UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria
Corso di Laurea in Ingegneria dell'Informazione

Tesi di Laurea Triennale

Studio e realizzazione di una interfaccia web per l'analisi dei log di interazione utente di biblioteche digitali multilingua

Relatore:
Prof. Giorgio Maria Di Nunzio

Laureando:
Marco Collautti

Anno Accademico 2010-11

Indice

Capitolo 1

Introduzione

L'esperienza giornaliera di navigazione che tutti noi possediamo e l'uso intensivo dei motori di ricerca come mezzi di reperimento di informazioni, portano l'attenzione dei gruppi di ricercatori sullo studio e l'analisi del ruolo centrale che oggigiorno è ricoperto da tali siti web nella crescita e nell'utilizzo della rete internet. Come riporta Alexa.com[1], infatti, al primo posto tra i 500 siti più visitati al mondo troviamo Google [1]: il più famoso motore di ricerca. Nella lotta per conquistare il maggior numero di utenti ciò che più è importante, e che decreta il successo di un servizio sugli altri, è la capacità di mostrare all'utente tra i primi risultati di una ricerca quelli che con buona probabilità sono i più vicini alle sue preferenze e ai suoi interessi.

La maggior parte dei web server[2] registrano automaticamente le richieste degli utenti durante la navigazione in particolari tipi di file chiamati log. I log non sono altro che una registrazione cronologica delle operazioni man mano che vengono eseguite, questi dati normalmente includono: le richieste *HTTP*, gli indirizzi IP dei visitatori, il tipo di browser utilizzato e la lingua in cui è utilizzato, la data e l'ora della visita. Quello che l'utente cerca col motore di ricerca (può esso essere una parola, una frase o molto più genericamente una sequenza di caratteri) è chiamato "query".

Tramite l'analisi dei dati di log di un motore di ricerca è possibile valutare la qualità del servizio offerto, valutazione che si rende particolarmente importante quando il sito web è utilizzabile in più lingue. Ciò che è fonda-

[1] http://www.alexa.com/ : Azienda statunitense sussidiaria di Amazon.com che si occupa di statistiche sul traffico di Internet.

[2] Per web server si intendono il software e l'hardware che si occupano di memorizzare e fornire agli utenti che ne fanno richiesta le pagine web visitate durante la navigazione.

mentale è che l'interazione tra l'utente e un sistema di accesso a determinate informazioni possa essere studiata e analizzata, sia per raccogliere le preferenze degli utenti, sia per poter imparare che cosa piace ed interessa di più. I dati di log possono essere analizzati per studiare l'uso dei motori di ricerca e per meglio adattare gli stessi ai risultati che gli utenti si aspettano di raggiungere quando li utilizzano. Uno dei maggiori ostacoli per la ricerca in questo campo è il fatto che gran parte dei servizi di ricerca tengono sì traccia dei loro log sui server, ma non sempre è possibile avere pubblico accesso agli stessi.

Nel 2009 un workshop tenuto a Londra [2], nell'ambito della coordination action TrebleCLEF[3], ha avuto tra i suoi argomenti principali l'utilizzo e la distribuzione di dati di log: come questi dati debbano essere resi pubblici per scopi di ricerca, se si debba raccogliere questi dati pensando già ad uno specifico obiettivo e come, e quali informazioni aggiuntive debbano essere collezionate e correlate con le query di ricerca.

In generale, non è possibile disporre di dati recenti e a lungo termine, rendendo la verificabilità e la ripetibilità degli esperimenti molto complicate. Ogni ricercatore compie i propri studi su differenti dati ed è impossibile poter confrontare i lavori dei diversi gruppi di ricerca. Anche se due gruppi lavorassero su una base di dati proveniente dallo stesso sito non è detto che lavorino su dati presi dallo stesso periodo, rendendo il confronto tra i diversi lavori ancora una volta complesso.

A livello internazionale esistono delle campagne di valutazione per confrontare in maniera sperimentale l'efficacia e l'efficienza dei motori di ricerca. La Cross-Language Evaluation Forum (CLEF)[4] è una di queste campagne. Nel 2009 in una track chiamata LogCLEF[5] ci fu un primo tentativo di rilasciare una collezione di dati di log con l'obiettivo specifico della verificabilità e ripetibilità. Un traguardo a lungo termine è quello di stimolare l'interesse dei ricercatori sui comportamenti degli utenti in ambienti multilingua e di proporre degli standard per la valutazione dei dati di log. Un'altra finalità di LogCLEF è l'analisi e la classificazione di query, il poter decretare il successo, o l'insuccesso, di una ricerca in modo da poter comprendere meglio quali siano i comportamenti di ricerche in ambito multilingue [3]. Nelle prime due edizioni di LogCLEF sono stati messi a disposizione dei partecipanti diverse

[3]http://www.trebleclef.eu/

[4]http://www.clef-campaign.org/

[5]http://www.uni-hildesheim.de/logclef/

collezioni di dati provenienti da diversi siti web tra i quali troviamo The European Library[6].

The European Library è un servizio che offre accesso alle risorse di 48 biblioteche nazionali d'Europa per un totale di 35 lingue diverse. Il suo scopo è quello di fornire una vasta collezione di materiali di diverse discipline e di offrire ai suoi visitatori un accesso al patrimonio culturale europeo. The European Library ospita risorse che possono essere sia digitali (libri, poster, mappe, registrazioni sonore, video, ecc.) sia bibliografiche. Le 48 biblioteche nazionali europee che partecipano al progetto sono tutte membri della Conferenza delle librerie nazionali (CENL)[7], una fondazione che punta a migliorare e rinforzare il ruolo delle librerie nazionali in Europa. Per l'Italia fanno parte della CENL le librerie nazionali di Firenze e Roma.

Dal secondo anno ai partecipanti di LogCLEF vennero forniti degli obiettivi ben precisi su cui focalizzare il proprio lavoro, ad esempio: si è richiesto di identificare la lingua di una query e di effettuare un confronto tra la lingua della query e l'indirizzo IP di chi ha eseguito quella query. In più altri possibili studi potevano essere fatti sull'uso di lingue diverse nella stessa ricerca oppure sul confronto tra la nazione della libreria, la lingua della query e la lingua dell'interfaccia di accesso al sito web. [2]

Alcune delle informazioni disponibili a chi lavorasse sui dati di log di The European Library erano:

- Indirizzo IP dell'utente;
- Una stringa alfanumerica autogenerata che identifica una sessione;
- Il contenuto della query;
- Tipo di azione compiuta dall'utente;
- Un codice identificativo della collezione che è interessata dalla ricerca;
- Ora e data dell'esecuzione dell'azione.

Solo per l'anno 2010 sono stati messi a disposizione circa 950,000 record di log.

[6] http://www.theeuropeanlibrary.org/
[7] http://web3.nlib.ee/cenl/

1.1 Obiettivi di LogCLEF 2011

In base alle idee e alle proposte sviluppate durante LogCLEF 2010, per l'anno 2011 sono state proposte tre tematiche [4]:

1. Identificazione della lingua: ai partecipanti è richiesto di riconoscere la lingua di una query;

2. Classificazione della query: ai partecipanti è richiesto di scegliere per ciascuna query una particolare categoria di interesse;

3. Successo di una query: ai partecipanti è richiesto di studiare il successo di una query. Il successo può essere definito in termini di tempo speso su una pagina, numero di elementi cliccati o tipo di azioni compiute durante la consultazione dei risultati di una ricerca.

Per quanto riguarda la prima tematica lo scopo è quello di creare un database di query multilingua classificate che costituiscano un insieme di dati da prendere come modello di riferimento in quanto a correttezza della scelta della lingua. Infatti se si avesse un algoritmo di riconoscimento linguistico e si fosse interessati a valutarne la validità nel compito dell'identificazione della lingua di alcune query, una possibilità potrebbe essere quella di confrontare le lingue scelte da utenti umani con i risultati ottenuti con questo algoritmo. Nel caso in cui le scelte fatte dal programma automatico di riconoscimento rispecchiassero in alta percentuale quelle fatte dalle persone vere, allora si potrebbe decretare il successo di questo software. Sebbene per alcuni tipi di query la scelta della lingua non dovrebbe rappresentare una grossa difficoltà (sarà semplice scegliere per "La Bibbia" la lingua italiana e per "The Bible" quella inglese), ci sono altri tipi di query che rendono difficile scegliere una lingua sulle altre (per esempio quando la query è il nome proprio di un autore), e dato che le query su cui si lavora provengono dai log di ricerche fatte su cataloghi di biblioteche spesso ci si potrà trovare in una simile difficoltà.

Quando ai partecipanti viene richiesto di classificare la query si richiede che scelgano tra diverse opzioni, le categorie tra le quali si può scegliere sono:

- Persone (che includono nomi, istituzioni e organizzazioni);

- Luoghi geografici;

- Eventi (eventi storici);
- Titoli di opera;
- Categorie specifiche (generalmente in latino, per esempio indicano specie animali, vegetali);
- Altro.

Per la terza proposta ai partecipanti viene richiesto di studiare il successo di una ricerca. Viene mostrata una sessione compiuta da un navigatore sul sito di The European Library e si chiede di classificare questa sessione secondo diversi parametri. Infatti quando viene eseguita una ricerca possono presentarsi diversi scenari [5]: in generale l'utente dopo una ricerca iniziale può totalmente modificare le parole chiave utilizzate a seconda dei risultati che otterrà. Quando la query viene modificata si può trattare di una generalizzazione: i risultati ottenuti sono troppo selettivi e quindi la query iniziale viene riformulata utilizzando termini più generici; un altro caso è quello di una specificazione: è il contrario della generalizzazione, cioè l'utente che compie una ricerca si rende conto che i risultati sono troppo generici e quindi modifica la query per avere risultati più specifici. Infine può essere il caso di drifting, la query viene modificata mantenendo lo stesso grado di specificità ma magari cercando un aspetto diverso dello stesso argomento.

