UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

Relazione

Un Framework collaborativo per il matching di dataset geografici

Candidato: *Nicholas Montenegro*

Relatori: *Andrea Marchetti*
Angelica Lo Duca
Davide Gazzè

Correlatore: *Alessandro Lenci*

Anno Accademico 2014-2015

ABSTRACT

Oggigiorno, grazie alla diffusione delle più moderne tecnologie web, sempre più organizzazioni sia pubbliche che private mettono a disposizione dataset geografici riguardanti luoghi fisici dislocati in varie parti del mondo. Questi dati hanno un enorme valore sia per studi politico/economico che turistici. Purtroppo, la maggior parte dei dataset non offrono un'elevata qualità. Il modo più semplice di ovviare a questa problematica è quella di fondere diversi dataset. In questa accezione il problema del match del medesimo luogo su dataset geografici diversi riveste un'importanza fondamentale. Ovviamente questa problematica è ben lungi dall'essere totalmente automatizzata, infatti esistono diverse applicazioni classiche che permettono l'annotazione manuale.

In questa tesi verrà introdotto il Geo Data Annotator (GDA), un framework collaborativo web-oriented per l'annotazione di dataset geografici. GDA fa della semplicità d'uso, dell'utilizzo delle tecnologie web uno dei suoi punti di forza. Inoltre, GDA essendo collaborativo permette a diversi utenti di annotare gli stessi dataset. Al fine di semplificare il lavoro dell'annotatore, GDA mette a disposizione due tipi di indici, uno geografico e uno basato sulla similarità tra stringhe.

Come caso di studio, nella tesi verrà descritta l'annotazione di 3 dataset geografici riguardanti la accommodation nel comune di Pisa. Il primo dataset, che funge da riferimento, deriva degli opendata messi a disposizione della regione Toscana, mentre il secondo e il terzo sono dati raccolti dai social media Google Places e Facebook.

La tesi discute i risultati della sperimentazione nell'annotazione del dataset di riferimento con Google Places e Facebook da parte di 3 diversi annotatori. Dagli esperimenti eseguiti si può dedurre che l'approccio utilizzato semplifica la complessità del task.

Inoltre GDA è stato valutato positivamente dalla comunità scientifica ed approvato presso il WWW-2015.

INDICE

INTRODUZIONE

Negli ultimi anni abbiamo potuto assistere ad un continuo crescere della quantità di dati ed informazioni geografiche reperibili su internet, sia open-data sia proprietarie. Esempi di sorgenti proprietarie sono:

- social media
- agenzie governative
- centri di ricerca
- aziende private

Il diffondersi delle nuove tecnologie web ha modificato i ruoli degli utenti sulla rete. L'utente generico è passato dalla posizione di consumatore a quella di produttore di dati. Dietro all'esplosione di reperibilità di informazioni geografiche (ma non solo) si nasconde il problema della qualità e quindi dell'affidabilità dei dati.

Questa situazione rappresenta un problema che pone seri limiti allo sfruttamento di questa ricca fonte di dati. Infatti una vasta gamma di settori (scienze sociali assistenza sanitaria, turismo, finanza, economia ecc.) fa regolarmente uso di questa tipologia di dati. Per risolvere questo problema sono stati sviluppati importanti strumenti come il *data matching* che permette l'identificazione di records nei vari datasets che definiscono la stessa entità geografica.

Nel corso del tempo sono stati proposti molti algoritmi di data matching, deduplica ed integrazione dei dati, ognuno di questi basato su procedure automatiche o semi-automatiche per la comparazione di due o più records. Tali tecniche sono state proposte al fine di conciliare le informazioni di un'entità in un singolo punto (luogo).

Con il termine deduplica in informatica si intende il processo mediante il quale si esegue una compressione dei dati atta a diminuire lo spazio per l'archiviazione. Lo stesso concetto applicato all'elaborazione dei dataset (il nostro caso) definisce la fase attraverso la quale vengono cancellati i record doppi, che rappresentano cioè la stessa entità.

L'integrazione invece è il processo nel quale, durante l'elaborazione dei dati, sono racchiuse le operazioni per adattare questi ultimi ad essere inseriti in un database. L'integrazione è una fase che può variare considerevolmente a seconda delle proprie necessità; alcuni esempi di queste operazioni sono la normalizzazione, l'eliminazione

di alcuni contenuti o l'estrazione di questi per poter essere ad esempio suddivisi ed archiviati in categorie.

Tuttavia, anche se questi algoritmi sono automatizzati, è ancora richiesto durante la fase di apprendimento del sistema un processo di annotazione manuale da parte degli esperti di dominio.

Da questo presupposto nasce il progetto GeoData Annotator, un'applicazione web-based che si pone l'obiettivo di ridurre la complessità del processo annotativo facilitando il lavoro umano ed al tempo stesso incrementare la qualità dei dataset geografici.

STATO DELL'ARTE

Il crescente interesse per l'elaborazione di dati provenienti da dataset geografici ha portato alla creazione di molte applicazioni. Di seguito viene riportata una panoramica dei principali strumenti disponibili descrivendone le caratteristiche, il funzionamento e le differenze con GeoData Annotator.

GeoDDupe

GeoDDupe è un'applicazione sviluppata presso il Dipartimento di Informatica dell'Università del Maryland[1] scritto con il linguaggio di programmazione C#. Quest'applicazione fa uso di algoritmi automatici per l'identificazione di record duplicati ed offre al tempo stesso un'interfaccia basata su reti relazionali per l'identificazione di quest'ultimi da parte dell'utente. GeoDDupe è uno dei primi progetti nati per lavorare esclusivamente su dati geografici. La rete relazionale è composta dai record, i nodi della rete, mentre gli archi sono rappresentati dalla relazione che lega questi record: in questo caso la somiglianza. Gli algoritmi di confronto analizzano non soltanto la vicinanza geografica, ma anche le informazioni che accompagnano il record.

[1] *http://linqs.umiacs.umd.edu/projects//geoddupe/*

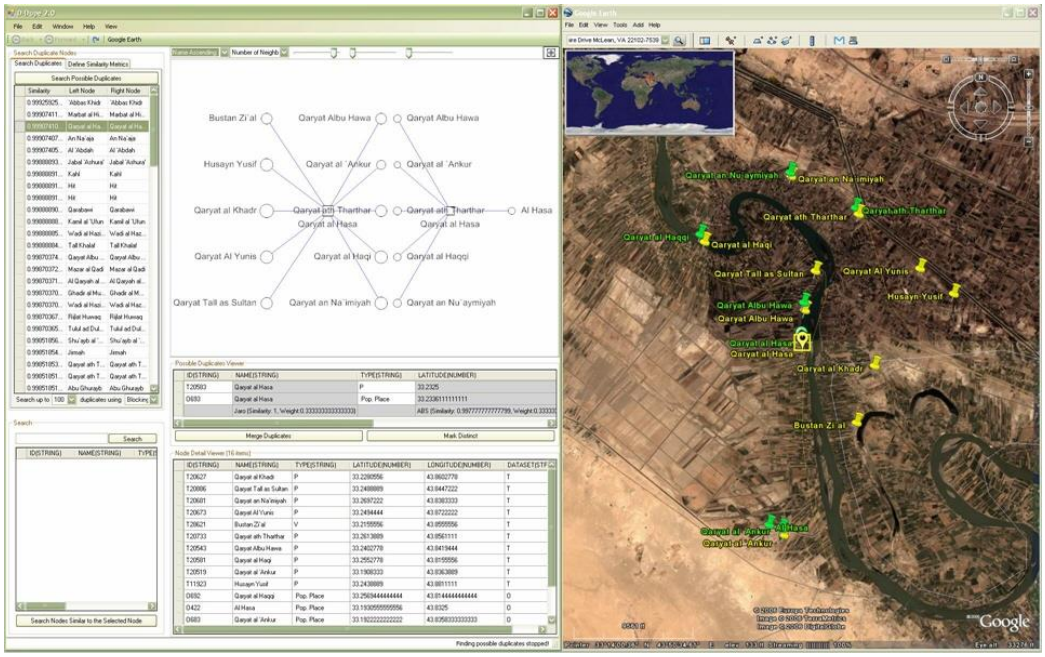

Figura 1: GeoDDupe: l'interfaccia geografica insieme con la rappresentazione relazionale dei record.

L'interfaccia di GeoDDupe si compone di tre finestre principali:

- La finestra dei potenziali duplicati: mostra una lista di record identificati come possibili duplicati dagli algoritmi di similarità ed ordinati secondo il grado di somiglianza in una scala che va da 0 (nessuna similarità) ad 1 (del tutto uguali). Gli utenti possono selezionare una coppia di possibili duplicati che verranno mostrati nella finestra relazionale.
- La finestra del contesto relazionale: mostra la relazione di vicinanza fra i due record potenzialmente duplicati, essa si divide in 5 sezioni che mostrano i due nodi duplicati, i nodi vicini (neighborhood nodes) che condividono, e quelli che non condividono, calcolati sulla base di un'area prestabilita (in km di distanza rispetto ai due nodi).
- La finestra delle informazioni sui record: mostra tutti gli attributi ed i relativi valori che fanno parte dei record come ad esempio il nome, l'indirizzo, la latitudine, la longitudine, ecc. Tali attributi possono essere selezionati (tutti o solo parte di essi) per calcolarne la similarità.

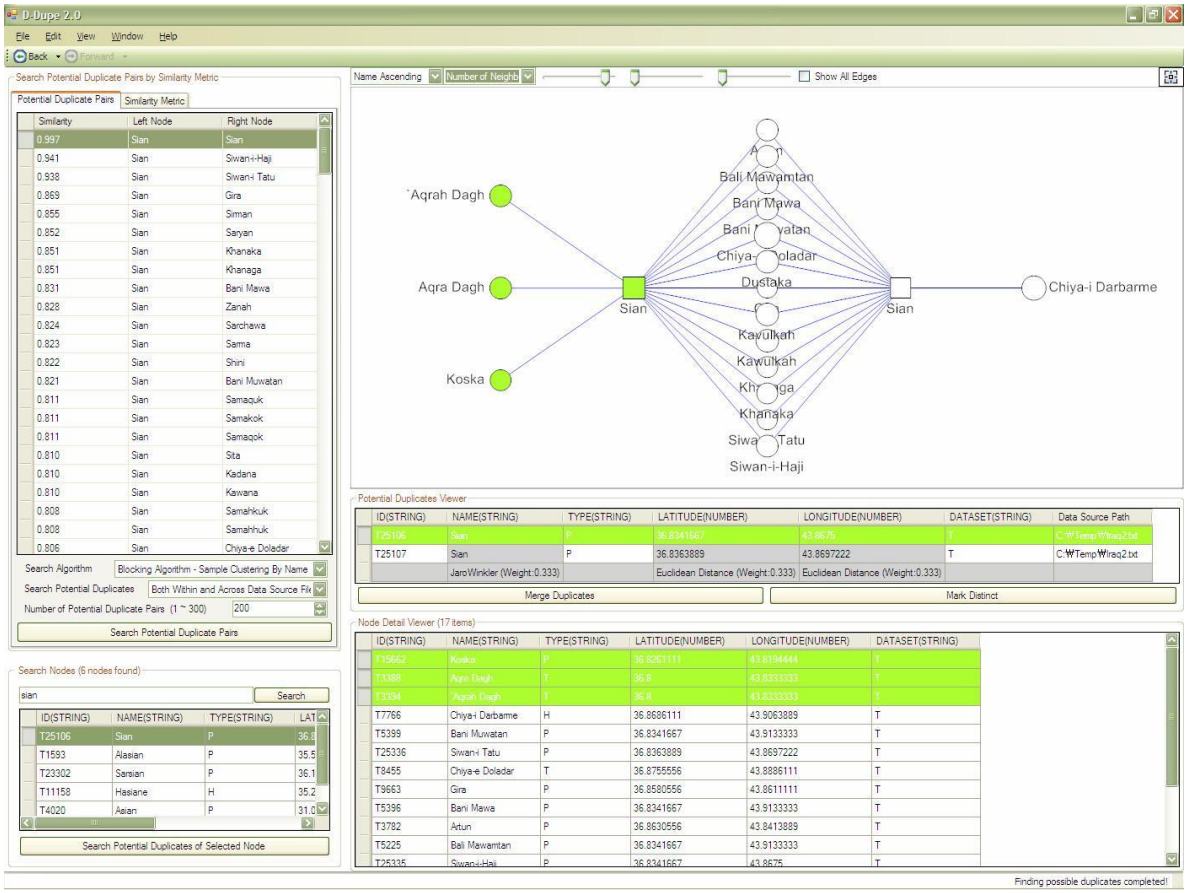

Figura 2: Interfaccia principale di GeoDDupe, si possono notare sulla parte sinistra la finestra dei potenziali duplicati, sulla parte destra superiore la finestra del contesto relazionale e nella parte destra inferiore quella delle informazioni sui record.

