

POLITECNICO DI MILANO

Polo Regionale di Como

Corso di Laurea Magistrale in Ingegneria Informatica



WikiFinder: un *Game with a Purpose* per lo studio della popolarità di Linked Data

Tesi di Laurea Magistrale

Autore: Roberto Gualandris

Matricola: 787170

Relatore: Emanuele Della Valle

Correlatore: Irene Celino

Anno Accademico 2012/2013

Sommario

INDICE DELLE FIGURE	3
1. INTRODUZIONE	5
1.1 DBpedia e il Semantic Web	6
1.2 Struttura tesi	8
2. STATO DELL' ARTE	9
2.1 I GWAP secondo Luis Von Ahn	9
2.2 Diversi modelli di GWAP	10
2.3 Esempi di GWAP.....	12
2.4 GWAP per classificare Linked Data	17
3. PROGETTAZIONE DELLA SOLUZIONE	21
3.1 Struttura del gioco	21
3.2 L'ispirazione.....	22
3.3 Le alternative.....	24
4. WIKIFINDER	29
4.1 Estrazione dati da Wikipedia e creazione griglie.....	29
4.2 Architettura Client-Server	35
4.3 Funzionamento	38
5. VALUTAZIONE DI WIKIFINDER	42
5.1 Metriche di valutazione	42
5.2 Risultati della valutazione	45
6. CONCLUSIONI	53
6.1 Sviluppi futuri	54
BIBLIOGRAFIA	55

INDICE DELLE FIGURE

Fig. 1.1 - Un esempio di relazioni tra risorse di DBpedia	7
Fig. 2.1 - Un schermata di gioco di ESP game	13
Fig. 2.2 -Una schermata del gioco Peekaboom.....	14
Fig. 2.3 - Una schermata del gioco Verbosity.....	15
Fig. 2.4 - Una schermata di gioco di Duolingo.....	16
Fig. 2.5 - La nuvola di Linked Open Data osservata a fine 2011 [10]	18
Fig. 2.6 - Schermata di gioco di Who Knows?	19
Fig. 2.7 - Una schermata di gioco di Better Relations	20
Fig. 3.1 - Una schermata di gioco di Ruzzle	23
Fig. 3.2 - Esempio di griglia 4x4, la lettera R indica la risorsa, la P il predicato.....	25
Fig. 3.3 - Esempio di griglia 3x3, la lettera S indica il soggetto, la P il predicato e la O l'oggetto	27
Fig. 3.4 - Esempio di griglia 3x5, la lettera S indica il soggetto, la P il predicato e la O l'oggetto	28
Fig. 4.1 - Esempio di interrogazione SPARQL	29
Fig. 4.2 - Esempio di percorsi possibili per ospitare le triple estratte da DBpedia	30
Fig. 4.3 - Template query Sparql: caso in cui il soggetto è già presente in griglia	32
Fig. 4.4 - Template query Sparql: caso in cui il predicato è già presente in griglia.....	32
Fig. 4.5 - Template query Sparql: caso in cui l'oggetto è già presente in griglia.....	32
Fig. 4.6 - Template query Sparql: caso in cui sia il predicato che l'oggetto sono già presenti in griglia	33
Fig. 4.7 - Template query Sparql: caso in cui sia il soggetto che il predicato sono già presenti in griglia	33
Fig. 4.8 - Template query Sparql: caso in cui sia il soggetto che l'oggetto sono già contenuti in griglia	33
Fig. 4.9 - Algoritmo in pseudocodice.....	34
Fig. 4.10 - Schema generale dell'architettura	35

Fig. 4.11 - Servizio di invio griglia all'interfaccia	36
Fig. 4.12 - Servizio di invio dati partita al server	37
Fig. 4.13 - Schermata Login	39
Fig. 4.14 - Schermata Home	39
Fig. 4.15 - Schermata griglia di gioco	40
Fig. 4.16 - Schermata risultato finale	40
Fig. 5.1 - Tabella indice di popolarità triple vere.....	45
Fig. 5.2 - Classifica frequenza triple false	46
Fig. 5.3 - Tabella triple vere individuate nei primi 10 secondi di gioco.....	47
Fig. 5.4 - Tabella triple popolari che sono state selezionate almeno da 5 giocatori distinti.....	50
Fig. 5.5 - Grafico di correlazione tra rank WikiFinder e rank Wikipedia inglese	52
Fig. 5.6 - Grafico di correlazione tra rank WikiFinder e rank Wikipedia italiana.....	52

1. INTRODUZIONE

Al giorno d'oggi, nonostante si pensi che la tecnologia e il progresso nel campo informatico abbiano compiuto passi da gigante, esistono ancora delle attività che, per quanto risultino banali per le capacità umane, sono invece molto complesse da elaborare per normali calcolatori. L'intelligenza artificiale, sebbene abbia raggiunto livelli qualitativi molto alti, non è ancora in grado di simulare molte delle abilità percettive che per la maggior parte degli uomini sono una caratteristica naturale.

Con il termine **Games with a Purpose** (GWAP) si identificano tutti quei giochi che sono stati progettati per un preciso scopo e che coniugano l'attività ludica (il divertimento come obiettivo principale del giocatore) con il raggiungimento di un obiettivo prefissato dagli sviluppatori. La peculiarità sta nel fatto che questo obiettivo è *nascosto* al giocatore il quale, inconsapevolmente, aiuta a risolvere problemi su larga scala. Grazie a questo meccanismo, coinvolgendo un buon numero di persone, è possibile creare un processo di **Computazione Umana** (in inglese, Human Computation) in grado di generare dati dai quali è possibile dedurre alcune evidenze attraverso una serie di analisi statistiche.

Il lavoro presentato in questa tesi è incentrato sullo sviluppo di un gioco, rispondente alle caratteristiche sopraindicate, proposto sotto forma di applicazione per smartphone (Android e IOS), il quale ha permesso di raccogliere interessanti informazioni sul livello di conoscenza e di popolarità di determinate nozioni raccolte all'interno del Web, in particolare da **Wikipedia**.

1.1 DBpedia e il Semantic Web

Per la creazione dell'applicazione proposta, è stato utilizzato il database di DBpedia [1] come fonte di dati della conoscenza umana. DBpedia è un progetto che è nato con l'intento di organizzare e annotare semanticamente l'enorme mole di informazioni strutturate che sono raccolte all'interno delle pagine di Wikipedia (ad es. nei cosiddetti *infobox* che appaiono a destra nelle pagine). Lo strumento più utilizzato per rappresentare queste informazioni nella comunità scientifica del Semantic Web è il linguaggio **RDF** [2] (Resource Description Framework) il quale permette di descrivere qualunque risorsa reperibile sul Web attraverso una definizione di proprietà e valori. Ogni risorsa è identificata da un **URI** (Uniform Resource Identifier), ossia un *nome* che la identifica univocamente, mentre le proprietà sono delle relazioni che legano tra loro risorse e valori e sono anch'esse identificate da URI. Un valore, invece, può essere costituito da un tipo di dato primitivo (stringa, intero, decimale, etc.) oppure dall'URI di una risorsa. L'unità base per rappresentare un'informazione in RDF è lo statement. Uno statement è una tripla quale, **Soggetto – Predicato – Oggetto**, dove il soggetto è una risorsa, il predicato è una proprietà e l'oggetto è un valore (e quindi potenzialmente anche un URI di un'altra risorsa che a sua volta può essere soggetto di un'altro statement).

Questo metodo di rappresentare delle informazioni è molto utile per poter ordinare la mole di dati che ogni giorno viene pubblicata sul Web, ma non aiuta ad indentificare quanto una risorsa sia più o meno conosciuta in termini di popolarità.

L'applicazione sviluppata per questo lavoro di ricerca ha come scopo quello di soddisfare il suddetto requisito, sottoponendo al giocatore informazioni estratte da DBpedia sottoforma di risorse proprietà e valori in maniera disordinata e chiedendogli di ricreare le relazioni esatte secondo la sua conoscenza.

In questo modo è stato possibile verificare il grado di conoscenza e di popolarità di cui godono alcune notizie e informazioni rispetto ad altre.

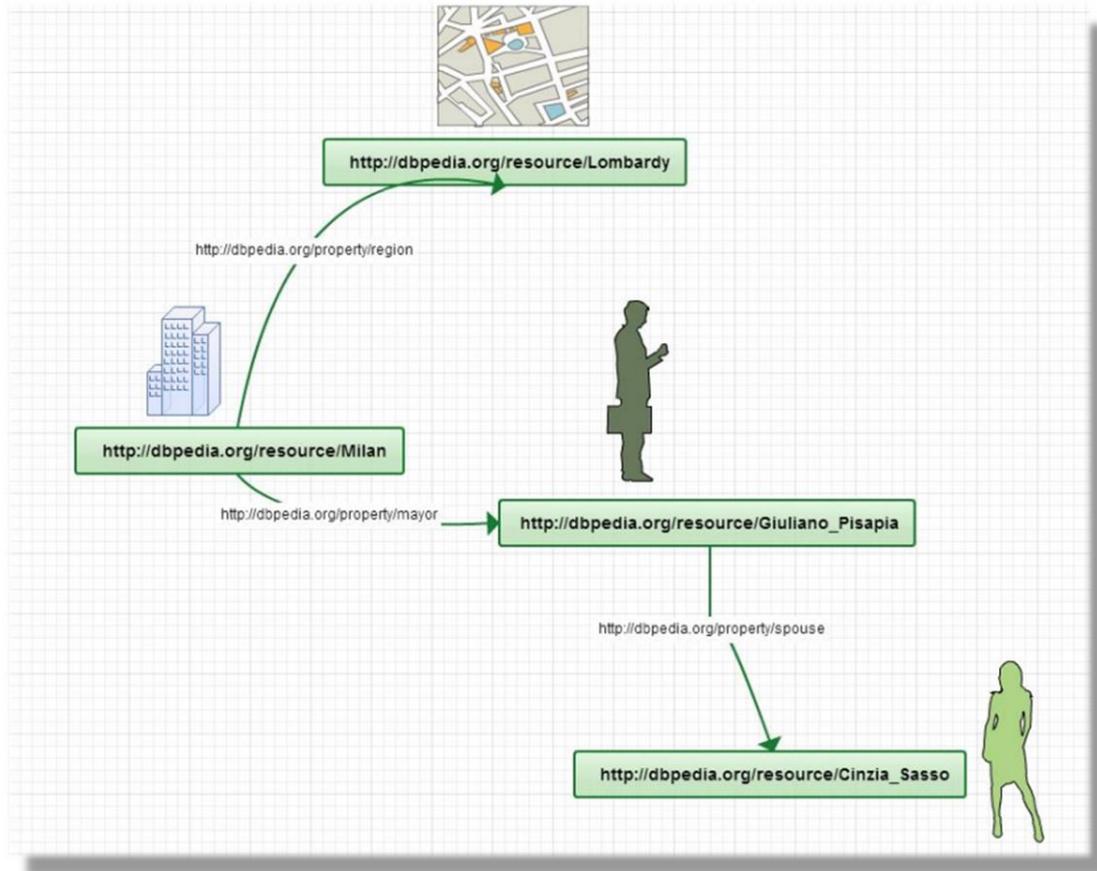


Fig. 1.1 - Un esempio di relazioni tra risorse di DBpedia

1.2 Struttura tesi

L'elaborato è strutturato nel seguente modo:

- Nel secondo capitolo è presentato un quadro generale del mondo dei GWAP. Sono descritti e analizzati alcuni dei principali giochi GWAP con lo scopo di comprendere le loro caratteristiche e funzionalità. Questi ultimi sono stati poi classificati in base al dominio e al metodo utilizzato per raccogliere e studiare le informazioni collezionate dai giocatori. I dati, infatti, sono stati il punto di partenza da cui si è sviluppata l'applicazione.
- Nel terzo si è analizzata la possibilità di realizzazione di un gioco per studiare la popolarità di statement pubblicati sul Web sotto forma di **linked data** (e, in particolare, dati estratti da DbPedia), nucleo centrale del progetto in esame. Sono state scandagliate tutte le caratteristiche dell'applicazione e studiate le alternative più consone per quanto riguarda sia lo sviluppo dell'architettura, sia il game play.
- Il quarto capitolo propone la soluzione migliore per l'implementazione del gioco: sono descritte l'architettura finale, gli algoritmi adottati e spiegato come l'applicazione è in grado di collezionare le informazioni rilevanti e come successivamente vengono immagazzinate.
- Il quinto capitolo riassume i risultati pratici dell'esperimento. In particolare, sono mostrate tutte le valutazioni sui dati collezionati utilizzando metodi di analisi tra i quali l'ALP (Average Lifetime Play), il Throughput, l'indice di popolarità delle triple e delle risorse poste in esame. Queste informazioni sono state confrontate per stilare le opportune considerazioni sulla buona riuscita del gioco.
- Infine, il sesto capitolo, delinea le conclusioni sulle ricerche effettuate e traccia le possibili linee future per un eventuale sviluppo migliorativo.

2. STATO DELL' ARTE

2.1 I GWAP secondo Luis Von Ahn

A coniare il termine di GWAP è stato l'informatico Luis Von Ahn [3], autore di una geniale intuizione. Ogni anno, persone in tutto il mondo, spendono milioni di ore giocando ai videogiochi, sia online tramite il pc di casa, sia su smartphone e tablet. Von Ahn osservò con molta attenzione questo comportamento e pensò di incanalare tutte queste energie e il tempo speso nel gioco per contribuire allo sviluppo di un lavoro utile. Ogni singolo giocatore può essere considerato come un singolo processore capace di elaborare i dati a cui è sottoposto; tanti processori messi in parallelo sono in grado di analizzare un numero maggiore di informazioni e di svolgere delle attività in un tempo sempre minore. Utilizzando questo principio, Von Ahn ipotizzò che, sottoponendo un problema in forma videoludica a milioni di giocatori, questi sarebbero stati in grado di risolverlo in maniera sorprendentemente più veloce ed efficace rispetto a un normale approccio automatico. A differenza dei calcolatori, gli umani hanno bisogno di essere stimolati per entrare a far parte di questa computazione collettiva. I giochi online rappresentano un'attrazione per incoraggiare le persone a partecipare a questo processo. Von Ahn capì dunque che era molto più efficace, in termini di costo/effort, sviluppare un gioco di questo tipo piuttosto che progettare ed elaborare un algoritmo per un calcolatore elettronico.

