

1. RICONOSCIMENTI

Ringrazio il mio relatore Fabio Casati, il mio correlatore Cris-
tian Parra e LifeParticipation¹ per avermi seguito durante
tutto il percorso dell’internship che ha portato a questa tesi
e al lavoro da me fatto; ringrazio anche Manuel Cattin Cosso
e chi mi ha sopportato nei momenti di stress di quest’anno.

2. REFERENCES

APPENDIX

A. IL BOOKLET COME APPLICAZIONE PER IL TESTING

Nella sezione dedicata alla soluzione è stato presentato bre-
vemente il Booklet, un’interfaccia pensata per svolgere una
delle funzioni di base di Reminiscens, cioè quella di mostrare
contenuti che possano risultare familiari all’utente in modo
da stimolare l’afflusso di ricordi e far partire la narrazione di
episodi correlati al contenuto presentato. Tale applicazione
è stata intesa per permettere, come sviluppo futuro, alle
famiglie degli anziani la personalizzazione del booklet, ag-
giungendo ad esempio immagini e musica di propria scelta;
il tutto per riuscire a creare un “aggregatore di ricordi” che
sia il più efficace possibile. Come anticipato in sezione ??, la
scelta da noi compiuta è quella di offrire un’interfaccia che
riprende un libro antico, con una grossa copertina in pelle e
il titolo dorato, il tutto con uno sfondo che rappresenta la
superficie in legno massiccio di un vecchio tavolo, in modo
da mostrare un ambiente familiare che possa essere il primo
passo per andare oltre il gap che indubbiamente esiste tra gli
anziani e le nuove tecnologie; proprio per come è stata pen-
sata, la UI si presta perfettamente ad un’attività preliminare
di testing. Per verificare la bontà delle scelte di design, così
come l’adeguatezza di alcuni dati raccolti dai moduli ETL,
si potrebbe pensare a un workshop avvalendosi della colla-
borazione di un centro ricreativo per anziani, strutturato in
questo modo: dopo aver diviso i collaboratori in (4?) gruppi,
a ogni gruppo verrebbe assegnato un iPad, con a bordo una
versione del booklet con contenuti diversi; l’idea è quella di
comporre ogni piattaforma di test con contenuti riguardanti
un diverso periodo della vita, tenendo conto delle diverse età
dei partecipanti e ottenendo quindi infanzia, giovinezza, età
adulta e presente. Ultimo parametro da considerare nella di-
visione dei gruppi e nella composizione dei booklet è quello
del luogo in cui centrare il calcolo del contesto da presentare
ai volontari: perchè il workshop possa restituire dei risultati
utili, c’è bisogno che ognuno possa vedere immagini, ascol-
tare musica e ricordare eventi che almeno in teoria possano
essere significative (e.g. una persona cresciuta a Roma dif-
ficilmente potrà rievocare memorie passate guardando delle
fotografie del Monte Bondone). Il feedback ricevuto da que-
sta e altre prove è indispensabile per capire se la strada che
stiamo percorrendo con Reminiscens è quella giusta e, in
caso di difetti del nostro approccio, fornirebbe importanti
linee guida per correggerlo.

B. DOCUMENTAZIONE API

A seguito del lavoro compiuto, viene qui rilasciata la do-

- **decade[]** : un array di decenni nella forma *dddd*. Ogni decade deve avere nome *decade[]* - invece di *decade* - per far sapere all’API che si tratta di un array.
- **lat[], lon[]** : due array di coordinate geografiche, Latitudini e longitudini devono avere nome rispettivamente *lat[]* e *lon[]* - invece di *lat* e *lon* - per far sapere all’API che si tratta di due array.

mentre la terza accetta solo il primo, cioè *decade[]*.

B.2 Lettura dei dati

- **GET** /media
- **GET** /events
- **GET** /people
- **GET** /works

Le prime tre route accettano i seguenti parametri:

- **decade**: una decade nella forma *dddd*.
- **place** : è la forma letterale rappresentante il luogo; quando usato, è cercato nel db; se assente, viene effettuato il Geocoding per ottenere le coordinate corrispondenti; può essere associato a *radius*; se usato insieme a *lat* e *lon*, viene ignorato.
- **lat** e **lon** : ovviamente latitudine e longitudine; devono sempre apparire in coppia; possono essere associate a *radius*; se utilizzate insieme a *place*, quest’ultimo viene ignorato.
- **radius** : rappresenta la distanza nella da *place* / *lat* e *lon* che l’API deve utilizzare per considerare validi o no i possibili risultati; di default ha valore 0.

mentre l’ultima accetta solo il primo, cioè *decade*.

C. STATO DELL'ARTE

A partire dal problema che cerchiamo di risolvere, è importante iniziare a calarsi nel concreto, rendendosi conto di quali sono i campi a cui ci si vuole ispirare per costruire un'architettura funzionale alla soluzione della questione.

