ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

Ο ΚΡΥΜΜΕΝΟΣ ΙΣΤΟΣ. ΜΕΛΕΤΗ ΚΑΙ ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ ΤΟΥ

ΦΟΙΤΗΤΗΣ:

Κουντουρζής Γεώργιος – Μιχαήλ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Παπαδόπουλος Δημήτρης

ΠΑΤΡΑ, 2013

Περίληψη

Στη πτυχιακή εργασία αυτή, μελετάται η έννοια του deep web, τα κύρια χαρακτηριστικά του γνωρίσματα και η διάκρισή του από το surface web. Μέσα από την ανάλυση που θα ακολουθήσει στα επόμενα κεφάλαια παρουσιάζεται ο ρόλος του αλλά και η χρησιμότητα του περιεχομένου του. Πέραν των βασικών εννοιών που θα αναλυθούν, θα γίνει αναφορά στις μεθόδους ανάκτησης των δεδομένων από το deep web.

1. ΕΙΣΑΓΩΓΗ

Ο κρυμμένος ιστός ή αλλιώς Deep Web (το οποίο είναι γνωστό και με τις ονομασίες Deepnet, DarkNet, Undernet) αναφέρεται στο μέρος εκείνο του World Wide Web, όπου το περιεχόμενό του δεν ανήκει στο Επιφανειακό Ιστό (Surface Web), και το οποίο δεικτοδοτείται από μία συνηθισμένη μηχανή αναζήτησης[1].

Ο ιδρυτής του BrightPlanet Mike Bergman[2] είναι αυτός που έδωσε αυτή τη φράση και την έννοια της (deep web). Αυτό το οποίο είχε πει είναι: "πως το να ψάχνει κανείς στο Internet σήμερα είναι σαν να σέρνει ένα δίχτυ στην επιφάνεια του ωκεανού: πολλά μπορεί να πιαστούν στο δίχτυ, αλλά υπάρχει ένας πλούτος πληροφοριών που βρίσκονται βαθιά και επομένως δεν μπορούν να πιαστούν".

Η παραπάνω φράση εννοεί ότι οι περισσότερες πληροφορίες στον παγκόσμιο ιστό, είναι θαμμένες/κρυμμένες μέσα σε ιστότοπους με ιστοσελίδες που παράγονται δυναμικά, και ως εκ τούτου οι διαδεδομένες μηχανές αναζήτησης δεν είναι σε θέση να τις εντοπίσουν. Άρα λοιπόν, οι παραδοσιακές μηχανές αναζήτησης (που όλοι γνωρίζουμε και χρησιμοποιούμε) δεν μπορούν να ανακτήσουν το περιεχόμενο του κρυμμένου ιστού. Οι σελίδες αυτές στην ουσία δεν υπάρχουν μέχρις ότου δημιουργηθούν δυναμικά, και να επιστραφούν ως το αποτέλεσμα μιας συγκεκριμένης αναζήτησης. Ο κρυμμένος ιστός όπως θα φανεί και στην συνέχεια της εργασίας είναι συγκριτικά πολύ μεγαλύτερο από τον επιφανειακό Ιστό.[3]

Με βάση τα παραπάνω, γίνεται αντιληπτό ότι όταν αναφερόμαστε στο Deep Web, αναφερόμαστε στην αόρατη (για τους περισσότερους) πλευρά του διαδικτύου. Στην πραγματικότητα, το ορατό μέρος του ιστού, που μπορεί κάποιος να έχει πρόσβαση μέσω των παραδοσιακών μηχανών αναζήτησης, είναι μόνο ένα μέρος του διαδικτύου. Ο κρυμμένος ιστός αφορά το μέρος εκείνο του διαδικτύου, όπου το περιεχόμενο του δεν είναι τόσο εύκολα προσβάσιμο.

[1] http://el.wikipedia.org/wiki/Deep_Web
[2] http://brightplanet.com/images/uploads/12550176481-deepwebwhitepaper.pdf

Ένα μέρος του κρυμμένου ιστού είναι προσβάσιμο μέσω του ανώνυμου δικτύου του Tor. Το Tor αρχικά χρηματοδοτήθηκε από το Εργαστήριο Ναυτικών Ερευνών των ΗΠΑ, όταν κυκλοφόρησε το 2002 και χρησιμοποιείται ευρέως σε όλο τον κόσμο για την προστασία της ανωνυμίας στην on-lene σύνδεση[3].

[3] http://www.apocalypsejohn.com/2012/12/dark-web.html

2. ΕΠΙΦΑΝΕΙΑΚΟΣ ΚΑΙ ΚΡΥΜΜΕΝΟΣ ΙΣΤΟΣ

Στο κεφάλαιο αυτό, γίνεται μια περιγραφική αναφορά στις δύο κύριες κατηγορίες που υπάρχουν στον Παγκόσμιο Ιστό, τον επιφανειακό ιστό (surface web) και τον κρυμμένο ιστό (deep web). Επίσης επισημαίνονται οι βασικές διαφορές των δύο κατηγοριών καθώς και τα κύρια χαρακτηριστικά τους.

2.1. **Επιφανειακός Ιστός** (Surface web)

Ο κρυμμένος ή επιφανειακός ιστός (surface web ή visible web ή indexable web) είναι εκείνο το τμήμα του Παγκόσμιου Ιστού που δύναται να ευρετηριοποιηθεί από τις παραδοσιακές μηχανές αναζήτησης. Στην αντίπερα όχθη, το κομμάτι εκείνο του παγκόσμιου ιστού που δεν είναι προσβάσιμο από τις παραδοσιακές μηχανές αναζήτησης, καλείται κρυμμένος ιστός (deep web ή invisible web).

Πρακτικά, οι μηχανές αναζήτησης κατασκευάζουν μια βάση δεδομένων του Ιστού χρησιμοποιώντας κάποια ειδικά προγράμματα (spiders ή web crawlers) τα οποία ξεκινούν τη διαδικασία ανίχνευσης από μια λίστα γνωστών σελίδων. Το spider κρατάει ένα αντίγραφο της κάθε σελίδας και το καταχωρεί στο ευρετήριο, αποθηκεύοντας χρήσιμες πληροφορίες οι οποίες θα επιτρέψουν να ανακαλεστεί μετά η σελίδα γρήγορα. Οι υπερσυνδέσεις σε νέες σελίδες, προστίθενται στη λίστα των σελίδων που θα πρέπει να ανιχνευτούν. Το αποτέλεσμα είναι όλες οι προσπελάσιμες σελίδες που έχουν σχέση με την αναζήτηση, να ταξινομηθούν στο ευρετήριο. Η συλλογή αυτών των αποτελεσμάτων, δηλαδή των προσπελάσιμων σελίδων, καθορίζει τον επιφανειακό ιστό.

2.2. **Μηχανές Αναζήτησης**

Το κύριο εργαλείο που χρησιμοποιείται για την αναζήτηση πληροφοριών στον Παγκόσμιο Ιστό είναι οι μηχανές αναζήτησης (web search engine). Όπως είναι γνωστό,

τα αποτελέσματα της αναζήτησης ταξινομούνται σε μία λίστα και παρουσιάζονται στις σελίδες αποτελεσμάτων της μηχανής αναζήτησης (SERPs - Search engine results page).

Στον ακόλουθο πίνακα παρατίθεται μια λίστα του χρονικού ιστορικού λειτουργίας των διαφόρων μηχανών αναζήτησης καθώς και της κατάστασης λειτουργίας τους σήμερα.

