ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

SEDE DI CESENA

FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI

CORSO DI LAUREA IN SCIENZE DELL' INFORMAZIONE

STUDIO DELLA VARIABILE UMANA NELL'INSERIMENTO DI DATI LIBERI

Relazione finale in

BASI DI DATI

Relatore

Prof. Dario Maio

Presentata da

Giuliano Vandi

Sessione

ANNO ACCADEMICO 2011 -2012

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

SEDE DI CESENA

FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI

CORSO DI LAUREA IN SCIENZE DELL' INFORMAZIONE

TITOLO DELLA TESI

STUDIO DELLA VARIABILE UMANA NELL'INSERIMENTO DI DATI LIBERI

Relazione finale in

BASI DI DATI

Relatore

Prof. Dario Maio

Presentata da

Giuliano Vandi

Sessione

ANNO ACCADEMICO 2011 -2012

INTRODUZIONE

Ogni risultato ottenuto con una macchina di Turing decidibile ha avuto inizio con una macchina di Turing che non lo è. Questa macchina è l'uomo.

Dalla definizione classica di macchina di Turing a 5 campi l'uomo non ne rispetta nessuno perché ognuno è alterato da molti fattori.

Fattori psicologici, condizioni ambientali, nuove esperienze condizionano le scelte del fattore umano.

Scrivere una tesi su questo argomento si ritiene utile per porre l'attenzione su un problema che forse viene troppo spesso ignorato ma che è ancora presente nonostante l'evoluzione informatica abbia raggiunto risultati importanti.

Questo fenomeno è ancora facilmente riscontrabile anche nei siti che pubblicano annunci di vendita. In questi casi si possono riscontrare errori di battitura che rendono impossibile localizzare il prodotto ma anche la mancanza di dati fondamentali come il numero di telefono che vanificano l'intento di chi propone l'inserzione.

Se è intuibile un errore di disattenzione nell'ambito professionale dove la ripetitività di una operazione è suscettibile di errori per una auto-ipnosi della mente che comanda l'operazione, risulta più difficile giustificare un errore nell'ambito della sfera privata in quanto di interesse personale.

I nuovi e famosi strumenti di ricerca impiegati nel settore del WEB sviluppatissimi visti gli interessi economici che li spingono non riescono a trovare queste informazioni, anche perché i siti di annunci non sono ancora search friendly e questi algoritmi non sono ancora utilizzati capillarmente su questi siti.

A queste considerazioni ci si è giunti dopo aver ricevuto l'incarico di analizzare e produrre una soluzione al problema della ricerca delle offerte di lavoro che pervengono ad un Ufficio di Collocamento.

Questo è dunque il motivo per cui si è scelto di investigare la variabile umana ancor prima di trovare una soluzione al problema proposto.

Nuove esigenze informative a partire dalla Information Retrieval (IR) e nuove ottiche nate dall'evoluzione della situazione generale richiedono lo studio di tutte le fasi del trattamento dell'informazione a partire dalla sua origine.

Fenomeni apparentemente slegati fra loro quali la presa di coscienza dell'importanza del capitale umano nel processo produttivo, la presa di consapevolezza di un mercato di riferimento più ampio, i mutamenti nel mercato del lavoro, l'introduzione del rating aziendale nel mondo bancario e nuovi strumenti informatici fanno emergere la necessità di valutare, in diverso modo e sotto diverse ottiche, imprese, processi e fattori produttivi. Occorrono all'uopo nuovi strumenti capaci di dar vita a banche dati ove l'informazione sia disponibile a livello disaggregato.

I cambiamenti in atto nel panorama economico-sociale nazionale e internazionale hanno fatto nascere la necessità di introdurre sistemi di valutazione di imprese e fattori produttivi. Alcune sommarie considerazioni lo dimostrano.

Oggi si ritiene che i 2/3 del valore della produzione sia dovuto all'apporto del capitale umano. Tuttavia, sia a livello macroeconomico che microeconomico mancava fino a pochi anni fa una stima del capitale umano come variabile statistica definita su individui e famiglie. Recenti studi hanno permesso di giungere a stime attendibili del capitale umano definito come quell'investimento in educazione e formazione professionale che genera il reddito di lavoro di lungo periodo della famiglia (Vittadini, Dagum, Costa, Lovaglio 2003).

Poco o nulla è stato invece fatto per stimare il capitale umano aziendale tanto è vero che, per l'azienda, l'investimento in capitale umano, a differenza dell'investimento in macchinari, è considerato quasi esclusivamente come spesa corrente nel bilancio.

Per poter studiare il capitale umano nell'ambito della politica economica in relazione alle altre variabili strategiche (reddito, ricchezza, debito ecc) e nell'impresa insieme alle altre variabili necessarie per definire gli investimenti e gli ammortamenti è necessario generalizzare e rendere fruibili a tutti questi nuovi metodi di valutazione del suo ammontare.

Simultaneamente si osserva che l'accumulo di conoscenze da parte del lavoratore nell'ambito del ciclo vitale, il ritmo più celere delle innovazioni e i conseguenti cambiamenti della legislazione del lavoro hanno comportato una crescita vertiginosa di mobilità orizzontale e verticale. Oggi la vita di un lavoratore è spesso un percorso tra diverse aziende e differenti mansioni e professioni.

Occorre perciò valutare la rispondenza delle caratteristiche di un lavoratore, che accumula senza soluzione di continuità conoscenze, alle mutevoli necessità del mondo delle imprese.

Lo studio dei risultati riscontrati su banche dati compilate senza l'ausilio di sistemi di controllo e degli annunci di vendita disponibili su internet hanno permesso di determinare che la variabilità della componente umana è troppo elevata.

Lo studio è quindi articolato su due fronti distinti e scollegati fra loro.

Attraverso un originale prospettiva della fase dell'inserimento dati si vogliono discutere le problematiche che portano a risultati inattesi durante la fase del recupero delle informazioni quindi da un lato si incarica di identificare gli aspetti cognitivi e psicologici che agiscono nella fase di inserimento dati.

Dall'altro si incarica di presentare una soluzione alla ricerca di mansioni e professioni nelle offerte di lavoro che le aziende richiedono ad un centro per l'impiego a San Marino.

Il testo ha inizio con una rassegna dei problemi che determinano l'incertezza nell'inserimento dei dati, presenta lo stato dell'arte a supporto delle basi

teoriche utilizzate e prosegue con una descrizione del problema principale, la codifica delle offerte di lavoro.

A dimostrazione dell'efficacia dell'algoritmo proposto si completerà con un campione significativo di risultati ottenuti.

Con quanto detto si auspica venga istituito un corso di laurea per aspetti cognitivi.

CAPITOLO 1 – VALUTAZIONE DEL PROBLEMA

1.1 INTRODUZIONE AL PROBLEMA

E’ risaputo che le problematiche riscontrate nel recupero delle informazioni nascono nel momento in cui viene progettato il sistema di acquisizione. Le energie concentrate per la fase di analisi del contenitore si esauriscono spesso prima della formulazione del progetto dell’interfaccia utente.

L’interfaccia utente dovrebbe tenere conto di problematiche interdisciplinari di natura cognitiva e psicologica per cui lo strumento informatico utilizzato diviene puramente un mezzo e non un fine.

In azienda spesso, gli operatori sono chiamati a fare più cose contemporaneamente, come inserire dati trasmessi al telefono, prendere fogli da un fascicolo per soddisfare le richieste di un utente, tutti fattori che appesantiscono il processo mentale aumentando la probabilità di errore.

Partendo dall’istante in cui il dito dell’operatore “cade” sul tasto che imputa i dati, cioè quell’ambiente costituito di variabili più o meno casuali (variabili umane che incidono sulla determinazione del valore del capitale umano disponibile in azienda) che incidono su chi inserisce i dati si studieranno le problematiche che incidono sul recupero di tali informazioni.

Discutere una tesi di informatica analizzando problematiche umanistiche non è sicuramente agevole, ma nella pratica quotidiana i risultati sono influenzati notevolmente da questa componente.

1.2 Valutazione del capitale umano

1.2.1 La valutazione dell'ammontare di capitale umano

Sotto il profilo statistico, coerentemente con la definizione data, il capitale umano (HC) viene definito come quella "variabile composita" non osservabile generata dagli indicatori formativi inerenti l'investimento in istruzione superiore il cui esito sulla capacità lavorativa è misurabile mediante gli indicatori riflessivi (Giorgio Vittadini - Valutazione del capitale umano, delle imprese, dei servizi e nuova Informazione statistica).

Per stimare il capitale umano sono state utilizzate i seguenti indicatori:

Indicatori formativi: Età; Sesso; Regione di nascita e di residenza; Stato civile; Anni di scolarità; Numero di figli; Area di residenza (regione e ripartizione geografica); Tipo di laurea; Voto; Anno di laurea; Età in cui si è entrati nel mercato del lavoro; Anni di contributi; Status lavorativo (dipendente, autonomo, ecc...); Tipo di Occupazione; Settore lavorativo.

Indicatori riflessivi: Ricchezza totale famiglia (attività reali + finanziarie); Reddito da lavoro (dipendente + autonomo - ammortamenti + pensione + aiuti CIG); Risparmio e Debito familiare; Grado di istruzione; Tipo di lavoro e settore lavorativo di ciascun genitore

Indicatori procedurali: Ripetitività della procedura, stress all'interno del posto di lavoro, quantità e attitudine personale alla moltitudine di variabili procedurali.

Il valore capitale umano oltre a fenomeni esterni all'operatore indicati precedentemente dipende anche da come è organizzata l'informazione da inserire nel sistema. Il paragrafo che segue mostra come la stessa informazione organizzata in maniera differente possa determinare una alleggerimento o un appesantimento del processo mentale dell'operatore che si riflette sul risultato finale.

1.3 DISTANZA LOGICA E SEMANTICA DELL'INFORMAZIONE

L'indecisione provocata da termini simili concorre con altri fattori alla creazione di un risultato diverso da quello aspettato.

Si ritiene di affermare che un risultato diverso da quello atteso è certamente un errore che l'informazione non deve avere.

Il livello a cui si posiziona l'indecisione provoca una propagazione dell'errore più o meno significativo.

Maggiore è la distanza semantica fra parole e maggiore è l'elaborazione mentale necessaria a processare il concetto e quindi anche la possibilità di errore.

Si vuole estendere il concetto di distanza semantica non solo agli aspetti verbali ma anche ai processi. Cioè si vuole intendere tutte quelle attività che concorrono all'azione che appesantiscono inutilmente e irrimediabilmente il processo.

La necessità di fare più cose contemporaneamente, come inserire dati trasmessi al telefono oppure l'insinuazione di una telefonata in una conversazione già avviata sono tutti fattori che appesantiscono il processo mentale e come tale aumentano la probabilità di errore.

1.3.1 Richiami a Modelli a rete gerarchica della Memoria Semantica

La memoria semantica si riferisce all'immagazzinamento e all'utilizzazione di conoscenze che riguardano le parole e i concetti, le loro proprietà e relazioni reciproche.

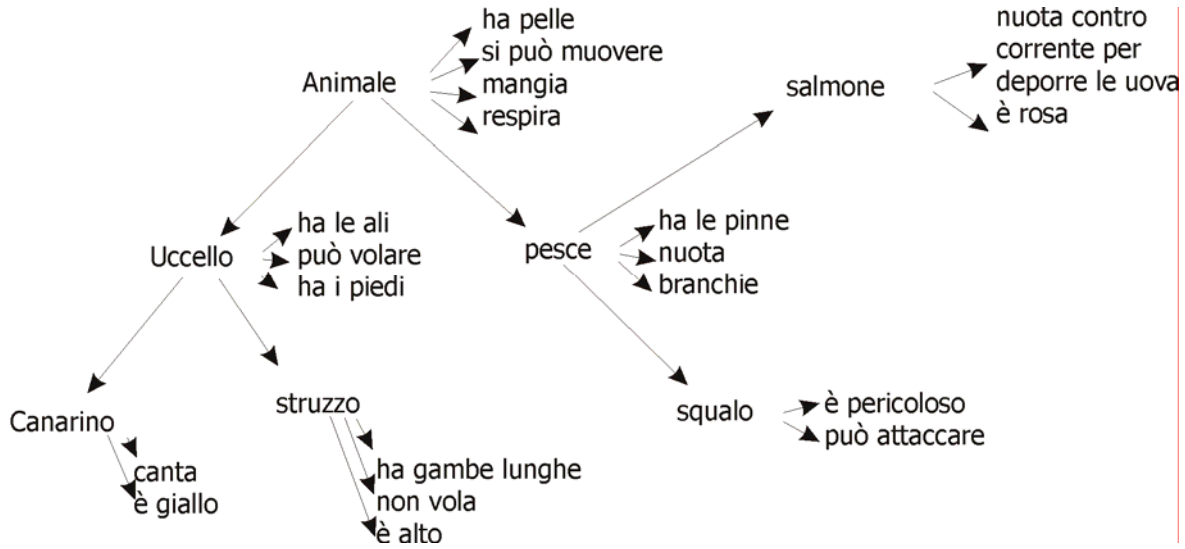
La memoria è quella parte (biologica o digitale) che conserva ciò che noi sappiamo (ad es.: i significati delle parole, di qui la sua denominazione).

