Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Informatica

Ricostruzione del Profilo Mediale tramite dati di Social Network Sites

Tesi di Laurea in Complementi di Basi di Dati

Relatore:
Danilo Montesi

Presentata da:
Marco Gemelli

Co-relatore:
Matteo Magnani

Sessione III
Anno accademico 2009-2010

Dedico questa tesi al mio nonno Aldo Nardi

Ringraziamenti

Ringrazio innanzitutto Danilo Montesi e Matteo Magnani per avermi dato l'opportunità di svolgere questo lavoro e per la fiducia che hanno riposto in me per il suo svolgimento.

Ringrazio gli amici e colleghi del dipartimento di Informatica, senza i quali non sarei riuscito a portare termine questo percorso di studi denso di progetti ed esami impegnativi.

Ringrazio gli amici di Mantova e Bologna, che in modi diversi mi hanno sostenuto non facendomi mai mancare il loro appoggio.

Ringrazio gli amici del Judo C.U.S. Bologna, grazie a i quali ho potuto scaricare fisicamente le tensioni in periodo di esami e tesi, aiutandomi a mantenere un equilibrio fisico e mentale costante.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Una parte rilevante di contenuti presenti sul Web è generata dagli utenti tramite i servizi messi a disposizione dai Social Network Site (SNS). In particolare gli SNS permettono agli utenti di pubblicare nuovi contenuti sul Web, farsi conoscere da altri utenti, stabilire nuove relazioni in rete e interagire tramite i contenuti generati.

L'interesse da parte degli utenti ad usufruire dei servizi messi a disposizione degli SNS è sempre maggiore [1]. Il tempo trascorso dagli utenti sugli SNS varia a seconda della nazione che considerano. Gli utenti Australiani trascorrono mediamente 6 ore di tempo ciascuno sugli SNS, i Giapponesi 3 ore ciascuno e gli europei 4 ore ciascuno [2]. Il valore degli SNS è in continua ascesa, grazie al crescente numero di utenti che li utilizzano generando informazioni d'immenso valore per settori quali il marketing, indagini sociologiche e politiche. Il valore stimato nel 2009 dei SNS principali è di: 10.000 milioni di dollari per Facebook, 6.531 milioni di dollari per Myspace, 1.827 milioni di dollari per Bebo e 1.678 milioni di dollari per Twitter.

I diversi SNS mettono a disposizione servizi differenti, ma tutti hanno in comune la peculiarità di permettere all'utente di gestire un proprio profilo per interagire con gli altri. Chiameremo i profili che gli utenti generano sugli SNS *profili virtuali*. Il profilo virtuale di un utente è formato da molte informazioni: i contenuti generati dall'utente, le relazioni che stabilisce con altri utenti e le interazioni con i contenuti degli altri utenti. In particolare siamo

interessati a identificare le informazioni che un utente genera per descriversi e presentarsi agli altri, come nome, cognome, età, nazionalità, lavoro, interessi e molti altri ancora. Tali informazioni compongono le peculiarità fisiche e sociali di un utente reale tramite la visione che esso ci propone attraverso un determinato SNS. Queste informazioni costituiscono gli *attributi* di un profilo virtuale.

Questo scenario è ulteriormente complicato dal fatto che i profili virtuali non vengono generati all'interno di un singolo SNS. Ogni utente può utilizzare più di un SNS per soddisfare i propri bisogni: Facebook per mantenersi in contatto con gli amici, Flickr per condividere e commentare foto, Youtube per condividere e guardare video, Blogspot per promuovere i propri Blog e molti altri SNS che offrono diversi servizi. Questo fa sì che ogni utente gestisca più di un profilo virtuale in rete per proporsi e mostrarsi agli altri. I profili virtuali di un utente non sono facili da identificare e analizzare, in quanto sono distribuiti su tutta la rete, eterogenei negli attributi che li costituiscono e spesso i valori degli attributi tra più profili virtuali presentano valori incoerenti.

Scopo di questo lavoro è progettare e realizzare un sistema che recuperi per un dato utente le istanze dei suoi profili virtuali allo scopo di integrarne gli attributi ricavandone un unico profilo che sia il più possibile completo ed esaustivo. Definiamo *Profilo Mediale* il profilo ricavato dall'osservazione degli SNS usati dall'utente. Il sistema sviluppato ci permetterà quindi di ottenere profili mediali che potrebbero essere utilizzati per indagini di tipo economico e sociologico.

Presentiamo ora l'organizzazione dei capitoli:

Capitolo 2 Illustriamo i lavori correlati al nostro progetto.

Capitolo 3 In questo capitolo spiegheremo l'architettura del sistema sviluppato per ricostruire i profili mediali. Verrà fornita una panoramica generale delle sue componenti, di come esse interagiscono tra di loro e quali problemi sono predisposte a risolvere.

Capitolo 4 Per ottenere un profilo mediale è necessario integrare diverse fonti di dati spesso incoerenti. A tale scopo introdurremo l'algoritmo implementato nel nostro sistema.

Capitolo 5 In questo capitolo analizzeremo alcuni dati estratti dall'aggregatore FriendFeed. Per capire quali e quanti SNS vengono utilizzati dagli utenti di questo aggregatore. Successivamente analizzeremo i dati rilevabili dai profili virtuali dei vari SNS per stabilire quali usare per ricostruire i profili mediali.

Capitolo 6 In questo capitolo valuteremo i metodi possibili per studiare la precisione del nostro algoritmo per la ricostruzione dei profili mediali e discuteremo i risultati ottenuti.

Capitolo 7 Conclusioni e sviluppi futuri del lavoro svolto.

Capitolo 2

Lavori correlati

Gli SNS sono usati da milioni di utenti in tutto il mondo e sono stati ampiamente impiegati in molti eventi globali come un'importante fonte per diffondere notizie e informazioni [3]. L'adozione degli SNS come modello principale di comunicazione online da parte degli utenti è in continua crescita a prescindere delle differenze culturali presenti nel mondo [4].

Le informazioni negli SNS, come evidenziato da boyd [5] hanno le proprietà di *searchability* e *persistence* che sono due requisiti fondamentali per qualsiasi ricerca. La grande quantità di dati che viene generata online ogni secondo li rende difficili da recuperare e analizzare in modo corretto. Questi dati sono complessi, ridondanti e spesso incoerenti tra di loro.

TruthFinder [6] è stato il primo progetto che si è proposto di risolvere il problema di scoprire il vero valore dei dati in presenza di più fonti che forniscono valori in conflitto. Si basa su un algoritmo iterativo che sfrutta la dipendenza reciproca tra l'accuratezza delle fonti e del consenso tra i valori che queste riportano. Allo stesso modo [7] e di più di recente [8] presentano algoritmi per stimare i veri valori dei dati riportati da una serie di fonti. Il problema di questi algoritmi è che funzionano solo su dati semplici o atomici, per esempio il prezzo di un determinato prodotto di consumo o la temperatura atmosferica prevista per una giornata. La maggior parte dei dati presenti sul Web sono dati complessi e strutturati, per esempio ogni quotazione del NASDAQ è formata dai dati di "valore indice", "variazione", "percentuale

chiusura", "percentuale apertura" e "min-max del giorno".

Un algoritmo probabilistico per integrare dati complessi dal Web è stato elaborato da Papotti [9]. Questo algoritmo genera per ogni attributo che viene integrato una distribuzione di probabilità dei possibili valori, permettendoci quindi di scegliere come valore "vero" quello ritenuto più probabile.

Tale soluzione però non è applicabile ai nostri dati. L'algoritmo di Papotti lavora su dati quantitativi (quotazioni di borsa, cambi valuta ecc..), mentre nel nostro caso la maggior parte dei dati sono qualitativi (nome, gusti musicali, ecc..).

Nell'algoritmo proposto l'attributo che cerchiamo di integrare viene modellato come una variabile casuale, poi gli si associa una distribuzione di probabilità iniziale per i suoi possibili valori, che tramite la statistica di Bayes viene raffinata reiterando un algoritmo che trova man mano una approssimazione migliore della distribuzione. Da qui si ottiene la distribuzione di probabilità dei possibili valori del nostro attributo.

Affinché questa operazione dia risultati attendibili è necessario che valga il teorema centrale del limite. I test eseguiti da Papotti lavorano su centinaia di sorgenti per uno stesso attributo rendendo tale assunzione statisticamente accettabile. Nel nostro caso invece per un singolo attributo che cerchiamo di integrare abbiamo poche sorgenti e pochi valori (non tutte le sorgenti ci fornisco il dato).

Le operazioni che i sistemi precedentemente descritti devono eseguire per cercare il valore reale di un dato tra molte fonti sono molteplici:

Record Linkage (RL) tale operazione consiste nel trovare i documenti che fanno riferimento alla stessa entità/informazione in due o più fonti distinte. Questa tecnica viene utilizzata quando si devono unire un insieme di dati che non dispongono di una chiave univoca per essere identificati [10].

Data Fusion (DF) è un operazione effettuata con tecniche che combinano dati provenienti da fonti multiple per raccogliere informazioni al fine di

realizzare inferenze statistiche. I dati risultanti saranno in generale più precisi e accurati rispetto al caso in cui fossero recuperati per mezzo di una singola fonte [11].

Data Integration (DI) questa operazione combina dati che risiedono in fonti diverse in modo da offrire agli utenti una visione unificata di questi dati [12].

Nel nostro lavoro per ricostruire il profilo mediale degli utenti svilupperemo un sistema che propone una soluzione a queste tre operazioni cruciali.

