Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un sistema automatico per l'individuazione di curricola presenti in Internet

Rosario Calabria, Giovanni Pilato, Francesco Ricotta,
Filippo Sorbello, Giorgio Vassallo

RT-ICAR-PA-03-18 **dicembre 2003**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: www.icar.cnr.it
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.na.icar.cnr.it
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: www.pa.icar.cnr.it

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Un sistema automatico per l'individuazione di curricola presenti in Internet

Rosario Calabria[2], Giovanni Pilato[1], Francesco Ricotta[2]
Filippo Sorbello[2], Giorgio Vassallo[2]

Rapporto Tecnico N.18: ***Data:***
RT-ICAR-PA-03-18 **dicembre 2003**

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sezione di Palermo Viale delle Scienze edificio 11 90128 Palermo
[2] Università degli Studi di Palermo Dipartimento di Ingegneria Informatica Viale delle Scienze 90128 Palermo

Indice

Introduzione

Internet può essere vista come un enorme basi di dati amorfa dove le informazioni si trovano senza un ordine prefissato. Per reperire le informazioni, tramite gli attuali motori di ricerca, occorre comporre delle **query** ossia delle domande, limitate a singole parole. Queste sono parole che si trovano all'interno del documento. Lo scopo del presente lavoro è di fornire una soluzione che automatizzi l'estrazione delle informazioni e che le ponga in una forma strutturata in base al loro contenuto semantico. In questo modo ci viene consentito di porre le informazioni in una base di dati per poterle consultare ponendo delle query anche complesse. XML viene incontro a questa necessità in quanto i suoi tag diversamente a quelli dell'HTML, sono strutturati in base alla semantica cioè all'informazione stessa e non in base all'impaginazione come nell'HTML. Si tratta quindi di un applicazione che presa una pagina HTML la converta in XML o analogamente in tabelle di una base di dati. Si può così realizzare un applicazione che analizzi le pagine di un sito web, appartenente ad una data categoria e ne estragga le informazioni significative per poi memorizzarle in una base di dati oppure visualizzarle all'utente direttamente all'utente impaginandole tramite un foglio di stile in XSL. Quindi di si hanno due ooperazioni da compiere: classificare un documento HTML ed estrarre le informazioni significative.
Per quanto riguarda la classificazione, viene proposto un metodo indipendente dalla lingua del documento. Viene utilizzata la cross-entropia [8] cioè una misura su base statistica della dipendenza tra variabili aleatorie. Questo al fine di creare un dizionario di parole sulle presenza o assenza in un documento formare un vettore di codifica. Per determinare la classe di appartenenza del documento viene addestrata in tal senzo una rete neurale particolarmente utile in quando in grado di eliminare le componenti di rumore.
Al fine, di estrarre le informazioni significative, il metodo proposto ha il vantaggio di poter essere utilizzato su qualunque documento di testo, non occorre infatti che questo sia in formato HTML. Viene fatto uso del contesto che nel nostro caso è costituito dalla parola che precede e da quella seguente

la sezione di testo che contiene l'informazione da estrarre. Con tali parole vengono formati degli elenchi ordinati in modo da privilegiare le parole la cui presenza nel testo indica, con buona probabilità, il fatto che accanto vi sia il campo da estrarre.
Il lavoro appena descritto è organizzato nei seguenti capitoli

- **Richiami di teoria**. Qui vengono trattati: la cross-entropia (o coefficiente d'incertezza fra due variabili aleatorie)e le reti neurali (a singolo strato). Argomenti propedeutici alla trattazione della soluzione proposta.
- **La soluzione proposta**. Questo capitolo è diviso in 2 sezioni. La prima per la classificazione di una pagina HTML tramite uso della cross-entropia. La seconda per l'estrazione dei campi dalle pagine HTML tramite uso del contesto.
- **Risultati sperimentali**. Viene fornita una dettagliata descrizione del lavoro svolto con grafici e tabelle riportanti i risultati su di un insieme di pagine HTML.
- **Conclusioni**.

Capitolo 1

Richiami di teoria

In questo capitolo vengono descritti degli strumenti di tipo teorico che verranno usati nella soluzione proposta. Il capitolo è diviso in due sezioni: nella prima viene introdotta la cross-entropia utile per la classificazione dei documenti in quanto fornisce una misura della dipendenza della presenza o meno di una parola dalla classe a cui appartiene un documento.

1.1 La cross-entropia

Supponiamo di avere due variabili casuali ciascuna delle quali può assumere valore vero oppure falso. Noi vogliamo misurarne il grado di associazione, cioè vogliamo sapere se la variabile I è dipendente dalla variabile j (con i diverso da j) e se lo è vogliamo misurare il grado di tale dipendenza. Vediamo come possiamo pervenire ad un informazione di tale tipo[8] .

Riferiamoci alla tabella, associata a due variabili X e Y, mostrata in figura:

	Classe=1	**Classe**=2		
Parola=1	N_{11}	N_{12}	...	$N_{1.} = N_{11} + N_{12}$
Parola=2	N_{21}	N_{22}	...	$N_{2.} = N_{21} + N_{22}$
...	...	...	...	...
	$N_{.1} = N_{11} + N_{21}$	$N_{.2} = N_{12} + N_{22}$	...	$N = \sum_{i,j=1}^{2} N_{ij}$

Tabella 1.1: Tabella di contingenza calcolata per una parola su tutti i documenti dell'insieme d'addestramento

Il metodo della cross-entropia si basa proprio sull'utilizzo di tale tabella chiamata tabella di contingenza. L'analisi del grado di associazione tra

queste variabili prende il nome di Analisi della Tavola di Contingenza o Cross-tabulation Analysis. Riferendoci alla tabella definiamo le probabilità.

$$p_{i,j} = \frac{N_{ij}}{N}$$

$$p_{i.} = \frac{N_{i.}}{N}$$

$$p_{.j} = \frac{N_{.j}}{N}$$

dove: pij è il rapporto tra il numero di casi in cui la variabile X assume il valore i-esimo mentre la variabile Y assume il valore j-esimo ed il numero totale di esperimenti. pi. è il rapporto tra il numero di casi in cui la variabile X assume il valore i-esimo ed il numero totale di esperimenti. p.j è il rapporto tra il numero di casi in cui la variabile Y assume il valore j-esimo ed il numero totale di esperimenti. Dopo aver eseguito N volte un esperimento su entrambe le variabili, Possiamo associare, a ciascuna delle I variabili un valore chiamato entropia per le variabili X e Y definite nel seguente modo: Entropia di **Y**:

$$H(X) = -\sum_{i=1}^{N} p_{i.} \ln p_{i.} \tag{1.1}$$

Entropia di **X**:

$$H(Y) = -\sum_{j=1}^{N} p_{.j} \ln p_{.j} \tag{1.2}$$

Nel calcolo delle (1.1) e (1.1) teniamo conto del fatto che:

