

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Ranking dei motori di ricerca . . . . .	3
1.1.1	Metodi di ranking endogeno . . . . .	4
1.1.2	Metodi di ranking esogeno . . . . .	5
1.2	Web spam . . . . .	6
1.2.1	Tecniche di boost . . . . .	7
1.2.2	Term Spamming . . . . .	8
1.2.3	Link Spamming . . . . .	9
1.2.4	Tecniche di hiding . . . . .	11
1.2.5	Click Spamming . . . . .	12
<b>2</b>	<b>Tecniche di spam detection</b>	<b>14</b>
2.1	Tecniche basate sul contenuto . . . . .	14
2.1.1	Prime feature per identificare lo spam . . . . .	14
2.1.2	Utilizzo di un classificatore per combinare le feature . . . . .	22

# Elenco delle figure

1.1	Tassonomia delle tecniche boost . . . . .	7
1.2	Tipi di pagine nel web per uno spammer . . . . .	10
1.3	Esempio di una spamfarm . . . . .	11
1.4	Tecniche di hiding . . . . .	12
2.1	Occorrenze dello spam classificate per dominio all'interno del dataset descritto in [10] . . . . .	16
2.2	Occorrenze dello spam classificate per lingua all'interno del dataset descritto in [10] . . . . .	16
2.3	Prevalenza di spam sulla base del numero di parole per pagina . . .	17
2.4	Prevalenza di spam sulla base del numero di parole all'interno dei titoli delle pagine . . . . .	18
2.5	Prevalenza di spam sulla base della frazione di parole di una pagina che sono tra le 200 più frequenti parole nel corpus . . . . .	19
2.6	Prevalenza di spam sulla base della lunghezza media delle parole per pagina . . . . .	20
2.7	Prevalenza di spam sulla base della quantità di testo delle ancore delle pagine . . . . .	21
2.8	Prevalenza di spam sulla base della frazione di contenuto visibile . .	22
2.9	Prevalenza di spam sulla base del rapporto di compressione . . . . .	23
2.10	Esempio di classificatore . . . . .	24

# Capitolo 1

## Introduzione

Questa tesi ha come obbiettivo lo studio e l'analisi delle tecniche di spam detection attualmente esistenti ed in particolare delle tecniche online. Nella prima parte le tecniche verranno classificate sulla base dei segnali che utilizzano. Successivamente verranno eseguiti dei test per valutare alcuni algoritmi di spam detection offline eseguiti durante la fase di crawling. Ed infine verranno presentati e discussi i risultati ottenuti. Al momento in cui si scrive non ci sono, o meglio sono poche, le tecniche online di spam detection, ovvero tecniche che rilevano lo spam durante la fase di crawling. Infatti quasi tutti i metodi tentano di fare il crawling dell'intera porzione di web di interesse e successivamente classificare le pagine in classi (dove di norma le classi sono due: *spam* oppure *non spam*).

Il fenomeno del web spam è sempre più presente all'interno del web: questo è dovuto al fatto che gli utenti tendono ad esaminare solo i primi risultati calcolati dai motori di ricerca e quindi se un sito fa parte degli  $n$  primi risultati può avere un ritorno economico legato alla quantità di traffico che viene generata per quel sito. Uno studio del 2005 descritto in [5] stima che la perdita finanziaria mondiale causata dallo spam è di circa 50 miliardi di dollari e nel 2009 (come descritto in [6]) è salita a 130 miliardi di dollari. Per questo motivo, recentemente tutte le più grandi compagnie di motori di ricerca hanno identificato il recupero di informazioni non pertinenti come una delle priorità da risolvere. Le conseguenze del web spam possono essere riassunte come segue[13]:

- la qualità delle ricerche è compromessa penalizzando i siti web legittimi;
- un utente potrebbe perdere la fiducia sulla qualità di un motore di ricerca e perciò passare con facilità all'utilizzo di un altro;
- inoltre i siti spam possono essere usati come mezzo per malware, pubblicazione di contenuto per adulti e attacchi di tipo “fishing”. Una prova tangibile si può vedere in [1], dove gli autori hanno eseguito l'algoritmo di *PageRank* su 100 milioni di pagine e hanno notato che 11 sui primi 20 risultati erano composti da siti con contenuto per adulti.

Queste considerazioni evidenziano che quando si progetta un motore di ricerca bisogna tenere conto delle pagine che potrebbero portare al mal funzionamento del motore stesso. Il lavoro prodotto sarà utilizzato per essere integrato all'interno di un web crawler distribuito ad alte prestazioni per il futuro sviluppo di un modulo di spam detection. L'esigenza di tale modulo è sorta a seguito dello sviluppo, presso il Dipartimento, di un crawler chiamato *BUBiNG*, altamente configurabile ma privo al momento di qualunque forma di rilevazione di siti e contenuti malevoli. Il problema è estremamente interessante sia dal punto di vista teorico che da quello pratico: infatti, sebbene siano numerose le tecniche descritte in letteratura per la determinazione di spam (usando come segnali sia il contenuto che la struttura dei link), è sorprendentemente scarso l'insieme di tali tecniche che possono essere usate on-line, cioè durante il crawl. Il problema diventa ancora più complesso se si aggiungono considerazioni legate ai vincoli di spazio di memoria disponibile e tempo di calcolo. Infatti in letteratura il processo di spam detection viene eseguito subito dopo la fase di crawling. Ovvero il processo è composto dai seguenti passi:

- crawling dell'intero web;
- fase di spam detection;
- indicizzazione.

Questo modello è utile perché molte delle tecniche utilizzate fanno delle analisi sul grafo che è il risultato della fine del processo di crawling. Da queste considerazioni

noi proviamo a fare delle analisi per determinare se il processo di spam detection può essere fatto durante la fase di crawling ovvero al momento in cui il crawler esegue il “fetch” di una pagina per determinare “on the fly” se la pagina è buona o ha un contenuto malevolo.

## 1.1 Ranking dei motori di ricerca

Prima di spiegare i vari metodi con cui si possono creare pagine web spam e successivamente quelli utili ad identificarlo, è necessario capire come i motori di ricerca sono capaci di valutare la rilevanza di una pagina web per una determinata query.

In linea di massima un sistema di reperimento di informazioni ovvero un motore di ricerca è dato da una collezione documentale  $D$  (un insieme di documenti) di dimensione  $N$ , da un insieme  $Q$  di interrogazioni, e da funzione di ranking ( $r : Q \times D \rightarrow R$ ) che assegna a ogni coppia formata da un’interrogazione e un documento un numero reale. L’idea è che a fronte di un’interrogazione a ogni documento viene assegnato un punteggio reale: i documenti con punteggio nullo non sono considerati rilevanti, mentre quelli a punteggio non nullo sono tanto più rilevanti quanto più il punteggio è alto. In particolare i metodi di ranking si dividono in *endogeni* ed *esogeni*. I primi metodi fanno uso del contenuto del documento per valutarne la rilevanza mentre i secondi fanno uso di una struttura esterna ad esempio il grafo composto dai collegamenti ipertestuali tra le pagine web, questo non implica che i metodi esogeni non possono fare uso del contenuto della pagina (per esempio il testo delle ancore). I criteri si dividono ulteriormente in statici (o indipendenti dall’interrogazione) e dinamici (o dipendenti dall’interrogazione). Nel primo caso, il punteggio assegnato a ciascun documento è fisso e indipendente da un’interrogazione  $q$  mentre nel secondo il punteggio assegnato a ciascun documento è dipendente da un’interrogazione  $q$ .