La parte più delicata da affrontare è il RL, in quanto associare e recuperare i profili virtuali di un utente in modo corretto è un'operazione difficile. Per risolvere questo problema partiremo dai dati estratti dall'aggregatore FriendFeed. Questo SNS permette agli utenti di aggregare i servizi Web che questi utilizzano in un unico canale. In questo modo disponiamo per ogni utente FriendFeed di un indice di potenziali servizi da cui estrarre i profili virtuali.

Per effettuare DI e DF abbiamo a disposizione un insieme di profili virtuali eterogenei e spesso incoerenti tra loro. Svilupperemo un algoritmo probabilistico per ricostruire il profilo mediale dell'utente. Tale profilo fornirà una vista più completa e precisa rispetto a quella che potrebbe fornire un singolo profilo virtuale dell'utente.

Capitolo 3

Architettura del sistema

In questo capitolo spiegheremo l'architettura del sistema sviluppato per ricostruire i profili mediali. Verrà fornita una panoramica generale delle sue componenti, di come esse interagiscono tra di loro e quali problemi sono predisposte a risolvere.

3.1 Componenti del sistema

Descriviamo ora l'architettura del sistema sviluppato per ricostruire i profili mediali degli utenti FriendFeed. Nella figura 3.1 vengono mostrate le componenti del nostro sistema; inizieremo la descrizione dell'immagine dall'alto verso il basso. Esterni al sistema abbiamo gli SNS, per ognuno di essi abbiamo sviluppato un **Crawler** specifico che è in grado di recuperare i profili virtuali in esso contenuti e salvarli persistentemente nel **Crawl Repository**. I **Crawler** consultano l'elenco dei servizi registrati **User Services** per scegliere quali profili scaricare. Per estrarre gli attributi a cui siamo interessati è stato sviluppato per ogni SNS un **Data Extractor** con il compito di scansionare i profili virtuali disponibili, gli attributi estratti vengono salvati nel repository **Profiles Data** in una rappresentazione intermedia comune a tutti i profili virtuali. Il **Data Merger** ha il compito di ricostruire il profilo mediale. In **User Services** disponiamo di un indice di servizi registrati e di un indice di utenti che sono potenzialmente intestatari di tali servizi. Per

ogni utente il **Data Merger** consulta l'elenco dei servizi usati dagli utenti, ove presenti prende i dati estratti dai profili virtuali in **Profiles Data** e ricostruisce il profilo mediale dell'utente memorizzandolo nel repository **Media Profiles**. Per agevolare l'acceso ai profili mediali si è sviluppato un **Profile Retriever** che indicizza i profili mediali presenti in **Media Profiles** e permette all'utente finale **System end user** di consultarli.
Nel caso le ricerche effettuate dal **System end user** non producano risultato, il **Profile Retriever** richiede al **Service Discover** di ricercare nel Web dei servizi che potenzialmente corrispondo all'interrogazione formula. I nuovi servizi acquisiti vengono memorizzati in **User Services**, da qui ripartirà la fase di crawling e ricostruzione di profili mediali in modo da soddisfare la richiesta del **System end user**.

Descriviamo ora più in dettaglio le singole componenti e i problemi che devono affrontare per assolvere il loro compito.

3.1.1 Crawler

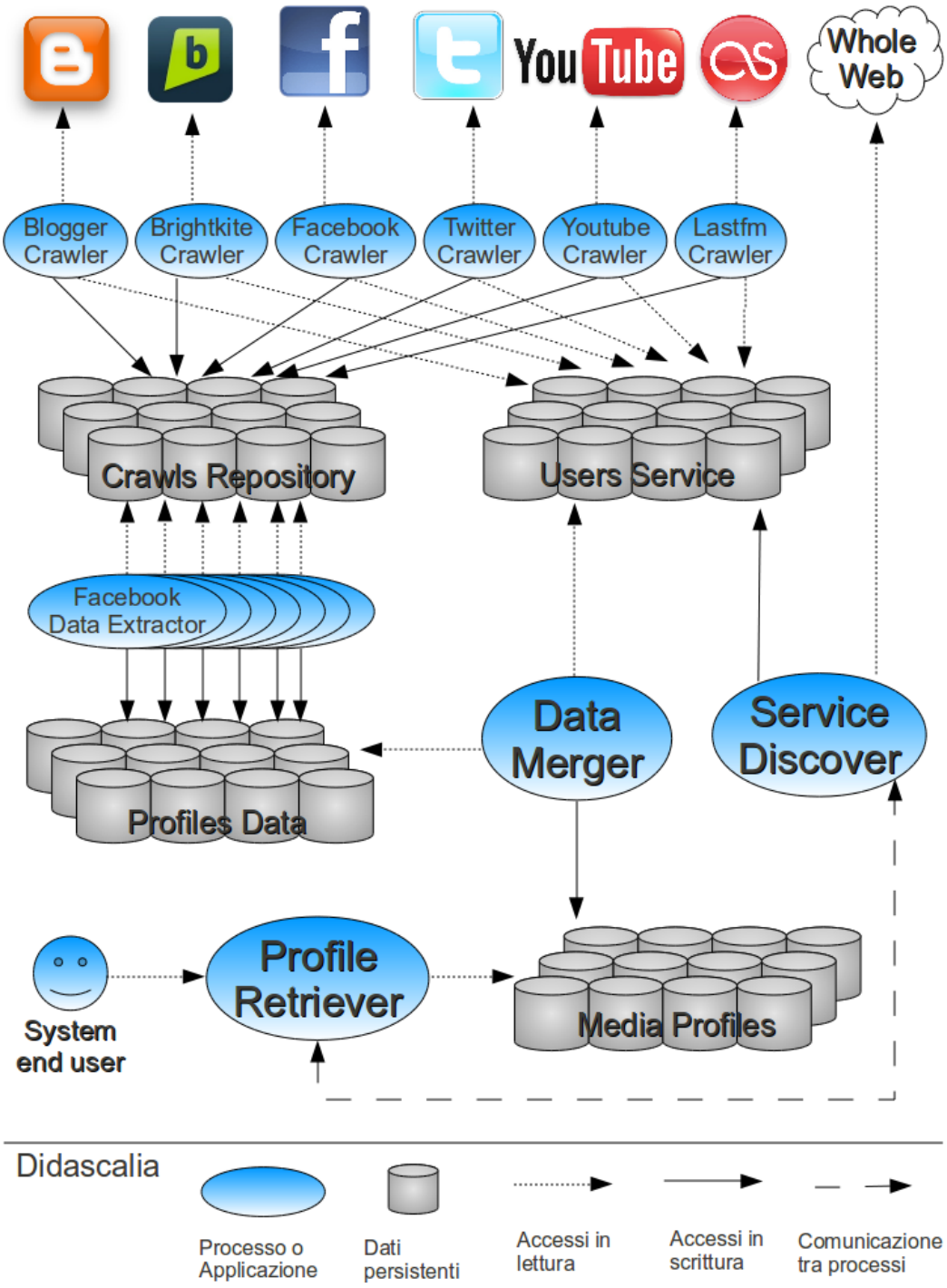
Per ogni singolo SNS è stato sviluppato un crawler dedicato. Ogni SNS rende disponibili i profili virtuali tramite API se presente, i formati in cui sono recuperabili sono essenzialmente HTML, XML e JSON.

I gestori di SNS cercano il più possibile di proteggersi da software come i crawler, poiché la loro attività è simile a quella di un software che effettua un attacco DOS (denin of service) in quanto sovraccaricano di richieste i server. Inoltre le informazioni sugli utenti sono dati sensibili e di gran valore per gli SNS che sono in generale restii a fornirli a terzi.

Ogni SNS si difende a modo proprio dai crawler, nel caso migliore restituendo un messaggio di errore oppure modificano le latenze di risposta sino a non rispondere più mandando in timeout le connessioni.

Analizzando come gli SNS si difendono dai crawler e come si recuperano i profili è stato possibile implementare per ogni SNS un crawler che scarica nel modo più rapido ed efficace possibile i profili virtuali.

Figura 3.1: Architettura del sistema



Per esempio Twitter permette un massimo di 150 richieste in un'ora alle sue API per ogni applicazione non registrata. Una volta raggiunto tale limite viene restituito un messaggio di errore e nel header della risposta HTTP viene riportato lo stato 400.

Nel caso di Blogger non viene trasmesso nessun messaggio di errore quando il crawler viene bloccato, ma viene gradualmente aumentata la latenza di comunicazione. Per massimizzare le richieste da questo provider si è appurato che conviene introdurre un intervallo di tempo casuale tra le richieste, in modo che il tempo trascorso tra due richieste consecutive non risulti regolare. Nel caso in cui venga bloccato, il crawler sospenderà la sua attività ripartendo in un secondo momento.

Mostriamo lo pseudo-codice del crawler realizzato per Twitter nel listato 3.1 e nel listato 3.2 lo speudocodice del crawler di Blogger.