Secondo i modelli a rete gerarchica della memoria Semantica presentati da Collins e Quillian [Alessandro Laudanna - Memoria semantica concetti processi semantici] può essere rappresentata con 3 tipi di oggetti:

- Unità (insiemi di oggetti, i "nodi" della rete).
- Proprietà (caratteristiche funzionali delle unità)
- Puntatori (relazioni tra unità o tra unità e proprietà)

Nella immagine sotto riportata i nodi sono rappresentati da: Animale, uccello, pesce, canarino, struzzo etc.etc.

Le proprietà sono le caratteristiche possedute dai nodi:ha pelle,ha le ali, ha le gambe etc. Mentre i puntatori sono i connettori logici orientati che uniscono i Nodi con le proprietà.



Graf. 1. Il modello storico utilizzato per descrivere e spiegare il il processo che provoca ritardo RT (Collins & Quillian 1969)

Maggiore è la distanza fra un nodo e la relativa proprietà e maggiore è la distanza semantica dell'informazione che provoca una elaborazione più lenta.

Ad esempio la frase “Un canarino mangia” richiede una elaborazione più lenta di “Un canarino è giallo”.

1.3.2 Modalità di rappresentazione

Secondo il modello di Collins e Quillian (rete gerarchica della memoria semantica) la rappresentazione avviene secondo 2 criteri:

1 Organizzazione gerarchica dei concetti

- Effetto di ampiezza di categoria:

  Es. “Un pettirosso e’ un uccello” vs. “Un pettirosso e’ un animale”

2 Principio di economia cognitiva: proprietà codificate al livello più alto possibile

- Economia cognitiva:

  Es. “Un uccello ha le piume” vs. “Un uccello ha la pelle”

Il recupero delle informazioni è visto come una ricerca delle “intersezioni” nella rete.

Poiché la ricerca richiede tempo, il tempo necessario per stabilire se un enunciato è vero o falso fornisce informazioni sulla struttura della rete.

Maggiore è la distanza semantica tra due nodi, maggiore il tempo necessario per recuperare l’informazione.

Il modello preso a riferimento presenta difficoltà e limiti pratici:

1.3.2.1 **Problemi empirici** - Limiti dell'assunzione gerarchica

Non è univocamente determinato l'ordine della gerarchia soprattutto con categorie ampie, ad esempio ritornando alla classe animali

- L'ordine: animale,mammifero,cane non rispetta l'assioma in quanto: "Un cane è un mammifero" più lento di "Un cane è un animale"
- Equivalentemente l'animale pettirosso appartiene allo stesso livello di struzzo ma: "Un pettirosso è un uccello" più veloce di "Uno struzzo è un uccello"

1.3.2.2 **Problemi empirici** - Limiti dell'economia cognitiva

Il principio di economia cognitiva afferma che gli esseri umani cercano di ottenere informazioni dall'ambiente circostante con il minimo sforzo, il che significa tendere a raggruppare gli elementi in categorie simili piuttosto che memorizzarle singolarmente.

In questo principio è fondamentale il dimensionamento verticale della categorizzazione, le cui caratteristiche più importanti sono fissate dal livello di dettaglio e/o dai membri inclusi per la formazione della categoria.

D'altro lato, la struttura del mondo percepito ci fa concentrare su quelle caratteristiche che normalmente concorrono, cioè, con la struttura correlazionale del mondo che ci circonda. Questo principio è fondamentale per la dimensione orizzontale della categorizzazione in cui ciò che prevale è la rappresentatività o la proto tipicità degli elementi che formano la classe.

Esempio	Distanza	RT	Esempio	Associazione	Ritardo (msec)
benjo-corde	0	1100	Pesciolino-labbra	debole	1210
arancio-commestibile	2	1060	arancio-commestibile	alta	1060

Tab. 1. Esempi di relazione in termini di tempo (RT) fra tasso di associazione e distanza gerarchica

I ritardi(RT) sono legati piu' al tasso di associazione che alla distanza nella gerarchia

1.3.2.3 **Problemi empirici -** Risultati su frasi negative non previsti dal modello

I risultati sperimentali condotti su proposizioni false o dalla risposta negativa portano a tempi di ritardo più lunghi mentre per frasi non plausibili portano tempi di ritardo più veloci. Questi risultati risultano inaspettati in quanto contrari a quanto previsto dalle performances del modello.

Frasi non plausibili:	Ritardo(msec)
La tigre ha una criniera.	1700
Un leopardo è un serpente.	1500

Tab. 2. Esempi di ritardo provocati da frasi non plausibili

1.3.2.4 Meccanismo non chiaro per la risposta a proposizioni false (ricerca autoterminante?)

1.3.2.5 Illusione di Mosè: nella memoria semantica le persone cercano corrispondenze non necessariamente esatte, ma anche solo approssimate (le meglio approssimate).

Nonostante le difficoltà del modello di Collins e Quillian, la procedura sperimentale consistente nel porre domande aí soggetti cosi che essi esplorino la propria memoria semantica si è dimostrata estremamente feconda. Un buon esempio di tali esperimenti è la dimostrazione di Reder e Kusbit [1991] della cosiddetta «illusione di Mosè». L'illusione di Mosè si riferisce al fatto che molte persone rispondono alla domanda: «Quanti animali di ciascuna specie Mosè portò con sé sull'arca?» dicendo «Due». Naturalmente, nessun animale venne portato sull'arca da Mosè. Era Noè.
L'illusione di Mosè è un fenomeno molto robusto, e può essere suscitata anche attraverso altre domande, come Reder e Kusbit mostrano con i seguenti esempi. Se la domanda è: «Di quale paese è stata presidente Margaret Thatcher?» la risposta sarà «Gran Bretagna», anche se Margaret Thatcher è stata primo ministro, non presidente. Quando viene chiesto: «Chi trovò la scarpetta di vetro che Biancaneve perse al ballo?» la risposta sarà «Il principe», anche se fu Cenerentola, e non Biancaneve, a perdere la scarpetta al ballo. E ancora, quando viene chiesto «In quale super-eroe si trasforma Clark Kent quando entra in una cabina dell'ascensore?» la risposta sarà «Superman», anche se Clark Kent in realtà si trasforma in una cabina del telefono, non dell'ascensore.

Le persone di solito non notano l'errore contenuto nella domanda, alla quale rispondono comunque. Come è possibile? Perché le persone rispondono a una versione riveduta e corretta della domanda? Per chiarire questi interrogativi, Reder e Kusbit hanno chiesto a dei soggetti di rispondere a una serie di domande, alcune delle quali contenevano errori sul tipo dell'illusione di Mosè. I soggetti però avevano modo di studiare alcune delle risposte corrette già prima che venissero poste loro le domande. Ciò avrebbe dovuto rendere le risposte più facili da recuperare, e avrebbe dovuto accrescere la probabilità che i soggetti rilevassero l'errore nella domanda. Tuttavia, questa tecnica non permise di ridurre l'illusione di Mosè. Ciò indica che le persone non ricercano una corrispondenza esatta tra l'informazione contenuta nella memoria semantica e quella contenuta nella domanda, ma s'accontentano di una corrispondenza approssimata. Cosi, quando

> devono rispondere a una domanda le persone spesso ricorrono all'informazione che più s'avvicina a quella richiesta. Se uno si limitasse a recuperare solo le corrispondenze esatte, si troverebbe spesso a non spiccicare verbo! Cosi, la migliore strategia generale sembra quella di recuperare l'informazione che, nella situazione attuale, costituisce la migliore approssimazione. La persona potrà non dare una risposta completamente corretta, ma darà almeno una risposta. II che forse è quel che molti studenti fanno quando rispondono alle domande d'esame.

Durante il recupero dell'informazione nella rete vengono attivati tutti i percorsi nei quali avviene la ricerca. L'attivazione si propaga dal nodo iniziale e poi si espande ai nodi vicini e ai nodi a questi collegati. Maggiore è l'attivazione di un nodo, migliore il recupero dell'informazione.

Si tralasciano ulteriori approfondimenti lasciando per inteso che l'argomento è stato trattato solo parzialmente e non esaustivamente perché esula dalla materia di studio.

Tutto quanto presentato fin ora permette per giungere alla significativa conclusione che una attenta progettazione del sistema informativo deve tenere conto di aspetti che le normali analisi di sistemi informativi non considerano. Ragionare a tutto tondo permette di ottenere risultati migliori e più coerenti, un valore aziendale maggiore con una partecipazione maggiore da parte del personale disponibile in azienda.

1.4 VALUTAZIONE DEI COSTI DI UNA INFORMAZIONE ERRATA

Disporre di una quantità significativa di informazioni sullo stesso argomento, inserito da persone diverse in tempi diversi e relative a problematiche diverse permette una prospettiva privilegiata dalla quale è possibile fare delle valutazioni significative diversamente non possibili. Scorrendo i record ci si

accorge di come siano visibili fenomeni più o meno importanti che possono provocare anomalie nel recupero del dato.

La presenza di uno spazio vuoto in un campo testo o di un campo non inizializzato non gestito determinano l'esito del risultato. Inoltre a fronte di una crescente mole di dati si tende a scoraggiare l'uso di operatori che effettuano la scansione carattere per carattere (Vedi operatore LIKE) limitando anche la possibilità di analisi. Molte politiche di gestione di database relazionali per evitare rallentamenti delle performance dei server non permettono l'uso dell'operatore LIKE.

Un sistema di acquisizione dati che non verifica l'informazione inserita liberamente presenta un errore molto significativo che dipende dal valore del capitale umano(dati rilevati sperimentalmente dal software gestionale dell'Ufficio del Lavoro di San Marino su campi di lunghezza 3 caratteri alfanumerici indicano un errore medio del 10%).

Occorre quindi identificare metodi, per una più veloce ed efficiente gestione dei contenuti testuali organizzando delle attività di *back office* che si incarichino di catalogare le informazioni per un recupero più efficiente.

Una organizzazione intelligente delle informazioni permette di ottenere risultati che al momento del concepimento del database non erano neppure richiesti. Ad esempio l'informazione sul recapito postale di un utente porta con se altre informazioni che permettono di determinare politiche mirate di marketing.

Oppure conoscere il recapito postale dei dipendenti e del relativo datore di lavoro su larga scala permette di determinare i flussi di traffico nelle ore di punta, ma anche lo studio sulla collocazione di servizi pubblici quali centri commerciali, palestre, stazioni di rifornimento etc etc.

Software che permettono l’inserimento di informazioni tramite campi testo liberi non certificati permettono maggiore espressività, ma è facile prevedere anche una proliferazione di errori tipografici notevoli.

Il settore della Pubblica Amministrazione soggetto a frequenti, inaspettate e non concordate variazioni dell’assetto normativo può trarre beneficio dal disporre di campi testo libero per limitare gli interventi di manutenzione del software.

D’altro canto disporre di una interfaccia i cui valori vengono selezionati da un insieme seppur teoricamente garantisti delegano l’inserimento del valore scelto ad un solo gesto.

Probabilmente distribuire l’inserimento su più operazioni permette maggior controllo sul dato inserito. A questo titolo strutturare un inserimento dati a livelli di tipo TOP-DOWN permette maggior controllo sul valore inserito e un miglior sistema per identificare gli errori.

1.5 STUDIO DI UN CASO PRATICO (DATO PRODOTTO DA UN UFFICIO PUBBLICO)

L'ufficio oggetto dello studio è un Centro per l'Impiego a San Marino (chiamato Ufficio del Lavoro). L'Ufficio è stato scelto per la sua disponibilità e per le caratteristiche dell'informazione.

Il progetto ha portato alla luce criticità fino ad allora inesplorate che hanno portato un notevole vantaggio al lavoro dell'Ufficio stesso.

1.5.1 Introduzione al problema e definizione dell'ambito di lavoro

Nella Repubblica di San Marino l'avviamento al lavoro è una Funzione Pubblica che viene esercitata dall'Ufficio del Lavoro e in particolar modo dalla sezione Collocamento con mansioni riconducibili ad un Centro per l'impiego.
Si vuole informatizzare la gestione delle offerte di lavoro che aziende private inviano all'Ufficio.
Il metodo cartaceo archiviato in faldoni presenta limiti alla fruizione e alla condivisione delle informazioni necessarie. L'archiviazione in faldoni ha funzionato fino a quando le esigenze erano contenute, le professioni semplici e la quantità di offerte di lavoro (e di domande) erano gestibili manualmente; oggigiorno il metodo dell'archiviazione cartacea non è perseguibile.

