# UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA
## FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI
## Corso di Laurea in Informatica



# Costruzione di un thesaurus per gli algoritmi di prossimità semantica

# Riassunto


**DISCo – LET**
Dipartimento Informatica, Sistemistica e COmunicazione –
Laboratorio Elaborazione Testi

**Supervisori**:
Prof. Huu LE VAN
Prof. Luca BERNARDINELLO

Relazione della prova finale di:
**Roberto FERMI**
Matricola n°048091



Anno Accademico 2004 – 2005

# Introduzione

Il contenuto della mia tesi riguarda l'Information Retrieval, ovvero quella disciplina che si occupa di studiare, progettare e realizzare sistemi informatici finalizzati al reperimento di documenti riguardanti le richieste degli utenti.
Questa tematica ha suscitato il mio interesse perché ritengo che nella società moderna l'IR rappresenta e rappresenterà sempre di più il modo più veloce ed efficace per memorizzare e ricercare informazioni su qualsivoglia argomento.
Lo scopo di questo stage è quello di realizzare un sistema per il reperimento di informazioni che stabilisca una valida metodologia per il confronto tra il bisogno informativo dell'utente (query) e la rappresentazione logica dei documenti (surrogati di documenti).
I parametri di ingresso di un IRS sono le informazioni rilevanti tratte dal testo originale e la richiesta dell'utente. Il processo di retrieval può essere sintetizzato nei seguenti passi:

- creazione di una rappresentazione di ogni oggetto, basata sulla sua descrizione;
- creazione di una rappresentazione del fabbisogno informativo, query, dell'utente;
- confronto tra le due rappresentazioni e scelta di quelle che meglio rappresentano la query dell'utente.

L'obbiettivo ottimale è quello di fornire all'utente solamente risultati totalmente pertinenti alla sua richiesta, escludendo tutto il resto. In pratica questo traguardo non è raggiungibile e, quindi, si utilizzano due parametri per valutare l'efficacia di un processo di retrieval: recall (richiamo) e precision (precisione). La precisione misura l'abilità del sistema nel recuperare solamente documenti rilevanti per la richiesta dell'utente. Il richiamo misura l'abilità del sistema nel recuperare tutti i documenti rilevanti.
Vi sono diversi metodi per effettuare ricerche su testi, ma l'idea che sta alla base di tutti è quella di ridurre il documento originale ad un surrogato di documento (un riassunto significativo) e, quindi, confrontare le frasi in esso contenute con la query.
Vi sono molte metodologie relative a questa attività di confronto; quelle più diffuse sono:

- Modello booleano
- Modello Fuzzy
- Modello probabilistico
- Modello a spazio vettoriale

Prima che un documento possa essere utilizzato come base per il reperimento delle informazioni, si devono eseguire alcune operazioni in modo da recuperare solamente i termini rilevanti. Le principali fasi di questo processo sono:

- Stop-list
- Stemming
- Weighting
- Thesaurus

Mediante l'unione delle operazioni precedenti è possibile realizzare un thesaurus che contiene sostanzialmente coppie di termini che hanno un certo grado di similitudine fra di loro.
Per quanto riguarda il confronto fra le frasi, le principali caratteristiche di un buon algoritmo di prossimità semantica sono la capacità di interpretare correttamente il fabbisogno informativo dell'utente (precision) e la velocità con cui viene restituito il risultato.
Lo stage è stato sviluppato in due parti: dapprima è stato realizzato un algoritmo per la costruzione di un thesaurus e, quindi, è stato possibile implementare un algoritmo per la ricerca della prossimità semantica tra le frasi. Nella prima parte viene trattato l'algoritmo per la costruzione del thesaurus, mentre nella seconda parte l'implementazione dell'algoritmo di Paice-Ramirez.

## Parte 1: Thesaurus

L'utilizzo dei thesaurus negli algoritmi di prossimità semantica è determinante, ma d'altra parte, tale implementazione richiede un tempo di calcolo molto elevato e, spesso, risulta inapplicabile per i sistemi informatici comuni. Sorge, dunque, il problema di ottimizzare il tempo di esecuzione dell'algoritmo, mantenendo contemporaneamente un livello ottimale di ricerca della somiglianza dei termini.
In un thesaurus vengono memorizzati i valori di somiglianza tra coppie di parole. Esistono diverse tecniche per la costruzione automatizzata dei thesaurus; quella utilizzata per questo stage è basata sulla ricerca nel documento campione di termini che possono essere considerati vocaboli simili per il loro. Fra i diversi algoritmi orientati al calcolo della somiglianza semantica tra le parole si è scelto quello relativo al coefficiente del Coseno, che risulta più efficace.
L'implementazione di un buon thesaurus è realizzata avvalendosi delle tecniche precedentemente descritte ed, in particolare, le fasi fondamentali sono:

- normalizzazione del testo in ingresso;
- applicazione del coefficiente del Coseno per l'analisi della somiglianza semantica dei termini;
- perfezionamento dei dati in uscita.

