Per costruire la cronologia base che deve essere uguale per tutti è stato chiesto ai dieci partecipanti al gruppo di valutazione di fornire cinque query che riguardino la loro attività recente o i loro interessi oppure che siano query molto frequenti per le loro esigenze informative quotidiane. In questo modo sono state ottenute $10 \cdot 5 = 50$ query. È stato richiesto che le interrogazioni fossero di 1 o 2 parole in italiano, eccezion fatta per parole tecniche (come computer o altri termini inglesi intraducibili). Il limite di lunghezza delle query è stato consigliato dal funzionamento della *awesome bar* che esegue *pattern matching* esatto fra parole delle query e le voci della cronologia. È stato deciso di usare la lingua italiana supponendo che i membri del team di valutazione usino l'italiano nella scelta dei path e dei filename dei propri documenti locali.

Altre 30 query piuttosto generiche sono state aggiunte per arricchire il bacino di query, senza tenere conto delle indicazioni degli utenti. Complessivamente sono state raccolte 80 query tutte diverse su cui eseguire la valutazione.

Costruzione della cronologia base

Per fornire una cronologia di base uguale per tutti gli utenti e di natura web, è stata sviluppata un'ulteriore applicazione ausiliaria. L'idea è quella di interrogare Google® con le 80 query raccolte in fase preliminare e di salvare i primi 16 risultati

di ogni interrogazione. In questo modo si assicura che al momento della valutazione, gli utenti ricevano come suggerimento una lista non vuota di pagine web.

Il valore di *frecency* assegnato alle pagine web presenti nella cronologia base è assunto avere un andamento decrescente rispetto alla posizione del documento nella lista dei risultati del motore di ricerca (più alta è la posizione in lista, più alto è il valore di *frecency* modellato per la pagina web in quella posizione). La funzione usata per la stima del valore iniziale di *frecency* da dare ai documenti della cronologia base è una funzione valutata per soli valori positivi e usata spesso in *Information Retrieval* per modellare funzioni analoghe.

$$f = C \cdot r^{-k}$$

Ove $k = 2.1$ e $C = 1000$ sono delle costanti stimate sperimentalmente e r è la posizione (*rank*) della pagina all'interno della lista dei risultati. In particolare si è notato, attraverso numerosi tentativi, che il valore di k fissato a 2.1 permetteva di avere una buona[2] distribuzione dei valori di *frecency*, sfruttando un'analogia con le stime del numero di click che gli utenti effettuano sulle pagine web proposte nei risultati dei motori di ricerca, in relazione alla loro posizione. È stato verificato in studi precedenti [12], infatti, che i primi risultati dei motori di ricerca sono visitati con frequenza elevata e il numero di click sulle pagine successive decresce secondo una funzione analoga a quella proposta. Dovendo costruire una cronologia che simuli valori di *frecency* dovuti al comportamento dell'utente, si è assunto che il comportamento degli utenti nei confronti dei risultati nella ricerca web modelli in modo sufficientemente accurato il grado di utilizzo delle pagine web presenti in cronologia e indirettamente il valore di *frecency* loro assegnato.

4.3 Descrizione del gruppo di valutazione

Fino ad ora si è parlato piuttosto genericamente di un gruppo di valutazione. È necessario però dare una descrizione più dettagliata di com'è composto.

[2]Si assume ragionevolmente che i valori di frecency associati a pagine dello stesso topic (presenti in una cronologia web) siano distribuiti a valori decrescenti secondo il grado di utilizzo (e di utilità) delle pagine stesse.

Sono state selezionate dieci persone fra laureandi, laureati o dottorandi, la maggior parte dei quali afferenti al dipartimento di Ingegneria dell'Informazione di questa università e operanti in ambito ingegneristico nell'area dell'Informazione.

L'età è omogenea e compresa fra i 24 e i 25 anni mentre la suddivisione fra ragazzi e ragazze è decisamente squilibrata: nove maschi e una sola femmina.

La ridotta dimensione del gruppo di valutazione è in linea con esperimenti preliminari analoghi eseguiti in alcuni laboratori di ricerca ma offre già una sufficiente significatività statistica dei risultati ottenuti, anche tenendo conto dell'ampia numerosità delle query.

4.4 Modalità di conduzione dell'esperimento

Ai membri del team di valutazione è stato fornito un *kit* contenente le istruzioni dettagliate per eseguire la valutazione, il database contenente la cronologia base, una copia dell'applicazione *Awesome++* e un foglio elettronico dove riportare i risultati precompilato con le query assegnate. La parte operativa che gli utenti hanno dovuto affrontare si può suddividere in tre fasi:

1. Configurazione e testing del sistema

2. Valutazione della cronologia base

3. Valutazione della cronologia arricchita con documenti locali

I membri del gruppo di valutazione sono stati chiamati ad eseguire in 9 giorni di tempo tutte e tre le fasi nell'ordine indicato, senza alcun tipo di supervisione. A chi l'ha richiesto sono state date spiegazioni tecniche allo scopo di risolvere piccoli problemi di installazione o configurazione o per chiarire le procedure operative da eseguire.

Nelle sottosezioni successive verrà sintetizzato il contenuto di ciascuna delle tre fasi. Il paragrafo qui di seguito è invece dedicato al metodo con cui le query son state assegnate e distribuite ai membri del team di valutazione.

Distribuzione delle query

Si è scelto di assegnare a ciascun utente un insieme di 10 query scelte fra le 80 raccolte in fase preliminare. L'assegnazione non è stata casuale ma ha seguito una certa struttura. Assumendo di usare la seguente notazione per la lista di query assegnate alla persona p_1:

$$Q_{p_1} = (q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10})$$

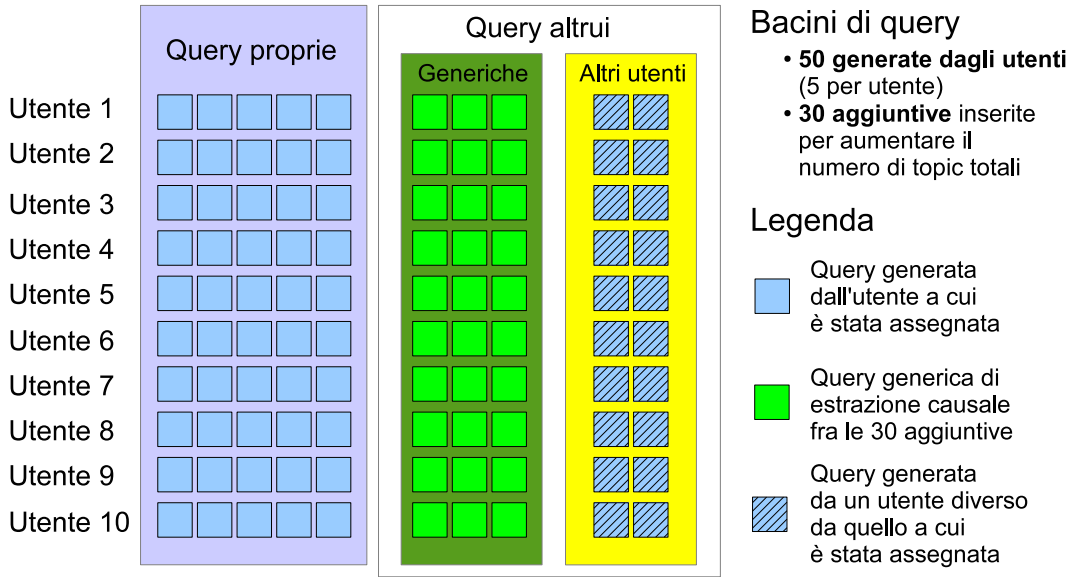

Figura 4.1: Schema di distribuzione delle query ai dieci utenti del gruppo di valutazione

La sottolista $(q_1, \ldots, q_5)$ corrispondente alle prime cinque query, è formata interamente dalle query fornite dall'utente p_1. Il sottoinsieme (q_6, q_7, q_8) è stato assegnato in modo casuale prendendo elementi dall'insieme delle 30 query non fornite dagli utenti. Le ultime due query della lista, (q_9, q_{10}), sono state pescate in modo casuale dalle query fornite dagli utenti (escluse quelle fornite dallo stesso p_1).

In questo modo ciascun utente si è trovato a valutare una lista di query che per metà proviene dall'utente stesso, per 2/10 è di altri membri del gruppo di valutazione e per i rimanenti 3/10 è costituita da query generiche generate in fase di ideazione dell'esperimento.

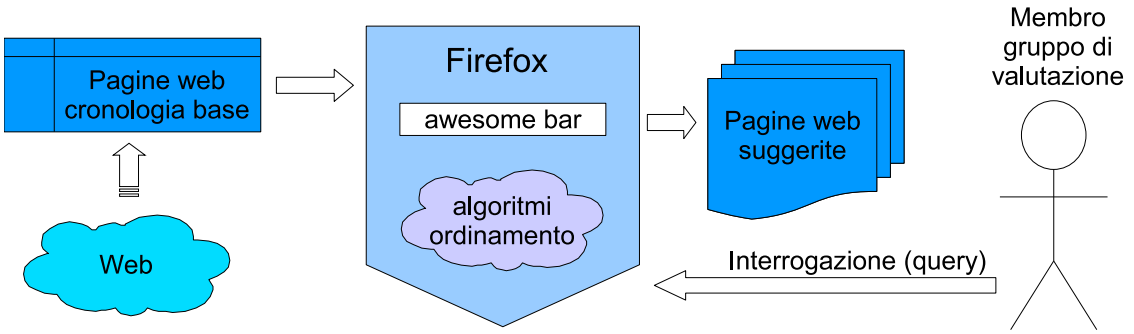

Figura 4.2: Diagramma funzionale per la valutazione della cronologia base

4.4.1 Configurazione e testing del sistema

In questa fase i partecipanti al gruppo di valutazione sono chiamati a configurare il proprio sistema di lavoro, eseguire alcuni settaggi di Firefox e ad installare lo strumento *Awesome++* per l'integrazione della cronologia con documenti locali. In particolare a ciascuno di loro è stato richiesto di:

- Verificare la presenza delle corrette versioni di Firefox e della Java Virtual Machine
- Localizzare il database *SQLite* del proprio profilo di Firefox che andrà sostituito con quello fornito e contenente la cronologia base
- Entrare nei dettagli di configurazione di Firefox necessari al corretto funzionamento della *awesome bar*
- Installare e verificare il funzionamento dell'applicazione *Awesome++* eseguendo un breve test

Alla termine di questa prima fase l'utente non solo avrà individuato tutti i componenti software con cui dovrà agire, ma si sarà già impratichito con l'interfaccia di *Awesome++* a sufficienza per avere una prima idea di come verranno eseguite a livello tecnico le due successive fasi di valutazione.

4.4.2 Valutazione della cronologia base

Per eseguire una valutazione è necessario dare i cosiddetti giudizi di rilevanza. In questa fase gli utenti sono invitati a dare giudizi di rilevanza dei documenti presenti nella cronologia base che sono di provenienza web. La rilevanza di un documento in generale è data dalla capacità del documento di rispondere all'esigenza informativa dell'utente. Tuttavia questo progetto è basato sulla ricerca in cronologia per mezzo della *awesome bar*. L'utilizzo che l'utente fa della *awesome bar*, oltre a usarla come mera barra degli indirizzi, è quello di poter ritrovare una pagina web già vista in passato e di cui si è dimenticato l'indirizzo. Dovendo estendere queste funzioni anche ai documenti locali, è necessario dare una interpretazione adatta all'esperimento. Si arriva perciò a definire *rilevante* un documento o una pagina web che:

- è quella già vista in passato e/o di cui si ha dimenticato l'indirizzo
- contiene frasi, immagini, contenuti multimediali attinenti alle parole della query

La natura della cronologia base che è sconosciuta agli utenti e per poter cogliere variazioni anche piccole nell'espressione dei giudizi di rilevanza, è stata adottata una scala a tre valori:

0 - pagina o documento non rilevante

1 - pagina o documento abbastanza rilevante

2 - pagina o documento molto rilevante

Operativamente, l'utente deve impostare Firefox come previsto nella precedente fase di configurazione, avendo cura di usare il database contenente la cronologia base. La valutazione vera e propria dei risultati di ciascuna query può essere sintetizzata in una procedura come quella seguente.

Per ogni query delle dieci a lui assegnate:

1. L'utente digita completamente (non parzialmente) la query nella *awesome bar* e attende pochi istanti affinché Firefox produca la lista dei primi dieci documenti suggeriti

2. Per ogni pagina proposta, l'utente ne valuta la rilevanza eventualmente navigandola

3. Per ogni pagina proposta, l'utente scrive il giudizio di rilevanza nella tabella riassuntiva del foglio elettronico, nella casella corrispondente tra la query corrente e la posizione del documento nella lista dei suggerimenti

Al termine l'utente avrà compilato una tabella 10 x 10 riempita coi giudizi di rilevanza dei primi dieci documenti di ciascuna delle dieci query. Se una query produce una lista di documenti di dimensione inferiore a dieci, l'utente avrà lasciato in bianco le caselle in coda.