Tra i metodi endogeni sono di maggiore importanza *tf-idf* e *BM25* mentre tra quelli esogeni i più diffusi in letteratura sono *PageRank* e *HITS*.

### 1.1.1 Metodi di ranking endogeno

Come già detto in precedenza i metodi di ranking endogeno utilizzano il contenuto di una pagina per assegnarle un punteggio. Possono essere anch'essi statici o dinamici (cioè dipendere o meno da un'interrogazione). L'algoritmo usato dai motori di ricerca per fare il rank delle pagine web basandosi sui campi di testo usa varie forme del *tf-idf*. Il *tf-idf* è un metodo di ranking endogeno dinamico che utilizza il contenuto di una pagina per assegnarle un punteggio. Il *tf-idf* è una misura composta da due misure più semplici: la *Term Frequency* e la *Inverse Document Frequency*. Il primo metodo assegna a un documento  $d$  il punteggio dato dalla somma dei conteggi dei termini  $t$  dell'interrogazione che compaiono nel documento stesso. In questo modo documenti in cui i termini dell'interrogazione compaiono più frequentemente avranno un punteggio più elevato. Utilizzare solo questo metodo non conviene in quanto è facilmente manipolabile. Inoltre non tiene conto del fatto che alcuni termini occorrono più frequentemente non perché rilevanti, ma perché altamente frequenti all'interno di *ogni* documento (ad esempio le congiunzioni). Il secondo metodo è definito come l'inverso del numero di documenti nella collezione che contengono il termine  $t$  [8]. Più precisamente:

$$idf_t = \log \frac{N}{df_t} \quad (1.1)$$

La combinazione del *tf* ed dell'*idf* produce una misura composta che permette di normalizzare il peso dei termini. Il *tf-idf* di un documento  $d$  rispetto a una query  $q$  è calcolato su tutti i termini  $t$  in comune come:

$$tf-idf(d, q) = \sum_{t \in d \text{ and } t \in q} tf(t, d) \cdot idf(t) \quad (1.2)$$

Con il *tf-idf* gli spammer possono avere due obiettivi: o creare pagine rilevanti per un gran numero di query o creare pagine molto rilevanti per una specifica query. Il primo obiettivo può essere ottenuto includendo un gran numero di termini distinti in un documento; il secondo, attraverso la ripetizione di determinati termini nel documento.

Un altro metodo di ranking endogeno dinamico è *BM25* [12] che è uno schema di pesatura basato sul *modello probabilistico* ed attualmente è il sistema di pesatura più usato.

### 1.1.2 Metodi di ranking esogeno

Uno dei metodi esogeni statici è *PageRank* descritto in [11]. *PageRank* usa le informazioni portate dai link in entrata (*inlink*) per determinare un punteggio globale di importanza di una pagina. Esso assume che esista un legame tra il numero di *inlink* di una pagina  $p$  e la popolarità della pagina  $p$ . Il concetto fondamentale dietro *PageRank* è che una pagina è importante se molte altre pagine importanti puntano ad essa. Questo concetto è mutualmente rinforzante ovvero l'importanza di una certa pagina influenza ed è influenzata dall'importanza delle altre pagine [4]. In dettaglio *PageRank* è basato sulla passeggiata naturale del grafo del web  $G$ . Più precisamente, la passeggiata viene perturbata nel seguente modo: fissato un parametro  $\alpha$  tra 0 e 1, a ogni passo con probabilità  $\alpha$  si segue un arco uscente, e con probabilità  $1 - \alpha$  si sceglie un qualunque altro nodo del grafo utilizzando una qualche distribuzione  $v$ , detta vettore di preferenza (per esempio, uniforme). Assumendo che non esistano pozzi, la matrice di transizione della catena è quindi rappresentata dalla combinazione lineare:

$$\alpha G + (1 - \alpha)1v^T \quad (1.3)$$

dove  $G$  è la matrice della passeggiata naturale su  $G$ . Il fattore  $\alpha$  è detto fattore di attenuazione di norma è impostato a un valore di 0,85.

Un altro metodo esogeno usato per il ranking delle pagine è *HITS* (*Hyperlink Induced Topic Distillation*) introdotto in [7]. Differentemente da *PageRank* esso assegna due punteggi di importanza a ogni pagina: uno di *hubbiness* e uno di *autorevolezza*. L'intuizione dietro a *HITS* è che invece di un singolo punteggio di importanza esista un concetto di pagina *autorevole*, cioè pagina con contenuto pertinente e interessante, e di *hub*, cioè pagina contenente numerosi collegamenti a pagine autorevoli. I due concetti si rinforzano *mutuamente*: una pagina autorevole è puntata da molte pagine centrali, e una buona pagina centrale punta a molte pagine autorevoli.

Questo approccio considera che nel web ci sono due tipi di pagine: quelle che contengono dei contenuti per un determinato argomento (*authoritative*) e quelle che contengono tanti link a delle pagine *authoritative* che sono chiamate pagine *hub*. Le pagine *hub* sono utili per scoprire le pagine *authoritative* [9].

L'algoritmo lavora su un sottografo del web ottenuto a partire da un'interrogazione. La selezione del sottografo può essere fatta in vari modi, un modo è quello di prendere un certo insieme di risultati ottenuto da un motore di base e generare un sottografo sulla base di una query e delle pagine che puntano a quelle ottenute dalla query. Per questo sottoinsieme di pagine otteniamo una matrice di adiacenza  $A$ . I punteggi di *hub* e *authority* per tutte le pagine del sottoinsieme possono essere formalizzate dalla seguente coppia di equazioni:

$$\begin{cases} \vec{a}_{t+1} = A^T \vec{h}_t \\ \vec{h}_{t+1} = A \vec{a}_{t+1} \end{cases} \quad (1.4)$$

Può essere dimostrato che la soluzione ottenuta applicando iterativamente il sistema 1.4 converge rispettivamente al principale autovettore di  $AA^T$  e  $A^T A$  [9][13].

## 1.2 Web spam

Con il termine web spamming si fa riferimento a tutti i metodi che tentano di manipolare gli algoritmi di ranking dei motori di ricerca per aumentare il valore di alcune pagine rispetto ad altre [4]. Dato il numero esorbitante di pagine che vengono create e pubblicate sul web, gli utenti competono per far comparire le proprie pagine tra le prime dei risultati di una query. Il fenomeno dello spamming o spamindexing ricade sulla qualità delle ricerche causando diversi problemi: indicizzazione di pagine che non sono utili, aumento del costo delle operazioni di query, malware e reindirizzamento verso contenuto per adulti; inoltre questo spinge gli utenti ad utilizzare altri motori di ricerca [13].