***Implementazione***

La normalizzazione del documento originale viene svolta in fasi successive. Per prima cosa vengono eliminati tutti i caratteri di punteggiatura, tranne il punto, e gli eventuali doppi spazi presenti. Quindi, mediante l'utilizzo di un apposito dizionario, sono eliminate tutte le parole non significative per il contesto del documento. Infine, la normalizzazione delle parole del documento (ovvero la ricerca della radice del termine) viene eseguita sfruttando l'algoritmo di stemming proposto da Porter. Questa procedura si basa sull'eliminazione sequenziale dei suffissi della parola, fino ad ottenere la forma base del termine. Il vocabolo così elaborato viene confrontato con un dizionario della lingua usata per verificare di avere ottenuto una parola esistente e non avere commesso errori di stemming; in caso il termine non venga trovato, viene ripristinato il valore originale. L'applicazione dell'algoritmo di Porter, però, è limitata alla lingua inglese per la presenza di un minor numero di eccezioni grammaticali. Applicando queste elaborazioni è possibile ridurre notevolmente il numero di termini del testo diminuendo, così, il tempo di calcolo necessario alla ricerca del valore di somiglianza semantica tra le coppie di termini.
Per la realizzazione del thesaurus è stato utilizzato il coefficiente del Coseno, in quanto è in grado di fornire migliori risultati di somiglianza semantica rispetto alle altre formule.
Una volta normalizzata ogni parola del documento, viene eseguito il calcolo del coefficiente del Coseno su ogni coppia di termini unici mediante la seguente formula:

$$somiglianza\_lessicale = \frac{\sum_{i=1}^{N_f} V_{px}[i] \cdot V_{py}[i]}{\sqrt{\sum_{i=1}^{N_f} V_{px}[i]^2 \cdot \sum_{i=1}^{N_f} V_{py}[i]^2}}$$

dove,

| | |
|---|---|
| $N_f$ | Rappresenta il numero di frasi del campione |
| $N_s$ | Rappresenta il numero di frasi significative presenti nel campione |

| | |
|---|---|
| $P_1, P_2, ..., P_n$ | Rappresenta le parole significative |
| $V_{pi}[N_f]$ | Vettore che contiene nella posizione $V_{pi}[j]$ il numero di occorrenze della parola $p_i$ nella frase $j-esima$ |

I vocaboli normalizzati sono organizzati sulle righe di una tabella; sulle colonne della stessa, invece, sono memorizzate le occorrenze dei termini in ogni frase. Il coefficiente del Coseno utilizza questa struttura dati per calcolare il valore si somiglianza semantica tra tutte le coppie di termini unici. In particolare, per risparmiare spazio nel database e tempo di calcolo, si è scelto di computare solamente le dipendenze dirette e non quelle inverse; in altre parole, se si trova il valore di somiglianza lessicale per la coppia di termini "A B", non verrà calcolato anche per "B A". Anche l'analisi del caso "A A" verrà saltata, in quanto il valore di somiglianza semantica sarà 1.
Il perfezionamento dei dati in uscita consiste nell'individuare una soglia del valore di somiglianza semantica al di sotto della quale la coppia di parole non deve essere memorizzata nel thesaurus e in questo modo è possibile definire la precisione del thesaurus che verrà costruito. Si è scelto il valore soglia uguale a 0.3, infatti, dopo attente analisi, è stato rilevato che al di sotto di questa soglia le parole sono difficilmente correlate tra di loro. Per ottenere dei buoni risultati di ricerca, quindi, si è scelto di ignorare coppie al di sotto della soglia minima. E' stato inoltre posto un limite inferiore alla lunghezza delle parole considerate: termini con un numero di caratteri inferiore a 2 vengono ignorati.
Nel caso la stessa coppia di termini venga analizzata più volte in diversi documenti, il valore di somiglianza semantica memorizzato nel thesaurus sarà la media tra tutti i valori di affinità rilevati fino a quel momento. Questo accorgimento è necessario per ottenere un valore realistico della somiglianza tra due termini. Se si ha una grande quantità di documenti da analizzare, infatti, sicuramente la stessa coppia verrà analizzata diverse volte ed avrà, sicuramente, valori differenti da documento a documento. Calcolandone la media si può tenere traccia della rilevanza in ogni documento, aumentando la precisione del thesaurus.
Per potere calcolare correttamente il valore medio è necessario che nel database venga memorizzato anche il numero di volte che la coppia di parole è stata analizzata. La struttura del database, quindi, sarà la seguente:

| Termine1 | Termine2 | Somiglianza | Occorrenze |
|---|---|---|---|

Il codice per la costruzione del thesaurus è stato sviluppato in ambiente multi-thread in modo da rendere possibile l'esecuzione parallela di più istanze di ogni classe. Questo permette di sfruttare pienamente le potenzialità dei nuovi calcolatori e ridurre considerevolmente i tempi di computazione.
Lo scopo di questo stage è quello di realizzare un thesaurus utilizzabile dagli algoritmi di prossimità semantica per ricerche in differenti ambiti. Il numero di termini presente nel database, quindi, deve essere sufficientemente consistente da permettere di ottenere buoni risultati per qualsiasi ricerca.
Poiché il thesaurus viene costruito solamente in base alla frequenza dei termini, è necessario che ogni documento preso in analisi abbia un contesto ben definito. In questo modo si eviteranno ambiguità tra i valori dei dati raccolti.
Per questa tesi sono stati analizzati 42 documenti con una media di circa 10.000 parole totali, ottenendo un thesaurus di 11.405 relazioni significative e 408 relazioni importanti.

## Parte 2: L'algoritmo di Paice-Ramirez

Dopo attente ricerche è stato rilevato che uno dei più semplici, ma affidabili algoritmi di prossimità semantica, basato sull'utilizzo di thesaurus, è quello sviluppato da Paice e Ramirez.

Il funzionamento si basa sulla quantificazione della vicinanza tra due frasi in relazione al grado di somiglianza esistente tra le parole della prima e della seconda frase. Per questo motivo è essenziale avere a disposizione un thesaurus contenente le coppie di parole che presentano tra loro un'affinità lessicale.

Forniamo alcune definizioni che si utilizzeranno per definire opportunamente il funzionamento dell'algoritmo:

- parola: qualsiasi sequenza finita non nulla di caratteri alfabetici avente lunghezza uguale o maggiore rispetto ad un valore minimo predefinito;
- termine: qualsiasi parola che non compare all'interno di una lista predefinita di parole non significative;
- frase: sequenza finita e non nulla di termini che termina con un punto.

Ora si possono considerare

$S=(S_1, S_2, ..., S_m)$

e

$T=(T_1, T_2, ..., T_n)$

due frasi composte, rispettivamente, da m ed n termini. La funzione che calcola la prossimità semantica $R$ tra le due frasi $S$ e $T$ ha la seguente forma:

$$R(S,T) = f(S_x, T_y)$$

***Implementazione***

Al fine di valutare correttamente il peso delle parole, è necessario definire un mapping, ovvero un'associazione tra i termini della prima frase ed i termini della seconda frase, in modo tale che i due termini associati presentino una somiglianza non nulla. Si può assumere che: ogni termine $S_x$ di $S$ può essere associato ad un termine distinto $T_y$ di $T$, oppure non essere mappato. Un termine di $S$ non può essere associato a più di un termine di $T$ e viceversa.

Possiamo definire $J(x)$ come una funzione che associa il termine $T_{J(x)}$ di $T$ con il termine $S_x$ di $S$. Per ogni $S_x$ di $S$ che non presenta mapping con termini di T, vale che $J(x) = 0$.

Un fattore di cui è importante tenere conto è l'ordine dei termini nelle due frasi esaminate. Occorre introdurre una funzione che rappresenti la tidyness (accuratezza) dell'ordinamento dei termini.

$$\Psi(J) = \psi_{\min} + \Phi(J) \cdot (1 - \psi_{\min})$$

dove,

$$\Phi(J) = \frac{1}{m-1} \sum_{x=1}^{m-1} K_x$$

e

$$K_x = \begin{cases} 1 & se \quad J(x+1) = J(x) + 1 \\ 0 & se \quad J(x+1) \neq J(x) + 1 \end{cases}$$

Anche nel caso dell'ordinamento peggiore, però, questa funzione non dovrebbe portare a "0" il risultato.

La funzione per il calcolo della prossimità semantica, considerando anche il vincolo di tidyness, diventa:

$$R(S,T) = \frac{1}{F}\sum_{x=1}^{m} W_x \cdot V \cdot \left(S_x \cdot T_{J(x)}\right) \cdot \Psi(J)$$

dove:

- per ogni coppia di termini associati $S_x$ e $T_{J(x)}$ si intende determinare un valore che ne definisca l'affinità. $W_x$ è il peso associato alla parola $S_x$
- $F$ è un fattore di normalizzazione
- $V \cdot \left(S_x \cdot T_{J(x)}\right)$ denota la prossimità semantica o lessicale tra i due termini $S_x$ e $T_{J(x)}$. $V$ viene calcolato sulla base dei valori contenuti nel thesaurus.
- $J(x)$ denota la funzione di mapping
- $\Psi(J)$ denota la funzione di tidyness