4.4.3 Valutazione della cronologia arricchita con documenti locali

La valutazione della cronologia arricchita deve avvenire con la cronologia base integrata con i documenti *desktop* dell'utente. È necessario perciò l'utilizzo dell'applicazione *Awesome++*. La definizione di documento rilevante è esattamente la stessa della fase precedente. All'utente però viene chiesto di segnalare i documenti di provenienza locale, scrivendo un asterisco accanto al giudizio di rilevanza nella tabella dei risultati.

Operativamente, l'utente deve impostare Firefox come previsto nella fase di configurazione, avendo cura di usare il database contenente la cronologia base. Successivamente è necessario avviare l'applicazione *Awesome++* per l'indicizzazione dei documenti sul *desktop* dell'utente e per la successiva integrazione della cronologia base con i documenti indicizzati. Alla fine di questa fase preliminare, l'utente avrà impostato Firefox in modo da lavorare con la cronologia arricchita di documenti locali. La procedura di valutazione vera e propria è del tutto analoga alla precedente.

Per ogni query delle dieci a lui assegnate:

1. L'utente digita completamente (non parzialmente) la query nella *awesome bar* e attende pochi istanti affinché Firefox produca la lista dei primi dieci documenti suggeriti

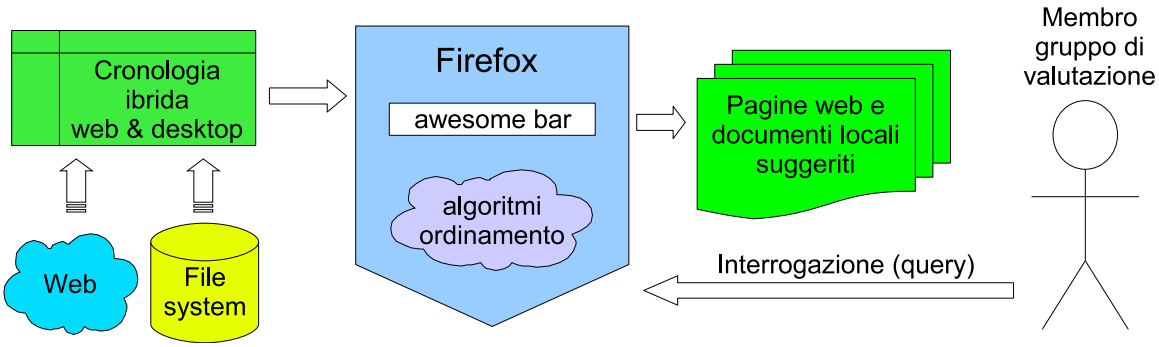

Figura 4.3: Diagramma funzionale per la valutazione della cronologia arricchita con documenti locali

2. Per ogni pagina proposta, l'utente ne valuta la rilevanza eventualmente navigandola

3. Per ogni pagina proposta, l'utente scrive il giudizio di rilevanza nella tabella riassuntiva del foglio elettronico, nella casella corrispondente tra la query corrente e la posizione del documento nella lista dei suggerimenti. L'utente apporrà il simbolo di asterisco (*) accanto ai giudizi di rilevanza relativi a documenti locali

Analogamente a prima, al termine l'utente avrà compilato una tabella 10 x 10 riempita coi giudizi di rilevanza dei primi dieci documenti di ciascuna delle dieci query. Se una query produce una lista di documenti di dimensione inferiore a dieci, l'utente avrà lasciato in bianco le caselle in coda. La particolarità di questa tabella è quella di mostrare con degli asterischi la posizione dei documenti locali nelle liste di documenti suggeriti dalla *awesome bar*.

4.5 Misure per la valutazione delle prestazioni

La natura sperimentale dell'esperimento e l'uso di collezioni sperimentali diverse per ciascun utente non permettono di usare le misure di performance come il richiamo (*recall*) o la precisione (*precision*). In particolare il richiamo non è adatto in quanto non è definito l'insieme dei documenti rilevanti presenti nella collezione sperimentale. La precisione invece potrebbe essere usata nella sua versione a rango

10 (P@10), visto che gli utenti valutano i primi dieci documenti di ciascuna lista di suggerimenti. Tuttavia una misura di precisione può essere poco significativa per la limitatezza del numero compelessivo di *run*[3] valutate e per la rigidità del parametro che necessita di avere dei giudizi di rilevanza binari (non rilevante o rilevante).

La scala a tre valori per i giudizi di rilevanza può essere facilmente interpretata e condensata da misure basate sul guadagno accumulato, come il *Direct Cumulated Gain* (CG), il *Discounted Cumulated Gain* (DCG) e le loro varianti normalizzate.

Il *Direct Cumulated Gain* è una misura che tiene conto che i documenti rilevanti nelle prime posizioni di una lista di documenti proposti dovrebbero essere ritenuti in qualche modo più rilevanti dei documenti rilevanti presenti in fondo alla lista. Ad esempio, se una query fornisce la seguente lista ordinata in termini di giudizi di rilevanza $G = (2, 1, 2, 1, 1, 1, 2, 1, 0, 2)$ il documento che corrisponde al giudizio 2 dovrebbe essere considerato più rilevante del documento che ha ottenuto il giudizio 2 presente in decima posizione.

$$CG[i] = \begin{cases} G[1], & i = 1 \\ G[i-1] + G[i], & i > 1 \end{cases}$$

Il calcolo del *Direct Cumulated Gain* per la lista dell'esempio porta ad ottenere $CG = (2, 3, 5, 6, 7, 8, 10, 11, 11, 13)$ che può essere confrontato col vettore ideale ottenibile dal calcolo del CG per il vettore $G_I = (2, 2, 2, 2, 1, 1, 1, 1, 1, 0)$, $CG_I = (2, 4, 6, 8, 9, 10, 11, 12, 13, 13)$

Il vettore normalizzato nCG, ottenuto dal rapporto fra CG e CG_I componente per componente, è chiamato *Normalized Cumulated Gain*:

$$nCG = \left(\frac{2}{2}, \frac{3}{4}, \frac{5}{6}, \frac{6}{8}, \frac{7}{9}, \frac{8}{10}, \frac{10}{11}, \frac{11}{12}, \frac{11}{13}, \frac{13}{13}\right)$$

Il *Discounted Cumulated Gain* è una misura analoga al *Direct Cumulated Gain* che in più cerca di modellare la perdita di rilevanza di un documento dovuta alla sua posizione all'interno della lista dei risultati. L'assunto è che man mano che si scende nella lista, minore è la probabilità che l'utente esamini il documento, a causa di tempo e sforzo impiegati, oltre che della conoscenza accumulata per la visita

[3]Il termine run è da intendersi come risultato di una query, cioè la lista ordinata di documenti restituiti dall'interrogazione del sistema.

dei documenti precedenti. Viene quindi inserito un fattore di sconto che progressivamente riduce il punteggio accumulato. Solitamente lo sconto avviene per mezzo della divisione del guadagno corrente per il logaritmo in base due del *rank* del documento.

$$DCG[i] = \begin{cases} G[i], & i < 2 \\ G[i-1] + G[i]/\log_2 i, & i \geq 2 \end{cases}$$

Il vettore DCG calcolato sulla lista G diventa:

$$DCG = (2, 3, 4.26, 4.76, 5.19, 5.58, 6.29, 6.63, 6.63, 7.23)$$

. Il vettore DCG_I calcolato sul vettore ideale G_I è il seguente:

$$DCG_I = (2, 4, 5.26, 6.26, 6.69, 7.08, 7.44, 7.77, 8.08, 8.08)$$

Il vettore normalizzato $nDCG$, ottenuto dal rapporto fra DCG e DCG_I componente per componente, è chiamato *Normalized Discounted Cumulated Gain*:

$$nCG = \left(\frac{2}{2}, \frac{3}{4}, \frac{4.26}{5.26}, \frac{4.76}{6.26}, \frac{5.19}{6.69}, \frac{5.58}{7.08}, \frac{6.29}{7.44}, \frac{6.63}{7.77}, \frac{6.63}{8.08}, \frac{7.23}{8.08}\right)$$

Capitolo 5

Risultati sperimentali

In questo capitolo verranno riassunti i risultati sperimentali ottenuti dal gruppo di valutazione su cui è stato condotto l'esperimento. Una prima sezione darà una breve descrizione statistica complessiva dei giudizi di rilevanza ottenuti, attraverso indici di sintesi come medie algebriche. La seconda sezione è invece dedicata alla misure delle prestazioni di efficacia del sistema sviluppato secondo i canoni dell'*Information Retrieval*.

5.1 Descrizione statistica dei risultati

Un primo sguardo ai risultati ottenuti dalle procedure di valutazione eseguite dai membri del gruppo di valutazione può essere dato osservando la *Tabella 5.1*. In essa sono riportati diversi indici che sintetizzano i soli risultati relativi ai documenti locali.

- F è la frequenza assoluta dei documenti locali, cioè il numero di documenti di provenienza *desktop* che sono stati proposti complessivamente all'utente

- S_{totale} è uno score che corrisponde alla somma dei giudizi di rilevanza ottenuti per i documenti locali su tutte le query assegnate all'utente

- $S_{proprie}$ è uno score che corrisponde alla somma dei giudizi di rilevanza ottenuti per i documenti locali sulle sole 5 query proposte dall'utente stesso

Tabella 5.1: Sintesi statistica dei risultati relativi ai documenti locali

Nome	F	S_{totale}	$S_{proprie}$	S_{altrui}
Andrea	35	69	43	26
Chiara	39	78	56	22
Francesco	11	19	12	7
Luca	14	26	24	2
Marco M.	37	65	24	41
Marco V.	32	41	37	4
Matteo	44	72	48	24
Michele	15	14	14	0
Silvio	33	66	56	10
Simone	38	73	28	45
Somma	298	523	342	181
Media	28.9			
$F_{proprie}$	189			
F_{altrui}	109			

- S_{altrui} è uno score che corrisponde alla somma dei giudizi di rilevanza ottenuti per i documenti locali sulle sole 5 query non proposte dall'utente stesso (si ricorda che 3 query sono generiche e 2 sono invece proposte da altri utenti)

- $F_{proprie}$ è il numero di documenti locali restituiti dal sistema di tutti gli utenti sulle sole query proposte dagli utenti stessi

- F_{altrui} è il numero di documenti locali restituiti dal sistema di tutti gli utenti sulle query non proprie

- *Media* è la riga dove viene riportata la media della frequenza dei documenti locali estesa a tutti gli utenti

In tabella si può notare che il numero di documenti locali ritrovati varia fra un minimo di 11 e un massimo di 44, con una media di 28.9 documenti *desktop* per utente. Le colonne $S_{proprie}$ e S_{altrui} mostrano che le somme dei giudizi di rilevanza relativi alle query fornite dagli utenti sono in genere superiori a quelle sulle interrogazioni aggiuntive altrui: il dato complessivo mostra 342 contro 181. C'è anche una tendenza a ottenere più documenti locali per le interrogazioni proprie degli utenti ($F_{proprie} = 189$) rispetto a quelle altrui ($F_{proprie} = 109$).

Tabella 5.2: Medie dei giudizi di rilevanza relativi alla cronologia per le due tipologie di documenti

Tipologia	M_{totale}	$M_{proprie}$	M_{altrui}
Documenti web	1.19	1.15	1.23
Documenti locali	1.76	1.81	1.66

La *Tabella 5.2* riporta invece i valori medi dei giudizi di rilevanza assegnati dagli utenti alle pagine web della cronologia base rispetto a quelli sui soli documenti locali. Ancora una volta si è fatta distinzione fra la media complessiva, quella relativa alle sole query proprie e quella eseguita sui soli giudizi di rilevanza dati ai risultati delle interrogazioni altrui. Dai dati si evince che la media dei giudizi di rilevanza sui documenti della cronologia base è inferiore rispetto a quella sui documenti locali. La media sui documenti generati dall'insieme delle query fornite dagli utenti è superiore rispetto a quelli dati ai documenti suggeriti usando query altrui per quanto riguarda i documenti locali (si ha la stessa tendenza evidenziata precedentemente dalle somme presenti nella *Tabella 5.1*); viceversa i membri del team di valutazione hanno valutato in media leggermente più rilevanti i documenti web sulle query non proprie rispetto a quelle fornite dagli utenti stessi.

5.2 Risultati della valutazione

Per valutare le prestazioni in termini di efficacia del nuovo sistema di reperimento, si deve procedere con il confronto fra i risultati della configurazione base e quelli ottenuti con l'utilizzo dello strumento *Awesome++*. Per chiarezza è bene sintetizzare ancora una volta in cosa differiscono i due sistemi.