C.1 Web semantico

con il termine web semantico, coniato dal suo ideatore Tim Berners-Lee, "si intende la trasformazione del World Wide Web in un ambiente dove i documenti pubblicati (pagine HTML, file, immagini, e così via) sono associati ad informazioni e dati che ne specificano il contesto semantico in un formato adatto all'interrogazione e l'interpretazione (es. tramite motori di ricerca) e, più in generale, all'elaborazione automatica."³ Questo non significa limitare le numerosissime forme in cui è possibile presentare i contenuti sul web, bensì è un modo per fornire una struttura capace di rispondere a delle query, permettendo la lettura automatica di pagine e documenti. Risorse con questa caratteristica sono enormemente utili per la facilità con la quale è possibile estrarre informazioni e adattarle alle proprie necessità; sono quindi queste la scelta più ovvia come semplice punto di partenza per la ricerca di contenuti atti a stimolare la reminiscenza. Una delle realtà più grosse che opera nel web semantico è DBpedia, il cui approccio è quello di parsare periodicamente i dump rilasciati da Wikipedia, estrarre le uniche informazioni strutturate delle pagine, le cosiddette infobox, unendo alcuni attributi delle versioni localizzate dello stesso articolo e rappresentando le risorse tramite Resource Description Framework (RDF). Ogni risorsa è definita tramite un URI, un identificatore unico dell'entità presentata. Il lavoro di DBpedia è eccellente, ma è involontariamente ostacolato dagli editori di Wikipedia, che per la compilazione delle infobox non hanno dei template ben definiti, finendo per rappresentare lo stesso attributo in articoli diversi con nomi diversi; DBpedia è interrogabile tramite SPARQL, un linguaggio SQL-like costruito per leggere RDF. Dopo il rilascio del primo dataset al pubblico nel 2007, la situazione attuale è quella di un grosso grafo di altri set di dati collegati tra di loro tramite l'accoppiamento di risorse appartenenti a insiemi diversi, ma rappresentanti la stessa entità. Pubblicare questi Linked Data rende molto più facile la ricerca di informazioni sul web, rendendole molto più precise e complete. Alcuni altri progetti di web semantico sono: Freebase, che a differenza di DBpedia è un progetto proprietario; Uniprot, una knowledge base contenente dati liberamente accessibili sulle sequenze di proteine; GeoNames, un database geografico contenente oltre 10 milioni di luoghi, accessibile e scaricabile sotto licenza Creative Commons. È disponibile online lo stato dei collegamenti tra datasets semantici in forma di grafo, aggiornato a settembre 2011?⁴<http://lod-cloud.net/versions/2011-09-19/lod-cloud.html>, mentre una descrizione di DBpedia in particolare è reperibile in [?].

Un approccio diverso nell'attribuzione di un significato ai contenuti presenti sul web è quello dei microformati⁴, un'estensione di markup che, tramite l'utilizzo degli attributi HTML class, rel e rev, consente l'attribuzione di regole se-

mantiche a normali pagine web: figura ?? mostra la differenza tra le informazioni sul contatto di Nicola Parrello in maniera canonica e le stesse, se riscritte con il microformato hCard (specifico per i contatti).

Attraverso i microformati, ad esempio, un software come un browser può estrarre facilmente informazioni e navigare le relazioni tra oggetti diversi, mantenendo comunque la normale leggibilità di una pagina web. Oltre ad hCard, solo hCalendar è stato formalizzato: altri microformati, come hAtom, hMedia e hNews (rispettivamente per feed Atom, contenuti multimediali e notizie) sono solo più o meno abbozzati, e non rappresentano quindi uno standard.

C.2 Web search e contestualizzazione dell'informazione

Il passo successivo nella ricerca di una soluzione al problema che è tema di questo documento, dopo aver scelto da dove reperire i dati, è quello di decidere come permettere a un eventuale utente di accedere ai contenuti che vogliamo proporre. L'approccio che sembra più naturale è quello di costruire un piccolo motore di ricerca, in modo tale da creare una maschera che permetta una richiesta di risorse in maniera uniforme, anche se da fonti diverse; inoltre, per avere una ricerca rapida e reattiva, il buon senso suggerisce che è una buona idea quella di indicizzare il materiale ricercabile. In questo campo è difficile non considerare Google come lo stato dell'arte, essendo loro il search engine più utilizzato al mondo. Perché utenti da ogni parte del globo possano utilizzare un servizio veloce ed efficiente, Google utilizza sette componenti dinamiche, che salvano e leggono dati in altre strutture; il procedimento può essere riassunto così:

1. Il Server URL parte da un URL di base e, leggendo il Document Index, invia gli URL ai Crawler;
2. I Crawler scaricano le pagine web e le mandano nello Store Server.
3. Lo Store Server comprime le pagine nel formato zlib (RFC1950)⁵, riducendo la loro dimensione a un terzo dell'originale, che vengono poi di un docId univoco e immagazzinate nel repository;
4. Il repository viene letto dall'Indexer, che decompime i documenti e li parsa, assegnando a ogni parola (a cui viene assegnato un wordId univoco) informazioni su posizione, grandezza del testo, numero di occorrenze e altro; ogni voce viene poi aggiunta ad un indice parzialmente ordinato. Dalle pagine lette, l'Indexer estrae anche i dati dei link in esse contenuti, oltre a servirsi del testo analizzato per costruire un lessico, utile per la vera e propria funzione di ricerca;
5. Lo URL Resolver combina i documenti con i dati dei link, per costruire tabelle di coppie di docId, utilizzati per calcolare il PageRank;
6. Il Sorter riordina l'indice (ordinato per docId) in un indice inverso (ordinato per wordId), aggiungendo altri dati per rendere la ricerca più precisa;
7. Infine viene calcolato il PageRank, una misura dell'importanza di una pagina web calcolata in base a quali e