2.2 Diversi modelli di GWAP

La prima caratteristica su cui Von Ahn incentrò la progettazione dei GWAP fu la capacità di sollecitare lo stimolo alla competizione che generalmente un giocatore abituale possiede in maniera naturale. Questa tipologia di gioco è detta **multiplayer** e consiste appunto, nel mettere due persone a confronto nella stessa partita, allo stesso momento.

I modelli [3] definiti dallo studioso hanno tutti alla base il concetto di multiplayer nel quale i due players giocano contemporaneamente, cercando di fare più punti possibili.

I modelli sono i seguenti:

- **Output Agreement Games:** sono quei giochi in cui due persone vengono scelte casualmente tra tutti i potenziali giocatori. In ogni partita ricevono lo stesso input ed entrambe dovrebbero produrre lo stesso output. Ovviamente i due sfidanti non possono comunicare tra di loro e non possono conoscere il risultato dell'avversario. L'input, ad esempio, può essere un'immagine e l'output una sua descrizione oppure delle parole chiave che la descrivono. Partendo dal fatto che i due giocatori non possono comunicare e non si conoscono, lo scopo del gioco è quello di produrre entrambi più elementi comuni possibili, in modo da raggiungere una sorta di **accordo** implicito. Sicuramente non è possibile garantire che il risultato prodotto sia conforme all'input, ma è possibile prevenire il gioco scorretto stabilendo in anticipo una black list di parole non ammesse, spingendo così l'utente a cercare di concepire output il più possibile relativi all'input.
- **Inversion Problem Games:** in questo modello vi sono sempre due players scelti casualmente ma, a differenza della precedente tipologia, essi hanno due ruoli differenti: in ogni partita uno ha l'incarico di descrivere qualcosa, l'altro ha il

compito di indovinare l'oggetto della descrizione. Il gioco inizia con l'invio da parte del primo giocatore di una descrizione riguardante ad esempio un'immagine attraverso una serie di parole chiave mirate, mentre il secondo deve cercare di indovinare avvicinandosi il più possibile all'elemento dell'immagine in questione. Ovviamente se la descrizione iniziale non è corretta, la seconda persona non potrà mai produrre una risposta ragionevole. Lo scopo del gioco è dunque raggiungere un'intesa, una sorta di **cooperazione sociale** che si viene a creare tra i due players pur non conoscendosi. A differenza degli output agreement games in cui entrambi compiono lo stesso lavoro, in questo schema c'è una rotazione dei ruoli ad ogni round in modo da renderlo bilanciato ed avvincente.

- **Input Agreement Games:** come nel primo modello, anche in questo, due estranei partecipano alla partita, ma con la differenza che ai due viene sottoposto un input che può essere lo stesso o differente. I giocatori devono produrre un output descrivendo nel modo più efficace l'input, cercando di capire se hanno entrambi lo stesso input o meno. Essendo lo scopo del gioco quello di raggiungere un comune accordo, è sconsigliato proporre descrizioni sbagliate al partner e, per scoraggiare entrambi dallo scommettere casualmente sulla soluzione, il sistema infligge pesanti penalizzazioni per soluzioni non corrette.

2.3 Esempi di GWAP

I più famosi GWAP, che hanno riscosso maggior successo e hanno prodotto i risultati migliori, sono riportati di seguito, per avere una panoramica chiara e esaustiva.

ESP Game

Molte applicazioni online, come i motori di ricerca, hanno spesso bisogno di descrizioni accurate per le immagini. Purtroppo, non esistono linee guida sulla procedura da utilizzare per catalogare ed etichettare con un'adeguata descrizione le milioni di immagini presenti sul Web o, tantomeno, un computer è in grado di eseguire una simile mole di lavoro. Le tecniche correnti utilizzate per schedare le immagini, spesso sono inadeguate, poiché assumono che il significato di una figura sia quello espresso nel testo ad essa adiacente che in molti casi non è completo o è fuorviante. L'etichettatura manuale è l'unica soluzione per eseguire un lavoro ben fatto e qualitativamente rilevante. Questa procedura però, può risultare noiosa e comporta un dispendio di energie notevole. ESP game [4] nasce con l'intenzione di risolvere questo problema, traendo vantaggio dalla computazione umana, collezionando i tag delle immagini ad ogni sessione di gioco. All'inizio della partita vengono selezionate casualmente due persone che sono estranee e non sono in grado di comunicare tra di loro. Ad entrambe è sottoposta un'immagine, lo scopo del gioco è descrivere la figura attraverso delle parole significative dette **Tag** che siano comuni per tutti e due gli utenti: ad ogni tag comune il punteggio aumenta. Il sistema ovviamente, previene l'uso di parole ovvie o molto comuni imponendo una lista di parole *tabù* che non è possibile utilizzare, stimolando la fantasia e l'intuizione di ogni giocatore. Le partite consistono in brevi match, poiché è presente un limite massimo di tempo per trovare l'accordo sui tag: minore è il tempo impiegato per trovare un tag comune, maggiore sarà il numero di punti guadagnato.

Il numero elevato di persone che si è cimentato in questo gioco, ha permesso di catalogare milioni di immagini nella maniera più semplice possibile, risparmiando tempo e denaro e allo stesso tempo facendo divertire gli utenti.

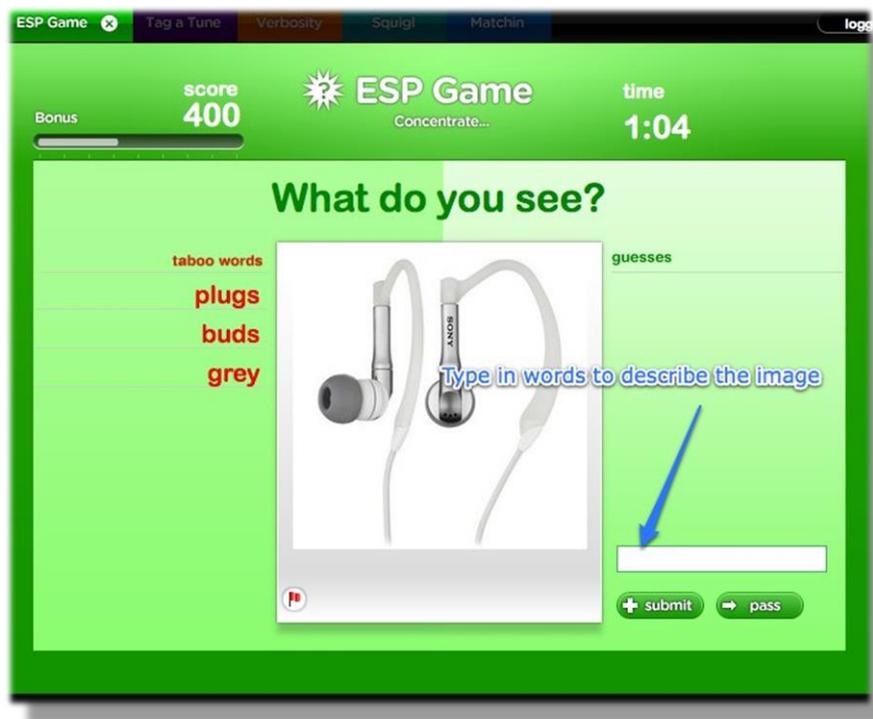


Fig. 2.1 - Un schermata di gioco di ESP game

PeekaBoom

Mentre Esp game è in grado di determinare **cosa** è presente in un'immagine, non dice nulla sul **dove** gli oggetti siano collocati. Queste informazioni di posizione sono fondamentali per allenare e testare gli algoritmi di computer vision. PeekaBoom [5] è un GWAP che permette di risolvere questo task, migliorando di molto le informazioni catalogate dal gioco precedentemente descritto. Più precisamente, identifica quali pixels appartengono a quale oggetto all'interno di un'immagine. Come tutti i giochi agreement-based, vengono selezionati due giocatori in maniera casuale ai quali viene assegnato un ruolo ciascuno: *Peek* and *Boom*. Peek inizia con una schermata vuota mentre Boom vede un'immagine e delle parole associate. Lo scopo di Peek è quello di indovinare le parole associate mentre Boom rivela, una dopo l'altra, alcune parti della

foto: ogni volta che Boom clicca sull'immagine, un'area di 20 pixels di raggio si scopre a Peek che deve capire quale parola sia associata all'immagine, scrivendola in una textbox. Peek, il quale vede quello che scrive Boom, può indicare se la parola si avvicini o meno a quella originale indicando se sia *calda* o *fredda*. Nel caso in cui l'intuizione di Boom fosse corretta, entrambi guadagnano punti per poi passare all'immagine successiva. Questo gioco ha riscosso molto successo e ha permesso di fornire una soluzione all'importante problema di localizzazione digitale di elementi all'interno di una figura.

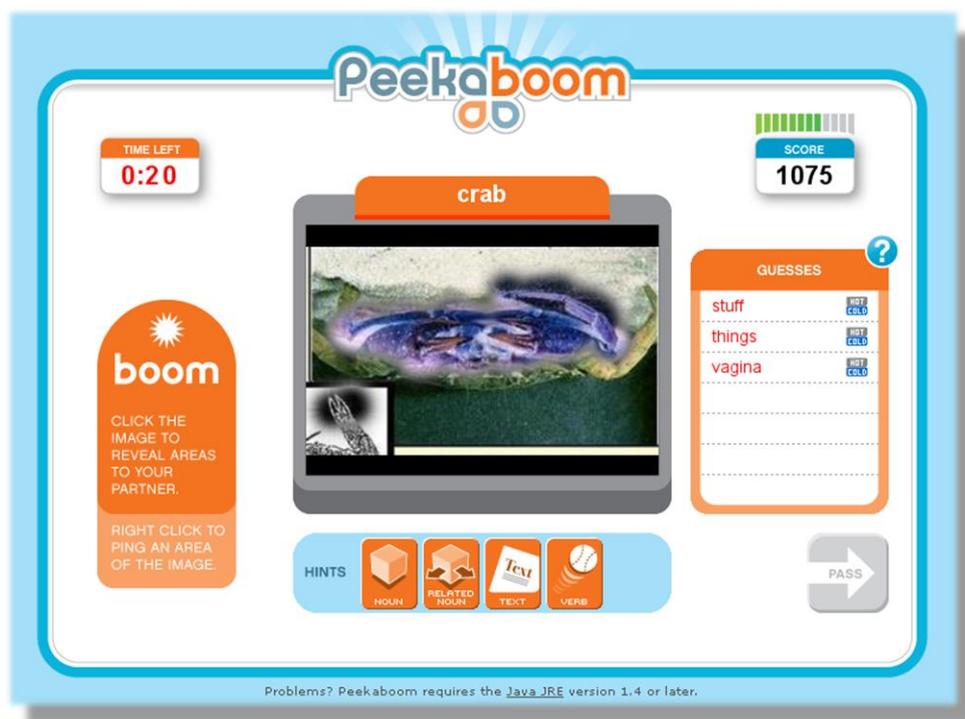


Fig. 2.2 - Una schermata del gioco Peekaboom

Verbosity

Verbosity [6] è un gioco ispirato al famoso *Taboo*. Lo scopo principale è quello di collezionare una conoscenza comune riguardo vari oggetti. Anche questo è un gioco basato su due utenti che assumono differenti ruoli: il primo diventa *narratore*, mentre il secondo, assume le vesti di *indovino*. Scelto in modo casuale quale dei due giocatori sarà

il narratore, a lui viene mostrata una parola, che il secondo giocatore dovrà cercare di indovinare grazie alla descrizione che avrà fatto il narratore usando piccole frasi che non contengano, ovviamente, il termine in questione e nemmeno uno molto simile. L'unico modo per indovinare è quello di dedurre la parola solamente dalle frasi scritte a schermo poiché non è possibile una comunicazione verbale. Il tempo impiegato per trovare il termine esatto è sinonimo della qualità della descrizione del narratore. Ovviamente anche questo gioco richiede che entrambe le persone giochino in simultanea.



Fig. 2.3 - Una schermata del gioco Verboosity

Duolingo

Duolingo [7] è uno degli ultimi giochi inventati da Von Ahn, che permette l'apprendimento e l'affinamento di una lingua straniera da parte dell'utente. Duolingo segue una logica applicata all'apprendimento delle lingue, e alla **traduzione**. In sostanza, si tratta di corsi intensivi per imparare una lingua straniera proposti, esteticamente, come un gioco con percorso e traguardi e, metodologicamente, come una serie di esercizi che prevedono **l'infinita ripetizione di fonemi, parole, strutture grammaticali**,

con la certezza che, alla lunga, la nuova lingua verrà assimilata nelle conoscenze dell'individuo. Questo è però solo uno degli aspetti interessanti del gioco, poiché lo scopo primario di Von Ahn e del suo staff è infatti quello di usare queste lezioni di massa per riuscire, un giorno, a tradurre i contenuti della rete per intero tramite le centinaia di milioni di frasi trasposte dagli utenti-studenti, che diventano a loro volta traduttori della rete, o per conto di essa, dando loro la possibilità di imparare e migliorare la conoscenza di una lingua straniera.