Configurazione base

- *Collezione sperimentale*: cronologia base di pagine web, uguale per tutti gli utenti
- *Cosa è stato valutato*: la rilevanza dei primi 10 documenti suggeriti dalla *Awesome bar* per ciascuna query

- *Funzionamento*: la *awesome bar* propone le pagine web presenti in cronologia base, eseguendo *pattern matching* fra la query e l'*URL* o il *Title* (Titolo) della pagina, ordinando secondo il parametro *frecency*

- *Uso dei risultati*: giudizi di rilevanza da usare come base di confronto per evidenziare le performance della configurazione con *Awesome++*

Configurazione con *Awesome++*

- *Collezione sperimentale*: cronologia base di pagine web arricchita con i documenti locali degli utenti (ciascun utente ha una cronologia arricchita che differisce da quella degli altri per i soli documenti di provenienza *desktop*)

- *Cosa è stato valutato*: la rilevanza dei primi 10 documenti suggeriti dalla *Awesome bar* per ciascuna query

- *Funzionamento*: la *awesome bar* propone le pagine web presenti nella cronologia arricchita (pagine web e documenti locali), eseguendo *pattern matching* fra la query e l'*URL* o il *Title* della pagina web oppure sul *Path* e il *FileName* dei documenti locali

- *Uso dei risultati*: i giudizi di rilevanza ottenuti con la cronologia arricchita forniranno un'indicazione sull'efficacia del sistema sviluppato

Risultati: misure di efficacia

Come anticipato nella sezione *4.5 Misure per la valutazione delle prestazioni*, le misure di efficacia adottate per la sintesi dei risultati sperimentali sono il *Cumulated Gain* e il *Discounted Cumulated Gain*, comprese le loro versioni normalizzate.

Vista la notevole quantità di dati raccolti, in questo capitolo si preferiscono sintetizzare i risultati attraverso l'uso di grafici che mostrano le curve dei vettori medi relativi alle diverse configurazioni del sistema. La maggior parte dei risultati sperimentali è disponibile in forma tabellare in *Appendice A*.

Nei grafici sono mostrate sei curve corrispondenti ai risultati ottenuti dalle due configurazioni del sistema, facendo distinzione fra le query proprie e quelle atrui. La legenda è da intendersi come segue:

- *Base - Query proprie*: risultati della configurazione base, relativi alle sole query fornite dagli utenti stessi (le prime cinque query della lista assegnata a ciascun utente)
- *Base - Query altrui*: risultati della configurazione base, relativi alle sole query generiche o fornite da utenti diversi rispetto all'utente a cui sono state assegnate (le ultime cinque query della lista assegnata a ciascun utente)
- *Base complessivo*: risultati complessivi della configurazione base
- *Arricchita - Query proprie*: risultati della configurazione arricchita con documenti locali attraverso l'uso di *Awesome++*, relativi alle sole query generiche o fornite da utenti diversi rispetto all'utente a cui sono state assegnate (le ultime cinque query della lista assegnata a ciascun utente)
- *Arricchita - Query altrui*: risultati della configurazione arricchita con documenti locali attraverso l'uso di *Awesome++*, relativi alle sole query generiche o fornite da utenti diversi rispetto all'utente a cui sono state assegnate (le ultime cinque query della lista assegnata a ciascun utente)
- *Arricchita complessivo*: risultati complessivi della configurazione arricchita con documenti locali attraverso l'uso di *Awesome++*

Dal grafico di *Figura 5.1* mostra le curve di *Cumulated Gain*. Chiaramente si ha un andamento crescente, poiché man mano che la posizione del documento aumenta si accumulano i giudizi di rilevanza di nuovi risultati. Le due linee con tratto più spesso mostrano che il giudizio di rilevanza medio accumulato a ciascun *rank* è superiore per il sistema con cronologia arricchita, rispetto al sistema base. Le curve sono molto ravvicinate solo nella prima posizione ($\Delta = 0.05$), poi il loro divario aumenta fino ad arrivare al valore di gap massimo $\Delta = 0.05$ per la posizione 10.

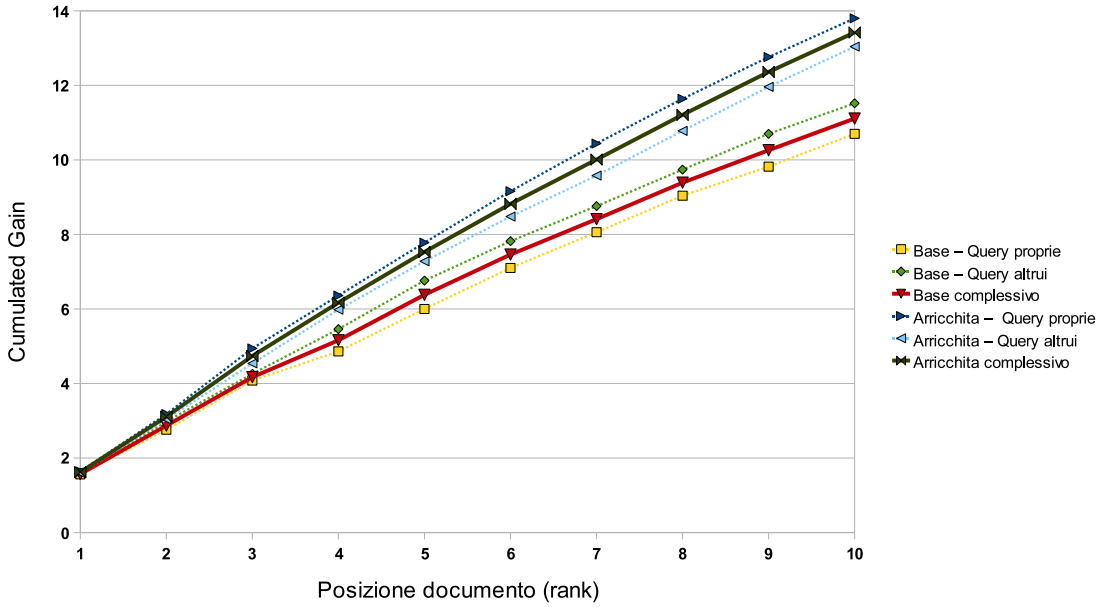

Figura 5.1: Curve medie di *Cumulated Gain* calcolate sui giudizi di rilevanza risultati dalla valutazione per la *cronologia base* e per la *cronologia arricchita* con documenti locali

Si noti che la posizione delle curve tratteggiate tenenti conto della tipologia dell'interrogazione rispetto alle curve complessive è invertita, per le due configurazioni del sistema: la curva celeste relativa alle query altrui per la configurazione arricchita è in posizione inferiore alla curva nera e mostra che gli utenti hanno giudicato in media meno rilevanti i documenti proposti per le query non proprie; la curva tratteggiata verde per le query altrui della configurazione con cronologia base è superiore rispetto alla curva rossa che registra l'andamento medio per quella configurazione: i membri del gruppo di valutazione hanno perciò valutato in media più positivamente le pagine web della cronologia base restituite per le query non proprie.

Passando al *Discounted Cumulated Gain* che sconta l'accumulo dei giudizi di rilevanza di un fattore logaritmico (*Figura 5.2*), si nota che la posizione relativa delle curve è analoga a quella del grafico precedente. L'andamento è meno lineare proprio per la presenza dello sconto applicato ai giudizi di rilevanza che si accumulano man mano che il *rank* aumenta. Le curve sono molto vicine in posizione 1 ($\Delta = 0.05$), esattamente come nel grafico precedente, perché lo sconto viene applicato dalla prima

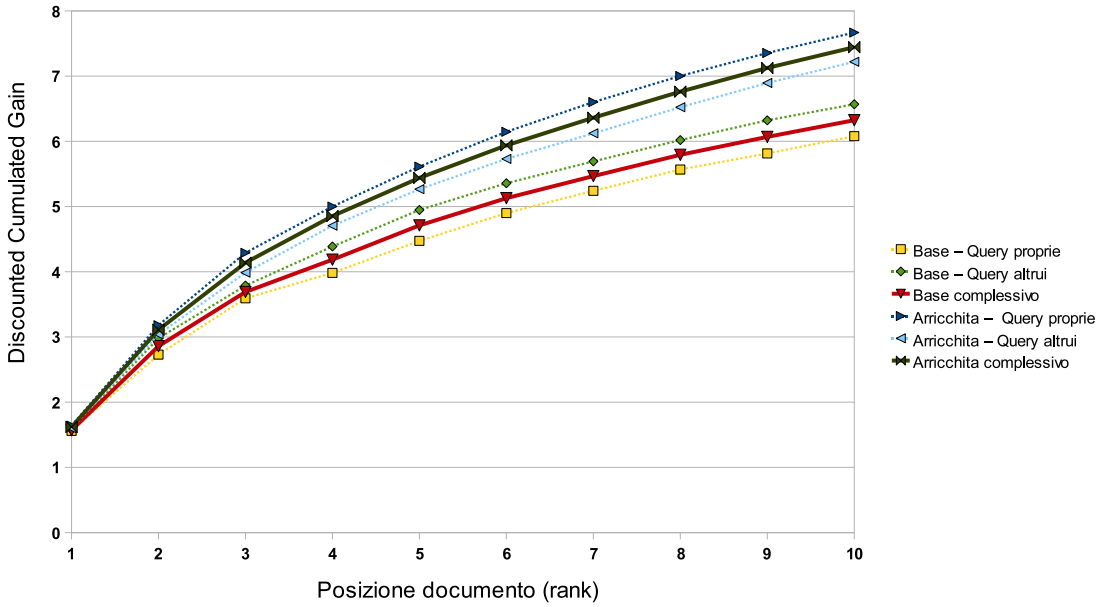

Figura 5.2: Curve medie di *Discounted Cumulated Gain* calcolate sui giudizi di rilevanza risultati dalla valutazione per la *cronologia base* e per la *cronologia arricchita* con documenti locali

posizione di valore uguale alla base del logaritmo (in questo caso da posizione 2). Il distacco massimo fra le due curve che riassumono gli andamenti complessivi delle due configurazioni si ha ancora in posizione 10 ed è quantificabile in $\Delta = 1.12$.

Il *Normalized Cumulated Gain* facilita l'interpretazione numerica e visiva, in quanto la normalizzazione implica che i valori dei guadagni accumulati siano, per ogni posizione, compresi nell'intervallo [0,1]. Se il valore normalizzato si avvicina a 1 significa che tende ad essere un valore ottimale per quel *rank*. Il grafico di *Figura 5.3* mostra che le curve di guadagno accumulato normalizzato mantengono la medesima posizione relativa. La curve complessive mostrano un gap massimo ($\Delta = 0.1$) in posizione 4. Le curve si incontrano e coincidono in posizione 10 e hanno valore normalizzato massimo e ottimale (valore 1). Questo si spiega entrando nel dettaglio del calcolo della normalizzazione, riprendendo l'esempio già presentato in *5.5 Misure per la valutazione delle prestazioni*.