³http://it.wikipedia.org/wiki/Web_semantico

⁴<http://microformats.org/>

⁵<http://www.ietf.org/rfc/rfc1950.txt>

quante sono le pagine che hanno dei link che puntano a essa.

Per una trattazione più approfondita, rimandiamo a [?].

Negli anni Google è diventato colmo di pubblicità, mostrata e scelta in base ai dati raccolti dalle ricerche degli utenti, ma certamente non è l'unica possibilità in quanto a motori di ricerca: un esempio su tutti è quello di Duck Duck Go che, a differenza di Google, è un Semantic Search Engine. Il sistema quindi si occupa di valutare l'effettivo significato dei termini di ricerca, eliminando in maniera più precisa i risultati irrilevanti. Ma quello che lo contraddistingue in maniera maggiore dagli altri è la sua attenzione alla privacy: Duck Duck Go infatti non conserva e non vende a terzi nessuna informazione sulle ricerche, permettendo una ricerca anonima e al sicuro da data leak, richieste legali da parte delle istituzioni e disonesti.⁶ Quello che può infastidire gli utenti, cioè gli annunci di cui sopra, sono però molto interessanti per la nostra ricerca, in quanto introducono un altro aspetto molto importante: perchè la ricerca sia semplice da utilizzare, e soprattutto piacevole, può essere necessario renderla per così dire automatica, nascondendo l'azione manuale dell'inserimento di parametri e mostrando i risultati direttamente come contesto di qualche altra azione. Premiata nel 2012 dal magazine Popular Science come innovazione dell'anno⁷, Google Now è un'estensione dell'applicazione mobile Google Search che, oltre ad essere un assistente vocale, analizza abitudini, ricerche e posizioni ricorrenti per fornire dati e risultati contestuali, mostrando all'utente le informazioni prima che egli ne faccia richiesta esplicita all'app.

C.3 Algoritmi spazio-temporali

Abbiamo definito la raccolta e la ricerca, ma un ultimo punto rimane fumoso: in base a quale criterio scegliere un set di risorse rispetto a un altro, con la condizione che queste siano effettivamente rilevanti ed efficaci nella stimolazione della reminiscenza? Il vissuto di un individuo può essere riassunto in una lista di eventi, di storie di vita; allo stesso tempo, un evento può essere identificato da una coppia $\langle D, L \rangle$, dove D è la coordinata temporale e L descrive la posizione nello spazio. Utilizzando questo modello, il criterio di cui sopra diventa quello della distanza spazio-temporale tra due entità, e il calcolo di questa distanza diventa il mezzo per effettuare un'indicizzazione preliminare delle risorse.

Un interessante esempio di sfruttamento estensivo delle informazione su tempo e spazio al fine di restituire dei risultati a una query è TimeTrails. TimeTrails è un sistema per l'estrazione e l'esplorazione di coordinate spazio-temporali che si possono trovare nei documenti di testo, composto di tre componenti principali: (1) una pipeline che, dopo aver letto i dati forniti da dei moduli che estraggono documenti di testo da varie sorgenti (e.g. la vetrina di Wikipedia⁸), si occupa di estrarre le date e i luoghi contenuti nel testo, normalizzarli (e.g. da "25 luglio 1991" a "25/07/1991" per le prime e da "Pergine Valsugana" a "46.06853620, 11.23528960" per i luoghi), e calcolare il numero e la posizione delle occorrenze

trovate, in modo da verificare se la coordinata dello spazio e quella del tempo identificano un evento ben preciso oppure se sono due riferimenti senza alcuna correlazione. Il risultato dell'elaborazione, cioè il documento originale unito a una sequenza ordinata di tuple $\langle D, L \rangle$, viene poi salvato in (2) un database ottimizzato per contenere dati su luoghi⁹, dal quale (3) un'interfaccia permette la ricerca testuale di documenti, con un risultato che verrà mostrato su una mappa sotto forma di traiettoria, rappresentata da tutte le tuple lette dal db. Nel caso la query dovesse restituire più di un documento, può risultare interessante vedere dove e quando le traiettorie si incrociano: se i documenti sono biografie, l'intersezione delle traiettorie in uno dei punti segnati dalle tuple $\langle D, L \rangle$ può indicare che i personaggi si sono incontrati. TimeTrails è presentato dai suoi ideatori in [?].

⁶<https://duckduckgo.com/privacy>

⁷<http://www.popsci.com/bown/2012/product/google-now>

⁸<http://it.wikipedia.org/wiki/Wikipedia:Vetrina>

⁹<http://postgis.net/>