L'obiettivo dei motori di ricerca è di ottenere ottimi risultati per identificare tutte le pagine web che sono rilevanti per una specifica query e presentarle secondo l'importanza che esse hanno. Di norma la rilevanza viene misurata attraverso la similarità testuale tra la query e le pagine mentre l'importanza è definita come la



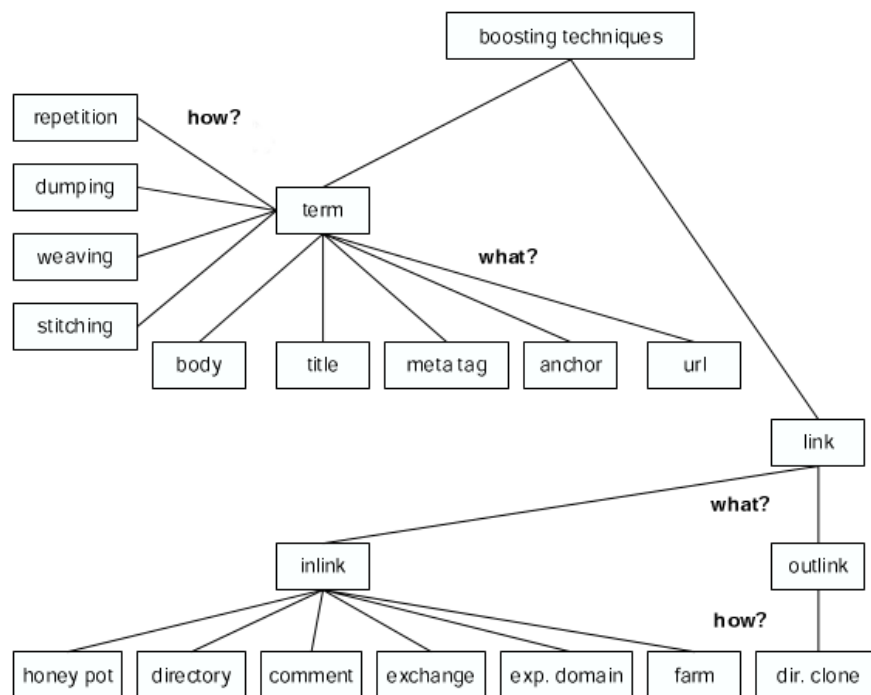


Figura 1.1: Tassonomia delle tecniche boost

popolarità globale della pagina e a volte è inferita dalla struttura dei link [4]. Ci sono due categorie di tecniche associate al web spam [4]:

- **tecniche boost** che cercano di far avere più importanza o rilevanza a delle pagine
- **tecniche hiding** che sono metodi per nascondere le tecniche di boost all'utente dal browser, anche se alcuni autori incorporano queste tecniche fra quelle di boost.

### 1.2.1 Tecniche di boost

Le tecniche di boosting si dividono in: *Term Spamming* e *Link Spamming*. Con l'avvento degli algoritmi di ranking basati sulla struttura del grafo il *Term Spamming* è stato trascurato. In figura 1.1 è specificata una possibile tassonomia delle tecniche boost [4].

### 1.2.2 Term Spamming

Nel valutare la rilevanza testuale i motori di ricerca considerano dove i termini di una query compaiono in una pagina. Il tipo di punto all'interno della pagina è chiamato *campo*. I più comuni campi di testo per una pagina  $p$  sono: il body della pagina, il titolo, i meta tag nell'header HTML e l'URL della pagina. Inoltre viene considerato anche come *campo*, il testo delle ancore (il tag  $a$ ) associate all'URL che puntano alla pagina  $p$  dato che descrive molto bene il contenuto della pagina. I campi di testo di  $p$  sono utilizzati per determinare la rilevanza di  $p$  rispetto ad una query (alcune volte i campi vengono pesati sulla base della loro importanza) e perciò chi fa *term spamming* utilizza tecniche di pesatura dei contenuti dei campi di testo in modo tale da aumentare l'efficacia dello spam [4]. Le tecniche di spamming possono essere raggruppate in base ai *campi* di testo dove viene fatto spamming. In base a questo distinguiamo [4]:

- *Body Spam*. In questo caso lo spam è nel corpo del documento. Questo è lo spam più diffuso.
- *Title Spam*. Molti motori di ricerca danno molta importanza ai termini che compaiono nel titolo. Quindi ha senso includere termini di spam all'interno del titolo della pagina.
- *Meta Tag Spam*. I tag che compaiono nell'header sono molto frequentemente soggetti a spam. Per questo i motori di ricerca danno poca importanza a questi campi o non li considerano. Di seguito viene mostrato un esempio di questo tipo di spam.

```
<meta name="keyword" content="buy, cheap, cameras, lens,
    accessories, nikon, canon">
```

- *Anchor Text Spam*. I motori di ricerca assegnano un peso maggiore al testo nelle ancore perché pensano che esse contengano un riassunto del contenuto della pagina. Perciò del testo di spam è incluso nel testo delle ancore dei collegamenti HTML di una pagina. In questo caso lo spamming non viene

fatto sulla pagina cui si vuole far avere un rank più alto ma sulle pagine che puntano ad essa.

```
<a href="target.html">free, great deals, cheap,  
    inexpensive, cheap, free</a>
```

- *URL Spam.* Alcuni motori di ricerca dividono l'URL delle pagine in un insieme di termini che sono usati per determinare la rilevanza di una pagina. Per sfruttare questo metodo di ranking, gli spammer creano lunghi URL che includono una grande sequenza di termini spam, un esempio può essere: *buy-canon-rebel-20d-lens-case.camerasx.com*.

Queste tecniche possono essere utilizzate insieme o separatamente. Un altro modo per raggruppare queste tecniche si basa sul tipo di termini che vengono utilizzati nei campi di testo [4], possiamo avere:

- Ripetizione di uno o più specifici termini.
- Inclusione di molti termini generici per creare pagine rilevanti per molte query.
- Intreccio di vari termini all'interno della pagina.
- Creazione di frasi di senso compiuto per l'elaborazione di contenuti generati velocemente attraverso la concatenazione di frasi da fonti diverse.

### 1.2.3 Link Spamming

Il *link spamming* è un tipo di spam che fa uso della struttura dei link tra le pagine web per favorire il rank di una pagina target  $t$ . In [4] si afferma che per uno spammer ci sono tre tipi di pagine nel Web: inaccessibili, accessibili(blog) e proprietarie (fig. 1.2). Le inaccessibili sono quelle che uno spammer non può modificare. Le accessibili sono pagine gestite da altri ma che possono essere modificate lievemente dallo spammer attraverso l'immissione di un post in un forum o in un blog o portali di questo genere. Le proprietarie sono pagine su cui gli spammer hanno il pieno controllo. Il gruppo di pagine proprietarie è chiamato *spam farm*.