Nell'approccio tradizionale la ricerca del candidato avviene tramite la compilazione di un modulo nel quale l'azienda identifica le caratteristiche del candidato e le trasmette all'Ufficio del Lavoro che le elabora e invia all'azienda quei candidati dotati di tutti o alcuni dei requisiti richiesti.
Successivamente l'Ufficio identifica le persone con le caratteristiche più idonee a soddisfare le richieste dell'azienda, individuando quei candidati che più si avvicinano ai requisiti richiesti.

Condizione necessaria per soddisfare le esigenze dell'Ufficio del Lavoro è mantenere massima fedeltà alla libera espressività delle esigenze della azienda e garantire attendibilità al risultato.
Lo sviluppo individua la necessità di codificare alcune caratteristiche chiave della figura professionale e di un successivo authoring dei risultati di ricerca.

L'attuale legislazione impone all'Ufficio di inquadrare le offerte di lavoro in una delle 6 liste previste: Laureati, Impiegati, Mano d'opera qualificata, Mano d'opera generica, Commercio, Magazzinieri e Autisti.
Inoltre, la modulistica in corso di validità al momento dell'analisi prevede l'acquisizione della mansione da esercitare, del livello di inquadramento previsto e un altro campo nel quale vengono descritte le attività da far svolgere al candidato.

Vista la confidenza acquisita nel tempo dalle aziende, nell'inquadrare le richieste in una delle 6 liste, si è scelto di utilizzare questa informazione come uno dei campi codificati, ma **sarebbe auspicabile codificare** (responsabilizzando il compilatore della richiesta) **la mansione richiesta**.
Ma analizzando il significato delle parole che titolano le 6 liste ci si rende conto di come le professioni ad oggi più comuni portino a dei casi di incertezza per le quali occorre "interpretare", ma d'altronde il Centro preferisce procedere in questo modo piuttosto che individuare nuove tipologie di inquadramento.

Casi di relazione promiscua possono verificarsi fra i laureati che svolgono attività impiegatizie o Mano d'opera generica con alcuni ruoli nel campo del commercio (aiuto cucina,lavapiatti,addetta alle pulizie) o della gestione delle merci (addetto alla movimentazione delle merci, addetto alle consegne) per cui identificare chiaramente l'uno o l'altro non è descrivibile univocamente, ecco perché la fase di authoring.

Casi critici di indecidibilità possono risolversi analizzando il ramo di attività economica in cui opera l'azienda (anche se non sempre ben localizzato).
Risulta infatti più facile identificare il genere di "manodopera generica" se l'azienda che effettua la richiesta opera nel ramo della ristorazione piuttosto che nell'edilizia o nella metal meccanica.

CAPITOLO 2 - COME FUNZIONA GOOGLE

2.1 STATO DELL'ARTE - L'ALGORITMO USATO DA GOOGLE

Il modello semplificato di Google consiste di 5 parti fondamentali: il repository, il sistema di indicizzazione, il sistema di ricerca, il sistema di presentazione, il frontend.

2.1.1 Sistema di Indicizzazione

Il sistema di indicizzazione è responsabile di:

- Eseguire il crawling dei documenti accedendo ai siti web ed ad altri insieme di documenti
- Identificare le frasi nei documenti
- Indicizzare i documenti in accordo con le loro frasi

Con la parola documento si intende qualsiasi tipo di file che può essere indicizzato e recuperato dal motore di ricerca, pagina web, immagini, file multimediali, documenti di testo pdf, documenti di word, postscript, RTF, fogli excell, documenti power point, etc.

Nei paragrafi che seguono si utilizzeranno le seguenti convenzioni:

- per indicare un generico insieme di dati si utilizzeranno le lettere maiuscole
- per indicare un singolo elemento di un insieme di dati si utilizzeranno le lettere minuscole ed eventualmente un pedice per distinguere elementi diversi.

Ogni motore di ricerca dispone di un insieme di documenti anche detta corpus su cui è eseguita l'indicizzazione.

Il corpus risulta fondamentale per il motore di ricerca in quanto riconosce la correlazione semantica fra due o più frasi in relazione alla frequenza con cui esse compaiono insieme nei documenti.

2.1.2 Il crawling

Il crawling dei documenti presenti sul web è eseguito da un sistema di indicizzazione che prende il mone di crawler oppure spider.

Lo spider si basa su un insieme di URL da visitare forniti dall'URL server.

A sua volta l'URL server riceve gli indirizzi dagli utenti o dai link acquisiti automaticamente dalle pagine web.

Lo spider non valuta il contenuto dei documenti ma è in grado di preselezionare le pagine da acquisire in funzione di:

- l'esame dei domini presenti nelle liste SPAM
- riconoscimento delle spider – traps (ad esempio form di imput, ID di sessione delle URL , accesso ristretto da cookie, frame, pagine di autenticazione, cioè tutti i contenuti dinamici non percorribili)
- esame delle esclusioni impostate attraverso il file robots.txt o le meta tag analizzate a breve
- esame di altri criteri che possono influenzare il reperimento della pagina (ad esempio latenze e down time del sistema di host

Lo spider inoltre verifica periodicamente che le risorse siano ancora disponibili, non siano state aggiornate o abbiamo cambiato indirizzo.

In caso di variazioni lo spider aggiorna le sue informazioni e fa in modo di mettere a disposizione contenuti aggiornati.

Il sistema di indicizzazione utilizza istanze multiple di spider che operano su partizioni del web.

Ogni spider può leggere simultaneamente migliaia di pagine da centinaia di siti.

Una partizione può contenere centinaia di siti ospitati su host appartenenti a una certa classe di indirizzi IP .

L'attività di crawling ed indicizzazione può richiedere molto tempo cosi invece di ricorrere ad una scansione massiva, si preferisce una scansione incrementale in grado di aggiornare i dati sensibili con frequenza giornaliera o addirittura più frequente.

I dati letti dagli spider sono compressi e caricati dallo store server in un repository.

Il repository è gestito da una piattaforma softwere che permette di manipolare strutture dati di dimensioni dell'ordine dei Petabyte (Un Petabyte = 1000 Terabyte).

Il repository è un tipo speciale di struttura dati tabellare chiamato webtable.

Nella webtable ogni riga corrisponde ad un pagina web e le colonne corrispondono agli attributi della pagina.

Gli attributi possono essere:

- indirizzo URL della pagina web
- codice HTML della pagina
- il linguaggio
- data scadenza del contenuto
- tipo di pagina

- elenco back link
- anchortext della pagina
- indirizzo IP server ospitante

L’URL della pagina web è la chiave di accesso alle righe in altri termini si assume l’URL come l’indice di identificazione delle pagine.

Il motore di ricerca può aggiungere (ad uso interno) ulteriori colonne corrispondenti ad attributi che qualificano il contenuto di ciascuna pagina.

La webtable ha una struttura multidimensionale , per ogni cella della tabella è memorizzata la cronologia delle modifiche applicate al dato di quella cella.

Ad esempio la webtable può includere diverse versioni di codice HTML o le liste dei back link che si sono succedute nel tempo.

Le versioni memorizzate sono identificate in modo univoco da un timestamp.

La natura multidimesionale di questa struttura dati permette al motore di ricerca di esaminare le variazioni nel tempo di alcune proprietà caratteristiche delle pagine e di conseguenza costruire dei criteri di valutazione per questi elementi.

2.1.3 Identificazione delle frasi nei documenti

Non è nota l'implementazione del motore di ricerca di google, tuttavia le caratteristiche generali di indicizzazione e recupero del motore reale sono molto prossime a quanto ci si accinge a descrivere.

In questo paragrafo si discute un particolare tipo di indicizzazione chiamata indicizzazione per frasi.

L'identificazione consiste nella scansione del contenuto della pagina alla ricerca delle frasi valide.

Con la parola frase si intende una generica sequenza di uno o più termini distinta se marcata in qualche modo rispetto al testo circostante (tra virgolette, in grassetto, sottolineato etc).

Un frase si dice predditiva rispetto ad un'altra frase se in una collezione di documenti al sua occorrenza indica la presenza dell'altra frase.

L'identificazione delle frasi nei documenti è determinata dalle seguenti fasi:

- scansione ed estrazione
- Aggiornamento della matrice di co-occorrenza ed esecuzione del pruning secondo la sequenza:1) calcolo del grado di correlazione tra le frasi 2) eliminazione delle frasi non predditive 3) individuazioni delle frasi incomplete
- individuazione delle frasi correlate che hanno un alto guadagno informativo
- organizzazione in cluster.

2.1.4 scansione ed estrazione

L’identificazione delle frasi avviene scorrendo lungo il testo una finestra di scansione di lunghezza fissata.

La lunghezza della finestra è espressa come numero di parole che essa contiene in particolare la prima parola nella finestra è essa stessa una frase e le altre frasi sono costruite progressivamente aggiungendo alla prima parola i termini successivi.

L’esame del contenuto della finestra elimina la punteggiatura e non filtra gli articoli, le congiunzioni, le preposizioni e altre parole di uso comune.

Ogni singola frase è sottoposta a criteri di selezione che si basano sull’uso di contatori ricalcolati ogni volta che la frase è individuata nel testo.

I contatori sono:

- numero di documenti in cui la frase appare
- numero complessivo di occorrenze della frase
- numero do occorrenze che vedano la frase evidenziata mediante marcatori di testo

come risultato dell’applicazione dei criteri una frase è classificata come valida, da ignorare oppure possibile.

Le frasi possibili e le frasi valide sono memorizzate insieme ai propri contatori di lista: lista delle frasi possibili e lista delle frasi valide.

La lista delle frasi possibili è utilizzata nella fase di applicazione dei criteri di selezione, infatti, per ogni frase la lista memorizza i contatori su cui si applicano i criteri per valutare se può essere considerata valida e quindi spostata nella lista delle frasi valide.

Nel caso la frase identificata risulta valida si procede con la fase successiva (fase di aggiornamento della matrice di co-occorrenza e pruning).

2.2 Fase di aggiornamento della matrice di co-occorrenza e pruning

2.2.1 Passo 1: Determinazione del grado di correlazione tra le frasi

Il motore di ricerca mantiene un repository dedicato cioè una struttura dati chiamata matrice di co-occorrenza.

Tale matrice ricopre un ruolo determinante nell'indicizzazione.

Il numero di righe e di colonne della matrice è pari al numero di frasi valide identificate nella collezione di documenti.

Ogni volta che si identifica una nuova frase valida, questa è inserita nella matrice di correlazione aggiungendo una riga e una colonna.

In ogni cella della matrice sono memorizzati di dati che il motore di ricerca usa sia per l'indicizzazione che per la ricerca.

Se la frase valida era già stata identificata in precedenza allora non sono aggiunte righe ma sono aggiornati i dati nelle celle già presenti.

In ogni cella identificata con coppie di indici (j,k) della matrice è memorizzato l'indicatore di co-occorrenza della frase fj rispetto alla frase fk .

Questo indicatore è determinato attraverso l'applicazione al testo di una finestra di scansione secondaria che ha una lunghezza di più o meno 30 termini .

L'indicatore di co-occorrenza è pari al numero di volte che la frase fj compare nella finestra secondaria assieme alla frase f_k.

2.2.2 Passo 2: Eliminazione delle frasi non predittive (pruning).

Obiettivo di questa fase è consolidare l'elenco delle frasi valide affinché esso possa contenere solo altre frasi in grado di predire altre frasi correlate.

Questo consolidamento avviene calcolando il guadagno informativo che sussiste tra la frase f_j e le altre frasi valide f_k.

Il guadagno informativo esprime il rapporto tra la frequenza di co-occorrenza attuale e la frequenza di co.occorrenza stimata tra le frasi f_j e f_k.

In particolare, la corrispondenza di ciascuna colonna fk.della matrice di co-occorrenza si eseguono con i calcoli della funzione:

$$I(j,k) = \frac{R(f_j, f_k)}{T} * \frac{1}{E(f_j) * E(f_k)}$$

Dove T è il numero totale dei documenti della collezione esaminata; $E(f_j)$ e $E(f_k)$ sono rispettivamente le percentuali di documenti della collezione che contengono le frasi f_j e f_k.

Queste percentuali sono calcolate come il rapporto tra il numero dei documenti in cui la frase appare e T.

$R(f_j, f_k)$ è l'indicatore di co-occorrenza, mentre $\frac{R(f_j, f_k)}{T}$ corrisponde alla frequenza di co-occorrenza attuale fra f_j e f_k.

Il secondo fattore moltiplicativo $\frac{1}{E(f_j) * E(f_k)}$ corrisponde alla frequenza di co-occorrenza stimata fra f_j e f_k.