Dato il vettore rappresentante una lista di giudizi di rilevanza di una query $G = (2, 1, 2, 1, 1, 1, 2, 1, 0, 2)$. Il vettore ideale relativo è quello che porta in testa i giudizi

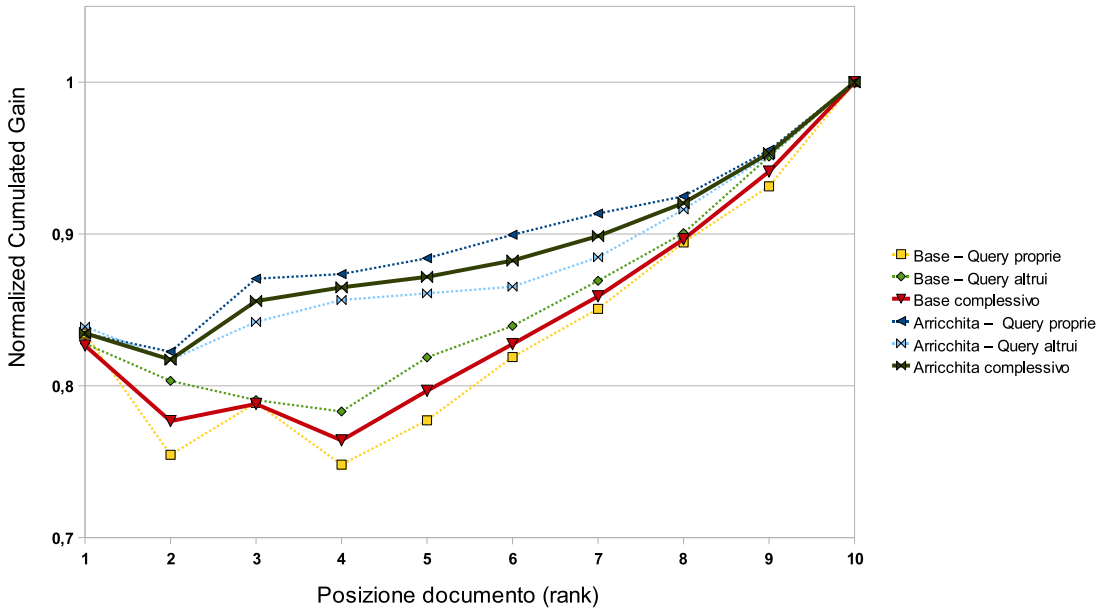

Figura 5.3: Curve medie di *Normalized Cumulated Gain* calcolate sui giudizi di rilevanza risultati dalla valutazione per la *cronologia base* e per la *cronologia arricchita* con documenti locali

di rilevanza di valore più alto (2), poi quelli intermedi (1) e in coda quelli più bassi (0): $G_I = (2, 2, 2, 2, 1, 1, 1, 1, 1, 0)$. Calcolando ora i *Cumulated Gain* per entrambi i vettori, si ottengono i seguenti due nuovi vettori $CG = (2, 3, 5, 6, 7, 8, 10, 11, 11, 13)$, $CG_I = (2, 4, 6, 8, 9, 10, 11, 12, 13, 13)$. Il vettore normalizzato nCG è ottenuto dal rapporto fra CG e CG_I, componente per componente:

$$nCG = \left(\frac{2}{2}, \frac{3}{4}, \frac{5}{6}, \frac{6}{8}, \frac{7}{9}, \frac{8}{10}, \frac{10}{11}, \frac{11}{12}, \frac{11}{13}, \frac{13}{13}\right)$$

È evidente che, per loro natura, tutti i vettori *Cumulated Gain* e tutti i vettori *Cumulated Gain Ideali* terminano con lo stesso valore. La differenza fra essi si presenta nell'ordine in cui i giudizi di rilevanza contribuiscono all'accumulo parziale. L'ultimo valore del vettore normalizzato sarà sempre pari a 1 e questo spiega il ricongiungimento di tutte le curve in posizione 10.

Gli andamenti delle curve dei risultati per la versione normalizzata del *Discounted Cumulated Gain* sono rappresentati in *Figura 5.4*. Ancora una volta, la posizione relativa delle curve è la medesima dei grafici precedenti. La distanza maggiore fra le spezzate dei valori di *Normalized Discounted Cumulated Gain* complessivi per le

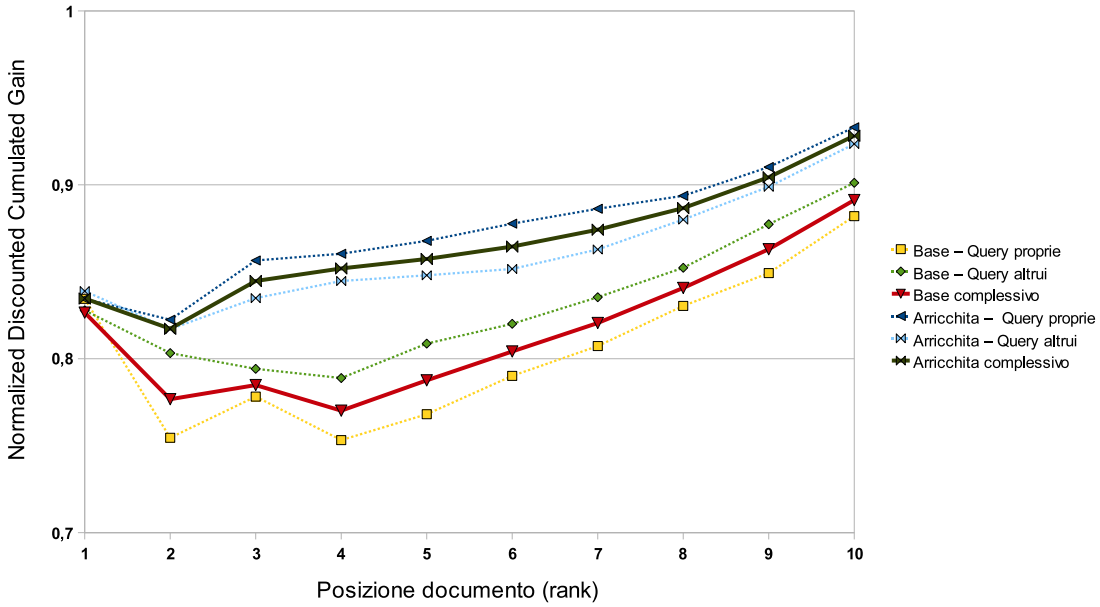

Figura 5.4: Curve medie di *Normalized Discounted Cumulated Gain* calcolate sui giudizi di rilevanza risultati dalla valutazione per la *cronologia base* e per la *cronologia arricchita* con documenti locali

due configurazioni si ha in posizione 4 e ha valore $\Delta = 0.8$. La minore distanza fra le stesse spezzate si ha in posizione 1 e ha valore $\Delta = 0.1$. Si nota che la configurazione del sistema con cronologia arricchita dà risultati sensibilmente migliori rispetto alla configurazione con cronologia base. Le curve, dopo un primo comportamento altalenante nelle prime quattro posizioni tendono a crescere per i valori di posizione dei documenti superiori a cinque, senza arrivare mai a 1.

Capitolo 6

Analisi dei risultati

In questo capitolo si discuteranno i risultati sperimentali ottenuti, cercando di individuare pregi e difetti del sistema proposto. Verranno anche puntualizzati i limiti del metodo utilizzato e la loro valenza.

6.1 Discussione dei risultati ottenuti

La *Tabella 5.1* mostra che ciascun utente ha recuperato complessivamente un numero di documenti locali variabile in un intervallo fra gli 11 e i 44, con una media di 28.9. Una variabilità così alta nel numero di documenti *desktop* può dipendere da tre fattori: la numerosità della collezione di documenti locali, il numero di documenti locali potenzialmente interessati dalla query, la capacità del sistema di portare nei primi 10 risultati i documenti locali. Il numero di documenti locali dipende dalla quantità di documenti presenti nel *dekstop* dell'utente e non può essere stimato anticipatamente. Il numero di documenti locali potenzialmente interessati dalle query varia in base alle interrogazioni stesse; per le query fornite dagli stessi utenti, si può presumere che il numero di documenti locali potenziali target di quelle query sia superiore. Questa assunzione è verificata dal fatto che il numero complessivo di documenti locali rilevati, ristretto alle sole query proprie ($F_{proprie} = 189$), è molto superiore a quello delle query altrui ($F_{altrui} = 109$). La capacità del sistema di ritrovare documenti locali portandoli nelle prime 10 posizioni dipende sostanzialmente dal valore di *frecency* assegnato ai documenti locali indicizzati. È da sottolineare che l'adattamento di *frecency* per i documenti di tipo *desktop* è incentrato sulla data di

ultima modifica del file e questo penalizza i documenti che sono presenti da molto tempo nel *file system*.

Le medie dei giudizi di rilevanza per tipologia di documento e riportate in *Tabella 5.2* mostrano che i membri del gruppo di valutazione tendono a valutare mediamente più rilevanti i documenti locali ($M_{totale} = 1.76$) rispetto a quelli web ($M_{totale} = 1.19$). Questa tendenza è comprensibile, visto l'utilizzo della particolare definizione di rilevanza con cui è stato chiesto di esprimere i giudizi. Ad ogni modo questo è anche il primo indice di funzionamento del sistema, poiché i documenti locali sono stati mediamente ritenuti più rilevanti di quelli della cronologia base formata da pagine web. È utile anche considerare le medie relative all'insieme delle query proprie e altrui. La media dei giudizi di rilevanza dati ai documenti locali relativi alle query degli utenti stessi ($M_{proprie} = 1.81$) è superiore rispetto a quella per la stessa tipologia di documenti ma per le query altrui ($M_{altrui} = 1.66$). Per spiegare questi valori si deve pensare che alcune query fra quelle altrui erano query molto generiche; perciò i giudizi di rilevanza su alcuni documenti locali reperiti ma magari non pienamente rispondenti al significato delle interrogazioni stesse sono stati penalizzati dagli utenti. Viceversa per quanto riguarda i documenti di provenienza web, i giudizi di rilevanza hanno premiato maggiormente i risultati delle query altrui ($M_{altrui} = 1.23$) rispetto a quelli delle query proprie ($M_{proprie} = 1.15$). La differenza numerica delle medie è molto piccola, ma una possibile interpretazione può essere una maggiore esigenza dei membri del gruppo di valutazione rispetto alle interrogazioni che riguardano la loro sfera di interessi personali; in qualche modo può essere vista come una sorta di maggiore criticità degli utenti rispetto ai risultati che più li riguardavano personalmente.

L'efficacia del sistema sviluppato è mostrata dall'andamento delle curve relative alle misure di *Cumulated Gain* e di *Discounted Cumulated Gain*. Con riferimento al grafico di *Figura 5.1*, le due curve continue mostrano che la configurazione con cronologia arricchita ottiene un guadagno accumulato superiore rispetto alla cronologia base. Questo accade perché i documenti locali reperiti si inseriscono nella lista delle pagine web, portando dei giudizi di rilevanza più elevati nelle prime dieci posizioni per due motivi: il primo è che il giudizio di rilevanza medio per i documen-

ti locali è in media più elevato; il secondo è che l'inserimento dei documenti locali fra quelli web fa slittare più in basso la coda della lista di documenti web, facendo uscire dalla finestra della top-10 i documenti web meno rilevanti. Complessivamente si ottiene una lista mista di documenti locali e di pagine web, tutti aventi giudizi di rilevanza medio-alti. Considerazioni del tutto analoghe valgono anche per il grafico del *Discounted Cumulated Gain* di *Figura 5.2*, con la differenza che lo sconto dovuto alla divisione per il logaritmo della posizione del documento porta le curve ad avere un andamento ancora crescente ma più smorzato rispetto alle spezzate quasi rettilinee descriventi il *Cumulated Gain* medio delle due configurazioni.

L'andamento quasi rettilineo delle curve in *Figura 5.1* è interpretabile alla luce della considerazione che il numero di posizioni valutate è molto piccolo (si valutano i primi 10 documenti proposti della *awesome bar*), rispetto alla dimensione delle collezioni sperimentali. Questa scelta, dettata soprattutto da esigenze sperimentali, porta nella finestra di documenti valutata dagli utenti i documenti con rilevanza maggiore, ottenendo delle liste di giudizi di rilevanza riempite quasi esclusivamente di 2 e 1. Ciò si traduce in curve di guadagno accumulato che, ad ogni posizione, crescono quasi sempre di 1 o 2 punti. È chiaro che eseguendo la media dei vettori di *Cumulated Gain*, si ottengono delle curve che smussano le variazioni a ciascun *rank* e possono essere interpolate con rette aventi coefficiente angolare pari alla media dei giudizi di rilevanza complessivi.

Per quanto riguarda le versioni normalizzate delle misure di performance, si nota dai grafici di *Figura 5.3* e *Figura 5.4* che le curve si attestano sopra il valore 0.7. Questo significa che i risultati ottenuti sono abbastanza buoni, visto che i vettori di *Normalized Cumulated Gain* e di *Normalized Discounted Cumulated Gain* sono a valori in [0,1]. L'andamento altalenante nelle prime posizioni è dovuto al fatto che le variazioni dei giudizi di rilevanza nelle prime posizioni porta ad ottenere dei guadagni cumulati che cambiano anche di molto. Man mano che il guadagno si accumula, le variazioni dovute ai giudizi di rilevanza incidono meno sulla somma parziale. Dalla posizione 4 in poi, l'andamento risulta crescente e tendente a 1 poiché il guadagno cumulato calcolato sui dati sperimentali è molto simile a quello del

vettore di guadagno cumulato ideale, che è dato dallo stesso vettore sperimentale, riordinato in modo decrescente.

Capitolo 7

Conclusioni e sviluppi futuri

In questo capitolo si evidenzieranno i risultati complessivi del progetto e verranno proposte alcune possibili migliorie da portare al sistema sviluppato. In ultima analisi si procederà a descrivere gli scenari futuri per applicazioni analoghe a *Awesome++* e il loro utilizzo in ambienti domestici o *corporate*.

7.1 Rilevanza dei risultati ottenuti

L'obiettivo di questo progetto era sviluppare uno strumento per ritrovare informazioni presenti nel *desktop* dell'utente, verificandone non solo il corretto funzionamento ma anche l'efficacia. La fase di valutazione è servita a quest'ultimo scopo.