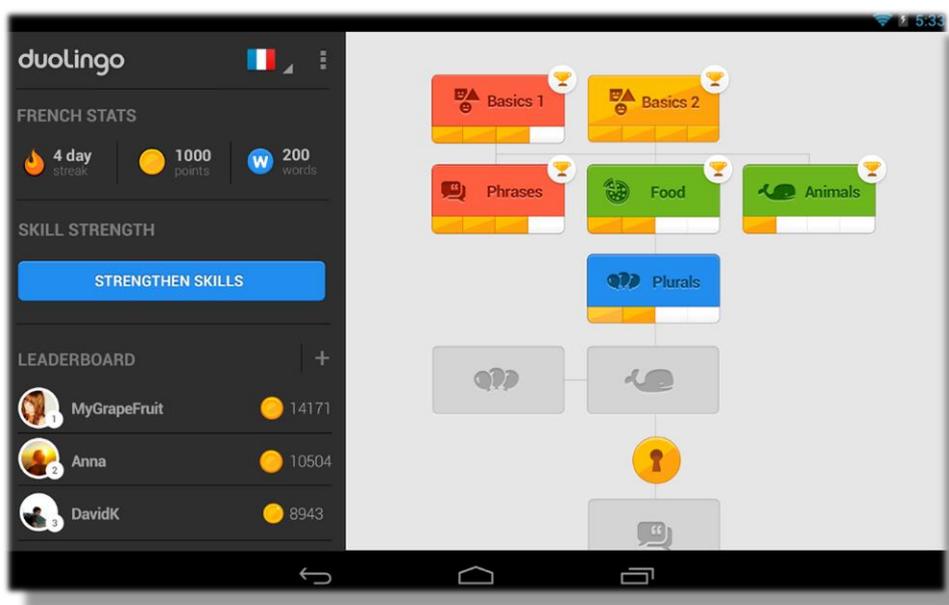


Fig. 2.4 - Una schermata di gioco di Duolingo

2.4 GWAP per classificare Linked Data

Con l'introduzione del Semantic Web, il modo di concepire il Web è cambiato radicalmente passando da una visione di condivisione dell'informazione strettamente riservata tra persone umane, ad una comunione di dati significativi tra le macchine stesse. Due dei più famosi risultati prodotti da questa trasformazione sono stati lo sviluppo del **Resource Description Framework** [2] e l'**Ontology Web Language** [8]. OWL è un linguaggio di markup per rappresentare esplicitamente significato e semantica di termini attraverso l'uso di vocabolari e delle relazioni tra gli stessi. Lo scopo di OWL è descrivere delle basi di conoscenze, effettuare delle deduzioni su di esse e integrarle con i contenuti delle pagine Web. Grazie a OWL in futuro sarà possibile, ad esempio, effettuare delle ricerche estremamente complesse nel Web evitando i problemi di omonimia e ambiguità presenti nelle normali ricerche testuali. Altro scopo di OWL è permettere alle applicazioni di effettuare delle deduzioni sui dati.

In principio i dati raccolti nella rete non erano sufficienti per trarre benefici tangibili da queste tecnologie innovative, solo con il suo progredire e diffondersi, il Web è stato arricchito di un database abbastanza vasto di informazione che ha permesso la nascita dell'iniziativa **Linked Open Data** che, nel corso degli anni e attraverso il contributo della comunità scientifica del Semantic Web, ha raccolto, catalogato e incluso numerosi dataset pubblicati sul Web come *Linked Data* [9], costituiti da diversi miliardi di triple RDF.

Estratto da Wikipedia, DBpedia [1] è il più importante dataset facente parte della nuvola dei Linked Open Data. Esso presenta una quantità innumerevole di informazioni collegate tra loro con relazioni di proprietà-valore, raccolte e immagazzinate in forma strutturata nelle pagine della famosa enciclopedia multimediale.

Nonostante il grande successo del Semantic Web e di Wikipedia, la crescita esponenziale di linked data ha causato nuovi problemi. Mentre all'inizio non c'era un volume di informazioni abbastanza ampio, e la ricerca di dati era più semplice e portava a risultati

più specifici, adesso è più facile ricevere come risultato di una ricerca solo contenuti migliori dal punto di vista quantitativo ma non da quello qualitativo.

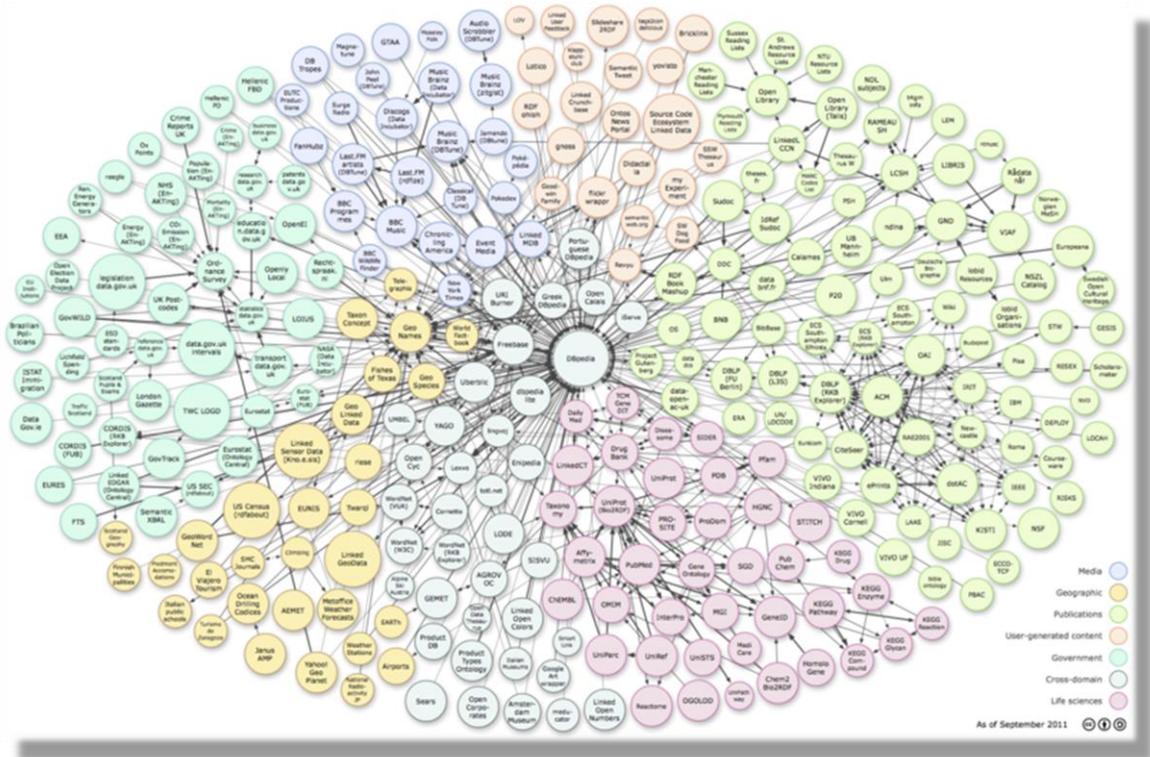


Fig. 2.5 - La nuvola di Linked Open Data osservata a fine 2011 [10]

La capacità umana di creare associazioni mentali gioca un ruolo chiave in questa problematica, poiché grazie al pensiero umano è possibile classificare in modo significativo l'informazione, inclusa quella della Linked Open Data Cloud.

A questo scopo sono stati creati diversi GWAP; i più conosciuti e relativi allo studio della popolarità di Linked Data sono **Who Knows?** e **Better Relations**, descritti di seguito.

Who Knows?

Who Knows [11] è un gioco in modalità single player il cui scopo è quello di classificare la popolarità di fatti. All'inizio della partita, il sistema estrae degli statement da DBpedia e crea una lista di domande differenti che posso essere a risposta multipla o meno. Queste domande vengono mostrate in sequenza al giocatore il quale deve rispondere entro un limite di tempo. L'utente riceve più o meno punti in base alla correttezza della risposta e al tempo impiegato per rispondere. La peculiarità sta nella possibilità della persona coinvolta nel gioco di contestare una risposta esatta che secondo lui non corrisponde alla realtà. Grazie a questa funzionalità è possibile raccogliere dei feedback utili per correggere errori o risolvere delle ambiguità.

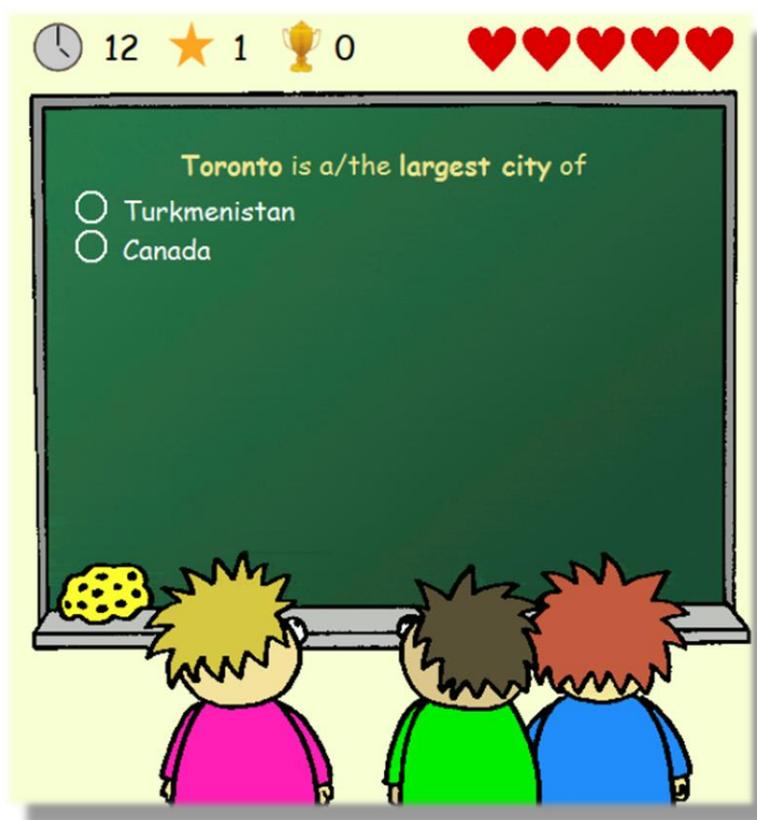


Fig. 2.6 - Schermata di gioco di Who Knows?

Better Relations

Appartenente alla famiglia degli Output Agreement Games, Better Relations [12] è un gioco che vede due utenti contrapposti tra di loro in maniera simultanea. In ogni round, che ha una durata di due minuti, il sistema mostra ai partecipanti un argomento e due fatti ad esso relativi estratti sempre da DBpedia. I giocatori devono scegliere quale, secondo il loro parere, è più importante rispetto all'altro. Nel caso le risposte siano uguali viene assegnato un punteggio positivo ad entrambi. In aggiunta sono presenti due pulsanti che l'utente può premere nel caso in cui o non riesca a decidere quale dei due fatti sia più rilevante rispetto all'altro, o nel caso in cui entrambe le notizie siano false o senza senso. Il sistema raccoglie così i feedback degli utenti e stila una classifica delle informazioni in base alla loro popolarità.



Fig. 2.7 - Una schermata di gioco di Better Relations

È a questi due ultimi modelli presentati che si è fatto riferimento per creare l'applicazione, che ricalca in parte l'obiettivo per cui sono state processate. Nello specifico, lo scopo del lavoro è quello di assegnare un rank alle relazioni contenute in DBpedia in base a quanto siano più o meno note alla gente comune.

3. PROGETTAZIONE DELLA SOLUZIONE

Come anticipato nell'introduzione, lo scopo del progetto è quello di realizzare, secondo l'approccio definito da Von Ahn, un'applicazione sotto forma di GWAP che abbia come *purpose* di ordinare un insieme di fatti (nello specifico, un set di triple estratte da DBpedia) in base alla loro popolarità. L'obiettivo è quindi quello di sfruttare le tecniche di Human Computation per ordinare questo dataset di informazioni in base alla loro *notorietà* presso il grande pubblico, ordinamento che è di difficile realizzazione tramite il solo utilizzo di processi automatici, che mancano infatti del background di conoscenza delle persone. Con lo sviluppo dell'applicazione, si avrà l'occasione di far svolgere questo compito in maniera naturale a degli utenti comuni, incentivandoli alla partecipazione attraverso un'esperienza di gioco divertente.

3.1 Struttura del gioco

La progettazione di un GWAP dal punto di vista concettuale, deve essere in grado di fondere due importanti aspetti: il divertimento dell'utente e l'utilità per uno scopo. Senza il primo, il secondo non ha possibilità di esistere, per cui è fondamentale che l'intera struttura di gioco sia orientata a coinvolgere e appassionare la persona in modo che sia naturalmente portata a giocare ripetutamente senza annoiarsi.

Proprio il numero di utenti e la quantità di partite effettuate sono i primi elementi con cui misurare il successo di un GWAP; infatti, maggiore è il numero di giocatori e di match giocati, più dati si possono raccogliere per risolvere il *purpose* annegato nel gioco e per effettuare le analisi necessarie a capire l'efficacia dello stesso nel raggiungere tale obiettivo.

Analizzando i risultati di ricerca descritti nel precedente capitolo è emersa come soluzione molto frequente, l'introduzione di un meccanismo di sincronia tra i due

partecipanti. Questo significa che il gioco deve avere un elevato numero di utenti liberi allo stesso momento: per iniziare una partita è necessario abbinare almeno due persone¹. Nel caso trattato però, si è ritenuto che un funzionamento del genere potesse essere troppo complicato dal punto di vista realizzativo e potesse limitare in qualche modo l'esperienza di gioco (facendo attendere un giocatore a lungo al fine di trovargli un avversario) e la buona riuscita del progetto (nel caso di non riuscire ad ottenere un numero sufficiente di partite giocate).

Si è deciso dunque di sostituire all'approccio di tipo *agreement* tipico dei giochi di Von Ahn, un criterio di sfida *uno contro tutti* che ha come obiettivo quello di scalare una classifica dei migliori giocatori. Sulla base della competizione tra gli utenti, si è cercato di costruire una struttura di gioco che assicurasse un soddisfacente numero di partite per verificare la capacità del GWAP di risolvere il suo *purpose*.

