TECNICHE DI WEB MINING PER SUPPORTARE L'ATTIVITA' DI NAVIGAZIONE IN RETE

G. Convertino[1], L. Di Pace[1], P. Leo[1], A. Maffione[1],
D. Malerba[2] e G. Vespucci[2]

Per risolvere i problemi causati dall'eccesso di informazioni disponibili su Internet occorrono nuovi strumenti di ricerca e di supporto alla navigazione. Questo articolo presenta i risultati di uno studio su tecniche di Web Mining, ovvero l'applicazione di algoritmi di Data Mining per scoprire le regolarità presenti in risorse disponibili su Web. In particolare, vengono proposti i primi risultati relativi al problema della classificazione automatica di pagine Web in base al loro contenuto testuale. I classificatori sono costruiti a partire da pagine rappresentative fornite da un singolo utente o da un gruppo durante l'attività di ricerca in rete. Lo studio ha portato alla realizzazione di un sistema di Web Mining, denominato WebClass, che offre servizi di assistenza alla navigazione.

1. INTRODUZIONE

Gli strumenti di ricerca di informazioni su Internet hanno subito una rapida evoluzione. Si è passati da strumenti legati alla struttura dei file system, come il WAIS e il GOPHER, a primitivi motori di ricerca come ARCHIE. L'affermazione del Web ha segnato la comparsa di servizi di ricerca più sofisticati come quelli offerti da motori di ricerca come Altavista, Excite, Yahoo, Lycos, e dei metasearcher (ad esempio ProFusion) in grado, questi, di combinare sinergicamente gli sforzi di più motori di ricerca. La tecnologia degli agenti software ha permesso di controbilanciare la genericità di un motore di ricerca e di fornire dei tool personalizzabili da parte del singolo utente o di gruppi, come ad esempio WebCompass, Net Attachè, Smart Bookmarks e Teleport Pro.

L'esigenza di disporre di strumenti generali e allo stesso tempo personalizzabili ha recentemente portato sia a tentativi di integrazione di grandi e sofisticati meccanismi di ricerca *general purpose* con servizi personalizzabili (vedi MyYahoo) sia al potenziamento degli agenti di ricerca personali che offrono anche dei servizi tipici dei metasearcher (vedi abstracting, sintesi e integrazione dei risultati). In questa prospettiva si colloca il lavoro di ricerca descritto in questo articolo. L'obiettivo è quello sviluppare un agente di ricerca *tematico* che offra i propri servizi ad un gruppo di utenti con interessi navigazionali omogenei, combinando l'esigenza di "generalità" con quella di "specificità" rispetto a un

[1] Java Technology Center, IBM Semea Sud, Bari, email: pietro_leo@it.ibm.com.
[2] Dipartimento di Informatica, Università degli Studi, Bari, email: malerba@di.uniba.it.

ristretto numero di argomenti. In questo articolo l'attenzione è rivolta a uno dei servizi messi a disposizione da tale agente: la classificazione automatica di pagine Web in base al loro contenuto testuale. A tale scopo è stata esplorata la possibilità di applicare tecniche di Data Mining per costruire i classificatori a partire da pagine rappresentative fornite da un utente o un gruppo di utenti durante l'attività di ricerca in rete. L'architettura del prototipo sviluppato, le tecniche di Data Mining adottate, e i risultati ottenuti da una prima sperimentazione sono descritti nel seguito.

2. WEB MINING: LO STATO DELL'ARTE

In analogia con il termine Data Mining, che denota l'applicazione di specifici algoritmi per individuare "regolarità" nei dati di un database, l'espressione Web Mining si riferisce all'applicazione di procedure analoghe per estrarre automaticamente informazioni dalle risorse presenti nel Web (sia documenti che servizi). L'obiettivo del Web Mining trova giustificazione nell'opinione diffusa che l'informazione presente nel Web è sufficientemente strutturata da consentire una efficace applicazione di tecniche statistiche e di apprendimento automatico [4]. Di fatto, è possibile scomporre l'obiettivo del Web Mining nei seguenti sottobiettivi:

1. *Scoperta di risorse*, cioè la localizzazione di documenti e servizi nella rete.
2. *Estrazione di informazioni* a partire dalle risorse individuate.
3. *Generalizzazione*, cioè l'apprendimento relativo alla struttura stessa del Web.