Year	Engine	Current status
1993	W3Catalog	Inactive
	Aliweb	Inactive
1994	WebCrawler	Active, Aggregator
	Go.com	Active, Yahoo Search
	Lycos	Active
1995	AltaVista	Inactive (URL redirected to Yahoo!)
	Daum	Active
	Magellan	Inactive
	Excite	Active
	SAPO	Active
	Yahoo!	Active, Launched as a directory
1996	Dogpile	Active, Aggregator
	Inktomi	Acquired by Yahoo!
	HotBot	Active (lycos.com)
	Ask Jeeves	Active (ask.com, Jeeves went away)
1997	Northern Light	Inactive
	Yandex	Active
1998	Google	Active
	MSN Search	Active as Bing
1999	AlltheWeb	Inactive (URL redirected to Yahoo!)
	GenieKnows	Active, rebranded Yellowee.com
	Naver	Active
	Teoma	Active
	Vivisimo	Inactive
2000	Baidu	Active
	Exalead	Acquired by Dassault Systèmes
2002	Inktomi	Acquired by Yahoo!
2003	Info.com	Active
2004	Yahoo! Search	Active, Launched own web search (see Yahoo! Directory, 1995)
	A9.com	Inactive
	Sogou	Active
2005	AOL Search	Active

	Ask.com	Active
	GoodSearch	Active
	SearchMe	Closed
2006	wikiseek	Inactive
	Quaero	Active
	Ask.com	Active
	Live Search	Active as Bing, Launched as rebranded MSN Search
	ChaCha	Active
	Guruji.com	Active
2007	wikiseek	Inactive
	Sproose	Inactive
	Wikia Search	Inactive
	Blackle.com	Active
2008	Powerset	Inactive (redirects to Bing)
	Picollator	Inactive
	Viewzi	Inactive
	Boogami	Inactive
	LeapFish	Inactive
	Forestle	Inactive (redirects to Ecosia)
	VADLO	Active
	DuckDuckGo	Active, Aggregator
2009	Bing	Active, Launched as rebranded Live Search
	Yebol	Active
	Mugurdy	Inactive due to a lack of funding
	Goby	Active
2010	Black Google Mobile	Active
	Blekko	Active
	Cuil	Inactive
	Yandex	Active, Launched global (English) search
	Yummly	Active
2011	Interred	Active
2012	Volunia	Active , only *Power User*

Εικόνα 1 - Αναπαράσταση ενεργών μηχανών αναζήτησης κατά τη περίοδο 1993 - 2012[4]

[4] Πηγή: http://en.wikipedia.org/wiki/Search_engine

Όπως απεικονίζεται και στο ακόλουθο σχεδιάγραμμα, η πιο δημοφιλής μηχανή αναζήτησης για το Μάιο του 2012, παραμένει με διαφορά η μηχανή αναζήτησης της Google.

Global:	
Google	87.6%
Bing	4.2%
Baidu	4.7%
Yahoo	2.4%
Ask.com	0.6%
Other	0.5%

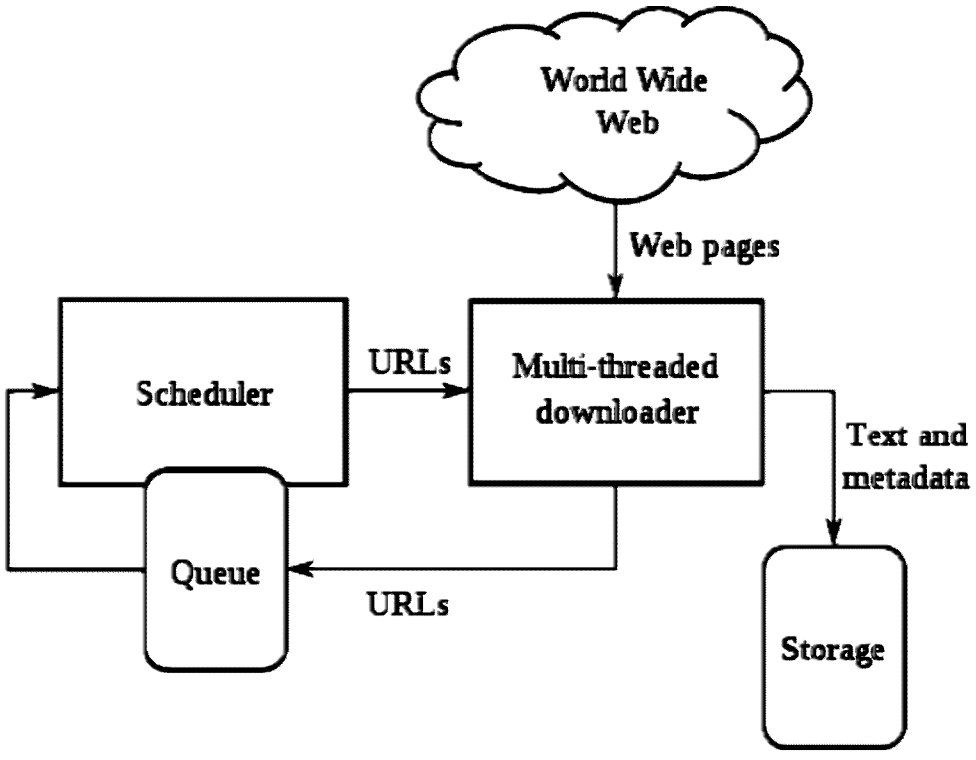
Εικόνα 2 - Ποσοστό χρήσης των μηχανών αναζήτησης[5]

Όπως αναφέρθηκε, οι μηχανές αναζήτησης χρησιμοποιούν ειδικά προγράμματα (spiders ή web crawlers) για την ανίχνευση των πληροφοριών στον Παγκόσμιο Ιστό. Ο web crawler αποτελείται από τέσσερα κύρια μέρη τα οποία αναλύονται ακολούθως:

[5] Πηγή: http://www.karmasnack.com/about/search-engine-market-share/

- **Scheduler**: Αποτελείται από δύο μονάδες. Η μία μονάδα αναλαμβάνει τον εντοπισμό διπλότυπων URLs και η άλλη μονάδα αναλαμβάνει την εισαγωγή και εξαγωγή των URLs στην ουρά (queue).
- **Downloader**: Ξεκινώντας από τα URLs εκκίνησης, ανακτά τις αντίστοιχες σελίδες, εξάγει τα URLs που περιέχονται στις σελίδες υπό τη μορφή ιστοσελίδων κι εν συνεχεία τα στέλνει στον Scheduler, ο οποίος με τη σειρά του τα προσθέτει στην ουρά (εφόσον δεν αποτελούν διπλο-εγγραφή). Αποτελείται από τις ακόλουθες τρεις μονάδες:
    - DNS Resolving
    - Ανάκτηση σελίδων μέσω του HTTP
    - Συντακτική ανάλυση HTML σελίδων με σκοπό την εξαγωγή υπερ-συνδέσμων και λοιπών στατιστικών δεδομένων.
- **Queue**: Δέχεται από τον Scheduler, τα URLs.
- **Storage**: Το μέρος στο οποίο αποθηκεύονται οι σελίδες από τον Downloader.

Στο ακόλουθο σχεδιάγραμμα εμφανίζεται η λειτουργία που εκτελείται.



2.3. Web Crawlers

Οι web crawlers (ή spiders όπως είναι ευρέως γνωστοί), είναι αυτοματοποιημένα προγράμματα που διαπερνούν το παγκόσμιο ιστό σύμφωνα με κάποια συγκεκριμένη τακτική. Η διαδικασία που επιτελεί ένας web crawler καλείται web crawling και είναι μία διαδικασία που χρησιμοποιείται κατά κύριο λόγο από τις υπηρεσίες δεικτοδότησης ώστε να «κατεβάσουν» τις σελίδες του διαδικτύου. Γενικά, ένας crawler ξεκινά από μία λίστα URLs που πρόκειται να επισκεφτεί. Συνεχίζει αναδρομικά βρίσκοντας τα links στις σελίδες που επισκέπτεται και τερματίζει αφού καλυφθούν κάποιες παράμετροι, π.χ. χρόνος εκτέλεσης, ποσότητα δεδομένων, κλπ.

Το πλήθος των web crawlers είναι αρκετά μεγάλο και αν εξαιρέσουμε τους εξειδικευμένους web crawlers (focused crawlers) παρατηρούμε πως οι περισσότεροι έχουν σαν σκοπό να συλλέξουν όλες τις HTML σελίδες από τις οποίες απαρτίζεται ένας δικτυακός τόπος μαζί με τα βοηθητικά αρχεία (pdf, video, css, javascript) και ουσιαστικά να δημιουργήσουν ένα offline-instance του δικτυακού τόπου τον οποίο προσπελαύνουν.