In breve il guadagno informativo esprime lo scostamento che sussiste tra il valore reale di co-occorrenza delle frasi f_j e f_k e il valore stimato.

Un valore del guadagno > 0 è un indicazione positiva del fatto che le frasi siano co-occorrenti mentre 0 indica che le due frasi non sono correlate.

Il guadagno informativo indicato con $I(j,k)$ è memorizzato per ogni coppia (fj,fk), nella corrispondente cella della matrice di co-occorrenza.

L'eliminazione delle frasi non predittive consiste nel verificare che almeno per la coppia (f_j,f_k), il guadagno $I(j,k)$ sia > di una certa soglia prefissata compresa tra 1.1 e 1.7 anche se il valore tipico è 1.5.

Se non esiste almeno una fase f_k tale per cui il suddetto guadagno informativo superi la soglia prefissata, allora la frase f_j è rimossa dalla lista delle frasi valide e si passa all'elaborazione della frase successiva.

2.2.3 Passo 3: Individuazione delle frasi incomplete

La lista delle frasi valide, a questo punto del processo di indicizzazione può includere delle frasi incomplete, ovvero frasi che sono in grado di predire solo estensioni di esse ma non di altre frasi.

Per esempio la frase incompleta "casa" può predire le frasi "casa in affitto" , "casa in vendita", etc.

Queste ultime frasi sono estensioni della frase incompleta "casa".

L'obbiettivo di questo passo consiste nella rimozione delle frasi incomplete che sono mantenute in una struttura dati dedicata chiamata lista delle frasi incomplete.

La liste delle frasi incomplete è utile durante la fase di valutazione delle query utente.

In particolare la query è comparata con le frasi incomplete e se corrisponde ad una frase incompleta il motore risalendo alle frasi estese suggerisce quelle complete o procedere direttamente alla ricerca.

Per individuare le frasi incomplete su ogni frase fj è eseguito uno string matching con le frasi f_k che predice (per cui il guadagno informativo è maggiore del valore di soglia).

Se tutte le frasi f_k sono estensioni di f_j allora f_j è incompleta e viene rimossa dalla lista delle frasi valide per essere caricata nella lista delle frasi incomplete.

Se ce almeno una f_k che non è estensione di f_j viene lasciata nella lista delle frasi valide.

Al termine di questa elaborazione l'elenco delle frasi valide consiste in un ampio numero di frasi estratte dalla collezione dei documenti.

Ciascuna delle frasi valide predice almeno un'altra frase che non è sua estensione.

Ogni frase valida è usata con frequenza accettabile per rappresentare in modo significativo concetti o idee espresse nella collezione dei documenti.

2.3 Fase selezione delle frasi altamente correlate.

Obiettivo di questa fase è selezionare le fasi correlate che hanno un alto guadagno informativo.

In criterio utilizzato per la selezione delle frasi è che il guadagno informativo deve essere maggiore del valore di soglia di 100.

Ricordando la definizione di guadagno informativo ciò significa che la frequenza di co-occorrenza attuale deve essere 100 volte superiore alla frequenza di co-occorrenza stimata (cioè le due frasi co-occorrono 100 volte in più rispetto alla stima.

Nel passo successivo viene generata una lista ordinata delle frasi correlate a f_j .

In questa lista le frasi sono ordinate per valori decrescenti in funzione del guadagno informativo rispetto a fj.

2.4 Fase organizzazione in cluster.

Un gruppo omogeneo di frasi che sono tra loro più fortemente associate di quanto lo siano con altre frasi in altri gruppi determinati tramite il calcolo dei rispettivi guadagni informativi è chiamato cluster.

I cluster sono usati durante il query time per organizzare i risultati della ricerca, permettendo di determinare quali documenti includere nei risultati e in quale ordine.

Determinare il cluster primario di una frase f_j costituito di frasi f_k passa per il calcolo dei guadagni informativi $I(j,k)$ e $I(k,j)$.

Se entrambi i guadagni sono maggiori di zero la frase fk appartiene al cluster primario f_j .

Procedendo in modo analogo su tutte le frasi rappresentative si costruisce l'intera struttura di cluster.

Il cluster primario di una frase può essere cluster secondario o terzo di un'altra frase etc.

Il cluster è rappresentato da una sequenza di bit .

Il numero di bit è pari al numero di parole rappresentative, il bit k-esimo è settato 1 se la frase f_k è nel cluster di f_j .

La sequenza di bit interpretata come numero decimale rappresenta l'identificatore univoco del cluster chiamato anche cluster id.

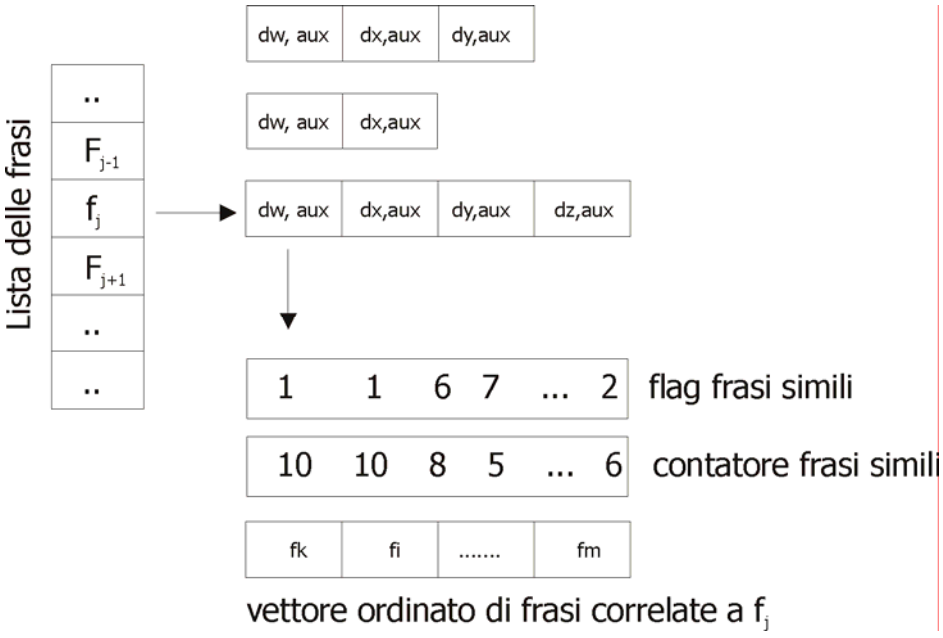
I dati relativi al cluster bit vector e al cluster id sono memorizzati nella matrice di co-occorrenza.

2.5 Indicizzazione dei documenti.

La struttura dati alla base del motore di ricerca è l'indice.

Nell'indice viene memorizzato l'elenco delle frasi valido contenuto nei documenti e per ogni frase fj viene mantenuta una lista (posting list) dei documenti nei quali la frase compare.

Ad ogni documento viene associato un insieme di informazioni ausiliarie relative alle frasi correlate al fj in ciascun documento.



Graf. 2. Rappresentazione della memoria delle frasi correlate

Con riferimento alla figura le informazioni ausiliarie sono:

- contatore frasi correlate: è una sequenza di numeri in cui ogni elemento corrisponde al numero di occorrenze nel documento dw di ogni frase

correlata a f_j Ogni elemento del vettore è in corrispondenza logica con una frase correlata a f_j .

- Flag frasi correlate: è una sequenza in cui ogni elemento è una coppia di bit . Ogni elemento del vettore è in corrispondenza logica con una frase correlata a f_j . Il primo bit indica la presenza della frase correlata del documento dw il secondo bit indica la presenza del documento dw di almeno una frase correlata secondaria. Una frase secondaria correlata e una frase correlata a f_k che a sua volta è correlata a f_j (si tratta di un secondo livello di correlazione).

La sequenza di bit " flag frasi correlate" di una frase fj rispetto al un documento dw può essere interpretata come un numero in base decimale.

In questo caso il valore numerico prende il nome di indice di rilevanza.

L'indice di rilevanza è tanto maggiore quanto più alto è il numero di bit impostati a 1 nel vettore, ovvero quanto più frasi correlate a f_j sono contenute nello specifico documento in esame.

In breve, l'indice di rilevanza esprime la rilevanza di una specifica frase per un dato documento.

Questa rilevanza è tanto più alta quante più frasi correlate alla frase in esame sono presenti nello stesso documento.

2.6 Sistema di ricerca

Il sistema di ricerca è responsabile dell'individuazione dei documenti rilevanti per la query di ricerca .

Il processo consiste in:

1. identificazione delle frasi nella query di ricerca
2. ordinamento dei documenti nei risultati della ricerca usando la presenza di frasi per influenzare l'ordine di importanza

2.7 Identificazione delle frasi nella query

La query di ricerca inserita dall'utente è analizzata in modo simile al come si è proceduto per l'indicizzazione del testo.

In pratica è eseguita una scansione con una finestra di testo lunga N=5 parole.

La finestra inizia con la prima parola nella query e si estende per 5 termini a destra.

Questa finestra è poi progressivamente traslata a destra di M-N termini dove M è il numero di parole della query.

Ad esempio la scansione della query " cercasi casa in vendita zona centro storico in provincia di Roma" prevede le seguenti frasi di query candidate estratte da questa finestra:

- "cercasi"
- "cercasi casa"

- “cercasi casa in”
- “cercasi casa in vendita”
- “cercasi casa in vendita zona”
- “cercasi casa in vendita zona centro”
- “cercasi casa in vendita zona centro storico”
- “cercasi casa in vendita zona centro storico in”
- “cercasi casa in vendita zona centro storico in provincia”
- “cercasi casa in vendita zona centro storico in provincia di”
- “cercasi casa in vendita zona centro storico in provincia di roma”

Per ciascuna frase di query candidata qj è verificata la presenza nella lista delle frasi incomplete.

Se la frase è presente nella lista il motori risale alle versione di frasi estese suggerendole all’utente o utilizzandole direttamente per la ricerca .

Per ogni frase di ricerca candidata qj ne è verificata la presenza all’interno dell’elenco delle frasi valide.

Ciascuna frase di query viene elaborata al fine di porre i termini che la compongono del giusto formato maiuscolo/minuscolo.

Ad esempio il testo “ stati uniti” è convertito in “Stati Uniti”.

Questo processo è chiamato capitalization.

Al termine delle verifiche suddette il motore dispone di una lista di frasi di query utilizzabili per il recupero dei documenti rilevanti.

La lista di query suddetta è ulteriormente ampliata usando le frasi correlate a ciascuna frase di query.

Ciò equivale ad estendere la ricerca fornendo all’utente risultati che includono termini e/o concetti aggiuntivi ma correlati.

Questo processo prende in nome di “query expansion”.

2.8 Recupero dei documenti rilevanti per la query

I documenti rispondenti alle frasi di query sono individuati intersecando le posting list associate a ciascuna frase di query, in altre parole si determinano i documenti in comune tra le frasi di query che sono contenute nell’indice.

Date due qualsiasi frasi di query q1 e q2 e le rispettive liste ordinate di frasi correlate Qr_1 e Qr_2 ci sono tre possibili casi di intersezione:

1. q_2 è una frase correlata a q_1
2. q_2 non è correlata a q_1, Qr_1 e Qr_2 non hanno frasi in comune
3. q_2 non è correlata a q_1 ma Qr_1 e Qr_2 hanno frasi in comune

Per individuare la relazione tra le due query q_1 e q_2 il motore recupera la posting list di q_1 ed esamina il vettore “frag frasi correlate” per ciascun documento dw , al fine di determinare se ce in bit corrispondente a q_2 .

Questo vettore di bit ci dice se q_2 è correlata a q_1 ed è presente del documento.

Nel caso 1 se questo bit non è settato a 1 per il documento dw allora vuol dire che q_2 non appare in quel documento ovvero q_1 e q_2 non sono correlate nel documento dw ,

Come risultato il documento dw può essere immediatamente eliminato perché non soddisfa entrambe del query.

Nel caso 2 se q_1 e q_2 non sono correlati è Qr_1 e Qr_2 non si intersecano allora il sistema procede con una intersezione delle posting list per individuare i documenti che hanno in comune.

Nel caso 3 se q_1 e q_2 non sono correlate ma Qr_1 Qr_2 si intersecano il motore esegue un intersezione delle posting list di q_1 e q2 finalizzate ad individuare i documenti che hanno in comune.

Se una frase è identificata come incompleta le posting list delle sue frasi vengono estese poi viene eseguita l'intersezione.

Il risultato dell'intersezione è un insieme di documenti rilevanti per la query dell'utente.

I documenti trovati possono essere ordinati in funzione del loro valore informativo.

2.9 Ranking dei documenti

2.9.1 Ordinamento basato sull'indice di rilevanza

I documenti trovati sono ordinati per valore decrescente dell'indice di rilevanza .

Di conseguenza i documenti contenenti la maggioranza delle frasi correlate alla query avranno un indice di rilevanza maggiore e saranno posizionati più in alto nella lista dei risultati.