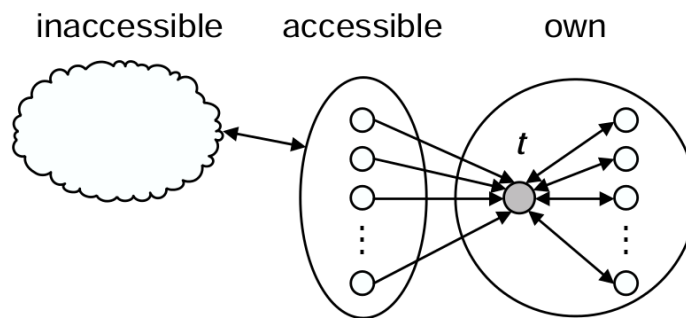


Figura 1.2: Tipi di pagine nel web per uno spammer

Molti motori di ricerca utilizzano due algoritmi per aumentare l'importanza basandosi sulle informazioni dei link: PageRank e HITS; sulla base di questi due tipi di algoritmi vengono definite due categorie principali di *link spamming*: *outgoing link spam* e *incoming link spam*. L'*outgoing link* è uno dei metodi più facili da implementare in quanto basta aggiungere dei link nella propria pagina, ad altre pagine che sono considerate buone, sperando di poter aumentare il punteggio di *hub*. Per la ricerca di link da includere nella pagina per cui si vuole incrementare il punteggio di *hub* si possono utilizzare delle directory che contengono liste di siti come DMOZ o Yahoo!. Queste directory organizzano i contenuti web in contenuti e in liste di siti relativi. Per quanto riguarda *incoming link*, ci sono diverse strategie che si possono adottare in modo tale da avere un numero elevato di link in entrata [4]:

- *Honeypot*: ovvero si creano un insieme di pagine che hanno un contenuto interessante (un esempio può essere una documentazione Linux) ma che hanno link nascosti alla pagina o alle pagine per cui si deve aumentare il valore di rilevanza.
- *Infiltrarsi in una directory web*: molte directory web permettono ai webmasters di postare link ai loro siti che hanno lo stesso contenuto.
- *Postare link nei blog, forum e wiki*: includere URL a pagine di spam come parte di un commento.

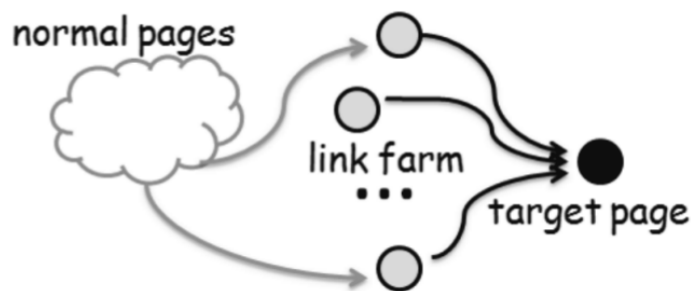


Figura 1.3: Esempio di una spamfarm

- *Scambio di link*: scambiare link con altre pagine di spam. Questa è una pratica comune tra chi fa spam ed esistono blog completamente finalizzati all'incontro di spammer per lo scambio dei link.
- *Comprare domini scaduti*: quando un dominio scade ci sono delle pagine che puntano ancora ad esso. Una tecnica è comprare questi domini e manipolare le pagine in modo tale da fare aumentare il rank di una pagina target.
- *Creare una spam farm*: con l'abbassamento dei costi si possono costruire delle spam farm che hanno come obbiettivo di aumentare la rilevanza di una pagina spam detta *target page*, un esempio è mostrato in fig. 1.3. Molte volte si utilizzano tecniche come *honeypot*. In questo caso il valore di page rank aggregato delle pagine è propagato alla pagina target. Una delle forme più aggressive di honeypot è l'*hijacking* [13], dove gli spammer prima attaccano un sito con una buona reputabilità e poi usano questo come parte della loro link farm.

#### 1.2.4 Tecniche di hiding

Le tecniche di hiding si possono classificare in: *content hiding*, *cloaking*, *redirection* (fig. 1.4) [4]. Nel *Content hiding* i termini o link di spam possono essere nascosti quando il browser visualizza una pagina. Una tecnica è quella di utilizzare lo stesso colore per i termini e lo sfondo. Mentre per i link basta non inserire il testo all'interno delle ancore che indirizzano a una pagina. Un'altra tecnica è quella di utilizzare degli script per nascondere il contenuto. Il *Cloaking* sfrutta il fatto che è facile

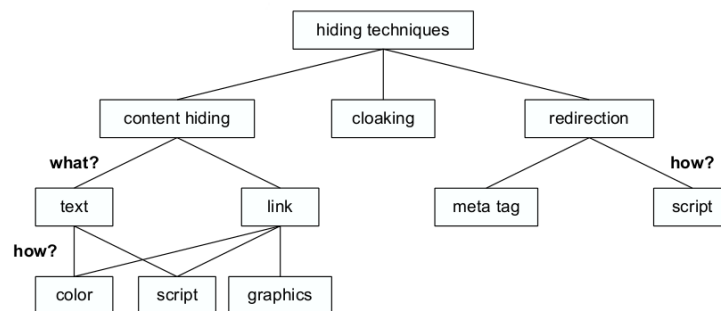


Figura 1.4: Tecniche di hiding

identificare quando la richiesta di una pagina è fatta da un crawler o da un browser: questa tecnica dato un URL, il server spam restituisce un documento HTML diverso a seconda che la richiesta sia fatta da un crawler o da un browser. Quindi vengono distribuiti due contenuti diversi in base al fatto che la richiesta al server spam sia fatta da un crawler o da un browser. La rilevazione di un crawler può essere effettuata in due modi: o si mantiene in memoria una lista di indirizzi di crawler oppure si usa l'header della richiesta HTTP andando a vedere il campo user-agent: se questo è diverso dai più comuni browser allora può essere un crawler. Nell'esempio sotto, lo user-agent della richiesta HTML indica l'uso del web browser Chrome.

```

Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/32.0.1700.102 Safari/537.36
  
```

La *Redirection* è un'altra tecnica che reindirizza il browser ad un altro URL appena la pagina è caricata. Un esempio di redirection server-side è mostrato di seguito.

```
header("Location: http://www.example.com/");
```

### 1.2.5 Click Spamming

Un ultimo metodo per fare web spam è il *Click Spamming* [13]. I motori di ricerca utilizzano dati sul flusso dei click per regolare le funzioni di ranking, quindi gli spammer generano click fraudolenti per manipolare il comportamento di queste fun-

zioni in modo tale da fare avere un rank migliore ai loro siti. Il metodo prevede che vengano fatte delle query e si clicchi sulla pagina di cui si vuole aumentare il rank. Tale metodo viene eseguito in modo automatico attraverso script che girano su diverse macchine per non fare sospettare il motore di ricerca delle numerose richieste provenienti da un'unica macchina [13].

## Capitolo 2

# Tecniche di spam detection

Il capitolo illustra alcune tecniche di spam detection presenti in letteratura. Le tecniche verranno suddivise sulla base del tipo di segnali che vengono utilizzati, come: il contenuto, il grafo ottenuto dalla fase di crawling ed altri segnali (es. header delle richieste HTTP). Nella prima parte del capitolo verranno illustrate le tecniche basate sul contenuto, nella seconda parte le tecniche che fanno uso di grafi ed infine le tecniche che fanno uso di altri tipi di segnali.