3.2 L'ispirazione

Lo spunto di partenza per lo sviluppo del lavoro è il famoso puzzle game **Ruzzle** [13] disponibile come applicazione sulla maggior parte degli smartphone. Ruzzle è un gioco basato sulla ricerca di parole all'interno di una griglia di 16 caselle nel formato 4x4 ognuna contenente una lettera dell'alfabeto e un valore numerico differente. Nell'arco di tre round il compito del giocatore è quello di toccare tali caselle e formare il maggior numero di parole possibili entro un limite di tempo, ottenendo un punteggio maggiore qualora utilizzasse le lettere evidenziate dal bonus. Al termine della partita, il punteggio realizzato viene confrontato con quello dell'avversario e vince chi ha realizzato quello migliore. Come si può dedurre, il gioco, pur non avendo alcun *purpose* nascosto, è sulla stessa linea di quelli analizzati nel capitolo precedente, perchè vengono messi a confronto due utenti i quali stavolta si sfidano tra di loro. La caratteristica di questo gioco è che

¹ I Gwap con meccanismo sincrono utilizzano spesso anche il trucco di simulare tale sincronia con partite pre-registrate. In ogni caso, anche per seguire questa strada, è necessario avere un alto numero di utenti per avere un alto numero di partite giocate.

adotta un **meccanico asincrono** ossia, un round può essere giocato anche non in contemporanea con l'avversario, ma in un secondo momento. In aggiunta c'è anche una modalità offline in cui l'obiettivo è quello di fare il punteggio maggiore all'interno di una classifica.

In termini di game design, Ruzzle può essere considerato un ottimo modello per l'interfaccia grafica dell'applicazione, poiché è possibile immaginare di sostituire al posto delle lettere le risorse o le proprietà o gli oggetti delle triple estratte da DBpedia e creare così una sorta di griglia in cui lo scopo del gioco sia quello di unire le caselle ricreando le relazioni esatte iniziali. A differenza di Ruzzle, in questo caso, non sarà possibile attribuire ad ogni risorsa un punteggio bonus poiché non è possibile stabilire a priori se la stessa faccia parte di una tripla più o meno difficile da individuare, difatti questo è proprio uno degli obiettivi dell'applicazione, ovvero cercare di stabilire quali dati siano più conosciuti e di conseguenza più facili da trovare e quali siano meno noti e quindi più ardui da individuare.

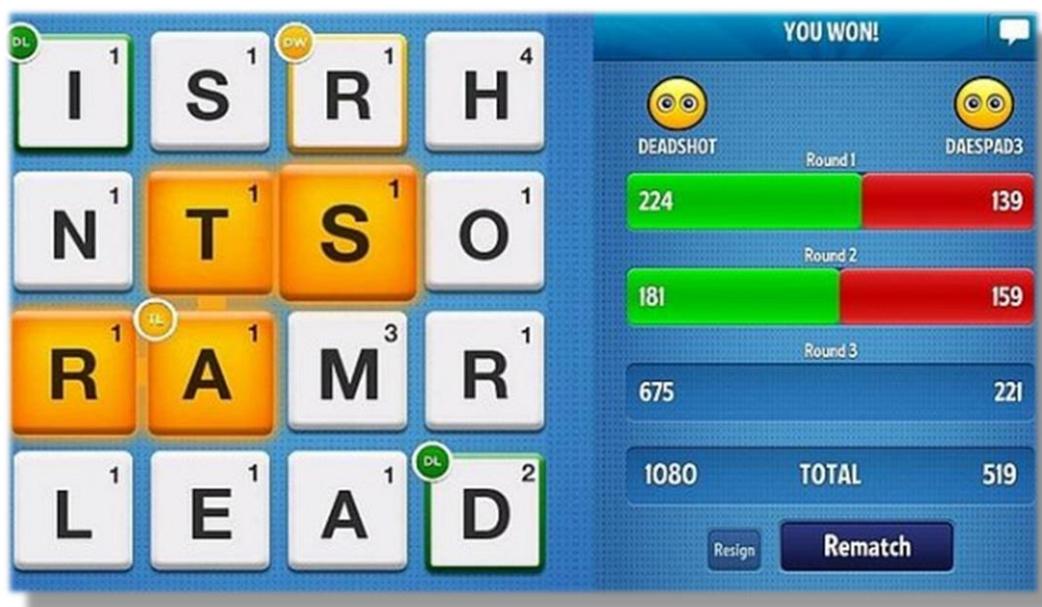


Fig. 3.1 - Una schermata di gioco di Ruzzle

3.3 Le alternative

Il prossimo problema da affrontare è quello di decidere quale struttura adottare per la creazione di una griglia che meglio si adatti al caso in esame. Sull'esempio di Ruzzle sono stati definiti tre modi per comporre la scacchiera di gioco, ogni metodo è stato analizzato per verificarne i pregi e i difetti in relazione all'utilizzo che ne conseguirà.

Prima alternativa

La prima alternativa riprende esattamente lo schema di Ruzzle, una griglia 4x4 di sedici caselle nelle quali sono incastrate diverse triple soggetto-predicato-oggetto in modo che siano in relazione tra loro. La relazione consiste nel fatto di condividere una stessa risorsa o uno stesso predicato tra più triple, in modo da creare dei *serpenti* partendo da una casella in cui è presente un soggetto e collegando via via altre caselle seguendo rigorosamente lo schema soggetto -> predicato -> oggetto (che a sua volta diventa soggetto). Più lungo è il percorso trovato, più punti è possibile guadagnare. Un esempio di un possibile percorso potrebbe essere:

❖ **Lombardia** -> *capoluogo* -> **Milano** -> *sindaco* -> **Pisapia** -> *sposato* -> **Cinzia Sasso**

dove le parole in grassetto identificano una risorsa (soggetto/oggetto) e le parole in corsivo identificano un predicato.

- *Vantaggi*: il gioco può risultare molto avvincente; si esaltano le capacità dell'utente di individuare connessioni anche tra risorse appartenenti ad ambiti totalmente differenti; può risultare molto longevo dal punto di vista del gameplay.
- *Svantaggi*: è difficile creare un algoritmo di estrazione dei dati da DBPedia che sia in grado di incastrare gli elementi nelle caselle tenendo conto delle relazioni che possiedono; potrebbe risultare leggermente confusionario per un occhio

non in grado di riconoscere al volo le corrispondenze tra una risorsa e il suo significato; richiede uno sviluppo molto elaborato e complesso sia dal punto di vista della progettazione che della raccolta dati.

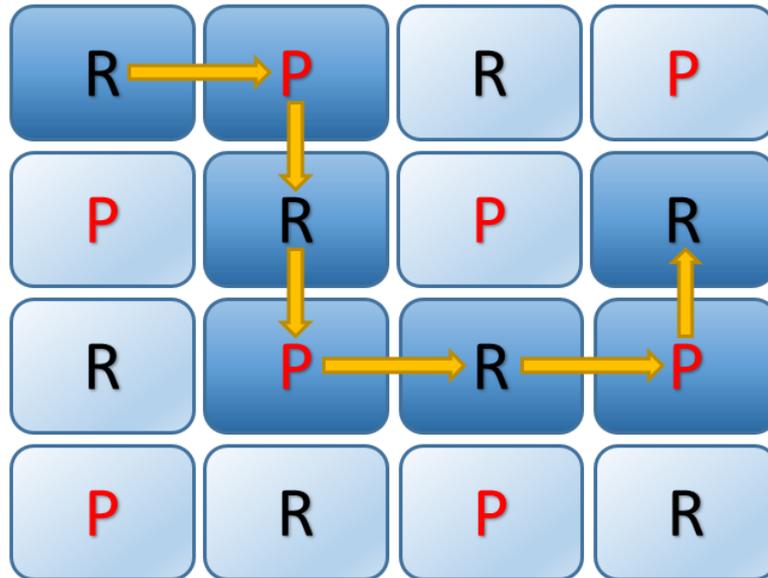


Fig. 3.2 - Esempio di griglia 4x4, la lettera R indica la risorsa, la P il predicato

Seconda alternativa

Nella seconda alternativa il numero e la disposizione delle caselle varia, la griglia diventa di grandezza 3x3 e la posizione centrale è occupata dall'unico predicato presente nella scacchiera. Nelle caselle intorno sono inserite una serie di risorse che questa volta assumono il ruolo di soggetto o oggetto a seconda della relazione che hanno con il predicato presente. In questo modo è possibile mantenere la struttura principale della tripla estratta da DBpedia, infatti nelle otto caselle occupate dalle risorse saranno distribuite equamente quattro soggetti e quattro oggetti. La caratteristica sta nel fatto che è possibile scegliere un argomento da cui partire per l'estrazione dei dati poiché, essendoci un unico predicato, tutte le risorse saranno di norma appartenenti ad uno stesso ambito (grazie alla definizione di dominio e codominio del predicato). Oltretutto è possibile anche che alcune risorse poste nella griglia, non siano legate in alcun modo

alla proprietà posta al centro, in modo da variare le possibili giocate. L'obiettivo del giocatore sarebbe dunque quello di scovare una per volta tutte le triple nascoste. Un esempio pratico potrebbe essere la sequenza di triple:

- ❖ **Mario Balotelli** -> *club* -> AC Milan
- ❖ **Mario Balotelli** -> *club* -> FC Internazionale
- ❖ **Kakà** -> *club* -> AC Milan
- ❖ **Kakà** -> *club* -> Real Madrid
- ❖ **Cristiano Ronaldo** -> *club* -> Real Madrid

dove in grassetto sono rappresentati i soggetti, in corsivo l'unico predicato (nell'esempio, il far parte o l'aver fatto parte di una determinata squadra) e sottolineati sono gli oggetti. Come si può notare un soggetto o un oggetto potrebbero essere in comune con più combinazioni.

- Vantaggi: la struttura è molto semplice ed intuitiva; il giocatore è facilitato dal fatto che le risorse appartengano ad un ambito ristretto; la progettazione è di facile sviluppo e implementazione per quanto riguarda l'aspetto tecnico.
- Svantaggi: le partite potrebbero risultare troppo semplici perchè, contando le varie combinazioni possibili di triple che è possibile inserire, il numero varia da 1 a 16 dove gli estremi sono i casi molto rari in cui nella griglia è presente solamente una tripla esatta oppure tutte le combinazioni soggetto – predicato – oggetto sono valide; il match potrebbe durare pochi secondi nel caso in cui la griglia sia troppo triviale; alla lunga l'esperienza di gioco potrebbe risultare troppo noiosa e poco stimolante.

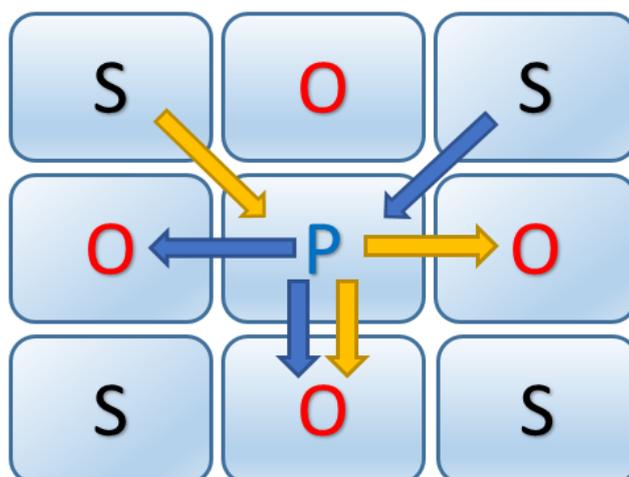


Fig. 3.3 - Esempio di griglia 3x3, la lettera S indica il soggetto, la P il predicato e la O l'oggetto

Terza alternativa

La terza alternativa fonde i due modelli descritti in precedenza adottando le caratteristiche di entrambi, semplificando il primo e migliorando il secondo. In questo schema la griglia diventa di dimensioni 3x5 dove è possibile individuare 3 colonne da 5 caselle ciascuna. A differenza delle altre forme, in questo caso, le tre colonne ricalcano la struttura base di una tripla ed individuano rispettivamente il soggetto, il predicato e l'oggetto. La scacchiera sarà dunque composta dalla prima colonna nelle cui 5 caselle saranno inserite risorse che assumeranno il ruolo di soggetto, dalla seconda colonna comprendente 5 caselle di predicati e dalla terza colonna la quale contiene 5 caselle con altrettante risorse sottoforma di oggetti. Lo schema è abbastanza chiaro, lo scopo del giocatore è quello di trovare le triple nascoste unendo le caselle da sinistra a destra. Caratteristica fondamentale è che la complessità appartenente alla griglia 4x4 è stata ridotta adottando un sistema di gioco che obbliga la connessione di singole triple, mentre la semplicità della griglia 3x3 è stata aumentata rimuovendo il vincolo sulla tipologia dell'argomento che la presenza di un unico predicato aveva introdotto.

- Vantaggi: giusto compromesso tra semplicità e difficoltà di gioco; struttura di gioco lineare e di facile comprensione da parte dell'utente; duttilità nel variare

gli ambiti delle risorse all'interno della griglia; maggior numero di combinazioni possibili da inserire all'interno della schacchiera; difficoltà gestibili nel realizzare lo sviluppo dell'architettura dell'applicazione.

- Svantaggi: l'aumento del numero di predicati e delle combinazioni tra sogetti e oggetti potrebbe portare alla formazione di griglie con risorse completamente differenti in termine di tipologia.