È in questo contesto che si colloca l'operato del mio lavoro di tesi. Si e' deciso di mettere a disposizione dei partecipanti di LogCLEF 2011 un'interfaccia per creare e collezionare un insieme di annotazioni manuali sui record di log in modo semplice, veloce ed intuitivo in modo da poter adempiere a ciascun task proposto. Il mio compito è stato quello di progettare, e realizzare un'interfaccia web utilizzabile per la valutazione e l'analisi delle query multilingua provenienti dai log del sito The European Library. Il risultato del mio lavoro è un sito web disponibile all'indirizzo
`http://ims.dei.unipd.it/websites/LogCLEF/Logs/login.php`
a cui si accede tramite autenticazione; al momento il sito viene utilizzato da vari gruppi di ricerca in tutta Europa. Contestualmente ai task di LogCLEF 2011 agli utenti che accedono al sito viene chiesto di rispondere a specifiche domande relative ad una determinata query che gli verrà proposta. Le annotazioni collezionate tramite questo sito vengono ogni volta salvate e verranno utilizzate nei modi e nei tempi previsti dalle specifiche di LogCLEF 2011.

Capitolo 2

Progettazione e realizzazione del database di annotazioni

Solo per l'anno 2010 dal sito The European Library sono stati raccolti circa 950,000 dati. Quest'imponente mole di informazioni deve essere organizzata in un modo tale da essere facilmente consultabile ed utilizzabile dai partecipanti di LogCLEF 2011. Inizialmente questi dati di log sono distribuiti sotto forma di file .csv (comma separated value). Ogni file contiene i dati di log di un intero mese; i file da gestire sono quindi in tutto 12, uno per ogni mese del 2010. I file .csv sono facilmente utilizzabili tramite un qualsiasi programma di gestione di fogli elettronici; sono organizzati come delle tabelle dove ogni riga corrisponde ad un record, ed ogni campo del record viene separato da quelli attigui da un simbolo di separazione del tipo "," oppure ";".

Vista la quantità di dati presente per ogni file i software di gestione dei fogli elettronici non si rivelano essere i migliori candidati per manipolarli. Infatti quello che serve è un rapido e semplice accesso ai dati, in maniera automatizzata e diretta da un sito web accessibile ai partecipanti di LogCLEF 2011. Inoltre ogni volta che un utente risponderà ad una domanda relativa ad una query la sua annotazione dovrà essere adeguatamente salvata ed organizzata in un altro file. È per questo che tutti i dati devono essere immagazzinati in un apposito database.

Un database, o base di dati, indica un insieme di archivi elettronici collegati secondo un particolare schema logico in modo tale da rendere possibile la gestione dei dati stessi in maniera semplice ed efficace, e di permettere

particolari operazioni quali: inserimento di nuovi dati, ricerca, cancellazione ed aggiornamento. I database più moderni (come quello che verrà qui utilizzato) sono basati su un modello relazionale. Generalmente un database è diviso in più tabelle e nella teoria delle basi di dati una tabella corrisponde ad una relazione, cioè corrisponde ad una rappresentazione visuale di essa. Nello specifico una relazione è la definizione di una tabella, insieme ai dati che vi vengono memorizzati.[6] Le tabelle sono organizzate per colonne chiamate attributi.

Un dato viene generalmente chiamato tupla (o record): una tupla è quindi formata da più attributi. Nella maggior parte dei casi siamo interessati ad avere, in una data tabella, l'unicità dei record: non ci possono essere dati uguali, o comunque ciò che è importante è riuscire ad identificare univocamente una tupla. Questa unicità è garantita fissando, per ogni tabella, una chiave primaria che non è altro che un insieme di attributi che serve ad essere sicuri che in una data tabella non ci siano tuple i cui valori per questi attributi siano uguali. Nessun record nella tabella può avere l'attributo identificato dalla chiave primaria identico a quello di un qualsiasi altro record della stessa (vincolo di unicità): il tentativo di inserimento di un tale record genera un errore di violazione della chiave primaria.

2.1 Progettazione e realizzazione

Quando abbiamo a che fare con dei database non ci resta che scegliere, tra i vari a disposizione, quale software di gestione si adatti meglio alle nostre necessità. Per la gestione di questo database è stata fatta la scelta di utilizzare *PostgreSQL*[1]: è un database relazionale che usa il linguaggio *SQL* per eseguire delle query sui dati. Il linguaggio *SQL* è un linguaggio testuale interattivo, che permette di interrogare e gestire i database mediante l'utilizzo di query. Con *SQL* si possono inserire, leggere, modificare o cancellare i dati nel database. Alcuni dei comandi basilari (usati con gli appositi parametri) possono essere[2]:

- CREATE: permette di creare un intero database o delle tabelle;
- INSERT: consente di inserire un dato in una certa tabella;

[1] http://www.postgresql.org/

[2] Per una esauriente lista dei possibli comandi si veda: http://www.postgresql.org/docs/9.0/static/sql-commands.html

- SELECT: viene utilizzato per fare una ricerca che soddisfi determinati criteri;
- DROP: utile quando vogliamo cancellare una tabella e tutti i dati in essa contenuti.

PostgreSQL può essere utilizzato sia attraverso una comoda interfaccia grafica, sia tramite la linea di comando ed è compatibile con tutti i sistemi operativi più diffusi.

Il database che viene gestito è costituito da un totale di 29 tabelle e i dati provenienti dai file .csv distribuiti ai partecipanti sono organizzati nelle tabelle ***actionlog***. È stata fatta la scelta di crearne una diversa per ogni mese dell'anno 2010 e ognuna verrà chiamata ***actionlog2010_i*** dove i è un numero da 1 a 12 a seconda del corrispondente mese; queste tabelle costituiscono la base da cui sviluppare l'intera architettura del database. Successivamente vengono introdotte altre 12 tabelle (sempre una per ogni mese) chiamate ***sessionlog2010_i***, dove i ha analogo significato. Queste ulteriori tabelle servono ad organizzare meglio i dati secondo caratteristiche che descriverò più avanti suddividendoli per sessioni di navigazione distinte. La tabella ***actor*** è quella che si occupa di gestire i profili dei partecipanti a LogCLEF: viene tenuta traccia degli username e password scelti per accedere al sito web. In seguito ho creato le tabelle ***decision***, ***extendedDecision***, ***sessionDecision***, ***categoryDecision*** che servono a memorizzare le annotazioni fatte dai partecipanti.

2.2 Tabelle ActionLog

In questa sezione presentiamo la definizione di ciascuna tabella: gli attributi (colonne) che le compongono e come vengono organizzati i dati in esse. Gli attributi in ordine sono:

- *id*: identificatore di un record. Il suo utilizzo è quello di costituire la chiave primaria della tabella, le diverse tuple saranno quindi univocamente identificate solo da questo campo;
- *userid*: identificatore dell'utente; nel caso in cui l'utente non abbia effettuato il login al sito allora verrà impostato a "guest";
- *userip*: l'indirizzo IP dell'utente (semi oscurato per motivi di privacy);

- *sesid*: è una stringa alfanumerica autogenerata il cui scopo è identificare una sessione di navigazione effettuata sulle pagine di The European Library. Quando un host si connette ad una pagina un determinato *sesid* viene associato direttamente alla macchina. Tramite questo attributo quindi non ci sarà possibile sapere se, per esempio, la sessione è stata compiuta da una, o più, persone diverse. Tutto quello che sappiamo è che un determinato terminale ha eseguito una certa sequenza di determinate azioni (di cui c'è traccia nel nostro Log) chiamata sessione;
- *lang*: identifica la lingua in cui si è impostata l'interfaccia del sito. Quando un utente accede alle pagine di The European Library potrà, se vuole, cambiare la lingua in cui esse sono mostrate. La lingua di default, in cui è visualizzato il sito, è l'inglese, per cui nella maggioranza dei casi ci troveremo a esaminare tuple in cui l'attributo *lang* sarà impostato a inglese, dato che molti utenti non andranno a modificare questa impostazione;
- *query*: è l'attributo più importante, è il testo digitato dall'utente quando esegue una ricerca sul sito;
- *action*: identifica il tipo di azione compiuta dall'utente (può essere per esempio una ricerca semplice, o avanzata ...);
- *colid*: l'identificatore della collezione interessata dall'azione dell'utente;
- *nrRecords*: il numero di record restituiti dalla collezione interessata dall'azione eseguita dall'utente;
- *recordPosition*: posizione dell'elemento nella lista complessiva dei record;
- *sboxid*: identificatore di una eventuale casella di ricerca remota;
- *objurl*: l'indirizzo URL dell'oggetto interessato;
- *date*: data e ora in cui una certa azione è stata eseguita.

Ora che abbiamo visto come sono definite le 12 tabelle ***actionlog*** non ci resta che crearle e in seguito popolarle con i dati proveniente dai log di The European Library e che vengono forniti sotto forma di file .csv. La creazione di una delle 12 tabelle si ottiene eseguendo l'istruzione:

```
CREATE TABLE actionlog2010_1
(
  id bigint NOT NULL,
  userid character varying(25) NOT NULL,
  userip character varying(15) NOT NULL,
  sesid character varying(32) NOT NULL,
  lang character varying(3) NOT NULL,
  query character varying(250),
  "action" character varying(30),
  colid character varying(250),
  nrrecords integer,
  recordposition character varying(25),
  sboxid character varying(50),
  objurl character varying(250),
  date timestamp without time zone,
  CONSTRAINT actionlog2010_pkey1 PRIMARY KEY (id)
);
```

Il nome della tabella, che è ***actionlog2010_1*** ci indica che stiamo creando la tabella ***actionlog*** relativa ai log di gennaio 2010, tutti gli altri dati corrispondono ai campi che abbiamo elencato prima. I vari *character varying(x)* stanno ad indicare che i dati che appartengono a quell'attributo saranno delle stringhe di lunghezza variabile fino ad un massimo di x caratteri. Quando invece leggiamo *integer* allora sappiamo che quei valori sono numerici. Il valore *NOT NULL* sta ad indicare che ogni record inserito nella tabella dovrà avere quell'attributo specificato (non può essere lasciato vuoto).