Dai risultati complessivi è emerso che l'applicativo riesce a rispondere alle esigenze degli utenti, specialmente per quanto riguarda le query proposte da loro stessi e perciò riferibili ai propri interessi personali o lavorativi. Buoni risultati sono stati evidenziati anche dalle query aggiuntive e generiche inserite per ampliare il numero di topic cui appartenevano le interrogazioni proposte dal gruppo di valutazione. Riscontri più scarsi sono stati ottenuti nei risultati di interrogazioni per mezzo di query proposte dai membri del team di valutazione ma valutate da altri. La distribuzione dei risultati positivi per i documenti di provenienza *desktop* suggerisce che lo strumento ha maggiore efficacia per le specifiche interrogazioni che abbiano attinenza con l'ambito di lavoro o le attività personali dell'utente. Questo comportamento del sistema è del tutto naturale, in quanto un utente si aspetta di (ri)trovare

documenti locali per le tematiche legate ai propri interessi personali o professionali, di cui ha riscontro nei documenti sul proprio *desktop*.

La valutazione complessiva dello strumento è perciò positiva nel suo funzionamento, ma certamente migliorabile in numerosi aspetti implementativi e di presentazione dei risultati.

7.2 Possibilità di sviluppo

7.2.1 Miglioramenti dello strumento *Awesome++*

Lo strumento sviluppato ha carattere sperimentale e preliminare e non è adatto ad un utilizzo commerciale, ma per il suo sviluppo si è tenuto conto di numerose considerazioni anche di carattere pratico, cercando di immergersi in una situazione di utilizzo reale. Le difficoltà maggiori per riuscire a far diventare lo strumento un'applicazione di facile utilizzo quotidiano risiedono nell'automatizzazione delle procedure di indicizzazione dei documenti locali e di arricchimento della cronologia. Per rendere automatica l'indicizzazione dei documenti *desktop* è necessario eseguire una scansione del *file system* alla ricerca di nuovi file o per l'aggiornamento delle modifiche che essi hanno subito (ad esempio per la data di ultima modifica). Poiché l'aggiornamento della cronologia e la sua integrazione con documenti locali deve avvenire quando Firefox è chiuso per le questioni di locking del *database*, una possibile soluzione è intercettare la chiusura o l'avvio del browser e di collegare l'aggiornamento dei dati della cronologia con uno dei due eventi.

La procedura di adattamento del parametro *frecency* ai documenti locali può essere migliorata eseguendo *logging* dell'attività dell'utente, in due modi. Il primo consiste nella reintroduzione della frequenza di utilizzo dei documenti; il secondo è legato al calcolo della data effettiva di ultimo accesso dei documenti locali. Questa soluzione richiede di introdurre un *daemon* nel sistema in grado di conteggiare il numero di aperture dei documenti e di saper distinguere l'accesso di un documento da parte del sistema operativo rispetto all'accesso volontario dell'utente. I dati ricavati da un programma che lavora in *background* possono non solo migliorare il calcolo

di *frecency* ma anche portare eventualmente allo sviluppo di un parametro diverso adatto ai documenti di provenienza *desktop*.

7.2.2 Linee guida per una valutazione più approfondita

Per la valutazione del sistema, non avendo a disposizione collezioni sperimentali adatte allo scopo, si è dovuto procedere con la progettazione di un esperimento ad hoc che si è rivelato sufficiente per una valutazione preliminare. I metodi di reperimento basati sul contenuto dei *desktop* degli utenti hanno generalmente bisogno di essere testati in sessioni di tempo più lunghe, in modo da evidenziare le risposte del sistema a variazioni nella numerosità dei documenti sul lungo periodo, specialmente quando il funzionamento è dipendente dalla cosiddetta attività recente dell'utente.

La numerosità del gruppo di valutazione è stata limitata per questioni soprattutto logistiche e perché in una tesi di laurea non è possibile utilizzare risorse umane non volontarie. Inoltre era necessario che i membri del gruppo di valutazione fossero delle persone fidate e con un minimo di pratica per la risoluzione di piccoli problemi di installazione e configurazione del tool. È evidente che per eseguire una valutazione più consistente si dovrebbe costituire un gruppo di valutazione più numeroso ed eterogeneo, come spesso accadeva soprattutto in passato. Alternativamente è possibile pubblicare dei kit di valutazione su un sito web e affidarsi alla buona volontà degli utenti di Internet[1]. In questo caso bisognerebbe certamente tenere conto dell'eterogeneità del gruppo di valutazione e di numerose altre problematiche correlate alla rilevazione di giudizi di rilevanza su una tipologia di utenti che in generale non possono essere considerati fidati.

Numerose varianti possono essere proposte sulla modalità di costruzione della cronologia base, sulla sua composizione (lingua dei documenti, delle query, etc.), sul numero di documenti da far valutare per ogni interrogazione, sulla redistribuzione delle query agli utenti e anche sulla scala di valori per i giudizi di rilevanza. È chiaro che le possibili nuove configurazioni o modalità con cui condurre l'esperimento sono

[1]Un servizio offerto da Amazon® è *Mechanical Turk*[23]: una piattaforma che raccoglie intelligenze umane in grado di eseguire dei task e inviare i risultati ottenuti ai richiedenti.

molteplici e andrà scelta la combinazione di parametri che più si adatta allo scopo della nuova valutazione.

7.3 Scenari futuri

In questo progetto si è arrivati a sviluppare uno strumento in grado di ritrovare documenti locali, con particolare rilevanza per l'attività recente dell'utente. Si è affrontato il problema in modo innovativo, cercando di dare una nuova interpretazione all'attività recente degli utenti e offrendo uno strumento diverso dal cosiddetto *desktop search*. Con la crescita della capacità di memorizzazione e con l'uso intensivo del web come fonte di informazione, gli utenti si troveranno sempre più spesso di fronte alla necessità di ricercare documenti nel proprio *file system*. La mera ricerca per nome offerta dai sistemi operativi può risultare inefficiente (spesso la funzione di ricerca deve eseguire al volo l'indicizzazione dell'intero *file system*) o inefficace.

Uno strumento come *Awesome++*, estratto dalla componente browser, può invece servire allo scopo, ammettendo che l'indicizzazione dei documenti locali avvenga in *background* non appena il sistema nota un cambiamento nell'insieme di documenti cui si è interessati. Inoltre, se si toglie la limitazione nel numero di documenti suggeriti, il tool può rappresentare una valida alternativa ai motori di ricerca *desktop* per la maggior parte delle richieste informative dell'utente, con il vantaggio che non dovendo indicizzare il contenuto dei documenti, la fase di scansione del *file system* e di selezione dei documenti risulta molto velocizzata.

Se invece si vuole mantenere il tool all'interno di un'architettura come quella di Firefox, il risultato è di ottenere un'utile estensione delle funzioni del browser, fornendo funzioni aggiuntive alla *Awesome bar* che viene già ampiamente utilizzata. Una barra degli indirizzi così intelligente potrà addirittura soppiantare il concetto dei segnalibri (bookmark) o preferiti più utilizzati, in quanto le statistiche sulle pagine visitate di recente forniscono automaticamente alla *awesome bar* le informazioni necessarie per presentare all'utente le pagine che egli visita più spesso.

È opportuno sottolineare anche la possibilità di usare lo strumento sviluppato in questo progetto come base per costruire, assieme a un buon generatore di query, un

sistema più complesso e automatico che proponga agli utenti direttamente pagine web o documenti locali di interesse dell'utente, basandosi esclusivamente sulla sua storia passata. Sarebbe come fornire una rassegna stampa ibrida web e *desktop* di documenti di possibile interesse.

È interessante evidenziare la tendenza di fusione dell'ambiente *desktop* con quello web legato alla Rete. L'estensione e l'integrazione di *Awesome++* in un browser multifunzione sembra proiettare questo progetto in un ambiente dove le sfumature fra ambiente locale e globale (Internet e il Web) sono molto più attenutate, anche in una prospettiva di *cloud computing* che è già realtà.

Ringraziamenti

Desidero ringraziare tutti i colleghi e gli amici che mi hanno accompagnato nel mio percorso universitario, nelle attività di studio e di lavoro ai progetti comuni ma anche nei momenti di svago e di viaggio.

Un ringraziamento particolare al Professore Massimo Melucci per la sua disponibilità, per i suoi preziosi suggerimenti nello sviluppo di questo progetto e per avermi dato l'opportunità di lavorare ad un argomento così innovativo e interessante.

Grazie ad Andrea, Chiara, Francesco, Luca, Marco M., Marco V., Matteo, Michele, Silvio e Simone, pazienti membri del gruppo di valutazione e senza il cui contributo non sarebbe stato possibile dare un senso all'applicazione sviluppata in questo progetto.

Uno speciale ringraziamento a Matteo con cui ho passato numerose serate a cercare di capire se era il mio software che non funzionava o se Linux si rifiutava di farlo funzionare.

Grazie alla mia famiglia e in particolare ai miei genitori, per avermi sempre sostenuto anche nei momenti di difficoltà, per aver creduto in me e per aver capito le scelte che mi hanno portato dove sono oggi. Spero che questo traguardo ripaghi in parte i vostri sforzi.

Appendice A

In quest'appendice vengono riportate, per ogni utente del gruppo di valutazione, la lista delle query assegnate e la tabella coi risultati sperimentali. In particolare nelle tabelle si possono leggere i giudizi di rilevanza che ciascun membro del team di valutazione ha assegnato ai primi 10 documenti proposti come risultati delle query, nelle due configurazioni del sistema.

Appendice A

Liste di query assegnate all'utente Andrea.

- Query proprie: quantistica, matlab, analogica, radio, elettronica organica
- Query generiche: fuoristrada, radioamatore, engineering
- Query di altri utenti: latex, orari

Tabella 1: Risultati sperimentali dell'utente Andrea

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
quantistica	2	2	2	0	2	2	2	1	0	1
matlab	2	2	2	1	0	2	2	2	2	1
analogica	1	0	0	0	2	2	0	0	2	0
radio	2	1	2	2	1	2	2	0	1	2
elettronica organica	1	1	2	2	1					
fuoristrada	2	2	2	2	2	2	2	2	1	1
radioamatore	2	2	2	2	2	2	2	1	0	2
engineering	1	0	0	1	2	1	1	1	1	1
latex	2	2	2	2	2	1	0	2	0	2
orari	2	1	2	1	1	2	2	2	2	1
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
quantistica	2	2	2	0	2	2	2	1	0	1
matlab	2*	2	2	2*	2*	2*	2*	2*	2*	2
analogica	1	2*	2*	2*	2*	2*	2*	2*	1*	2*
radio	2	1	2	2	2*	2*	2*	2*	2*	2*
elettronica organica	1	1	2	2	1					
fuoristrada	2	2	2*	2*	2	2	2	2	2	2
radioamatore	2	2*	2*	2*	2	2	2	2	2	2
engineering	1	0	0	1	2	1	1	1	1	1
latex	2	2	2*	2*	2*	2*	2*	2*	2*	2*
orari	2	1	2	1	1	2	2	2	2	1

Liste di query assegnate all'utente Chiara.

- Query proprie: foot, spartiti, biomeccanica, presentazione, schema
- Query generiche: 2009, biologia, università
- Query di altri utenti: reti biologiche, lyrics

Tabella 2: Risultati sperimentali dell'utente Chiara

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
foot	1	2	1	0	1	0	0	0	0	1
spartiti	2	2	2	2	2	2	2	2	2	2
biomeccanica	2	0	1	1	2	1	0	1	1	1
presentazione	1	1	1	2	0	1	1	2	1	1
schema	0	1	0	0	0	1	0	1	2	2
2009	0	2	1	0	2	1	1	1	1	1
biologia	2	2	1	1	1	2	1	2	1	0
università	2	2								
reti biologiche	2	2	2							
lyrics	2	2	2	2	2	2	2	2	2	2
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
foot	1	2	1	0	1	0	0	2*	2*	2*
spartiti	2	2*	2*	2*	2	2	2	2	2	2
biomeccanica	1	2	2*	2*	2*	2*	2*	2*	2*	2
presentazione	2	2*	2*	2*	2*	2*	2	1	2	0
schema	2*	2*	2*	2*	2*	2*	2*	2*	2*	2*
2009	0*	2	2*	2*	2*	2*	2*	2*	2*	2*
biologia	1	2	2*	2	2	2	2	2	2	1
università	2*	2*	2	2						
reti biologiche	2	2	2							
lyrics	2	2	2	2	2	2	2	2	2	2

Appendice A

Liste di query assegnate all'utente Francesco.