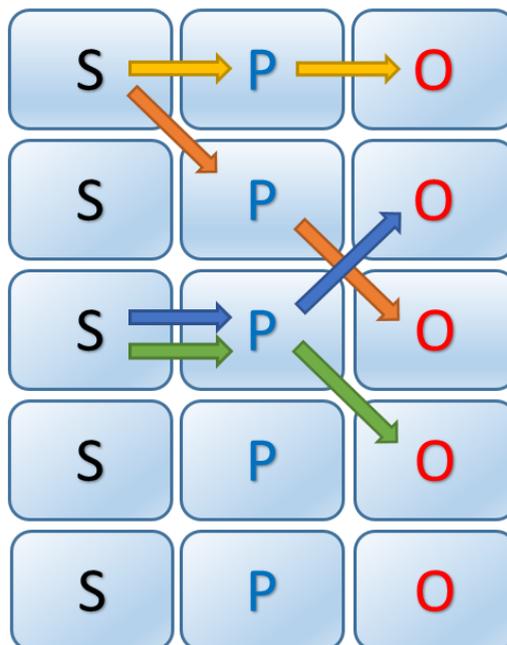


Fig. 3.4 - Esempio di griglia 3x5, la lettera S indica il soggetto, la P il predicato e la O l'oggetto

4. WIKIFINDER

Dopo un attento studio delle possibili alternative elencate in precedenza, la scelta è ricaduta sull'utilizzo dell'ultimo modello di struttura. La griglia 3x5 offre uno schema di gioco semplice ed intuitivo, capace di rendere l'esperienza di gioco fluida e non troppo complicata. La varietà delle combinazioni possibili in cui sistemare le triple permette di evitare ripetizioni nella creazione delle griglie e di assicurare una discreta longevità al gioco. Il nome scelto per l'applicazione è **WikiFinder**, poiché l'obiettivo del gioco è quello di trovare le correlazioni esatte tra le varie risorse provenienti da Wikipedia.

4.1 Estrazione dati da Wikipedia e creazione griglie

Per estrarre dati da DBpedia è innanzitutto necessario eseguire delle interrogazioni sul database utilizzando un linguaggio simile ad SQL chiamato **SPARQL** (SPARQL Protocol and RDF Query Language). SPARQL [14] è un linguaggio di interrogazione per dati rappresentati tramite RDF ed è uno degli elementi chiave delle tecnologie legate al Semantic Web poiché ci permette sia di esplorare informazioni senza conoscerne le relazioni, sia di eseguire delle interrogazioni su diverse banche dati RDF tramite una singola query.

Un esempio di query in SPARQL è il seguente dove la dicitura *prefix* indica un prefisso per abbreviare il namespace delle URI delle risorse.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX owl: <http://dbpedia.org/ontology/>
3 SELECT DISTINCT *
4 WHERE {
5     ?s    rdf:type    owl:SoccerPlayer .
6 }
7 LIMIT 10
```

Fig. 4.1 - Esempio di interrogazione SPARQL

In particolare, in questa query, specifichiamo che stiamo cercando tutti i soggetti (?s) che appartengono (*rdf:type* è il predicato) alla classe dei calciatori (*owl:SoccerPlayer* è l'oggetto). Il risultato sarà un elenco di risorse del tipo:

http://dbpedia.org/resource/Mario_Balotelli.

L'algoritmo

La creazione delle griglie, comporta dunque l'esecuzione di una serie di interrogazioni al database di DBpedia, ognuna delle quali tiene conto di determinati vincoli stabiliti per il graduale riempimento di tutte le caselle. Nello specifico, la prima operazione, comporta la scelta casuale di un numero predefinito di **percorsi** tra tutte le possibili combinazioni che una griglia 3x5 permette. Per percorso si intende una sequenza di caselle adiacenti sia in direzione orizzontale che in direzione obliqua che prevede la scelta di un soggetto, di un predicato e di un oggetto tra le tre colonne. Ovviamente il percorso partirà sempre da sinistra nella colonna dei soggetti e terminerà a destra in quella degli oggetti. Numerando le righe da 1 a 5 e nominando le colonne con le lettere A, B e C, un percorso

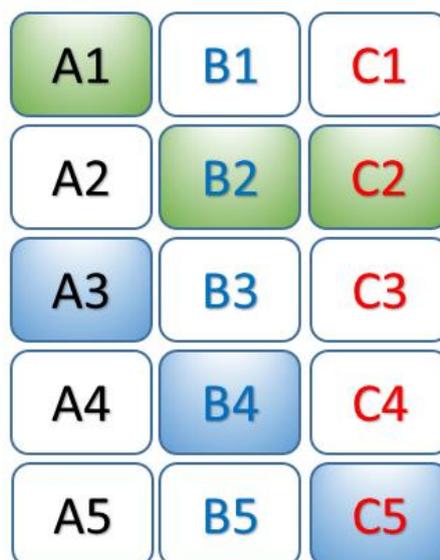


Fig. 4.2 - Esempio di percorsi possibili per ospitare le triple estratte da DBpedia

potrebbe essere definito dalle sequenze: A1-B2-C2 oppure A3-B4-C5 (cf. figura 4.2). Le caselle identificate da questi percorsi saranno quindi le potenziali *ospitanti* delle triple estratte.

La seconda operazione prevede l'estrazione dei dati e il riempimento delle caselle seguendo una serie di determinati controlli. Inizialmente si parte dal primo percorso tra quelli selezionati e si cerca tramite una query SPARQL una tripla da cui partire. Per estrarre solamente dei dati significativi che potessero essere utili al lavoro, è stato posto un vincolo alla tipologia del predicato e alla forma dell'oggetto: ogni predicato dovrà essere stato definito nel namespace <http://dbpedia.org/property> oppure nel namespace <http://dbpedia.org/ontology> che includono le principali proprietà definite in DBpedia; gli oggetti dovranno essere esclusivamente delle risorse identificate da un'URI e non dagli RDF literal (come stringhe o numeri). Una volta riempite le caselle appartenenti al primo percorso, si passa al secondo.

Quindi, si controlla se siano presenti riquadri già occupati in precedenza. In caso di riscontro positivo, viene generata una query SPARQL che estragga tutte le triple che abbiano in comune il contenuto delle caselle occupate, altrimenti ne viene estratta un'altra in maniera casuale. Nella pratica, supponendo di avere la sequenza A2-B2-C1 dove C1 risulta già popolata dalla risorsa *AC Milan*, l'interrogazione cercherà tutte le triple che avranno come oggetto la suddetta risorsa e ne verrà scelta una per riempire ad esempio il percorso A1-B1-C1. Seguendo questo meccanismo l'algoritmo popola tutte le caselle della griglia controllando che non vi siano inseriti dei doppioni all'interno di una stessa colonna. Nel caso la procedura non riesca a trovare dei risultati validi per riempire i riquadri di un determinato percorso, questo sarà scartato.

Terminato il processo, si individuano le eventuali caselle rimaste vuote alle quali si assegnano risorse o predicati scelti in maniera casuale imponendo che essi non abbiano alcuna relazione con gli elementi adiacenti, eliminando il rischio di avere inavertitamente una tripla esatta non prevista. Di seguito sono elencati alcuni template di query utilizzate per la ricerca delle triple tramite SPARQL.

```

1  select  <$$subject.$$> ?p ?o
2  where  {
3      <$$subject.$$> ?p ?o .
4      FILTER(regex(str(?p), "http://dbpedia.org/ontology/") or
5              regex(str(?p), "http://dbpedia.org/property/"))
6      FILTER(isURI(?o)) .
7      FILTER($$BlackList$$)
8  }

```

Fig. 4.3 - Template query Sparql: caso in cui il soggetto è già presente in griglia

```

9  select  ?s <$$predicate$$> ?o
10 where {
11     ?s <$$predicate$$> ?o .
12     FILTER(isURI(?o)) .
13 } LIMIT 1000

```

Fig. 4.4 - Template query Sparql: caso in cui il predicato è già presente in griglia

```

15 select  ?s ?p <$$objects$$>
16 where  {
17     ?s ?p <$$objects$$> .
18     FILTER(regex(str(?p), "http://dbpedia.org/ontology/") or
19             regex(str(?p), "http://dbpedia.org/property/"))
20     FILTER(isURI(?o)) .
21     FILTER($$BlackList$$)
22 }

```

Fig. 4.5 - Template query Sparql: caso in cui l'oggetto è già presente in griglia

```

23  select ?s <$$predicate$$> <$$objects$$>
24  where {
25      ?s <$$predicate$$> <$$objects$$> .
26  }

```

Fig. 4.6 - Template query Sparql: caso in cui sia il predicato che l'oggetto sono già presenti in griglia

```

28  select <$$subject.$$> <$$predicate$$> ?o
29  where {
30      <$$subject.$$> <$$predicate$$> ?o .
31      FILTER(isURI(?o)) .
32  }

```

Fig. 4.7 - Template query Sparql: caso in cui sia il soggetto che il predicato sono già presenti in griglia

```

34  select <$$subject.$$> ?p <$$objects$$>
35  where {
36      <$$subject.$$> ?p <$$objects$$>.
37      FILTER(regex(str(?p), "http://dbpedia.org/ontology/") or
38              regex(str(?p), "http://dbpedia.org/property/"))
39      FILTER($$BlackList$$)

```

Fig. 4.8 - Template query Sparql: caso in cui sia il soggetto che l'oggetto sono già contenuti in griglia

Di seguito è mostrato il processo eseguito dall' algoritmo in linea generale tramite la scrittura in pseudocodice delle parti fondamentali.

```

1  griglia = Array();
2  percorsi = Array();
3  tripla_selezionata = Array();
4  percorsi = scegliPercorsi(ArrayCombinazioniPossibili);
5  black_list = Array(proprietàDaEvitare)
6  foreach percorsi as percorso { { /////IL PERCORSO È UN ARRAY CONTENENTE
   LA TRIPLA NELLA FORMA TIPO "A1-B2-C3" /////}
7      sogg = percorso[0];
8      pred = percorso[1];
9      ogg = percorso[2];
10     if ((presenteInGriglia(sogg) == false) & (presenteInGriglia(pred) == false) & (
        presenteInGriglia(ogg) == false)) {
11         query = cerco una tripla qualsiasi;
12     }
13     else if ((presenteInGriglia(sogg) == true) & (presenteInGriglia(pred) == true) & (
        presenteInGriglia(ogg) == true)) {
14         flag_occupate = true;
15     }
16     else if ((presenteInGriglia(sogg) == true) & (presenteInGriglia(pred) == false) &
        (presenteInGriglia(ogg) == false)) {
17         query = cerco triple con quel soggetto;
18     }
19     else if ((presenteInGriglia(sogg) == false) & (presenteInGriglia(pred) == true) &
        (presenteInGriglia(ogg) == false)) {
20         query = cerco triple con quel predicato;
21     }
22     else if ((presenteInGriglia(sogg) == false) & (presenteInGriglia(pred) == false) &
        (presenteInGriglia(ogg) == true)) {
23         query = cerco triple con quell'oggetto;
24     }
25     else if ((presenteInGriglia(sogg) == true) & (presenteInGriglia(pred) == true) & (
        presenteInGriglia(ogg) == false)) {
26         query = cerco triple che abbiano quel soggetto e quel predicato;
27     }
28     else if ((presenteInGriglia(sogg) == false) & (presenteInGriglia(pred) == true) &
        (presenteInGriglia(ogg) == true)) {
29         query = cerco triple che abbiano quel predicato e quell'oggetto;
30     }
31     else if ((presenteInGriglia(sogg) == true) & (presenteInGriglia(pred) == false) &
        (presenteInGriglia(ogg) == true)) {
32         query = cerco triple che abbiano quel soggetto e quell'oggetto;
33     }
34
35     if (flag_occupate = false) {
36         risultato = Sparql(query);
37         if (risultato) { { /////CONTROLLI SUL RISULTATO DELLA QUERY /////}
38             tripla_selezionata = scelgoTriplaCasuale(risultato);
39             soggetto = tripla_selezionata[0];
40             predicato = tripla_selezionata[1];
41             oggetto = tripla_selezionata[2];
42             While ((presenteInGriglia(soggetto) == true) OR (presenteInGriglia(
                predicato) == true) OR (presenteInGriglia(oggetto) == true)) {
43                 tripla_selezionata = scelgoTriplaCasuale(risultato);
44             }
45             griglia[sogg] = soggetto;
46             griglia[pred] = predicato;
47             griglia[ogg] = oggetto;
48         }
49     }
50     flag_occupate = false;
51 }

```

Fig. 4.9 - Algoritmo in pseudocodice

4.2 Architettura Client-Server

Una volta create un numero sufficiente di griglie, è necessario progettare un'architettura client-server per l'archiviazione delle griglie e per lo scambio dei dati prima, dopo e durante una partita. Il primo aspetto da tenere in considerazione è la quantità di dati scambiati durante l'interazione tra l'applicazione e il server. Per non influire sulla fluidità e sulla giocabilità si è optato per l'invio e la ricezione di informazioni solamente all'apertura dell'applicazione (con il trasferimento dei dati della griglia da giocare dal server al client) e al termine di ogni match (con il trasferimento delle informazioni sulle triple selezionate dal giocatore dal client al server). In questo modo è stato possibile prevenire eventuali perdite di connessione o rallentamenti della rete che avrebbero potuto rovinare l'esperienza ludica dell'utente. L'architettura prevede come prima cosa, un database dove archiviare le scacchiere di gioco, l'anagrafe degli utenti, le partite effettuate, i punteggi e le singole triple scelte durante i round. Questi dati sono memorizzati su un server il quale, tramite l'utilizzo di servizi creati ad hoc per l'applicazione, funge da tramite per le comunicazioni tra l'interfaccia e il database: ogni volta che il client ha bisogno di informazioni specifiche, effettua una richiesta al server che, a sua volta, interroga il database prelevando i dati richiesti. Questi, successivamente, sono inviati come risposta alla richiesta sotto forma di **JSON** [15]. JSON è un formato adatto per lo scambio di informazioni in applicazioni di questa tipologia poiché presenta una semplicità di interpretazione cruciale per molti linguaggi di programmazione.



Fig. 4.10 - Schema generale dell'architettura

I servizi

I servizi presenti che gestiscono il collegamento tra interfaccia e database sono stati tutti progettati tramite il linguaggio **PHP** [16] (Hypertext Preprocessor) comune in molti ambienti di programmazione. Il PHP è un linguaggio concepito principalmente per la creazione di pagine Web dinamiche, ossia pagine il cui contenuto è generato sul momento dal server e quindi in grado di mostrare un output diverso a seconda delle esigenze, garantendo così una maggiore interattività con l'utente.

Sono stati creati servizi per le funzionalità principali quali il login, la registrazione, l'invio delle griglie, l'invio dei risultati, la gestione dei punteggi e delle soluzioni. Nelle figure 4.11 e 4.12 sono riportati due tra quelli più significativi: l'invio all'interfaccia della griglia all'inizio di una partita e l'invio al server dei risultati al termine del match.