Davanti ad una coppia di duplicati, l'utente ha a disposizione tre possibili scelte: segnarli come merge, segnarli come due marker distinti o ignorare temporaneamente il matching per poterlo rivalutare successivamente. GeoDDupe definisce varie tipologie di indici di misura per gli attributi: gli algoritmi di Levensthein, Jaccard, Jaro, JaroWinkler, MongeElkan, etc. (ciascuno con i suoi vantaggi ed in grado di produrre risultati più o meno diversi per potersi adattare alle necessità dell'utente). Mentre per il calcolo della distanza geografica (latitudine e longitudine) viene messo a disposizione il calcolo della distanza euclidea.

La rappresentazione del grafo relazionale può essere visualizzata secondo lo schema dei nodi condivisi e non condivisi, oppure in una rappresentazione 2D basata sulla vicinanza-lontananza dei nodi rispetto a quelli duplicati; il tutto può essere anche visualizzato in una mappa satellitare messa a disposizione dal progetto Google Earth.

Da un punto di vista architetturale GeoDDUpe è stato sviluppato in C# appoggiandosi al toolkit open source *Piccolo*, sviluppato internamente all'Università del Maryland. L'applicazione è concepita per ricevere in input dataset in formato di testo, MS Access oppure un database attraverso l'API ODBC (Open DataBase Connectivity). GeoDDupe è strutturato in tre parti: Model (gestione dei processi per l'elaborazione dei dati), View e Controller. Il Model è ulteriormente suddiviso in più moduli: algoritmi per data-mining, per la struttura dei grafi e per la gestione dell'input/output; questa scelta è dovuta all'intenzione di rendere GeoDDupe aperto all'aggiunta di algoritmi sviluppati dagli utenti per differenti calcoli sulla similarità o per l'input e l'output di differenti formati come l'XML.

DuDe

DuDe è l'acronimo di Duplicate Detection, è un toolkit sviluppato all'Università di Potsdam in Germania[2]; nasce per essere facile da usare e da estendere, al fine di supportare una grande varietà di formati di dati e per poter implementare ulteriori algoritmi oltre a quelli già presenti di default.
Il processo di elaborazione dei dati di DuDe si compone di sei sezioni:

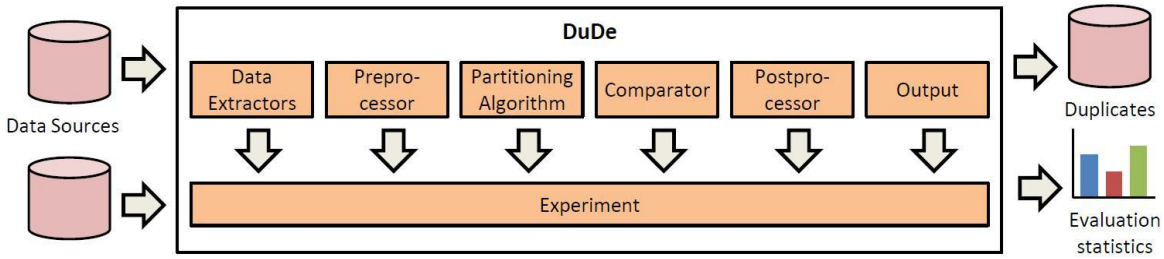

Figura 3: Schema di elaborazione dei dati su DuDe.

- Estrazione dei Dati: i dati ricevuti in input possono essere in vari formati (MySQL, PostgreSQL, DB2, CSV file, XML, ecc.), in questa fase i dati vengono convertiti in formato JSON, viene definito un identificatore per ogni record (consistente in uno o più attributi) e viene aggiunto un ID che identifica la sorgente dei record permettendo così il confronto fra diversi dataset.
- Pre-processore: il preprocessore è usato per ricavare statistiche durante il processo di estrazione come ad esempio il numero di records o di attributi. Queste informazioni vengono poi messe a disposizione per le fasi successive di elaborazione.
- Algoritmo di partizionamento: si occupa della fase di identificazione dei duplicati. L'algoritmo di partizionamento crea le coppie di record da far analizzare al comparatore scegliendo i record più vicini fra loro basandosi su algoritmi come il *Sorted Neighborhood Method* o il *Blocking Method*.

[2] *http://hpi.de/naumann/projects/data-quality-and-cleansing/dude-duplicate-detection.html*

- Comparatore: il comparatore confronta i due record di ogni singola coppia per calcolarne la similarità, che assume un valore da 0 a 1 (dove 1 significa uguaglianza).

  Vengono distinte tre differenti comparazioni:

  - Comparazione basata sulla struttura: si basa sull'analisi della struttura del singolo record, ad esempio il calcolo sulla similarità della struttura degli attributi.
  - Comparazione basata sui contenuti: analizza i valori degli attributi, per questa analisi DuDe mette a disposizione (di default) 19 differenti comparatori come ad esempio la distanza di Levensthein, Jaro-Winkler, ecc..
  - Multi-comparatori: similarità combinata calcolata sul risultato di più comparatori.

  I comparatori servono tuttavia solo per il calcolo della similarità fra coppie di dati e non identificano duplicati o differenti records. In questo caso viene confrontata la similarità con una soglia (threshold), a seconda se sia maggiore o minore di quest'ultima se ne decreta l'esito.
- Post-processore: riceve le coppie di dati e genera statistiche riguardo il processo, come ad esempio il tempo d'esecuzione, il numero di coppie di record generate e quello dei duplicati riconosciuti. Se esiste un gold standard viene eseguito un confronto e ne viene calcolata la precision, recall, f-measure, etc..
- Output: Sono previsti vari formati di output, dal CSV (con o senza informazioni addizionali) al JSON fino al formato di testo semplice.

Per lo sviluppo del toolkit DuDe è stato utilizzato il linguaggio Java. Tale linguaggio, oltre a rendere il sistema maggiormente aperto all'aggiunta di funzionalità, permette la compatibilità cross-platform e l'esecuzione su tutti i sistemi operativi che supportano Java.

FEBRL

FEBRL (**F**reely **E**xtensible **B**iomedical **R**ecord **L**inkage) è un sistema sviluppato dall'Università di Canberra in collaborazione con il New South Wales Department of Health di Sydney[3]. L'obiettivo del progetto è lo sviluppo di nuove tecniche per la pulitura, standardizzazione e deduplicazione dei dataset in ambito sanitario.

Il sistema è sviluppato in linguaggio Python, che è una piattaforma perfetta per lo sviluppo di dati strutturati in liste e dizionari, inoltre la grande quantità di moduli per l'estensione esistenti permette l'accesso a numerose tipologie di database ed alla costruzione di una GUI avanzata.

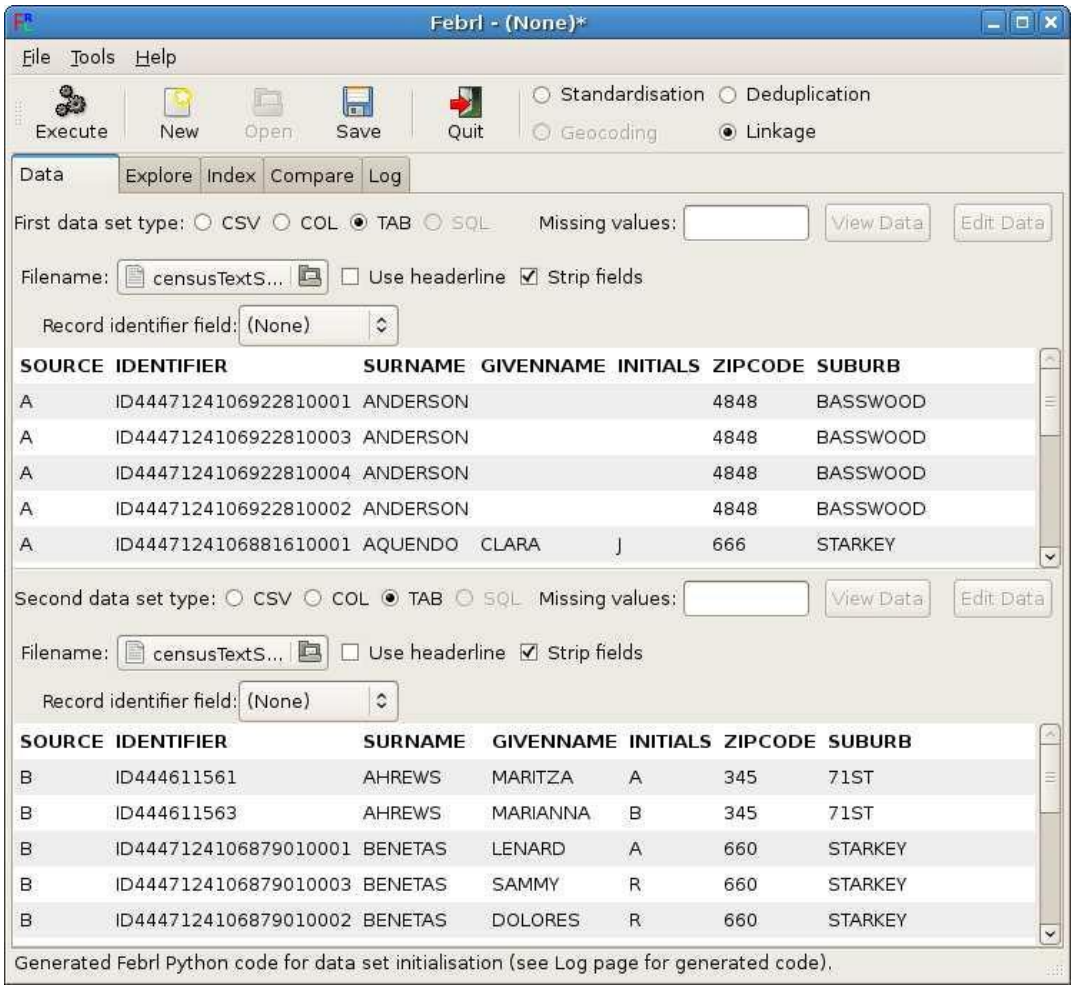

Figura 4: FEBRL: interfaccia per l'unione di due dataset.