$$lim_{p \to 0} ln(p_i) = 0$$

H varia tra 0 e ln(I). E' 0 solo se una tra le pi è uno mentre le altre sono uguali a zero. In questo caso la variabile assume un valore prefissato. Il massimo valore ln(I) viene assunto da H quando i pi sono tutti uguali.
L' entropia tra due variabili X ed Y insieme vale:

$$H(X,Y) = -\sum_{j=1}^{2} p_{ij} \ln p_{ij} \tag{1.3}$$

Infine l'entropia di Y noto X e viceversa di X noto Y:

$$H(Y|X) = -\sum_{i,j} p_{ij} ln \frac{p_{ij}}{i.} \tag{1.4}$$

$$H(X|Y) = -\sum_{i,j} p_{ij} ln \frac{p_{ij}}{.j} \tag{1.5}$$

Dove pij indica il rapporto tra il numero di volte in cui rispettivamente la variabile X assume il valore i-esimo e la variabile Y assume invece il valore j-esimo. A questo punto abbiamo tutte le carte in regola per introdurre il coefficiente di incertezza, cioè una misura del grado di dipendenza di Y su X; denoteremo tale dipendenza con $U(Y|X)$, tale valore è dato da:

$$U(Y|X) = \frac{H(Y) - H(Y|X)}{H(Y)} \tag{1.6}$$

tale valore varia tra 0 ed 1; se vale 0 allora X ed Y non sono associate, se vale 1 allora la conoscenza di X può aiutarmi a predire il valore di Y. Per valori compresi tra 0 ed 1, estremi esclusi, il coefficiente $U(Y|X)$ mi da la percentuale di entropia di Y persa se X è già nota. Supponiamo di considerare X come la variabile dipendente ed Y come quella indipendente, allora scambiando X con Y possiamo calcolare il coefficiente $U(X|Y)$ nel seguente modo:

$$U(X|Y) = \frac{H(X) - H(X|Y)}{H(X)} \tag{1.7}$$

Trattando X ed Y in maniera simmetrica possiamo considerare il coefficiente simmetrico di incertezza:

$$U(X,Y) = 2\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)} \tag{1.8}$$

perciò a tal proposito se le due variabili sono indipendenti allora avremo $H(X,Y) = H(X)+H(Y)$ per cui sostituendo nella 1.1 $U(X,Y) = 0$, se invece le due variabili sono completamente dipendenti allora $H(X) = H(Y) = H(X,Y)$ per cui sostituendo nella 1.1 avremo $U(X|Y) = 1$.

1.2 Le reti neurali

La rete neurale è uno strumento per l'apprendimento il cui funzionamento è simile a quello del cervello umano. Sostanzialmente il cervello apprende tramite esperienze, cioè associa delle azioni a percezioni di eventi. Il vero modo di funzionare del cervello è tuttora sconosciuto ma alcuni suoi aspetti sono noti. Infatti si sa che è composto da degli elementi chiamati neuroni. Vediamo come sono fatti e come funzionano i neuroni del cervello. Questi hanno 4 parti fondamentali: dendriti, nucleo, asse e sinapsi. Vediamo come funziona un neurone . Il nucleo riceve i segnali tramite le dendriti che costituiscono il collegamenti con i segnali provenienti dagli altri neuroni, li elabora

e produce un segnale che viene trasmesso ad altri neuroni tramite l'asse e le sinapsi. Noi ci riferiamo a dei neuroni artificiali, come quello in figura 1 il cui principio di funzionamento è simile a quello descritto. Questo ha un certo numero di ingressi a ciascuno dei quali è associato un peso, nel primo stadio viene eseguita la somma dei valori degli ingressi ciascuno dei quali moltiplicato per il peso associato. Nel secondo stadio, viene applicata a questa somma pesata, una funzione generalmente non lineare che generalmente prende il nome di funzione di trasferimento il cui valore è posto all'uscita del neurone. Le funzioni di trasferimento supportate solitamente sono: sigmoide, seno, tangente iperbolica, soglia. Quindi ogni neurone è caratterizzato dai pesi e dal tipo di funzione di trasferimento adottata. I neuroni sono connessi tra di loro formando quella che chiamiamo rete neurale. Noi sfruttiamo la rete neurale allo scopo di risolvere classi di problemi che non sono stati risolti con metodi tradizionali.
Una classe di problemi potrebbe essere, ad esempio, quella inerente alla codifica binaria di simboli. In questo caso occorrebbe come funzione di trasferimento la sogliatura. Questa restituirebbe il valore massimo quando la somma pesata degli ingressi supera il valore della soglia viceversa l'uscita avrebbe il valore minimo. In questo modo la rete implementa una rete logica. Se invece vogliamo che l'uscita possa assumere infiniti valori, come nel caso si stia realizzando un meccanismo per muovere una manopola, allora occorre come funzione di trasferimento ad esempio la sigmoide.

1.2.1 Le Unità Logiche a Soglia (ULS)

Un singolo neurone avente come funzione di trasferimento il gradino viene chiamata Unità Logica a Soglia (ULS). Questo tipo di neuroni sono particolarmente utili perché da soli possono implementare una classe di funzioni chiamate funzioni linearmente separabili. Queste sono funzioni logiche chiamate così perché lo spazio dei vettori d'ingresso che producono una uscita può essere separato dallo spazio dei vettori d'ingresso che producono un uscita bassa, tramite una superficie lineare. Ad esempio le funzioni AND, OR e la funzione maggioranza che assume valore 1 quando più della metà dei suoi ingressi è 1. Non è invece linearmente separabile la XOR. Vediamo da un altro punto di vista quanto detto. Indichiamo I il vettore degli ingressi della ULS e W quello dei pesi. Nella funzione a gradino possiamo impostare come parametro il valore della soglia cioè quel valore che una volta superato dalla somma pesata fa produrre un uscita alta. Per comodità è possibile inserire tale valore come ultima componente del vettore dei pesi W. A questa componente corrisponde l'ultima componente del vettore d'ingresso. In definitiva se N è il numero di ingressi e di pesi, allora sia I che W hanno dimensione N+1.

l'N+1 esima componente del vettore degli ingressi I è posta ad 1 mentre la corrispondente N+1 esima componente del vettore dei pesi è uguale al valore della soglia invertita di segno. Quindi, al momento dell'addestramento, essendo la soglia diventata un componente del vettore W potrà essere modificata come lo sono i pesi. Una ULS produce in uscita 1 quando $W \times I > 0$ ossia quando il prodotto scalare tra i vettori, a cui abbiamo appena aumentatati di 1 la dimensione, W e I è positivo. Da qui abbiamo che possiamo dividere lo spazio in due sottospazi: $W \times I > 0$e $W \times I < 0$. Il confine tra i due sottospazi è descritto dal piano: $W \times I = 0$ nel caso lo spazio sia a due dimensioni allora il piano diventa una retta. Nella figura 2.3 è mostrato un esempio di funzione logica AND con due ingressi dove è mostrata la retta di separazione dei due spazi.