Listing 3.1: Speudocodice del crawler di Twitter

```
for each u in TwitterUser do
  request_uri = ``http://api.twitter.com/1/users/show.xml?
     screen_name='' + u.username
  response = http_get(request_uri)
  if response.status == 200 then
     /*richiesta andata a buon fine*/
    save_in_crawl_repository(response.http_body)
  else if response.status == 400 then
    /*crawler bloccato da Twitter, bisogna aspettare 1 ora prima
        di poter continuare a scaricare*/
    wait(1hour)
    redo /*rieffettua la richiesta http per l'ultimo utente, la
       cui richiesta non era andata a buon fine*/
  end
end
```

Listing 3.2: Speudocodice del crawler di Blogger

```
for each u in BloggerUser do
  try
    request_uri = ``http://www.blogger.com/profile/'' + u.
       username
    response = http_get(request_uri)
```

```
    save_in_crawl_repository(response.http_body)
  catch TimeOutException e
    /*crawler bloccato da Blogger, bisogna aspettare  prima di
       poter continuare a scaricare*/
    wait(rand([30minutes,1hour]))
    redo /*rieffettua la richiesta http per l'ultimo utente, la
       cui richiesta non era andata a buon fine*/
  end
  wait(rand([30seconds,60seconds])) /*aspetta prima di
     effettuare una nuova richiesta*/
end
```

3.1.2 Data Extractor

Per estrarre dai profili virtuali gli attributi necessari alla ricostruzione del profilo mediale è stato sviluppato per ogni SNS un data extractor in grado di scansionare il profilo virtuale e recuperarne gli attributi, salvandoli in una rappresentazione intermedia comune a tutti i profili virtuali.

Siccome gli attributi non sono sempre strutturati in modo rigido, alcuni attributi vanno "derivati" da altri. Per esempio non sempre siamo in grado di estrarre singolarmente gli attributi città, regione e nazione di residenza; frequentemente siamo di fronte ad un unico attributo che contiene tutte queste informazioni spesso scritte dall'utente stesso senza una formattazione predefinita. Sarà compito di ogni singolo data extractor usare un insieme di regole di derivazione per estrarre gli attributi con una granularità il più fine possibile. Per esempio molti utenti scrivono la propria residenza con la formattazione "*città*, *regione*, *nazione*", con l'espressione regolare /([^,]*),([^,]*),([^,]*)/ possiamo testare se la residenza corrisponde con tale formato per scomporla in città di residenza, regione di residenza e nazione di residenza. Nel caso in cui siamo in presenza di un dato che può essere un username o il nome reale dell'utente, si può sempre adorare una espressione regolare /(\w+) (\w+)/ per verificare che siamo in presenza di nome e cognome per poterli correttamente estrarre.

Anche se non finalizzati all'estrazione di dati sono stati sviluppati una serie di data extractor che hanno il compito di effettuare delle statistiche/analisi sui profili disponibili. I dati presentati nel §5 sono stati ottenuti con queste componenti e il loro scopo è stato quello si fornire statistiche in modo da aiutarci a scegliere come progettare il sistema in funzione dei dati disponibili.

Dai profili possiamo ricavare molti contenuti formattati in testo libero, come descrizioni, stato corrente dell'utente ecc.. Questa peculiarità è intrinseca nei Blog ma è anche presente negli altri SNS. Dall'analisi di alcuni di questi testi si nota che molti utenti scrivono la propria età e altri dati interessanti in questi discorsi, che se opportunamente estratti posso essere usati come attributi per ricavare il profilo mediale.

Tuttavia l'estrazione di dati dal testo libero risulta essere molto ardua. In questo contesto si è provato l'utilizzo del software Boxer [13] per ottenere una rappresentazione semantica dei discorsi in testo libero in modo da individuare le relazioni semantiche che ci interessano per estrarne dei dati.

Per testare l'efficacia di questo software si è effettuata un'analisi su un campione di profili in cui a priori sapevamo essere presenti dati di nostro interesse. I risultati ottenuti si sono rilevati deludenti, in quanto vengono effettuati molti errori. Per esempio per estrarre l'età di una persona l'utilizzo di questo sistema non sembra dare alcun aiuto in più rispetto a un tradizionale pattern matching. Siccome a fronte di errori di analisi diversi cambiano le relazioni individuare, risulta difficile individuare quali sono le relazioni da prendere in considerazione per estrarre i dati. Inoltre le rappresentazioni semantiche dedotte da Boxer risultano essere per lo più errate.

Tali risultati non sono completamente imputabili al software, in quanto test effettuati su testo scritto in un inglese corretto hanno dato risultati migliori. I testi scritti negli SNS sono lontani dall'essere scritti in un inglese corretto e questo fa sì che la rappresentazione semantica derivata non sia corretta. Per ottenere buoni risultati con l'analisi semantica sarebbe opportuno adattare questo software per l'analisi di questi specifici testi. Il software è stato addestrato tramite un training set di frasi scritte in inglese corretto e annotate da linguisti, non c'è quindi da stupirsi che i risultati sui testi

presenti sugli SNS non vengano analizzati correttamente.

Per le nostre analisi ci servirebbe un software in grado di derivare rappresentazioni semantiche corrette partendo da testi scritti in un inglese scorretto e corrente. Per ottenere questo bisognerebbe disporre di un training set apposta che sia rappresentativo dei testi presenti sugli SNS.

Per il nostro lavoro si è scelto di non utilizzare questo tipo di analisi nell'implementare i data extractor, tuttavia conviene tenere in considerazione per sviluppi futuri il possibile utilizzo di tecniche di Natural Language Processing.

3.1.3 Data Merger

Il Data Merger è il componente fondamentale del nostro sistema. Per ricavare il profilo mediale di un utente, il Data Merger dispone dell'indice dei servizi registrati per ogni utente e per ogni servizio di un insieme di attributi ricavati dai rispettivi profili virtuali.

Per stabilire quali sevizi appartengono ad un utente è necessario effettuare un'operazione di Record Linkage in modo da stabilire quali di essi utilizzare in fase di integrazione. Non disponendo di un identificatore univoco tra i diversi SNS per il medesimo utente, dobbiamo stabilire un criterio di similarità tra i profili virtuali per stabilire quali fanno riferimento al medesimo utente.

Analizzando come gli utenti registrano i propri account su vari SNS scopriamo un comportamento interessante, sembra che gli utenti tendano a registrarsi sui diversi SNS cercando di mantenere lo stesso username (o uno molto simile). Per esempio l'utente registrato su Blogger come *ralphpaglia* ha pubblicato sul suo Blog dei link a Youtube, Twitter e Stumbleupon; di cui è intestatario e registrato con il medesimo username. Oppure l'utente registrato su Wordpress come *griflet* è registrato su Brightkite e Stumbleupon con un username molto simile *griflet7*. In pratica gli utenti cercano il più possibile di registrarsi col medesimo identificativo (o uno molto simile) tra i servizi che utilizzano nel Web.

Questo comportamento è molto frequente e regolare negli utenti, quindi per selezionare i servizi da integrare per ricavare il profilo mediale di un

utente, applicheremo una funzione di similarità tra gli username dei servizi recuperati, stabilendo un valore di soglia per scartare o accettare quel servizio.

Una volta effettuata questa operazione di Record Linkage disponiamo di un indice di utenti con associati i rispettivi profili virtuali, con questi dati ricostruiremo i profili mediali di questi utenti.

Disponendo di più profili virtuali con i medesimi attributi, il Data Merger ha il compito di integrare i diversi dati (spesso incoerenti) per ricostruire gli attributi del profilo mediale. Siccome questa operazione è molto delicata rimandiamo la sua formalizzazione nel capitolo §4.

3.1.4 Profile Retriever

Una volta ottenuti i profili mediali ci si è posto il problema di renderli accessibile all'utente finale. Si è dunque implementato un sistema di Information Retrieval (IR) che permette di effettuare un recupero mirato delle informazioni presenti nei profili mediali. Gli utenti saranno in grado di formulare delle interrogazioni che rappresentino il profilo degli individui, sarà compito del sistema soddisfare il fabbisogno informativo dell'utente selezionando i profili mediali che meglio rispondono all'interrogazione formulata.

Siccome il profilo mediale può essere visto come un documento (un insieme di campi con valori), abbiamo scelto il framework Ferret [14] tra quelli disponibili per sviluppare il sistema di IR per il nostro sistema. Questo sistema ci permette di generare un indice dei profili mediali, su cui effettuare le interrogazioni con un formalismo molto versatile denominato Ferret Query Language (FQL). Utilizzando opportunamente la FQL possiamo realizzare un'interfaccia di rapida comprensione per l'utente finale permettendogli la formulazione delle interrogazioni in modo intuitivo. Nell'immagine 3.2 riportiamo l'esempio di una interrogazione che ricerca tutti i profili mediali ricavati con l'integrazione di 4 profili virtuali. Per ogni attributo viene proposto un valore con la relativa probabilità che sia vero, l'utente può inoltre visualizzare gli altri possibili valori che l'attribuito può assumere.

3.1.5 Service Discover

Questa componetene ha il compito di mantenere aggiornare il Service Repository, popolandolo di nuovi servizi su cui le altre componenti del sistema possano operare. Per adempiere al suo compito passa al vaglio il Web alla ricerca di nuovi account degli utenti degli SNS. Le fonti principali da cui si possono recuperare nuovi account/servizi sono gli SNS che rendono pubblici l'indice dei loro utenti e i Blog presenti sul Web.