```
1 <?php
2 //sELEZIONE GRIGLIA E INVIO AL CLIENT
3 function getGrid($utente) {
4     global $db;
5     $query='SELECT * FROM grid WHERE id NOT IN (SELECT id_griglia FROM partita
6     WHERE utente ='. $utente.' ) ORDER BY RAND() LIMIT 0,1;';
7     $result=mysql_query($query,$db) or pLog(mysql_error());
8     if (mysql_num_rows($result)>0) {
9         return gridByQuery($result);
10    }
11 }
12 function gridByQuery($result) {
13     while($row=mysql_fetch_array($result)) {
14
15         $grid[]=array(
16             "id_grid"=> $row["id"],
17             "a1"=> $row["a1"],
18             "a2"=> $row["a2"],
19             "a3"=> $row["a3"],
20             "a4"=> $row["a4"],
21             "a5"=> $row["a5"],
22             "b1"=> $row["b1"],
23             "b2"=> $row["b2"],
24             "b3"=> $row["b3"],
25             "b4"=> $row["b4"],
26             "b5"=> $row["b5"],
27             "c1"=> $row["c1"],
28             "c2"=> $row["c2"],
29             "c3"=> $row["c3"],
30             "c4"=> $row["c4"],
31             "c5"=> $row["c5"]
32         );
33     }
34     return $grid;
35 }
```

Fig. 4.11 - Servizio di invio griglia all'interfaccia