Οι crawlers που έχουν κατασκευαστεί για το διαδίκτυο αγγίζουν σε αριθμό τις μερικές χιλιάδες, καθώς η κατασκευή τους είναι σχεδόν τετριμμένη. Στη συνέχεια θα παρουσιάσουμε συγκεκριμένους crawlers που αξίζουν προσοχής για τα ιδιαίτερα χαρακτηριστικά που παρουσιάζουν.

2.3.1. **Δημοφιλείς** Crawlers

Web Crawler

Ο web crawler, πρόκειται για έναν από τους πρώτους crawlers που κατασκευάστηκαν από τον Pinkerton το 1994 [10]. Βασίστηκε στη βιβλιοθήκη WWW, προκειμένου να είναι σε θέση να κατεβάζει σελίδες από το διαδίκτυο ενώ χρησιμοποιούσε ένα δεύτερο πρόγραμμα προκειμένου να διαβάζει τα URLs τα οποία πρέπει να προσπελάσει. Ο αλγόριθμος προσπέλασης ήταν κατά πλάτος αναζήτηση του γραφήματος μίας ιστοσελίδας σε συνδυασμό με αποφυγή των σελίδων που έχει ήδη επισκεφθεί. Ένα αξιοσημείωτο στοιχείο ήταν η δυνατότητα να ακολουθεί συγκεκριμένα μόνο links σε ένα δικτυακό τόπο - και όχι όλα - βάση του ερωτήματος που έθετε ο χρήστης. Ήταν κάτι σαν ένας crawler πραγματικού χρόνου που φυσικά μπορούσε να ανταποκριθεί πλήρως λόγω του μικρού μεγέθους που είχε το διαδίκτυο. Ο Web Crawler από το 2001 είναι τμήμα της Infospace η οποία τον χρησιμοποιεί ως την βάση για την ομώνυμη μετα-μηχανή αναζήτησης[6].

Google Crawler

Ο Google Crawler ή αλλιώς Googlebot είναι ένας από τους πιο σημαντικούς Crawlers που κατασκευάστηκαν και διατηρούνται ακόμα και σήμερα, με σημαντικές βέβαια βελτιώσεις. Ο Google Crawler των Brin και Page [7], βασίζεται στις γλώσσες προγραμματισμού C++ και Python και παρουσιάζει εξαιρετικά μεγάλη πολυπλοκότητα. Επειδή η χρήση των σελίδων που κατέβαζε ο Crawler προοριζόταν για εκτενή αναζήτηση μέσα σε σειρές από κείμενα, ο συγκεκριμένος Crawler βασίστηκε στη διαδικασία indexing. Στο μηχανισμό υπάρχει ένας URL εξυπηρετητής που αποστέλλει λίστες με URLs προς τους Crawlers του συστήματος οι οποίοι λειτουργούν παράλληλα. Οι Crawlers εξάγουν από τις σελίδες το κείμενο αλλά και όσα URLs εντοπίζουν. Αυτά στέλνονται πίσω στον URL εξυπηρετητή για έλεγχο και σε περίπτωση που δεν τα έχει επισκεφθεί ποτέ ο Crawler προστίθενται στη λίστα του εξυπηρετητή. Συγκεκριμένες πληροφορίες υλοποίησης για τον Crawler που χρησιμοποιεί η μηχανή αναζήτησης

[6] http://www.webcrawler.com/

Google δεν είναι διαθέσιμες στο ευρύ κοινό μιας και ο αλγόριθμος του crawling που χρησιμοποιείται μεταβάλλεται διαρκώς προκειμένου:

α) να ανταποκρίνεται στις μεταβαλλόμενες απαιτήσεις του Web και

β) να αντιμετωπίζει τις επιθέσεις των spammers.

Mercator

Ο Mercator είναι ένας κατανεμημένος και τμηματοποιημένος Web Crawler γραμμένος εξ' ολοκλήρου σε γλώσσα προγραμματισμού Java. Η τμηματοποίηση του προκύπτει από τη χρήση δύο διαφορετικών πρωτοκόλλων [8].

- Protocol modules

  Τα τμήματα πρωτοκόλλων είναι υπεύθυνα για την ομαλή σύνδεση του μηχανισμού στις σελίδες και για την εξασφάλιση πως ο μηχανισμός θα είναι σε θέση να «κατεβάσει» τη σελίδα.

- Processing modules

  Από την άλλη μεριά τα τμήματα επεξεργασίας είναι αυτά που αφορούν την ανάλυση της σελίδας και την εξαγωγή του κειμένου και συνδέσμων από αυτή. Η απλή διαδικασία επεξεργασίας περιλαμβάνει ανάλυση της σελίδας και εξαγωγή των συνδέσμων που αυτή περιέχει ενώ σε μία πιο σύνθετη μορφή της περιλαμβάνει αλγορίθμους για την αποτελεσματική εξαγωγή του κειμένου.

- WebFoundain

  Πρόκειται για έναν κατανεμημένο τμηματικό Crawler παραπλήσιο του Mercator, με τη διαφορά ότι είναι γραμμένος σε C++ . Περιλαμβάνει έναν κεντρικό μηχανισμό και μία σειρά από "ant" (μερμήγκι) μηχανισμούς [9]. Πρόκειται δηλαδή για το ρυθμιστή της κατάστασης και τους εργάτες. Ο μηχανισμός αυτός

περιέχει στοιχεία που τον κάνουν πολύ φιλικό προς τις σελίδες που επισκέπτεται. Σκοπός του είναι η διατήρηση ενός offline instance του διαδικτύου. Αυτό έχει σαν αποτέλεσμα, μία από τις μετρικές τις οποίες προσμετρά ο συγκεκριμένος μηχανισμός να είναι το κατά πόσο οι σελίδες που διαθέτει ανταποκρίνονται στις πραγματικές σελίδες που βρίσκονται on-line στους δικτυακούς τόπους και όχι απλά σε μία παλαιότερη έκφανσή τους. Για να πετύχει μεγαλύτερο freshness όπως ονομάζεται η συγκεκριμένη μετρική, χρησιμοποιεί διαφορετική συχνότητα επίσκεψης στις σελίδες που έχει αποθηκευμένες στη βάση δεδομένων του.

· WebRACE

Πρόκειται για έναν Crawler ο οποίος είναι γραμμένος σε Java και αποτελεί ένα κομμάτι ενός γενικότερου συστήματος που ονομάζεται eRACE [10]. Το συγκεκριμένο σύστημα λαμβάνει εντολές από τους τελικούς χρήστες για να ξεκινήσει να κατεβάσει σελίδες και συμπεριφέρεται σαν proxy server. Το σύστημα μπορεί να εξυπηρετήσει και αιτήσεις για αλλαγές στοιχείων σε σελίδες: μόλις μία σελίδα αλλάξει, τότε ο crawler την ξανακατεβάζει και ειδοποιεί τον τελικό χρήστη που ενδιαφέρεται πως η σελίδα έχει αλλάξει και πως πλέον στον proxy είναι αποθηκευμένη μία νέα σελίδα. Το πιο σημαντικό στοιχείο του συγκεκριμένου crawler είναι η χαρακτηριστική διαφορά που παρουσιάζει συγκριτικά με όσους crawlers έχουμε δει. Στο συγκεκριμένο crawler δεν υπάρχει έ να feed URL από το οποίο θα ξεκινήσει να αναζητά σελίδες. Το URL feed είναι δυναμικό και διαμορφώνεται από τα ερωτήματα των χρηστών. Μετά τη χρήση του καταστρέφεται και ο μηχανισμός βρίσκεται σε αναμονή μέχρι να του δοθεί κάποιο νεότερο ερώτημα.

· Ubicrawler

Ο Ubicrawler [11] είναι ένας κατανεμημένος crawler γραμμένος σε Java και δε διαθέτει κεντρικοποιημένη διαδικασία. Είναι κατασκευασμένος από έναν αριθμό από όμοιους "agents" και μία συνάρτηση –ανάθεση που αναθέτει σε κάθε agent κάποια εργασία. Οι agents δεν επικοινωνούν μεταξύ τους άμεσα αλλά όλες οι

διαδικασίες διευθετούνται από την κεντρική συνάρτηση ανάθεσης. Καμία σελίδα δεν προσπελαύνεται διπλή φορά καθώς κάθε agent φροντίζει να ενημερώσει για τις σελίδες που έχει επισκεφθεί εκτός και αν κάποιος από τους agents καταστραφεί. Πρόκειται για έναν πολύ σταθερό crawler, σχεδιασμένο με τέτοιον τρόπο ώστε να πετυχαίνει μέγιστη κλιμάκωση και μικρή ευαισθησία σε σφάλματα.