- Query proprie: esame di stato, socket library, tinyos, tutorial c, 6lowpan
- Query generiche: web service, risultati, reti
- Query di altri utenti: dispensa, ubuntu

Tabella 3: Risultati sperimentali dell'utente Francesco

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
esame di stato	2	1	2	1	0	0				
socket library	1	2	2							
tinyos	2	0	0	0	2	0	1	1	0	
tutorial c	2	2	1	1	1	0	0	0	0	0
6lowpan	1	1								
web service	2	2	0	0	2	0	2	1	0	0
risultati	2	2	2	2	2	1	2	2	2	2
reti	1	0	0	0	1	0	2	0	1	0
dispensa	1	0	1	0	1	0	1	0	0	0
ubuntu	2	2	1	0	2	2	1	0	1	1
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
esame di stato	2*	1*	2	1	2	1	0	0		
socket library	1	2	2							
tinyos	2*	2*	2	0	0	0	2	0	1	1
tutorial c	2*	2	2	1	1	1	0	0	0	0
6lowpan	2*	1*	1	1						
web service	2*	2	2	0	0	2	0	2	1	0
risultati	2	2	2	2	2	1	2	2	2	2
reti	2*	1	0	0	0	1	0	2	0	1
dispensa	1*	1	0	1	0	1	0	1	0	0
ubuntu	2*	2	2	1	0	2	2	1	0	1

Liste di query assegnate all'utente Luca.

- Query proprie: exchange, azure, basket, free ride, cognitive
- Query generiche: photoshop, sport, viaggio
- Query di altri utenti: latex, mappe

Tabella 4: Risultati sperimentali dell'utente Luca

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
exchange	2	1	0	0	0	0	1	1	0	2
azure	2	2	1	0	2	2	1	2	0	0
basket	2	2	2	1	0	2	0	0	0	2
free ride	2	2	2	0	2	2	2	2	0	2
cognitive	0	0	0	0	0	0	0	0	0	0
photoshop	2	2	0	0	0	0	2	2	2	0
sport	0	2	2	2	2	1	0	2	2	1
viaggio	0	0	0	2	0	0	0	0	2	0
latex	2	2	2	2	2	2	2	2	2	2
mappe	2	2	2	2	2	2	2	2	2	2
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
exchange	2	2	2*	2*	2*	2*	2*	2*	0*	0
azure	2*	2	2	1	0	2	2	1	2	0
basket	2*	2*	2	2	2	2	0	2	1	0
free ride	2	2	2	0	2	2	2	2	0	2
cognitive	0	2*	2*	2*	0	0	0	0	0	0
photoshop	2*	2	2	0	0	0	0	2	2	2
sport	0	2	2	2	2	1	0	2	2	1
viaggio	0	0	0	2	0	0	0	0	2	0
latex	2	2	2	2	2	2	2	2	2	2
mappe	2	2	2	2	2	2	2	2	2	2

Appendice A

Liste di query assegnate all'utente Marco M.

- Query proprie: proteine, reti biologiche, dati 3d, linux, film
- Query generiche: giochi, gentoo, unipd
- Query di altri utenti: informatica musicale, presentazione

Tabella 5: Risultati sperimentali dell'utente Marco M.

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
proteine	0	0	0	0	0	2	0	1	0	0
reti biologiche	2	0	1							
dati 3d	0	0	0	1	0	1	1	0	0	0
linux	2	2	1	2	2	0	2	1	0	1
film	1	1	2	1	1	2	0	0	1	1
giochi	0	0	0	0	0	0	0	0	0	0
gentoo	2	0	2	2	2	0	0	0	2	1
unipd	2	0	0	1	0	1	1	0	0	0
informatica musicale	2	1	0	0	2	2	2	0	0	1
presentazione	2	1	0	2	0	2	0	0	0	0
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
proteine	2*	2*	2*	2*	0	0	0	0	0	1
reti biologiche	0	0	1							
dati 3d	0	1	2*	2*	2*	2*	2*	2*	2*	2*
linux	2	2	1	2	2	0	2	1	0	1
film	1	1	2	1	1	2	0	0	1	1
giochi	0	0	0	0	0	0	0	0	0	0
gentoo	2	0	2	2	2	0	0	0	2	1
unipd	2	1*	1*	1*	1*	1*	1*	1*	1*	1*
informatica musicale	2	1	2*	2*	2*	2*	2*	2*	2*	2*
presentazione	2	1	2*	2*	2*	2*	2*	2*	2*	2*

Liste di query assegnate all'utente Marco V.

- Query proprie: banca, programmazione web, google web toolkit, fotografia, mappe
- Query generiche: curriculum, vacanze, esame
- Query di altri utenti: cognitive, prototipo

Tabella 6: Risultati sperimentali dell'utente Marco V.

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
banca	1	1	1	0	2	2	2	1	2	1
programmazione web	2	2	2	0	1	1	1	1	1	0
google web toolkit	1	0	1	1	0	1	1	2	1	0
fotografia	2	0	2	2	2	2	1	1	0	0
mappe	2	2	2	2	2	2	2	1	1	0
curriculum	2	2	2	1	2	1	0	2	0	0
vacanze	2	2	2	2	2	1	1	1	0	2
esame	1	2	1	1	1	1	0	0	0	0
cognitive	2	0	1	1	1	1	1	1	1	1
prototipo	2	1	0	0	0	0	1	1	2	1
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
banca	1*	2	2	1*	1*	1*	1*	1	1*	1*
programmazione web	0*	2	2	1*	1*	1*	1*	1*	1*	1*
google web toolkit	1	0	2*	2*	2*	2*	2*	2*	2*	2*
fotografia	2	0	2	2	2	2	1	1	0	0
mappe	2	1*	1*	1*	1*	1*	1*	1*	2	2
curriculum	2*	2	2*	2	2	1	2	1	0	2
vacanze	2	2	2	2	2	1	1	1	0	2
esame	1	2	1	1	1	1	0	0	0	0
cognitive	2	0	1	1	1	1	1	1	1	1
prototipo	2	1	0	0	0	0	1	1	2	1

Appendice A

Liste di query assegnate all'utente Matteo.

- Query proprie: robot, prototipo, orari, khr manual, gossip
- Query generiche: connettore, papiro, guida
- Query di altri utenti: programmazione web, matlab

Tabella 7: Risultati sperimentali dell'utente Matteo

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
robot	2	2	2	1	1	1	2	2	2	1
prototipo	2	1	2	1	2	0	2	2	1	1
orari	2	1	2	1	1	2	2	2	2	2
khr manual										
gossip	2	2	2	1	2	1	2	0	2	2
connettore	1	2	2	2	0	1	1	2	2	1
papiro	2	2	2	2	0	1	1	2	0	2
guida	1	2	2	2	2	2	2	1	1	2
programmazione web	2	2	2	2	2	2	1	2	2	0
matlab	2	2	2	2	1	1	1	2	1	2

Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
robot	2	2	2*	2*	2*	2*	2*	2*	2*	2*
prototipo	2*	2	1	2*	2*	1	2	2	0	2
orari	2	1	2*	2*	2*	2*	2*	2*	2*	2*
khr manual	2*	2*	2*							
gossip	2	2	2*	2*	2	1	2	0	2	0
connettore	2*	1	2	2*	2	2	0	1	1	2
papiro	2	2	2*	2*	2*	2*	2	2	1	2
guida	1	2	2	2	2*	2*	2*	2*	2*	2*
programmazione web	2	2	2	2	2	2	1	2	2	0
matlab	2	2	0*	0*	0*	0*	0*	0*	0*	0*

Liste di query assegnate all'utente Michele.

- Query proprie: sottotitoli, chitarra, java, lyrics, documentation
- Query generiche: spartiti, musica, traduttore
- Query di altri utenti: radio, fotografia

Tabella 8: Risultati sperimentali dell'utente Michele

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
sottotitoli	2	2	2	0	1	1	0	0	1	0
chitarra	2	2	0	0	1	2	0	0	1	0
java	2	2	2	2	2	0	1	1	1	2
lyrics	1	1	1	1	1	1	1	1	2	0
documentation	1	0	1	1	2	1	2	2	1	1
spartiti	0	1	0	1	2	0	1	0	0	1
musica	0	0	2	0	2	1	0	0	1	1
traduttore	2	2	2	2	2	0	1	0	2	0
radio	2	2	2	2	2	2	0	1	0	1
fotografia	2	0	1	2	2	0	0	0	0	0
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
sottotitoli	2	2	2	0	1	1	0	0	1	0
chitarra	2	2	0	0	1	2	0	0	1	0
java	2	2*	2*	2*	2	2	2	2	0	1
lyrics	1	1	1	1	1	1	1	1	2	0
documentation	1	0	1*	1*	1*	1*	1*	1*	1*	1*
spartiti	0	1	0	1	2	0	1	0	0	1
musica	0	0	2	0	0*	0*	0*	0*	2	1
traduttore	2	2	2	2	2	0	1	0	2	0
radio	2	2	2	2	2	2	0	1	0	1
fotografia	2	0	1	2	2	0	0	0	0	0

Appendice A

Liste di query assegnate all'utente Silvio.

- Query proprie: labview, latex, manuale, tesi, ubuntu
- Query generiche: ebook, guerra, misure
- Query di altri utenti: socket library, azure

Tabella 9: Risultati sperimentali dell'utente Silvio

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
labview	2	2	2	0	1	1	0	2	2	2
latex	2	2	1	2	1	1	0	2	0	0
manuale	2	0	0	0	2	0	2	2	1	2
tesi	1	0	2	0	2	1	1	1	1	2
ubuntu	2	2	2	1	2	2	1	1	0	1
ebook	2	1	1	2	2	2	1	0	2	0
guerra	2	2	0	1	1	2	1	1	2	1
misure	1	2	2	0	2	0	0	0	2	0
socket library	1	0	0							
azure	2	2	2	1	0	1	2	2	0	0
Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
labview	2	2	2*	2*	2*	2*	2*	2*	2*	2*
latex	2	2	2*	2*	2*	2*	2*	2	2	2
manuale	2	0	0	2*	2*	2*	2*	2*	2*	2*
tesi	2	0	2*	2*	2*	2*	2*	2*	2*	2*
ubuntu	2	2	2	1	2	2	1	1	0	1
ebook	2	1	1	2*	2*	2*	2	2	2	1
guerra	2	2	0	1	1	2	1	1	2	1
misure	1	2	2*	2*	2	0	2	0	0	0
socket library	1	0	0							
azure	2	2	2	1	0	1	2	2	0	0

Liste di query assegnate all'utente Simone.

- Query proprie: informatica musicale, dispensa, hockey pista, espresso, vicenza
- Query generiche: wireless, album, sistemi operativi
- Query di altri utenti: proteine, esame di stato

Tabella 10: Risultati sperimentali dell'utente Simone

Risultati della valutazione sulla cronologia base										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
informatica musicale	2	2	2	1	2	2	2	1	0	2
dispensa	2	1	2	1	1	1	0	1	1	1
hockey pista	2	2	2	0	1	2	2	2	0	2
espresso	2	1	2	2	2	2	2	2	2	2
vicenza	2	2	2	2	2	2	2	2	2	1
wireless	2	0	2	2	1	1	0	2	1	0
album	1	2	2	1	2	2	0	1	2	2
sistemi operativi	2	2	2	1	2	2	2	2	1	2
proteine	2	2	2	2	2	2	1	2	2	2
esame di stato	2	2	2	2	0	1	1			

Risultati della valutazione sulla cronologia arricchita										
Query/Docs	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
informatica musicale	2	2	2*	2*	2*	2*	2*	2*	2*	2*
dispensa	2	1	2*	2*	2*	2	1	1	1	0
hockey pista	2	2	2	2*	0	1	2	2	2	0
espresso	2	1	2	2*	2	2	2	2	2	2
vicenza	2	2	2*	2	2	2	2	2	2	2
wireless	2	0	2	2*	2*	2*	2*	2*	2*	2*
album	1	2	2*	2*	2*	2*	2*	1*	1*	1*
sistemi operativi	2	2*	2*	2*	2*	2*	2*	2*	2*	2*
proteine	2	2	2	2	2	2	1	2	2	2
esame di stato	2	2	2	2	0	1	1			

Appendice B

In questa appendice vengono riportate le istruzioni fornite a ciascun utente del gruppo di valutazione, per la configurazione e il testing del sistema. Il registro usato è quello di una guida piuttosto informale e sono presenti suggerimenti tecnici e logistici. Chi volesse installare e provare l'applicazione sviluppata, può seguire le istruzioni dettagliate riportate in questa guida.

Configurazione e testing del sistema

Premessa

Come anticipato via email, il tool che ho sviluppato serve per aggiungere alla cronologia di Firefox alcune voci riferite ai documenti [doc, docx, ppt, pptx, (e corrispondenti estensioni per OpenOffice) pdf, ps] che l'utente ha nel proprio computer. Per fare ciò è necessario eseguire alcune operazioni di configurazione del sistema.

- Avere installata la Java Virtual Machine (o installarla per l'occasione)
- Avere installato Moxilla Firefox 3.5 o successivo (o installarlo per l'occasione)

Installazione della Java Virtual Machine

La JVM può essere facilmente scaricata e installata dal sito della Sun Microsystems seguente

`http://java.com/it/download/`

Il file dovrebbe essere di circa 10MB e dopo l'installazione è possibile eseguire un test per verificare che la JVM sia installata correttamente.