Per quanto riguarda il primo sottobiettivo, i tradizionali strumenti di ricerca di informazioni in rete mettono a disposizione dell'utente finale varie modalità per la localizzazione di risorse. Queste spaziano dal paradigma "*classificazione e browsing"* ai puri meccanismi di ricerca di tipo *Boolean* o *free-text* (cioè, a parole chiave) ad approcci misti. Attualmente gli strumenti disponibili per la scoperta di risorse Web ricorrono ai Web Robot, detti Spider, che scandiscono milioni di documenti Web e costruiscono indici di ricerca basati sulle parole in essi contenute. Le modalità appena descritte sono *general purpose*, e spesso gli indici costruiti includono informazioni irrilevanti per uno specifico utente o gruppo di utenti. Gli approcci più sofisticati, come quello presentato in questo lavoro, applicano tecniche di categorizzazione automatica al fine di ottenere una classificazione personalizzabile dei documenti Web indicizzati dai motori di ricerca [7]. Tali tecniche sono peraltro utili anche al raggiungimento del secondo sottobiettivo, l'estrazione di informazioni. In questo caso i testi in linguaggio naturale (come articoli, descrizioni di brevetti, ecc.) vengono trasformati in rappresentazioni strutturate predefinite, dette *template*, che, quando riempite, rappresentano un estratto dell'informazione presente nel testo [11]. Una volta che è stato automatizzato il processo di scoperta e di estrazione di informazioni sul Web, il terzo sottobiettivo impone che si generalizzino gli stessi processi

navigazionali nel Web. Lo studio riportato in questo articolo va anche in questa direzione, in quanto affronta uno dei maggiori ostacoli: la classificazione delle pagine Web in base al loro contenuto.
In questo scenario è possibile considerare l'applicazione di diverse tecniche di Data Mining al fine di apprendere le categorie di pagine ritenute "interessanti" per degli utenti Web, cioè un vero e proprio *profilo di interessi*. Tale profilo può essere quindi utilizzato in diverse circostanze, ad esempio per suggerire in anticipo il contenuto (la "classe") dei documenti indirizzati dai link contenuti nella pagina Web corrente.
Negli ultimi anni sono stati realizzati vari sistemi in grado di apprendere gli "interessi" di un utente Internet. NewsWeeder [5], ad esempio, è un sistema per il filtraggio di News Internet che apprende ad attribuire un punteggio per ogni documento a partire da pagine valutate dal un utente in funzione del proprio livello di interesse. WebWatcher [1] è un sistema interattivo che aiuta i propri utenti a localizzare informazioni di interesse nel Web durante l'attività di browsing. Ciò è realizzato acquisendo dall'utente un insieme di parole chiave, suggerendo link e ricevendo un feedback sul grado di interesse per le pagine visitate. Letizia [6] è un Web Browser Assistant in grado di indurre gli interessi dei propri utenti relativamente alle pagine Web osservandone l'attività di browsing e quindi di esplorare in anticipo il Web con una strategia di ricerca in ampiezza. Syskill & Webert [8] è un'altro sistema che raccoglie valutazioni del tipo "interessante"/"non-interessante" circa le pagine Web visitate e apprende il profilo di interessi dell'utente. In quest'ultimo lavoro sono state confrontate sei diverse tecniche per la costruzione automatica del profilo, e si è potuto concludere che i classificatori Bayesiani "naive" offrono dei vantaggi rispetto agli altri nei domini considerati. Un altro confronto è stato effettuato da Quek [9] considerando anche le informazioni di "struttura" direttamente presenti nelle pagine Web, come la posizione del testo all'interno del documento (titolo, paragrafo, ecc.) e le informazioni tipografiche (grassetto, corsivo, etc.).