Crawlers ανοιχτού κώδικα

Μία σειρά από crawlers ανοιχτού κώδικα διανέμονται ελεύθερα στο διαδίκτυο. Κυρίως είναι προϊόντα κάποιου ιδιώτη που κατασκευάζονται για να καλύψουν συγκεκριμένες ανάγκες που έχουν οι τελικοί χρήστες, ανάγκες που συχνά δεν καλύπτονται από τους εμπορικούς crawlers. Η χρήση τους έχει συνήθως ως εξής. Κάποιος χρήστης που δεν καλύπτεται από έναν εμπορικό crawler λαμβάνει τον κώδικα ενός open source συστήματος και το αλλάζει με σκοπό να το φέρει στα μέτρα του. Συνήθως οι open source crawlers δεν έχουν εξειδικευμένες λειτουργικότητες ωστόσο προσφέρονται στους τελικούς χρήστες οι οποίοι μπορούν να τους τροποποιήσουν ελεύθερα.

Μερικά παραδείγματα από crawlers ανοιχτού κώδικα ακολουθούν

- GNU Wget[7]
- Heritrix[8]
- ht://Dig[9]
- HTTrack[10]
- Larbin[11]

[7] http://www.gnu.org/software/wget/
[8] http://crawler.archive.org/
[9] http://www.htdig.org/
[10] http://www.httrack.com/

- Methabot[12]
- Nutch[13]
- WebSPHINX[14]
- WIRE – Web Information Retrieval Environment[15]

2.3.2. **Εστιασμένο** crawling

Ένας γενικού σκοπού Web Crawler συγκεντρώνει όσο περισσότερες σελίδες μπορεί από ένα δεδομένο σύνολο από URL's. Αντίθετα, ένας εστιασμένος ή αλλιώς focused crawler είναι σχεδιασμένος για να συγκεντρώνει μόνο έγγραφα ενός συγκεκριμένου θέματος ή ενδιαφέροντος και επομένως ελαττώνει την κίνηση του δικτύου και κατά συνέπεια και τον απαιτούμενο χρόνο περάτωσης. Ο σκοπός του focused crawler είναι να αναζητεί επιλεκτικά σελίδες που είναι σχετικές με ένα προκαθορισμένο σύνολο από θεματικές έννοιες. Οι έννοιες καθορίζονται όχι με βάση κάποιες λέξεις-κλειδιά, αλλά χρησιμοποιώντας έγγραφα-παραδείγματα. Αντί λοιπόν να συλλέγονται και να δεικτοδοτούνται όλα τα διαθέσιμα έγγραφα του ιστού ώστε να υπάρχει η δυνατότητα να μπορούμε να απαντήσουμε σε κάθε πιθανή τυχαία ερώτηση, ένας focused crawler αναλύει τα όρια του crawling εντοπίζοντας δεσμούς που είναι πολύ πιθανό να συσχετίζονται με το crawling που γίνεται, αποφεύγοντας παράλληλα άσχετες θεματικές περιοχές του ιστού. Αυτό έχει ως συνέπεια την σημαντική ελάττωση των απαιτούμενων πόρων σε υλικό και εύρος ζώνης και βοηθάει την διαδικασία του crawling ώστε να έχει πιο επικαιροποιημένο περιεχόμενο.

Ο focused crawler περιέχει τρία βασικά συστατικά:

[11] http://larbin.sourceforge.net/index-eng.html
[12] http://www.bithack.se/methabot/
[13] http://lucene.apache.org/nutch/
[14] http://www.cs.cmu.edu/~rcm/websphinx/
[15] http://www.cwr.cl/projects/WIRE/

i. έναν ταξινομητή (clasifier), ο οποίος λαμβάνει αποφάσεις συσχέτισης όσων αφορά στις σελίδες που διαπερνούνται ουτοσώστε να αποφασιστεί αν θα γίνει το λεγόμενο link expansion (επέκταση και ακολούθηση των δεσμών της σελίδας)

ii. έναν distiller, ο οποίος καθορίζει το μέτρο του κατά πόσον οι σελίδες που διαπερνιούνται παραμένουν εντός θεματικών εννοιών

iii. έναν crawler με δυναμικά αναπροσαρμοζόμενη προτεραιότητα που ελέγχεται από τον ταξινομητή και τον distiller.

Η πιο σημαντική εκτίμηση της ικανότητας του focused crawler δίνεται από το harvest ratio που αυτός έχει. Η μετρική αυτή μας δίνει το ρυθμό κατά τον οποίο οι σχετικές σελίδες συλλέγονται και οι μη σχετικές απορρίπτονται από τη διαδικασία του crawling. Το harvest ratio θα πρέπει να είναι μεγάλο αλλιώς ο focused crawler περνάει πολύ χρόνο απλά απορρίπτοντας μη σχετικές σελίδες και πιθανά να είναι καλύτερα σε αυτή την περίπτωση να χρησιμοποιηθεί ένας κλασικός crawler.

2.3.3. **Αλγόριθμοι για εστιασμένο** crawling

Οι Focused crawlers βασίζονται σε δύο ειδών αλγορίθμους για την διατήρηση του περιεχομένου για τη διατήρηση του περιεχομένου τους εντός των προκαθορισμένων εννοιών [12]. Οι αλγόριθμοι που αναλύουν τον ιστό χρησιμοποιούνται για να κρίνουν την συσχέτιση και την ποιότητα των ιστοσελίδων, ενώ οι αλγόριθμοι αναζήτησης καθορίζουν την βέλτιστη σειρά με την οποία τα URLs πρέπει να επισκεφθούν.

Αλγόριθμοι ανάλυσης του ιστού

Γενικά, οι αλγόριθμοι αυτού του τύπου μπορούν να κατηγοριοποιηθούν σε δύο κατηγορίες: σε αυτούς που βασίζονται στην ανάλυση του περιεχομένου και σε αυτούς που βασίζονται στην ανάλυση των δεσμών των σελίδων.

Οι αλγόριθμοι που κάνουν ανάλυση περιεχομένου, εφαρμόζουν τεχνικές δεικτοδότησης για την ανάλυση αλλά και την εξαγωγή λέξεων – κλειδιών έτσι ώστε να καθορίσουν αν το περιεχόμενο μιας σελίδας είναι σχετικό με το πεδίο του crawler. Η κατηγορία αυτή ενσωματώνει την γνώση για το πεδίο στην ανάλυση των σελίδων βελτιώνοντας τα αποτελέσματα. Για παράδειγμα, ελέγχονται οι λέξεις μιας ιστοσελίδες σε σύγκριση με μια προκαθορισμένη λίστα λέξεων του πεδίου. Συχνά ανατίθεται επίσης μεγαλύτερο βάρος σε λέξεις και φράσεις που ανήκουν στον τίτλο ή σε κεφαλίδες της σελίδας (πληροφορία που εντοπίζεται με βάση τα HTML tags). Η URL διεύθυνση επίσης περιέχει σημαντική πληροφορία για τη σελίδα που έχει να κάνει με τον προορισμό της ή για το πεδίο γνώσης της. Παράλληλα η τοποθέτηση των δεσμών έχει ιδιαίτερη σημασία όσον αφορά το πεδίο γνώσης μιας ιστοσελίδας [13]. Πιο συγκεκριμένα, ο συγγραφέας της σελίδας A, που τοποθετεί ένα link προς τη σελίδα B, θεωρεί ότι η σελίδα B σχετίζεται με την A.