[3] *http://datamining.anu.edu.au/software/febrl/febrldoc/manual.html*

FEBRL è sviluppato sotto licenza open source, sia per l'accessibilità del codice sorgente, sia per l'intenzione di rendere il progetto rapido da sviluppare ed orientato all'estensione e l'implementazione di nuovi algoritmi e funzionalità, oltre che rendere il progetto aperto a nuovi utenti che intendono sperimentare nuove tecniche di elaborazione ed unione tra i dataset. Al momento del rilascio, FEBRL era l'unico sistema per la pulitura, standardizzazione e deduplicazione dei dataset basato su un'interfaccia grafica. Il progetto nasce con l'obiettivo di rendere le sue funzionalità accessibili anche ad un'utenza non tecnica, da qui la scelta di dotarlo di una GUI.

L'interfaccia di FEBRL si compone di un'unica finestra, la quale contiene diverse tab, una per ogni funzione del programma ad esempio per la fase di input dei file, della scelta dei metodi, delle operazioni da eseguire, della verifica del risultato, ecc..

La prima fase di elaborazione di FEBRL prevede la pulitura e la normalizzazione dei dati provenienti dai vari dataset che vengono collegati, deduplicati e copiati in un nuovo unico dataset. Successivamente vengono create le coppie di record da analizzare, è possibile selezionare dei metodi di sorting ed indicizzazione come il BlockingIndex o il SortingIndex che permettono di ridurre il numero di coppie da analizzare; viene successivamente creata una primary key che può essere un nuovo campo od uno o più campi già esistenti.

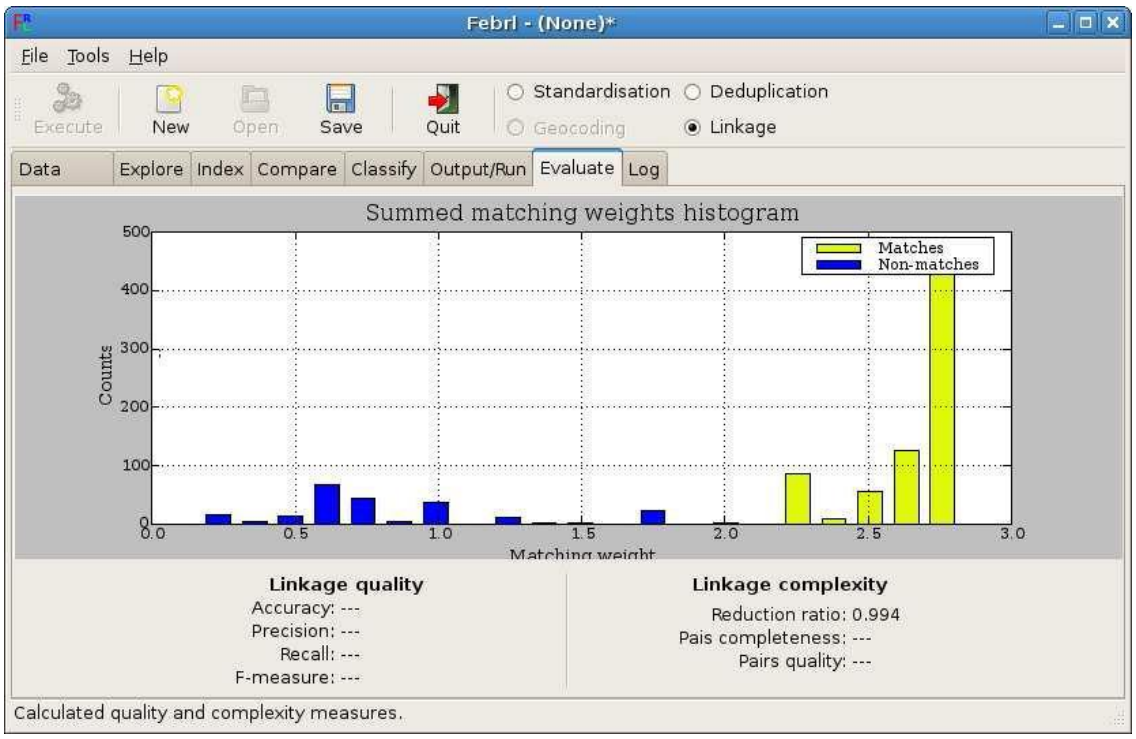

Figura 5: Interfaccia del riepilogo del matching tra i record.

La fase successiva è quella dedicata alla comparazione dei valori dei campi e FEBRL mette a disposizione di default 26 diversi algoritmi di comparazione, alcuni di questi specializzati nell'analisi di specifici dati come date, ore, valori numerici o stringhe e la similarità viene calcolata con un valore compreso tra 0 e 1. Si procede poi al processo di confronto dei record nelle coppie dove, anche qui, numerosi strumenti di classificazione (supervisionati e non) vengono messi a disposizione. Viene infine fornito il file di output in formato Python.

FRIL

FRIL (**F**ine-grained **R**ecord **I**nteraction and **L**inkage tool) è un tool di comparazione fra dataset per l'unione di record sviluppato dal dipartimento di informatica e matematica dell'università di Emory ad Atlanta in collaborazione con il National Center on Birth Defects and Development Disabilities per il monitoraggio dei difetti congeniti nell'area metropolitana di Atlanta[4].

Allo stato attuale FRIL confronta 12.700 record provenienti dal Metropolitan Atlanta Congenital Defects Program (MACDP) con 1,25 milioni di record riguardo i certificati di nascita. L'obiettivo per cui il tool è stato sviluppato è monitorare come i difetti e le malformazioni alla nascita cambino in base al periodo temporale ed in che modo l'ambiente possa incidere su questi fenomeni oltre che poter quantificare il tasso di mortalità collegato ai difetti congeniti.

All'utente viene data la scelta sull'utilizzo di determinati algoritmi per l'unione di record fra quelli che FRIL mette a disposizione, L'utente può scegliere il metodo di ricerca, settare le distanze per la misurazione della similarità ed il modello decisionale per accettare o rifiutare un match. Sono inoltre stati sviluppati strumenti di apprendimento automatici per permettere suggerimenti sul trattamento dei parametri.

L'output di FRIL consiste di tre insiemi cui vengono suddivisi i record:

- L'insieme dei record matchati
- L'insieme dei record non matchati
- L'iniseme dei record che possono essere potenzialmente matchati: questi possono successivamente essere sottoposti al processo di elaborazione con altri parametri di controllo.

[4] *http://www.ncbi.nlm.nih.gov/pubmed/18985680*

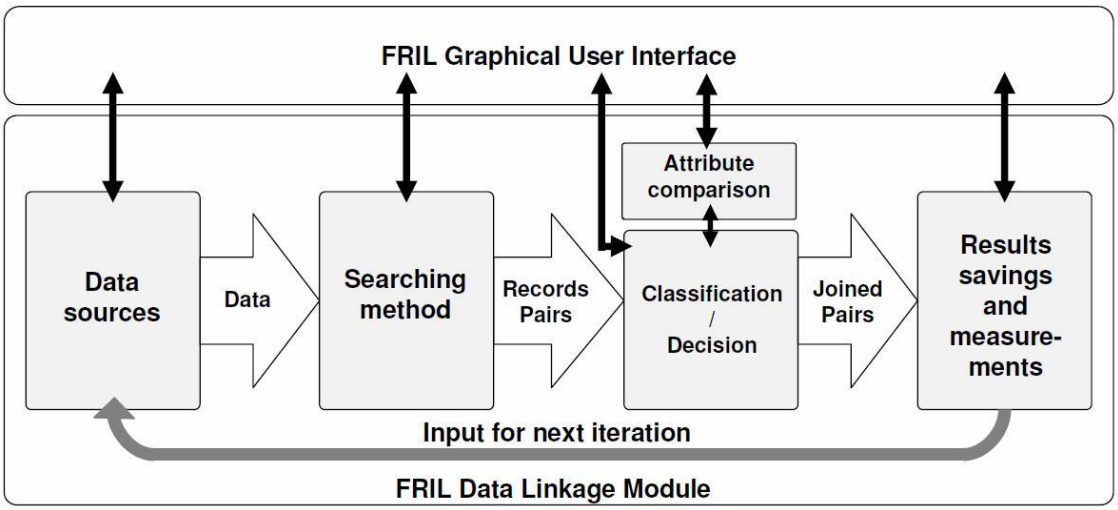

Figura 6: FRIL: architettura e flusso di lavoro.

FRIL implementa due metodi di ricerca: *Nested Loop Method* e *Sorted Neighborhood Method* con il primo che analizza la totalità dei record incrociandoli l'uno con l'altro mentre il secondo disegna una finestra di una grandezza prestabilita intorno ad un record confrontandolo solo con quelli che rientrano al suo interno.

Per quanto riguarda il confronto fra i valori degli attributi, FRIL prevede la scelta di un attributo dominante che avrà un ruolo critico rispetto agli altri nella misurazione della similarità; nel modello decisionale è possibile comunque impostare il peso per ogni attributo.

Merge ToolBox

Merge ToolBox (MTB) o Matching ToolBox è un tool sviluppato all'Università di Costanza in Germania[5] per comparare le performance sulla misurazione di similarità fra le stringhe dei cognomi tedeschi. Il linguaggio di programmazione scelto è il Java per la sua portabilità cross-platform anche se durante la fase di sviluppo e sperimentazione (durata un anno) è stato preferito Perl . Attualmente viene impiegato per la prototipazione di comparazione empirica fra stringhe e per i tool di pre-processamento dei dati, inoltre è stato utilizzato per progetti di ricerca in campo sociologico, economico ed epidemologico.

Specificamente MTB nasce per venire incontro all'esigenza degli statisti di rilevare e correggere (possibilmente in modo automatico) gli errori umani commessi in fase di inserimento dei record come lapsus di memoria, errori di spelling o tipografici. Si tratta di un problema cruciale dal momento che il matching dei dati che presentano anche lievi errori possono minare seriamente il risultato del processo portando a falsi positivi o negativi.

Matching ToolBox nasce dalla mancanza di programmi pubblicamente disponibili dedicati allo scopo (vanno infatti esclusi Matcher-2, nato come tool interno all'US Bureau of the Census, GRLS anch'esso sviluppato per lo Statistics Canada e OXLINK e AUTOMATCH, nati esclusivamente per la ricerca medica).

Al di là di questo, perfino la ricerca riguardo misurazioni di similarità sui giornali scientifici al tempo del rilascio di Merge ToolBox (2004) era sporadica e basata su errori generati artificialmente.

Il tool si compone di tre sotto-programmi che vengono eseguiti in sequenza:

- Il pre-processore: si occupa di trasformare i dati in input (vengono accettati vari formati come CSV e Xbase) in formato STATA (**sta**tistic and da**ta**, formato comunemente usato nella comunità scientifica per lo scambio di dati in quanto standardizzato) per poter essere elaborati. Un'ulteriore funzione è quella di rimuovere le stringhe non necessarie (come titoli accademici o

[5] *https://www.uni-due.de/soziologie/schnell_forschung_safelink_mtb.php*

nobiliari), standardizzare i prefissi come Mc o Mac (tramite una lista prefissata) e sostituire i caratteri speciali.