Questo approccio è conveniente poiché sul piano semantico i documenti sono maggiormente correlati all'argomento della frase di query.

Questo approccio fornisce documenti che sono altamente rilevanti anche se i documenti stessi non contengono una elevata occorrenza delle parole specificate nella query.

Infatti in questo caso i documento pur non contenendo le parole della query (o contenendone poche) possono avere un ampio numero di frasi correlate ai termini della query e dunque possono risultare più rilevanti dei documenti che

contengono con un alta frequenza i termini della query ma che hanno poche frasi correlate.

2.9.2 Ordinamento basato su scoring

Lo scor finale di un documento è una combinazione di più risultati parziali pesati:

score finale = 30% * (body hit score) + 70% * (anchor hit score)

Il valore di boby hit score di un documento è in relazione con il numero di frasi correlate alle frasi query e che sono contenute nel documento.

Il valore di anchor hot score di un documento d è calcolato individuando, l'insieme R di documenti che contengono un link che punta a d e il cui anchor text del link è una frase di query.

In seconda istanza per ogni documento di R si moltiplica l'indice di rilevanza (chiamato outbound score component) per l'indice di rilevanza del documento d.

Questo prodotto prende il nome di inbond score component.

La somma di tutti gli inbond score component calcolati per ogni documento di R è l'anchor hit score.

L'inbond score componet misura il grado di attinenza per d della frase di query q usata come anchor text.

Se ci sono documenti che puntano a d usando la frase di query q e contengono le frasi correlate a q, allora maggiore sarà questa attinenza.

In altre parole il motore ritiene rilevante che un documento identificato con una query q (che contiene frasi correlate a q) sia puntato da altri documenti aventi link in uscita con anchor text q e contenenti frasi correlate a q.

I documenti sono ordinati per valori di score decrescenti.

Come ulteriore affinamento dei risultati della ricerca il motore può ulteriormente rimuovere i documenti.

In alcuni casi i documenti possono riguardare molti argomenti questo è vero soprattutto con documenti molto lunghi.

Generalmente gli utenti preferiscono documenti che sono altamente specifici su di un argomento particolare espresso dalla query.

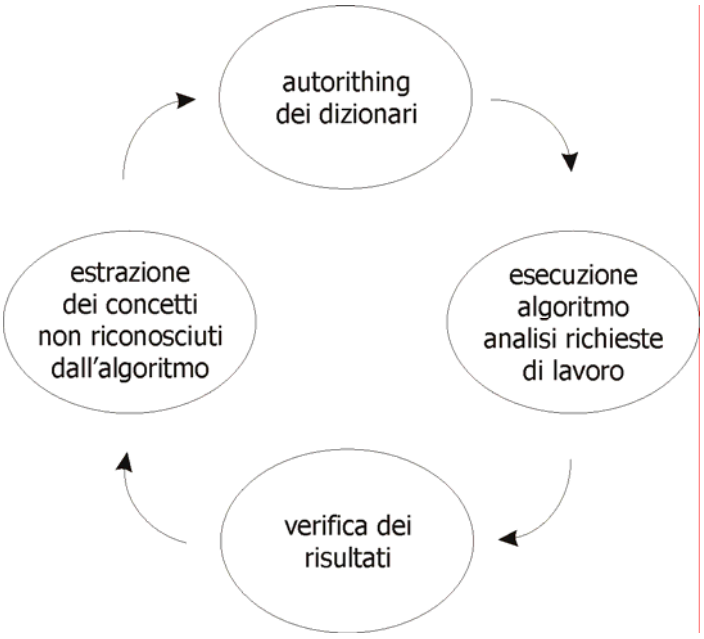
Per indiziare questo aspetto il motore usa le informazioni memorizzate nel “cluster bit vector” nella frase di query e rimuove tutti i documenti nei quali ci sono più di un certo numero di cluster.

In particolare questo numero di cluster è 2 e di conseguenza il motore di ricerca rimuove tutti i documenti che contengono più di due cluster, parametro che può essere impostato dall’utente nella query time.

CAPITOLO 3 - PRESENTAZIONE ALGORITMO

Sebbene per molti aspetti l'algoritmo di Google appare efficace per il problema presentato, della ricerca delle professioni (vedi paragrafo 1.5), le specifiche richieste analizzate da parte dell'Ufficio committente non permettono di applicarlo al problema esposto.

I motivi sono dovuti alla incapacità di distinguere concetti sulla base di parole analoghe riferite a concetti diversi e all'esigenza di garantire la supervisione tramite una fase di autorithing dei dati.



Graf. 3. Ciclo di funzionamento del procedura per il riconoscimento delle professioni tramite l'algoritmo presentato

Inoltre la struttura dati impiegata dall'Ufficio è profondamente differente dalla struttura dati necessaria alla ricerca di pagine web utilizzata da Google.

L’idea che ha permesso di giungere alla realizzazione di questo algoritmo è arrivata dopo aver appreso la magia operata da google e dall’aver fatto confidenza con il dato che ci si apprestava a lavorare.

Effettuare estrazioni su campi testo utilizzando parole chiave o parti di essa, estrazione con doppia negazione oppure verificare valori fuori range o non coerenti ha rappresentato un ottimo benchmark. Cosi l’algoritmo è un “semplice” processo batch di estrazioni.

3.1 ALGORITMO SOUNDEX PER AGGIRARE L'OPERTORE LIKE (DIGEST DEL MESSAGGIO)

Il cuore dell'algoritmo di ricerca presentato opera una codifica delle parole tramite un message digest della funzione SOUNDEX scritta specificatamente per questo problema.

L'operatore LIKE comodo ma costoso in termini computazionali viene cosi sostituito in un processo di Back-office che effettua un digest di tutte le parole che costituiscono il messaggio da analizzare.

L'idea base consiste in un sistema iterativo che:

1) scompone la stringa di testo in singole parole.
2) Esegue un digest di ogni parola attribuendo un codice simile ad un codice hash.
3) Memorizza in una tabella con tre campi: il numero di protocollo della richiesta, il codice di ogni singola parola e la parola stessa

Di seguito viene presentata la struttura della funzione soundex scritta in psudocodice:

```
funzione Soundex (S: stringa)

per ogni carattere di S
    se
    vale  "B", "F", "P", "V" allora Code = 1
    vale  "C", "G", "Q", "S", "Z" allora Code = 2
    vale  "D", "T" allora Code = 3
    vale  "L" allora Code = 4
    vale  "M", "N" allora Code = 5
    vale  "R" allora Code = 6
    vale  "0" allora Code = 7
    vale  "1", "2", "3", "4", "5", "6", "7", "8", "9", "J", "X", "K" allora Code = 8
    vale  " ", "" allora Code = 9
    accoda il valore di Code alla variabile di lavoro R
    fine ciclo
restituisce il codice hash dei primi 5 caratteri della stringa
```

Graf. 4. Funzione soundex

La funzione SOUNDEX rappresenta il cuore dell'algoritmo ragion per cui si descriverà più approfonditamente rispetto alla trattazione di altre funzioni accessorie.

La funzione SOUNDEX è caratterizzata da 3 punti salienti:

- codifica solo consonanti e numeri
- Mantiene una distanza significativa fra parole
- Considera solo la prima parte della parola perché più significativa

La scelta di non codificare le vocali è dovuta al fatto che sono le consonanti a costituire l'ossatura della parola.

Il non comprimere il codice generato fa in modo che le parole conservino la distanza semantica che avevano allo stato iniziale.

Inoltre dovendo evitare di comprimere la parola si è dovuto prendere la prima parte del codice che comunque rappresenta la parte più significativa.

Queste caratteristiche non sono comunque sufficienti a mantenere l'univocità del codice rispetto a 2 parole semanticamente differenti ma è un rimedio efficiente se integrato alle soluzioni messe in campo dalla funzione Match.

Parola	Codice SOUNDEX
approssimativamente	11622
prossimamente	16225
Programma	16265
Programmazione	16265
Precipita	16213
Precipitevolissimevolmente	16213

Tab. 3. Esempi dei valori restituiti dalla funzione SOUNDEX

Come mostrano i risultati delle performances della funzione SOUNDEX le problematiche legate sovrapposizione di codici riferite a parole diverse rendono necessaria la registrazione della parola stessa (terzo campo) per effettuare una successiva verifica “fine” del risultato ottenuto.

Con questa funzione è facile determinare casi di vicinanza sintattica fra le parole ma non la vicinanza semantica per la quale è necessario un ulteriore accorgimento che verrà presentato successivamente.

3.2 FUNZIONE EDITDISTANCE

La funzione difference è una particolare funzione editdistance che implementa una distanza di hamming fra la stringa di test e la stringa prototipo con in più una amplificazione dell’errore se il numero dei caratteri della stringa prototipo è maggiore del numero di caratteri della stringa di test

```
Funzione difference (modello :stringa, archivio :stringa) As Integer

Num_caratteri=minimo(modello e archivio)
Per ogni carattere di Num_caratteri
    Incrementa la variabile errori per ogni corrispondenza 1:1 fra modello e archivio che risulta diversa
Fine ciclo
Incrementa ulterioremente di un altro errore per ogni carattere esistente in più fra modello e archivio
Rem la differenza è calcolata relazionandola alla lunghezza del modello rispetto all'archivio
restituisce il valore della differenza di hamming rapportandola alla lunghezza delle stringhe
difference = errori * parte_intera(n_car_archivio / n_car_modello)
```

Graf. 5. funzione difference

3.3 FUNZIONE MATCH

La funzione match esegue lo string matching per trovare la miglior approssimazione di una parola test all'interno di una stringa indicata dalla variabile archivio.

In questo paragrafo la funzione match è riportata in pseudocodice per rendere più comprensibile la funzione di tale codice. Nei paragrafi successivi verrà presentata la stessa funzione scritta con un linguaggio più simile al visual basic per mostrare le differenze implementative.

La funzione restituisce una gamma di valori compresi fra 0-100 e sono fondamentalmente i valori restituiti dalla funzione difference.

Il valore 100 rappresenta un valore di fondo scala cioè l'errore massimo, un valore di fatto non raggiungibile per i contenuti che la funzione si trova a dover gestire.

Mentre il valore 0 indica il riscontro perfetto fra la parola test e una occorrenza della stessa all'interno del vettore archivio.

```
Funzione Match(test :stringa, archivio :stringa)
Se (n_car_archivio - n_car_test) < 0  allora Match = 100
Altrimenti
        Splitta la variabile archivio in un vettore di parole
        Per ogni parola del vettore di parole archivio di indice i esegui soundex(archivio[i]) e
        confrontala con soundex(test) e registra nella variabile errori il numero di errori minimi
        riscontrati in tutti i test restitutiti dalla funzione distance(test,archivio[i])
Fine se
Restituisce un valore equivalente al numero di errori
```

Graf. 6. Funzione match che individua quanti errori esistono fra una parola test e una qualsiasi parola contenuta nella stringa archivio

Di seguito è presentata la funzione MATCH implementata in un linguaggio di programmazione simile al visual basic.

```
Funzione Match(parola_test tipo stringa, archivio tipo stringa) tipo intero
NcarTest = numero caratteri ( parola_test )
```

```
NcarArchivio = numero caratteri ( archivio )

Cod_Test = Soundex( parola_test )
Arch = archivio

Ripeti
                    Errori = 100
                    Cod_Arch = Soundex( arch )
                    Se Cod_Arch = Cod_Test allora
                      Se errori>difference ( parola_test , arch ) allora /* difference() restituisce il
                    numero di caratteri diversi fra le 2 parole passate a parametro*/
                              errori>difference ( parola_test , arch )
                      fine se
                    A = A + NcarTest
fine se

i = space_before( A, arch )          /* restitisce la posizione in cui ha inizio la parola corrente
                    all'interno della frase arch */
f=space_after( A, arch )   /* restitisce la posizione in cui termina la parola corrente all'interno
                    della frase arch*/

arch = estrai_stringa( archivio , I , f-i+1 )

Finchè A< NcarArchivio – NcarTest + 1



Se NcarTest> NcarArchivio esci con valore restituito 100
```

Graf. 7. Funzione match 2° implementazione

3.4 FUNZIONI ACCESSORIE SPACE_BEFORE E SPACE AFTER

Il linguaggio di programmazione utilizzato non dispone di una funzione per eseguire lo SPLIT della stringa da analizzare, cosi sono state create delle funzioni che con altri linguaggi di programmazione non sono necessari. La mancanza di queste funzioni determina anche altre scelte operative che con altri linguaggi di programmazione “maggiormente attrezzati” non sono necessarie.

Le funzioni SPACE_AFTER e SPACE_BEFORE necessitano di 2 valori di input: posizione del carattere, stringa da analizzare e restituiscono rispettivamente il valore della posizione all’interno della stringa dove termina la parola e dove inizia.