3. L'ARCHITETTURA DEL SISTEMA

I sistemi di assistenza al reperimento di informazioni su Internet tipicamente estendono le funzionalità del browser e forniscono i loro servizi in locale o in remoto. Nel nostro caso, l'architettura all'interno della quale sono collocati i servizi di assistenza è del tipo "suite", cioè è composta da un insieme di moduli integrati che forniscono una soluzione "scalabile" rispetto a esigenze crescenti in termini di funzionalità richieste dagli utenti. In particolare, la funzionalità di classificazione automatica di pagine Web fornita dal modulo *WebClass* è stata integrata nell'architettura per Intermediari Web *WBI* (*Web Browser Intelligence*) (vedi Figura 1) [2].
WBI estende architetturalmente le funzionalità offerte da un HTTP/Proxy Server,

svolgendo un'azione di intermediazione tra il Web Browser e Internet. In particolare, esso cattura ogni richiesta di accesso ad Internet proveniente dal Web Browser e quindi realizza le proprie funzionalità manipolando e filtrando opportunamente tali richieste attraverso un certo numero di agenti software specializzati appartenenti a tre diverse tipologie: Monitor, Editor e Generator (MEG). Gli agenti di tipo Monitor permettono di analizzare il contenuto del flusso di dati che intercorre tra il Web Browser e Internet al fine di apprendere pattern ricorrenti e altro. Gli Editor intercettano il flusso di comunicazione e sono in grado di modificare le risposte alle richieste di accesso ad Internet formulate dall'utente. Infine i Generator possono aggiungere dinamicamente nuovo contenuto informativo alle risposte ottenute via Internet. In generale l'insieme di MEG opportuni da attivare a seguito di una data richiesta da parte del Web Browser è determinato da un semplice meccanismo di configurazione a regole.
Utilizzando l'infrastruttura architetturale "a intermediario" appena descritta sono state realizzate varie applicazioni che spaziano dall'assistenza al browsing, ad avanzati meccanismi di document caching per Internet browser, alla gestione dei Cookie e così via. All'interno dell'architettura appena descritta sono state anche collocate le funzionalità di classificazione automatica definite all'interno di WebClass. Queste funzionalità sono governate da un opportuno set di MEG.
Attraverso i nuovi servizi offerti da WebClass l'architettura WBI potrà fornire i necessari strumenti per realizzare svariati servizi di assistenza rivolti sia a singoli utenti (nella configurazione "Personal Assistant") o gruppi di utenti (nella configurazione "Servizio di Assistenza"). Ad esempio si potranno classificare in anticipo le pagine Web man mano visitate dall'utente, effettuare esplorazioni autonome e off-line, abstracting di pagine, come pure semi-automatizzare l'organizzazione del bookmark dei riferimenti ai siti Internet di un singolo utente o di un gruppo di utenti.

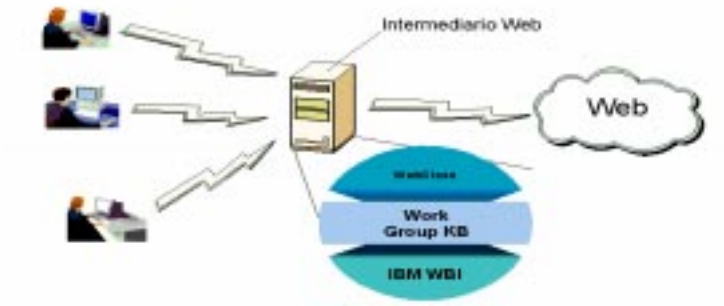

Figura 1: L'architettura del sistema.

4. LA CLASSIFICAZIONE DI PAGINE WEB

La costruzione di un *profilo di interessi* relativo a pagine Web passa per la soluzione di due problemi: decidere quale informazione è interessante e determinare le sue modalità di estrazione dalle stesse pagine Web.
Per quanto riguarda il primo problema si assume che la rilevanza di una pagina Web dipenda essenzialmente dal suo contenuto testuale, quindi criteri di giudizio "esterni", come il carattere di novità del documento o l'affidabilità dell'informazione, ed eventuali contenuti di tipo non testuale della pagina, non vengono considerati. Inoltre, si assume che l'utente sia in grado di specificare un insieme di *classi* corrispondenti ai vari argomenti di interesse, e di fornire un insieme di esempi significativi per ciascuna delle classi (insieme di addestramento).
Per il secondo problema è importante definire sia il *linguaggio di rappresentazione* da utilizzare per descrivere le parti testuali delle pagine HTML e sia le tecniche di Data Mining in grado di generare la conoscenza classificatoria, cioè il profilo di interessi, da utilizzare in un processo di classificazione automatica di pagine Web.