- misurazione della similarità fra record: viene offerta un'ampia varietà di scelta riguardo le misurazioni di similarità fra stringhe come ad esempio l'analisi di bigrammi e trigrammi, algoritmi di edit distance, ecc.. Vengono formate le coppie di record che possono potenzialmente comporre un match divise in sottogruppi a seconda delle variabili selezionate per l'analisi. Un file di log con le opzioni selezionate viene fornito al termine dell'esecuzione, due file di log possono essere confrontati per la comparazione fra due database sorgenti. Le due fasi iniziali del programma (pre-processing e analisi dei record) possono essere eseguite su shell di sistema o, in alternativa, su interfaccia grafica fornita dal programma, come per la terza fase.
- Modulo di modifica manuale: creata per il collegamento fra i record non riconosciuti dal programma che devono, quindi, essere trattati manualmente dall'utente. Si tratta della fase più laboriosa ed è stata intenzione degli sviluppatori cercare di renderla più user-friendly possibile. Si compone di due finestre indipendenti che mostrano i record e di un sistema di ricerca basato su pattern (algoritmo AGREP) permettendo di selezionare i record da entrambi i database e di unirli fra loro.

Confronto con GeoData Annotator

Dopo aver visionato i principali tool attualmente disponibili, si giunge infine al GeoData Annotator (GDA), oggetto di analisi di questo elaborato. GDA è un sistema di annotazione e deduplica basato sull'accuratezza (ground-truth) sviluppato dal Consiglio Nazionale delle Ricerche di Pisa. L'obiettivo che GeoData Annotator si pone è esattamente quello degli altri framework, ciò che lo differenzia e, in tal senso lo rende attualmente unico, sono le due caratteristiche sulle quali è stato sviluppato: è un sistema web-based ed è un framework collaborativo.

La collaborazione fra gli utenti è il punto di forza del sistema, il quale permette a più persone di fornire il proprio contributo e cooperazione per eseguire il matching dei record provenienti da due differenti dataset.

L'interfaccia, come già accennato, è basata sul web e questo permette di estendere il concetto di collaborazione ad una scala ben più ampia di quanto possono, nei limiti delle loro possibilità, gli altri software. Un approccio web-based consente infatti la cooperazione fra utenti situati in ogni parte del mondo, i quali possono contribuire alle annotazioni ed al matching dei record.

Nessun tool o framework precedentemente analizzato è dotato di una tale potenzialità che acquista ulteriore risalto in un'era in cui la globalizzazione, la comunicazione e l'interconnessione fra utenti è di primaria importanza.

GEO DATA ANNOTATOR

Il Geo Data Annotator è un framework interattivo che ripone la sua forza nella collaborazione[6]. Collaborazione che avviene tra vari operatori umani per poter annotare e, quindi, costruire dataset basati sull'accuratezza, unendo le informazioni provenienti da due diversi dataset geografici.

Con accuratezza si definisce la fedeltà che una serie di valori misurati ha rispetto al campione reale che rappresenta. Nel nostro caso, trattandosi di dataset geografici, i valori misurati sono le coordinate geografiche insieme con i dati (nome e indirizzo) che esprimono una reale entità fisica (come ad esempio un luogo o un edificio) e che cercano quindi di rappresentare con la massima accuratezza (fedeltà) possibile.

Il progetto, considerate le problematiche che intende affrontare e che sono già state analizzate e gli obiettivi che si prefigge, è stato elogiato ed approvato all'International World Wide Web Conference (WWW2015) tenutosi a Firenze nel 2015.

METODOLOGIA

Gli scopi primari che un'applicazione per il supporto al processo di elaborazione di dataset dovrebbe avere sono la riduzione della complessità dello stesso processo di elaborazione ed il perfezionamento della qualità generale del dataset annotato.

Come già introdotto in precedenza, la complessità dei dataset geografici è un serio problema che può tuttavia essere mitigato dall'utilizzo di un sistema ben progettato. La qualità del processo di annotazione, invece, è direttamente collegata al numero di errori commessi dagli annotatori (errori umani).

Si vanno adesso ad analizzare i due aspetti fondamentali che determinano la qualità di un'applicazione per l'elaborazione dei dataset.

[6] *http://www.iit.cnr.it/node/33767*

Riduzione della complessità:

La complessità derivante dalla corrispondenza dei match è il più grande problema da affrontare quando si annota un dataset geografico, questo deriva direttamente dalla bassa scalabilità dello stesso processo di annotazione. Infatti il matching dei dati, da un punto di vista tecnico, è composto dai collegamenti fra i record piuttosto che fra i record stessi, questo perché ogni record di un dataset deve essere confrontato con ogni record dell'altro dataset.

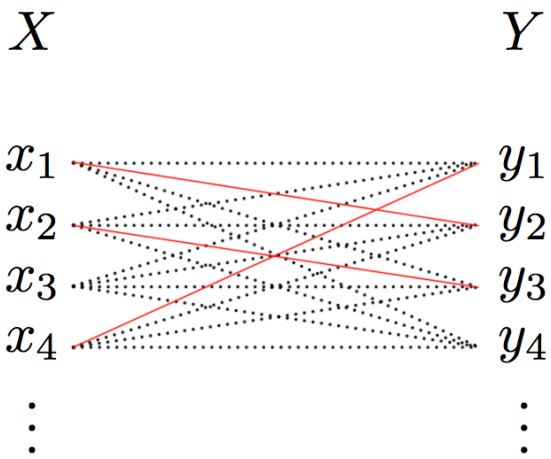

Figura 7: Tutti i possibili confronti fra due dataset (le linee rosse rappresentano i matching).

Questo risulta in un numero massiccio di potenziali collegamenti che, senza l'ausilio di algoritmi, dovrebbero essere annotati manualmente dall'utente. Se supponiamo che X e Y sono due dataset geografici, il numero totale di collegamenti da annotare è così calcolato:

Il valore di corrisponderà quindi al totale spazio di comparazione e sarà dato dal prodotto cartesiano dei due dataset X e Y. Come si può vedere in figura 7, ogni linea rappresenta un'annotazione fra i record in X con quelli in Y, le linee rosse rappresentano gli effettivi matching. Solitamente gli elementi di matching sono solo una piccola porzione rispetto a tutti i collegamenti fra i record dei dataset.

Valutazione della Qualità:

Un secondo aspetto cruciale in un sistema di annotazione risiede nell'identificazione degli errori commessi durante le annotazioni, problema strettamente correlato alla valutazione ed al miglioramento del processo di annotazione. Il GeoData Annotator, grazie alla sua natura basata sulla collaborazione, può venire incontro al problema sfruttando il concetto della concordanza degli annotatori (*concordance* o *annotator agreement*).

Tipicamente, il compito del matching dei dati fa si che più annotatori umani eseguano tale operazione su i vari record di un progetto. Gli stessi dati sono quindi annotati da tutti gli utenti che eseguono il compito singolarmente, senza cioè sapere come gli altri stiano eseguendo la stessa operazione per poter evitare così di essere condizionati dalle scelte altrui.

Nel corso del tempo sono state elaborate alcune metriche per il calcolo e l'interpretazione dei risultati di questo tipo di annotazione, tra le più comuni troviamo la *Kappa di Cohen* o la *Pi di Scott*. Tuttavia, mentre queste misurazioni possono essere applicate alla perfezione per misurare risultati di massimo due annotatori, si rivelano insoddisfacenti quando questi sono in numero maggiore. In questi casi, in cui rientra pienamente il GeoData Annotator, è consigliato adottare la Kappa di Fleiss (vedi cap. Conclusioni e Future Works).

IL FRAMEWORK

TECNOLOGIE UTILIZZATE

Per la realizzazione di GeoData Annotator si è fatto uso delle principali tecnologie web oggi adottate.
Dalla parte client troviamo:

- HTML (HyperText Markup Language)
- JavaScript
- jQuery
- CSS3

E' stata utilizzata la libreria *CryptoJS* per la criptazione delle password secondo l'algoritmo MD5. Per la visualizzazione della mappa si è ricorso all'API Google Maps.
Si è fatto anche uso di *overlappingMarkerSpiderfier*, una libreria JavaScript scritta per le API di Google Maps che permette la clusterizzazione dei marker e della libreria *infobox* messa a disposizione da Google per la creazione delle finestre informative dei marker.

Dal lato server si è optato per l'utilizzo del linguaggio PHP alla versione 5.6.7. Infine per l'archiviazione dei dati è stata scelta la tecnologia MySQL.

ARHITETTURA

Il funzionamento di GeoData Annotator si compone di varie fasi:

- Autenticazione: ad ogni utente viene richiesta la registrazione e l'esecuzione del login per poter utilizzare il framework.
- Creazione o scelta di un progetto: è possibile scegliere di lavorare su di un progetto già esistente, in tal caso viene fornita una lista di tutti i progetti di cui l'utente è proprietario o ne sono stati comunque forniti i permessi. In alternativa è possibile creare un nuovo progetto di cui l'attuale utente loggato ne sarà ovviamente proprietario.

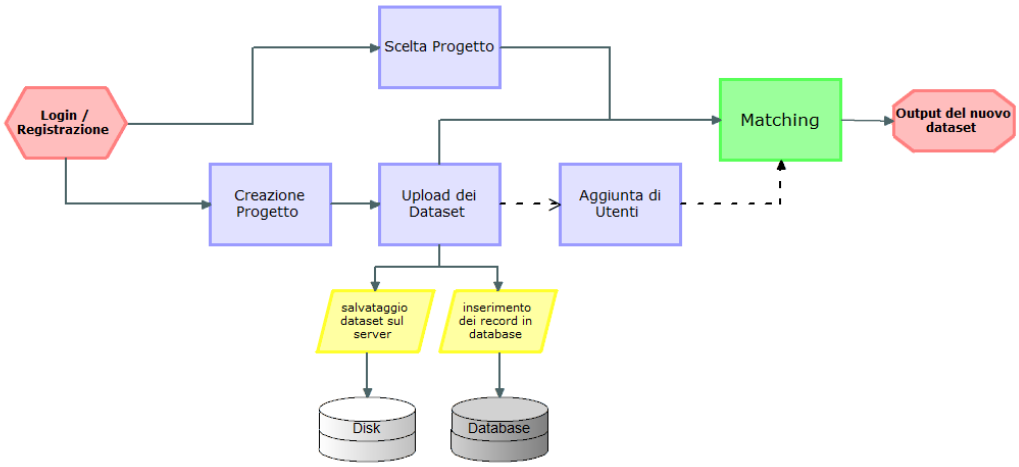

Figura 8: Diagramma di flusso sul funzionamento di GeoData Annotator.

A questo punto, a seconda della scelta effettuata al punto precedente sono possibili due alternative:

- Creare il progetto: viene anzitutto chiesto di inserire un nome valido per il progetto e opzionalmente una breve descrizione, successivamente viene chiesto di caricare i due dataset contenenti i record che saranno oggetto dell'elaborazione ed è possibile coinvolgere più persone nel progetto aggiungendo utenti.
- Matching: è la fase centrale del framework ed è quella dove gli utenti effettuano il matching dei record.