Ο όρος «δεσμοί εισόδου» αναφέρεται στα links που «δείχνουν» προς τη σελίδα. Συνήθως, όσο μεγαλύτερος είναι αυτός ο αριθμός, τόσο πιο υψηλά βαθμολογείται μία σελίδα. Η λογική πίσω απ' αυτό είναι παρόμοια με αυτή της ανάλυσης των αναφορών των συγγραφέων: ένα κείμενο που γίνεται αναφορά συχνά από άλλους συγγραφείς, θεωρείται καλύτερο από κάποιο που δεν έχει καμία αναφορά από άλλους. Η υπόθεση είναι ότι αν δύο σελίδες έχουν ένα link μεταξύ τους είναι πολύ πιθανό να έχουν το ίδιο θεματικό πεδίο. Συγκεκριμένα, στο [14], υπολογίζεται ότι η πιθανότητα οι σελίδες που έχουν link μεταξύ τους να έχουν παρόμοιο περιεχόμενο κειμένου είναι υψηλή αν επιλέγονται τυχαία σελίδες από τον ιστό.

Το anchor text είναι η λέξη ή η φράση που έχει ένας δεσμός ως το κείμενο που εμφανίζεται στον browser. Το κείμενο αυτό μπορεί να δώσει μία καλή πηγή πληροφορίας σχετικά με την σελίδα που δείχνει ο δεσμός, ένα θέμα που έχει θιγεί από πολλές μελέτες, π.χ. [15]. Τέλος είναι σχετικά λογικό να δίνεται ένα επιπλέον βάρος σε κάποιον δεσμό που «έρχεται» από κάποια διάσημη πηγή (π.χ. yahoo.corn). Οι γνωστότεροι αλγόριθμοι που βασίζονται στην ανάλυση των δεσμών είναι ο PageRank [7] και ο HITS [16].

Αλγόριθμοι αναζήτησης του ιστού

Οι αλγόριθμοι αυτού του τύπου χρησιμοποιούνται για να καθοριστεί η βέλτιστη σειρά με την οποία τα URLs πρέπει να επισκεφθούν. Παρότι πολλοί διαφορετικοί αλγόριθμοι αναζήτησης έχουν προταθεί, οι δύο πιο συνηθισμένοι είναι ο κατά πλάτος και ο κατά βάθος. Ο αλγόριθμος αναζήτησης κατά πλάτος είναι η απλούστερη στρατηγική για το crawling μιας και δε ν χρησιμοποιεί ευρετικά για την απόφαση σχετικά με το επόμενο link που πρόκειται να ακολουθηθεί. Όλα τα URLs στο τρέχον επίπεδο θα επισκεφθούν με τη σειρά που ανακαλύπτονται. Παρότι αυτή η στρατηγική δεν διαφοροποιεί ιστοσελίδες διαφορετικής ποιότητας ή διαφορετικού πεδίου, εντούτοις είναι αρκετά καλή για το χτίσιμο συλλογής μιας γενικού σκοπού μηχανής αναζήτησης. Πρόσφατα όμως [17], δείχθηχε ότι παρότι απλή, η στρατηγική αυτή μπορεί να χρησιμοποιηθεί και για συλλογές συγκεκριμένου πεδίου. Η λογική είναι ότι αν τα URLs εκκίνησης είναι σχετικά με το πεδίο που αναζητούμε, είναι πιθανό οι σελίδες του επόμενου επιπέδου να είναι επίσης σχετικές με το πεδίο, κ.ο.κ. Η στρατηγική αναζήτησης κατά πλάτος έχει επίσης χρησιμοποιηθεί και σε συνδυασμό με το εστιασμένο crawling [18] όπου οι σελίδες διαπερνιούνται αρχικά κατά πλάτος και στη συνέχεια οι μη σχετικές σελίδες φιλτράρονται από τη συλλογή με χρήση αλγορίθμου που αναλύει το περιεχόμενο. Σε σχέση με την στρατηγική αναζήτησης κατά πλάτος, η τεχνική αυτή μπορεί να χρίσει πολύ μεγαλύτερες συλλογές συγκεκριμένου πεδίου με πολύ λιγότερο θόρυβο. Παρόλα αυτά, επειδή πολλές μη σχετικές σελίδες ανακτώνται και επεξεργάζονται με τον αλγόριθμο της ανάλυσης του περιεχομένου, η μέθοδος αυτή έχει χαμηλή αποδοτικότητα.

Η στρατηγική αναζήτησης «το καλύτερο πρώτα» είναι η πιο γνωστή την τρέχουσα περίοδο στον τομέα των εστιασμένων crawlers [18, 19, 20, 21]. Σε αυτή τη στρατηγική τα URLs επισκέπτονται απλά με τη σειρά που εντοπίζονται. Αντίθετα μερικά ευρετικά (συχνά αποτελέσματα που προκύπτουν από αλγόριθμους ανάλυσης του ιστού) χρησιμοποιούνται για την κατάταξη των URLs στην σειρά με την οποία λαμβάνει χώρα το crawling. Τα αποτελέσματα που μοιάζουν πιο 'ελπιδοφόρα', γίνονται crawl πρώτα και

επομένως η τεχνική αυτή πλεονεκτεί σε σχέση με την κατά πλάτος. Παρόλα αυτά, έχει και ορισμένα μειονεκτήματα. Στο [22] δείχνεται ότι οι crawlers με τη στρατηγική αυτή ενδέχεται να χάνουν σημαντικές σελίδες και να έχουν ως αποτέλεσμα χαμηλή ανάκληση όσον αφορά στην τελική συλλογή και αυτό διότι το ευρετικό που χρησιμοποιείται είναι γενικά τοπικό (είναι εφικτό να ελεγχθούν μόνο οι κοντινοί γείτονες μιας σελίδας πριν παρθεί η απόφαση για το πόσο σχετική είναι).

2.3.4. **Κατανεμημένο** crawling

Η δεικτοδότηση του ιστού είναι μία μεγάλη πρόκληση λόγω της αύξησής του και λόγω της δυναμικής φύσης του. Καθώς το μέγεθος του παγκόσμιου ιστού αυξάνει, έχει γίνει επιβεβλημένη η παραλληλοποίηση της διαδικασίας του crawling προκειμένου να ολοκληρώνεται το κατέβασμα των σελίδων μέσα σε λογικά χρονικά πλαίσια. Μία μοναδική διεργασία crawling, ακόμα και αν είναι πολυνυματική, παραμένει μη επαρκής για μεγάλες μηχανές αναζήτησης που χρειάζεται να κατεβάζουν μεγάλες ποσότητες δεδομένων περιοδικά. Επίσης, όταν χρησιμοποιείται μία διεργασία, όλα τα δεδομένα που ανακτώνται περνάνε από το ίδιο μοναδικό physical link. Κατανέμοντας την διαδικασία του crawling σε πολλές διεργασίες και πολλά συστήματα μπορεί να βοηθήσει στην κατασκευή ενός κλιμακώσιμου, εύκολα παραμετροποιήσιμου συστήματος που στην ουσία είναι ανεκτικό στις βλάβες. Παράλληλα, ο διαχωρισμός του φόρτου ελαττώνει τις απαιτήσεις σε υλικό και ταυτόχρονα αυξάνει την συνολική ταχύτητα και αξιοπιστία. Κάθε εργασία λαμβάνει χώρα με πλήρως κατανεμημένο τρόπο και επομένως δεν χρειάζεται κεντρικοποιημένος έλεγχος του crawling.