2.2.1 Popolamento delle tabelle ActionLog

Adesso che le tabelle sono state create non resta che popolarle con i dati proveniente dai file .csv. Per fare questo esiste un comodo comando di *PostgreSQL* che permette di fare correttamente l'import usando la virgola come spaziatore, cioè all'interno di ogni riga i dati vengono divisi nei corrispondenti attributi da una virgola. Il comando è:

```
copy actionlog2010_1 FROM 'log2010-01-01_2010-01-31.csv'
WITH DELIMITER ',' CSV
```

dove in questo caso il nome del file .csv e della tabella, ci indicano che stiamo facendo l'import per i dati di gennaio 2010.

id	userid	userip	sesid	lang	query	action
25123219	guest	85.238	FD6E831814..	hu	("stephen hawking")	view_full

colid	nrrecords	recordposition	sboxid	objurl	date
a0071	69	7		http://search.theeuropeanlibrary.org/...	2010-01-01 12:19:16.857

Figura 2.1: Esempio di dati presenti nella tabella ActionLog

2.3 Tabelle SessionLog

A questo punto si è interessati a raggruppare tutti i dati presenti nelle tabelle ***actionlog*** in un modo particolare e che sia utile all'adempimento corretto dei task di LogCLEF. Per esempio, abbiamo visto come uno degli obiettivi sia quello di valutare il successo di una ricerca da parte di un utente. Per valutarne il successo viene mostrata una sessione intera compiuta da un utente sul sito di The European Library e si chiede di classificare questa sessione secondo alcuni criteri basati sull'interazione dell'utente con i risultati della ricerca. Per rispondere correttamente a questo compito si richiede di mostrare ai partecipanti al progetto un'intera sessione di navigazione. Esiste per questo il campo *sesid*, che come abbiamo visto viene associato direttamente ad una macchina e ne identifica una sessione, che non è altro che un susseguirsi di azioni di cui il log tiene traccia. Ci possono essere casi però in cui, da una macchina, venga eseguita una ricerca, per esempio, alle 9 del mattino, poi vengano consultati i risultati e poi venga effettuata un'altra ricerca, totalmente scorrelata dalla prima 3 ore dopo. È chiaro il fatto che ci troviamo davanti a due sessioni diverse, ma la tabella ***actionlog***, così com'è, difficilmente potrà aiutarci in tale senso. È per questo che sono state create le tabelle ***sessionlog2010_i*** dove i è un numero da 1 a 12 a seconda del corrispondente mese. Le tabelle sono così definite:

```
CREATE TABLE sessionlog2010_1
(
  id bigint NOT NULL,
  sesid character varying(32) NOT NULL,
  date timestamp without time zone,
  counter character varying(32),
  CONSTRAINT sessionlog_pkey1 PRIMARY KEY (id)
);
```

Gli attributi, a parte *counter*, hanno lo stesso significato (e valore) dei corrispondenti attributi delle tabelle ***actionlog***. L'attributo *counter* è stato

qui introdotto in quanto servirà ad identificare le sessioni diverse. Ad ogni sessione di navigazione viene assegnato un diverso *counter*, e ci potrà essere il caso in cui più record abbiano identico *sesid* ma *counter* diverso. È stato deciso che i record con identico *sesid* vengano separati in *counter* diversi quando ci sia un ritardo, tra un'azione e l'altra, di più di 30 minuti.

2.3.1 Popolamento delle tabelle SessionLog

È a questo punto che introduco il secondo linguaggio di programmazione qui utilizzato: *PHP* (hypertext preprocessor)[3]. In particolare *PHP* è un linguaggio di scripting: questi tipi di linguaggi sono stati creati per interagire con altri programmi molto più complessi. In origine i linguaggi di scripting erano sequenze di comandi che invece di essere digitati uno alla volta nella shell di sistema venivano scritti su un file di testo e poi eseguiti tutti insieme in maniera automatizzata. Il *PHP* viene usato principalmente in applicazioni web lato server e bene si adatta al funzionamento di pagine *HTML*. Il motivo principale per cui uso *PHP* è la perfetta interazione con *PostgresSQL*, per il quale offre un semplice e potente interfacciamento, sia per la consultazione che per la modifica di database.

Per poter popolare la tabella ***sessionlog*** a partire dai dati della tabella ***actionlog*** ho creato uno script chiamato `fillSessionLog.php` . Lo scopo dello script `fillSessionLog` è quello, scelto uno dei 12 mesi, di importare tutti i dati dalla tabella ***actionlog*** corrispondente e di separare i record con *sesid* diversi in sessioni diverse, assegnando ad ognuna un *counter* diverso. Inoltre, nel caso in cui ci si trovi ad avere record con lo stesso *sesid* ma le cui azioni siano state eseguite con più di 30 minuti di differenza, allora si dividerà la sessione in più parti, assegnando anche in questo caso *counter* diversi.

Per prima cosa importiamo tutti i dati necessari nella tabella ***sessionlog*** con l'istruzione:

```
INSERT into $sessionTable (id, sesid,date)
SELECT id, sesid,date FROM $actionTable ORDER BY id;
```

Questa operazione copia gli attributi *id, sesid, date* dalla tabella ***$actionTable*** nella tabella ***$sessionTable***[4] ordinandoli per *id*. Per cominciare

[3] http://www.php.net/

[4] Dato che lo script deve essere valido per tutte e 12 le tabelle, all'inizio del file `fillSessionLog` sono presenti le variabili *$actionTable* e *$sessionTable*. Basterà impostare

ad assegnare i corretti *counter* si fa una prima scansione di tutti i dati nella tabella ***sessionlog*** e ogni volta che si trova un *sesid* diverso viene assegnato un *counter* diverso; record con *sesid* uguale avranno, per ora, *counter* uguale (incurante del tempo passato tra un'azione e l'altra). Il *counter* non è altro che un numero, quindi il primo record analizzato avrà *counter* = 1, e così tutti i record con lo stesso *sesid*. Quando si trova un *sesid* diverso si assegna *counter* = 2 e così via incrementando ogni volta. Questo procedimento viene fatto con un'istruzione di `UPDATE`, istruzione prevista dal linguaggio SQL, che permette di aggiornare i dati presenti in una tabella modificandone uno o più attributi.

Ora si rende necessario eseguire la seconda azione: separare i record con *sesid* uguale ma le cui azioni siano state eseguite con troppa distanza per poterle definire appartenenti alla stessa sessione di navigazione. Quello che serve ora è avere tutti i dati nella tabella ordinati per *counter* (in modo da avere tutti i record con lo stesso *sesid* vicini) e in ordine cronologico. Nel caso in cui ci si trovasse ad avere due record che abbiano stesso *sesid* e quindi stesso *counter* (diciamo per esempio sia *counter* = 1234) ma che debbano essere separati in due sessioni diverse, allora si mantiene il primo counter invariato e al secondo record viene modificato il *counter* concatenando una lettera alla fine (il *counter* diventerà 1234a). Se fosse necessario dividere in ulteriori sessioni allora al posto del carattere "a" si potrà utilizzare "b" e così via. Lo script `fillSessionLog` procede ora ad analizzare in questo modo tutti i record nella tabella a coppie di due, prendendo due record contigui:

- Dato che è disponibile nella tabella l'attributo *date*, ma questi tipi di dati non sono in una forma direttamente confrontabile, viene calcolato il timestamp dei due record, cioè il numero di secondi che sono passati dal 01/01/1970. Successivamente viene calcolata la differenza tra i due valori;

- Se il *counter* del primo record è solo numerico (non ha nessuna lettera concatenata alla fine) ed è uguale al *counter* del secondo record e se la differenza tra i due è maggiore a $30 * 60 = 1800$ secondi allora al secondo counter viene concatenata alla fine una "a" (da 1234 diventerà 1234a);

a dovere il nome di queste due prima di eseguire lo script per adattarlo a tutti e 12 i mesi di log.

ID	SESID	DATE	COUNTER
25122990	7ACFABCF86A879ABC 0FD05F2F4D5A034	2010-01-25 15:12:48.576	1
25123037	8A67EDAD394C71312 4BC0730025D04A9	2010-01-25 16:06:46.434	2
25123038	8A67EDAD394C71312 4BC0730025D04A9	2010-01-25 16:59:54.725	2a
25123042	0FBAD7F94EEA6087B B35E1F65FA67B8A	2010-01-25 17:42:17.106	3
25123032	0FBAD7F94EEA6087B B35E1F65FA67B8A	2010-01-25 18:26:21.667	3a
25123112	0FBAD7F94EEA6087B B35E1F65FA67B8A	2010-01-25 18:26:25.879	3a
25123115	0FBAD7F94EEA6087B B35E1F65FA67B8A	2010-01-25 21:13:37.767	3b

Figura 2.2: Esempio di diversi *counter* presenti nella tabella SessionLog

- Se l'ultimo carattere del *counter* del primo record non è numerico (è quindi un carattere, sia esso "a", "b",...) allora se la parte numerica del primo è uguale al secondo *counter* e se la differenza tra i due timestamp:
  - è maggiore a 1800 secondi: al secondo record viene assegnato un nuovo *counter* incrementando di uno (in ordine alfabetico) il carattere finale del primo *counter*;
  - è minore a 1800 secondi: il secondo *counter* viene reso uguale a quello del primo;

Questo della corretta identificazione delle sessioni di navigazione è un problema noto in questo campo e, solitamente, si procede nello stesso modo in cui si è proceduto qui. Si sceglie arbitrariamente una soglia temporale e si dividono le azioni in sessioni diverse secondo questa soglia.

Capitolo 3

Progettazione ed implementazione dell'interfaccia web

In questo capitolo presentiamo la progettazione e l'implementazione del sito web per l'annotazione delle query. Abbiamo visto come i dati provenienti dai dati di log di The European Library siano stati collezionati nelle tabelle ***actionlog*** e ***sessionlog***; questa organizzazione è stata un passaggio preliminare per preparare al meglio tutte le informazioni necessarie al corretto svolgimento dei task assegnati ai partecipanti all'edizione 2011 di LogCLEF.