4.1 La scelta degli attributi

La rappresentazione delle pagine Web avviene mediante coppie attributo-valore, dove un *attributo* corrisponde a qualche parola estratta dalle pagine di addestramento fornite dall'utente, mentre il *valore* corrispondente è calcolato utilizzando la frequenza relativa (TF, Term Fequency) della parola all'interno della pagina, eventualmente pesata in modo da tener conto della posizione della parola all'interno delle strutture HTML (titolo, sottotitolo, grassetto, link, e così via). Ad esempio, per descrivere le classi *Astronomy*, *Jazz*, *Auto* e *Moto*, usate nella sperimentazione riportata in seguito, il sistema potrebbe considerare i seguenti attributi: *solar*, *earth*, *jazz*, *blue*, *vehicle*, *fuel*, *motorcycle*, *bike*. In quest'ultimo caso una pagina Web viene descritta da un vettore di otto valori numerici, uno per ogni attributo. Gli attributi sono individuati dal sistema estraendo in modo opportuno un certo numero di parole dall'insieme delle pagine Web di addestramento. Questa fase di *preprocessing* è basata su una variante del TF-IDF [5] in grado di tenere in conto sia del concetto di "classe" sia dell'effetto di *spamming,* tipico delle pagine Web. Inizialmente, tutte le pagine Web di addestramento sono analizzate lessicalmente. Quindi sono rimosse tutte le parole (*token*) troppo corte e le stringhe alfanumeriche, così come le *stopword* (articoli, preposizioni, congiunzioni). In questa fase di preprocessing la posizione dei token all'interno della struttura HTML non viene considerata.
L'insieme degli attributi che complessivamente saranno utilizzati per descrivere tutte le pagine di addestramento viene determinato secondo la modalità di seguito

descritta a partire dai token estratti dalle varie pagine. In particolare, data la *j*-esima pagina di addestramento della *i*-esima classe, per ogni token *t* appartenente alla pagina viene calcolata la frequenza del token nel documento, denotata **TF(i,j,t)**. Per ogni classe *i* e per ogni token *t* vengono individuati i valori massimi **MaxTF(i,t)** fra tutti i valori **TF(i,j,t)** relativi agli esempi-pagine della classe. Per ogni classe *i* e token *t* viene inoltre determinato il valore **PF(i,t)** (Page Frequency) che rappresenta la percentuale di documenti della classe *i* contenenti il token *t*. L'insieme di tutti i token estratti dalle varie pagine di una data classe *i* definisce una sorta di *dizionario "empirico" di classe* che può essere utilizzato nel descrivere gli argomenti legati a tale classe. E' quindi possibile ordinare ogni dizionario in modo decrescente rispetto al prodotto $\mathbf{MaxTF(i,t)*PF(i,t)^2}$ di seguito denominato MaxTF-PF^2 (Max Term Frequency - Square Page Frequency). Così facendo, l'insieme delle parole più comunemente utilizzate all'interno delle pagine di una data classe appariranno nelle prime posizioni del dizionario di classe ordinato. Alcune di queste parole sono sicuramente parole gergali e tipiche della classe mentre altre sono parole comuni e utilizzate anche in pagine di altre classi e pertanto definibili *quasi-stopword*. Al fine di spostare le quasi-stopword nelle parti "basse" dei dizionari di classe il valore MaxTF-PF^2 di ogni token *t* verrà moltiplicato per un fattore **1/CF(t)**, dove **CF(t)** (Category Frequency) è il numero dei *dizionari di classe* nelle quali il token *t* compare. Riordinando i dizionari così ottenuti si noterà che le parole più rappresentative di ogni classe rispetto al contesto delle classi considerate compariranno nelle prime posizioni e potranno quindi essere considerate come attributi di classe. Ad esempio, in Tabella I sono riportate le parole che sistema WebClass pone nelle prime 8 posizioni dei dizionari di classe *Astronomy*, *Jazz*, *Moto* e *Auto* quando determina l'insieme di attributi da utilizzare per descrivere le pagine considerate nella sperimentazione riportata in seguito. In questo caso sono state utilizzate per ogni classe 16 pagine di addestramento estratte casualmente dall'ontologia di classificazione del motore di ricerca Yahoo.

Tabella I: Esempio di token estratti automaticamente da WebClass.