Figura 3.2: Interfaccia del Profile Retriever

ID	UserID	ServiceCount	FullName	FirstName	LastName	Age	Gender
12	12	4	<belén gonzález,0.2050>	<belén,0.2558>	<gonzález,0.2697>	<26,0.4962>	<female,0.5000>
60	60	4	<nana yuuki,0.2075>	<nana,0.5024>	<yuuki,0.5000>	<top,0.0000>	<female,0.5025>
139	139	4	<mehdi taheri,0.2991>	<mehdi,0.5025>	<taheri,0.5025>	<top,0.0000>	<male,0.5024>
193	193	4	<rich harris,0.2448>	<rich,0.5013>	<harris,0.5013>	<35,0.4310>	<female,0.2119>
203	203	4	<simon berry,0.2942>	<simon,0.5025>	<berry,0.5025>	<top,0.0000>	<female,0.2122>
240	240	4	<money09,0.1544>	<melanie,0.5023>	<rogers,0.5023>	<42,0.4762>	<female,0.4762>
251	251	4	<aabrilru,0.2805>	<angel,0.4897>	<abril ruiz,0.4897>	<34,0.4897>	<male,0.5023>
268	268	4	<thepostman,0.2419>	<aaron,0.5025>	<post,0.5025>	<38,0.4469>	<male,0.5020>
310	310	4	<abo46n2,0.2515>	<adam,0.5025>	<bohannon,0.5023>	<26,0.4310>	<male,0.4975>
314	314	4	<david,0.2117>	<top,0.0000>	<top,0.0000>	<24,0.3220>	<male,0.4975>
334	334	4	<acenoface1,0.2746>	<ace,0.5021>	<top,0.4980>	<40,0.4897>	<male,0.4975>
354	354	4	<activemoms,0.2513>	<active moms,0.2538>	<sherk,0.4975>	<43,0.4310>	<female,0.5024>
367	367	4	<adamdeb,0.2369>	<adam,0.5023>	<debreczeni,0.2355>	<top,0.0000>	<male,0.5023>
375	375	4	<library playground,0.2544>	<library,0.5006>	<playground,0.5000>	<top,0.0000>	<male,0.4469>
383	383	4	<adams,0.2211>	<adams,0.5023>	<smith,0.5023>	<28,0.3372>	<male,0.5025>
385	385	4	<adam vincent gilmer,0.5025>	<adam,0.5025>	<vincent,0.2818>	<40,0.4310>	<male,0.5025>
392	392	4	<addison-wesley verlag,0.2428>	<pia,0.4897>	<kleine wieskamp,0.4897>	<47,0.4310>	<female,0.4762>

Capitolo 4

Merge dei dati

Per ottenere un profilo mediale è necessario integrare diverse fonti di dati spesso incoerenti. A tale scopo introdurremo l'algoritmo implementato nel nostro sistema.

4.1 Indagine su soluzioni esistenti

Il Web mette a disposizione una sempre crescente quantità di dati, sono state sviluppate molte applicazioni che integrano dati forniti da un gran numero di fonti. I dati presenti sul web sono imprecisi, e diverse fonti possono fornire dati con valori in conflitto. Risolvere tali conflitti e determinare quali valori sono corretti è il problema cruciale di queste applicazioni e del nostro sistema.

L'algoritmo probabilistico implementato da Papotti [9] non è applicabile nel nostro caso, in quanto necessita di molti campioni per operare mentre nel nostro contesto per ogni singolo utente disponiamo di poche fonti per integrarne gli attributi del profilo mediale. Nella tabella 4.1 è riportata la frequenza di utenti che sono intestatari di un certo numero di servizi SNS, tale statistica è stata ricavata da un campione di utenti recuperati dall'aggregatore FriendFeed. Come si può vedere la maggior parte degli utenti sono registrati con pochi servizi, al più sull'ordine di grandezza della decina. Questo ci impedisce di applicare la statistica Bayesiana per integrare gli attributi degli utenti, in quanto non abbiamo abbastanza campioni per inferire una

distribuzione di probabilità con algoritmi probabilistici già noti e ritenuti attendibili.

Dall'articolo [9] si evince che è importante misurare/stimare l'attendibilità delle fonti che forniscono i dati per sviluppare un algoritmo probabilistico di integrazione. Nell'algoritmo di Papotti l'attendibilità di una sorgente è stata misurata guardando quante altre sorgenti riportano il medesimo valore copiato per un dato. Nel nostro caso i dati non sono forniti dai gestori degli SNS ma dagli utenti stessi e anche la replicazione dei dati è dunque in mano agli utenti e non alle sorgenti. Questo criterio per misurare l'attendibilità delle fonti non risulta applicabile nel nostro contesto.

4.2 Algoritmo di integrazione dei dati

L'algoritmo che andremo a sviluppare per integrare i dati ricavati dai profili virtuali terrà conto del fatto che ci sono pochi valori per disambiguare i singoli attributi e che i dati sono per lo più qualitativi e non quantitativi.

Il profilo mediale sarà rappresentato da un record probabilistico e per ogni attributo (es: nome, età, ecc) verrà fornita una distribuzione di probabilità dei suoi possibili valori.

Osservando i dati estratti dagli SNS possiamo distinguere gli attributi di un profilo in due categorie: nella prima avremo attributi semplici come età, nome e città natale. In questi attributi sappiamo che c'è un solo valore corretto da ricercare tra le osservazioni disponibili; nella seconda vi sono attributi multipli come gusti musicali o libri preferiti per i quali ci possono essere più valori corretti tra le osservazioni disponibili.

Dato un insieme U di utenti e un insieme A di tutti gli attributi del profilo mediale, il nostro scopo è calcolare per ogni utente $u \in U$ la distribuzione di probabilità di tutti gli attributi $a \in A$ dato un insieme di osservazioni per quell'attributo $O(a)$. Queste distribuzioni di probabilità calcolate a posteriori costituiranno un record probabilistico rappresentante il profilo mediale dell'utente.

Tabella 4.1: Frequenza del numero di servizi registrati per utente

Numero servizi registrati per utente	**Frequenza**
1	78685
2	33582
3	27496
4	23324
5	19915
6	16154
7	13092
8	10600
9	8252
10	6468
11	5117
12	3833
13	3089
14	2324
15	1824
16	1478
17	1201
18	927
19	720
20	569
21	483
22	410
23	334
24	262
25	198
26	184
27	157
28	118
.....	
.....	
.....	
284	2
287	1
292	1
294	1
.....	
	Totale 1.293.482

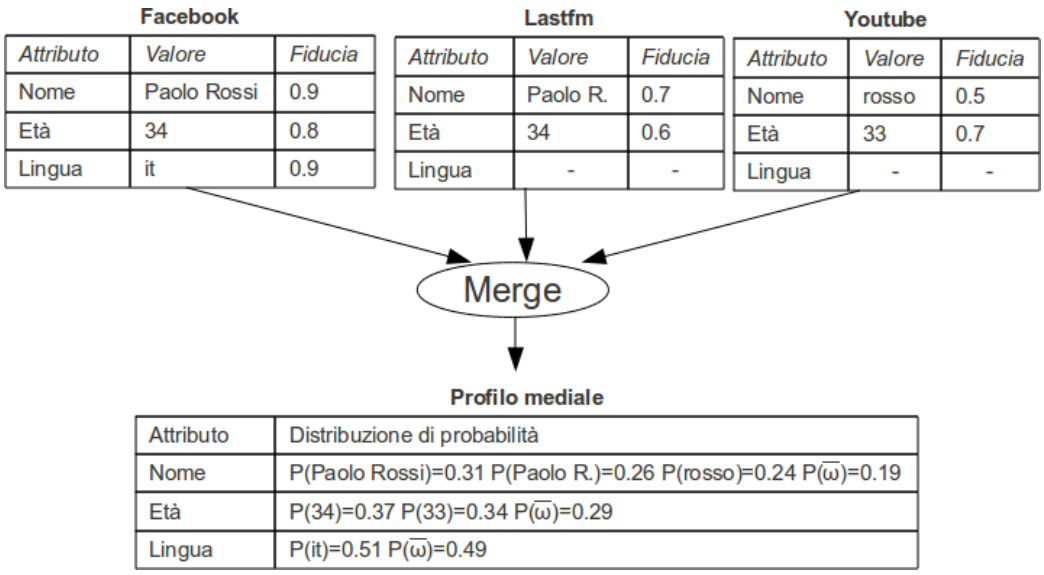
Diamo le seguenti definizioni per il nostro algoritmo:

- A insieme degli attributi del profilo mediale che vogliamo integrare.
- $O(a)$ multi-insieme delle osservazioni disponibili per l'attributo a. Per chiarezza quando scriviamo $\forall o \in O(a)$ intendiamo iterare su tutti gli elementi del multi-insieme, incluse le ripetizioni.
- $trust : O(a) \to [0, 1]$ affidabilità/fiducia che diamo all'osservazione.

4.2.1 Attributo Semplice

Nella figura 4.1 portiamo come esempio il merge di alcuni attributi semplici: nome, età e lingua. Nell'esempio ci interessa far emergere il valore "Paolo Rossi" per l'attributo nome, in quanto è un valore proveniente da una fonte che possiamo ritenere attendibile e anche perché risulta essere più completo come valore rispetto a quelli delle altre fonti. Per l'età ci interessa far emergere il valore 34, i quanto sono presenti più osservazioni che riportano

Figura 4.1: Esempio di merge di attributi semplici



questo valore. Per quanto riguarda la lingua, essendoci una sola osservazione avremo solo quel valore noto nella distribuzione di probabilità.

Com'è possibile notare nelle distribuzioni di probabilità degli attributi è stato introdotto un ulteriore valore $\overline{\omega}$ non presente nei profili virtuali, per il quale si è calcolata una probabilità $P(\overline{\omega})$. Questo valore è incognito e rappresentante un'osservazione non disponibile tra i profili virtuali che potrebbe essere vera per l'attributo. La sua introduzione serve per evitare di prendere un singolo valore come certo nel caso in cui si disponga di una sola osservazione per un attributo (come la lingua dell'utente nell'esempio). La probabilità $P(\overline{\omega})$ com'è ragionevole pensare decresce all'aumentare delle osservazioni disponibili e della loro fiducia, in quanto queste due condizioni portano ragionevolmente a pensare che il valore che stiamo cercando sia presente tra quelli osservati.