Data una query, agli utenti è richiesto di:

1. Identificarne la lingua;
2. Classificarla secondo particolari categorie di interesse;
3. Valutarne il successo analizzando le azioni compiute durante la sessione di navigazione corrispondente.

Il sito web è così organizzato: dopo che si è effettuato l'accesso tramite username e password all'utente viene mostrata una query presa a caso dal database a disposizione; sequenzialmente vengono mostrati alcuni dettagli su questa query, ogni volta che un dettaglio è mostrato viene chiesto all'utente di rispondere a particolari domande; la risposta data (la sua annotazione) viene salvata in un apposita tabella e l'utente prosegue nel soddisfare gli obiettivi preposti. Quando il partecipante avrà risposto a tutte le

domande prestabilite riguardo quella query potrà decidere di lasciare il sito o di continuare valutando una query diversa.

3.1 Strumenti utilizzati

Gli strumenti utilizzati per la realizzazione del sito sono:

- HTML
- PHP
- SQL
- CSS

L'*HTML* è il linguaggio standard con cui sono scritte le pagine web, descrive le modalità di impaginazione, formattazione o visualizzazione grafica del contenuto, testuale e non, di una pagina web attraverso quelli che sono chiamati tag di formattazione. Lo scopo di un browser, quando si accede ad una pagina durante la navigazione, è quello di leggere il file *HTML* e visualizzarlo sotto forma di pagina web, il browser non mostra i tag usati ma li usa per interpretare il contenuto della pagina. È per questo che è chiamato un linguaggio di markup e non è un linguaggio di programmazione. Il *CSS*, invece, è un linguaggio utilizzato per definire la sola formattazione dei documenti *HTML*, verrà utilizzato alla fine del lavoro per avere impatto sull'aspetto grafico delle pagine web realizzate e renderle più piacevoli alla vista e all'utilizzo. L'*SQL* e il *PHP* sono stati introdotti rispettivamente al paragrafo 2.1 a pagina 8 e al paragrafo 2.3.1 a pagina 13.

Una parte fondamentale del sito, ed è principalmente quello che l'utente vede e con cui interagisce durante la navigazione, è costituita dai *form HTML*. I *form* sono una sezione di un documento *HTML* che contengono speciali elementi chiamati controlli (possono essere caselle da spuntare, menù a tendina, ecc.)[1] e vengono utilizzati quando c'è la necessità che l'utente fornisca determinati dati al server. I *form* in generale permettono agli utenti di immettere un input testuale che venga inviato al server, oppure di effettuare una scelta tra più possibilità. Nel nostro caso i *form* vengono utilizzati per permettere ai partecipanti di eseguire le loro annotazioni che verranno poi inviate al server e memorizzate.

[1] http://www.w3.org/TR/html4/interact/forms.html

Figura 3.1: Esempio di *Form HTML* per inserire nome e cognome

Affinché i dati inseriti nei *form* vengano inviati effettivamente al server è necessario inserire un tasto di "Submit". Quando l'utente clicca su tale tasto allora i dati vengono inviati ad una determinata pagina specificata nell'attributo *action* del *form*.

Un esempio di un form che permette all'utente di fornire nome e cognome è realizzato con il seguente codice:

```
<form action="prova.php" method="post">
Nome: <input type="text" name="nome" />
Cognome: <input type="text" name="cognome" />
<input type="submit" value="Submit" />
</form>
```

Tramite questo *form* l'utente fornisce il nome e il cognome. I due valori così inseriti andranno ad inizializzare due variabili: *$_POST['nome']* e *$_POST['cognome']* che saranno utilizzabili solo dalla pagina `prova.php` .

Prima di procedere ad una dettagliata descrizione della progettazione e dell'implementazione del sito web si elencano qui i file che lo costituiscono:

- `pg_connect.php`: script utilizzato per la connessione al database;
- `login.php`: gestisce l'autenticazione al sito web;
- `logout.php`: gestisce l'uscita dal sito web;
- `interact.php`: annotazione manuale sulla lingua della query proposta;
- `interact2.php`: seconda annotazione manuale sulla lingua;
- `interact3.php`: annotazione sul successo di una query conosciuta l'intera sessione di navigazione;
- `interact4.php`: annotazione a riguardo della categoria della query;
- `finalStep.php`: pagina finale che si occupa di indirizzare l'utente verso l'annotazione su una nuova query o verso l'uscita dal sito.

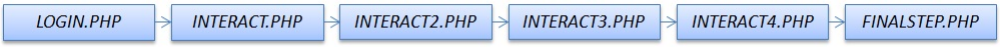


Figura 3.2: Sequenza in cui i partecipanti visitano le pagine

3.2 Gestione partecipanti a LogCLEF 2011

La collezione dei partecipanti a LogCLEF 2011 che accedono al sito e compiono annotazioni manuali sulle query è gestita con la tabella ***actor***. Con questa tabella si tiene traccia delle persone che sono autorizzate ad accedere al sito e inoltre è possibile associare ogni annotazione fatta e salvata nell'apposita tabella (che vedremo più tardi) con la persona che l'ha eseguita. La tabella ***actor*** è creata tramite il comando:

```
CREATE TABLE actor
(
 id character varying NOT NULL,
 pwd character varying (15) NOT NULL,
 fullname text,
 contactmail text,
 country isocountrytwo,
 lang isolanguagetwo,
 CONSTRAINT actor_pkey PRIMARY KEY (id)
);
```

Il tipo di dato *text* indica un valore testuale senza limiti di lunghezza. Gli attributi *country* e *lang* sono stati creati a parte tramite un'istruzione del tipo `CREATE TYPE` e corrispondono ai codici di due lettere che identificano le nazioni e le lingue conformi allo standard ISO 3166-1:1997[2] e ISO 639-1:2002[3].

Per ora il popolamento della tabella viene fatto in maniera manuale: tramite l'esecuzione di una query *SQL* vengono inseriti nella tabella i dati di un partecipante; non si esclude tuttavia in futuro di creare una pagina apposita per la registrazione dei nuovi utenti in maniera automatizzata. Gli attributi che obbligatoriamente devono essere specificati per poter autorizzare la creazione di un nuovo utente sono l'id (che corrisponde allo username) ed una password.

[2]http://www.iso.org/iso/catalogue_detail?csnumber=24591
[3]http://www.iso.org/iso/catalogue_detail.htm?csnumber=22109

3.3 Connessione al database

Come si è già avuto modo di sottolineare l'uso di *PHP* permette un naturale ed immediato interfacciamento con i database gestiti con *PostgreSQL*. All'interno del sito lo script `pg_connect.php` si occupa di stabilire la connessione al database e di rendere possibile a tutte le altre pagine comunicare con lo stesso.

Tramite l'istruzione:

```
$db = pg_connect('dbname='.$dbname.' user='.' '.$user.
' password='.' '.$password);
```

lo script stabilisce una connessione col database *$dbname* e si autentica con le credenziali *$user* e *$password*. Una volta che la connessione è stata stabilita viene inizializzata la variabile *$db* che mantiene un riferimento per poter eseguire le azioni volute sul database.

In qualsiasi altra pagina del sito sarà sufficiente inserire la riga:

```
require('pg_connect.php');
```

per avere a disposizione la variabile *$db*.

3.4 Autenticazione al sito

Prima di poter procedere con l'annotazione manuale i partecipanti a LogCLEF 2011 devono effettuare l'autenticazione al sito accedendo alla pagina `login.php`.

Lo scopo di far autenticare un utente ad un sito è prima di tutto la possibilità di negare l'accesso a chi non è autorizzato, ma ancora più importante è il poter sapere esattamente quale utente stia compiendo una determinata azione sul sito web. Nel caso in cui, per esempio, ci siano più utenti attivi contemporaneamente allora si deve essere in grado di associare ogni sequenza di azioni ad un determinato utente. Così come è architettato il protocollo *HTTP* che regola il mondo web non è in grado di memorizzare quali azioni siano state compiute nel passato da un determinato utente, si dice infatti che l'*HTTP* è un protocollo *stateless*.

Per ovviare a questo problema ci sono due strade: utilizzare i *cookie* o utilizzare il sistema di *sessioni* introdotto da *PHP*. I *cookie* sono dei file che vengono salvati sul computer dell'utente e che salvano certe informazioni; sono inviati dal server al client per mandare informazioni su un certo stato

e poi rimandati indietro dal client al server ogni volta che il client accede allo stesso server. Le *sessioni* invece non richiedono nessuna scrittura di file sul disco degli utenti ma agiscono esclusivamente da lato server. Al contrario dei *cookie* che rimangono sul disco dell'utente per sempre (a meno di cancellazioni) i file creati sul server dalle *sessioni PHP* vengono cancellati non appena l'utente effettui il logout dal sito o chiuda il browser. Negli standard web entrambi i metodi sono comunemente accettati ed utilizzati, ma nel nostro caso si è arbitrariamente scelto di utilizzare le *sessioni PHP*.

Il modo corretto di inizializzare una *sessione PHP* è quello di eseguire il comando *session_start()* . In questo modo viene inizializzata una variabile superglobale chiamata *$_SESSION* che è costituita da un array dove si possono memorizzare alcune informazioni utili. Nel nostro caso si memorizzerà all'interno di questo array lo username dell'utente connesso; questa variabile così inizializzata sarà disponibile ad ogni pagina per tutta la durata della sessione.

Quando l'utente accede alla pagina `login.php` dovrà immettere in un *form* il suo username e la password con le quali è stato inserito nella tabella ***actor***. Una volta cliccato il pulsante submit vengono quindi inizializzate le variabili *$_POST['username']* e *$_POST['pass']* e si viene reindirizzati nuovamente alla stessa pagina. A questo punto però essendo inizializzate le due variabili, grazie ad un costrutto `IF` viene eseguita una parte di codice diversa rispetto a prima. Lo script *PHP* procede a cercare nella tabella ***actor*** se esiste una tupla con lo *username* e la *password* uguali a quelli inseriti dall'utente. Nel caso in cui non ci sia un match all'utente è chiesto di inserire nuovamente username e password; nel caso ci sia un match viene fatta iniziare una nuova sessione tramite il comando *session_start()* e viene inizializzata una variabile super globale *$_SESSION['username']* e l'utente viene reindirizzato automaticamente ad un'altra pagina in modo che possa iniziare il suo compito.