1.2.2 Topologie delle reti

Se la funzione che si vuole approssimare non è linearmente separabile, allora una sola ULS non è sufficiente. In tal caso ricorriamo alle reti di neuroni. Il tipo di raggruppamento dei neuroni nella rete ne determinano le caratteristiche. Spesso i neuroni sono raggruppati in più strati per semplificate la costruzione della rete. Il primo strato contiene i neuroni direttamente interfacciati con il mondo esterno, l'ultimo strato è.formato da neuroni che forniscono le uscite. I neuroni dello strato intermedio non hanno alcun collegamento con il l'esterno della rete. Noi distinguamo 2 tipi di rete neurale in base al tipo di collegamento fra i neuroni. Il primo è chiamato con connessioni in avanti (Feed Forward) mentre il secondo è chiamato ricorrente. Nelle reti con connessioni in avanti, in ogni strato, ogni neurone è ha come ingressi le uscite dei neuroni dello strato precedente ed invece le uscite sono gli ingressi dei neuroni appartenenti allo strato successivo. Queste reti sono caratterizzate da tre parametri: il tipo di funzione di trasferimento di ciascun neurone, il numero di neuroni di ciascuno strato ed il numero di strati che compongono la rete. Per convenzione, per numero di strati ci si riferisce agli strati di pesi. Nella figura 2.2 è mostrata una rete con connessione in avanti avente 2 strati di pesi.
Qua ci va una figura sui percettroni
La scelta della topologia della rete è un parametro molto importante perché se dovessimo utilizzare troppo pochi neuroni, allora la rete non potrebbe rappresentare la funzione. Viceversa se ne usassimo troppi, allora si adatterebbe troppo ai vettori d'addestramento con la mancanza di generalizzazioni con il risultato che ricevendo in ingresso un vettore gia utilizzato per l'addestramento sarebbe prodotto l'uscita corretta se invece il vettore d'ingresso non era gia stato usato per l'addestramento allora non avremmo buone proba-

bilità che l'uscita sia corretta. Quest'ultima situazione viene chiamata sovra adattamento. Ogni rete avente uno strato nascosto è in grado di approssimare qualsiasi funzione continua. Invece con una a rete avente due strati nascosti si può rappresentare una qualsiasi funzione. Il numero di neuroni per ciascuno strato può anche crescere esponenzialmente con il numero di ingressi. Esistono degli algoritmi per trovare la topologia piu adatta per le reti in base al tipo di funzione da rappresentare. Uno di questi e di ricerca ma con uno spazio degli stati, che rappresenta le possibili strutture della rete, troppo ampio con notevole dispendio di memoria. Un altro tipo di algoritmo consiste nell'analizzare strutture di rete discendenti dall'attuale in questione risparmiando così memoria dovendo considerare solo un sottoinsieme di tutte le possibili strutture della rete. Si può partire da una rete grande per poi rimpicciolirla o, viceversa, partire da una rete piccola per poi ingrandirla. Un altro algoritmo per ottimizzare la struttura della rete è quello del danno cerebrale ottimo che consiste nell'eliminare i pesi da una rete totalmente connessa. I pesi che possono essere eliminati vengono determinati mediante un algoritmo basato sulla teoria dell'informazione. Questa operazione viene ripetuta ogni volta che si hanno prestazioni uguali o migliori a quelle del passo precedente. In questo modo si possono eliminare tre quarti dei pesi è viene migliorata la prestazione nella prova. Inoltre possono anche essere eliminati neuroni in eccesso. Per eliminare i pesi in eccesso può essere utilizzato l'algoritmo della pavimentazione. Questo algoritmo consiste nel cominciare con un solo neurone a che viene addestrato con il massimo numero possibile di esempi. Altri neuroni vengono aggiunti per poter dare la risposta corretta sugli esempi su cui non viene fornita la risposta corretta. In questo modo si hanno un numero di neuroni sufficienti a trattare tutto l'insieme d'addestramento.

1.2.3 Le reti RBF

Le funzioni continue dipendenti da un punto del piano e dalla distanza da tale punto vengono chiamate funzioni a simmetria radiale. Le reti la cui funzione d'attivazione è a simmetria radiale rispetto ad un punto del piano, vengono chiamate RBF ossia Radial based Function. In questo tipo di reti, l'addestramento avviene la regressione lineare e consiste nella modifica oltre ai pesi, del centro della funzione di attivazione. Le funzioni d'attivazione a simmetria radiale più utilizzate sono: la gaussiana, la multiquadrica inversa ed i polinomi di Hermite. Le reti RBF sono in grado di approssimare qualsiasi funzione non lineare utlizzando un solo strato nascosto inoltre l'addestramento è più semplice e più rapido rispetto alle reti MLP.

1.2.4 Addestramento delle reti

La rete neurale per poter funzionare deve prima essere addestrata cioè attribuire i valori ad i pesi in modo tale da avere valori per le uscite appropriati agli ingressi. I valori iniziali dei pesi sono scelti in modo casuale. Ci sono 2 tipi di addestramento: quello supervisionato e quello adattativo (o non supervisionato). Nell'addestramento supervisionato alla rete vengono mostrati delle coppie ingresso-uscita di esempio. Questo insieme di coppie viene chiamato insieme d'addestramento. Per ciascuna di tale coppia preleviamo l'ingresso per la rete neurale, la rete valuta l'uscita, questa viene confrontata con quella della coppia. Se tra i due valori c'è una differenza, allora vengono modificati i pesi della rete in modo da ridurre lo scarto tra l'uscita ed il valore fornito dall'esempio. Questo scarto è l'errore che a seconda dell'algoritmo d'apprendimento ne può essere valutato il quadrato oppure anche il cubo. Nell'addestramento adattativi invece, la rete viene addestrata senza la conoscenza delle uscite.desiderate cercando di trovare una logica di raggruppamento degli ingressi. Noi ci aspettiamo che la rete, dopo essere stata addestrata, sia in grado di generalizzare cioè sia in grado di produrre appropriate uscite anche quando utilizzati degli ingressi non inclusi nell'insieme d'addestramento. La funzione che viene implementata dalla rete neurale dipende dalla topologia (numero di neuroni ed il modo in cui vengono connessi tra loro) e dai pesi di ciascun neurone. Se una rete non ha cicli viene chiama rete con connessione in avanti, se invece se non vi sono solo connessioni in avanti viene chiamata rete ricorrente. Le reti con neuroni disposti a strato, e con gli elementi di ciascuno strato riceventi gli ingressi dallo strato precedente, vengono chiamate reti multistrato con connessioni in avanti. Secondo la notazione, quando ci si riferisce a tali reti, con numero di strati, si intendono quelli dei pesi. Nella figura 2 è mostrata un esempio di tale rete. Nell'adattamento di curve, il numero di gradi di libertà deve essere minore del numero di punti di cui si dispone. Le reti neurali assumono un comportamento simile. Nelle reti, il numero di gradi di libertà coincide con il numero di pesi. Ad esempio, in una rete 2 a due strati di pesi con connessioni in avanti, indicando con N il numero di ingressi, H il numero di unità nello strato nascosto, ed M il numero di uscite, il numero di gradi di libertà della rete e quindi dei pesi è: $H \times (N + M)$. Se i neuroni non utilizzino come funzione non lineare la soglia e che l'insieme d'addestramento non sia linearmente separabile, possiamo usiamo avere un errore utilizzando una sola unità nascosta. Altrimenti aumentando il numero di neuroni nello strato nascosto e andando oltre questo valore, l'errore di classificazione non tende al suo minimo ma avviene ciò che viene chiamato sovra-adattamento cioè la rete si adatta al rumore che si aggiunge ai vettori dell'insieme d'addestramento.