### 2.1 Tecniche basate sul contenuto

#### 2.1.1 Prime feature per identificare lo spam

Uno dei primi studi sul web spam è descritto in [2], in questo articolo vengono eseguite delle analisi per confrontare alcune proprietà delle pagine spam e con contenuti in modo tale da estrarre dei valori tramite cui si può stimare se una pagina sia spam. Bisogna precisare che questi valori sono il risultato di studi empirici effettuati su alcuni dataset. Le proprietà vengono classificate come segue:

- Proprietà degli URL. Come descritto in precedenza il *link spam* è una forma di spam dove gli spammer cercano di aumentare il rank derivato dagli algoritmi basati sulla struttura dei link. Quindi uno spammer cerca di creare automaticamente tante pagine spam di bassa qualità che puntano a una pagina target  $p$ . Alcune analisi sulle proprietà dei link mostrano che gli URL di un HOST sono



buone feature per identificare lo spam. In particolare il nome di un HOST con molti caratteri, punti, barre e numeri è un buon indicatore di spam. Perciò un modo semplice per classificare le pagine sarebbe quello di usare un valore di soglia tale per cui superata tale soglia la pagina venga classificata come spam.

- Host name resolution. Alcuni motori di ricerca (es. Google) data una query  $q$ , calcolano un rank più alto a un URL  $u$  se i termini che compongono il nome dell'host di  $u$  combaciano con i termini della query. Gli spammer per sfruttare questo meccanismo popolano gli URL delle pagine spam con termini contenuti in query molto frequente che sono rilevanti per un certo settore.
- Proprietà del contenuto: le pagine generate automaticamente hanno tutte lo stesso template, ad esempio ci sono numerosi siti di spam che dinamicamente generano pagine che hanno uno stesso numero di parole. Una tecnica per determinare lo spam è quella di clusterizzare le pagine in base alla somiglianza dei template. Visto che le pagine di spam sono molto simili tra loro, identificando molte pagine con la stessa struttura è probabile che siano di spam.

Oltre a queste proprietà base che servono per identificare lo spam in [10], un lavoro del 2006, vengono descritti una serie di metodi per l'individuazione dello spam. Ogni metodo è altamente parallelizzabile, può essere eseguito in un tempo proporzionale alla dimensione della pagina e identifica lo spam di ogni pagina scaricata. Inoltre questi metodi possono essere combinati con tecniche di machine learning per creare un algoritmo di rilevazione di spam più efficiente. Questo lavoro è un proseguo del lavoro precedentemente descritto [2]. In primis sono stati identificati quali domini e pagine (classificate sulla base della lingua) contenessero più spam. Il risultato ha evidenziato che i domini “.biz, .us, e .com” sono quelli con maggiore contenuto di spam mentre per quanto riguarda le pagine contenenti più spam sono le francesi, tedesche e inglesi. I risultati sono rappresentati nei due grafici in figura 2.1 e in figura 2.2. Questi risultati si basano sul dataset messo a disposizione degli autori che è stato ricavato utilizzando MSN Search crawler nell'agosto del 2004, per maggiori informazioni [10].

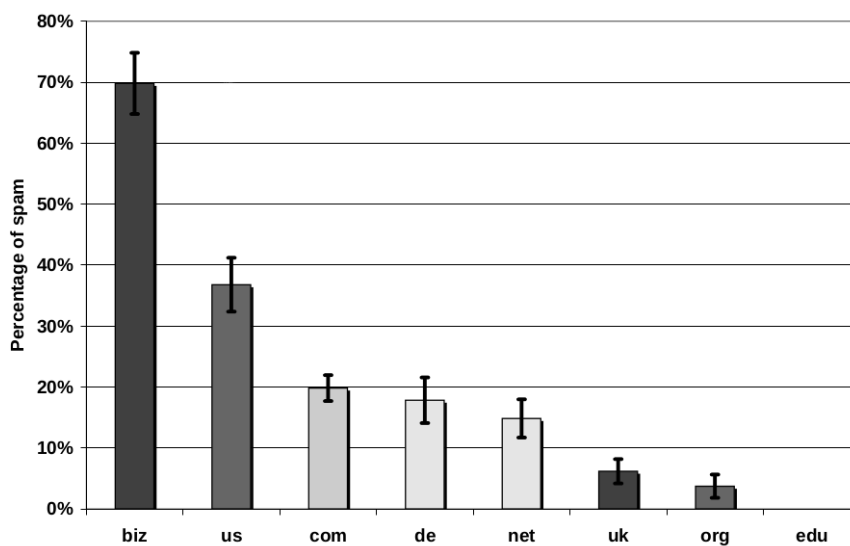


Figura 2.1: Occorrenze dello spam classificate per dominio all'interno del dataset descritto in [10]

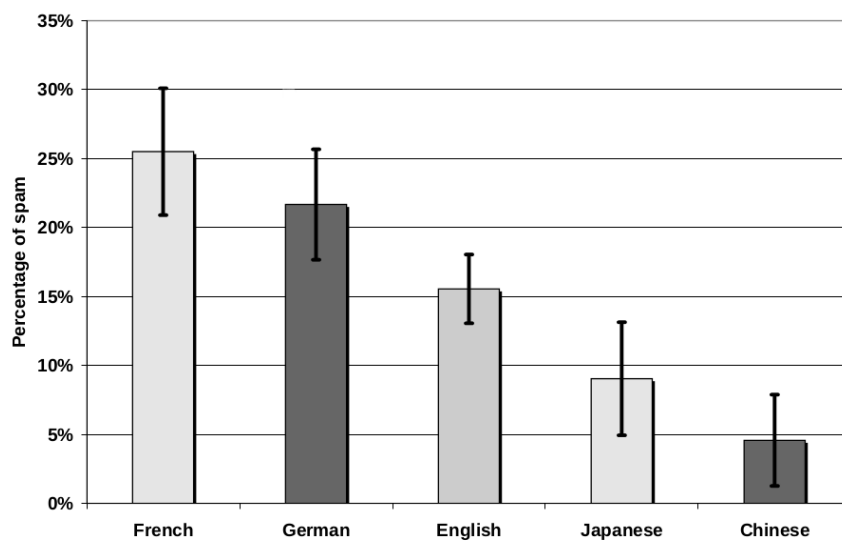


Figura 2.2: Occorrenze dello spam classificate per lingua all'interno del dataset descritto in [10]

Una pratica molto comune nel costruire pagine spam è la cosiddetta “Keyword stuffing”. Durante questo processo il contenuto della pagina aumenta con un numero di parole popolari che sono irrilevanti con il resto della pagina. In molti casi per poter aumentare le probabilità di essere messa in cima al rank di molte query, il contenuto di una pagina spam viene aumentato con tante parole estrane all’argomento della pagina. In figura 2.3 viene plottato la distribuzione delle parole per ogni pagina del data set. Oltre alla distribuzione delle parole viene raffigurata la percentuale di pagine per ogni range di parole che sono considerate spam. Dal grafico viene notato che la prevalenza di spam è più alta nelle pagine con pmolte parole. Perciò c’è una correlazione tra prevalenza di spam e numero di parole. Il conteggio delle parole da solo non è una buona euristica visto che porta un alto tasso falsi positivi.