Tutte le pagine che compongono il sito iniziano con le seguenti righe:

```
if (!isset($_SESSION['username'])){
    die("Restricted area");
}
```

Queste istruzioni servono a verificare che sia stata inizializzata la variabile *$_SESSION['username']*: in caso positivo la navigazione può continu-

are, in caso negativo significa che l'utente non ha effettuato correttamente l'accesso e all'utente è negato poter vedere il contenuto di quella pagina.

3.5 Scelta della query da sottoporre all'utente

Subito dopo l'autenticazione l'utente viene reindirizzato alla pagina `interact.php`: sarà in questo momento che in maniera automatizzata verrà scelta la query su cui fare le annotazioni. La query rimarrà sempre la stessa anche durante i passaggi per le pagine successive, sarà possibile proseguire con una nuova solamente dopo essere passati per la pagina `finalStep.php` (oppure facendo il refresh della pagina `interact.php`).

In `interact.php` viene per prima cosa scelto casualmente uno dei 12 mesi di log disponibili e vengono definite due variabili superglobali corrispondenti ai nomi delle tabelle ***actionlog*** e ***sessionlog*** corrispondenti in modo che siano utilizzabili da tutte la pagine che ne abbiano bisogno.

Successivamente si sceglie a caso un *id* tra tutti quelli a disposizione nella tabella ***sessionlog*** considerata; in particolare si sceglie l'*id* corrispondente alla prima azione eseguita durante una sessione di navigazione sul sito The European Library. Questo viene fatto tramite le istruzioni:

```
$querySession = pg_query($db,"SELECT DISTINCT on (counter)
id FROM $sessionTable ORDER BY counter, date;");
$randomNumber = rand(1,pg_num_rows($querySession));
$result = pg_fetch_all($querySession);
$id = $result[$randomNumber]['id'];
```

in questo modo si è sicuri di avere a disposizione tutti i *counter* esistenti in quella tabella (tramite l'istruzione di `SELECT DISTINCT`) ordinati per data. Con le istruzioni successive si sceglie un numero a caso tra 1 e il numero di *counter* diversi presenti. A questo punto dopo aver estratto i risultati della prima query eseguita e averli inseriti in un array chiamato *$result* si può scegliere un *id* a caso tra tutti quelli possibili; avendo ordinato i risultati della prima query per *date* si è sicuri che l'*id* preso sarà il primo (per ordine cronologico) nella sessione identificata da quel particolare *counter*.

Un primo controllo sulla validità della query così scelta viene effettuato verificando che l'utente che la dovrà valutare non abbia già effettuato annotazioni su di essa in passato. Tale verifica viene fatta controllando che nella tabella ***decision*** (che serve a memorizzare le scelte fatte dalla pagi-

na `interact.php`) non esista già un record con gli stessi *id* e *userid*. Nel caso in cui tale utente abbia già valutato questa query allora, con le stesse istruzioni di prima, viene scelto un nuovo *id* fino a quando non si trovi una query adatta.

A questo punto però è necessario verificare altre specifiche riguardanti la query considerata: dato che uno dei task da adempiere riguarda la valutazione dell'intera sessione di navigazione in termini di successo di una ricerca allora non sono da considerarsi utili al fine dell'annotazione query corrispondenti a sessioni di navigazione composte da una sola azione, infatti tali sessioni non permetterebbero di eseguire un'analisi sul successo di utilizzo della query; in conseguenza di ciò si è scelto di non prendere in considerazione le sessioni che contengono un numero totale di azioni corrispondenti a ricerche semplici o avanzate uguale a uno. Per essere valida, quindi, una sessione dovrà contenere due o più ricerche (siano esse semplici o avanzate): in questo modo siamo sicuri di sottoporre ai partecipanti a LogCLEF 2011 sessioni di navigazione con un certo numero di azioni compiute che probabilmente presentino anche modifiche sulla query di ricerca, e che non si limitino ad una semplice consultazione dei risultati della prima, e unica, ricerca.

Questa verifica viene fatta analizzando i risultati della seguente query:

```
SELECT action, count(*) FROM $actionTable WHERE id IN
(SELECT id FROM $sessionTable WHERE counter =
(SELECT counter FROM $sessionTable WHERE id = $id))
AND (action = 'search_sim' OR action = 'search_adv')
GROUP BY action;
```

Questa istruzione è da interpretare:

- Utilizzo l'*id* selezionato in precedenza memorizzato nella variabile *$id*;
- Dalla tabella ***sessionlog*** estraggo il *counter* relativo a quel record con l'istruzione `'SELECT counter FROM $sessionTable WHERE id = $id'`;
- Seleziono tutti gli *id* che compaiono nella sessione identificata dal *counter* appena ottenuto con il costrutto `'WHERE id IN([...])'`;
- Ottenuti tutti questi *id* controllo l'attributo *action* di ciascun record e conto quante azioni siano uguali a *search_sim* o *search_adv*.

	action character varying(30)	count bigint
1	search_adv	2
2	search_sim	1

Figura 3.3: Esempio di possibile risultato di un'interrogazione *SQL* di tipo *count()*

Un possibile output di questa istruzione è mostrato in figura 3.3.

Questa interrogazione *SQL* viene eseguita ciclicamente fino a quando (ogni volta viene scelto un id diverso) non viene trovato un record appartenente ad una sessione con un numero totale di *search_sim* e *search_adv* strettamente maggiore a uno.[4]

Dopo questi controlli preliminari è finalmente possibile mostrare la query scelta all'utente in modo che possa valutarla nel merito delle necessità dell'edizione 2011 di LogCLEF.

3.6 Prima annotazione: lingua della query

In `interact.php` si adempie al primo task proposto: si richiede di riconoscere, per la query proposta, la lingua in cui essa sia stata formulata.

Dell'intero record che si può estrarre dalla tabella ***actionlog*** una volta scelto l'*id* all'utente viene mostrato semplicemente l'attributo *query*: sarà quindi una stringa di caratteri in letteratura chiamata "web search query" dato che proviene da una casella di ricerca. In alcuni casi l'utente può trovarsi di fronte ad una query strutturata, cioè una query che contiene al suo interno operatori booleani (come `AND, OR`) che servono a specificare meglio i risultati che si vogliono ottenere dal motore di ricerca.

Per poter effettuare la propria scelta il partecipante deve interagire con un *form HTML* di tipo *select*. Quando in un *form* si introduce il *tag* <*select*> significa che si vuole introdurre la possibilità di scelta tra più possibilità con un menù a tendina. Le varie possibilità tra cui gli utenti possono scegliere sono specificate dal *tag* <*option*>.

In particolare in `interact.php` gli utenti possono scegliere tra tutte le lingue ufficiali della comunità europea, più tutte le lingue non ufficiali della

[4]Dato che viene scelto un nuovo record a questo punto è necessario controllare nuovamente che l'utente non abbia già effettuato annotazioni su quella query. Il controllo viene rifatto nello stesso modo spiegato prima.

UE ma ufficiali nei 48 paesi che fanno parte della CENL e che quindi hanno le biblioteche nazionali catalogate in The European library. Le possibili lingue tra cui si può scegliere sono: albanian, armenian, azerbaijani, bosnian, bulgarian, croatian, czech, danish, dutch, english, estonian, finnish, french, german, georgian, greek, hungarian, icelandic, irish, italian, latin, latvian, lithuanian, luxembourgish, macedonian, maltese, norwegian, polish, portuguese, romanian, russian, serbian, slovak, slovene, spanish, swedish, turkish, ukrainian. Oltre a queste lingue si è optato per dare la possibilità di scegliere anche tra altre due opzioni: unknown e undecided nel caso in cui la lingua sia sconosciuta oppure non sia possibile decretare quale sia la lingua di appartenenza della query.

Il *form* si occupa di registrare la scelta fatta dall'utente e di trasmetterla alla pagina specificata nell'attributo *action*; in questo caso la scelta viene inviata alla pagina `interact2.php`: si è scelto infatti che le annotazioni fatte vengano effettivamente memorizzate nella tabella competente nell'istante in cui l'utente accede alla pagina successiva rispetto a quella in cui ha effettuato la scelta. Questo significa che la scelta fatta in `interact.php` verrà memorizzata da una porzione di codice scritta nella pagina `interact2.php` e così a seguire per tutte le successive annotazioni.

Dato che l'inserimento di un record nella tabella ***decision*** viene fatto in `interact2.php` c'è la necessità di avere a disposizione diverse informazioni in questa seconda pagina che invece sono disponibili in `interact.php`. Per poter scambiare dati tra una pagina e l'altra sfruttiamo il *form* appena utilizzato per la scelta della lingua. Infatti questo *form* inizializza un array chiamato *$_POST* in cui possiamo memorizzare i dati che ci servono; questo array sarà disponibile solamente nella pagina `interact2.php` (dato che questa è la pagina specificata nell'attributo *action* del *form*).

Per come sono strutturati i *form* la lingua viene nativamente inviata alla pagina di destinazione dato che rappresenta la scelta fatta dall'utente utilizzando il *form* stesso. Per gli altri dati invece si è deciso di inviarli sottoforma di attributi nascosti: verranno inviati alla seconda pagina automaticamente senza che l'utente se ne accorga. I dati inviati in maniera invisibile sono il nome utente di chi ha appena effettuato la scelta e l'*id* del record della query appena valutata.

3.6.1 Tabella decision

La scelta fatta in `interact.php` dai partecipanti viene memorizzata nella tabella ***decision***. Le informazioni che servono essere memorizzate nella tabella riguardano prima di tutto la lingua scelta per la query sottoposta e l'utente che l'ha fatta, oltre che, ovviamente, un identificativo per individuare immediatamente il record valutato. In secondo luogo siamo anche interessati a sapere quando è stata fatta l'annotazione.