Astronomy	Solar	Earth	Moon	Planet	Sun	Atmosphere	Telescope	Eclipse
Jazz	Jazz	Blue	Note	Music	Record	Album	Rythm	Recording
Auto	Vehicle	Fuel	Cylinder	Wire	Car	Oil	Question	Clean
Moto	Motocycle	Bike	Road	Honda	Ride	Yamaha	Address	Tire

Ottenuti i dizionari di classe questi vengono uniti e viene determinato un set di attributi unico e rappresentativo di tutte le classi. Il numero complessivo di attributi da considerare è variabile e scelto dall'utente. In Figura 2 sono schematizzate le fasi di preprocessing.

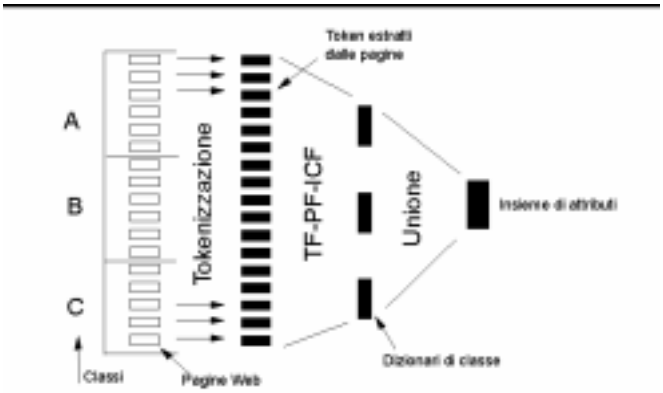

Figura 2: Processo di estrazione degli attributi.

4.2 La rappresentazione delle pagine Web

Il set di attributi così determinato è utilizzato per descrivere le pagine Web. Il valore da associare agli attributi è calcolato considerando la frequenza relativa della parola corrispondente rispetto al numero complessivo delle parole contenute nella pagina. Tale frequenza è quindi pesata in modo da considerare la posizione occupata dalla parola all'interno dei tag presenti nella struttura HTML della pagina, come <Title>, <H1>, <H2>, e così via. L'obiettivo è quello di enfatizzare il valore della frequenza relativa di una parola che compare nel titolo, o in altre strutture. Attualmente il peso dell'informazione di tipo strutturale di una pagina può contribuire nei seguenti modi: *nessuna influenza* (peso 1.0), *influenza additiva* (il peso attribuito ad una parola è ottenuto come somma dei pesi associati alle varie strutture HTML in cui la parola occorre), e *influenza moltiplicativa* (il peso attribuito ad una parola è ottenuto come prodotto dei pesi associati alle varie strutture HTML in cui la parola occorre).

4.3 I metodi di classificazione

L'interazione dell'utente o del gruppo di utenti con WebClass avviene in due passi: inizialmente l'utente naviga nel Web e colleziona riferimenti a pagine significative delle classi di interesse (pagine di addestramento). Nel secondo passo il sistema assiste l'utente nella navigazione classificando autonomamente le pagine Web. La decisione di sospendere la raccolta delle pagine di addestramento e di invocare i servizi di assistenza è responsabilità dell'utente o dell'amministratore del gruppo di utenti.

La predizione della classe di appartenenza di una pagina Web è realizzata attraverso tre modalità alternative:

1. *alberi di decisione* costruiti per induzione a partire dalle pagine di

addestramento;
2. distanza dai *prototipi di classe* costruiti a partire dalle pagine di addestramento;
3. algoritmi di *nearest-neighbour* [3] basati sulla valutazione della similarità rispetto alle pagine collezionate.

Le prime due modalità prevedono la precostruzione di classificatori (alberi di decisione o prototipi) in una fase di addestramento (*training* del sistema). Al contrario, la terza modalità non precostruisce classificatori espliciti per le classi, bensì usa le pagine selezionate direttamente durante la fase predittiva. L'intero processo di classificazione è descritto in Figura 3.

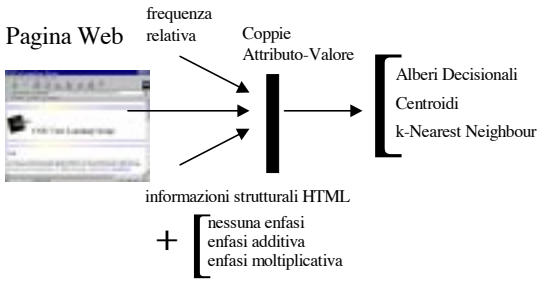

Figura 3: Processo di classificazione di una pagina Web.