Il mapping è calcolato utilizzando una matrice $m \times n$ dove sulle righe sono indicati i termini della frase del documento e sulle colonne le parole della query. Ad ogni intersezione viene inserito il valore di somiglianza semantica tra i due vocaboli considerati, prelevandolo dal thesaurus.
Il mappaggio diretto dei termini è una tecnica molto semplice e veloce per determinare un'associazione tra le parole, ma non è efficiente in quanto "miope". Per ovviare a questo problema il mapping è stato realizzato utilizzando un algoritmo che tenga conto della distribuzione dei valori dei pesi nella matrice, creando una seconda matrice W. I valori contenuti in W sono pesi che penalizzano una eventuale scelta del termine corrente.

$$w_{ij} = \sum_{k=1}^{n,k\neq i} m_{ik} + \sum_{k=1}^{m,k\neq j} m_{ik} + \left(1 - m_{ij}\right)$$

Il valore scelto è quello minimo, ovvero quello corrispondente alla minima perdita. Quando il valore viene scelto, l'algoritmo cancella la riga e la colonna corrispondente all'elemento selezionato. In questo modo, si evita la sua scelta nelle fasi successive.
La scelta dell'ordinamento per la corrispondenza tra i termini è memorizzata in un vettore che verrà utilizzato per calcolare la funzione di tidyness, ovvero per analizzare se i termini della query sono posti nello stesso ordine delle parole della frase considerata.
Particolare attenzione va posta al valore finale di prossimità semantica restituito dall'algoritmo. Questo, infatti, è ottenuto effettuando una media pesata tra i coefficienti di prossimità semantica di ogni singola frase del documento. Il peso è calcolato da un'apposita formula e dipende dal numero di frasi presenti nel testo e dal valore della singola frase ed è così definito:

$$peso = e^{(prossimità * coefficiente)}$$

$$Coefficiente = \ln(numero\_parole) \cdot \frac{100}{80}$$

La complessità delle formule impiegate in questo algoritmo, rende difficile una valutazione complessiva della qualità del codice. Per ovviare a questo problema si è scelto di eseguire test separati che valutino le singole componenti dell'algoritmo.

***Valutazione dei sinonimi***

Il test condotto consiste nell'eseguire differenti ricerche sullo stesso testo inserendo come query dei sinonimi. Si è verificato che l'algoritmo utilizza correttamente lo strumento thesaurus in quanto restituisce dei risultati paragonabili con l'effettiva somiglianza semantica tra le parole analizzate.

***Valutazione dell'ordinamento***

Per valutare l'ordinamento delle parole della query rispetto alla frase analizzata del documento, è stata introdotta nel testo un'apposita frase contenente tre vocaboli chiave; i test sono stati condotti modificando l'ordine dei termini della query ed analizzando le variazioni della prossimità semantica rilevata. Dai test si può rilevare che l'algoritmo è in grado di analizzare l'ordinamento dei termini del fabbisogno informativo dell'utente rispetto alla frase del documento, per ottimizzare il valore di prossimità semantica.

***Valutazione della lunghezza delle frasi***

Un altro aspetto da analizzare è l'influenza che la lunghezza della frase ha sul valore di prossimità semantica indicato dall'algoritmo. Per eseguire questa valutazione si può modificare la lunghezza delle frasi di un testo ed analizzare la variazione dei risultati. Dagli esiti dei test si può evincere che il valore di prossimità semantica viene influenzato correttamente dalla lunghezza della frase considerata.

***Valutazione di precisione e richiamo***

Al fine di verificare l'efficienza dell'algoritmo sviluppato è necessario analizzare i parametri di precisione e richiamo, che caratterizzano i sistemi per il reperimento di informazioni. Dai risultati ottenuti, si deduce che l'algoritmo ha una buona abilità nel recuperare solamente i documenti rilevanti per l'utente, mentre la sua capacità di recuperare tutti i documenti effettivamente importanti per la ricerca dell'utente è inferiore.

## Conclusioni

Le principali difficoltà incontrate nella realizzazione di questo stage sono state quelle di ridurre al minimo i tempi di computazione per poter fornire i risultati in tempi accettabili dall'utente e quella di dare una corretta interpretazione alla formula generale di Paice-Ramirez. Dagli esiti raccolti si può affermare che l'obbiettivo generale di questo stage è stato raggiunto in modo più che soddisfacente: i tempi di calcolo sono contenuti, gli algoritmi si adattano ai più moderni calcolatori sfruttandone a pieno le potenzialità e i dati di output sono coerenti.

L'esperienza di questo stage mi ha permesso di approfondire gli aspetti cruciali dell'IR e di studiare nuovi elementi della programmazione, dell'ottimizzazione del codice e della configurazione hardware dei calcolatori.