La definizione della tabella ***decision*** è:

```
CREATE TABLE decision
(
  id bigint NOT NULL,
  lang isolanguagetwo NOT NULL,
  userid character varying(32) NOT NULL,
  date timestamp with time zone,
  CONSTRAINT decision_pkey PRIMARY KEY (id, userid)
);
```

La chiave primaria della tabella è costituita dalla coppia *id* e *userid* in modo da essere sicuri che un utente non esegua annotazioni sulla stessa query più volte.

All'inizio dello script `interact2.php` vengono recuperate le informazioni necessarie andando ad estrarre i seguenti valori dall'array *$_POST*:

```
$username = $_POST['user'];
$id = $_POST['id'];
$lang = $_POST['lang'];
```

Questi dati vengono poi inseriti nella tabella ***decision*** tramite l'istruzione:

```
INSERT INTO decision values ($id, '$lang', '$username',
current_timestamp);
```

In caso di buon fine dell'operazione verrà visualizzato un avviso di corretto svolgimento, in caso negativo sarà una notifica di allarme a segnalare il problema.

3.7 Seconda annotazione sulla lingua della query

In `interact2.php` viene offerta agli utenti la possibilità di effettuare una seconda scelta sulla lingua della query che hanno valutato in precedenza. Per fare questa seconda annotazione viene però fornito un dettaglio in più

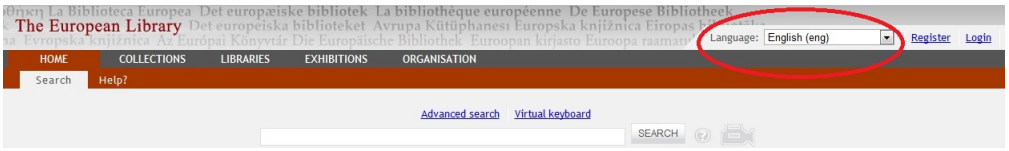


Figura 3.4: Opzione per cambiare la lingua dell'interfaccia del sito di The European Library

che possa aiutare nella decisione. Tra tutti i possibili dati disponibili nella tabella ***actionlog*** si è optato per mostrarne solo uno in più rispetto alla sola query: viene mostrata la lingua in cui è stata impostata l'interfaccia del sito The European Library.

Purtroppo non sempre questa ulteriore informazione può essere d'aiuto: non è comune che chi utilizzi The European Library cambi la lingua in cui esso è visualizzato, spesso si sceglie di visualizzare il sito in lingua inglese (che è la lingua di default) e raramente gli utenti la cambiano. Ciò nonostante si è scelto che visualizzare tale indizio possa essere di valido aiuto ai partecipanti LogCLEF.

Far vedere qual è la lingua in cui si utilizza il sito può essere di grande aiuto quando la query di ricerca è relativa ad un ambito specifico, per esempio quello botanico, in cui la maggior parte dei nomi sono in latino e non vengono tradotti nelle varie lingue.

La scelta tra le diverse lingue viene fatta utilizzando un *form HTML* simile a quello in `interact.php`. Una particolarità di questo *form* è quello di mostrare, come opzione di default, la scelta che l'utente ha effettuato in `interact.php`: essendo infatti molto probabile che la lingua venga confermata si è deciso di implementare questa funzione per rendere più rapidi i tempi di scelta della lingua.

3.7.1 Tabella extendedDecision

L'annotazione eseguita viene salvata nella tabella ***extendedDecision*** e il compito di eseguire l'istruzione *SQL* apposita viene svolto in `interact3.php`: infatti dopo che l'utente ha cliccato sul tasto di submit nel *form* la scelta fatta e le informazioni necessarie a proseguire vengono inviate alla pagina `interact3.php`.

La tabella ***extendedDecision*** è creata con la seguente definizione:

```
CREATE TABLE extendedDecision
```

```
(
  id bigint NOT NULL,
  lang isolanguagetwo NOT NULL,
  userid character varying(32) NOT NULL,
  date timestamp with time zone,
  CONSTRAINT extendedDecision_pkey PRIMARY KEY (id, userid)
);
```

Esattamente come per la tabella ***decision*** la chiave primaria è costituita dalla coppia di attributi *id* e *userid*.

3.8 Annotazione sul successo di una ricerca

Con le pagine `interact.php` e `interact2.php` i partecipanti a LogCLEF 2011 adempiono correttamente al primo task loro proposto. Gli obiettivi sono però tre e con la pagina `interact3.php` agli utenti è richiesto di fare uno studio sulle query loro sottoposte diverso da quello precedente.

In questa pagina, infatti, è richiesto di studiare il successo di una data query: è richiesto quindi implicitamente che venga studiato il buon esito di un'intera ricerca. Quello che viene sottoposto all'attenzione dei partecipanti non è solamente ciò che è stato digitato da qualcun altro nella casella di ricerca di The European Library, ma tutta la serie di eventi che da questa ricerca sono generati: vengono consultati i risultati, alcuni in maniera approfondita e altri in maniera superficiale, oppure i termini digitati vengono modificati e qui lo studio può essere portato avanti sul come vengono essi cambiati. Magari i risultati della ricerca sono stati così soddisfacenti che chi l'ha eseguita ha deciso di condividerne i contenuti via email con qualcun altro, oppure si è informato su dove siano reperibili i materiali cartacei. Oltre a questo l'attenzione può essere spostata anche sull'osservare da dove siano originate la maggior parte delle ricerche: se dalla home page del sito o se da caselle su siti esterni.

Di tutte queste diverse azioni è ovviamente tenuta traccia nei log che vengono forniti ai partecipanti a LogCLEF: nella tabella ***actionlog*** ogni record ha specificato il campo *action* che serve ad associare ad ogni dato il tipo di azione a cui fa riferimento.

I possibili valori per il campo *action* sono:

- *search_sim*: ricerca iniziale da una semplice casella di ricerca;

- *search_adv*: ricerca iniziale da una casella di ricerca di tipo avanzato;
- *search_res*: ricerca iniziale da una casella di ricerca in una pagina di risultati;
- *search_res_rec_any,search_res_rec_all*: ricerca iniziale eseguita da una visualizzazione completa di una ricerca precedente (eseguita cliccando su un'icona a lente di ingrandimento);
- *search_url*: ricerca iniziale da un indirizzo URL (una ricerca può provenire da siti esterni);
- *view_brief*: visualizza il titolo corto;
- *view_full*: visualizza il titolo lungo (siamo nel caso in cui un utente clicca su un link col titolo in una pagina con pochi (20 per pagina) o molti risultati);
- *jump_to_page*: quando un utente visualizza i risultati di una ricerca può saltare ad una determinata pagina di record;
- *available_at*: L'utente clicca su un link del tipo "Disponibile presso:";
- *see_online*: L'utente ha cliccato su un link del tipo "Mostra online";
- *page_brief*: L'utente ha cliccato sui bottoni "Prossimo" o "Precedente" nelle pagine di visualizzazione dei record;
- *col_set_theme*: Collezioni selezionate dalla lista categorie;
- *col_set_theme_country*: Collezioni scelte dalla lista delle nazioni dalla pagina dei risultati;
- *col_set_country*: Collezioni scelte dalla lista che contiene tutte le collezioni (ordinate per nazionalità);
- *col_set_subj*: Collezioni scelte dalla lista dei soggetti;
- *col_set_desc*: Collezioni scelte tramite una ricerca tra le descrizioni;
- *col_set_defaultCollections*: Reinizializzazione della lista di default;
- *option_save_session_favorite*: Salvataggio della sessione come preferita;
- *option_send_mail*: Record inviato via email;

- *options_save_reference*: Record salvato per scopi di riferimento;
- *service_[denmark]—[hungary]—[netherlands]—[uk]—[all]*: Utilizzo del link di servizio per i record;
- *show_help_helpfilename*: Richiesta di aiuto.

Agli utilizzatori del sito viene richiesto di focalizzare la propria attenzione su un solo aspetto di tutto ciò: il query refinement. Viene chiesto di analizzare le modifiche (se presenti) alla query utilizzata durante una sessione di navigazione su The European Library.

A chi accede a `interact3.php` viene mostrata un'intera sessione utilizzando i dati, e le informazioni, precedentemente organizzati a dovere nella tabella ***sessionlog***, è a questo punto che torna indispensabile il campo *counter* precedentemente introdotto proprio per il corretto svolgimento di questo task.

Dalle pagine precedenti in `interact3.php` tutto quello che si conosce è l'*id* della query su cui si stanno eseguendo le annotazioni; quello che si vuole far vedere per poter procedere è l'intera sessione di navigazione associata al *counter* della query, in particolare per ogni record che appartiene alla sessione vengono mostrati: la query (che può man mano variare), il tipo di azione svolto e il timestamp di quando è stata compiuta. Mentre il *counter* è disponibile nella tabella ***sessionlog*** le altre informazioni sono disponibili nella tabella ***actionlog***. Per poter estrapolare alcuni dati da una tabella e altri dati dalla seconda si utilizza una query che utilizzi l'operatore `JOIN`. Il `JOIN` è un particolare operatore che serve a correlare tra di loro i dati contenuti in tabelle diverse. In particolare verrà utilizzato l'`INNER JOIN` che combina i valori delle due tabelle di partenza basandosi su una certa regola di confronto, in questo caso l'uguaglianza dell'attributo *id* dei record nelle due tabelle.