Gli alberi di decisione sono generati utilizzando il sistema di apprendimento automatico OC1 [10], particolarmente adatto ai domini con forte prevalenza di attributi numerici. Infatti, il sistema OC1 è anche in grado di indurre alberi decisionali multivariati, benchè questa capacità non sia stata sfruttata nel nostro caso. WebClass genera un albero decisionale univariato per ogni classe di addestramento, considerando come esempi positivi tutte le pagine rappresentative della classe e come esempi negativi le rimanenti. In questo modo il risultato della classificazione di una nuova pagina potrà essere: 1) nessuna classificazione: la pagina è rigettata poichè non riconosciuta come appartenente a qualcuna delle classi definite dall'utente; 2) classificazione singola: la pagina è assegnata a una sola classe fra quelle predefinite; 3) classificazione multipla: la pagina è attribuita a più classi fra quelle predefinite.

Per quanto riguarda la seconda modalità di predizione, WebClass è in grado di classificare una pagina Web sulla base della similarità tra il vettore rappresentativo della pagina e un prototipo calcolato per ogni classe. Il prototipo di ciascuna classe è definito come il vettore *centroide* fra tutti i vettori relativi alle pagine Web di addestramento della classe. Ad esempio, in Tabella II sono riportati i quattro centroidi associati alle classi prese in considerazione per la sperimentazione descritta nel paragrafo successivo. In questo caso i centroidi proposti sono stati calcolati su 64 esempi di addestramento equidistribuiti tra le 4 classi.

Tabella II: Prototipi estratti da WebClass

	solar	*earth*	*Jazz*	*blue*	*Vehicle*	*fuel*	*motocycle*	*bike*
Astronomy	0.004	0.005	0	0	0	0	0	0
Jazz	0	0	0.026	0.003	0	0	0	0
Auto	0	0	0	0	0.005	0.004	0	0
Moto	0	0	0	0	0	0	0.01	0.009

Una nuova pagina viene attribuita alla classe il cui prototipo risulta essere meno distante da essa. La distanza è calcolata in funzione del coseno dell'angolo formato tra il vettore prototipo e il vettore rappresentativo della pagina.
Relativamente alla terza modalità, è stato implementato l'algoritmo di k-nearest neighbour (k-NN). Anche in questo caso la distanza è calcolata in funzione del coseno dell'angolo formato tra i vettori all'interno dello spazio vettoriale. Il k-NN attribuisce la classe di appartenenza della pagina Web da classificare analizzando la distribuzione delle classi di appartenenza dei k esempi di addestramento meno distanti. Come noto, uno dei limiti del k-NN è l'"intolleranza" agli attributi irrilevanti. Tuttavia, nel nostro caso il problema è meno evidente poiché la selezione degli attributi è già stata effettuata in precedenza, in modo da ridurre la presenza di attributi poco significativi.

5. RISULTATI SPERIMENTALI

Al fine di verificare l'efficacia degli algoritmi in costruzione è stata organizzata una sperimentazione consistente nell'addestrare il sistema al riconoscimento di quattro classi di pagine Web: *Astronomy*, *Jazz*, *Auto* e *Moto*. Le classi *Auto* e *Moto* sono concettualmente vicine fra loro e distanti da *Astronomy* e *Jazz*. In ogni caso, gli argomenti legati alle classi possono avere dei punti di contatto a livello "testuale"; ad esempio, non è difficile trovare delle pagine sul jazz con frasi del tipo "la *star* del gruppo", che potrebbero far pensare a una qualche relazione con l'astronomia.
Per definire il set di pagine di addestramento da fornire al sistema è stata considerata l'ontologia di classificazione resa disponibile dal search engine Yahoo. Le 192 pagine selezionate soddisfano i seguenti requisiti: 1) sono state individuate casualmente seguendo i link proposti dal motore di ricerca; 2) sono ricche di parti testuali in lingua inglese; 3) sono equidistribuite tra le quattro classi.
E' stata adottata la tecnica del cross-validation con tre insiemi di validazione (three-fold) in modo da poter stimare le statistiche di interesse. Per ogni prova 2/3 delle pagine sono state prese come insieme di addestramento lasciando il rimante terzo come insieme di validazione. L'insieme degli attributi utilizzati in ogni prova è stato determinato a partire dal sottoinsieme dei documenti di addestramento selezionati esclusivamente per quella prova, in modo tale da poter anche testare la bontà del metodo di selezione degli attributi. Si è deciso di estrarre 25 parole diverse da ogni dizionario di classe, ottenendo quindi un vettore di cento coppie