GeoData Annotator è un framework web-based che utilizza un database per ospitare i dataset geografici, i dati degli utenti, e i match effettuatuati su di un server. In tal modo, gli utenti (client) possano connettervisi e lavorare, anche in contemporanea, da remoto ovunque si trovino (vedi cap. Database e Tecnologie Utilizzate).

AUTENTICAZIONE

Il modulo di autenticazione e registrazione comprende tutti quei meccanismi che permettono il login ed il logout richiesti per l'utilizzo del framework. Questo si compone di una serie di funzioni JavaScript, chiamate JSON e script PHP. In questa sezione sono raccolte, oltre le funzioni che gestiscono la fase di accesso e registrazione, anche quelle relative all'aggiunta di utenti ad un progetto esistente.
Nella seguente tabella sono riportate le funzioni JavaScript (lato client) rapportate ai relativi script PHP lato server:

Lato Client	***Lato Server***	***Funzionamento***
Login	*login.php*	*gestisce il login degli utenti*
Logout	*logout.php*	*gestisce logout e cancellazione della sessione*
VerifyUserName	*verifyUserName.php*	*verifica se il nome scelto in fase di registrazione non è già in uso*
VerifyUserToProject	*verifyUserToProject.php*	*verifica se è possibile aggiungere un utente al progetto*
AddUser2Projet	*addUserToProject.php*	*aggiunge l'utente al progetto se possibile*
RegisterUser	*registerUser.php*	*gestisce la registrazione di un nuovo utente*

I metodi *Login* e *Logout* permettono l'autenticazione dell'utente (dopo previa registrazione) e l'accesso ai suoi progetti ed a quelli a cui è stato aggiunto. L'accesso avviene attraverso l'inserimento del proprio username e della password, la quale viene criptata tramite l'algoritmo MD5 grazie alla libreria *CryptoJS*.
Il metodo *VerifyUsername* controlla in fase di registrazione se l'username scelto dall'utente sia effettivamente libero o non sia già in utilizzo da altri iscritti, dal momento che questo deve necessariamente essere univoco. La funzione *VerifyUserToProject* controlla se l'utente selezionato che si vuole aggiungere al

proprio progetto esista effettivamente e se non sia già parte di quest'ultimo. *AddUser2Project* viene invocata nel caso risulti possibile aggiungere l'utente al progetto.

Ognuna delle 5 funzioni client-side sopra descritte invoca un script PHP su lato server, 4 di queste attraverso una chiamata AJAX ed una (*Logout*) con un redirect per la distruzione della sessione.

Tutte le funzioni legate all'autenticazione fanno uso del metodo POST per lo scambio di dati tra client e server per poter garantire la massima sicurezza possibile.

INTERFACCIA UTENTE

In questo paragrafo verranno illustrate le funzionalità implementate in questa versione del GeoData Annotator. In particolare, verranno discusse l'operazione di caricamento di un dataset geografico, il matching fra i marker e la gestione dei progetti.

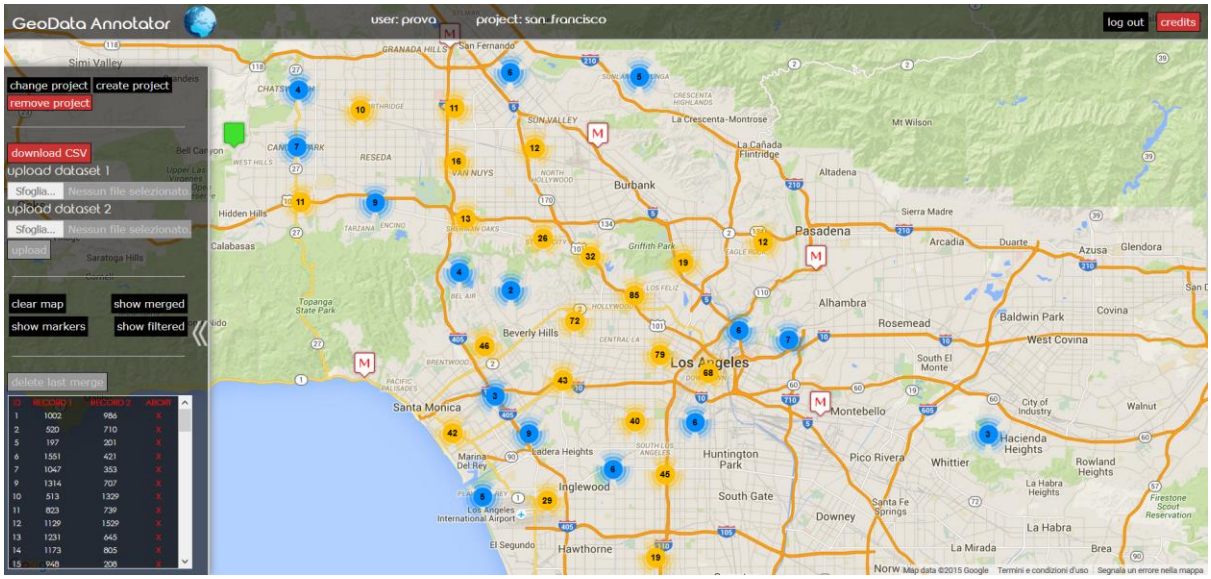

Figura 9: Panoramica dell'interfaccia di GeoData Annotator.

Il dettaglio delle funzioni implementate è mostrata nella tabella seguente:

Funzione	***Descrizione***
VerifyProject	*verifica, durante la creazione di un progetto, se il nome scelto non è già in uso*
CrtPrj	*crea un nuovo progetto*
CrtPrjList	*genera la lista dei progetti in base a quelli disponibili per l'utente*
LoadProject	*carica un progetto già esistente di cui si è proprietari o partecipanti*
RemoveProject	*elimina un progetto*

UploadDatasets	*carica i dataset di un progetto sul server*
DBInsert	*registra nel database il dataset caricato associandolo al progetto*
ExtractPlaces	*passa le informazioni del dataset allo script per l'estrazione dei dati*
InsertPlaces	*riceve i dati dei record e li consegna allo script per l'inserimento nel database*
LoadMarkers	*richiede i record al caricamento della pagina*
ExtractMerged	*estrae i record dei merge collegati al progetto ed all'utente (se sono presenti)*
Filter	*filtra i marker che devono essere disegnati sulla mappa con quelli rappresentanti i merge*
DrawMarker	*disegna sulla mappa i marker prescelti*
ShowMerged	*seleziona solo i marker che rappresentano merge*
ShowMarker	*seleziona solo i marker che rappresentano record*
ShowFiltered	*invoca Filter*
ClearMap	*svuota la mappa da tutti i marker*
SelectMarkers	*gestisce i marker selezionati per il merge ed invoca la funzione per il calcolo della similarità*
MergeMarkers	*esegue il merge una volta selezionati due marker*
AddMergedMarker	*aggiunge il nuovo marker rappresentante il merge tra quelli da disegnare*
DeleteMerge	*elimina un merge precedentemente effettuato*
DeleteLastMerge	*elimina l'ultimo merge effettuato*
CreateMergeList	*crea la lista dei merge effettuati dall'utente*

La funzione *VerifyProject* verifica che, durante la fase di creazione del progetto, il nome scelto non sia già in uso permettendo o impedendo l'esecuzione della funzione *CrtPrj* che si occupa dell'effettiva creazione ed inizializzazione. La funzione *LoadProject* carica i dati relativi al progetto scelto ed all'utente, mentre *RemoveProject* cancella definitivamente un progetto con tutti i dati collegati, compresi quindi i dataset caricati ed i merge effettuati.
Una volta effettuata la creazione di un progetto è necessario caricarne i relativi dati, per fare ciò viene eseguita una serie di processi consecutivi. *UploadDatasets* è la prima funzione ad essere lanciata ed è quella che esegue l'upload su server dei dataset (attualmente viene accettato il formato CSV), questi devono essere due e vengono caricati contemporaneamente. Per ogni file viene invocata la funzione *DBInsert* che inserisce nel database il relativo CSV caricato collegandolo al progetto. *ExtractPlaces* passa il nome e l'id del dataset allo script *processData.php* che cerca ed estrae i dati. La funzione *InsertPlaces* riceve i dati e li passa ad *insertData.php* per l'inserimento nel database. Alla fine del processo ne viene visualizzato l'esito, da questo momento è possibile operare sul progetto.

Una volta caricati i dati viene lanciato un reload della pagina per l'inserimento nella mappa dei marker, questo avviene con l'invocazione di *LoadMarkers* ed *ExtractMerged*, la prima estrae tutti i marker collegati al progetto mentre la seconda i merge effettuati dall'utente. La funzione *Filter*, come suggerisce il nome, filtra i marker con quelli del merge scegliendo quali dovranno essere disegnati sulla mappa dal metodo *DrawMarker*.
Le funzioni *ShowMerged*, *ShowMarkers* e *ShowFiltered* mostrano rispettivamente le sole annotazioni eseguite dall'utente, i soli marker originali del progetto o entrambe le possibilità filtrando i dati esattamente come avviene al caricamento del progetto. Il metodo *ClearMap* invece pulisce la mappa da qualunque marker disegnato, indipendentemente dalla tipologia; è importante far notare che tale azione non cancella i dati caricati o i merge effettuati dall'utente, ne tantomeno compie operazioni nel database ma si limita ad operare al solo livello grafico nella rappresentazione geografica dei dati.
Con la pressione del tasto A è possibile selezionare due marker dalla mappa per effettuare il merge, operazione gestita dalla funzione *SelectMarkers* la quale, una volta memorizzati , invoca il metodo *MergeMarker*. Questo recupera le informazioni

dai due marker e li invia tramite chiamata JSON a *mergeData.php* che li memorizza nel database come annotazione effettuata dall'utente. Successivamente la funzione *AddMergedMarker* crea il nuovo marker che sostituisce i due precedenti, viene invocata nuovamente *Filter* per aggiornare la mappa. Il metodo *DeleteMerge* permette all'utente di annullare l'ultimo merge effettuato nel caso si sia trattato di un errore e di ripristinare i due marker originali; la cancellazione avviene a tutti i livelli: grafico, lato client fino al database.

Durante tutte queste operazioni viene fatto uso di alcune funzioni secondarie non menzionate in precedenza, queste sono:

- *infoBox*: disegna i box contenenti le informazioni dei marker.
- *CalculateZoom*: calcola il centramento della mappa nella sola zona interessata dai marker.
- *CheckDatasets*: controlla il campo per il caricamento dei dataset disabilitandolo nel caso l'operazione sia già stata effettuata.
- *MergedAnimation*: attiva l'animazione di conferma del merge.
- *DownloadCSV*: richiede l'output del progetto con i relativi dati in formato CSV.
- *Similarity*: richiede il calcolo della similarità delle stringhe e quella geografica e ne gestisce la visualizzazione a schermo.
- *ZoomOnMarker*: effettua lo zoom sul marker di un merge una volta selezionato dalla lista.

SISTEMA DI MATCHING

GeoData Annotator basa il matching dei record secondo due differenti criteri: quello geografico e quello sulla similarità delle stringhe.

Il confronto geografico è reso possibile grazie alla rappresentazione visiva dei place su di una mappa. Su quest'ultima vengono visualizzati marker di diverso colore a del dataset di appartenenza o se risultanti dall'operazione di merging.. Passando con il cursore del mouse sopra un marker, l'interfaccia permette di visualizzare le informazioni riguardanti il place come il nomee el'indirizzo.