La query utilizzata per recuperare tutte le informazioni necessarie è:

```
SELECT a.query,a.action,s.date
FROM $sessionTable AS s INNER JOIN $actionTable AS a
ON s.id = a.id
WHERE counter = '$counter' ORDER BY s.date;
```

ovvero una volta recuperato il *counter* e salvato nella variabile *$counter* (con un'operazione precedente non mostrata) si prendono dalla tabella *$actionTable* i campi *query* e *action*, e dalla tabella *$sessionTable* il campo *date*

This is the complete session:

Query	Action	Date
("charles sherlock")	search_sim	2010-03-29 10:51:02.3
("charles sherlock")	search_res	2010-03-29 10:54:56.774
("charles sherlock")	view_full	2010-03-29 10:55:26.104
("charles sherlock")	search_adv	2010-03-29 10:56:38.152
("charles sherlock")	search_adv	2010-03-29 10:57:03.494
("charles sherlock")	view_brief	2010-03-29 10:57:44.689
("charles sherlock")	view_brief	2010-03-29 10:57:47.799
("charles sherlock")	view_brief	2010-03-29 10:57:50.079
("charles sherlock")	view_brief	2010-03-29 10:58:20.347
("charles sherlock")	col_set_country	2010-03-29 11:03:27.977
("charles sherlock")	search_sim	2010-03-29 11:03:46.528
("charles sherlock")	view_brief	2010-03-29 11:04:02.51
("charles sherlock")	search_res	2010-03-29 11:04:16.617

Figura 3.5: Esempio di sessione di navigazione così come viene presentata ai partecipanti a LogCLEF 2011

di tutti i record che hanno il *$counter* considerato; i risultati estratti dalle due tabelle vengono uniti basandosi sull'uguaglianza del campo *id*. Successivamente l'insieme di azioni così recuperate viene mostrata all'utente con un'organizzazione tabulare come mostrato in figura 3.5

Con un *form HTML* l'utente può fare la sua annotazione; le possibilità tra cui esso può scegliere sono:

- Same query: la query utilizzata nella query non è stata cambiata durante la sessione;
- Generalization: la query è stata modificata in modo da ottenere risultati più generici rispetto a quelli ottenuti inizialmente;
- Specification: la query è stata modificata in modo da ottenere risultati più specifici rispetto a quelli ottenuti inizialmente;
- Drifting: la query è stata totalmente modificata in un'altra con lo stesso grado di specificità ma inerente ad aspetti diversi dello stesso argomento;
- Not applicable: nel caso in cui la scelta non sia effettuabile;
- More than two differents queries: può succedere che la query venga modificata più di una volta, in questo caso non si può procedere con questa valutazione.

Una volta effettuata la scelta l'utente viene reindirizzato alla pagina `interact4.php` che si occuperà, prima di permettere all'utente di effettuare l'ultima annotazione, di inserire nell'apposita tabella la scelta appena fatta sulla sessione.

3.8.1 Tabella sessionDecision

La scelta compiuta in `interact3.php` viene inserita nella tabella denominata ***sessionDecision*** da un'istruzione *SQL* eseguita nella pagina successiva, cioè `interact4.php`.

Quello che in questo caso si vuole che sia salvato sono le informazioni a riguardo della sessione valutata (tramite il campo *counter*) in modo da aver accesso a tutte le query interessate e, ovviamente, la scelta fatta dall'utente memorizzata nel campo *classification*.

La tabella è quindi così creata:

```
CREATE TABLE sessionDecision
(
  id bigint NOT NULL,
  counter character varying(32) NOT NULL,
  userid character varying(32) NOT NULL,
  classification character varying(32) NOT NULL,
  date timestamp with time zone,
  CONSTRAINT sessionDecision_pkey
  PRIMARY KEY (id, userid, classification)
);
```

A differenza delle altre tabelle in questo caso la chiave primaria oltre alla coppia *id* e *userid* contiene anche il campo *classification*: in futuro, infatti, si potrebbe decidere di permettere all'utente di valutare più volte la stessa sessione nel caso in cui la query venga modificata più di una volta, mentre adesso c'è solo la possibilità di segnalare tale evento scegliendo l'opzione *"More than two differents queries"*. Mettendo questa tripletta come chiave primaria l'utente potrà effettuare più di una annotazione, avendo cura di scegliere ogni volta un'opzione diversa: risulteranno così inseriti nella tabella più record con gli stessi *id* e *userid* ma con l'attributo *classification* diverso, in modo da non violare il vincolo di unicità della chiave primaria nella tabella.

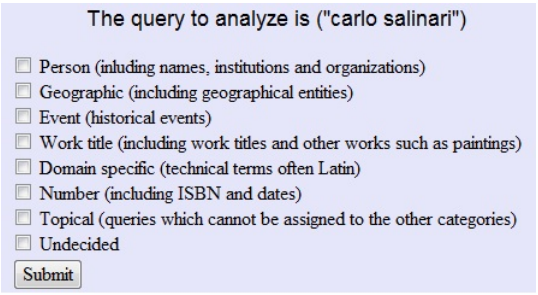


Figura 3.6: Categorie messe a disposizione per la scelta in `interact4.php`

3.9 Classificazione di una query

Tramite la scelta che si effettua nella pagina `interact4.php` l'utente adempie all'ultimo task rimasto: data una query deve essere in grado di associarla ad una (o più) categorie di interesse.

Le possibili opzioni tra le quali si può scegliere sono:

- *Person*: nomi propri di persone, ma include anche nomi, organizzazioni e istituzioni;
- *Geographic*: luoghi ed entità geografiche;
- *Event*: eventi di tipo storico;
- *Work title*: titoli di opere (anche dipinti);
- *Domani specific*: termini tecnici (spesso in latino);
- *Number*: numeri di qualsiasi tipo, anche codici ISBN o date;
- *Topical*: query che non possono essere assegnate alle altre categorie;
- *Undecided*: se non si è sicuri della categoria d'interesse.

Dato che una query può essere strutturata e può rientrare in più categorie di interesse (si pensi ad esempio ad una ricerca autore e titolo di una sua opera) allora nel *form* che gestisce questa annotazione si dà la possibilità di scegliere più opzioni tra quelle proposte. Il *form* è quindi di tipo *checkbox* e permette di scegliere una o più opzioni e quelle scelte vengono salvate all'interno di un array che verrà inviato alla pagina specificata nell'attributo *action*; in questo caso le annotazioni vengono inviate sotto forma di array chiamato *category[ ]* alla pagina `finalStep.php` .

3.9.1 Tabella categoryDecision

Appena l'utente viene indirizzato alla pagina `finalStep.php` cliccando sul tasto submit nel *form* della pagina `interact4.php` le scelte lì effettuate vengono inserite nella tabella ***categoryDecision***.

La tabella è così definita:

```
CREATE TABLE categoryDecision
(
  id bigint NOT NULL,
  category character varying(32) NOT NULL,
  userid character varying(32) NOT NULL,
  date timestamp with time zone,
  CONSTRAINT categoryDecision_pkey
  PRIMARY KEY (id,category, userid)
);
```

Dato che è possibile fare più scelte allora la chiave primaria non è costituita solo dalla coppia *id* e *userid* ma anche dall'attributo *category* in modo tale che ogni utente possa, per ogni query, scegliere più *category* diverse.

3.10 Fine delle annotazioni

La pagina `finalStep.php` dopo essersi occupata di inserire nella tabella ***categoryDecision*** l'ultima annotazione fatta dai partecipanti presenta all'utente la possibilità di scelta: a questo punto infatti si può scegliere di continuare con le annotazioni oppure di uscire dal sito.

Nel primo caso l'utente verrà reindirizzato alla pagina `interact.php` dove una nuova query verrà scelta casualmente; cliccando sul tasto di *submit* verrà inviato, sotto forma di attributo nascosto, lo *username* dell'utente, che è tutto quello di cui ha bisogno la pagina `interact.php` per funzionare correttamente. Nel secondo caso l'utente viene indirizzato alla pagina `logout.php` che si occupa di distruggere la sessione corrente tramite l'istruzione *session_destroy()* e poi rimandare l'utente alla pagina di *login* (essendo distrutta la sessione qualunque altra pagina sarebbe inaccessibile).

3.11 Ottimizzazioni grafiche del sito

Fino a quando il sito non è stato completato e veniva usato solo in locale per eseguire delle prove l'aspetto estetico non era importante. Quando si

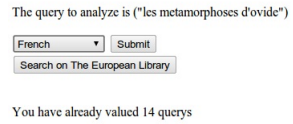
scrive una pagina *HTML* lo stile è molto semplice: lo sfondo bianco, il font non particolarmente ricercato e tutti gli elementi della pagina posizionati sulla sinistra. Quando però arriva il momento di mettere online il sito e renderlo accessibile a tutti i partecipanti a LogCLEF 2011 le pagine hanno bisogno di una rivisitazione grafica che renda la navigazione più gradevole.

Una pagina web è generalmente formata da due parti: i contenuti veri e propri e la formattazione, cioè l'aspetto con cui i contenuti sono mostrati all'utente. L'*HTML* gestisce una pagina web grazie all'utilizzo di diversi *tag* e i browser che si usano per accedere ai vari siti interpretano il codice *HTML* mostrando all'utente le formattazioni predefinite per ogni tipo di *tag* che incontrano. Quando nella storia del web hanno iniziato a comparire *tag* del tipo <*font*> e attributi per definire i colori del testo la programmazione web si è fatta molto complicata. L'implementazione di grandi siti web dove le informazioni a riguardo dei font e dei colori erano aggiunte ad ogni singola pagina comportava un lavoro lungo e dispendioso. Per risolvere questo problema il W3C (world wide web consortium)[5] ha creato il *CSS* (cascading style sheets). [7] Il *CSS* è un linguaggio che permette di separare il contenuto dalla formattazione, infatti alla base della sua creazione c'è la volontà di mantenere il contenuto delle pagine web sempre definito dal codice *HTML*, mentre la formattazione viene trasferita su un codice completamente separato, il *CSS* appunto. Il *CSS* definisce come le pagine *HTML* debbano essere visualizzate e solitamente lo stile è salvato in un file (.css) chiamato foglio di stile esterno a quelli *HTML*, in modo che sia possibile cambiare il look di tutte le pagine in un sito web modificando un singolo file.

Un foglio di stile è composto da una serie di regole e ognuna di queste definisce il modo in cui vogliamo che un certo elemento sia visualizzato. Una regola generalmente è composta da due parti: un selezionatore e una o più dichiarazioni. Il selezionatore si riferisce all'elemento che vogliamo definire, mentre ogni dichiarazione consiste di una proprietà e un valore. La seguente regola:

```
#warning{
        font-weight:bold;
        color:red;
        font-size: 16pt;
}
```

[5] http://www.w3.org/



(a) Pagina web senza nessuno stile applicato

(b) Pagina web con stile CSS applicato

Figura 3.7: Stile pagina web con e senza *CSS*

indica che l'elemento identificato dall'*id warning* (è il testo che verrà visualizzato quando un'operazione di inserimento di un'annotazione in una tabella non è andata a buon fine) dovrà essere visualizzato in grassetto, di colore rosso e con un font alto 16pt.