Installazione di Firefox

L'ultima versione di Firefox può essere scaricata e installata dal sito seguente, scegliendo l'installer adatto al proprio sistema operativo

`http://www.mozillaitalia.it/home/download/`

Per gli utenti più esperti

A chi avesse voglia/tempo e capacità per lavorare con più profili di Firefox, suggerisco di creare un profilo ad hoc per la valutazione seguendo le istruzioni presenti nella seguente pagina

`http://support.mozilla.com/it/kb/Gestione+dei+profili`

Questo eviterà di eseguire il successivo backup del profilo. Chi lavora con un profilo ad hoc deve assicurarsi di avviare e di lavorare con tale profilo in tutte le fasi seguenti.

Localizzazione del database su cui Firefox salva la cronologia

In Windows:

1. Aprire il menu di avvio: `Start - Esegui`
2. Digitare `%APPDATA%` e premere Invio (o cliccare OK) e si aprirà una finestra esplorazione contenente diverse cartelle.
3. Cliccare ed entrare nella cartella `Mozilla`
4. Cliccare ed entrare nella cartella `Firefox`
5. Cliccare ed entrare nella cartella `Profiles`
6. Cliccare ed entrare nella cartella `xxxxxxxx.default` (dove `xxxxxxxx` è una stringa di caratteri casuali e default è il nome del profilo di default. Se avete più profili di Firefox, selezionate il profilo dove volete eseguire la valutazione)

7. Copia-incollate in un file di testo il percorso che avete raggiunto, che dovrebbe avere la seguente forma:

   `C:\Documents and Settings\Utente\Dati applicazioni\Mozilla\Firefox\Profiles\xxxxxxxx.default`

8. Individuate nella cartella il file `places.sqlite` che è quello che contiene la cronologia e sul quale dovrete lavorare diverse volte

9. Per comodità, copia-incollate nel file di testo il percorso completo di quel file

   `C:\Documents and Settings\Utente\Dati applicazioni\Mozilla\Firefox\Profiles\xxxxxxxx.default\places.sqlite`

10. Il vostro file di testo dovrà perciò contenere il percorso del profilo di Firefox e il percorso completo del file `places.sqlite`

In Linux:

1. I linuxisti in quanto utenti più esperti riceveranno istruzioni più succinte. Invididuare la cartella del profilo di firefox che dovrebbe essere la seguente:

   `~/.mozilla/firefox/xxxxxxxx.default/`

   (dove `xxxxxxxx` è una stringa di caratteri casuali e default è il nome del profilo di default. Se avete più profili di firefox, selezionate il profilo dove volete eseguire la valutazione)

2. Copia-incollate in un file di testo il percorso del profilo e il path completo del file `places.sqlite` ivi contenuto.

Per entrambi i sistemi operativi, se avete difficoltà a individuare il profilo, consultate la pagina

`http://support.mozilla.com/it/kb/Profili`

IMPORTANTE: eseguire un backup del profilo

Poiché successivamente sarete chiamati a sostituire il vostro file places.sqlite con quello che vi fornirò io e che conterrà la cronologia-base, vi consiglio di eseguire un backup del vostro profilo, copiando l'intera cartella che lo contiene (quella del tipo `xxxxxxxx.default`) in un'altra posizione del vostro *file system*.

Attenzione: NON SALTATE QUESTA FASE perché nel file places.sqlite sono presenti anche i vostri segnalibri Firefox e la vostra cronologia passata. Perciò vi invito calorosamente ad eseguire il backup del profilo, così da ripristinare il vostro sistema non appena finita la valutazione.

Configurazione di Firefox

È necessario eseguire alcuni settaggi per quanto concerne la barra degli indirizzi (URL-bar, chiamata anche *Awesome bar* dalla versione 3.0 di Firefox in poi). Procedere come segue. Aprire il browser e digitare nella barra degli indrizzi (proprio come se fosse l'indirizzo di una pagina web che vorreste navigare) la stringa `about:config` e premere Invio. Verrà visualizzato un messaggio del tipo `Questa operazione potrebbe invalidare la garanzia`. Cliccare quindi sul bottone `Farò attenzione, prometto`. Si aprirà una tabella con numerose voci per la configurazione di Firefox. Cerare manualmente o tramite il filtro le seguenti voci e settarle ai valori corrispondenti.

Nome parametro: `browser.urlbar.maxRichResults` - Nuovo valore: 10

Nome parametro: `browser.places.smartBookmarkVersion` - Nuovo valore: 0

Nome parametro: `places.frecency.updateIdleTime` - Nuovo valore: 0

Una volta impostati i nuovi valori, lo stato di ciascun parametro risulterà essere cambiato in personalizzato. Chiudere Firefox per *salvare* le nuove impostazioni.

Chiudi Firefox, è necessario, se non l'hai ancora fatto.

Primo utilizzo del tool Java *Awesome++*

Dopo essersi assicurati che la Java Virtual Machine sia correttamente installata e funzionante, sarà necessario eseguire un primo test per capire se il tool *AwesomePlus*

funziona adeguatamente.

- Raggiungere la directory `kit_tuonome` dove è stato scompattato il file `kit_tuonome.rar` che ho fornito ed entrare nella cartrella `tool_windows` o `tool_linux`, a seconda del sistema operativo in uso

- Verificare che siano presenti una cartella `lib`, il file `AwesomePlus.jar`, il file di testo `README` La cartella lib contiene le librerie necessarie a far funzionare il tool; il file README è un file costruito di default; il file AwesomePlus.jar è il mio tool, mentre places.sqlite è il file che contiene la cronologia-base.

- In Windows, avviare il file `AwesomePlus.jar` facendo doppio-clic col mouse. In Linux lanciare il file `AwesomePlus.jar` da shell. Se tutto procede correttamente, verrà aperta una finestra a video dal titolo `Progetto` ***`Awesome++`*** `/ Melis Giovanni` con diversi bottoni (in Windows verrà aperta anche una seconda finestra per la Console. In Linux invece lo standard-output è scritto sulla shell).

A questo punto il tool si è avviato correttamente. Quella successiva è una brevissima fase di testing che permette di impadronirsi dell'interfaccia grafica.

[Prima di iniziare, eseguire il backup del proprio profilo di Firefox se non è ancora stato fatto]

1. Premere il bottone `Seleziona cartella/disco` e selezionare una cartella contenente pochi documenti aventi le estensioni selezionate in *Tipi di documenti*. Se l'operazione è andata a buon fine, verrà stampato in console il messaggio

   `Cartella o drive selezionato: PATH_SELEZIONATO`

2. Premere il bottone `Seleziona database SQLite` e selezionare il file `places.sqlite` presente nella cartella del profilo di Firefox (la stessa che avevate copia-incollato in fase di configurazione nel file testuale). Se l'operazione è andata a buon fine, verrà stampato in console un messaggio analogo a

```
Database SQLite selezionato: C:\Documents and Settings\Utente\
Dati applicazioni\Mozilla\Firefox\Profiles\xxxxxxxx.default\
places.sqlite
```

3. Chiudere Firefox

4. [Questa fase può richiedere alcuni minuti] Premere sul bottone `Indicizzazione Locale`. In console saranno stampati i percorsi dei file che verranno indicizzati. Se l'operazione va a buon fine, in console viene stampata una dicitura analoga alla seguente (consultare la successiva sezione *Possibili problemi/bug comuni*)

   ```
   Numero di documenti da inserire in cronologia: 5
   Tempo di elaborazione: 0.016
   Tempo stimato di arricchimento della cronologia: 2.25 secondi
   (circa 0 minuti)
   ```

5. [Questa fase può richiedere alcuni minuti] Procedere quindi con l'arricchimento, premendo il bottone `Integrazione cronologia`. In console verrà stampata la percentuale di avanzamento. Se l'operazione va a buon fine, in console viene stampata una dicitura analoga alla seguente

   ```
   Integrazione della cronologia web con file da disco
   Avanzamento: 20%
   Avanzamento: 60%
   Avanzamento: 100%
   Tempo integrazione: 2.765 secondi (circa 0 minuti)
   Integrazione della cronologia con file da disco conclusa
   ```

6. Ora è possibile aprire Firefox e provando a digitare il nome di uno dei file indicizzati nella barra degli indirizzi, Firefox dovrebbe proporlo nel menu di autocompletamento.

7. Se non sono state stampate eccezioni sulla console, il primo test può essere considerato completato e si può passare alla fase di valutazione vera e propria.

Possibili problemi e bug comuni

Su Windows non sono stati riscontrati problemi.

In linux il tool non sempre riesce a distinguere un *collegamento a una directory* da una dirctory vera e propria. I collegamenti ad altre directory possono perciò instaurare dei riferimento ciclici. Questo porta a dei loop infiniti nella fase di `Indicizzazione locale`.

Ho perciò sviluppato una piccola procedura che segnala le directory che potrebbero contenere riferimenti ciclici. Qualora il tool ne incontrasse una, in console verrebbe prodotta una serie di messaggi di questo tipo:

```
ATTENZIONE: POSSIBILE RIFERIMENTO CICLICO NELLA CARTELLA
<PATH_DELLA_CARTELLA>
L'utente è pregato di controllare ed eventualmente escludere la
cartella all'origine. Grazie
```

L'utente linuxista è invitato a controllare la console per individuare la presenza di tali messaggi e verificare che effettivamente si sia in presenza di un loop. In tale caso bisogna fermare, riavviare il tool e aggiungere quella directory ad una blacklist attraverso il bottone `Escludi cartella/directory`. Per aggiungerne più di una, premere ancora il bottone di esclusione e scegliere una seconda directory e così via. Dopo di che, avviare di nuovo l'indicizzazione. Se vengono segnalate ancora directory con riferimenti ciclici, ripetere la procedura.

Per evitare di dover eseguire questa procedura troppe volte, anticipo che sono stati riscontrati problemi con le directory `/sys`, `/proc` e `~/.wine`

Se in fase di lancio dell'applicazione, in shell vengono scritti dei warning, non preoccuparsi e ignorarli.

Appendice C

In questa appendice vengono riportate le istruzioni fornite a ciascun utente del team di valutazione per la conduzione dell'esperimento. Il registro usato è quello di una guida piuttosto informale. Chi volesse ripetere l'esperimento di valutazione, può seguire le istruzioni dettagliate riportate in questa guida.

Premesse per la valutazione

L'obiettivo della valutazione è giudicare la rilevanza dei primi 10 documenti proposti dalla *awesome bar*, quando l'utente digita le query a lui assegnate. Il giudizio andrà ripetuto in due configurazioni del sistema:

1. quando la cronologia corrisponde alla cronologia-base che vi fornirò (che è una cronologia Web uguale per tutti)

2. quando la cronologia-base è integrata coi documenti presenti nel vostro computer

Per fare ciò è necessario:

- avere completato la fase di configurazione (vedi *Appendice B*)

- avere ricevuto correttamente il file Excel o Calc da compilare coi giudizi di rilevanza

- avere a disposizione un'ora circa (escluso il tempo di elaborazione del tool)

Cos'è la rilevanza di un documento e come dare i giudizi

La rilevanza di un documento, in generale, è la capacità del documento di rispondere all'esigenza informativa dell'utente. Tuttavia, nella fattispecie i documenti presenti nella cronologia-base sono sconosciuti agli utenti, mentre i documenti personali che saranno presentati nella configurazione 2 (cronologia-base arricchita) sono conosciuti. Inoltre i task per cui gli utenti scrivono nella *awesome bar* (barra degli indirizzi), non sono quelli di una ricerca informativa come quella espressa per i motori di ricerca. Sono sostanzialmente i seguenti:

- Ritrovare una pagina o un documento già visto in passato e di cui si è dimenticato l'indirizzo

- Trovare pagine o documenti attinenti all'argomento rappresentabile con le parole che compongono la query

Questo ragionamento e la particolarità dell'utilizzo della *awesome bar* modificano la generica definizione di rilevanza come segue:

Una pagina web o un documento locale è rilevante se:

- è quella già vista in passato di cui si è dimenticati l'indirizzo o il path

- contiene frasi, immagini, contenuti multimediali attinenti alle parole della query

Per la natura della cronologia-base che è inizialmente sconosciuta agli utenti e per rendere più fine l'espressione dei giudizi di rilevanza, deve essere utilizzata una scala a tre valori 0,1,2:

```
0 - Pagina o documento non rilevante
1 - Pagina o documento abbastanza rilevante
2 - Pagina o documento molto rilevante
```

Valutazione della configurazione con cronologia base

In questa fase l'utente è chiamato a dare giudizi di rilevanza ai documenti proposti dalla *awesome bar* di Firefox che pesca i risultati dalla cosiddetta cronologia-base, cioè una cronologia web che è uguale per tutti.