Per aiutare ulteriormente l'utente nella valutazione sulla somiglianza geografica il framework esegue un calcolo sulla distanza che si basa sulla formula della Great-circle Distance la quale esegue la misurazione tenendo conto anche della curvatura della Terra.

Come già accennato, GeoData Annotator mette a disposizione dell'utente, come ulteriore strumento per il matching, l'analisi della similarità tra stringhe. IN particolare è stato utilizzato l'algoritomo di Levenshtein. Durante la seleziona un marker ne vengono estratti il nome e l'indirizzo, vengono uniti andando a creare una sola stringa, lo stesso procedimento viene ripetuto per il secondo marker selezionato. Quando entrambe le stringhe sono state create viene eseguita una chiamata AJAX ad uno script lato server (*similarita.php*) che ne analizza la similarità restituendone l'esito sotto forma di risultato del calcolo effettuato. Insieme a questo viene fornito anche un ulteriore esito più immediato e di più semplice lettura: sono stati creati quattro livelli di valutazione differenti a seconda della similarità fra le due stringhe che aiutano l'utente a decidere se si tratta di un merging valido o meno.

Distanza di Levenshtein all'interno di GDA

Come detto precedentemente, per l'analisi della similarità delle stringhe ci si è avvalsi di un tipo di misurazione molto comune nella teoria dell'informazione: la distanza di Levenshtein, algoritmo che prende il nome dal suo ideatore (Vladimir Levenshtein) che lo ha sviluppato nel 1965 e che il linguaggio PHP implementa già tra le sue funzionalità di base.

La Distanza di Levenshtein, comunemente chiamata Edit Distance o Distanza di Edit, è un tipo di misurazione che mette a confronto due stringhe calcolando il

numero di operazioni da effettuare per far si che una stringa venga trasformata nell'altra. Il confronto avviene carattere per carattere valutandone l'uguaglianza ed eventualmente le azioni da intraprendere per la sua modifica. Le operazioni previste nel calcolo originariamente ideato da Levenshtein sono 3, ognuna con un costo:

- *cancellazione*: si procede con l'eliminazione di un carattere dalla stringa non presente nell'altra, il costo è di 1.
- *aggiunta*: si aggiunge un carattere mancante che è invece presente nella stringa di confronto, anche qui il costo è di 1.
- *sostituzione*: viene sostituito un carattere perché differente da quello nell'altra stringa, il costo per questa operazione è di 2 perché prevede prima l'eliminazione del carattere diverso e, successivamente, l'aggiunta di quello nuovo.

La distanza di Levenshtein nasce per la trasformazione delle stringhe, tuttavia si è rivelata un ottimo strumento anche per l'analisi della similarità. A questo scopo, la funzione che esegue il calcolo implementata in PHP, utilizza un settaggio differente dei costi delle operazioni, dove vengono impostati tutti quanti ad 1.
Il risultato finale della misurazione viene trasformato in un valore percentuale per poter essere meglio interpretato dall'utente .

DATABASE

Come mostrato in figura 10, il GeoData Annotator si appoggia ad un database MySQL per l'archiviazione di tutti i dati riguardanti il framework.

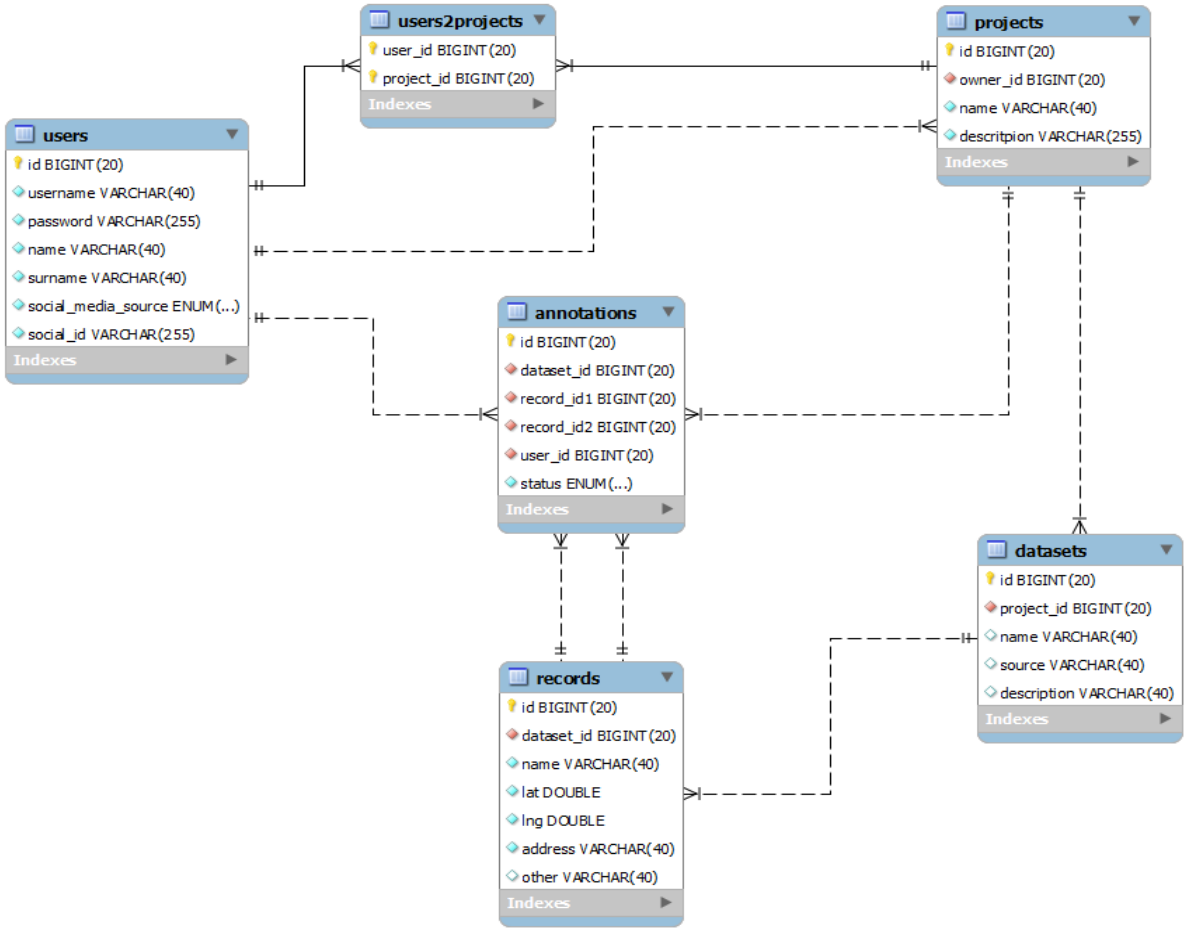

Figura 10: GeoData Annotator: schema entità-relazione del database.

Il database si compone di sei tabelle:

- *users:* contiente i dati degli utenti al momento della registrazione. Ad ogni utente viene dato un id univoco (che ha funzione di chiave primaria della tabella), nome e cognome dell'utente e vengono inseriti l'username, la password (criptata con il metodo MD5).
- *projects:* contiene i dati sui progetti che vengono registrati, contiene 4 attributi: *id* (chiave primaria), *owner_id* (riferimento all'utente proprietario del progetto), *name*, *description* (facoltativo).
- users2projects: gestisce i collegamenti, e quindi i permessi, fra gli utenti ed i progetti, si compone di 2 attributi*: user_id*, *project_id*.

- *datasets:* contiene le informazioni sui dataset caricati ed inseriti nel database collegandoli al relativo progetto, si compone di 5 attributi: *id* (chiave primaria), *project_id* (riferimento al progetto di cui il dataset fa parte), *name*, *source* (nome del file nella cartella di upload dei dataset), *description*.
- *records:* in questa tabella dove vengono inseriti tutti i record provenienti dai dataset, si compone di 7 attributi: *id* (chiave primaria), *dataset_id* (riferimento al dataset da cui il record proviene), *name*, *lat* (latitudine), *lng* (longitudine), *address*, *other*.
- *annotations:* in questa tabella vengono registrate le annotazioni (merge) effettuate dagli utenti. Essa si compone di 6 attributi: *id* (chiave primaria), *dataset_id* (riferimento al progetto di cui l'annotazione fa parte), *record_id1* (riferimento al primo record facente parte del merge), *record_id2* (riferimento al secondo record facente parte del merge), *user_id* (id dell'utente che lo ha effettuato), *status*.

SISTEMA DI INDICIZZAZIONE

L’indicizzazione permette di ridurre la complessità nel matching rendendo il lavoro dell’utente più facile ed affidabile supportandolo nelle decisioni. Gli indici consistono di valutazioni eseguite dal computer basate sull’esecuzione di algoritmi, nel nostro caso della similarità, che restituiscono un risultato da confrontare con un metro di giudizio o una soglia (threshold) per fornire all’utente una valutazione utile alle scelte che deve compiere. Quando si esegue il merging GeoData Annotator calcola l’indice di similarità geografica e quello sulla similarità fra le stringhe, per entrambi sono previste delle soglie con cui ne vengono confrontati i risultati. Gli indici servono a fornire all’utente dei suggerimenti per aiutarlo nella valutazione del merging.

L’indice sulla similarità fra le stringhe prevede la valutazione secondo 4 livelli differenti a seconda del risultato ottenuto con la Distanza di Levenshtein:

- distanza minore di 12 : similarità molto alta
- distanza da 12 a 17 : similarità discreta
- distanza da 17 a 27 : similarità bassa
- distanza uguale a 27 o maggiore : similarità molto bassa o nulla

La valutazione viene accompagnata da colori differenti che variano dal rosso al verde in base alla somiglianza e dal risultato del calcolo.