In ogni pagina web creata all'interno del *tag* <*head*> c'è un'istruzione che permette di importare da un file .css (chiamato `style.css`) gli stili impostati per la visualizzazione:

```
<head>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

All'interno del sito tutti gli elementi visualizzabili sono identificati tramite la specificazione dell'attributo *id*, per esempio il form che gestisce il login inizia con la seguente definizione:

```
<form id = "login" name="input" action="login.php" method="post">
```

e quindi verrà identificato dall'*id* = *"login"*. Agendo così per ogni elemento si otterrà il risultato mostrato in 3.7 messo a confronto con lo stile *HTML* di default.

In tutte le pagine sono presenti nell'intestazione due immagini che agiscono come collegamenti ipertestuali: se si clicca sul logo dell'università di Padova si verrà reindirizzati (in una nuova scheda) al sito di UNIPD[6], cliccando sul logo di *PROMISE* si verrà reindirizzati al sito corretto[7].

Nella pagina `interact.php`, inoltre, è presente un pulsante che permette di eseguire una ricerca su The European Library con la stessa query che si sta valutando in quel momento, in più in fondo alla pagina sarà possibile vedere un contatore per sapere quante altre query siano state analizzate fino ad ora da quell'utente.

[6] http://www.unipd.it/
[7] http://www.promise-noe.eu/

Capitolo 4

Conclusioni

L'analisi e lo studio di log di siti internet rientra nel più ampio campo dell'*information retrieval* che è quella disciplina che si occupa della ricerca e del recupero mirato dell'informazione e di cui i motori di ricerca rappresentano la più conosciuta applicazione. Un processo di *information retrieval* comincia quando un utente inserisce una query in un sistema di ricerca e le query utilizzate dagli utenti rappresentano formalmente la necessità di ricerca di un certo tipo di informazione. [8]. L'analisi e l'ottimizzazione dei motori di ricerca e del loro funzionamento sono oggigiorno campi su cui sempre più ricercatori stanno ponendo la loro attenzione.

Tra tutte le campagne di studio nell'ambito dell'*information retrieval* logCLEF è una iniziativa di ricerca a livello internazionale per l'analisi, la valutazione e la classificazione di query allo scopo di interpretare e comprendere i comportamenti di navigazione e ricerca in ambienti multilingua e, in ultima fase, di migliorare gli stessi servizi di ricerca.

L'edizione 2011 di logCLEF non è tanto mirata allo studio dell'efficienza di tali motori, quanto allo studio dell'interazione che hanno gli utenti con essi e all'analisi delle diverse query utilizzate durante la ricerca. Lo scopo di questa tesi è stato quello di progettare e realizzare una interfaccia disponibile via web che permettesse ai partecipanti di eseguire delle annotazioni in modo manuale su delle query che vengono loro sottoposte.

Il sito così come è impostato verrà utilizzato per l'edizione 2011 di logCLEF, ma per come è stato progettato e implementato apportandovi piccole modifiche verrà utilizzato anche per edizioni e progetti futuri: i log di The European Library sono stati infatti correttamente archiviati nel database *PostgreSQL* e il lavoro fatto per dividere i record nelle diverse sessioni di

navigazione può sicuramente essere utile per altri scopi. Nelle varie tabelle oltre agli attributi visti e utilizzati ne sono disponibili anche altri e può essere quindi possibile concentrare le analisi su questi.

Tramite l'interfaccia web risultato di questa tesi i partecipanti a logCLEF 2011 posso eseguire delle annotazioni manuali su delle query loro sottoposte in modo da formare un insieme di dati classificati che costituiscano una collezione da prendere come modello di riferimento in quanto a correttezza in riferimento alle scelte fatte. Viene richiesto che, data una query, venga correttamente identificata la lingua in cui essa sia stata scritta, che venga classificata secondo particolari categorie di interesse e che venga valutato il successo della ricerca associata. Le query vengono prese in modo casuale tra tutte quelle disponibili nei log del sito The European Library che cataloga i materiali di 48 biblioteche nazionali d'Europa.

Il sito web si occupa di mostrare agli utenti una query e di permettere poi loro di eseguire manualmente determinate scelte in relazione alle annotazioni che si richiede essi facciano per soddisfare gli obiettivi dell'edizione 2011 di logCLEF. Le annotazioni vengono poi inserite in un database apposito e verranno poi utilizzate nei modi e nei tempi previsti dal progetto. Oltre alla parte di progettazione dell'interfaccia è stato fatto un fondamentale lavoro di organizzazione dei dati messi a disposizione e provenienti dai log di The European Library. La parte cruciale è stata la suddivisione dei log in sessioni di navigazione distinte, in modo che per ogni query sia possibile estrarre le informazioni necessarie per conoscere quali siano state le azioni compiute da un utente sul sito dopo aver eseguito la ricerca. Questa suddivisione in sessioni è stata fatta per il corretto svolgimento del task relativo allo studio sul successo di una ricerca, e ciò che viene analizzato è come una determinata query di partenza possa essere modificata per meglio adattare i risultati della ricerca alle proprie necessità.

Durante logCLEF 2011 sono state effettuate annotazioni manuali su 1290 query diverse. I gruppi di ricerca coinvolti con i task sono stati quattro (su 17 inizialmente registrati), per un totale di 24 utenti che hanno avuto l'autorizzazione all'accesso al sito web e lo hanno utilizzato. I gruppi che hanno partecipato sono *DAEDALUS* (Spagna), *UBER-UVA* (Germania, Olanda), *CUZA* (Romania) e *ESSEX* (Regno Unito).

- *DAEDALUS* [9]: questo gruppo di ricerca ha analizzato se ci sia qualche effetto misurabile sul successo di una ricerca nel caso in cui la lingua

in cui essa sia stata effettuata e la lingua scelta per l'interfaccia del sito differiscano. Le analisi dei dati mostrano che, in generale per ogni lingua, il fatto che la lingua in cui viene effettuata la ricerca coincida, o meno, con la lingua dell'interfaccia non ha apparentemente nessun impatto sul successo della ricerca;

- *UBER-UVA* [10]: hanno investigato i comportamenti in ambito multilingua degli utenti in termini di osservazione delle diverse lingue utilizzate (per la ricerca, per la visualizzazione dell'interfaccia ecc.). Una parte del loro lavoro ha anche interessato lo studio del successo di una ricerca confrontato con la lingua dell'interfaccia e alla nazionalità dell'utente;

- *CUZA* [11]: hanno presentato uno studio sull'applicabilità del task di identificazione della lingua nel caso in cui il testo da analizzare sia molto corto, come nel caso delle query di ricerca;

- *ESSEX* [12]: questo gruppo è l'unico della presente edizione che ha utilizzato anche i dati provenienti dai log di un altro sito, cioè German EduServer (Deutscher Bildungsserver (DBS))[1]. Hanno infatti così presentato un confronto tra le due collezioni di dati (la seconda è quella proveniente da The European Library) confrontando la quantità di diverse query presenti e il numero delle sessioni di navigazione. Inoltre questo gruppo di ricerca si è focalizzato sullo studio di sistemi automatici di suggerimento delle query.

Ritornando al sito web realizzato si può affermare che una sicura possibile miglioria può essere relativa al modo in cui sono gestite le nuove iscrizioni: potrà essere creata una pagina che permetta ad un utente di registrarsi automaticamente al sito e di avere così accesso all'interfaccia di annotazione.

Dal punto di vista dell'utilizzo dell'interfaccia si potrà, tramite l'uso di *javascript*, creare caselle di testo autocompletanti che possano sostituire gli attuali menù a tendina. Sempre con l'uso di *javascript* possono essere apportate altre modifiche sempre nell'ambito del look and feel del sito.

[1] http://www.eduserver.de/

Bibliografia

[1] *Top 500 sites on Alexa.com*. 2011. URL: http://www.alexa.com/topsites.

[2] Giorgio Maria Di Nunzio, Johannes Leveling e Thomas Mandl. "Multilingual log analysis: LogCLEF". In: *Proceedings of the 33rd European conference on Advances in information retrieval*. ECIR'11. Dublin, Ireland: Springer-Verlag, 2011, pp. 675–678. ISBN: 978-3-642-20160-8.

[3] *Topic and goal of LogCLEF*. 2011. URL: http://ims.dei.unipd.it/websites/LogCLEF/Topic_and_Goal.html.

[4] *Tasks of LogCLEF*. 2011. URL: http://ims.dei.unipd.it/websites/LogCLEF/Tasks.html.

[5] *Different categories of queries*. 2010. URL: http://ir.cis.udel.edu/sessions/guidelines10.html#queries.

[6] *Database basati sul modello relazionale*. 2011. URL: http://en.wikipedia.org/wiki/Relational_model.

[7] *CSS (Cascading Style Sheets)*. 2011. URL: http://www.w3schools.com/css/css_intro.asp.

[8] *Information retrieval*. 2011. URL: http://en.wikipedia.org/wiki/Information_retrieval.

[9] Sara Lana-Serrano, Julio Villena-Román e José Carlos González-Cristóbal. "DAEDALUS at LogCLEF 2011: Analyzing Query Success and User Context". In: *This volume*. 2011.

[10] Maria Gäde et al. "Interface Language, User Language and Success Rates in The European Library". In: *This volume.* 2011.

[11] Alexandru-Lucian Gînscă, Emanuela Boroş e Adrian Iftene. “Adapting Statistical Language Identification Methods for Short Queries”. In: *This volume.* 2011.

[12] M-Dyaa Albakour e Udo Kruschwitz. “University of Essex at LogCLEF 2011: Studying Query Refinement”. In: *This volume.* 2011.