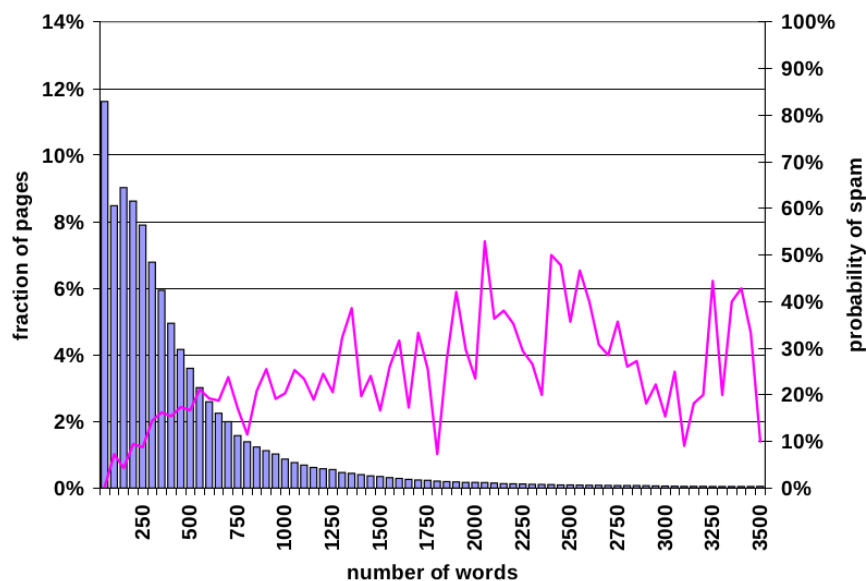


Figura 2.3: Prevalenza di spam sulla base del numero di parole per pagina

La tecnica “Keyword stuffing” viene utilizzata anche per la costruzione dei titoli delle pagine spam dal momento che alcuni motori di ricerca assegnano un peso maggiore ai termini della query presenti all’interno del titolo della pagina. Il grafico in figura 2.4 rappresenta la distribuzione del numero di parole all’interno dei titoli delle pagine. Come per gli altri grafici viene plottata la rispettiva percentuale di

pagine spam. Dal grafico si vede che un'eccesso di parole all'interno del titolo è un indicatore (come per il contenuto della pagina) che una pagina è spam. Le

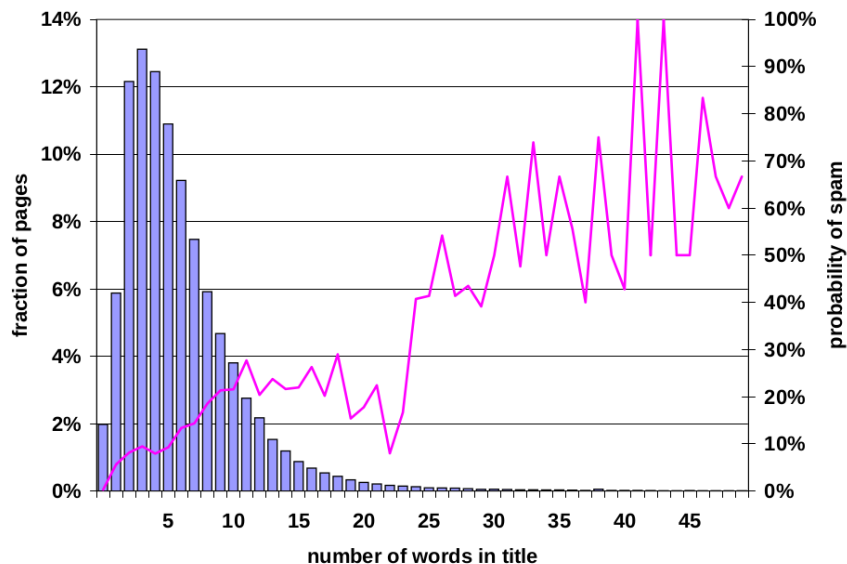


Figura 2.4: Prevalenza di spam sulla base del numero di parole all'interno dei titoli delle pagine

parole che vengono utilizzate nel processo di “keyword stuffing” vengono selezionate casualmente o da un ristretto gruppo di query comuni. Per tentare di esaminare il comportamento con cui vengono selezionate e costruite le frasi vengono esaminate le pagine per determinare se sono costituite da un eccesso di parole molto comuni. Prima vengono identificate le  $n$  parole più comuni all'interno del corpus poi viene calcolata, per ogni pagina, la frazione delle parole comuni contenute in ogni pagina. Questo processo viene ripetuto per ogni scelta di  $n$ . In figura 2.5 viene mostrato il grafico per  $n=200$ . Il grafico è basato sulla frazione di parole di una pagina che sono contenute nell'insieme delle 200 parole più comuni nella porzione delle pagine inglesi del data set utilizzato in [10]. Il grafico ha una caratteristica gaussiana e suggerisce che la maggior parte delle pagine di spam sono generate tessendo parole da un dizionario con una scelta casuale.

Un metodo per identificare pagine spam che sono generate automaticamente è descritto in [3]. Queste pagine spam sono costruite dinamicamente legando insieme

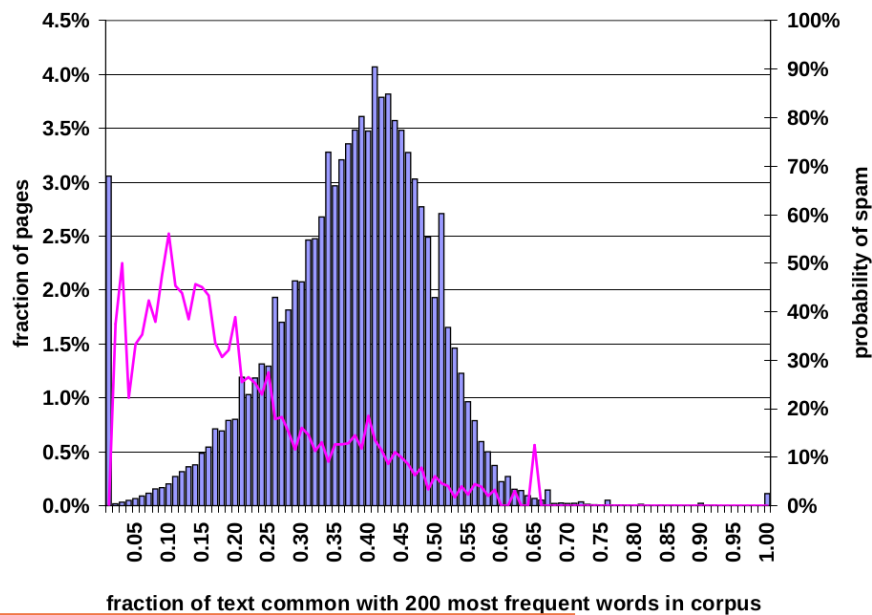


Figura 2.5: Prevalenza di spam sulla base della frazione di parole di una pagina che sono tra le 200 più frequenti parole nel corpus

frasi grammaticali ben formate. Le pagine create secondo questo metodo vengono generate al volo e cambiano completamente a ogni download. Queste pagine vengono create per essere indicizzate dalla maggior parte dei motori di ricerca. I link di queste pagine puntano ad altre pagine che sembrano essere su un altro host ma vengono risolte da uno stesso IP. Apparte l'uso di differenti host name ma che in realtà corrispondono ad unico host per non creare l'illusione di una struttura nepotica questo meccanismo viene usato anche per eludere le politiche di politeness di un web crawler volte al non sovraccarico di qualsiasi host se tali politiche sono basate sul nome dell'host. Infine generando pagine con frasi formate in modo corretto impediscono agli utenti di individuare l'inganno.