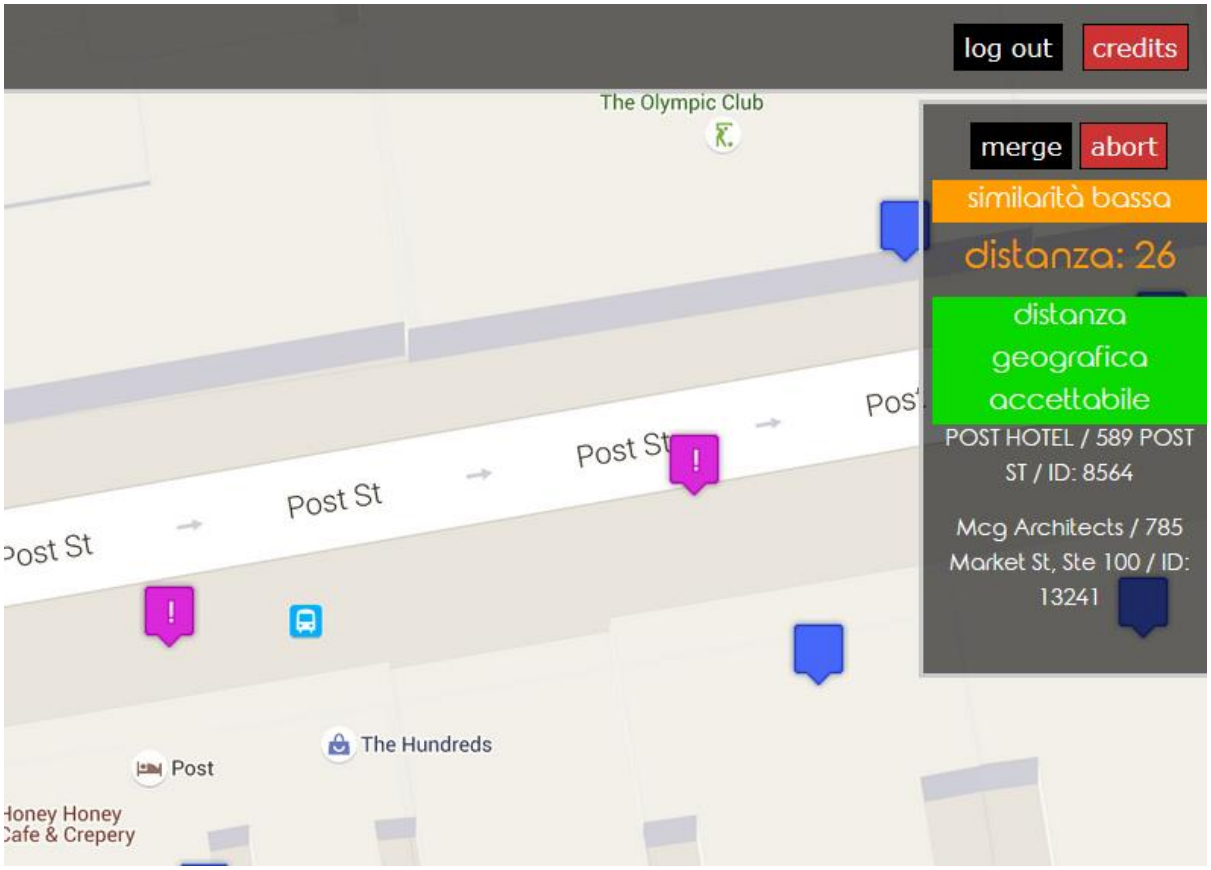

Figura 11: GeoData Annotator: esempio di calcolo sulla similarità fra le stringhe e sulla distanza geografica di due record.

L'indice di similarità geografica è basato sulla formula della Great-Circle Distance che permette di calcolare la distanza fra due punti terrestri tenendo conto anche della curvatura della Terra, sistema di calcolo sufficientemente preciso per questo scopo.
La formula per la distanza geografica è la seguente:

Dove *x* ed *y* sono una coppia di coordinate geografiche, è la latitudine, è la longitudine, ed R è il raggio terrestre (6378 Km).
Anche per il calcolo sulla distanza geografica è stata fissata una soglia (100 metri) oltre la quale i marker vengono considerati dal programma troppo distanti e ne viene sconsigliato il merge.

SISTEMA PER LA VERIFICA DELLA QUALITÀ

Come già introdotto nel capitolo dedicato alla valutazione della qualità dei dataset, GeoData Annotator nasce per facilitare la risoluzione di questa problematica. Il framework infatti è strutturato in modo da coinvolgere più utenti che possono svolgere le proprie annotazioni senza essere condizionati da quelle effettuate dagli altri partecipanti al progetto.

Quando un utente crea un progetto gli viene fornita la possibilità di coinvolgere più persone nel processo di annotazione semplicemente aggiungendo utenti (già registrati nel framework) al proprio progetto. Questi, una volta effettuata l'operazione (che solo il proprietario del progetto ha l'autorizzazione ad eseguire) possono iniziare fin da subito con la fase di annotazione. Fase che avviene, come già specificato, in modo del tutto autonomo ed indipendente.

Al termine del processo (o anche durante la sua esecuzione se si è interessati a vederne il progresso) GeoData Annotator mette a disposizione del proprietario del progetto il risultato complessivo ottenuto da tutti gli utenti all'interno del quale vengono inseriti tutti i merge effettuati.

A questo punto, grazie ai dati forniti, l'utente può ricavarne un dataset di record più attendibile applicando la Kappa di Fleiss per il calcolo dell'accuratezza, funzione che sarà resa disponibile in futuro nel framework (vedi cap. Future Works).

ESPERIMENTI E TEST

CASO DI STUDIO: STRUTTURE ALBERGHIERE DI PISA

Questo capitolo illustrerà un esperimento di utilizzo del GDA per le strutture alberghiere situate nei comuni di Pisa, Marina di Pisa, Tirrenia e Calambrone. A questo esperimento hanno preso parte tre annotatori.
Per questo esperimento sono stati utilizzati 3 dataset geografici. Il primo, considerato di riferimento, è il dataset ufficiale rilasciato dalla regione Toscana[7]. Il secondo e il terzo sono dataset scaricati rispettivamente da Google Places e Facebook., Questi ultimi dataset sono risultati dell'applicazione *Tour-pedia*[8].

Metodologia utilizzata:

Le metodologia utilizzata era atta a misurare sia i risultati che la soddisfazione dell'utente.
Sono stati svolti due test che hanno visto a confronto il dataset della regione Toscana con quello di Google Places nel primo caso e con quello di Facebook nel secondo.
La scelta di questi confronti è sorta dalla volontà di voler confrontare un dataset ufficiale rilasciato da un ente governativo con le informazioni reperibili tramite social media.
Ognuno dei tre annotatori ha svolto il proprio compito di merging dei record in totale autonomia per entrambi i test.

Dataset Utilizzati:

I dataset utilizzati per il test hanno una provenienza ed una tipologia differenti. Sono stati utilizzati un openData messo a disposizione dalla regione Toscana[1] filtrato dei soli record legati alla provincia di Pisa, il secondo dataset proviene invece dai database di Google mentre il terzo è stato ottenuto attraverso Facebook.

[7] *http://dati.toscana.it/dataset/rt-strutric*

[8] *nell'abito del progetto europeo Opener*

Il dataset della Toscana, che nella sua forma originaria, prevede più di 15000 record in quanto comprendente tutte le strutture della regione. Per ottenere il dataset ridotto, si è provveduto a filtrare i record riguardanti la provincia di Pisa. Nella sua forma ridotta, il dataset contiene 424 record di strutture alberghiere. Il dataset prodotto da un crawling di Google Places ha fornito 358 entità, mentre da Facebook sono state raccolte solo 90 entità.

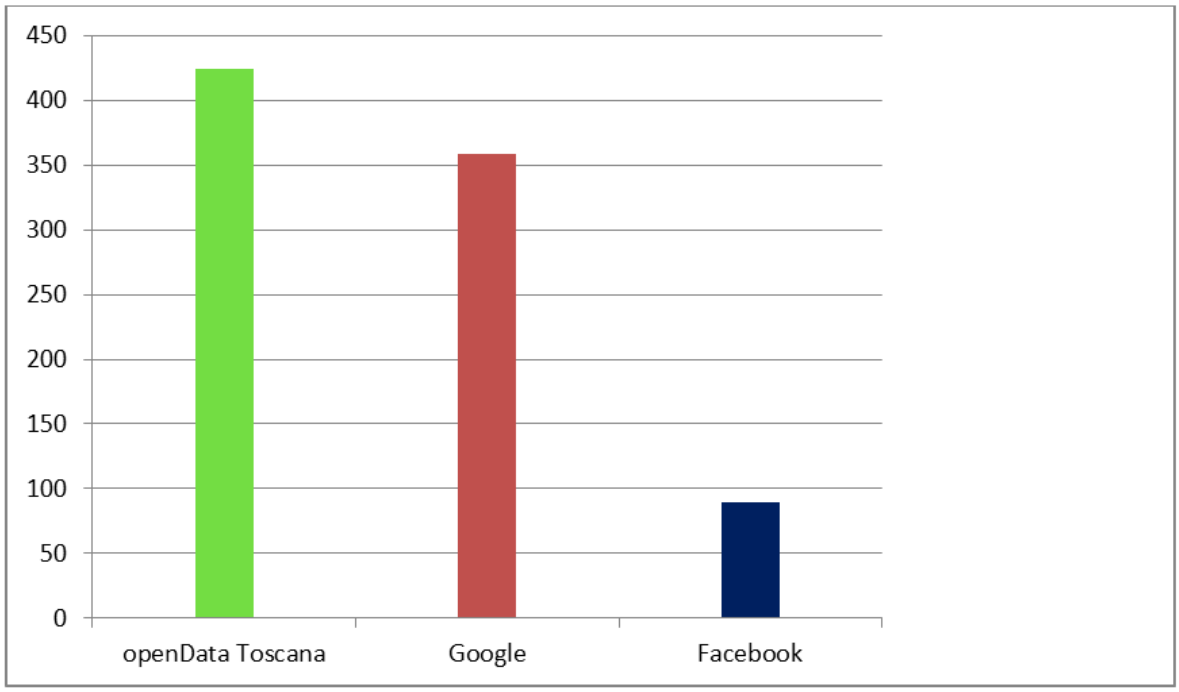

Figura 12: Rapporto fra i dati iniziali a disposizione.

Nel grafico in figura 12 è possibile visionare la differenza nella quantità di record fra i 3 dataset dove spicca la maggiore grandezza di quello messo a disposizione come fonte ufficiale dalla regione Toscana.

Tutti e tre i dataset a disposizione, compreso quello ufficiale della regione Toscana, presentavano degli errori (di cui alcuni molto gravi) nelle coordinate registrate che hanno reso difficoltosa la rilevazione delle entità rappresentate e la successiva fase di merging.

Nel confronto fra il dataset della regione Toscana con quello di Google, in un rapporto di 424 e 358 record si è ottenuto il seguente risultato:

- Annotatore 1: 122 annotazioni
- Annotatore 2: 104 annotazioni
- Annotatore 3: 97 annotazioni

Nel confronto fra il dataset della regione Toscana con quello di Facebook, in un rapporto di 424 e 90 record si è ottenuto il seguente risultato:

- Annotatore 1: 35 annotazioni
- Annotatore 2: 22 annotazioni
- Annotatore 3: 22 annotazioni

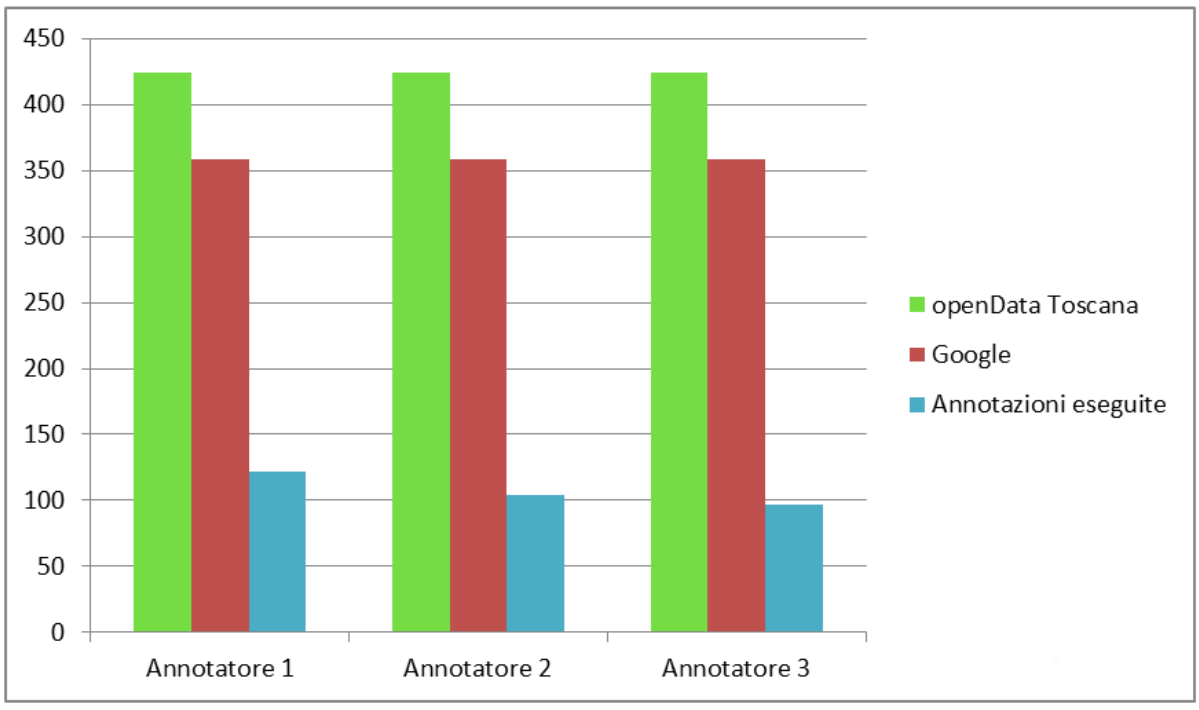

Figura 13: Rapporto nel primo test fra i dati in input e le annotazioni suddivise per gli annotatori.