```

57 // invio risultati al server, calcolo media punti ed inserimento eventuale
punteggio migliore
58 function sendResults($partita,$grid,$utente,$obj,$punteggio) {
59     global $db;
60     $oggetto = $obj['triple'];
61     $a=1;
62     $date = new DateTime();
63     $timestamp = $date->format('U');
64     foreach ($oggetto as $tripla) {
65         $soggetto = $tripla['soggetto'];
66         $predicato = $tripla['predicato'];
67         $oggetto = $tripla['oggetto'];
68         $intervallo = $tripla['intervallo'];
69         $corretta = $tripla['corretta'];
70         $query = "INSERT INTO triple (`id`,`partita`,`grid`,`soggetto`
,`predicato`,`oggetto`,`ordine`,`tempo_da_inizio`,`giusta`)
71         VALUES ('" . ($timestamp+$a) . "-" . $utente . "' , '" . $partita . "' , '" . $grid . "' ,
'" . $soggetto . "' , '" . $predicato . "' , '" . $oggetto . "' , '" . $a . "' , '" . $intervallo
. "' , '" . $corretta . "')";
72         //print $query;
73         $result=mysql_query($query,$db) or pLog(mysql_error());
74         if (!$result) {
75             $error[]="Impossibile aggiornare il database ";
76             return $error;
77         }
78         $a++;
79     }
80     $query = "INSERT INTO partita (`utente`,`id_partita`,`id_griglia`
,`punteggio`)
81     VALUES ('" . $utente . "' , '" . $partita . "' , '" . $grid . "' , '" . $punteggio . "')";
82     //print $query;
83     $result=mysql_query($query,$db) or pLog(mysql_error());
84     if (!$result) {
85         $error[]="Impossibile aggiornare il database ";
86         return $error;
87     }

```

Fig. 4.12 - Servizio di invio dati partita al server

L'interfaccia

Per rendere l'app compatibile con i più famosi sistemi operativi, si è scelto di adottare una soluzione cross-platform in grado di garantire il riutilizzo del codice a seconda che si tratti di un sistema operativo come Android o come Apple iOS. I vantaggi sono molteplici, ma soprattutto è determinante che un codice unico capace di coprire diverse piattaforme, dia la possibilità di minimizzare i tempi di sviluppo, di aggiornare, modificare e ottimizzare un'applicazione in maniera univoca e infine garantisca una buona manutenibilità che rimane sotto il pieno controllo dello sviluppatore.

Per ottenere questo risultato si è fatto uso del framework **Apache Cordova** [17] che permette di utilizzare gli standard HTML 5 e CSS 3 per scrivere e disegnare la propria applicazione. Tramite un set di **API** è possibile accedere alle funzioni native del device come fotocamera, accelerometro, GPS e combinato con l'uso di altri framework quali **jQuery Mobile** fa sì che l'applicazione venga generata semplicemente seguendo il tradizionale approccio di un'applicazione Web. Tutti i meccanismi di gestione client-side, come la creazione delle pagine HTML, del riconoscimento delle gesture e dell'elaborazione dei dati ricevuti dal server sono stati implementati con linguaggio JavaScript.

4.3 Funzionamento

All'accesso dell'applicazione la prima schermata che si presenta all'utente è quella del login. Nel caso non si disponesse delle credenziali necessarie per effettuare l'autenticazione, è possibile eseguire la registrazione inserendo un indirizzo email valido, un nickname identificativo e una password. Qualora la persona avesse dimenticato quest'ultima, viene in aiuto un servizio di re-invio tramite posta elettronica delle credenziali di accesso inserendo solamente l'email utilizzata durante la registrazione. Una volta effettuato il login, la schermata successiva offre un riepilogo sui migliori tre punteggi registrati dal gioco, suddivisi in due categorie: la prima ordina gli utenti in base al loro miglior punteggio in una singola partita, la seconda classifica i giocatori in base alla media migliore tra punteggi e numero di match. Un pulsante nell'header permette di visualizzare una classifica più generale dove è possibile scoprire la propria posizione attuale indicata a caratteri rossi per meglio distinguerla a prima vista. Nell'header, oltre al tasto classifica, è presente anche il tasto logout che permette di uscire ed eventualmente riconnettersi con un account diverso. Nel footer, invece, è presente un link che porta alla pagina del regolamento dove sono ben descritte tutte le regole per

un corretto utilizzo dell'applicazione. Infine, sempre nella schermata principale, è posto in evidenza il tasto play che ovviamente permette di iniziare la partita.



Fig. 4.13 - Schermata Login

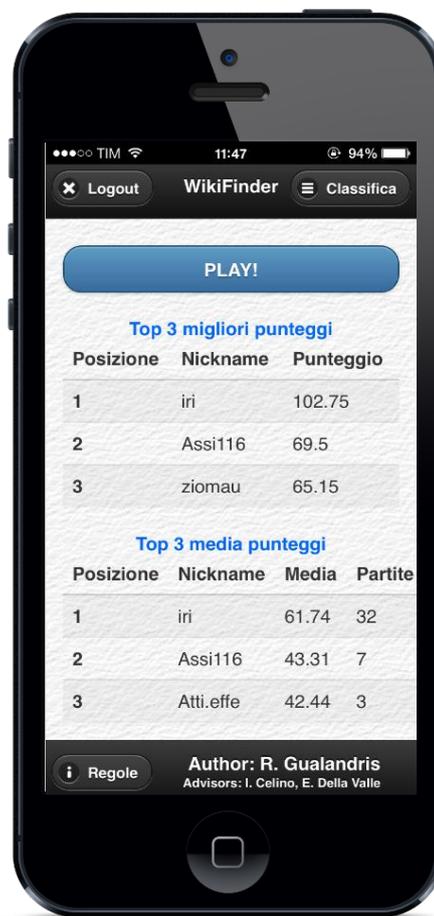


Fig. 4.14 - Schermata home

Una volta premuto il tasto play il match ha inizio e quella che l'utente si trova davanti è una griglia di 15 caselle divise in 3 colonne, ciascuna di un colore diverso per meglio distinguere i soggetti, i predicati e gli oggetti. Nel footer è presente il conto alla rovescia che parte da 60 secondi mentre nell'header il titolo fa spazio al punteggio che si aggiorna in tempo reale e al feedback visivo che ci mostra se la tripla selezionata è giusta o sbagliata. In caso di selezione positiva, viene mostrata la parola *GIUSTO* assieme all'icona di una spunta verde, nel caso di selezione negativa viene esposta la scritta

ERRORE assieme all'icona di una X rossa. Un feedback visivo è molto importante per rendere al giocatore l'idea della situazione del match che sta giocando, in tempo reale.



Fig. 4.15 - Schermata griglia di gioco



Fig. 4.16 - Schermata risultato finale

Per selezionare le triple all'interno della griglia, è obbligatorio partire toccando una casella contenente un soggetto e procedere a selezionare quella seguente trascinando il dito verso destra nelle direzioni consentite senza staccarlo dallo schermo. Nel momento in cui il dito viene sollevato dallo schermo, se sono state selezionate le tre caselle desiderate, la tripla viene immediatamente registrata e il punteggio viene

aggiornato. Per cui, se inavvertitamente avessimo scelto una tripla sbagliata, il sistema è programmato per considerarla ugualmente nel calcolo finale.

Il punteggio viene aggiornato in tempo reale durante ogni partita e segue un pre-determinato schema di calcolo. Per prima cosa il tempo totale a disposizione per ogni partita viene diviso in cinque fasce e ad ogni fascia è assegnato un numero di punti a scalare (sempre più basso con il passare del tempo: 100 nei primi 12 secondi, poi 90, 80, 70, 60 nelle successive fasce). A seconda dell'istante di tempo in cui viene individuata la tripla, il punteggio sarà differente: minore è il tempo impiegato per trovare una tripla giusta, maggiore è il punteggio guadagnato. Il punteggio guadagnato per ogni tripla esatta viene normalizzato dividendolo per il numero di soluzioni ammesse dalla griglia. In questo modo è possibile uniformare e confrontare punteggi ottenuti con un numero diverso di soluzioni. Ad esempio, se nella griglia ci sono 8 triple esatte, ipotizzando che il giocatore selezioni una tripla corretta nei primi 12 secondi, il punteggio sarà calcolato dal rapporto tra i punti assegnati alla prima fascia (100 punti) con il numero di soluzioni totali (8 triple): in questo caso la selezione della tripla corretta vale $100/8$ ossia 12,5 punti. Come anticipato in precedenza, questo calcolo viene fatto ogni volta che l'utente trova una tripla esatta e il punteggio viene sommato di conseguenza.

Qualora la tripla evidenziata non fosse corretta, al punteggio parziale viene sottratto un punto, così da scoraggiare una scelta a tappeto di tutte le combinazioni piuttosto che una selezione attenta e ragionata. Allo scadere del tempo limite di 60 secondi un messaggio di avviso indica quante triple esatte sono state individuate sul totale di quelle presenti e quanti punti sono stati realizzati; questi dati contribuiscono ad aggiornare le classifiche dei migliori giocatori.

5. VALUTAZIONE DI WIKIFINDER

Per poter analizzare e confrontare i dati raccolti durante la fase di test del gioco, è necessario stabilire alcuni criteri e metodi che permettano una valutazione precisa e oggettiva sulla qualità delle informazioni raccolte e la verifica dell'eventuale raggiungimento degli obiettivi del lavoro.

5.1 Metriche di valutazione

Per valutare WikiFinder, sono state definite diverse metriche che danno informazioni diverse e complementari per analizzare la bontà del risultato ottenuto. Oltre a metriche strettamente legate al *purpose* di questo GWAP (cioè al ranking di fatti in base alla popolarità), sono state usate anche metriche tipiche della valutazione dei GWAP.

Popolarità

Per stabilire quanto un fatto sia popolare, non è sufficiente contare il numero di volte in cui la tripla è stata selezionata durante le varie partite. Infatti, è necessario tenere presente che ogni utente è libero di giocare il numero di partite desiderate e che, una stessa griglia, può essere stata giocata un numero di volte minore rispetto ad un'altra. Per ottenere dei risultati che siano equilibrati e che non dipendano dal numero di match effettuati, il metodo più funzionale per stilare una classifica della popolarità delle triple messe in gioco è il calcolo della percentuale di selezioni della tripla in base al numero di volte in cui la tripla è stata presentata all'interno di una griglia. Una tripla avrà il 60% di popolarità se questa verrà selezionata almeno in sei partite su dieci in cui la griglia dove era contenuta è stata giocata. Una percentuale del 60% o del 70% ci permette di stabilire, sulla base della quantità di informazioni raccolte, una valutazione sensata dell'**indice di popolarità** di cui godono determinati fatti.

Allo stesso modo, è possibile calcolare la percentuale di frequenza con cui triple **non vere** sono state selezionate durante le partite. Questo indice, speculare al precedente, è allo stesso modo interessante perchè dà informazioni su quello che i giocatori pensano di conoscere o ritengono verosimile: l'errore costante di una tripla, da parte di più utenti, potrebbe avvalorare la validità della combinazione (oppure verificare eventuali errori durante la creazione delle griglie).

Un altro metodo di verifica utile per l'analisi, è la classificazione delle relazioni esatte in base a quante volte siano state selezionate nei **primi dieci secondi** del gioco. Come si può intuire, in un gioco di questo tipo è importante l'impatto che ha la partita nei primi momenti: le triple associate nei primi secondi, sono quelle che all'utente coinvolto nel match probabilmente sono più note e quindi popolari. Tener conto di questi dati, anche qualora non avessero raggiunto il 70% di popolarità globale, è utile e vantaggioso per le considerazioni statistiche.

Average Lifetime Play

L'attrattività del gioco ha un importante ruolo nella valutazione finale, da essa derivano buona parte dei meriti per la riuscita dello scopo finale. Von Ahn fa uso di questo fattore chiamato **Average Lifetime Play** o ALP [3] nello studio dei suoi risultati. Tale indice dipende principalmente dall'abilità di mantenere vivo l'interesse del giocatore rendendo l'esperienza interessante e divertente. L'ALP si calcola dividendo il tempo totale di gioco per il numero effettivo di giocatori, dove per numero effettivo si intendono solamente i giocatori che hanno completato almeno una partita. Con questo indice si è in grado di approssimare il tempo medio speso da ogni utente nel gioco.

Throughput

È possibile definire con il termine **Throughput**, il numero medio di task risolti per unità di tempo (dove il task è il compito *nascosto* nel gioco che si vuol far portare a termine dai giocatori). Nel caso del famoso ESP game [4], questo valore è stato misurato in circa 233 etichettature all'ora. Questo metro di valutazione è importante per stabilire le

performance raggiunte dal gioco nel risolvere i problemi per cui è stato concepito. Più questo indice è alto, minore è stato il tempo di gioco necessario per raggiungere l'obiettivo finale. Il risultato è determinato da una media globale calcolata su tutte le sessioni di gioco attraverso un ragionevole periodo di tempo.

Nel caso di WikiFinder, il task che si vuole risolvere è stabilire se un fatto espresso da una tripla sia o meno popolare. La velocità con la quale il gioco riesce a stabilirlo determina le prestazioni in termini di Throughput.

Players Expected Contribution

L'indice di **Expected Contribution** viene utilizzato per misurare la capacità di un GWAP di sfruttare i giocatori come risolutori di task [3]. Una volta conosciuti la media dei task risolti per unità di tempo (Throughput) e il tempo medio speso durante il gioco da ogni giocatore (ALP), è possibile combinare queste due metriche per definire il contributo di ogni utente al raggiungimento dell'obiettivo, espresso in numero medio di task risolti da ogni giocatore. Questo indice determina la media dei problemi che una singola persona è stata in grado di risolvere in un GWAP ed il risultato è ottenuto facendo il prodotto tra il Throughput e l'ALP. Questo indice serve a determinare quanti utenti sarebbero necessari per risolvere un dato numero di task.

Coefficiente di Spearman

Per poter verificare ulteriormente che i dati raccolti abbiano generato dei risultati sensati e realistici, l'uso del **coefficiente di Spearman** è fondamentale per calcolare la correlazione tra i dati di WikiFinder e quelli reali di Wikipedia. La correlazione indica la tendenza che hanno due variabili a *variare insieme* dove all'aumentare di una aumenta anche l'altra. Il coefficiente di correlazione di Spearman [18] indica la correlazione tra i rank di due variabili, cioè tiene conto dell'ordinamento relativo dei valori delle variabili, anziché dei valori assoluti che tali variabili assumono. Nel caso di WikiFinder, si tratta di mettere a confronto la popolarità delle risorse secondo i rank risultanti dall'applicazione e secondo quelli di Wikipedia e verificare se tra di essi esista una relazione.

5.2 Risultati della valutazione

In questa sezione sono riportati i risultati ottenuti dalla valutazione delle metriche definite nella sezione precedente.

Risultati relativi alla Popolarità

La Figura 5.1 riporta la tabella dell'indice di popolarità delle triple più conosciute. La classifica è ordinata in modo decrescente: nelle prime posizioni è evidenziata una popolarità del 100%, ad indicare che i rispettivi fatti sono state individuati tutte le volte che sono stati giocati in una griglia durante la partita. Nella tabella sono stati riportati dati fino ad un indice del 80%.

Percentuale	Soggetto	Predicato	Oggetto
100%	Michelangelo Buonarroti	birth place	Italia
100%	Romina Power	spouse	Al Bano
100%	Titanic (film 1997)	director	James Cameron
93%	Toscana	include	Provincia di Grosseto
92%	Aeroporto di Padova	city	Padova
92%	Alex Britti	occupation	Cantautore
92%	Università degli Studi di Padova	city	Padova
92%	Vicenza	province	Provincia di Vicenza
91%	Balla coi lupi	director	Kevin Costner
90%	Novara	province	Provincia di Novara
90%	Società Sportiva Calcio Napoli	league	Serie A
86%	Giuseppina del Belgio	birth place	Belgio
86%	Milano	Major	Pisapia
86%	Oscar Luigi Scalfaro	successor	Carlo Azeglio Ciampi
86%	Umberto Tozzi	occupation	Musicista
85%	Rock and roll	derivative	Rock
85%	Umberto Agnelli	state of origin	Italia
83%	Sant'Antonio Abate (Italia)	saint	Antonio abate
82%	Alexander (film)	starring	Anthony Hopkins
82%	Francesco Maurolico	birth place	Sicilia
80%	Filippa Plantageneta	birth place	Inghilterra
80%	Rieti	country	Italia
80%	Unione Calcio Sampdoria	league	Serie A
80%	Vicolungo	province	Provincia di Novara

Fig. 5.1 - Tabella indice di popolarità triple vere

Nella Figura 5.2 sono mostrati i risultati speculari alla tabella precedente ossia, l'indice di ripetizione delle triple false durante i match di gioco. La metodologia di calcolo è la stessa della precedente ma, come si può osservare, la percentuale può essere maggiore di 100 poiché è possibile che durante una stessa partita un giocatore abbia selezionato due volte la stessa tripla, convinto che questa fosse esatta. Con l'analisi di questi risultati è anche possibile osservare ed individuare eventuali anomalie ed errori all'interno delle scacchiere: il ripetersi dello stesso errore in più match potrebbe essere indice dell'esattezza della tripla evidenziata. Ad esempio, nella Figura 5.2, la tripla *Fabio Cannavaro – team – Nazionale italiana di calcio* ha ottenuto un indice dell'108%: l'errore rilevato non sussiste, perché la combinazione è ovviamente giusta; in questo caso, si è trattato di un bug nella fase di generazione delle griglie di gioco.

Percentuale	Soggetto	Predicato	Oggetto
115%	Elisa (cantante italiana)	occupation	Canto (musica)