Definiamo per gli attributi semplici $\Omega(a)$ come lo spazio campionario dell'attributo a, che sarà formato da tutti i valori distinti delle sue osservazioni $O(a)$ più il valore incognito $\overline{\omega}$. Ora per l'esempio in figura 4.1 mostriamo i valori di $O(a)$, $\Omega(a)$ e $trust(o)$ $\forall o \in O(a)$.

Esempio:
$O(\text{Nome}) = \{\text{Paolo Rossi}, \text{Paolo R.}, \text{rossi}\}$
$O(\text{Età}) = \{34_1, 34_2, 22\}$ $O(\text{Lingua}) = \{\text{it}\}$

$\Omega(\text{Nome}) = \{\text{Paolo Rossi}, \text{Paolo R.}, \text{rossi}, \overline{\omega}\}$ $\Omega(\text{Età}) = \{34, 22, \overline{\omega}\}$
$\Omega(\text{Lingua}) = \{\text{it}, \overline{\omega}\}$

$trust(\text{Paolo Rossi}) = 0.9$ $trust(\text{Paolo R.}) = 0.7$
$trust(\text{rosso}) = 0.5$ $trust(34_1) = 0.8$ $trust(34_2) = 0.6$ $trust(33) = 0.7$
$trust(33) = 0.7$ $trust(\text{it}) = 0.9$

Per ricavare la distribuzione di probabilità di un attributo a si è creato un algoritmo del torneo tra tutti i valori in $\Omega(a)$. Alla fine del torneo avremo per ogni $\omega \in \Omega(a)$ un punteggio $score(\omega, a)$ che rappresenta l'attendibilità che diamo a ω per a.

Dopo queste prime osservazioni definiamo nel dettaglio l'algoritmo del torneo tenendo conto delle considerazioni precedentemente esposte.

Siccome possono esserci per ogni $\omega \in \Omega(a)$ più osservazioni, è necessario definire la fiducia che diamo ai valori nello spazio campionario $trust(\omega, a)$ (questa definizione è un'altra rispetto a $trust(o)$ data per le osservazioni). Si è scelto di calcolare $trust(\omega, a)$ con il Teorema della probabilità totali. Tale teorema ci garantisce che al crescere del numero di osservazioni la fiducia risulta crescente, ed il valore di fiducia risultante si mantiene nell'intervallo $[0, 1]$.

$$trust(\omega, a) = P(\bigcup_{\forall o \in O(a) | o = \omega} trust(o)) \tag{4.1}$$

Per effettuare il confronto tra i valori nello spazio campionario ci serve una funzione di confronto. Poiché ci interessa far emergere i valori ritenuti i più completi, abbiamo scelto di usare una funzione di similarità, in modo che emergano i valori che presentano meno differenze tra di loro. Tale funzione vale 1 nel caso i valori confrontati siano uguali, mentre tende a 0 tanto più essi sono differenti.

$$sim : \Omega(a) \times \Omega(a) \to [0, 1] \tag{4.2}$$

Siccome $\overline{\omega}$ è un valore incognito non possiamo calcolare $sim(\omega, \overline{\omega})$, ma possiamo assumere che converga ad un determinato valore quando confrontato con gli altri $\omega \in \Omega(a) | \omega \neq \overline{\omega}$.

Ora facciamo un'assunzione pessimistica e consideriamo il valore incognito $\overline{\omega}$ come il più rappresentativo del valore reale che stiamo cercando, ne deriva che tutti gli altri elementi in $\Omega(a)$ gli saranno molto simili. Per $\overline{\omega}$ assumiamo quindi che quando viene confrontato con gli altri valori in $\Omega(a)$, la similarità tende a 1 da sinistra e che la fiducia che gli diamo tende anch'essa a 1 da sinistra: $\forall \omega \in \Omega(a) | \omega \neq \overline{\omega}\ sim(\omega, \overline{\omega}) \to 1^-$ e $trust(\overline{\omega}) \to 1^-$.

Ora possiamo definire la funzione $score(\omega, a)$ per stendere la classifica tra i valori nello spazio campionario. Si è scelto di calcolare il punteggio di ω come la somma delle sue similarità con tutti gli altri valori nello spazio

campionario (escluso se stesso) pesandolo con la fiducia.

$$score(\omega, a) = \sum_{\omega' \in \Omega(a) | \omega' \neq \omega} sim(\omega, \omega') trust(\omega, a) \tag{4.3}$$

Ora che sappiamo come calcolare $score(\omega, a)$ $\forall \omega \in \Omega(a)$, possiamo definire $P(a_\omega)$ come la probabilità che a sia uguale a ω. La distribuzione di probabilità si calcola normalizzando i punteggi ottenuti con la somma di tutti i punteggi risultanti a fine torneo.

$$P(a_\omega) = \frac{score(\omega, a)}{\sum_{\omega' \in \Omega(a)} score(\omega', a)} \tag{4.4}$$

Per ogni attributo di cui vogliamo ricavare la distribuzione di probabilità sarà quindi necessario definire un'opportuna funzione di similarità $sim(\omega, \omega')$ per confrontare i valori nello spazio campionario. I valori di fiducia $trust(o)$ delle osservazioni sono scelti a priori, in base alla fiducia che diamo alla fonte da cui quell'osservazione è stata recuperata.

L'assunzione pessimistica che $\overline{\omega}$ sia molto rappresentativo del valore reale ricercato deriva dal fatto che disponendo di pochi campioni per ricavare la distribuzione di probabilità a posteriori degli attributi non è possibile utilizzare un metodo statistico ritenuto attendibile. Questo ci permette di rendere più realistica la distribuzione di probabilità ricavata a posteriori su un numero esiguo di osservazioni.

È ragionevole pensare che $P(a_{\overline{\omega}})$ sia vero nel caso in cui disponiamo di poche osservazioni poco fidate, mentre $P(a_{\overline{\omega}})$ diminuisce quando abbiamo molte osservazioni disponibili di cui ci fidiamo.

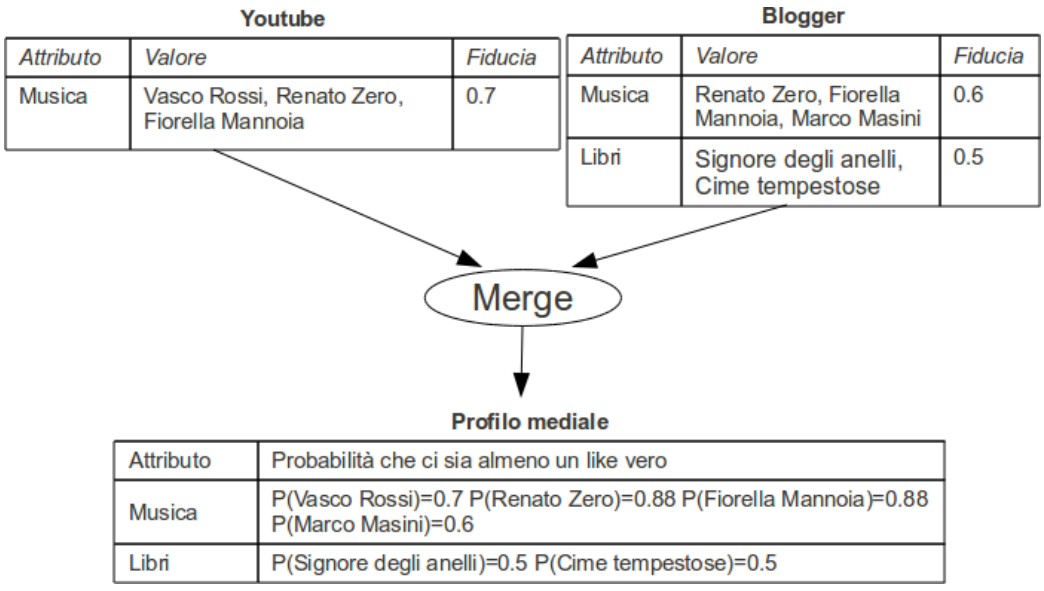
Esempi variazione di $P(a_{\overline{\omega}})$ in funzione delle osservazioni disponibili
$O(Name) = \{\text{Susa}\}$ $P(a_{\overline{\omega}}) = 0.7$ $P(\text{Susan}) = 0.3$
$O(Name) = \{\text{Susan}, \text{Susan Agital}\}$ $P(a_{\overline{\omega}}) = 0.5$ $P(\text{Susan}) = 0.2$
$P(\text{Susan Agital}) = 0.3$
$O(Name) = \{\text{Susan}, \text{Susan Agital}, \text{Susanne}\}$ $P(a_{\overline{\omega}}) = 0.44$ $P(\text{Susan}) = 0.15$
$P(\text{Susan Agital}) = 0.4$ $P(\text{Susanne}) = 0.1$

4.2.2 Attributo Multiplo

Per gli attributi multipli siamo interessati a stabilire per ogni libro/musica/ecc per cui l'utente abbia espresso una preferenza quanto questa sia attendibile. Pertanto non avremo come risultato una distribuzione di probabilità che stabilisce qual è il libro/musica/ecc preferito dall'utente, ma avremo un insieme di eventi che ci dice con che probabilità ogni singola preferenza è vera.

Nella figura 4.2 portiamo come esempio il merge degli attributi multipli Musica Preferita e Libri Preferiti. Come possiamo vedere per la Musica Preferita, ci sono due autori che sono stati elencati su entrambi gli SNS usati dall'utente, mentre gli altri due sono citati su due fonti distinte. Questo ci porta ragionevolmente a pensare che se un utente ripete più spesso un parere favorevole verso un autore su diversi SNS, la probabilità che questo parere sia vero deve aumentare. I pareri espressi su una singola fonte mantengono invece una probabilità di veridicità pari a quella della fonte da cui sono stati osservati.