I grafici nelle figure 13 e 14 mostrano la proporzione fra le grandezze dei due dataset oggetto dei test e le annotazioni eseguite. Si nota come il risultato sia stato ampiamente influenzato dalla ridotta grandezza del dataset proveniente da Facebook in confronto a quello di Google al pari della terza fonte (openData Toscana).

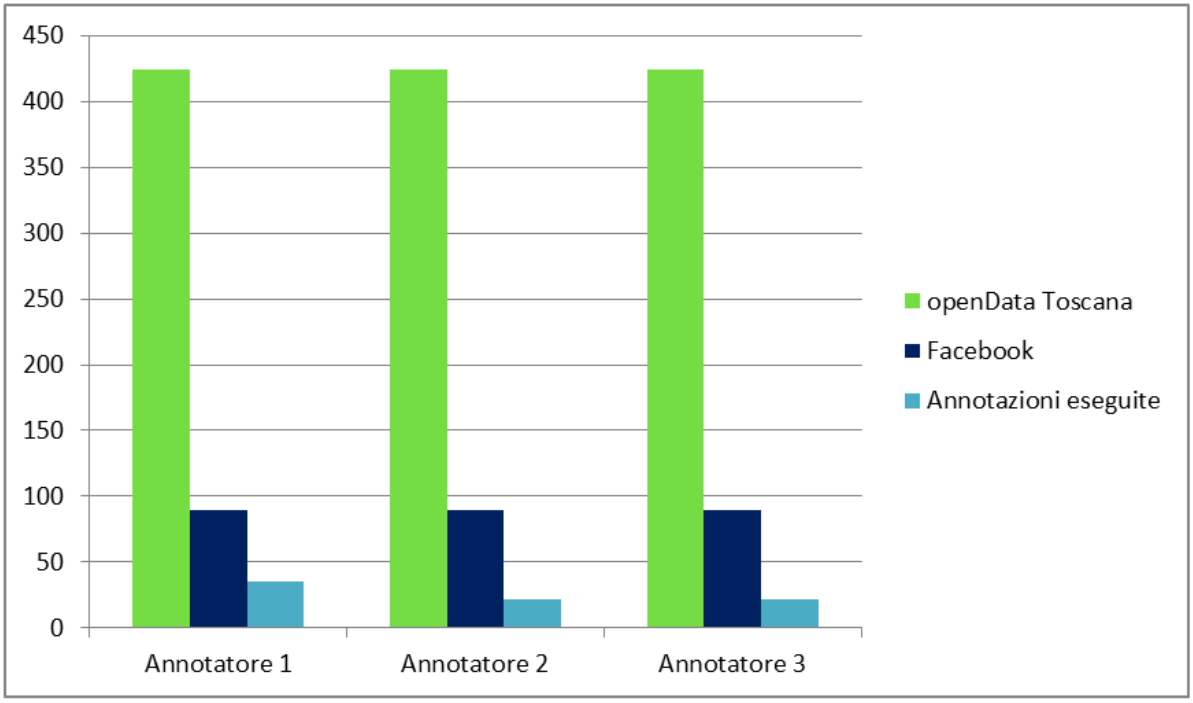

Figura 14: Rapporto nel secondo test fra i dati in input e le annotazioni suddivise per gli annotatori.

Dai test effettuati è stato possibile calcolare il rapporto in percentuale fra le annotazioni eseguite ed i dataset di partenza raccolti ed organizzati nelle seguenti tabelle:

	openData Toscana	***Google***
annotatore 1	*28,77%*	*34,08%*
annotatore 2	*24,53%*	*29,05%*
annotatore 3	*22,88%*	*27,1%*

Tabella 1: risultati in percentuale del primo test (openData Toscana e Googel).

	openData	***Facebook***
annotatore 1	*8,26%*	*38,89%*
annotatore 2	*5,19%*	*24,44%*
annotatore 3	*5,19%*	*24,44%*

Tabella 2: risultati in percentuale del secondo test (openData Toscana e Facebook).

CONCLUSIONI E FUTURE WORKS

In questo elaborato è stata introdotta la problematica del merging dei dataset geografici. Questo problema, oggi, è particolarmente sentito visto il proliferarsi di dataset sia open che provenienti da Social Media differenti. Nonostante l'esistenza di algoritmi che automatizzano tale processo, è necessaria l'annotazione manuale dei risultati da parte degli esperti di dominio sia come training per i sistemi automatici sia come verifica dei risultati stessi.

La tesi, dopo un analisi delle problematiche della complessità di questo task, introduce il GeoData Annotator (GDA), un framework web-based per l'annotazione di record di dataset geografici. GDA supporta il processo di annotazione riducendone la complessità attraverso l'utilizzo di indici basati sull'analisi geografica che delle stringhe.

Queste funzioni permettono la riduzione del tempo necessario per il merging di due dataset geografici che si traduce in una diminuzione del tempo necessario per l'esecuzione del task.

Oltre a queste funzionalità, che si trovano anche in altri software, GDA permette il supporto per la collaborazione all'annotazione, infatti diverse persone possono annotare lo stesso progetto. Inoltre, essendo un'applicazione web, non è necessaria alcuna installazione nel client dell'utente. Queste funzionalità rendono GDA, di fatto, uno strumento all'avanguardia in questo campo. A riprova di quanto scritto, la comunità scientifica ha apprezzato il lavoro svolto, testimonianza di ciò è il primo paper su GDA che è stato approvato alla International World Wide Web Conference (WWW2015). GeoData Annotator sarà rilasciato come un'applicazione Web pubblicamente disponibile.

Gli esperimenti condotti, anche se in un caso locale come le accommodation di Pisa, hanno permesso di dimostrare l'efficacia dello strumento. Durante lo sviluppo è stata infatti riposta molta attenzione nella semplicità di utilizzo, con la certezza che l'uso di uno strumento piacevole permetta la diminuzione dell'errore umano in un task considerato lungo.

Ovviamente la differenza di annotazione tra diversi annotatori è una problematica ben conosciuta in letteratura scientifica. Quindi si può pensare di estendere GDA con un meccanismo per la valutazione della qualità delle annotazioni.

Infatti un'estensione di GDA potrebbe essere il calcolo del Kappa di Fleiss come strumento per la verifica della qualità delle annotazioni fatte in termine di accuratezza.

Ovviamente lo sviluppo di GDA passa anche dall'aumento della capacità del tool di importare diversi tipi di datasets con schemi dati diversi. Per questo, un'altra possibile estensione potrebbe essere la possibilità di inserire nuovi dataset non solo in formato CSV, ma anche in SQL o XML.

Come già visto nel capitolo sullo Stato dell'Arte, alcuni software implementano un'interfaccia per la visualizzazione dei record che non fa uso di mappe geografiche, una rappresentazione, questa, più schematica ed analitica delle annotazioni. Questa visualizzazione permette una più facile navigabilità di un dataset da parte degli utenti. Per questo prevediamo, in futuro, di ampliare GDA con questa tipologia di visualizzazione oltre che a diversi tipi di export delle annotazioni fatte.

Uno dei punti di forza di GDA è lo sfruttamento di indici sia geografici che di testo anche se, allo stato attuale, è stata implementata solo la distanza di Levenshtein. Per questo è possibile pensare ad estensioni di GDA per l'uso della distanza di Jaccard o del coseno di similarità per l'analisi di stringhe. Ciò darà all'utente finale la possibilità di scegliere quale algoritmo utilizzare per le proprie annotazioni permettendo così al GDA di potersi meglio adattare a diverse esigenze.

BIBLIOGRAFIA

- Stefano Cresci, Davide Gazzè, Angelica Lo Duca, Andrea Marchetti, Maurizio Tesconi. *GeoData Annotator: A Web Framework for Collaborative Annotation of Geographical Datasets*. Istituto di Informatica e Telematica C.N.R. Pisa. 2015. *http://www.www2015.it/documents/proceedings/companion/p23.pdf* (visitato il 2 novembre 2015).

- Jurafsky, Daniel, e James H. Martin. *Speech and Language Processing: An introduction to Natural Language Processing, Computational Liguistics, and Speech Recognition*. New Jersey, Prentice Hall, 1999.

- Hyunmo Kang , Vivek Sehgal , Lise Getoor. *GeoDDupe: A Novel Interface for Interactive Entity Resolution in Geospatial Data.* Istituto di Informatica dell'Univrsità del Maryland. *http://linqs.umiacs.umd.edu/projects//geoddupe/* (visitato il 2 novembre 2015).
- Draisbach, Uwe, and Felix Naumann. "DuDe: The duplicate detection toolkit."*Proceedings of the International Workshop on Quality in Databases (QDB)*. Vol. 100000. No. 1000000. 2010.
- Christen, Peter, and Tim Churches. "Febrl-Freely extensible biomedical record linkage." *Made available in DSpace on 2011-01-05T08: 29: 34Z (GMT). No. of bitstreams: 4 TR-CS-02-05. pdf. jpg: 1669 bytes, checksum: 20bae79b191f4edb7f3143e73a8ee60a (MD5) 1523-01.2003-06-27T03: 04: 07Z. xsh: 356 bytes, checksum: 0d2e257a7c88d3fd4f6c4d95e169f36e (MD5) TR-CS-02-05. pdf: 557699 bytes, checksum: 4348ee16e5bf6d0de74667b7d6ec58c4 (MD5) TR-CS-02-05. pdf. txt: 131823 bytes, checksum: 1aa6b52187b652236b425dbdf6922eef (MD5) Previous issue date: 2002-10* (2002).
- Pawel Jurczyk, James J. Lu, Li Xiong, Janet D. Cragan, Adolfo Correa. *FRIL: A Tool for Comparative Record Linkeage*. American Medical Informatics Associations (AMIA) 2008 Annual Symposium.

- Rainer Schnell, Tobias Bachteler. *A Toolbox for Record Linkage.* Università di Costanza. *https://www.uni-due.de/soziologie/schnell_downloads.php* (visitato il 2 novembre 20159.