108%	Fabio Cannavaro	team	Nazionale di calcio dell'Italia
108%	Ladri di biciclette	director	Ruggiero Mastroianni
100%	Black the Sun	genre	Rock
100%	Joe Doherty (singer)	occupation	Canto (musica)
93%	Matthew Stiff	birth place	Inghilterra
91%	Somewhere Out There (James Horner)	music composer	James Horner
88%	Alice Keppel	parent	Abraham Haskel Taub
88%	San Lorenzo	venerated in	Chiesa cattolica
85%	Piave	river mouth	Italia
85%	Provincia di Sondrio	province	Lombardia
85%	Z-Star	birth name	Alfred Tomatis
83%	Jandro (footballer)	currentclub	Futbol Club Barcelona
80%	Richard Grossman	birth place	Inghilterra
79%	Marco Carta	occupation	Musicista
79%	Mediterraneo (film)	music	Leopoldo Trieste
79%	Sebastião Lazaroni	birth place	Buenos Aires
78%	Conservative Party (Romania, 1880–1918)	party	Aleksandras Stulginskis
77%	Indagine su un cittadino al di sopra di ogni sospe	scrittore	Tullio Pinelli
77%	Ladri di biciclette	director	Federico Fellini
75%	Anna Maria Luisa de' Medici	resting place	Cimitero degli Allori

Fig. 5.2 - Classifica frequenza triple false

Un'altra considerazione che emerge osservando la tabella è che alcuni abbinamenti, che a molti giocatori sono sembrati veri, risultano effettivamente sbagliati. Un esempio concreto è nella tripla *Elisa – occupation – Canto (musica)*. L'erronea interpretazione della proprietà *occupation* in relazione con l'oggetto *Canto (musica)*, conduce all'errore il giocatore: in DBpedia la proprietà in questione per la cantante Elisa è legata a *Musician* (Musicista), mentre l'oggetto *Canto* è messo in relazione con la proprietà *Instrument* da cui si può supporre che *Canto*, ossia la voce, sia inteso come strumento utilizzato dall'artista. In questo caso i giocatori sono stati tratti in inganno dal fatto che il nome della proprietà fosse poco auto-esplicativo e dal fatto che considerare il canto uno strumento potrebbe sembrare non ovvio. Questo risultato quindi dà importanti feedback non sulla popolarità dei fatti in questione, ma sulla modellazione delle proprietà (e dei rispettivi domini e codomini) nell'ontologia di DBpedia.

La tabella nella Figura 5.3 riporta invece la classifica per numero di selezioni delle triple vere che sono state individuate nei primi dieci secondi di ogni partita. Questi dati evidenziano le associazioni maggiormente di spicco all'interno delle griglie di gioco. Ad

Numero Selezioni	Soggetto	Predicato	Oggetto
11	Macerata	province	Provincia di Macerata
11	Titanic (film 1997)	director	James Cameron
8	Michelangelo Buonarroti	birth place	Italia
8	Romina Power	spouse	Al Bano
7	Eros Ramazzotti	genre	Musica leggera
6	Alex Britti	occupation	Cantautore
6	Milano	Major	Pisapia
6	Novara	province	Provincia di Novara
6	Oscar Luigi Scalfaro	successor	Carlo Azeglio Ciampi
6	Vicenza	province	Provincia di Vicenza
5	Andrea Bocelli	occupation	Opera
5	Avatar (film 2009)	scrittore	James Cameron
5	Eros Ramazzotti	birth place	Italia
5	Juventus Football Club	chairman	Andrea Agnelli
5	Little Tony	home town	Tivoli
5	Macerata	region	Marche
5	Tevere	river mouth	Mar Tirreno
4	8½	scrittore	Federico Fellini
4	Andrea Pirlo	Club	Juventus
4	Balla coi lupi	director	Kevin Costner

Fig. 5.3 - Tabella triple vere individuate nei primi 10 secondi di gioco

una prima analisi, sembra effettivamente che tali triple possano essere considerate mediamente popolari per un utente medio.

Risultati della valutazione di WikiFinder come GWAP

Per calcolare l'ALP, sono stati considerati i **42** utenti che hanno giocato, durante il periodo di test, **411** partite totali. Considerando che la durata di una partita è di 1 minuto, è possibile approssimare il tempo totale di gioco a 411 minuti. Il calcolo finale dell'Average Lifetime Play ci dice che ogni utente ha giocato in media per **9,786 minuti**. Pur essendo molto lontano dai 91 minuti di ALP dell'ESP game, questo numero è comunque incoraggiante e indica che i giocatori di WikiFinder si sono divertiti abbastanza da giocare in media circa 10 round di gioco, pari a circa un terzo dei round disponibili nel gioco durante periodo di test.

Nel caso del Throughput, bisogna stabilire cosa indica che una tripla sia popolare (e quindi che il task sia risolto). È abbastanza naturale utilizzare il numero di volte in cui una tripla è stata selezionata come *proxy* della sua popolarità; per fare ciò è stato utilizzato come numero di task risolti il numero di triple selezionate almeno n volte e come tempo di risoluzione il tempo di gioco necessario a raggiungere n selezioni. Essendo il Throughput definito come spiegato nella sezione precedente, è stato calcolato questo valore al variare del numero n di selezioni, come visualizzato nella tabella 1.

<i>Numero di selezioni</i>	2	3	4	5	6	7	8
<i>Throughput</i>	0,2	0,153	0,123	0,107	0,098	0,091	0,088

Tabella 1 - Throughput in funzione del numero di selezioni

Dal momento che un semplice agreement tra 2 giocatori (come accade nella maggioranza dei GWAP di Von Ahn) non sembra indicativo della notorietà del fatto, si è

stabilita come soglia sul numero di selezioni la metà circa dell'ALP (approssimato a **5**). Nella tabella riportata in Figura 5.4, sono elencate alcune delle terzine che hanno soddisfatto il requisito ed è evidenziato il numero di partite impiegate per superare tale soglia. Sotto questa ipotesi, il numero di task risolti è **166** e la somma dei tempi di gioco per il superamento della soglia è **1547**. Quindi il Throughput di WikiFinder è uguale a circa **0,107** task risolti al minuto. Evidentemente questo valore non è molto alto e indica che probabilmente sono necessarie delle modifiche alla modalità di gioco, o per aumentare il divertimento degli utenti (che aumenterebbe anche l'ALP) o per migliorare il calcolo della popolarità, ad esempio basandosi su qualche meccanismo di valutazione dell'affidabilità dei giocatori. Infatti, come evidenziato nella tabella precedente, se si assumesse di potersi fidare al 100% di tutti i giocatori e si abbassasse la soglia del numero di selezioni al semplice agreement tra 2 giocatori, il Throughput aumenterebbe di quasi il doppio.

Infine, per quanto riguarda il calcolo del Player Expected Contribution, il risultato è dato dal prodotto dei due indici calcolati in precedenza: $ALP \times Throughput$. In questo caso il valore è uguale a **1,05** task risolti per giocatore, in linea con i valori precedenti.

Numero Partite Impiegate	Soggetto	Predicato	Oggetto
5	Alex Britti	occupation	Cantautore
5	Balla coi lupi	director	Kevin Costner
5	Cerro Maggiore	province	Provincia di Milano
5	Francesco Maurolico	birth place	Sicilia
5	Michelangelo Buonarroti	birth place	Italia
5	Milano	Major	Pisapia
5	Provincia di Sondrio	subdivision name	Lombardia
5	Rock and roll	derivative	Rock
5	Romina Power	spouse	Al Bano
5	Titanic (film 1997)	director	James Cameron
5	Umberto Tozzi	occupation	Musicista
5	Università degli Studi di Padova	city	Padova
5	Vicenza	province	Provincia di Vicenza
5	Vicolungo	province	Provincia di Novara
6	Aeroporto di Padova	city	Padova
6	Alexander (film)	starring	Anthony Hopkins
6	Eros Ramazzotti	birth place	Italia
6	Germán Beltrán	clubs	Girona Futbol Club
6	Juventus Football Club	chairman	Andrea Agnelli
6	Lo sceicco bianco	starring	Alberto Sordi
6	Look Through My Eyes	genre	Musica leggera
6	Novara	province	Provincia di Novara
6	Oscar Luigi Scalfaro	successor	Carlo Azeglio Ciampi
6	Piemonte	leader party	Lega Nord
6	Rieti	country	Italia
6	Sant'Antonio Abate (Italia)	saint	Antonio abate
6	Società Sportiva Calcio Napoli	league	Serie A
6	Toscana	include	Provincia di Grosseto
7	Andrea Bocelli	occupation	Opera
7	Andrea Pirlo	Club	Juventus

Fig. 5.4 - Tabella triple popolari che sono state selezionate almeno da 5 giocatori distinti

Effettiva popolarità tramite confronto con una *ground truth*

Per stabilire l'effettiva veridicità dei risultati ottenuti, è stato confrontato l'output di WikiFinder con una *ground truth* per valutarne la correlazione, utilizzando il coefficiente di correlazione per rank di Spearman definito nella sezione 5.1.

L'ipotesi principale è che il rank ottenuto con WikiFinder corrisponda all'effettivo rank di popolarità dei soggetti delle triple. Per verificarlo, è stato scelto come *ground truth* della popolarità il numero di visite che i soggetti delle triple hanno avuto sulle versioni inglese e italiana di Wikipedia. Per ottenere queste informazioni è stato usato il sito <http://stats.grok.se>, che mostra le statistiche e il numero di visite per ogni articolo di Wikipedia in varie lingue, potendo definire un lasso di tempo in cui effettuare la ricerca. In questo caso è stato preso in esame il mese di ottobre 2013 che corrisponde al periodo di test dell'applicazione.

Una volta calcolati il rank dei soggetti come risultato da Wikifinder (dove ha rank più basso il soggetto che è stato selezionato più volte dai giocatori) e i rank delle pagine corrispondenti su Wikipedia (dove ha rank più basso la pagina che è stata visitata più volte), ci si è chiesto se il rank di Wikifinder sia un buon predittore del rank di Wikipedia. Per rispondere al quesito, è stato calcolato il coefficiente di *Spearman*. Questa valutazione dimostra che esiste effettivamente una correlazione positiva e statisticamente significativa tra i rank, sia tra quello di Wikipedia inglese e il rank di Wikifinder (coefficiente: 0.2780101, p-value: 0.01197), che tra il rank di Wikipedia italiana e il rank di Wikifinder (coefficiente: 0.2975131, p-value: 0.006989).

Come era prevedibile, la correlazione è leggermente più forte, e un po' più significativa, nel caso italiano, dal momento che i giocatori che hanno partecipato alla fase di test erano tutti italiani. Nelle Figure 5.5 e 5.6 sono riportati i grafici di tali correlazioni tra i rank, mettendo sull'asse delle ascisse il rank di Wikifinder e sull'asse delle ordinate il rank di Wikipedia. Geometricamente, il coefficiente di correlazione corrisponde grosso modo alla pendenza della retta di regressione lineare (in blu nei grafici, rappresentata insieme alla fascia grigia che corrisponde all'intervallo di confidenza).

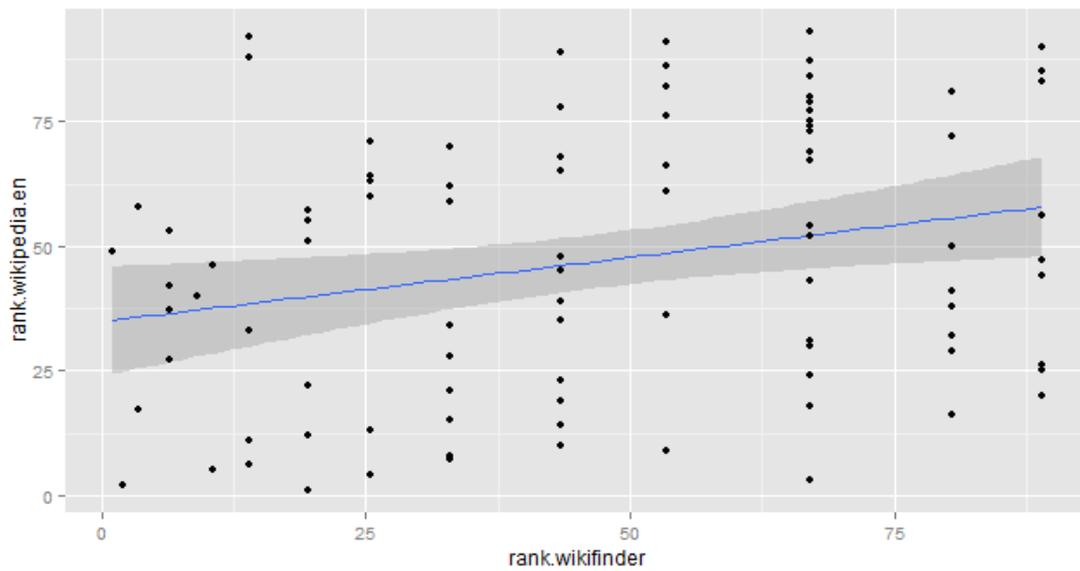


Fig. 5.5 - Grafico di correlazione tra rank WikiFinder e rank Wikipedia inglese

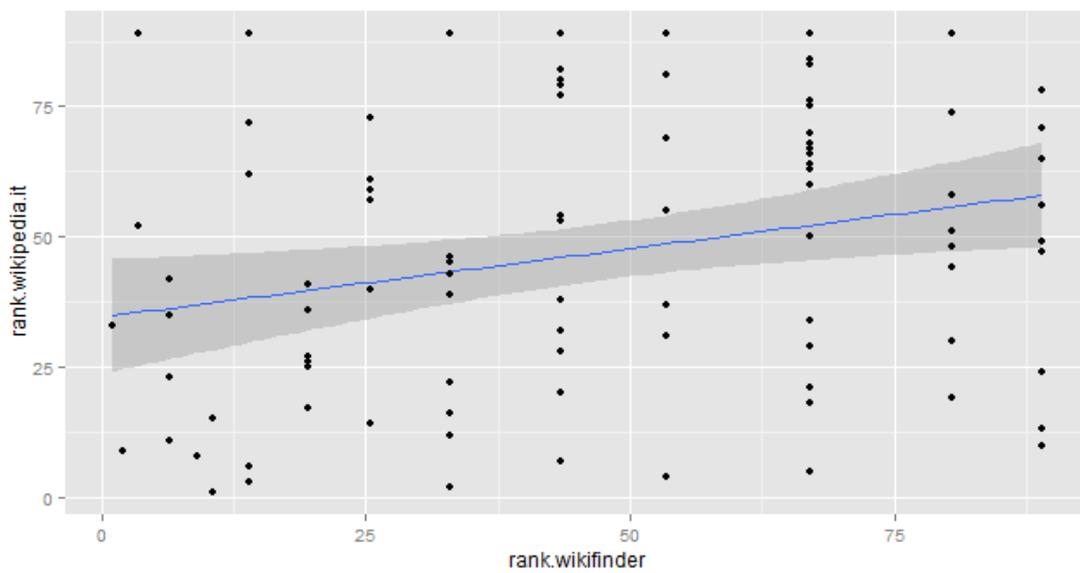


Fig. 5.6 - Grafico di correlazione tra rank WikiFinder e rank Wikipedia italiana

6. CONCLUSIONI

Con questo lavoro è stato possibile descrivere un metodo per ordinare i dati da DBpedia in base alla loro notorietà, utilizzando un approccio di Human Computation basato su GWAP.

Per prima cosa sono stati analizzati i modelli di GWAP presenti in letteratura. Studiati a fondo i loro requisiti e aspetti fondamentali, sono emerse le peculiarità utili per lo sviluppo del lavoro. Grazie a questa analisi è stato possibile progettare la struttura dell'applicazione WikiFinder.

Garantire il divertimento del giocatore e allo stesso tempo raccogliere i dati finali è stato il nucleo centrale dello sviluppo dell'app. Per questo motivo, sono state analizzate le possibili configurazioni per lo sviluppo dell'interfaccia, tenendo conto degli aspetti principali delle tecniche di GWAP.

Successivamente, è stata mostrata la soluzione finale dello sviluppo presentando in modo dettagliato tutte le sue parti e spiegandone il corretto funzionamento.

Infine, è stata compiuta un'analisi approfondita sui dati raccolti al termine del periodo di test, applicando diverse metriche che hanno permesso di verificare la buona riuscita del gioco rispetto ai requisiti posti all'inizio del lavoro.

Dopo le opportune considerazioni è emerso che i risultati raccolti durante il periodo di prova del gioco sono rappresentativi dell'effettiva popolarità dei fatti. Tali valori hanno permesso di fare un quadro dettagliato sia di ciò che è noto sia di ciò che è sconosciuto agli utenti partecipanti. Nonostante alcune risorse potessero essere piuttosto sconosciute rispetto ad altre e, considerando l'ostacolo della lingua inglese con cui erano riportati alcuni termini, il giudizio finale è soddisfacente come dimostrato dalla valutazione inclusa nel capitolo precedente.

6.1 Sviluppi futuri

L'applicazione allo stato attuale necessita ovviamente di alcuni miglioramenti e accorgimenti per rendere ancor più significativa e divertente l'esperienza di gioco. Alcuni spunti per un futuro lavoro potrebbero essere i seguenti:

- prendere come riferimento non solo il database di DBpedia inglese ma anche quello di DBpedia italiana, attualmente in costante sviluppo, in modo da avere due versioni dello stesso gioco;
- eseguire un controllo delle proprietà inserite nelle griglie di gioco, evitando di prendere in considerazione quelle poco chiare che possano creare confusione al giocatore;
- affinare l'algoritmo di creazione delle griglie inserendo un controllo che permetta di individuare quelle triple vere venutesi a creare inavvertitamente con il processo automatico di riempimento delle caselle: durante le partite è capitato di incontrare una tripla effettivamente esatta non segnalata come giusta;
- mostrare al termine del match tutte le triple che non sono state scoperte dall'utente, in questo modo l'utente sarà in grado di apprendere informazioni fino a quel momento sconosciute e avrà un ulteriore *reward* di tipo informativo oltre al divertimento del gioco. Successivamente, si potrebbero rimettere in gioco queste triple in nuove partite per verificare se l'insegnamento ricevuto in precedenza è stato colto dal giocatore, ma così facendo si perderebbe lo scopo per cui è concepito il gioco;
- evitare di inserire nelle griglie associazioni che siano semplici da trovare poiché intuitive per il tipo di soggetto e di proprietà utilizzate (ad esempio, *Macerata – province – Provincia di Macerata* è una tripla di questo tipo). Questa situazione infatti, ha un impatto negativo sullo scopo dell'applicazione poiché il giocatore non seleziona la tripla in base alla notorietà del fatto ma semplicemente per una certa intuitività. L'idea futura potrebbe essere quella di sviluppare un gioco diverso in cui lo scopo è quello di computare i tipi che la proprietà accetta per soggetti e oggetti.

BIBLIOGRAFIA

- [1] CHRISTIAN BIZER - JENS LEHMANN - GEORGI KOBILAROV - SÖREN AUER - CHRISTIAN BECKER - RICHARD CYGANIAK - SEBASTIAN HELLMANN, *DBpedia: A Crystallization Point for the Web of Data*, «Journal of Web Semantics: Science, Services and Agents on the World Wide Web», Issue 7, 2009, pag. 154–165
- [2] CHRIS BIZER - RICHARD CYGANIAK - TOM HEATH, *How to publish linked data on the web*, [<http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/drafts/LinkedDataTutorial/>], (ult. cons. Novembre 2013).
- [3] LUIS VON AHN - LAURA DABBISH, *Designing games with a purpose*, Commun. ACM 51(8): 58-67 (2008)
- [4] LUIS VON AHN - LAURA DABBISH, *Labeling images with a computer game*, CHI 2004, pag. 319-326
- [5] LUIS VON AHN - RUORAN LIU - MANUEL BLUM, *Peekaboom: a game for locating objects in images*, CHI 2006, pag. 55-64.
- [6] LUIS VON AHN - MIHIR KEDIA - MANUEL BLUM, *Verbosity: a game for collecting common-sense facts*, CHI 2006, pag. 75-78
- [7] *Duolingo*, [<http://www.duolingo.com/>], (ult. cons. Novembre 2013)
- [8] PASCAL HITZLER - MARKUS KRÖTZSCH - BIJAN PARSIA - PETER F. PATEL-SCHNEIDER - SEBASTIAN RUDOLPH, *OWL 2 Web Ontology Language Primer (Second Edition)*, W3C Recommendation 11 December 2012 [<http://www.w3.org/TR/owl-primer>].
- [9] CHRISTIAN BIZER - TOM HEATH - TIM BERNERS-LEE, *Linked Data: The Story So Far*, Int. J. Semantic Web Inf. Syst. 5(3): 1-22 (2009).
- [10] *Lod Cloud*, [<http://lod-cloud.net/state/>], (ult. cons. Novembre 2013).
- [11] NADINE LUDWIG - JÖRG WAITELONIS - MAGNUS KNUTH - HARALD SACK, *Who Knows? - Evaluating linked data heuristics with a quiz that cleans up dbpedia*, in «Proc. of the 8th Extended Semantic Web Conference (ESWC)», 2011.
- [12] JÖRN HEES - THOMAS ROTH-BERGHOFER - RALF BIEDERT - BENJAMIN ADRIAN - ANDREAS DENGEL, *BetterRelations: Using a Game to Rate Linked Data Triples*, KI 2011: pag. 134-138.
- [13] *Ruzzle*, [<http://www.ruzzle-game.com/>], (ult. cons. Novembre 2013).
- [14] STEVE HARRIS - ANDY SEABORNE, *SPARQL 1.1 Query Language*, W3C Recommendation 21 March 2013, [<http://www.w3.org/TR/sparql11-query/>].
- [15] *JSON*, [http://json.it/javascript_object_notation_ajax.php], (ult. cons. Novembre 2013).

- [16] *PHP*, [<http://php.net/>], (ult. cons. Novembre 2013).
- [17] *Apache Cordova*, [<http://cordova.apache.org/>], (ult. cons. Novembre 2013).
- [18] CHARLES SPEARMAN, *The proof and the measurement of association between two things*, *Armer J. Psychol*, pag. 72-101.