Figura 4.2: Esempio merge di attributi multipli



Siamo quindi interessati a determinare per il profilo mediale dell'utente l'insieme di preferenze espresse in un attributo a e la probabilità che queste siano vere.

Definiamo per un attributo multiplo a l'insieme delle preferenze del profilo mediale $\Omega(a)$ come l'insieme di tutti i valori distinti delle sue osservazioni $O(a)$. Per ogni preferenza osservata in $o \in O(a)$ abbiamo associato l'evento:

$$E_{o,a} = \text{“la preferenza } o \in O(a) \text{ di } a \text{ è attendibile”}$$

La probabilità dell'evento $E_{o,a}$ è uguale alla fiducia dell'osservazione.

$$P(E_{o,a}) = trust(o) | o \in O(a) \tag{4.5}$$

Per il profilo mediale siamo interessati a calcolare per ogni preferenza espressa $\omega \in \Omega(a)$ la probabilità che ci sia almeno una preferenza vera tra le sue osservazioni, per farlo definiamo l'evento:

$$E_{\omega,a} = \text{“per il parere } \omega \in \Omega(a) \text{ di } a \text{ c'è almeno un parere vero tra quelli espressi nelle sue osservazioni”}$$

Per calcolare $P(E_{\omega,a})$ utilizziamo sempre il Teorema della probabilità totali, che ci garantisce che al crescere del numero di pareri espressi la probabilità del parere verso un determinato libro/film/ecc risulta crescente.

$$P(E_{\omega,a}) = P(\bigcup_{\forall o \in O(a) | \omega = o} trust(o)) \tag{4.6}$$

Per potere utilizzare questo teorema assumiamo che gli eventi $E_{o,a}$ siano stocasticamente indipendenti tra di loro. Per esempio supponiamo che un utente abbia espresso un parere favorevole verso “Vasco Rossi” sul proprio profilo virtuale di Youtube e uno sul suo profilo virtuale di Blogger. Se usiamo il teorema delle probabilità totali per trovare la probabilità che almeno uno dei due sia vero dobbiamo calcolare $P(Vasco|Youtube \bigcup Vasco|Blogger) = P(Vasco|Youtube) + P(Vasco|Blogger) - P(Vasco|Youtube \bigcap Vasco|Blogger)$. Le prime due probabilità le abbiamo in quanto abbiamo scelto che siano uguali alla fiducia che

diamo all'osservazione, ma per l'intersezione delle due dobbiamo calcolare $P(Vasco|Youtube \bigcap Vasco|Blogger) = P(Vasco|Youtube)P(Vasco|Blogger; Youtube)$. Se consideriamo questi eventi stocasticamente indipendenti allora possiamo calcolarle questo valore come il prodotto delle due probabilità $P(Vasco|Youtube \bigcap Vasco|Blogger) = P(Vasco|Youtube)P(Vasco|Blogger)$.

Con tale assunzione facciamo si che i parerei espressi siano indipendenti dalle fonti ma dipendenti solo dall'utente in quanto è lui che in tempi e modalità diverse compila i propri profili virtuali sui diversi SNS. Tale assunzione può portarci a sovrastimare alcuni valori, ma per questo primo lavoro basterà tale definizione.

Con questo metodo non avremo come risultato una distribuzione di probabilità, ma un insieme di eventi $E_{\omega,a}$ che ci dice per ogni parere espresso nel profilo mediale la probabilità che sia vero su almeno un profilo virtuale.

Capitolo 5

Analisi sperimentale

In questo capitolo analizzeremo alcuni dati estratti dall'aggregatore FriendFeed. Per capire quali e quanti SNS vengono utilizzati dagli utenti di questo aggregatore. Successivamente analizzeremo i dati rilevabili dai profili virtuali dei vari SNS per stabilire quali usare per ricostruire i profili mediali.

5.1 FriendFeed

FriendFeed è un SNS che consente l'aggregazione in tempo reale degli aggiornamenti provenienti da altri SNS e qualsiasi altro servizio Web che renda disponibili i propri contenuti tramite feed RSS o Atom.

FriendFeed accede agli altri SNS e alle piattaforme Blog più diffuse attraverso le rispettive API, è questo il caso di Twitter, Facebook, Blogger, Last-fm, Flickr e di numerosi altri servizi i cui aggiornamenti vengono recuperati identificandosi come titolari del contenuto.

I servizi offerti da FriendFeed permettono all'utente la creazione di un unico canale di informazioni che riunisce le molteplici attività dell'utente in rete. Inoltre FriendFeed dispone di un servizio autonomo per pubblicare contenuti, commentare i post di altri utenti e creare una rete sociale con gli altri utenti iscritti al sito.

I servizi registrati dagli utenti FriendFeed sono utili per ricostruire il profilo mediale di un utente tramite i suoi profili virtuali pubblicati in rete.

Aggregare per ogni singolo utente del Web i servizi che adopera è un'operazione difficile, in quanto riconoscere che a usufruire di determinati servizi in rete sia lo stesso individuo è un'operazione complessa. Basti pensare che spesso i servizi sono utilizzati in forma anonima, o nel caso non sia così rimane comunque da gestire il problema delle omonimie tra individui. I servizi aggregati dagli utenti FriendFeed ci risolvono in parte questo problema, in quanto sono gli stessi utenti che creano un indice dei servizi che essi utilizzano. Di fatto non tutti i servizi registrati ci offrono la garanzia che l'utente che li gestisce sia lo stesso utente FriendFeed che li ha aggregati e bisogna quindi adoperare un criterio per filtrare i servizi di cui l'utente potrebbe non essere titolare.

5.2 Statistiche sui servizi aggregati

Per questo lavoro si è partiti da un campione di 497.806 utenti di FriendFeed, per i quali disponiamo di un totale di 1.293.482 servizi registrati.

Come prima analisi osserviamo i servizi registrati per scoprire con che frequenza determinati servizi/SNS vengano registrati dagli utenti. La tabella 5.1 mostra per ogni tipo di servizio registrabile su FriendFeed il numero di occorrenze con cui si presenta nel campione e con che percentuale concorre a formare il campione totale. Notiamo subito che la maggior parte dei servizi registrati sono Blog, a seguire gli SNS più noti. Una seconda analisi è stata fatta sul campione di servizi registrati come Blog. La tabella 5.2 mostra per ogni Blog provider con che frequenza i Blog sono gestiti nel campione. Come è possibile vedere la maggior parte dei Blog è gestita da Blogspot, ma nei risultati che sono stati tagliati è presente anche una lunga lista di Blog autonomi che sono gestiti senza passare tramite un Blog provider.

Idealmente sarebbe opportuno estrarre informazioni da ogni singolo servizio registrato per aiutarci nella ricostruzione del profilo mediale, tuttavia non tutte queste fonti sono complete o utili a tale scopo, pertanto sceglieremo un inseme di servizi e lavoreremo specificatamente su quelli in modo da

ottenere con precisione un insieme di informazioni che siano rilevanti per il nostro scopo.

I servizi scelti da analizzare sono Blogspot, Facebook, Flickr, Lastfm, Stumbleupon, Twitter, Youtube e Brightkite. Questi servizi sono stati scelti tenendo conto di aspetti sia quantitativi che qualitativi. Questi SNS costituiscono una parte consistente del campione e permettono di identificare facilmente il profilo virtuale dell'utente che li gestisce. Inoltre i profili sono ben strutturati e potenzialmente ricchi di informazione.

5.3 Statistiche sui profili virtuali e mediali

Per ricostruire il profilo mediale di un individuo dobbiamo prima scegliere quali attributi di quest'ultimo vogliamo ricostruire. Prima di questo però è opportuno indagare quali attributi sono disponibili nei profili virtuali del campione, in modo da scegliere per il profilo mediale un insieme di attributi che sia il più possibile condiviso tra i rispettivi profili virtuali.

Per ogni SNS gestito è stata effettuata una statistica che rileva con che frequenza un determinato attributo del profilo virtuale occorre. Successivamente si è ragionato sul significato/contenuto degli attributi. Nelle tabelle dalla 5.4 alla 5.11 sono stati riportati i risultati di questa ultima analisi (alcuni attributi sono stati esclusi in quanto superflui).

Come possiamo vedere i profili virtuali ricavabili dagli SNS sono ricchi di informazioni. Gli attributi più comuni tra i vari SNS sono quelli che trattano dati personali quali nome, cognome, nazionalità e residenza. Per ricostruire il profilo mediale ci concentreremo su questi ultimi. Il profilo mediale che ricostruiremo sarà formato dai seguenti attributi: nome, lingua, età, stato civile, lingua, email, lavoro, residenza, libri preferiti, musica preferita, film preferiti e interessi.

Per ottenere un profilo mediale di un utente FriendFeed dobbiamo: (I) scaricare i profili virtuali degli SNS usati dall'utente; (II) estrarre dai profili virtuali gli attributi scelti per il profilo mediale; (III) ricostruire gli attributi del profilo mediale tenendo conto dei valori che assumono tra i diversi profili virtuali.

Analizziamo ora gli attributi dei profili mediali ottenuti. Come mostra il grafico in figura 5.1 se consideriamo l'attributo Età (queste considerazioni valgono anche per gli altri attributi) otteniamo una maggior frequenza di quest'ultimo nei profili mediali rispetto alla frequenza con cui l'attributo occorrerebbe se consideriamo solo i profili virtuali ricavati dagli altri SNS. Questo beneficio deriva dal processo di integrazione che recupera informazioni da più fonti distinte. Nella tabella 5.3 riportiamo con che frequenza siamo riusciti a ricostruire gli attributi dei profili mediali degli utenti FriendFeed.