Configurazione del sistema

[Chi lavora con un profilo di Firefox ad hoc, deve assicurarsi di usare quel profilo nell'intera fase di valutazione. Non dovrà eseguire il backup del profilo]

Configurare Firefox come spiegato nel paragrafo `Configurazione di Firefox` dell'*Appendice B* (sostanzialmente eseguire le operazioni in `about:config` e riavviare Firefox).

[Se non lo si è ancora fatto, eseguire una copia di backup del profilo di Firefox]

- Chiudere Firefox
- Sostituire il file `places.sqlite` presente nella cartella del profilo di Firefox (il path della directory ve l'eravate salvato in un file testuale in fase di configurazione) con il file `places.sqlite` che ho fornito nel tuo kit (questo file places.sqlite è quello che contiene la cronologia-base).
- Avviare Firefox
- Entrare in modalità di navigazione anonima (in Firefox andare nel menu `Strumenti - Avvia navigazione anonima`, oppure premere Shift+Ctrl+P). Questo serve per evitare che Firefox modifichi al volo la cronologia-base.

Ora Firefox è configurato correttamente per eseguire la valutazione dei risultati delle query.

Fase di valutazione vera e propria

Aprire il file Excel o Calc che vi è stato fornito col vostro kit. La prima colonna della tabella `Risultati della valutazione della cronologia di base` con-

tiene le query a te assegnate. Alcune sono quelle da te segnalate, altre sono state segnalate da altri.

1. Digitare *completamente* (non fermarsi in mezzo alla parola) la prima query nella barra degli indirizzi di Firefox
2. Aspettare che la *awesome bar* proponga la lista dei suggerimenti
3. Valutare la prima pagina proposta [D1] (eventualmente cliccandola e navigandoci)
4. Scrivere il proprio giudizio di rilevanza (0 non rilevante, 1 abbastanza rilevante, 2 molto rilevante) nell'incrocio tra la prima query e il documento D1 della tabella Excel o Calc
5. Eventualmente riscrivere la query nella *awesome bar*, valutare il secondo documento proposto [D2] (eventualmente aprendolo e navigandoci) e scrivere il giudizio di rilevanza nella tabella all'incrocio tra la prima query e il documento D2
6. Ripetere dal punto 1 fino ad esaurire i documenti proposti (sono al massimo 10. Se sono meno di 10, lasciare vuota la cella corrispondente nella tabella)

Ripetere questa procedura per tutte le dieci query in modo da riempire la tabella `Risultati della valutazione della cronologia di base` del documento Excel (o Calc). Salvare il documento Excel (o Calc).

Bene. Hai terminato la fase di valutazione delle query sulla cronologia-base. Se hai ancora tempo, procedi alla valutazione delle query sulla cronologia-arricchita. Altrimenti la fase successiva può essere fatta anche in un secondo momento.

Valutazione della configurazione con cronologia arricchita

In questa fase l'utente è chiamato a dare giudizi di rilevanza ai documenti proposti dalla *awesome bar* di Firefox che pesca i risultati dalla cosiddetta cronologia-arricchita, cioè la cronologia web che è uguale per tutti, arricchita attraverso l'uso

del tool Java dei documenti locali [doc, docx, ppt, pptx, (e corrispondenti estensioni per OpenOffice) pdf, ps] presenti nel *file system* dell'utente.

Configurazione del sistema

[Chi lavora con un profilo di Firefox ad hoc, deve assicurarsi di usare quel profilo nell'intera fase di valutazione. Non dovrà eseguire il backup del profilo]

Configurare Firefox come spiegato nel paragrafo `Configurazione di Firefox` dell'*Appendice B* (sostanzialmente eseguire le operazioni in `about:config` e riavviare firefox).

[Se non lo si è ancora fatto, eseguire una copia di backup del profilo di Firefox]

1. Chiudere Firefox

2. Sostituire il file `places.sqlite` presente nella cartella del profilo di Firefox (il path della directory ve l'eravate salvato in un file testuale in fase di configurazione) con il file `places.sqlite` che ho fornito nel tuo kit (questo file places.sqlite è quello che contiene la cronologia-base)

3. In Windows, avviare il file `AwesomePlus.jar` facendo doppio-clic col mouse. In Linux lanciare il file `AwesomePlus.jar` da shell. Se tutto procede correttamente, verrà aperta una finestra a video dal titolo `Progetto` ***`Awesome++`*** `/ Melis Giovanni` con diversi bottoni (in Windows verrà aperta anche una seconda finestra per la Console. In Linux invece lo standard-output è scritto sulla shell).

4. Premere il bottone `Seleziona cartella/disco` e selezionare il drive o la cartella coi documenti da indicizzare. Se l'operazione è andata a buon fine, verrà stampato in console il messaggio

   `Cartella o drive selezionato: PATH_SELEZIONATO`

5. Premere il bottone `Seleziona database SQLite` e selezionare il file `places.sqlite` presente nella cartella del profilo di Firefox (la stessa che ave-

vate copia-incollato in fase di configurazione nel file testuale). Se l'operazione è andata a buon fine, verrà stampato in console un messaggio analogo a

```
Database SQLite selezionato: C:\Documents and Settings\Utente\
Dati Applicazioni\Mozilla\Firefox\Profiles\xxxxxxxx.default\
places.sqlite
```

6. Chiudere Firefox

7. [Questa fase può richiedere alcuni minuti] Premere sul bottone `Indicizzazione Locale`. In console saranno stampati i percorsi dei file che verranno indicizzati. Se l'operazione va a buon fine, in console viene stampata una dicitura analoga alla seguente (consultare la sezione `Possibili problemi/bug comuni` nell'Appendice precedente)

   ```
   Numero di documenti da inserire in cronologia: 5
   Tempo di elaborazione: 0.016
   Tempo stimato di arricchimento della cronologia: 2.25 secondi
   (circa 0 minuti)
   ```

   Preciso che il tempo stimato di arricchimento della cronologia è pesantemente sovrastimato per eccesso.

8. [Questa fase può richiedere alcuni minuti] Procedere quindi con l'arricchimento, premendo il bottone `Integrazione cronologia`. In console verrà stampata la percentuale di avanzamento. Se l'operazione va a buon fine, in console viene stampata una dicitura analoga alla seguente

   ```
   Integrazione della cronologia web con file da disco
   Avanzamento: 20%
   Avanzamento: 60%
   Avanzamento: 100%
   Tempo integrazione: 2.765 secondi (circa 0 minuti)
   Integrazione della cronologia con file da disco conclusa
   ```

9. Se non sono state stampate eccezioni sulla console, la configurazione può essere considerata conclusa e si può passare alla fase di valutazione vera e propria.

Con la procedura appena descritta, la cronologia-base è stata arricchita coi documenti presenti nel tuo computer ed è stato costruito correttamente il dataset su cui poter eseguire la nuova valutazione.

1. Avviare Firefox

2. Entrare in modalità di navigazione anonima (in Firefox andare nel menu `Strumenti - Avvia navigazione anonima`, oppure premere Shift+Ctrl+P). Questo serve per evitare che Firefox modifichi al volo la cronologia-base.

Ora Firefox è configurato correttamente per eseguire la valutazione dei risultati delle query.

Fase di valutazione vera e propria

- Digitare *completamente* (non fermarsi in mezzo alla parola) la prima query nella barra degli indirizzi di Firefox

- Aspettare che la *awesome bar* proponga la lista dei suggerimenti

- 1.alutare la prima pagina proposta [D1] (eventualmente cliccandola e navigandoci) NOTA: è noto che Firefox non apre alcune tipologie di documenti locali che invece sono state indicizzate. Se o quando vuoi consultare un documento locale, perciò, dovrai accederci manualmente e aprendolo col programma più opportuno.

- Scrivere il proprio giudizio di rilevanza (0 non rilevante, 1 abbastanza rilevante, 2 molto rilevante) nell'incrocio tra la prima query e il documento D1 della tabella Excel o Calc. ATTENZIONE: se quello che stai dando è un giudizio di rilevanza relativo a un documento locale, aggiungi un asterisco accanto al valore numerico (es.: 1*).

- Eventualmente riscrivere la query nella *awesome bar*, valutare il secondo documento proposto [D2] (eventualmente aprendolo e navigandoci) e scrivere il giudizio di rilevanza nella tabella all'incrocio tra la prima query e il documento D2

- Ripetere dal punto 1 fino ad esaurire i documenti proposti (sono al massimo 10. Se sono meno di 10, lasciare vuota la cella corrispondente nella tabella)

Ripetere questa procedura per tutte le dieci query in modo da riempire la tabella `Risultati della valutazione sulla cronologia arricchita con documenti locali` del documento Excel (o Calc). Salvare il documento Excel (o Calc).

La fase di valutazione è terminata. Inviami quanto prima il file Excel (o Calc) coi risultati della tua valutazione.

Ripristino del sistema

Per riprisinare il tuo sistema è sufficiente che:

- Ripristini il backup del tuo profilo di Firefox (dovrebbe essere sufficiente sostituire il file `places.sqlite` attualmente presente nel profilo con quello che avevate salvato nel backup)

- Entri in `about:config` e ripristini le tre voci modificate nella fase di configurazione ai loro valori di default

Bibliografia

[1] Bruce Croft, Donald Metzler, Trevor Strohman, *Search Engines: Information Retrieval in Practice*, Addison-Wesley, 2009.

[2] M. Agosti, M. Melucci, *Reperimento dell'Infromazione. Information Retrieval. Concetti, architetture e modelli dei motori di ricerca*. Università degli Studi di Padova, 2007.

[3] S. Chernov, G. Demartini, E. Herder, M. Kopycki, W. Nejdl, *Evaluating Personal Information Management Using an Activity Logs Enriched Desktop Dataset*, In Proceedings of 3rd Personal Information Management Workshop (PIM 2008), Florence, Italy, April 5-6, 2008.

[4] Beaza-Yates, Ribeiro Neto *Modern Information Retrieval*, Addison Wasley, First edit., 1999.

[5] Tom Noda and Shawn Helwig *Benchmark study of* desktop *search tools*, Best Practice Reports, UW E-Business Consortium, April 2005.

[6] Chang-Tien Lu, Manu Shukla, Siri H. Subramanya and Yamin Wu, *Performance evaluation of* desktop *search engines*, Information Reuse and Integration (IRI-2007), IEEE International Conference, August 2007.

[7] Stefania Costache, *Using Your Desktop as Personal Digital Library*, TCDL Bulletin, Current 2006, Volume 2, Issue 2 (URL: http://www.ieee-tcdl.org/Bulletin/v2n2/costache/costache.html).

[8] Paul-Alexandra Chirita and Wolfgang Nejdl, *Analyzing User Behavior to Rank Desktop Items*, International Conference on String Processing and Information Retrieval No13, Glasgow, ROYAUME-UNI (2006).

[9] Alexandru Chirita, Julien Gaugaz, Stefania Costache, Wolfgang Nejdl, *Desktop Context Detection Using Implicit Feedback*. In Proceedings of the Workshop on Personal Information Management held at the 29th ACM International SIGIR Conf. on Research and Development in Information Retrieval, Seattle, USA.

[10] Sergey Chernov, Pavel Serdyukov, Paul-Alexandru Chirita, Gianluca Demartini, and Wolfgang Nejdl, *Building a Desktop Search Test-bed*. In Proceedings of 29th European Conference on Information Retrieval (ECIR), Rome, Italy, 2-5 April, 2007.

[11] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, Fernando Diaz, *Towards recency* rank*ing in web search*. In WSDM '10: Proceedings of the third ACM international conference on Web search and data mining (2010), pp. 11-20.

[12] Laura A. Granka, Thorsten Joachims, Geri Gay. *Eye-tracking analysis of user behavior in WWW search*, SIGIR, 2004.

[13] Ecosia. URL: http://ecosia.org/

[14] Firefox, pagina ufficiale in italiano.
URL: http://www.mozilla-europe.org/it/firefox/

[15] Mozilla Foundation, pagina ufficiale, URL: http://www.mozilla.org/

[16] Mozilla Developer Center, URL: https://developer.mozilla.org/

[17] SQLite, pagina ufficiale, URL: http://www.sqlite.org/

[18] API Java per SQLite, URL: http://www.zentus.com/sqlitejdbc/

[19] Estensione per Firefox, SQLite Manager, URL: https://addons.mozilla.org/en-US/firefox/addon/5817

[20] Definizione di *frecency* su Wiktionary,
URL: http://en.wiktionary.org/wiki/frecency

[21] Guida al calcolo di *frecency*
URL: https://developer.mozilla.org/en/The_Places_frecency_algorithm

[22] TREC, sito ufficiale, URL: http://trec.nist.gov

[23] Amazon Mechanical Turk, sito ufficiale, URL: https://www.mturk.com/mturk/