Un altro dato trovato dagli autori per determinare se una pagina è spam riguarda la lunghezza media delle parole delle pagine. Dal dataset preso in considerazione in [10] si nota che la distribuzione risultante della lunghezza media delle parole è simile a una gaussiana con moda e mediana corrispondenti a una media di 5.0. La maggior parte delle parole hanno una lunghezza media compresa tra 4.0 e 6.0. Come si nota dal grafico in figura 2.6 le parole con lunghezza media 10 sono certamente spam.

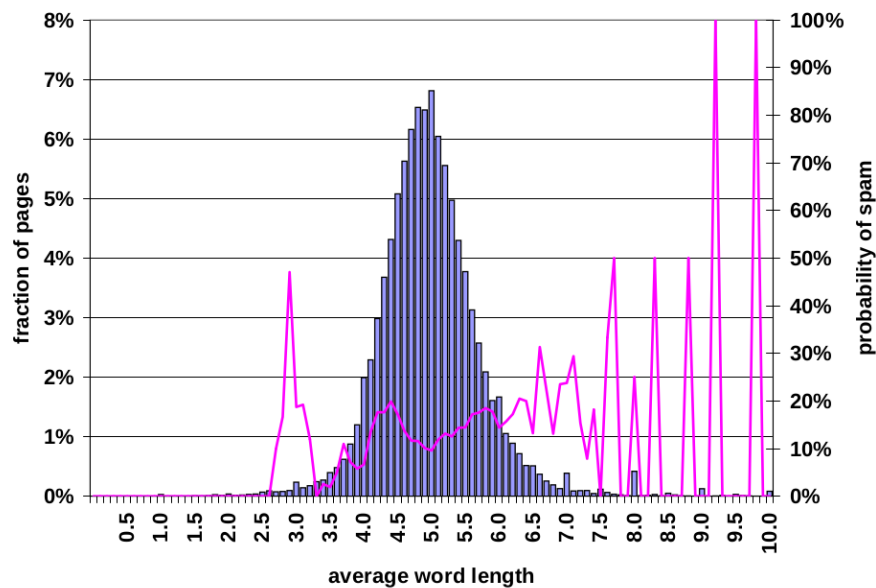


Figura 2.6: Prevalenza di spam sulla base della lunghezza media delle parole per pagina

Un'altra proprietà delle pagine web che consente di stimare se una pagina è spam oppure no è la quantità di testo che è contenuta all'interno delle ancore (il tag “ $\text{a}_i$ ” delle pagine web). Infatti una pratica comune dei motori di ricerca è considerare il testo delle ancore dei link in una pagina come annotazioni che descrivono il contenuto della pagina che viene puntata dal link. L'idea principale è che se la pagina  $a$  ha un link alla pagina  $b$  con testo dell'ancora, ad esempio, “computer” allora potremmo concludere che  $b$  parli di computer, anche se questa keyword non compare all'interno della pagina  $b$ . Alcuni motori di ricerca tengono conto di questo durante il ranking e potrebbero considerare la pagina  $b$  come risultato di una query contenente la keyword “computer”. Sfruttando questo meccanismo alcune pagine di spam vengono create solo per contenere del testo all'interno delle ancore per valorizzare il ranking di altre pagine. Queste pagine di norma sono solo cataloghi di link ad altre pagine. Per capire meglio il fenomeno è stato calcolato la frazione di tutte le parole del testo delle ancore all'interno di una pagina esclusi i markup rispetto al contenuto della pagina. In figura 2.7 viene visualizzato il grafico risultante. Si nota che un'alta frazione di testo delle ancore aumenta la probabilità che la pagina sia spam ma usare questa euristica da sola potrebbe portare un alto numero di falsi positivi.

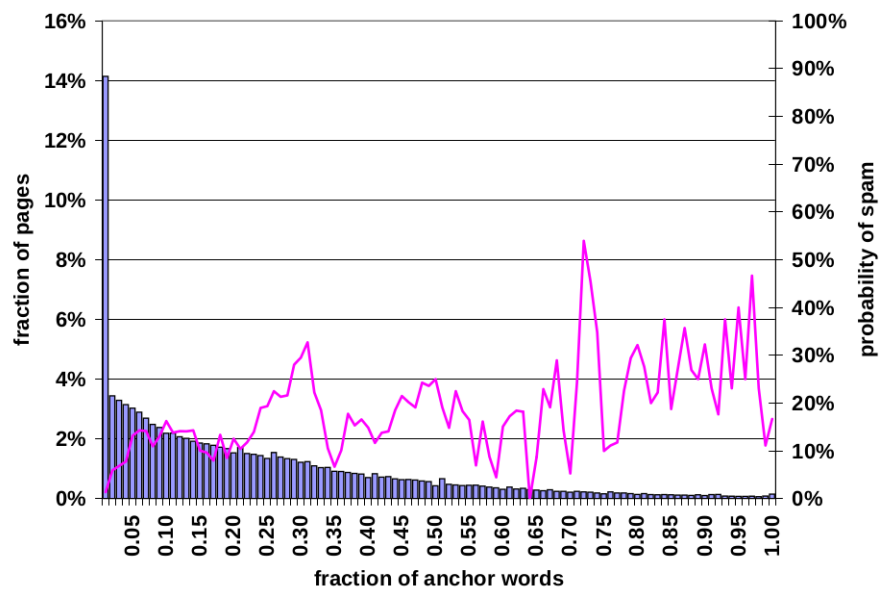


Figura 2.7: Prevalenza di spam sulla base della quantità di testo delle ancore delle pagine

Oltre a queste proprietà anche la quantità di contenuto visibile è utile per capire se una pagina è spam. Infatti calcolando la frazione di contenuto visibile all'interno di una pagina: definita come la lunghezza in termini di byte di tutte le parole non di markup diviso l'interna dimensione della pagina, si nota dal grafico in figura 2.8 della distribuzione delle frazioni di contenuto visibile che le pagine di spam hanno meno markup delle pagine normali. Questo fa intendere che molte pagine spam hanno il solo scopo di dover essere indicizzate dai motori di ricerca e non di essere fruite da un utente.

Come detto in precedenza i motori di ricerca possono dare un maggiore peso a pagine che contengono le keyword contenute nella query più volte all'interno della pagina ad esempio utilizzano come metodo di ranking il "term-frequency". Alcune pagine spam per trarre vantaggio da questo meccanismo replicano i contenuti molte volte per aumentare il rank. Per rilevare la ridondanza di contenuti (ottenuta dal processo di replica) e perciò stimare se la pagina sia spam, viene calcolato il rapporto di compressione ovvero la dimensione della pagina non compressa divisa per la dimensione della pagina compressa. In figura 2.9 viene visualizzato la distribuzione del rapporto di compressione e la likelihood che la pagina sia spam. Dal