Figura 5.1: Frequenza dell'attributo età nel profilo mediale rispetto agli altri profili virtuali

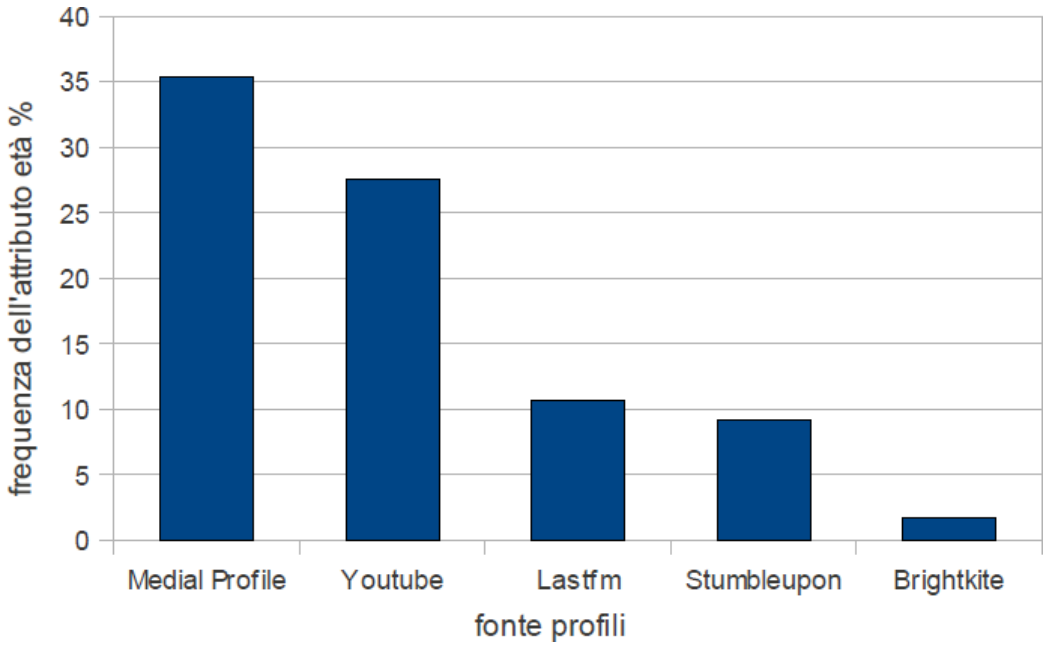


Tabella 5.1: Frequenza servizi registrati

Servizio	**Numero servizi registrati**	**% Servizi registrati**
blog	264.916	20.48
twitter	229.339	17.73
flickr	82.892	6.41
youtube	80.915	6.26
feed	64.875	5.02
delicious	64.485	4.99
facebook	53.191	4.11
googlereader	46.626	3.60
googletalk	45.452	3.51
lastfm	42.122	3.26
linkedin	41.304	3.19
digg	40.481	3.13
stumbleupon	30.229	2.34
picasa	28.620	2.21
tumblr	20.264	1.57
pandora	11.522	0.89
amazon	11.012	0.85
vimeo	10.334	0.80
brightkite	7774	0.60
reddit	7369	0.57
...	...	...
...	...	...
...	...	...
...	...	...
smugmug	1231	0.10
fotolog	844	0.07
baidu	630	0.05
meneame	607	0.05
photobucket	478	0.04
polyvore	454	0.04
skyrock	341	0.03
email	318	0.02
smotri	222	0.02
tipjoy	137	0.01
Totale 61	Totale 1.293.482	Totale 100%

Tabella 5.2: Provider per i blog rilevati

Blog provider	**Numero blog registrati**	**% blog registrati**
blogspot.com	52075	19.66
wordpress.com	14446	5.45
twitter.com*	3495	1.32
google.com	2953	1.11
yahoo.com	2511	0.95
feedage.com	2504	0.95
amazon.com*	2051	0.77
tumblr.com	1589	0.60
blogfa.com	1566	0.59
livejournal.com	1483	0.56
typepad.com	1346	0.51
easyjournal.com	1314	0.50
bloglines.com	1219	0.46
squidoo.com	868	0.33
insanejournal.com	1150	0.43
live.com	1074	0.41
easydigitalsales.com	990	0.37
fc2.com	923	0.35
squidoo.com	868	0.33
...	...	...
...	...	...
...	...	...
...	...	...

* Con FriendFeed è possibile registrare come Blog una qualunque fonte di feed

Tabella 5.3: Frequenza attributi dei profili mediali

Dato	**Frequenza**
FullName	97.16
Age	35.32
Gender	57.57
Language	87.24
Relationship status	14.01
Email	3.93
Locality of residence	25.93
Region/State of residence	12.47
Country of residence	26.92
Occupation	17.56
Favorit music	5.14
Favorit movies	4.71
Favorit books	4.56
Interests	7.47

Tabella 5.4: Attributi dei profili di Blogger

Dato	**Descrizione**	**Frequenza**
viste profilo	contatore delle visite al profilo	100.00
su blogger da	da quanto tempo è iscritto	100.00
nazione	nome del paese/nazione	72.77
genere	maschio o femmina	67.91
località	nome città in cui vive	62.06
settore	ambito lavorativo	53.51
regione	nome regione in cui vive	53.35
professione	lavoro svolto	41.69
segno zodiacale	segno zodiacale dell'utente	40.74
interessi	interessi dell'utente	39.72
email	contatto email	28.22
musica preferita	elenco delle preferenze musicali	27.73
film preferiti	elenco dei film preferiti	26.53
libri preferiti	elenco dei libri preferiti	25.95

Tabella 5.5: Attributi dei profili di Brightkite

Dato	**Descrizione**	**Frequenza**
nome completo	nome cognome o username	100.00
genere	maschio o femmina	100.00
descrizione	presentazione fornita dall'utente	57.87
età	età dell'utente	50.26

Tabella 5.6: Attributi dei profili di Facebook

Dato	**Descrizione**	**Frequenza**
nome completo	nome completo dell'utente	100.00
cognome	cognome dell'utente	99.35
lingua e nazionalità	lingua e nazionalità dell'utente in ISO code	99.35
nome	nome dell'utente	99.35
genere	maschio o femmina	78.62
soprannome	soprannome dell'utente	10.52
sito Web	link al proprio sito esterno	0.63
username	username su facebook	0.39
città natale	nome città natale	0.03
bio	descrizione dell'utente	0.02
residenza attuale	dove si trova ora l'utente	0.02

Tabella 5.7: Attributi dei profili di Flickr

Dato	Descrizione	Frequenza
registrato il	mese e anno in cui l'utente si è iscritto	98.00*
sono	maschio o femmina e stato civile	61.12
sito Web	sito esterno dell'utente	48.26
nazione	nome del paese/nazione dell'utente	35.59
città	nome della città in cui vive	34.65
città natale	nome città natale dell'utente	30.48
lavoro	lavoro svolto dall'utente	29.84
email	contato email	4.13
msn messenger	contatto msn	1.11
yahoo im	contatto yahoo	0.89
aim	contatto aim	0.82
icq	contatto icq	0.33

* il 2% manca in quanto con le impostazioni per la privacy è possibile bloccare la pubblicazione dia alcuni dati

Tabella 5.8: Attributi dei profili di Lastfm

Dato	Descrizione	Frequenza
username	username dell'utente	100.00
genere	maschio o femmina	98.14
stato	ISO country code della nazione in cui vive	75.66
nome	nome e cognome (non sempre attendibile)	63.44
età	età dell'utente	57.61

Tabella 5.9: Attributi dei profili di Stumbleupon

Dato	Descrizione	Frequenza
nome	nome cognome dell'utente oppure username	93.61
stato	paese/nazione in cui vive	93.61
genere	maschio o femmina	87.33
età	età dell'utente	72.81
regione	nome della regione in cui vive	65.38
città	nome città di residenza	52.42

Tabella 5.10: Attributi dei profili di Twitter

Dato	**Descrizione**	**Frequenza**
username	username dell'utente	100.00
contatore amici	numero amici dell'utente	100.00
verificato	l'identità dell'utente è stata verificata	100.00
lingua	lingua dell'individuo	100.00
nome	nome e cognome (non sempre attendibile)	100.00
ultimo feed	contenuto dell'ultimo tweet pubblicato	93.08
residenza	dove vive l'utente	84.63
descrizione	presentazione dell'utente	78.18

Tabella 5.11: Attributi dei profili di Youtube

Dato	**Descrizione**	**Frequenza**
username	username su youtube	100.00
statistics	alcune statistiche sull'attività dell'utente	100.00
location	dove abita l'utente	96.65
gender	maschio o femmina	92.97
age	età	79.43
firstName	nome	48.02
lastName	cognome	29.57
description	presentazione dell'utente	26.81
hometown	città natale	25.79
occupation	lavoro	18.93
relationship	stato civile	16.39
company	azienda per cui lavora	13.61
hobbies	passatempi	13.16
music	musica preferita	8.91
school	titolo di studio	8.10
movies	film preferiti	7.81
books	libri preferiti	7.54

Capitolo 6

Valutazioni

In questo capitolo valuteremo i metodi possibili per studiare la precisione del nostro algoritmo per la ricostruzione dei profili mediali e discuteremo i risultati ottenuti.

6.1 Metodi di valutazione

Per valutare un sistema come questo è necessario disporre di un data set con i valori reali dei dati da integrare. Una volta applicato il nostro algoritmo di integrazione possiamo calcolare la precisione con la quale riconosce i valori reali degli attributi dei profili mediali.