attributo-valore per ogni documento.
I metodi di classificazione considerati sono quattro: centroidi (*Ctr*), 1-nearest neighbor (*NN*), k-NN con k pari a sette, e alberi di decisione (OC1). Le statistiche stimate sono i tassi *Precision* e *Recall*[3] usati tradizionalmente in *information retrieval*, e mediati sulle tre prove per ogni classificatore. I risultati riportati in Tabella III fanno riferimento a trentasei prove complessive, ottenute dalla combinazione dei quattro classificatori considerati, delle tre prove per ogni classificatore e dei tre metodi di trattamento dell'informazione sulla struttura HTML: nessuna (0), enfasi additiva (+) ed enfasi moltiplicativa(*).

Tabella III: Statistiche *Precision* e *Recall* relative alla sperimentazione.

			CLASSI			
			Astronomy	**Auto**	**Jazz**	**Moto**
%PRECISION	0	**Ctr**	97,9	94,2	100	97,9
		kNN	96,1	91	100	98
		NN	96,1	88	100	93,5
		Oc1	97,7	92,9	100	100
	+	**Ctr**	97,9	92,4	100	95,8
		kNN	96,1	89,4	100	100
		NN	94,4	86,8	100	93,7
		Oc1	97,6	93,1	100	100
	*	**Ctr**	97,9	92,4	100	92,1
		kNN	96,1	89,1	100	95,8
		NN	59,3	85,7	92,6	92,1
		Oc1	95,5	89,2	100	100
%RECALL	0	**Ctr**	97,9	97,9	100	93,7
		kNN	97,9	97,9	100	87,5
		NN	97,9	93,7	100	83,7
		Oc1	87,5	83,3	100	93,7
	+	**Ctr**	97,9	95,8	100	97,7
		kNN	97,9	100	97,9	85,5
		NN	97,9	91,6	97,9	85,4
		Oc1	85,4	83,3	100	91,6
	*	**Ctr**	95,8	93,7	100	91,7
		kNN	97,9	95,8	97,9	87,5
		NN	95,8	79,1	95,8	83,3
		Oc1	85,4	87,5	100	91,6

Dall'analisi dei risultati sperimentali si possono trarre almeno tre conclusioni:

[3]Data una query Q relativa ai documenti di classe C contenuti in un database DB i valori di Precision e Recall sono definiti come segue:
Precision = #documenti di classe C ritrovati dalla query Q / #documenti ritrovati dalla query Q
Recall = #documenti di classe C ritrovati dalla query Q / #documenti di classe C nel database DB

1. il metodo di estrazione degli attributi si è rivelato efficace, in quasi tutte le prove sono stati ottenuti valori di *Precision* e *Recall* superiori al 90%. Si osservi che una estrazione casuale delle pagine in risposta avrebbe portato a un valore di *Precision* intorno al 25% e un valore di *Recall* variabile dallo 0 al 100% in funzione della numero di pagine selezionate per la risposta.
2. Tutti i metodi di classificazione sperimentati si sono dimostrati complessivamente efficaci nel problema di Web Mining considerato con una prevalenza del metodo basato sui prototipi di classe.
3. I meccanismi adottati per considerare l'informazione relativa alla struttura delle pagine HTML non sembra portare alcun vantaggio.

Un ulteriore test di WebClass è stato effettuato richiedendo ad un operatore umano di rintracciare su Internet un certo numero di pagine che a suo giudizio rappresentassero le classi prese da noi in considerazione. L'utente, utilizzando dei motori di ricerca tradizionali come Altavista ed Excite, e il metasearcher ProFusion, ha rintracciato 283 pagine. Queste sono state classificate utilizzando quei classificatori che hanno mostrato i migliori risultati in una delle prove del precedente esperimento. In particolare, sono stati testati i quattro classificatori senza tener conto della struttura HTML. I risultati ottenuti sono riportati in Tabella IV e confermano le percentuali osservate nell'esperimento precedente.