2.3.5. **Ο κρυμμένος ιστός**

Οι δομημένες βάσεις δεδομένων του Web έχουν εξελιχθεί στον βασικό αποθηκευτικό χώρο που χρησιμοποιείται στο διαδίκτυο για την αποθήκευση σχεσιακών δεδομένων. Δεδομένου όμως το ότι αυτά τα δεδομένα δεν είναι άμεσα προσπελάσιμα στους

παραδοσιακούς Web Crawlers βάσει των διεπαφών που αυτοί έ χουν, το κομμάτι αυτό του ιστού αναφέρεται συχνά ως κρυμμένος ιστός (deep Web ή hidden Web)[16]

Για να μπορέσουμε να έχουμε πρόσβαση τελικά σε αυτόν τον τεράστιο όγκο δεδομένων υπάρχουν δύο βασικές σχεδιαστικές επιλογές. Η πρώτη είναι η data warehouse-like προσέγγιση[17,18] στην οποία τα δεδομένα συγκεντρώνονται από ένα μεγάλο πλήθος πηγών του Web και αξιοποιούνται (για εξόρυξη πληροφορίας) με κεντρικοποιημένο τρόπο. Η δεύτερη προσέγγιση είναι η MetaQuerier [23] που προσφέρει ένα πεδίο «αφαίρεσης» των βάσεων δεδομένων του ιστού δίνοντας έτσι ένα ενδιάμεσο σχήμα στους χρήστες. Μαζί με πολλές ακόμη διαφορές μεταξύ αυτών των προσεγγίσεων, έχουν διαφορετικούς στόχους όσον αφορά στην off-line απόκτηση δεδομένων. Για την περίπτωση της data-warehouse, η ενσωμάτωση δεδομένων από δομημένες πηγές του ιστού και η συγκέντρωσή τους σε ένα κεντρικό σημείο είναι ένα σημαντικό αρχικό βήμα εφόσον τα ερωτήματα των χρηστών απαντώνται αποκλειστικά από τα δεδομένα που αποθηκεύονται στο κεντρικοποιημένο warehouse. Σε αντίθεση, η τεχνική του MetaQuerier, θέτει λιγότερες απαιτήσεις στην απόκτηση των δεδομένων διότι αρκετά ερωτήματα τυπικά επαρκούν για αντιστοίχηση ερωτημάτων μέσω κάποιου σχήματος [24].

Ενώ δεν είναι ακόμη σαφές αν η προσέγγιση τύπου MetaQuerier ή αυτή που βασίζεται στο warehousing είναι πιο κατάλληλη για την πρόσβαση στον κρυμμένο ιστό, η απόκτηση δεδομένων από δομημένες πηγές του ιστού είναι ένα ενδιαφέρον πεδίο από μόνο του. Πολλές εφαρμογές μπορούν να αναπτυχθούν (και έχουν ήδη αναπτυχθεί) για να αξιοποιούν τα δομημένα δεδομένα από τις κρυμμένες πηγές του ιστού. Για παράδειγμα στο [25] οι εφαρμογές αξιοποιούν τα δεδομένα για τεχνικές εκπαίδευσης συγκεκριμένων πεδίων γνώσης. Επίσης εφαρμογές που εφαρμόζουν «σύγκριση αγορών» ενσωματώνοντας δεδομένα από διαφορετικούς (πιθανά ανταγωνιστικούς) προμηθευτές. Τα δεδομένα του κρυμμένου ιστού μπορούν επίσης να βοηθήσουν στην δόμηση άλλων λιγότερο δομημένων εγγράφων. Για την ακρίβεια πολλές μηχανές αναζήτησης έχουν

[16] http://www.press.umich/edu/jep/07-01/bergman.html
[17] http://froogle.google.com
[18] http://shopping.msm.com/

αρχίσει να παρέχουν υπηρεσίες αναζήτησης προϊόντων βασισμένες σε δομημένα δεδομένα που μαζεύουν από πλήθος πηγών του ιστού[19].

Στην βιβλιογραφία, υπάρχουν δύο τρόποι για να ανακτηθούν τα δεδομένα από τις διάφορες πηγές του ιστού. Η πιο αποτελεσματική μέθοδος είναι αν αφήσουμε τις ίδιες τις πηγές να εξαγάγουν τις βάσεις τους με βάσει κάποιου είδους αδειοδότηση έτσι τα δεδομένα τους μπορούν να δεικτοδοτηθούν άμεσα. Δυστυχώς, στο αυτόνομο, μη-συνεργατικό και ανταγωνιστικό περιβάλλον του παγκόσμιου ιστού, αυτή η προσέγγιση δύσκολα κλιμακώνεται λόγω του πλήθους των ιστότοπων και λόγω του ότι απαιτεί ένα σημαντικό χρονικό διάστημα χώρια τον ανθρώπινο παράγοντα. Η εναλλακτική προσέγγιση βασίζεται σε έναν κρυφό Web Crawler που θέτει ερωτήματα διαρκώς στην κρυμμένη βάση δεδομένων «ξετυλίγοντας» το περιεχόμενό της. Τα ερωτήματα μπορούν να τίθενται είτε μέσω των φορμών που έχουν ήδη αυτοί οι ιστότοποι ή μέσω των δημοσιευμένων Web Services Interfaces που έχουν πολλοί ιστότοποι (π. χ. Amazon Web Service).

Σε αντίθεση με τις παραδοσιακές τεχνικές Crawling, το Crawling που βασίζεται σε ερωτήματα χαρακτηρίζεται από βρόχο που κάνει τ ην αλλαγή των ερωτημάτων (Query-harvest Decomposite loop) και που επαναληπτικά αποκαλύπτει την πληροφορία. Ένας τέτοιος Crawler αρχίζει με μερικά αρχικά ερωτήματα με μορφή attribute-value, π.χ. Actors, Hanks, Tom, Brand, IBM ανακτώντας και αποθηκεύοντας την πληροφορία που επιστρέφεται είτε σε HTML είτε σε XML μορφή τοπικά. Τα δεδομένα που επιστρέφονται με κάθε ερώτημα θεωρούνται πιθανά για να σχηματίσουν ένα μελλοντικό ερώτημα. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να γίνουν όλα τα πιθανά ερωτήματα ή μέχρι να λάβει χ ώρα κάποιο άλλο κριτήριο τερματισμού.

Σημαντική προσπάθεια έχει γίνει για να αυτοματοποιηθεί η προηγούμενη η προηγούμενη διαδικασία. Προς αυτή τη κατεύθυνση, οι τεχνικές προκλήσεις έγκειται στην αυτόματη συμπλήρωση φορμών [26] και στην εξαγωγή δομημένων δεδομένων [27]. Όμως ένα σημαντικό και γενικά δύσκολο πρόβλημα είναι το εξής: πως μπορούμε να επιλέξουμε

[19] http://froogle.google.com

καλά ερωτήματα ώστε να έχουμε ικανοποιητική κάλυψη της γνώσης που περιέχεται στην βάση δεδομένων μέσα πάντα σε ανεκτό κόστος επικοινωνίας με τον server;

Διαισθητικά, ενώ η τελική κάλυψη της γνώσης που είναι εφικτή είναι ουσιαστικά προκαθορισμένη από τα αρχικά ερωτήματα, τα κόστη επικοινωνίας είναι ευθέως εξαρτώμενα από την μέθοδο επιλογής των ερωτημάτων που τελικά χρησιμοποιείται. Στην πράξη, είναι συχνά αδύνατο για έναν Crawler να εξαντλήσει κάθε πιθανό ερώτημα στη βάση δεδομένων. Επομένως, μία μέθοδος αποτελεσματικής επιλογής ερωτημάτων είναι απαραίτητη για να πετύχουμε 'καλή' κάλυψη με λογικό κόστος επικοινωνίας. Παρά την εύκολη διατύπωση, το παραπάνω πρόβλημα δεν είναι απλό. Για παράδειγμα στο [28], οι συγγραφείς δείχνουν ότι μία επαρκής λύση για το πρόβλημα είναι τεχνικά non-trivial. Επίσης για να επιτευχθεί ισοδύναμη κάλυψη της βάσης δεδομένων, μία καλή μέθοδος επιλογής των queries μπορεί να έχει σημαντικά λιγότερο overhead σε σχέση με την naive μέθοδο. Στην ίδια εργασία επίσης αποδεικνύεται ότι το σημαντικότερο ζήτημα για το deep web crawling είναι στην κατάλληλη επιλογή των queries και το πρόβλημα μοντελοποιείται ως ένα πρόβλημα διάτρεξης γράφου. Υπό αυτή την έννοια, ο στόχος είναι η εύρεση ενός Weighted Minimum Dominating Set στον αντίστοιχο γράφο που αντιστοιχίζει πεδία με τιμές.