Funzione space_after(pos: intero, archivio :stringa)
Conta quanti caratteri diversi dallo spazio vuoto si trovano verso destra nella stringa archivio a partire dalla posizione “pos”
Restituisce un valore dato dalla differenza fra pos e il conteggio effettuato

Graf. 8. Funzione space_after per individuare la fine della stringa

Funzione space_before(pos: intero, archivio :stringa)
Conta quanti caratteri diversi dallo spazio vuoto si trovano verso sinistra nella stringa archivio a partire dalla posizione “pos”
Restituisce un valore dato dalla somma fra pos e il conteggio effettuato

Graf. 9. Funzione space_before per individuare l’inizio della stringa

3.5 STRUTTURA DATI

In questa paragrafo vengono presentate le variabili e le strutture dati che intervengono.

Struttura dati	descrizione
Query	È la frase inserita dall'utente che è alla ricerca della categoria specifica
Query[]	È il Vettore che contiene le parole contenute nella frase inserita dall'utente
Mansione[]	Parola contenuta nell'archivio utilizzata per cercare la Mansione_certificata[] corrispondente
Lista[]	Insieme a Mansione[] contribuisce a determinare la Mansione_certificata[] corrispondente, l'utente può cercare la richiesta in più Liste
Mansione_certificata[]	È determinata dalla coppia di chiavi: Mansione[] e Lista[] ed è la parola contenuta nell'archivio che determina il risultato
Errori	E' l'indice di attendibilità della risposta trovata. È il numero di incongruenze fra la parola utilizzata per il match (presa da Query[]) e la parola contenuta nell'archivio delle parole chiave.Errori=100 rappresenta il massimo errore possibile in quanto le parole con senso compiuto possono avere un massimo di circa 20 caratteri, ciò significa che il risultato trovato per la Query[] non è attendibile. Errori=0 rappresenta la certezza del risultato in quanto esiste una corrispondenza 1:1 con la parola contenuta in Mansione[]
Archivio	È l'insieme delle parole utilizzate per la ricerca della soluzione

Tab. 1. Elenco degli elementi utilizzati dall'algoritmo

3.6 funzione MAIN

In questo paragrafo viene presentata la funzione main scritta in pseudocodice. Questa funzione fa da contenitore alle funzioni già presentate perché richiama direttamente o indirettamente tutte le funzioni scritte per la soluzione del problema proposto.

```
Inizializzare la tabella delle Richieste_Elaborate

Ripeti
Query = Concatena (Mansione richiesta dall'utente ," ", Altre caratteristiche richieste dall'utente)
QUERY[]=SPLITT( Query)
Err=100
Mansione[]=Estrae tutte le Mansioni della Lista[]
Per ogni Mansione[] e Err > 0
                Se Err < Match (Mansione , Query[] )    allora
                        Err = Match (Mansione , Query[] )
                        Parola_test = Mansione
                Fine se
Fine per

Fine ripeti
Inserisci nella tabella mansioni_revisionate i dati della richiesta originale e i campi Mansione e Err

Fine algoritmo
```

3.7 CODICE COMPLETO (VBA)

```
Function space_before(ByVal A As Integer, stringa As String) As Integer
Dim pos, i As Integer
Dim flag As Boolean

pos = 1
i = A
flag = False
If A > 1 Then
While i >= 1 And flag = False
If Mid(stringa, i, 1) = "," Or Mid(stringa, i, 1) = ";" Or Mid(stringa, i, 1) = "(" Or Mid(stringa, i, 1) = ")" Or
Mid(stringa, i, 1) = " " Or Mid(stringa, i, 1) = "/" Or Mid(stringa, i, 1) = "\" Or Mid(stringa, i, 1) = "-" Or
Mid(stringa, i, 1) = "." Then
pos = i + 1
flag = True
End If
i = i - 1
Wend
End If
```

```
If flag = False Then
pos = 1
End If
space_before = pos
End Function
Function space_after(ByVal A As Integer, stringa As String) As Integer
Dim pos, i As Integer
Dim flag As Boolean
pos = A + 1
i = A + 1
flag = False
If A < Len(stringa) Then
While i <= Len(stringa) And flag = False
If Mid(stringa, i, 1) = "," Or Mid(stringa, i, 1) = ";" Or Mid(stringa, i, 1) = "(" Or Mid(stringa, i, 1) = ")" Or
Mid(stringa, i, 1) = " " Or Mid(stringa, i, 1) = "/" Or Mid(stringa, i, 1) = "\" Or Mid(stringa, i, 1) = "-" Or
Mid(stringa, i, 1) = "." Then
pos = i - 1
flag = True

End If
i = i + 1
Wend
End If
If flag = False Then
pos = Len(stringa)
End If
space_after = pos
End Function
Function Match(modello As String, archivio As String) As Integer
Dim M, A, M1, A1 As Integer
Dim n_car_modello, n_car_archivio, errori, i, f As Integer
Dim risultato, risultato_storico As Double
Dim cod_modello, cod_archivio, cod_archivio1 As String
Dim arch, arch1 As String
n_car_modello = Len(modello)
n_car_archivio = Len(archivio)
If (n_car_archivio - n_car_modello) < 0 Then
Match = 100
Exit Function
End If
M = 1
A = 1
cod_modello = Soundex(modello)
arch = Trim(Mid(archivio, A, n_car_modello))
cod_archivio = Soundex(arch)
errori = 100
While A < (n_car_archivio - n_car_modello) + 1

Rem questa è l'originale -> arch = Trim(Mid(archivio, a, n_car_modello))
Rem questa invece è la nuova
i = space_before(A, archivio)
f = space_after(A, archivio)
If f < n_car_modello + i Then
f = space_after(n_car_modello + i - 1, archivio)
End If
arch = Trim(Mid(archivio, i, f - i + 1))
arch1 = Trim(Mid(archivio, f - n_car_modello + 1, n_car_modello))
Rem fine modifica
A = A + 1
If Mid(archivio, A + 1, 1) = "," Or Mid(archivio, A + 1, 1) = ";" Or Mid(archivio, A + 1, 1) = "(" Or Mid(archivio, A
```

```
+ 1, 1) = ")" Or Mid(archivio, A + 1, 1) = " " Or Mid(archivio, A + 1, 1) = "/" Or Mid(archivio, A + 1, 1) = "\" Or
Mid(archivio, A + 1, 1) = "-" Or Mid(archivio, A + 1, 1) = "." Then
A = A + 1
End If
While Mid(archivio, A + 1, 1) = "," Or Mid(archivio, A + 1, 1) = ";" Or Mid(archivio, A + 1, 1) = "(" Or
Mid(archivio, A + 1, 1) = ")" Or Mid(archivio, A + 1, 1) = " " Or Mid(archivio, A + 1, 1) = "/" Or Mid(archivio, A +
1, 1) = "\" Or Mid(archivio, A + 1, 1) = "-" Or Mid(archivio, A + 1, 1) = "."
A = A + 1
Wend
cod_archivio = Soundex(arch)
cod_archivio1 = Soundex(arch1)
If cod_archivio = cod_modello Or cod_archivio1 = cod_modello Then
If errori > difference(modello, arch) Then
errori = difference(modello, arch)
End If
If errori > difference(modello, arch1) Then
errori = difference(modello, arch1)
End If
A = A + n_car_modello
End If
Wend
If errori < 100 Then
Match = errori
Exit Function
End If
If cod_archivio = cod_modello Then
Match = difference(modello, arch)
Exit Function
Else
Match = 100
Exit Function
End If
End Function
Function difference(ByVal modello As String, ByVal archivio As String) As Integer
Dim errori As Integer
Dim caratteri As Integer
modello1 = UCase$(Trim$(modello))
archivio1 = UCase$(Trim$(archivio))
n_car_modello = Len(modello1)
n_car_archivio = Len(archivio1)
If n_car_modello > n_car_archivio Then caratteri = n_car_modello
If n_car_modello < n_car_archivio Then caratteri = n_car_archivio
If n_car_modello = n_car_archivio Then caratteri = n_car_archivio
errori = 0
For A = 1 To (caratteri)
If A <= n_car_modello And A <= n_car_archivio Then
If Mid(modello1, A, 1) <> Mid(archivio1, A, 1) Then
errori = errori + 1
End If
Else
errori = errori + 1
End If
Next A
Rem la differenza è calcolata relazionandola alla lunghezza del modello rispetto all'archivio
difference = errori * Int(n_car_archivio / n_car_modello)
End Function

Function Soundex(ByVal S As String) As String
Dim i As Long
Dim Code As Integer
```

```
Dim Last As Integer
Dim R As String
R = ""
Code = 0
Last = 0
S = UCase$(Trim$(S))
For i = 1 To Len(S)
Select Case Mid$(S, i, 1)
Case "B", "F", "P", "V"
Code = 1
Case "C", "G", "Q", "S", "Z"
Code = 2
Case "D", "T"
Code = 3
Case "L"
Code = 4
Case "M", "N"
Code = 5
Case "R"
Code = 6
Case "0"
Code = 7
Case "1", "2", "3", "4", "5", "6", "7", "8", "9", "J", "X", "K"
Code = 8
Case " ", """"
Code = 9
Case Else
Code = 0
End Select
If (i = 1) Then
R = Mid$(S, 1, 1)
Else
If (Code <> 0 And Code <> Last) Then R = R & Code
End If
Last = Code
Next i
Soundex = Mid$(R & "00000", 1, 5)
End Function
Private Sub Form_Open(Cancel As Integer)
Dim richieste, mansioni_revisionate As Recordset
Dim richieste_elaborate, richieste_elaborate1 As Recordset
Dim stoppo, errori As Integer
Dim db As Database
Dim parola_revisionata, parola, parola_storica, parola_revisionata_migliore As String
Dim query As String
Dim tabe As Recordset
Dim nome_query, par As String
Dim trovato As Boolean
DoCmd.SetWarnings False
DoCmd.OpenQuery "100 crea mansioni_revisionate_appoggio"
DoCmd.OpenQuery "101 aggiunge campo ID"
DoCmd.OpenQuery "102 sposta mansioni_revisionate_appoggio"
DoCmd.OpenQuery "103 elimina mansioni_revisionate_appoggio"
DoCmd.SetWarnings True

Set db = CurrentDb
num_pratiche = 0
Set richieste = db.OpenRecordset("_richieste", dbOpenTable)
Set richieste_elaborate = db.OpenRecordset("richieste_elaborate", dbOpenDynaset)
If Not richieste_elaborate.EOF Then
```

```
richieste.MoveFirst
While Not richieste_elaborate.EOF
richieste_elaborate.Delete
richieste_elaborate.MoveNext
Wend
End If
If Not richieste.EOF Then
richieste.MoveFirst
End If

While Not richieste.EOF
num_pratiche = num_pratiche + 1
errori = 100
risultato_mansione = 100
risultato_altre_caratteristiche = 100
parola_revisionata = ""
parola_prova = ""
parola_revisionata = ""
parola_revisionata_migliore = ""
Rem suddivide il contenuto di stringa in sottostringhe
stringa = richieste!elenco
trovato = False
ii = 0
While ii < Len(stringa) And Not trovato
ii = ii + 1
i = ii
While Mid(stringa, ii, 1) <> " " And Mid(stringa, ii, 1) <> "-" And ii <= Len(stringa)
ii = ii + 1
Wend
Rem PAR contiene la sottostringa da mettere in query
par = Trim(Mid(stringa, i, ii - i))



query = "select * from mansioni_revisionate where elenco like """ + par + """"
query = query + " or elenco like ""*" + par + """ or elenco like ""*" + par + "*""
query = query + " or elenco like """ + par + "*"" order by id"
Set mansioni_revisionate = db.OpenRecordset(query, dbOpenDynaset)
If Not mansioni_revisionate.EOF Then
mansioni_revisionate.MoveFirst
End If

While Not mansioni_revisionate.EOF And errori > 0

If richieste!Mansione = "Resp. Logistica" And mansioni_revisionate!parola = "Logistica" Then
ii = ii
End If

If Not IsNull(richieste!Mansione) Then
risultato_mansione = Match(mansioni_revisionate!parola, richieste!Mansione + " " +
richieste![Altre_Caratteristiche])
parola_prova = mansioni_revisionate!parola
parola_revisionata = mansioni_revisionate![mansione_revisionata]
Else
risultato_mansione = 100
End If
Rem If risultato_mansione = 100 Or risultato_altre_caratteristiche = 100 And errori = 100 Then
Rem stringa = MsgBox("ci sono più parole che sono corrette per la pratica num: " +
Str(richieste!ID_richiesta), vbOKOnly, "titolo")
Rem End If
```

```
If risultato_mansione < errori Then
errori = risultato_mansione
parola_revisionata_migliore = parola_revisionata
parola = parola_prova
End If

mansioni_revisionate.MoveNext
Wend
mansioni_revisionate.Close
richieste_elaborate.AddNew
richieste_elaborate![Cod_Op] = richieste![Cod_Op]
richieste_elaborate![Ragione_Sociale] = richieste![Ragione_Sociale]
richieste_elaborate!Mansione = richieste!Mansione
richieste_elaborate![Altre_Caratteristiche] = richieste![Altre_Caratteristiche]
richieste_elaborate!elenco = par
richieste_elaborate!Telefono = richieste!Telefono
richieste_elaborate!ID_richiesta = richieste!ID_richiesta
richieste_elaborate!Validita = richieste!Validita
richieste_elaborate!Operatore = richieste!Operatore
richieste_elaborate![Ultimo_aggiornamento] = richieste![Ultimo_aggiornamento]
richieste_elaborate!Sesso = richieste!Sesso
richieste_elaborate!Prot = richieste!Prot
richieste_elaborate![Data_prot] = richieste![Data_prot]
richieste_elaborate!N = richieste!N
richieste_elaborate!Qualifica = richieste!Qualifica
If errori < 100 Then
richieste_elaborate![mansione_elaborata] = parola_revisionata_migliore
richieste_elaborate![parola] = parola
Else
richieste_elaborate![mansione_elaborata] = ""
richieste_elaborate![parola] = ""
End If
richieste_elaborate!errori = errori
richieste_elaborate.Update
trovato = True
Wend
richieste.MoveNext
Wend
richieste.Close
richieste_elaborate.Close
End Sub
```