Dopo l'addestramento, la rete viene provata su vettori non utilizzati in fase di addestramento. Questi vengono chiamati insieme di validazione. L'errore valutato sull'insieme di validazione sovrastima quello sull'insieme d'addestramento. Per via di esperienze, si trovato opportuno che su tutti i vettori a disposizione, ne vengano utilizzati 2/3 come insieme d'addestramento mentre il restante 1/3 come insieme di validazione.

1.2.5 Addestramento di tipo supervisionato

In questo tipo di addestramento, la rete conoscendo, per ciascun ingresso, i valori desiderati dell'uscita, corregge i pesi mediante un meccanismo di retropropagazione dell'errore di uscita della rete calcolato come differenza tra il valore desiderato ed il valore vero. Il meccanismo con cui vengono modificati ip pesi, viene chiamato funzione d'apprendimento. Questa procedura viene eseguita più volte. Esistono diversi algoritmi per la correzione dei pesi, questi sono trattati in seguito. Non sempre l'errore converge a zero durante l'addestramento. Questo può essere dovuto ad ambiguità nell'insieme d'addestramento cioè possono esserci due o più vettori d'ingresso per i quali è prevista la stessa uscita. Un altro motivo per cui l'errore non converge a zero è quello della insufficienza di dati nell'insieme d'addestramento. Quindi occorre agire sull'insieme d'addestramento oltre che sulla topologia. Le reti MLP ed RBF vengono addestrate tramite algoritmi d'addestramento supervisionato.

1.2.6 L'addestramento non supervisionato o adattativo

Nell'addestramento non supervisionato alla rete vengono posti gli ingressi senza le uscite desiderate. Si tratta di organizzare gli ingressi nel modo opportuno. T. Kohonen ha sviluppato una rete auto-organizzante per questo tipo di addestramento. Questa rete ha un solo strato di neuroni che vengono opportunamente raggruppati. Il problema consiste proprio nel trovare il migliore raggruppamento.Tale tipo di addestramento viene anche chiamato adattativi perché sono i neuroni che modificano la topologia adattandosi agli ingressi. Questo modo di addestrare è tutto argomento di ricerca. In pratica, si tratta nel simulare il cervello umano che deve prendere delle decisioni per situazioni nuove. Si è preferito utilizzare l'addestramento supervisionato perché offre buoni risultati.

1.2.7 La formula di regressione di Hermite

Definiamo i polinomi ortogonali di Hermite [1]:

$$
\begin{aligned}
&H_0(x) = 1\\
&H1(x) = 2x\\
&\vdots\\
&Hr(x) = 2[rHr-1(x) - (r-1)Hr-2(x)]
\end{aligned}
$$

Questi polinomi sono ortogonali nell'intervallo $]-\infty, +\infty[$ rispetto alle funzioni peso:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{1.9}$$

e sono normalizzati tramite la seguente relazione:

$$\int_{-\infty}^{+\infty} H_r^2(x)\phi^2(x)dx = r!\pi^{-\frac{1}{2}}2^{r-1} \tag{1.10}$$

Le relazioni ortonormali di Hermite sono così definite:

$$h_r(x) = (r!)^{-\frac{1}{2}}\pi^{\frac{1}{4}}2^{\frac{r-1}{2}}H_r(x)\phi(x) \tag{1.11}$$

con:

$$-\infty < x < +\infty$$

La funzione f^* senza brusche variazioni può essere così definita:

$$f^*(x) = \sum_{r=1}^{R} c_r^* H_r(x) \tag{1.12}$$

La derivata prima della funzione ortonormale di Hermite è facilmente calcolabile:

$$h_r^{'}(x) = (2r)^{\frac{1}{2}}h_{r-1}(x) - xh_r(x) \tag{1.13}$$

$$f^{*'}(x) = \sum_{r=1}^{R} c_r^* h_r^{'}(x) = \sum_{r=1}^{R}[(2r)^{\frac{1}{2}}h_{r-1}(x) - xh_r(x)] \tag{1.14}$$

1.2.8 L'algoritmo di discesa del gradiente coniugato (CGRD)

[1] Questo è un algoritmo per l'addestramento di reti neurali più veloce rispetto alla discesa lungo il gradiente negativo ed inoltre e non richiede

l'impostazione del learning rate. La ricerca del minimo dell'errore è basata sulla determinazione di un insieme di direzioni.che non interferiscano tra di loro chiamate direzioni coniugate. Per ciascuna di queste direzioni si procede sino a quando non è stato incontrato un minimo riconoscibile per via del fatto che in tale punto il gradiente della funzione da minimizzare è ortogonale al valore assunto dalla funzione. Fissiamo un punto di partenza P e determiniamo lo sviluppo in serie di Taylor della funzione f centrato nel punto P:

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i}\Big|_P dx_i + \frac{1}{2}\sum_{i,j}\frac{\partial^2 f}{\partial x_i x_j}dx_i dx_j + \ldots \approx c - \mathbf{b^T x} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{x^T A x} \tag{1.15}$$

dove:

$$c = f(\mathbf{P})$$

$$\mathbf{b} = -\nabla f\Big|_{\mathbf{P}}$$

$$[\mathbf{A}]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

In base alle posizioni fatte, il gradiente può essere calcolato nel seguente modo:

$$\nabla f = \mathbf{Ax} - \mathbf{b}$$

il gradiente dipendentemente dalla direzione lungo cui ci si muove

$$\delta(\nabla f) = \mathbf{A}\delta\mathbf{x}$$

Supponiamo di esserci mossi lungo la direzione descritta dal vettore **u** verso un minimo e per il passo successivo ci muoviamo lungo un'altra direzione descritta dal vettore **v**. La condizione secondo la quale il movimento lungo il vettore **v** non copre il percorso fatto in precedenza lungo il vettore **u** consiste nel fatto che il gradiente della funzione f deve essere perpendicolare al vettore **u**:

$$\mathbf{u}\delta(\nabla f) = \mathbf{uAv} = 0$$

Se i vettori **u** e **v** soddisfano questa relazione allora vengono chiamati **direzione coniugata**. Se questa relazione è soddisfatta da un insieme di vettori questi vengono chiamati **insieme coniugato**. E' possibile avere un insieme di N direzioni coniugate indipendenti dalla formula ricorsiva di Powell:

$$\mathbf{u_{i+1}} = -\mathbf{g_{i+1}} + \beta_{i+1}\mathbf{u_i} + \mu_{i+1}\mathbf{u_d}$$

dove:

$$\mathbf{g_i} = \nabla f\Big|_{\mathbf{P}}$$

e con i seguenti valori iniziali:

$$d = i = 1$$

$$\mathbf{u_0} = \mathbf{g_0}$$

e con:

$$\beta_{i+1} = \frac{\mathbf{g_{i+1}^T}[\mathbf{g_{i+1}} - \mathbf{g_i}]}{\mathbf{u_i^T}[\mathbf{g_{i+1}} - \mathbf{g_i}]}$$

$$\mu_{i+1} = \frac{\mathbf{g_{i+1}^T}[\mathbf{g_{d+1}} - \mathbf{g_d}]}{\mathbf{u_d^T}[\mathbf{g_{d+1}} - \mathbf{g_d}]}$$

Le seguenti condizioni sono chiamate **condizioni di ripartenza**:

$$|\mathbf{g_{i+1}^T g_i}| \geq 0.2 \left\|\mathbf{g_{i+1}}\right\|^2$$

$$-\frac{1}{2}\left\|\mathbf{g_{i+1}}\right\| \leq \mathbf{u_{i+1}^T g_{i+1}} \leq -0.8 \left\|\mathbf{g_{i+1}}\right\|^2$$

Quando queste sono soddisfatte poniamo:

$$d = i$$

$$\mathbf{u_{i+1}} = -\mathbf{g_{i+1}} + \beta_{i+1}\mathbf{u_i}$$

Reti neurali facenti uso del CGRD

aNet è una rete neurale ad un singolo strato nascosto i cui neuroni hanno una delle seguenti funzioni d'attivazione: lineare, sigmoide, tangente iperbolica, sinusoide e polinomi di Hermite. Le unità dello strato uscita sono tutte lineari. L'addestramento avviene tramite l'algoritmo di discesa del gradiente coniugato che abbiamo appena visto. Questo algoritmo presenta risulta essere più veloce, rispetto alla discesa lungo il gradiente negativo. La possibilità di utilizzare i polinomi di Hermite come funzione di attivazione dei neuroni, ha i seguenti vantaggi: · la possibilità di calcolarli facilmente in modo ricorsivo · la possibilità di calcolarne facilmente la derivata prima · la rete neurale presenta un uscita senza brusche variazioni con immunità al rumore additivo agli ingressi Vediamo come avviene l'addestramento in aNet avente polinomi di Hermite come funzione d'attivazione delle unità nascoste. La funzione d'attivazione della k-esima unità nascosta è:

Dove hrk(x) sono le prime R funzioni ortonormali di Hermite e dove c*rk sono i relativi coefficienti. Il numero di unità nascoste ed il numero di funzioni ortonormali di Hermite sono scelti in modo sperimentale. Il numero R dei pesi c*rk appartenenti allo strato nascosto sono settati prima di iniziare l'addestramento in modo che f*k(x) risulti essere l'approssimazione di

una funzione a forma di campana perché, per via sperimentale, si è trovato che questa consente migliori generalizzazioni. Durante l'addestramento, i coefficienti c*rk sono considerati come dei pesi e quindi variano con questi utilizzando l'algoritmo CGRD con le condizioni di riavvio che abbiamo già visto. Le funzioni di attivazione, quindi vengono continuamente cambiate durante l'addestramento finchè non viene trovato il minimo errore in corrispondenza all'insieme d'addestramento. La scelta di utilizzare l'algoritmo CGRD è dovuta ai buoni risultati sperimentali ottenuti qui riassunti:

- alta velocità d'apprendimento
- ottime prestazioni per funzioni polinomiali ad alto grado
- contenuta necessità di memoria infatti la memoria occupata cresce linearmente

con il numero di variabili richieste Indicando con M il numero di unità d'uscita, H il numero di unità nascoste e con N il numero di unità d'ingresso e con R il grado dei polinomi di Hermite, l'algoritmo CGRD ha un costo computazionale proporzionale a: R(N+H+M) cioè al numero di pesi della rete. Però i coefficienti c*rk non dovrebbero essere considerati dei veri pesi.

Capitolo 2

La soluzione proposta

Un documento è HTML viene prima classificato, cioé viene determinato se appartiene oppure no al tipo di documento per il quale di intendono estrarre le informazioni desiderate. Successivamente procediamo all'estrazione dei campi. In questo modo le informazione possono essere strutturate per essere tradotte in formato XML. Nel seguito analizziamo distintamente la classificazione e l'estrazione dei campi. I metodi descritti presentano il vantaggio di essere indipendenti dalla lingua del documento HTML e si basano sulla disponibilitá di un insieme di documenti da utilizzare come esempi. Inoltre non é richiesto il collegamento ad alcun tipo di base di dati con notevole risparmio di memoria.

2.1 La classificazione di un documento HTML

Per codificare i documenti, a questi, vengono tolti i tag HTML e le parole di uso cumune allo scopo di eliminare ad informazioni che costituiscono un rumore. In questo modo perveniamo al solo testo. Successivamente lo codifichiamo tramite un vettore ad elementi binari. Il riconoscimento della classe di appartenenza del documento avviene tramite l'uso di una rete neurale ricevente in ingresso tale vettore.
Ogni elemento del vettore é associato ad una parola facente parte di un opportuno insieme che chiameremo **insieme delle parole discriminanti** questo assume il valore 0 se la parola é presente nel testo e 1 se invece é presente.
Il documento da classificare deve essere diverso da quelli facenti parte di tali esempi.
Nel seguito verrá descritto il modo in cui si puó pervenire all'insieme delle

parole discriminanti.

2.1.1 La scelta delle parole discriminanti

La formazione di tale insieme avviene tramite, la composizione di un insieme di documenti HTML che chiameremo **insieme d'addestramento**. Per ogni documento dell'insieme d'addestramento, eliminiamo i tag e le parole di uso comune. Successivamente formiamo un elenco composto da tutte le parole appartenenti ai documenti dell'insieme d'addestramento. Tale elenco viene ordinato in base ad una misura del grado di associazione delle parole al tipo a cui appartiene il documento. Selezione di un gruppo di N parole per caratterizzare ciascun documento.