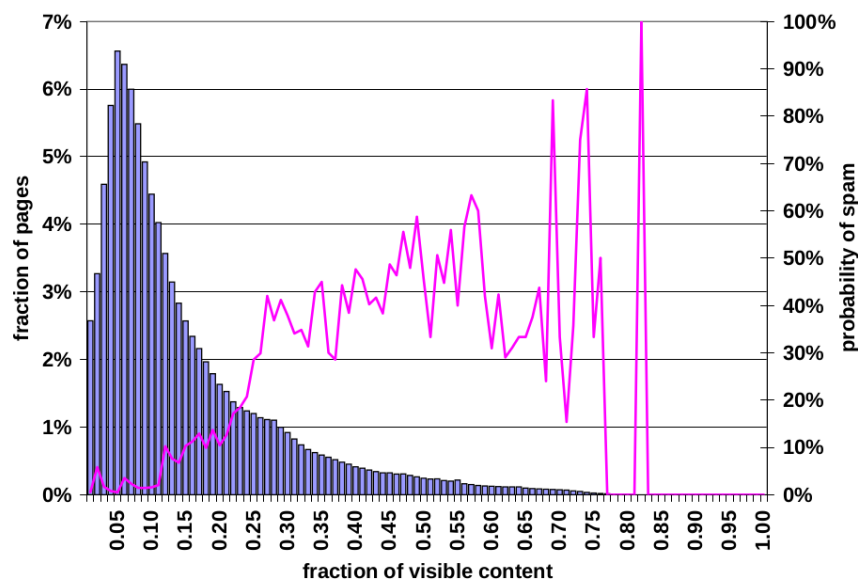


Figura 2.8: Prevalenza di spam sulla base della frazione di contenuto visibile

grafico si nota che quanto più il valore di compressione è elevato molto più probabilmente la pagina può essere considerata spam in particolare la probabilità che una pagina sia spam risiede sulla parte destra del grafico, il 70 per cento delle pagine con un rapporto di compressione maggiore di 4.0 sono giudicate spam. Questo è dovuto al fatto che una compressione su una pagina piena di contenuti ridondanti (spam) sarà più efficace che su una pagina caratterizzata da contenuti casuali (non spam). E perciò il rapporto tenderà a crescere quanto più la pagina sarà caratterizzata da contenuti ridondanti.

### 2.1.2 Utilizzo di un classificatore per combinare le feature

Sempre in [10] le euristiche vengono combinate considerando il problema di rilevamento dello spam come un problema di classificazione. In questo caso viene creato un modello di classificazione il quale data una pagina web userà le caratteristiche della pagina in modo tale di classificarla in una delle due classi: spam o non spam. Costruire un classificatore precede una fase di training durante la quale i parametri del classificatore sono determinati e una fase di testing durante la quale le performance del classificatore sono valutate. Per ogni pagina all'interno del dataset viene



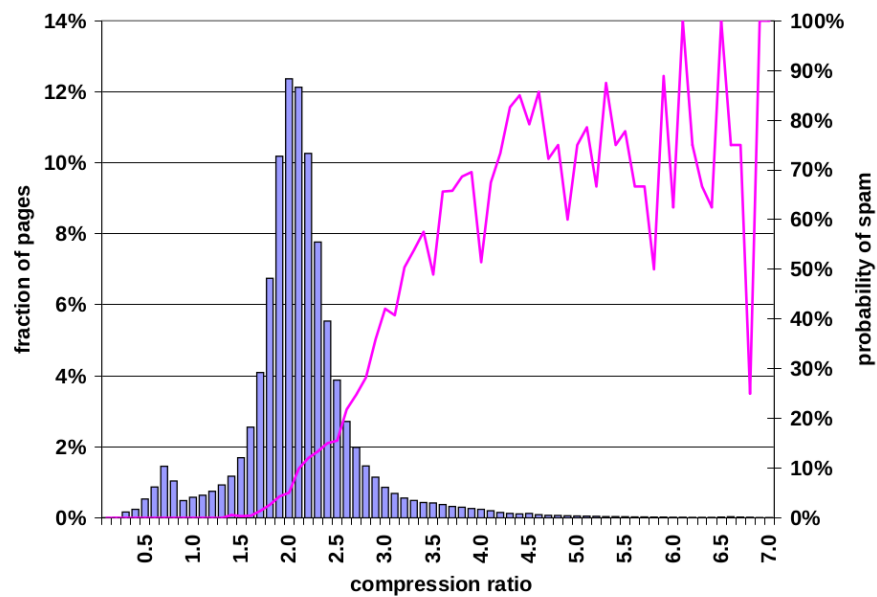
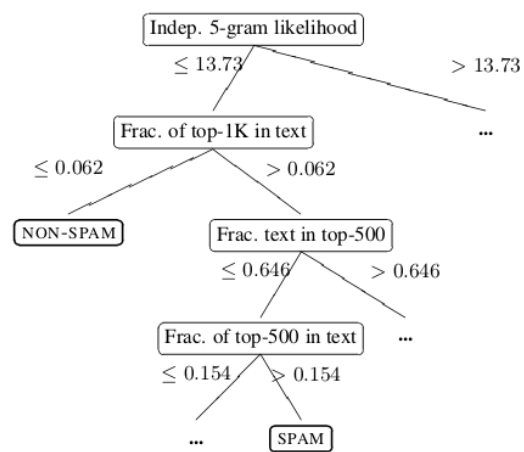


Figura 2.9: Prevalenza di spam sulla base del rapporto di compressione

calcolato il valore di ogni feature (le varie euristiche) e usiamo questi valori per istruire il classificatore. VUendo un classificatore di tipo “decision-tree”, l’algoritmo di classificazione funziona come segue: dato un insieme di dati da training e un insieme di feature l’algoritmo crea un diagramma di flusso come una struttura ad albero. Ogni nodo dell’albero corrisponde al test da valutare per una particolare feature mentre ogni arco è un valore di uscita del test ed infine le foglie corrispondono alle classi. Per applicare l’albero alle pagine viene controllato il valore della proprietà definita nel nodo radice dell’albero per quella pagina e confrontato con la soglia indicata dai lati uscenti poi si seguono i lati sulla base dei valori ottenuti finché non si arriva ad assegnare la classe per quella pagina. Un esempio del classificatore è rappresentato in figura 2.10 Per migliorare il classificatore si possono usare tecniche come bagging o boosting. Le tecniche creano un insieme di classificatori che vengono combinati.

*Figura 2.10: Esempio di classificatore*

# Bibliografia

- [1] Nadav Eiron, Kevin S. McCurley, and John A. Tomlin. Ranking the web frontier. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 309–318, New York, NY, USA, 2004. ACM.
- [2] Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases: Colocated with ACM SIGMOD/PODS 2004*, WebDB '04, pages 1–6, New York, NY, USA, 2004. ACM.
- [3] Dennis Fetterly, Mark Manasse, and Marc Najork. Detecting phrase-level duplication on the world wide web. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 170–177, New York, NY, USA, 2005. ACM.
- [4] Zoltan Gyongyi and Hector Garcia-Molina. Web spam taxonomy. Technical Report 2004-25, Stanford InfoLab, March 2004.
- [5] Nicholas R. Jennings. The global economic impact of spam. *Ferris Research*, 2005.
- [6] Nicholas R. Jennings. Cost of spam is flattening - our 2009 predictions. *Ferris Research*, 2009.
- [7] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

- [8] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. Pages 117–119.
- [9] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. Pages 474–476.
- [10] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on World Wide Web, WWW '06*, pages 83–92, New York, NY, USA, 2006. ACM.
- [11] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.
- [12] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.
- [13] Nikita Spirin and Jiawei Han. Survey on web spam detection: Principles and algorithms. *SIGKDD Explor. Newsl.*, 13(2):50–64, May 2012.