3.8 Limiti noti dell'algoritmo

Il valore delle risposte offerte dal programma dipende dalla correttezza e dalla vastità dei concetti memorizzati a disposizione dell'algoritmo.
L'algoritmo non utilizza reti neurali e non dispone di metodi per l'auto apprendimento.
Ogni concetto non riconosciuto che al termine dell'esecuzione dell'algoritmo avrebbe errori pari a 100 è inserito nella tabella della conoscenza (chiamata elenco_mansioni).
Questo macchinoso procedimento è ancora l'unico metodo per garantire il contenimento del proliferare di concetti sbagliati che porterebbero a risultati inaspettati.

3.9 Verifica delle performances dei risultati

Con l'algoritmo che viene presentato viene anche determinato se la risposta è considerata attendibile oppure no, dettaglio che l'algoritmo Google non determina con i requisiti necessari.
Questo valore è un numero fra 0 e 100 dove 0 rappresenta una risposta perfettamente identica rispetto a quanto trovato nella memoria dell'algoritmo (e quindi considerata una risposta certa), mentre 100 rappresenta una parola della quale l'algoritmo non è in grado di dare alcuna valutazione della risposta, mentre i valori intermedi sono valutabili singolarmente anche se l'esperienza maturata nell'uso del programma ha dimostrato come errori pari a 1 o 2 rappresentano una soluzione comunque valida.

Per descrivere il significato dell'errore si potrebbe associare la risposta che presenta errore 100 alla domanda posta dal software "cos'è questo?", per cui è necessario integrare la base della conoscenza del software con nuove informazioni.

Ad ogni modo il problema è in grado di distinguere parole significative per concetti differenti appartenenti (a patto di indicargli quali sono questi ambiti differenti), riesce quindi a distinguere alcuni concetti. Ad esempio la parola "impiegato amministrativo" rappresenta l'intenzione di cercare un "Laureato in Economia e commercio" nel caso in cui venga cercato nella lista dei laureati piuttosto che un "Ragioniere" se viene cercato nella lista degli impiegati.
Oppure se viene cercato un "Responsabile di cantiere" nella lista degli Impiegati o Geometri si intende un "Geometra" se cercato nella lista della Manodopera qualificata è un Muratore con esperienza.

3.10 RISULTATI

Di seguito sono riportati alcuni risultati significatici ottenuti eseguendo l'algoritmo riportato.
Sul lato sinistro della tabella sono indicati i valori di input mentre sul lato destro i valori di output.
In particolare sul lato sinistro la colonna Match indica la parola che è contenuta nell'archivio che rappresenta l'intelligenza dell'algoritmo.
I campi mansione, elenco e altre_caratteristiche rappresentano i campi che le aziende compilano e sottopongono all'ufficio.

Sul lato destro il campo mansione_elaborata rappresenta la mansione che l'algoritmo elabora mentre errori indica il numero di errori fra il valore in Match e mansione.

input				output	
Match	mansione	elenco	altre_caratteristiche	mansione_elaborata	errori
frutta	addetto orto frutta	COMMERCIO		Commesso - Banconista	0
frutta	Operatore Ortofrutta	COMMERCIO		Commesso - Banconista	0
banconiere	Banconiera	COMMERCIO		Commesso - Banconista	1
vendite	Addetto vendite	COMMERCIO		Commesso	0
Ragioniera	Segretario-a/Ragioniere-a	RAGIONIERI		Impiegato Amministrativo	1

tornitore	Tornitore	MQS	Esperienza su macchine a controllo numerico	Metalmeccanico - tornitore	0
comm.le	Impiegato/a comm.le	IMPIEGATI	sostituzione maternità, diplomata, capacità di lavoro in team e con agenti/fornitori ecc.	Impiegato Commerciale	0
commerciale	Responsabile Commerciale	LAUREATI	esperienza settore elettronico ed elettrico	Impiegato Commerciale	0
Controllo Gestione	Add. Controllo Gestione	LAUREATI	Laurea economia o ing. Gestionale - lingua inglese - trasferte geografiche	Controllo di gestione – Laurea economia	0
Amm.ne	Add.amm.ne del personale	IMPIEGATI	conoscenza programma paghe gips di beligotti, competenze amministrative	Impiegato Amministrativo	0
commerciale	commerciale senior	IMPIEGATI	competenza decennale settore packaging - lingua inglese e francese	Impiegato Commerciale	0
Match	mansione	elenco	altre_caratteristiche	mansione_elaborata	errori
gestionale	Project Manager	LAUREATI	Laurea Ingegneria Gestionale, perfetta conoscenza lingua inglese con esperienza all'estero	Esperto dei progetti - Ingegneria Gestionale	0
centralino	addetto centralino	IMPIEGATI	lingua inglese e tedesco	Receptionist	0
informatic	impiegato informatico	LAUREATI	laurea informatica o statistica - spec. business intelligence data warehouse	Impiegato informatico - statistica	1
operaio	Operaio	MG	Lavoro a turni anche notturno	Operaio	0

gestionale	add. logistica	LAUREATI	laurea in ingegneria gestionale, conoscenza inglese età 25 35 anni con esperienza	Esperto Logistica - Ingegneria Gestionale	0
centralino	Add. Centralino	IMPIEGATI	Ottima capacità relazionale. Ottima conoscenza inglese e tedesco, disp. Ai turni fino alle 18,30	Receptionist	0
sistemista	Sistemista Senior	LAUREATI	ottima conoscenza networking - vlan - lingua inglese -	Sistemista - Informatica	0
operaio	Operaio	MG		Operaio	0
contabilità	Impiegato/a	RAGIONIERI	conoscenza paghe, contabilità e dich. Redditi	Impiegato Amministrativo	0
Match	mansione	elenco	altre_caratteristiche	mansione_elaborata	errori
Rappresentante	rappresentante	IMPIEGATI	settore spedizioni e trasporti	Rappresentante	0
vendite	Addetta trattamento viso e trucco	COMMERCIO	vendita prodotti	Commesso	1
commessa	Commessa	COMMERCIO	20/21 anni	Commesso	0
portiere	Portiere notturno	COMMERCIO	esperienza	Portiere	0
computer	Segretario/a	IMPIEGATI	Con esperienza computer, disponibile festivi e week-end e inglese o altre lingue	Receptionist	0
tuttofare	Tuttofare	COMMERCIO		Tuttofare	0
computer	segretar io/a	IMPIEGATI	con esperienza computer,disponibilità festivi si e week-end	Receptionist	0
tuttofare	Tuttofare	COMMERCIO	Donna	Tuttofare	0

cuoco	cuoco - aiuto cuoco	COMMERCIO	cuoco - aiuto cuoco - pizzaiolo	Cuoco	0
lavapiatti	Lavapiatti	COMMERCIO	Servizio doppio turno	personale di cucina	0
cameriere	Cameriere	COMMERCIO	lingua inglese e russa	cameriere	0
cameriere	Cameriere	COMMERCIO		cameriere	0
produzione	add. Produzione	MG	solo da mobilità	Operaio	0
impiegato	Impiegato/a	IMPIEGATI		Impiegato generico	0
Controllo Gestione	impiegato/a	LAUREATI	laurea farmacia - CTF - controllo qualità	Impiegato Tecnico - Farmacia	0
Impiegato amm	impiegato/a	RAGIONIERI	pratiche d'importazione - monofase	Impiegato Amministrativo	0
operaio	Operaio	MG		Operaio	0
operaio	Operaio	MG	FINO 31/12/11	Operaio	0
Operaia agricola	operaio comune	MG		Operaio Generico	0
Match	mansione	elenco	altre_caratteristiche	mansione_elaborata	errori
Magazziniere	Magazziniere	MAGAZZINIERI	Utilizzo carro ponte e muletto, utilizzo computer	Magazzinieri	0
operaio	Operaio	MG	Lavoro a turni anche notturni	Operaio	0
operaio	Operaio	MG	disponibile lavoro a turni anche notturni	Operaio	0
meccanica	ing. meccanico	LAUREATI	lingua inglese	Ingegnere Meccanico - Ingegneria Meccanica	1
	Ragioniere/a	PERITI	con esperienza	Impiegato Amministrativo	0

Tab. 2. Incrocio dati in ingresso e risultati ottenuti dando applicazione all'algoritmo

Il report di seguito presentato è quanto ottenuto eseguendo il programma su un campione di dati reali di 1793 record.

I risultati ordinati per numero di occorrenze di mansioni omonime rilevate riportano anche il numero di errori, ovvero di incertezza sul risultato.

Errori dell'ordine di 1 sono da considerarsi comunque attendibili al 100% mentre per valori maggiori di 2 errori è necessaria una fase di autorithing.

Classi di grandezza di errori superiori a 4 necessitano inoltre una fase di autorithing ulteriore, cioè l'integrazione del dizionario di parole con nuove voci.

mansione elaborata	numero occorrenze	0 err	1 err	2-3 err	4-10 err	>5 err	incidenza della classe sul totale
Impiegato Amministrativo	135	46	65		12	12	7,53%
Operaio	124	104	20				7,48%
Impiegato Commerciale	123	120	1	2			8,02%
Commesso	118	99	19				8,36%
cameriere	112	102	10				8,66%
personale di cucina	93	90	3				7,87%
Impiegato generico	89	68	15	2	2	2	8,18%
barista	58	57	1				5,81%
Magazzinieri	56	54	1	1			5,95%
pulizie	54	54					6,10%
Autista	42	42					5,05%
Impiegato tecnico - Disegnatore	33	33					4,18%
Muratore	32	32					4,23%
Impiegato tecnico - Perito Informatica	27	25	2				3,73%
personale di cucina - pizzaiolo	25	25					3,59%
Impiegato Commerciale - Estero	24	24					3,57%
Laurea Economia	24	21			1	2	3,70%
Tecnico Riparatore	23	14	7		1	1	3,69%
Laurea Scientifica - Informatica	21	18	1	2			3,49%
Parrucchiera	21	18	3				3,62%
Estetista	19	19					3,40%
Laurea Scientifica - Ingegneria	15	14			1		2,78%
Tuttofare	15	15					2,86%
Autista - Addetto alle Consegne	14	14					2,75%
Idraulico	14	14					2,82%
Meccanico - auto	14	14					2,90%

mansione elaborata	numero occorrenze	0 err	1 err	2-3 err	4-10 err	>5 err	incidenza della classe sul totale
Assistente dentista	13	13					2,78%
Impiegato tecnico - Perito	12	12					2,64%
Commesso - Banconista	11	11					2,48%
Elettricista	11	11					2,55%
Imbianchino	10	10					2,38%
Laurea Scientifica	10	7	1	1		1	2,43%
Addetto	9	9					2,24%
Impiegato Marketing	9	9					2,30%
Impiegato tecnico	9	5	3			1	2,35%
Muratore - Carpentiere	9	9					2,41%
Addetto Manutenzione	8	8					2,19%
Autista - Operatore macch. Operatrici	8	8					2,24%
Impiegato Commerciale - Responsabile	8	7			1		2,29%
Metalmeccanico	8	8					2,35%
Impiegato tecnico - Geometra	7	6	1				2,10%
Impiegato Tour Operator	7	7					2,15%
Laurea Marketing	7	7					2,19%
Laurea Scientifica - Biologia	7	7					2,24%
macellaio	7	7					2,30%
Montatore Strutture Meccaniche	7	6	1				2,35%
OSS	7	7					2,41%
Commesso - Cassiere	6	6					2,11%
Impiegato tecnico - Perito Chimico	6	4	2				2,16%
Laurea Scientifica - Ingegneria Gestionale	6	6					2,21%
Operatori servizi sanitari	6	6					2,26%
Saldatore	6	6					2,31%
Addetto sicurezza	5	5					1,97%
Agente di Commercio	5	5					2,01%
Falegname	5	5					2,05%
gelataio	5	2	3				2,09%
Impiegato Settore Risorse Umane	5	5					2,14%
Impiegato tecnico - Perito Elettro Meccanico	5	5					2,18%
Impiegato tecnico - Perito Elettronica	5	5					2,23%
Laurea Giurisprudenza	5	5					2,28%
Operatore call center	5	5					2,34%
Responsabile	5	5					2,39%
Cablatore quadri elettrici	4	4					1,96%
Fabbro	4	4					2,00%
Facchino	4	4					2,04%
Impiegato tecnico - Perito Elettrotecnica	4	4					2,08%