Composizione di un insieme di documenti HTML chiamato insieme d'addestramento

I documenti HTML appartenenti a tale gruppo devono devono essere diversi da quelli che verranno analizzati dal processo di classificazione completo. Le classi a cui appartengono i documenti sono due: quella per cui estrarremo le informazioni cercate e tutte le restanti classi.

Determinazione, per ogni parola, di una misura dell'associazione alla classe di appartenenza

Il coefficente che associamo a ciascuna parola è la **cross-entropia** che fornisce una misura della dipendenza tra due variabili aleatorie. Questo coefficiente assume valori compresi tra 0 ed 1. Il valore 0 indica che tra le due variabili aleatorie sono indipendenti l'una dall'altra mentre il valore 1 indica che la due variabili sono totalmente dipendenti. Per calcolarla definiamo le due variabili aleatorie binarie **Parola** e **Classe**. Dove, per ogni documento, **Parola** assume il valore 1 quando la parola é presente e 2 altrimenti mentre **Classe** assume invece il valore 1 oppure 2 quando il documento analizzato appartiene rispettivamente alla prima oppure alla seconda classe. Definiamo anche il termine N_{ij} con i e j che possono assumere valori 1 o 2, e che indica il numero di volte in cui, su tutto l'insieme d'addestramento, parola assume il valore i e Classe invece assume il valore j. Da tali coefficienti ci costruiamo: $N_{1.}$, uguale alla somma di N_{11} e di N_{12}; $N_{2.}$, uguale alla somma di N_{21} e di N_{22}; $N_{.1}$, uguale alla somma di N_{11} e di N_{21}; $N_{.2}$, uguale alla somma di N_{12} e di N_{22} ed infine **N** uguale alla somma di $N_{11}, N_{12}, N_{21}, N_{22}$.In definitiva **N** coincide con il numero di documenti analizzati.

Quanto detto sinora puó essere riassunto in una tabella che prende il nome di **tabella di contingenza** (Tab. 2.1).

	Classe=1	**Classe**=2	
Parola=1	N_{11}	N_{12}	$N_{1.} = N_{11} + N_{12}$
Parola=2	N_{21}	N_{22}	$N_{2.} = N_{21} + N_{22}$
	$N_{.1} = N_{11} + N_{21}$	$N_{.2} = N_{12} + N_{22}$	$N = \sum_{i,j=1}^{2} N_{ij}$

Tabella 2.1: Tabella di contingenza calcolata per una parola su tutti i documenti dell'insieme d'addestramento

Dividendo per **N** tutti i coefficienti appena calcolati, perveniamo a:

$$\begin{aligned} p_{ij} = N_{ij}/N \quad con: \ i,j \in \{1,2\} \\ p_{1.} = N_{1.}/N \qquad p_{2.} = N_{2.}/N, \\ p_{.1} = N_{.1}/N \qquad p_{.2} = N_{.2}/N \end{aligned} \tag{2.1}$$

Da questi calcoliamo i coefficienti di **entropia**, si tratta di una misura della casualitá della variabile aleatoria. Questi coefficienti assumo valori compresi tra 0 e $\ln 2$. Assume il valore 0 quando la variabile aleatoria assume un valore fisso, se invece tutti i valori sono equamente distribuiti assume il valore massimo $\ln 2$.

Entropia di **Parola**:

$$H(Parola) = -\sum_{i=1}^{2} p_{i.} \ln p_{i.} \tag{2.2}$$

Entropia di **Classe**:

$$H(Classe) = -\sum_{j=1}^{2} p_{.j} \ln p_{.j} \tag{2.3}$$

Entropia fra **Parola** e **Classe**:

$$H(Parola, Classe) = -\sum_{j=1}^{2} p_{ij} \ln p_{ij} \tag{2.4}$$

In fine determiniamo il coefficiente di cross-entropia é:

$$U(Parola, Classe) = 2 * \frac{H(Parola) + H(Classe) - H(Parola, Classe)}{H(Parola) + H(Classe)} \tag{2.5}$$

Dopo aver calcolato il coefficiente di cross-entropia per tutte le parola, ordiniamo l'elenco delle parole, in ordine decrescente, in base a tale coefficiente.

Formazione dell'insieme delle parole discriminanti

Dall'insieme appena ordinato selezioniamo un gruppo di N parole. Per vedere quante parole selezionare, analizziamo la distribuzione dei valori di cross-entropia calcolati. Selezioniamo come ultima parola, quella tale da avere a seguire, una discontinuita oppere un flesso nella distribuzione nella distribuzione dei coefficienti di cross-entropia. In questo modo l'insieme delle parole discriminanti viene notevolmente ridotto con il notevole vantaggio di ridurre la dimensione del vettore con cui viene codificato ogni documento.

Codifica dei documenti in base alle parole discriminanti

Sia i documenti che fanno parte dell'insieme d'addestramento per la rete neurale che quelli da analizzare per determinarne la classe di appartenenza, vengono codificati tramite un vettore binario la cui dimensione coincide con quella dell'insieme d'addestramento. Ciascun vettore di codifica di un documento viene cosruito nel seguente modo: viene fatta una scansione dell'insieme delle parole discriminanti. Nel corrispondente elemento del vettore viene posto 0 se tale parola non è presente, altrimenti viene posto 1.

2.1.2 Il riconoscimento della classe del documento

Facendo uso di una rete neurale opportunamente addestrata possiamo determinare la classe a cui appartiene un documento. La rete riceve in ingesso un documento codificato nello stesso modo in cui si é proceduto in precedenza cioé tramite le parole discriminanti. L' addestramento della rete neurale avviene fornendo alla rete esempi formati da coppie composte da vettori di codifica del documento e la classe a cui appartiene il documento.

2.2 Estrazione dei campi

I tag HTML non portano informazione sulla semantica ma solo sull'impaginazione del testo quindi, per l'estrazione dei campi, vengono tolti. In questo modo l'analisi si basa solo sul testo e precisamente su due aspetti: il contesto ed Il lessico.
Con contesto intendiamo la parola che precede (contesto sinistro) e quella successiva (contesto destro) al campo. Per via sperimentale, infatti si é visto che l'analisi della sola parola che precede o viceversa che segue il campo, é sufficiente per risalire al campo cercato. Il tutto sta nel formare due elenchi