Nel §5.2 di [9] viene presentato un esempio per testare questo genere di sistemi: per una dominio particolare come le quotazioni del NASDAQ esiste una fonte autoritaria che si può usare per stabilire il valore reale dei dati che vengono integrati da altre fonti non autoritarie. Tuttavia per il nostro algoritmo non disponiamo di un simile scenario in quanto non esiste "un'anagrafe digitale" che ci fornisca i valori reali per gli attributi dei profili mediali, permettendoci quindi ti misurare la precisione dell'algoritmo.

Un'altra possibilità è presentata nel §5.1 di [9] che propone di testare l'algoritmo su un data set generato artificialmente con un motore casuale. Nel nostro caso dovremmo quindi generare un insieme di profili artificiali e per ognuno di essi un insieme di profili virtuali. Questa prima fase può essere

realizzata usando le statistiche presentate nei capitoli §5.2 e §5.3 di questa tesi per determinare la probabilità che un utente adoperi un determinato servizio e le probabilità con cui vi pubblichi determinati dati. La difficoltà reale del generare un data set artificiale consiste nel formulare un insieme di regole che emulino il comportamento degli utenti quando essi compilano i dati per i propri profili virtuali. I dati personali come il nome vengono spesso perturbati in nickname o abbreviazioni, mentre dati complessi come la città di residenza sono spesso scritti in testo libero senza un'adeguata strutturazione rendendo l'estrazione di tali valori soggetta ad errori. In fine bisogna stimare una quantità di profili virtuali errati da introdurre per alcuni utenti, in quanto come esposto in §3.1.3 c'è il rischio che non tutti i profili virtuali usati per ricostruire un profilo mediale siano stati scelti correttamente.

La scelta di generare un data set artificiale su cui effettuare il test viene anch'essa scartata, in quanto essendo un'operazione che difficilmente genererà un data set attendibile non ci permette di misurare in modo corretto la precisione con cui il nostro sistema ricostruisce i profili mediali.

6.2 Valutazione dei risultati ottenuti

Per questo primo lavoro non c'è stato abbastanza tempo a disposizione per riuscire ad effettuare un test che dia una misura quantitativa della precisione dell'algoritmo implementato. Le osservazioni che faremo sono qualitative e ristrette a un piccolo insieme di profili mediali analizzati manualmente. Tale verifica "umana" è stata effettuata in quanto possiamo ragionevolmente assumere che una persona sia capace di verificare che il profilo mediale ricavato sia corretto rispetto ai profili virtuali da cui è stato generato.

Per eseguire il nostro algoritmo sono stati scelti a priori i valori di fiducia da dare agli attributi estratti dai profili virtuali e per farlo si è tenuto conto sia dell'uso che viene fatto di determinati SNS da parte degli utenti sia di come questi attributi sono stati estratti. Per esempio gli attributi nome e cognome estratti da Facebook avranno una fiducia maggiore rispetto a quelli degli altri SNS, in quanto questo particolare SNS è fatto apposta per farsi trovare e mantenersi in contatto con i propri amici (su Facebook è molto di-

sincentivato l'anonimato), mentre se estraiamo nome e cognome da altri SNS dobbiamo spesso derivarli da un singolo dato che potrebbe addirittura essere un nickname. Nello stabilire questi valori di fiducia si è tenuto molto conto di come i Data Extractor hanno dovuto lavorare per ricavare i singoli attributi e su come sono strutturati i Crawl da cui andavano estratti. I risultati ottenuti variano molto a seconda degli attributi che andiamo a integrare.

Attributi semplici quali lingua, età e genere sembra che vengano integrati meglio degli altri. Questo è dovuto al fatto che sono molto facili da estrarre e il loro dominio è molto ristretto rispetto agli altri attributi (il dominio di genere in particolare è formato solo dai valori "maschio" e "femmina"). Per quanto riguarda il nome e cognome di una persona si sono visti risultati positivi, anche se in alcuni casi sono presenti errori. I dati di residenza strutturati in città, regione e stato risultano essere quelli integrati peggio, in quanto la loro estrazione sembra essere la più soggetta a errori, inoltre rispetto agli altri attributi sono meno frequenti, dandoci così un insieme ristretto di osservazioni su cui fare operare l'algoritmo.

Gli attributi multipli di preferenze musicali, film e libri anche se poco frequenti sono stati integrati abbastanza bene. Spesso gli stessi gestori di SNS invitano ad elencare le proprie preferenze in forma di elenco e questo ha fatto si che la fase di estrazione sia stata eseguita molto bene, rilevando varie ripetizioni di preferenze espresse dal medesimo utente tra gli SNS che egli usa. Gli errori si presentano nel caso in cui un utente esprima il proprio parere in modo diverso sui vari SNS. Per esempio esprimendo parere favorevole per Vasco Rossi su un SNS e su un altro SNS il parere favorevole per Vasco, il sistema per com'è attualmente implementato li vede come due pareri favorevoli distinti invece che uno solo. Per migliorare ulteriormente l'integrazione degli attributi multipli bisognerebbe risolvere queste ambiguità sui dati estratti.

Capitolo 7

Conclusioni e sviluppi futuri

Con questo studio abbiamo iniziato a fronteggiare il problema di ricostruire il profilo mediale degli utenti degli SNS mediante i profili virtuali che questi generano.

I problemi rilevati durante questo lavoro sono molteplici. Come abbiamo potuto esporre aggregare i profili virtuali degli individui non è un'operazione immediata e priva di errori, inoltre i dati ricavabili dai profili virtuali sono molto scarsi e di difficile estrazione. Questo fa si che integrare i dati con tecniche statistiche possa essere ritenuto poco attendibile anche se non impossibile.

Il tallone di Achille di questo studio è che non è risultato possibile formulare un test per il sistema sviluppato in modo da misurare la precisione con la quale vengono ricostruiti i profili mediali, lasciandoci così senza una metrica per misurare la qualità dell'algoritmo proposto per ricostruire i profili mediali.

Le migliorie apportatili risultano essere molteplici e sono sia quantitative che qualitative. Dal punto di vista quantitativo abbiamo bisogno di molti dati per ricostruire il profilo mediale, ed è quindi necessario estendere il sistema in modo tale che esso ricavi i profili virtuali da più fonti distinte possibili. Dal punto di vista qualitativo bisogna migliorare l'estrazione degli attributi dai profili virtuali in modo da portarli ad una forma intermedia il più strutturata possibile. L'algoritmo proposto lavora bene sino a che si dispongono

di pochi dati, in quanto per definizione un algoritmo del torneo presenta una complessità quadratica. Nel nostro caso abbino gestito otto SNS diversi, avendo spesso solo quattro valori per integrare ogni singolo attributo, rendendo quindi la complessità dell'algoritmo sostenibile, ma nel caso si riuscisse a disporre di una maggiore quantità di dati risulterà necessario riformulare l'algoritmo per questo nuovo contesto.

Nonostante tutte le problematiche riscontrate siamo stati in grado di sviluppare un sistema che ricostruisce i profili mediali degli utenti di FriendFeed, dandoci una visione più ricca e dettagliata di questi utenti. Queste informazioni aggregate sugli individui possono risultare molto utili in campi come il marketing e indagini politiche, in quanto siamo in grado di profilare molti individui senza doverli interpellare di persona uno ad uno.

Bibliografia

[1] Roger E. Bohn James E. Short: "How Much Information? 2009 Report on American Consumers" Published in: Global Information Industry Center

[2] The Economist "A world of connections A special report on social networking" Published in: January 30th 2010 January 30th 2010 University of California, San Diego on december 2009 Conference on Semantics in Text Processing. from Inaccurate Data Sources" Published in: In WWW2010 Conference - poster track, 2010.

[3] Fabio Celli, F. Marta L. Di Lascio, Matteo Magnani, Barbara Pacelli, and Luca Rossi: "Social Network Data and Practices: the case of Friendfeed" Published in: In International Conference on Social Computing, Behavioral Modeling and Prediction SBP 2010

[4] Cosenza, V.: Osservatorio Facebook http://www.vincos.it/osservatorio-facebook, retrieved on August 31, 2009.

[5] Danah Boyd: Taken Out of Context: "American Teen Sociality in Networked Publics" PhD thesis, University of California-Berkeley, School of Information 2008

[6] X. Yin, J. Han, and P. S. Yu. "Truth discovery with multiple conflicting information providers on the web" Published in: IEEE Trans. Knowl. Data Eng. 2008.

[7] M. Wu and A. Marian. "Corroborating answers from multiple web sources" Published in: WebDB, 2007

[8] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. "Corroborating information

[9] Lorenzo Blanco, Valter Crescenzi, Paolo Merialdo, and Paolo Papotti: "Probabilistic Models to Reconcile Complex Data from disagreeing views" Published in: Proc. WSDM, New York, USA, 2010.

[10] Elmagarmid, Ahmed; Panagiotis G. Ipeirotis, Vassilios Verykios (January 2007): "Duplicate Record Detection: A Survey" Published in: IEEE Transactions on Knowledge and Data Engineering 2009

[11] Royal Military Academy: "An Introduction to Data Fusion" retrieved on: http://www.sic.rma.ac.be/Research/Fusion/Intro/content.html

[12] Alon Y. Halevy: "Answering queries using views: A survey" pp. 270–294. Published in: The VLDB Journal 2009

[13] Johan Bos: "Wide-Coverage Semantic Analysis with Boxer" Published in: Proceeding STEP '08 Proceedings of the 2008

[14] Dave Balmain, Ferret search library http://ferret.davebalmain.com/