mansione elaborata	numero occorrenze	0 err	1 err	2-3 err	4-10 err	>5 err	incidenza della classe sul totale
Laurea Scientifica - Architettura	4	4					2,13%
Laurea Scientifica - Ingegneria Elettronica	4	4					2,17%
Laurea Scientifica - Ingegneria Meccanica	4	1	3				2,22%
Lavandaia	4	4					2,27%
Montatore Strutture	4	4					2,33%
Ottico	4	4					2,38%
pasticciere	4	4					2,44%
Posatore Mattonelle	4	4					2,50%
Responsabile Commerciale	4	2			1	1	2,56%
Responsabile di cantiere	4	4					2,63%
Responsabile ristorazione	4	2			1	1	2,70%
Sarta	4	4					2,78%
Asfaltista	3	3					2,14%
Carrozziere	3	3					2,19%
Fornaio	3	3					2,24%
Giardiniere	3	3					2,29%
Gommista	3	3					2,34%
Impiegato tecnico - operatori spettacolo	3	3					2,40%
Impiegato Ufficio Acquisti	3	3					2,46%
Laurea Lingue	3	2	1				2,52%
Laurea Scientifica - Chimica	3	3					2,59%
Laurea Scientifica - Geologia	3	3					2,65%
Lavanderia Stiratrice	3	3					2,73%
Metalmeccanico - tornitore	3	3					2,80%
Operatore Agricolo	3	3					2,88%
Scultore	3	3					2,97%
Addetto assistenza\manutenzione	2	2					2,04%
Arredatore	2	2					2,08%
Assistente anziani	2	2					2,13%
Autista trasporto pompaggio	2	2					2,17%
Carpentiere metallico	2	2					2,22%
Cartongessista	2	2					2,27%
Conciatore pelli	2	2					2,33%
Domestica	2	2					2,38%
Educatore	2	2					2,44%
Giardiniere add giardinaggio	2	2					2,50%
Imbianchino - Decoratore	2	2					2,56%
Impiegato Logistica	2	2					2,63%
Impiegato tecnico - Perito Meccanico	2		2				2,70%
installatore impianti gpl/gas/metano	2	2					2,78%
Istruttore sportivo	2	2					2,86%

mansione elaborata	numero occorrenze	0 err	1 err	2-3 err	4-10 err	>5 err	incidenza della classe sul totale
Laurea Scientifica - Ingegneria Civile	2	2					2,94%
Litografo	2	2					3,03%
Medicina - Farmacia	2	2					3,13%
Modellista Stilista	2	2					3,23%
Operatore Telemarketing	2	1		1			3,33%
Organizzatore eventi	2	2					3,45%
Recupero Crediti	2	2					3,57%
ristorazione collettiva	2	2					3,70%
Tipografo	2	2					3,85%
Verniciatore	2	2					4,00%
Vivaista	2	2					4,17%
	1					1	2,17%
Add macchine cartiera	1	1					2,22%
add. rivestimento degli arti	1	1					2,27%
Addetto montaggio video	1	1					2,33%
Agente di Vendita	1	1					2,38%
Agente Pubblicitario	1	1					2,44%
Assistente dentista - Igienista	1	1					2,50%
Assistente Governante	1	1					2,56%
Assistenza Bagnanti	1	1					2,63%
Autista - Camionista	1	1					2,70%
Autista - Facchino add.Trasclochi	1	1					2,78%
Autista trasportatore	1	1					2,86%
Carpenteria Metallica	1	1					2,94%
Collab. Famigliare - Colf	1	1					3,03%
Dirigente Comunità	1	1					3,13%
Elettrauto	1	1					3,23%
Formatore	1	1					3,33%
Giardiniere - Vivaista	1	1					3,45%
Impiegato tecnico - Grafico	1	1					3,57%
Impiegato tecnico - Perito Termotecnico	1				1		3,70%
Infermiere	1		1				3,85%
Laurea biotecnologie	1	1					4,00%
Laurea in Scienze Naturali	1	1					4,17%
Laurea Matematica	1	1					4,35%
Laurea Pedagogia	1	1					4,55%
Laurea Scientifica - statistica	1			1			4,76%
Magazziniere	1	1					5,00%
Marmista - scultore	1	1					5,26%
Massaggiatori	1	1					5,56%
Meccanico - auto - Collaudatore	1	1					5,88%
Medicina	1	1					6,25%
Medicina - Erboristeria	1	1					6,67%
Medicina - Naturopata	1	1					7,14%

mansione elaborata	numero occorrenze	0 err	1 err	2-3 err	4-10 err	>5 err	incidenza della classe sul totale
Medicina - Olistico	1	1					7,69%
Medicina - Tecnologie Alimentari	1	1					8,33%
Modellista	1	1					9,09%
Oparatore Radiofonico	1	1					10,00%
Operatore televisivo	1	1					11,11%
OSS - Responsabile	1	1					12,50%
Plastichino	1	1					14,29%
Portiere	1	1					16,67%
posatore sottofondi	1	1					20,00%
Prototipazione	1	1					25,00%
Responsabile Gestione	1	1					33,33%
Scienze della Formazione	1	1					50,00%
Tuttofare uomo di fatica	1	1					100,00%

Tab. 3. Report risultati con errori

CONCLUSIONI

L’obiettivo di questa tesi era presentare un algoritmo capace di ricondurre a parole normalizzate le richieste di lavoro che pervengono ad un Ufficio di Collocamento.

Le richieste di lavoro vengono scritte da operatori che non sono tenuti a rispettare regole di compilazione per cui il dato grezzo risultato risulta essere difficile da classificare.

Attraverso lo studio dei fattori che provocano indecidibilità del dato e che incidono sugli operatori che effettuano l’inserimento dati si rimarca la necessità di approcciare lo studio dell’interfaccia utente verso canali tematici non tipici della formazione informatica.

Uno sviluppo massiccio dei sistemi di comunicazione come quello che stiamo vivendo, sviluppato per un nuovo modello di utente che non ha fra le proprie specifiche la conoscenza approfondita dello strumento con il quale si trova ad operare deve fare i conti con nuove aree tematiche non ancora appieno approfondite sul piano informatico.

Affrontare l’analisi del problema delle offerte di lavoro dell’Ufficio di collocamento ha permesso di osservare dinamiche e ipotizzare nuovi modelli di interfaccia utente basati su:

- Fattori ambientali
- Fattori psicologici
- Fattori cognitivi
- Capitale umano

Non possono rimanere estranei nella progettazione dell’interfaccia utente. Nel corso della trattazione della tesi questi argomenti sono stati trattati nel primo capitolo anche se non con il rigore che il buon metodo matematico richiederebbe.

La difficoltà nel reperire materiale da poter consultare è stato un deterrente fondamentale alla stesura approfondita e accurata dei contenuti del primo capitolo.

Nel secondo capitolo si presenta ciò che sotto gli occhi di tutti rappresenta il modello indiscutibilmente più testato al mondo per quanto riguarda la ricerca di informazioni. Si è ritenuto iniziare lo studio della soluzione al problema presentato studiando quello che allo stato dell'arte rappresenta una certezza.

Il metodo di ricerca utilizzato come modello, è l'algoritmo di Google, attraverso la presentazione della sua struttura dati, dei criteri di ricerca e ottimizzazione e fornitura dei risultati si è creato quel background culturale fondamentale per poter formulare una qualsiasi ipotesi di soluzione.

Analizzando le specifiche delle esigenze della catalogazione delle pagine Web e d'altro canto della memorizzazione e codifica delle offerte di lavoro si è arrivati alla formulazione di un algoritmo capace di ricondurre a valori classificati attraverso un procedimento automatico gran parte delle offerte di lavoro inserite.

Il metodo messo a punto basa la propria capacità nel disporre di una quantità di informazioni (dizionario) sufficientemente ampio da poter rispondere alle domande che vengono sottoposte al sistema.

L'algoritmo non è dotato di sistemi di auto apprendimento per evitare la proliferazione di risultati inaspettati.

Per quanto dispendioso possa apparire si ritiene che una fase di autorithing dei valori utilizzati come prototipo sia fondamentale.

Quindi non spaventa l'incapacità di arrivare ad una soluzione automatica sotto ogni richiesta fornita, ma bensì alla certezza che i risultati ottenuti siano corretti.

Nel paragrafo 1.3.2.5 si richiama l'attenzione all' illusione di Mosè, si riferisce al fatto che molte persone rispondono alla domanda: «Quanti animali di ciascuna specie Mosè portò con sé sull'arca?» dicendo «Due». Naturalmente, nessun animale venne portato sull'arca da Mosè. Era Noè.

Con questo algoritmo si studia il dizionario delle parole conosciute affinché per l'errore non possa esistere una risposta ad una domanda approssimata.

Perno centrale dell'algoritmo è l'uso della funzione SOUNDEX implementata specificatamente per questo algoritmo.

Inoltre per distinguere semanticamente le parole sono state categorizzate in classi per cui (se la redazione del dizionario è sviluppata con cura) la risposta è deterministica. Per questo è necessario porre particolare cura nella stesura di un dizionario completo ma soprattutto strutturato correttamente.

A completamento del percorso di ricerca una verifica finale effettuata con una funzione EDITDISTANCE basata su distanza di Hamming restituisce l'errore riscontrato.

A supporto di quanto esposto ma soprattutto della qualità dell'algoritmo è stata presentata un ampio paragrafo con i risultati ottenuti.

Ma nuovi studi saranno orientati verso un'altra tecnologia di ricerca chiamata Probabilistic latent semantic analysis (PLSA) chiamata anche Indicizzazione semantica latente probabilistica che saranno oggetto di confronto con i risultati ottenuti con l'algoritmo presentato. All'inizio scartata perché si temeva che potesse rendere difficile la gestione di casistiche particolar, ora si ritiente interessante proporre una comparazione fra i due metodi.

Tutto quanto presentato fin ora permette per giungere alla significativa conclusione che una attenta progettazione del sistema informativo deve tenere conto di aspetti che le classiche analisi di sistemi informativi non considerano. Ragionare a tutto tondo permette di ottenere risultati migliori e più coerenti, un valore aziendale maggiore con una partecipazione maggiore da parte del personale disponibile in azienda.

BIOGRAFIA

TOSCANO LORENZO(2009): Seo strategy, Uniservice, Trento

http://www.google.com/patents?printsec=abstract&id=AuaoAAAAEBAJ&output=text&pg=PA8

ALLAN M. COLLINS, M. ROSS QUILLIAN(1969): Retrieval time from semantic memory, http://matt.colorado.edu/teaching/categories/cq69.pdf

CUENCA, M. J. y J. HILFERTY (1999): Introducción a la lingüística cognitiva, Barcelona,Ariel.

TAYLOR, J. R. (1995, [2009]): Linguistic categorization, Oxford, Oxford University Press.

EVANS, V. y M. GREEN (2006): Cognitive linguistics. An introduction, Edinburgh, Edinburgh University Press.

VITTADINI GIORGIO: http://www.istat.it/it/files/2011/02/Vittadini.pdf

LAUDANNA ALESSANDRO: Memoria semantica concetti processi semantici, Università di Salerno.

REDER &.KUSBIT, (1991): Moses illusion, http://www.psy.cmu.edu/faculty/reder/ph_rlm.pdf

KEYWORD DENSITY & PROMINENCE [online]:

- *http://labs.translated.net/relazioni-semantiche/*
- *http://en.wikipedia.org/wiki/Probabilistic_latent_semantic_analysis*
- *http://en.wikipedia.org/wiki/EM_algorithm*

Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. *Series B (Methodological)*

Thomas Hofmann, Learning the Similarity of Documents : an information-geometric approach to document retrieval and categorization, *Advances in Neural Information Processing Systems 12, pp-914-920, MIT Press, 2000*

Sommario