Tabella IV: Statistiche *Precision* e *Recall* relative alle 283 pagine di test.

	Ctr			**OC1**			**7-NN**			**NN**		
	Pr	**Re**	**FS[4]**	**Pr**	**Re**	**FS[4]**	**Pr**	**Re**	**FS[4]**	**Pr**	**Re**	**FS[4]**
Astronomy	98,5	97,1	97,8	100	100	100	97,1	97,1	97,1	97,2	98,6	97,9
Auto	97,3	100	98,6	94,6	95,9	95,2	100	98,6	99,3	98,6	97,3	97,9
Jazz	100	97	98,4	100	92,5	96,1	97,1	100	98,5	100	100	100
Moto	98,6	100	99,3	96	100	97,1	100	98,6	99,3	98,6	98,6	98,6

6. CONCLUSIONI

WebClass è un sistema di Web Mining in grado di classificare automaticamente pagine Web. Esso è implementato interamente in Java ad eccezione del sistema OC1 realizzato in C, ed è integrato nell'architettura WBI (Web Browser Intelligence). WebClass è stato testato ottenendo risultati interessanti nel riconoscimento di pagine Web estratte casualmente utilizzando l'ontologia di classificazione fornita dal search engine Yahoo. Particolarmente incoraggianti, relativamente alla sperimentazione effettuata, sono apparsi i risultati relativi alla tecnica di selezione degli attributi basata sulla combinazione di tre statistiche:

[4] F-Score=2*Precision*Recall/(Precision+Recall)

Term Frequency, *Page Frequency* e *Category Frequency*.
Per il futuro si prevede di avviare una sperimentazione più estesa, che coinvolga un maggior numero di classi possibilmente vicine concettualmente e/o organizzate secondo legami gerarchici, ed eventualmente faccia uso anche di test-set standard utilizzati nell'ambito della classificazione automatica di testi. Inoltre si indagheranno le problematiche legate al mutamento degli interessi dell'utente (aggiunta/rimozione di classi).

7. RINGRAZIAMENTI

Gli autori ringraziano la Prof.ssa Floriana Esposito per i preziosi suggerimenti e Giulio De Luise per il contributo dato nello sviluppo di WebClass e nella sperimentazione con gli alberi di decisione.

8. BIBLIOGRAFIA

1. R. Armstrong, D., Freitag, T., Joachims, and T. Mitchell: *WebWatcher: A learning Apprentice for the World Wide Web*. Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, 1995.
2. R. Barrett, P.P., Maglio, and D.C. Kellom: *How to personalize the Web*. IBM Almaden Research Center. USA. (www.networking.ibm.com/wbi/wbisoft.htm)
3. T.M. Cover and P.E. Hart: *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory 13, 1967, 21-27.
4. O. Etzioni: *The World-Wide Web: Quagmire or Gold Mine?* Communications of the ACM 39,1 (january 1996), 65-68.
5. K. Lang: *Learning to Filter Netnews*. Proceedings of the 12th International Conference on Machine Learning, 331-339, 1995.
6. H. Lieberman: *Letizia: An Agent That Assist Web Browsing*. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 924-929, 1995.
7. D. Malerba, G., De Luise, G., Convertino, L., Di Pace, P., Leo, and A., Maffione: *WebClass: a Web Mining tool*. Atti del Workshop Congiunto dei Gruppi di Lavoro dell'Associazione Italiana per l'Intelligenza Artificiale (AI*IA) su Apprendimento Automatico e Linguaggio Naturale, Torino, 1997.
8. M. Pazzani, J., Muramatsu and D., Bilsus*: Syskill & Webert: Identifying interesting web site*. AAAI Spring Symposium on Machine Learning in Information Access, Stanford, March 1996 (anche in *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI 96*, 54-61, 1996).
9. C.Y. Quek: *Classification of World Wide Web Documents*. Technical Report, Carnegie Mellon University, Pittsburgh, PA, 1996.
10. K.M. Sreerama, S, Kasif, S., Salzberg, and R., Beigel*: OC1: Randomized induction of oblique decision trees*. Proceedings of the Eleventh National Conference on Artificial Intelligence, 322-327.
11. Y. Wilks: *Information Extraction as a core language technology*. Information Extraction. SCIE-97, Springer Verlag, 1997, 1-9.