rispettivamente per le parole precedenti e quelle successive. Questi si possono formare creando un insieme di documenti HTML a cui sono stati tolti i tag e gli articoli tale insieme di documenti lo chiamiamo **insieme d'addestramento**, procedendo parola per parola su ciascuno di tali documenti e inserendo nell'elenco del sontesto sinistro le parole che precedono il campo nonché nell'elenco del contesto destro quelle che lo seguono.
Le parole, facenti parte di tali elenchi, però non è detto che figurino, nel documento,accanto al campo cercato e quindi, non si può basare la ricerca del campo sul fatto che ci aspetta di trovalo accanto a tale parola. Per questo motivo ia due elenchi devono essere tolte le parole che figurano molte volte non accanto al campo. Si può allora procedere nell'ordinare gli elenchi in base ad un opportuno coefficiente chiamato **distanza di Tanimoto** [9] ed analizzando la distribuzione di tale coefficiente enll'elenco ordinato per palori decrescenti, eliminare quelle parole che figurano dopo una discontinuita.
Sempre per via sperimentale si é visto che portano informazioni consistenti entrambi i contesti per alcuni campi mentre per altri é sufficiente il solo contesto sinistro.
Poiché gli articoli sono presenti in grandi quantitá nel testo e quindi non porterebbero alcun contributo alla nostra ricerca del campo, questi vengono tolti.
L'analisi del solo contesto non é sufficiente in quanto questa ci fornirebbe solo un insieme di porzioni di testo **candidate** cioé fra le quali, con una buona probabilitá, é presente anche il target. Per ridurre tale insieme poniamo delle condizioni sul campo. Si tratta di euristiche sul lessico o anche sul numero di parole che lo compongono.
Il processo di riconoscimento del campo deve essere fatto su di un documento non presente nell'insieme d'addestramento e consiste dopo aver tolto i tag e gli anticoli, nello scandire il testo rimanente parola per parola prelevando come canditate quelle porzioni di testo che sono precedute da parole facenti parte dell'elenco dei precedessori e seguite (solo nel caso di uso di entrambi i contesti) da quelle facenti parte dell'insieme dei successori. Infine procediamo tramite le euristiche sul lessico per ridurre l'insieme dei candidati.

2.2.1 L'individuazione del contesto

Nel seguito procediamo nella descrizione del modo in cui questi vengono costruiti.
Dopo aver estratto i tag e tolto gli articoli viene fatta una scansione di ogni parola del testo di ogni documento facente parte dell'insieme d'addestramento. Ogni parola che precede il campo target viene postaformazione dell'insieme dei predecessori (ed eventualmente dei successori). Ordinamento

tale insieme in base ad un opportuno coefficente Selezione delle prime parole dell'insieme dei predecessori (ed eventualmente dei successori).

Estrazione delle parole da ogni documento HTML

Ad ogni documento facente parte dell'insieme d'addestramento vengono rimossi i tag dato che questi non portano informazione sulla semantica contenuta nel documento ma solo informazioni sull'impaginazione. Vengono tolti anche i segni di punteggiatura. Non vengono rimosse tutte le parole facenti parte del titolo e del corpo del documento.

Eliminazione dal documento HTML delle parole di uso comune

Occorre evitare di prelevare parole che occorrono frequentemente all'interno del documento a prescindere dal fatto che siano seguite o meno dal campo. Queste sono:

- Gli articoli
- Le congiunzioni
- Le proposizioni semplici e quelle articolate
- I verbi di uso comune

Formazione dell'insieme dei predecessori

Sia le parole facenti parte dei predecessori che quelle facenti parte dei successori devono avere i seguenti requisiti:

- deve figurare accanto al campo almeno 1 volta
- deve figurare non accanto al campo il minor numero possibile di volte

Dalla seconda condizione, ci si rende conto che le parole devono essere selezionate in modo tale che quando una di esse é presente nel documento, é lecito aspettarsi che accanto vi sia il campo cercato.
Dell'insieme fanno parte quelle parole che figurano almeno una volta accanto al campo (a sinistra oppure a destra a seconda del contesto che si intende estrarre) in ciascun documento facente parte dell'insieme d'addestramento a cui sono stati tolti i tag HTML.
Per rispettare tali condizioni si possono selezionare tutte le parole che compaiono accanto al campo almeno una volta.

Ordinamento dell'insieme dei predecessori (oppure dei successori)

Dopo essere pervenuti all'insieme dei predecessori del campo, possiamo valutare, per ciascuna parola di tale insieme, la seguente funzione denominata **distanza di Tanimoto** [1] :

$$F(A,T) = \frac{f(AT)}{f(A) - f(T) - f(AT)} \tag{2.6}$$

Dove, riferendoci a tutto l'insieme d'addestramento:

A = parola appartenente all'elenco appena generato

T = campo

f(A) = numero di occorrenze di A (parola)

f(T) = numero di occorrenze del campo

f(AT) = numero di occorrenze di A accanto al campo Tutte le occorrenze vengono calcolate su tutto l'insieme d'addestramento come se questo fosse un unico documento.

Selezione delle prime parole dell'insieme dei predecessori

Se dovessimo considerare tutte le parole dell'insieme considerato allora si indurrebbe all'errore il processo di riconoscimento del campo. Il numero delle parole da selezionare dipende dalla differenza tra il valore assunto dalla distanza di Tanimoto per una parola e quella successiva: se vi è una differenza più grande rispetto a quella relativa fra le altre parole allora è opportuno troncare l'insieme.
Se si operasse il taglio prima rispetto al punto suddetto, allora si rischierebbe di non trovare il campo, durante il processo di riconoscimento. Viceversa, se il taglio venisse operato oltre la posizione corretta, allora si indurrebbe all'errore riconoscendo come campo porzioni di testo che invece non lo sono.

2.2.2 Le euristiche sul lessico

Il campo, oltre che dal contesto, puó essere individuato da euristiche lessicali. Queste possono essere:

[1] Dati due insiemi non vuoti A e T la distanza di Tanimoto è $F(A,T) = \frac{|A \cap T|}{|A \cup T|}$. Nel nostro caso i due insiemi sono costituiti da coppie di parole adiacenti, avendo considerato il campo come un unica parola. A è l'insieme delle coppie di parole fra cui una è la parola considerata. T è l'insieme delle coppie di parole fra cui una è il campo.

- Il numero di parole che lo compongono
- Il numero di caratteri di ciascuna parola che lo compone
- Il fatto che alcune o tutte le parole che lo compongono abbiano iniziale maiuscola
- La presenza (o assenza) di una sequenza di lettere e/o numeri.
- La terminazione di un periodo con un punto.

Le uristiche possono essere utili anche per scomporre un campo in sottocampi. Il sottocampo puó essere individuato tramite delle parole chiave che delimitano il sottocampo o che costituiscono il sottocampo stesso.

2.2.3 Il riconoscimento del campi

Il riconoscimento del campo avviene tramite la scansione del documento e l'individuazione delle parole che figurano nell'elenco delle parole adiacenti. A seconda del tipo di campo occorre utilizzare:

- Entrambi gli elenchi, quello dei precedessori e quello dei successori.
- solo un elenco, quello delle parole che lo precedono.

Questo processo porta ad un insieme di "candidati" cioé porzioni di testo che possono essere il campo. Per ridurre il numero dei candidati operiamo una selezione basata sulle euristiche.

Riconoscimento di un campo facendo uso del solo contesto sinistro

Disponendo del documento HTML dove si presume sia contenuto il campo da estrarre, dopo l'estrazione delle parole dal documento HTML e l'eliminazione delle parole di uso comune. Procediamo nel seguente modo:
Scandiamo tutte le parole del documento e, per ciascuna di esse, se fa parte dell'insieme dei predecessori, preleviamo le parole che la seguono nella quantitá determinata dalle euristiche e ci posizionami sulla parola successiva a tale sequenza, altrimenti ci posizionamo sulla parola successiva.
Occorre precisare che la scansione avviene a partire dalla prima parola del documento e quindi il campo puó essere individuato solo se questo comincia, a prescindere dal numero di parole che lo compongono, a partire dalla seconda parola.
Se il campo dovesse essere la prima parola del documento l'analisi del solo contesto sinistro non ne permetterebbe l'individuazione. Questo tipo di

problema non si pone utilizando entrambi i contesti.
La presenza di un ordine, nell'elenco dei precedessori porta alla presenza del campo cercato fra i primi elementi facenti parti parte dell'insieme dei candidati. Questa caratteristica consente di ridurre drasticamente la dimensione nell'insieme dei candidati.

Riconoscimento di un campo facendo uso di entrambi i contesti

L'uso di entrambi i contesti consente di ridurre il numero di elementi proposti in uscita fra i quli è presente anche il campo. Indichiano con **Sx** l'insieme, ordinato tramite la distanza di Tanimoto, dei precedessori del campo, con **Dx** l'insieme, anch'esso ordinato tramite la distanza di Tanimoto delle parole successive al campo. Noi procediamo con la costruzione dei seguenti quattro insiemi:

Sx-Dx Porzione di testo, preceduta da una parola facente parte dell'insieme PrecSx e seguita da una parola facente parte all'insieme No.

Sx-No Porzione di testo, preceduta da una parola facente parte dell'insieme PrecSx e seguita da una parola facente parte all'insieme No.

No-Dx Porzione di testo, preceduta da una parola facente parte dell'insieme No e seguita da una parola parola facente all'insieme Succ.

No-No Porzione di testo, preceduta e seguita da una parola facente parte dell'insieme No.

Questi insiemi sono complementari.
I primi tre insiemi contengono informazioni significative. Il quarto, invece deve contenere il minor numero possibile di elementi.
É stato accertato, per via sperimentale, che l'insieme **Sx-Dx**, per ogni documento contenente il campo cercato, è formato da un solo elemento (al piú due in casi sporadici) corrispondente al campo cercato a prescindere dalla presenza o meno di un ordine nei due insiemi **Sx** e **Dx**. Questo costituisce un notevole vantaggio avendo portato ad un solo elemento la dimensione dell'insieme dei candidati.
Quindi, dopo l'Estrazione delle parole dal documento HTML e l'eliminazione delle parole di uso comune. Procediamo nel seguente modo per costruire **Sx-Dx**:
Per ogni parola, se è preceduta da una parola facente parte di **Sx**, prendi la porzione di testo, di dimensione fissata tramite euristiche, che precede un elemento dell'insieme **Dx** e posizionarsi sulla parola immediatamente successiva a quest'ultimo. Altrimenti passa direttamente alla parola successiva.
Procediamo in modo analogo per la costruzione di **Sx-No** e di **Sx-No**.

Capitolo 3

Risultati sperimentali

Risultati dall'analisi del solo contesto sinistro

Nelle 2 seguenti tabelle sono riportati i risultati sperimentali su 100 documenti. Inoltre, quando viene trovato l'elenco di candidati al campo cercato, se il campo è stato trovato ed è quindi presente in tale elenco, allora questo è il primo elemento di tale elenco. Nella tabella 3.1 sono riportati i risultati inerenti i campi che su ogni documento sono presenti in un unica istanza.

Nome del campo	α	β	γ	δ
Data di nascita	86	90,69	8,13	1,16
Indirizzo	81	86,41	4,93	8,64
Nazionalità	32	93,75	0	6,25
Obblighi di leva	17	88.24	11,76	0
Numero di telefono	76	85,52	11,84	2,63
Numero del cellulare	38	81,57	7,89	10,52
Numero del fax	7	100	0	0
Indirizzo e_mail	78	83,3	2,56	14,10

Tabella 3.1: Tabella di contingenza inerente ai campi che, in ogni documenti sono presenti in un unica istanza

Dove le colonne hanno i seguenti significati:

α : Numero di documenti che contengono il campo

β : Percentuale di documenti nei quali il campo è stato identificato correttamente

γ : Percentuale di documenti nei quali è stato commesso un errore nell'identificazione del campo

δ : Percentuale di documenti nei quali non è stato trovato il campo

Nella tabella 3.2 sono invece riportati i risultati sperimentali inerenti ai campi che possono avere più istanze presenti nel documento. In questa tabella sono presenti 2 colonne i più (γ e η).

Nome del campo	α	χ	η	λ
Titolo di studio	92	97,82	3,13	81,15
Tipo lavoro	47	95,74	6,17	65,23
Attività lavorativa	76	100	18,40	73,32
Conoscenze acquisite	82	100	17,92	92,68

Tabella 3.2: Tabella di contingenza inerente ai campi che, in ogni documenti sono presenti in più di una istanza

dove:

χ :Valore medio del numero di documenti in cui è stato trovato qualcosa

η : Valore medio del numero di candidati trovati in ciascun documento

λ : Valore medio della percentuale di campi identificati in ciascun documento

Bibliografia

[1] S. Gaglio, G. Pilato, F. Sorbello, G. Vassallo. Using the Hermite regression formula to design a neural architecture with automatic learning of the hidden activation functions, Lecture Notes in Artificial Intelligence N.1792 - Springer Verlag - pp.226-237, 2000

[2] Eui-Hong Sam Han, George Karypis, Vipin Kumar. Text Categorization Using Weight Adjusted k -Nearest Neighbor Classification, Lecture Notes in Computer Science, 2035, 53-59, 2001

[3] Andrew McCallum, Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification, AAAI-98 Workshop on Learning for Text Categorization, 1998.

[4] Heide Brucher, Gerhard Knolmayer, Marc-Andrè Mittermayer. Document Classification Methods for Organizing Explicit Knowledge, Proc of OKLC, 5-6 April 2002

[5] Nicholas Kushmerick. Wrapper Induction: Efficiency and Expressiveness, Artificial Intelligence J. 118(1-2):15-68 (special issue on Intelligent Internet Systems), 2000

[6] Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites, Proceedings of 27th International Conference on Very Large Data Bases, pp109-118, 2001

[7] Wei Han, David Buttler, Calton Pu. Wrapping Web Data into XML. Sigmod Record, 2001, 30(3): 33-38

[8] Cambridge Universit. Numerical recipes in c: the art of scientific computing 1992

[9] K. R. Sloan Jr., Steven L. Tanimoto, Progressive Refinement of Raster Images, IEEE Transactions on Computers, Volume 28, Number 11, pp. 871-874, November 1979.