2.4. **Κρυμμένος Ιστός – Επιφανειακός Ιστός**

Στην υποενότητα αυτή παρουσιάζονται κάποια στατιστικά δεδομένα που αφορούν τις ποιοτικές και ποσοτικές διαφορές του κρυμμένου ιστού (deep web) από τον επιφανειακό (surface web). Τα δεδομένα αντλήθηκαν από τις εργασίες [3-5] και παρατίθενται ακολούθως:

- Ο όγκος πληροφοριών στον κρυμμένο ιστό υπολογίζεται περίπου 500 φορές μεγαλύτερος από αυτόν του επιφανειακού ιστού.
- Ο κρυμμένος ιστός περιλαμβάνει 7.500 terabytes όγκο δεδομένων, ενώ ο επιφανειακός ιστό μόλις 19.

- Ο ρυθμός ανάπτυξης του κρυμμένου ιστού είναι πολύ υψηλότερος σε σχέση με τον ρυθμό ανάπτυξης του επιφανειακού ιστού.
- Υπολογίζεται ότι αυτή τη στιγμή υπάρχουν περισσότερα από 2.000.000 sites στον κρυμμένο ιστό.
- Τα ανεξάρτητα έγγραφα στον κρυμμένο ιστό εκτιμώνται περίπου στα 550.000.000, τη στιγμή που ο επιφανειακός ιστός περιέχει περίπου 1.000.000.
- Οι περισσότερες πληροφορίες στον κρυμμένο ιστό, διατηρούνται από ακαδημαϊκά ιδρύματα και ερευνητικούς οργανισμούς, γι' αυτό το λόγο άλλωστε πολλοί ερευνητές του αντικειμένου υποστηρίζουν ότι η ποιότητα των πληροφοριών που βρίσκεται στον κρυμμένο ιστό είναι πολύ υψηλότερη από αυτή που βρίσκεται στον επιφανειακό.
- Το 95% των πληροφοριών που βρίσκονται στον κρυμμένο ιστό είναι προσβάσιμες, χωρίς την απαίτηση εγγραφής ή χρηματικού αντιτίμου.
- Το περιεχόμενο του κρυμμένου ιστού έχει μεγαλύτερη συνάφεια με την απαιτούμενη πληροφορία, σε σχέση με τον επιφανειακό ιστό.
- Περίπου το 55% του περιεχομένου του κρυμμένου ιστού είναι αποθηκευμένο σε βάσεις δεδομένων με συγκεκριμένη θεματική ενότητα.

Στις ενότητες που θα ακολουθήσουν θα αναλυθούν περεταίρω οι διαφοροποιήσεις μεταξύ του κρυμμένου και του επιφανειακού ιστού, και θα δοθεί ιδιαίτερη έμφαση στα χαρακτηριστικά του κρυμμένου ιστού.

3. ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΚΡΥΜΜΕΝΟΥ ΙΣΤΟΥ

Στη διεθνή βιβλιογραφία, ο κρυμμένος ιστός (deep web) αναφέρεται και ως invisible web ή hidden web. Η κατανόηση της πρακτικής του έννοιας δεν είναι εύκολα κατανοητή, ωστόσο η έννοια αυτή δεν μπορεί να προσδιοριστεί αυστηρώς από κάποιο ορισμό. Με απλά λόγια αυτό που στην ουσία εννοούμε με τον όρο κρυμμένος ιστός είναι το περιεχόμενο εκείνο το όποιο υπάρχει στο web αλλά δεν μπορεί να προσπελαστεί από τις μηχανές αναζήτησης γενικού σκοπού. Το περιεχόμενο αυτό μπορεί να αφορά αρχεία, σελίδες κειμένου και γενικότερα οποιαδήποτε άλλη πληροφορία η οποία δεν μπορεί να ανακτηθεί από τις μηχανές αναζήτησης γενικού σκοπού.

3.1. **Περιεχόμενο και μέγεθος του** Deep Web

Για την καλύτερη κατανόηση του είδους και του όγκου πληροφοριών που περιλαμβάνει ο κρυμμένος ιστός, αρχικά αναφέρεται ότι ο τρόπος μέσω του οποίου οι μηχανές αναζήτησης ανακτούν το περιεχόμενο του web, είναι κάποια ειδικά προγράμματα, οι web crawlers, οι οποίοι στην ουσία ακολουθούν συνδέσμους (links). Η χρήση της τεχνικής αυτής είναι πολύ αποτελεσματική στην εύρεση πληροφοριών από τον επιφανειακό ιστό (Surface Web), όχι όμως κι από τον κρυμμένο ιστό. Αυτό είναι απόλυτα λογικό αν σκεφτεί κανείς ότι ο web crawler πλοηγείται στον ιστό μέσω των συνδέσμων που είναι διαθέσιμοι σε κάθε σελίδα. Αν λοιπόν υπάρχει μια σελίδα για την οποία δεν υπάρχει σύνδεσμος σε καμία άλλη, τότε δεν μπορεί να εντοπιστεί από τον web crawler. Τέτοιου είδους σελίδες αποτελούν μέρος του κρυμμένου ιστού.

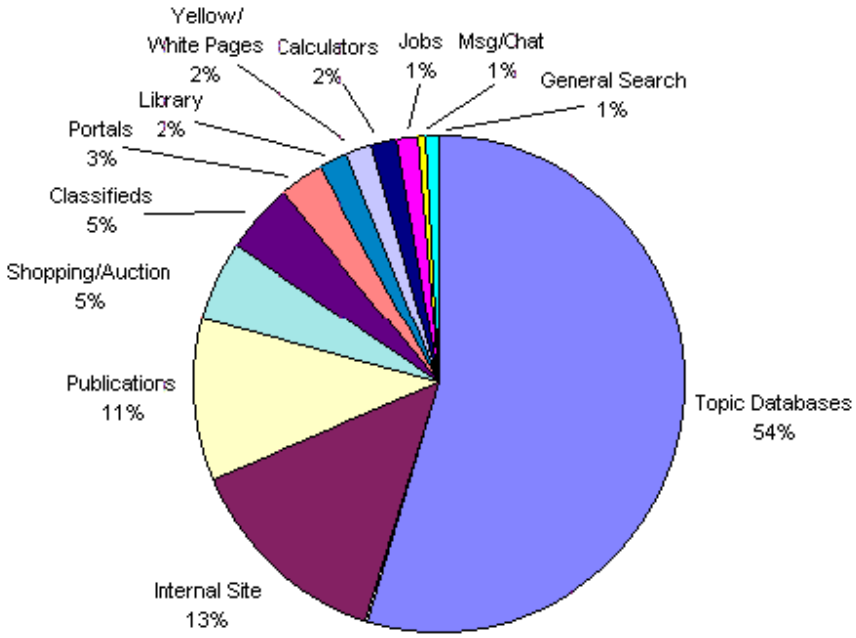
3.1.1. **Περιεχόμενο του** deep web

Γενικά, ο κρυμμένος ιστός μπορεί να περιέχει τις ακόλουθες κατηγορίες:

- *Μη συνδεδεμένες σελίδες*: Δεν υπάρχει σύνδεσμος που να κατευθύνει σε αυτή τη σελίδα. Πρακτικά με αυτό τον τρόπο εμποδίζονται οι web crawlers να αποκτήσουν πρόσβαση στο περιεχόμενό τους.
- *Περιεχόμενο βάσεων δεδομένων*: Οι web crawlers δεν μπορούν να αλληλεπιδράσουν με τις φόρμες αναζήτησης που παρέχονται στους πραγματικούς χρήστες.
- *Σελίδες που δεν έχουν **HTML** μορφή και περιέχουν κυρίως οπτικοακουστικό υλικό*: Λόγω της μη ύπαρξης "αρκετού" κειμένου, οι μηχανές αναζήτησης δεν μπορούν να δημιουργήσουν λέξεις κλειδιά για να εντοπίσουν αυτό το περιεχόμενο. Για παράδειγμα, μια σελίδα που περιέχει μόνο γραφικά δεν θα μπορούσε να προσπελαθεί καθώς δεν υπάρχει κάποια λέξη κλειδί που θα μπορούσε να καταχωρηθεί στη μηχανή αναζήτησης.
- *Σελίδες με εκτελέσιμα ή συμπιεσμένα αρχεία*: Συνήθως είναι προσπελάσιμα αλλά πολλές φορές οι μηχανές αναζήτησης τα απορρίπτουν εσκεμμένα (συνήθως για λόγους προστασίας).
- *Περιεχόμενο περιορισμένης πρόσβασης*: Αφορά σελίδες που δεν επιτρέπουν την περιήγηση στο περιεχόμενο τους (ή σε μέρος του περιεχομένου τους). Συνήθως, τέτοιου τύπου sites χωρίζονται σε δύο μεγάλες κατηγορίες:
    - εκείνα τα οποία απαιτούν εγγραφή του χρήστη (registration), δηλαδή οι σελίδες εκείνες στις οποίες απαιτείται η εισαγωγή κάποιου "ονόματος χρήστη (username)" και "κωδικού (password)" και ως εκ τούτου δεν μπορούν να προσπελαστούν από τους web crawlers.
    - εκείνα τα οποία χρησιμοποιούν ειδικά προγράμματα, τα οποία αποτρέπουν την πρόσβαση των web crawlers στο περιεχόμενό τους.
- *Δυναμικό Περιεχόμενο*: Περιεχόμενο το οποίο δημιουργείται δυναμικά ανάλογα με τις απαιτήσεις του χρήστη, δηλαδή δυναμικές σελίδες οι οποίες προκύπτουν ως απάντηση (response) σε ένα ερώτημα (query) ή δυναμικές σελίδες που μπορούν να προσπελαστούν μέσω κάποιας φόρμας. Αυτό είναι σύνηθες στις αναζητήσεις που πραγματοποιούνται σε βάσεις δεδομένων, γι' αυτό το λόγο

άλλωστε το μεγαλύτερο μέρος των εγγράφων που βρίσκονται στο κρυμμένο ιστό ενυπάρχει σε τέτοιου τύπου βάσεις δεδομένων.

Σύμφωνα με τον Michael K. Bergman [3], ο κρυμμένος ιστός περιλαμβάνει σε ποσοστά τις ακόλουθες ενότητες περιεχομένων.



Εικόνα 3 - Κατανομή του κρυμμένου ιστού με βάση το περιεχόμενο

Όπως παρουσιάζεται και στο ανωτέρω γράφημα, το 54% αφορά βάσεις δεδομένων με συγκεκριμένη θεματική ενότητα. Αν σε αυτό το ποσοστό προσθέσουμε το 13% των εσωτερικών εγγράφων που βρίσκονται σε μεγάλους ιστότοπους και το 11% που αφορά τις δημοσιεύσεις, τότε διαπιστώνουμε ότι η συντριπτική πλειοψηφία του κρυμμένου ιστού (78%) καλύπτεται από αυτές τρεις μεγάλες κατηγορίες

3.1.2. **Μέγεθος του** deep web

Το ακριβές μέγεθος του κρυμμένου ιστού, είναι ουσιαστικά αδύνατο να υπολογιστεί. Στην ουσία μπορεί μόνο μια εκτίμηση να γίνει για το μέγεθός του. Πιο συγκεκριμένα, η εκτίμηση του όγκου των δεδομένων που υπάρχει στο κρυμμένο ιστό αποτελεί ανοικτό πρόβλημα από το 1998. Στη διάρκεια αυτών των χρόνων αναπτύχθηκαν διάφορες τεχνικές για την εκτίμηση του μεγέθους του. Στην εργασία του Jie Liang [5] γίνεται μια σύνοψη όλων των τεχνικών που αναπτύχθηκαν τα τελευταία χρόνια σχετικά με τον υπολογισμό του όγκου των δεδομένων που υπάρχουν στον κρυμμένο ιστό.

Η βασική τεχνική για τον υπολογισμό των σχετικών μεγεθών του όγκου δεδομένων προτάθηκε από τους Bharat και Broder [1] και αφορά το ακόλουθο πιθανοκρατικό μοντέλο, όπου αν θεωρηθούν τα σύνολο A και B τότε:

- P(A), η πιθανότητα ένα στοιχείο να ανήκει στο A.
- P(A∩B | A), η πιθανότητα ένα στοιχείο να ανήκει στη τομή των A και B και ταυτόχρονα να ανήκει και στο A.

Τότε,

$$P(A \cap B|A) = \frac{|A \cap B|}{|A|}$$

Ισοδύναμα προκύπτει ότι

$$P(A \cap B|B) = \frac{|A \cap B|}{|A|}$$

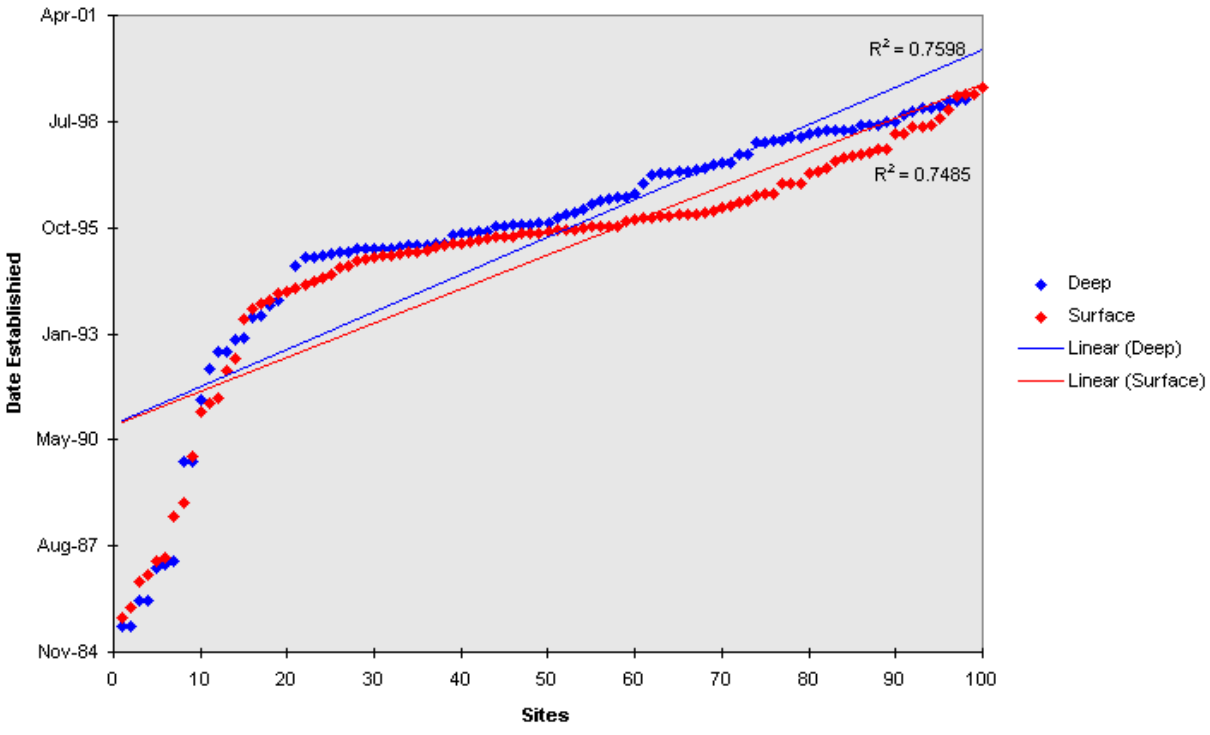
Από τις δύο προηγούμενες εξισώσεις εξάγεται η εξίσωση

$$\frac{|A|}{|B|} = \frac{P(A \cap B|B)}{P(A \cap B|A)}$$

Σύμφωνα με έρευνα που διεξήχθη στο Πανεπιστήμιο Berkeley της Καλιφόρνια το 2001, ο κρυμμένος ιστός αποτελείται περίπου από 91.000 terabytes, εν αντιθέσει με τον επιφανειακό ιστό, όπου είναι περίπου 167 terabytes. Περίπου 550 εκατομμύρια ανεξάρτητων εγγράφων είναι διαθέσιμα στον κρυμμένο ιστό, τη στιγμή που ο επιφανειακός ιστός διαθέτει περίπου ένα εκατομμύριο [3].

Ο κρυμμένος ιστός είναι η μεγαλύτερη σε ανάπτυξη αποθήκη πληροφοριών στο Διαδίκτυο. Οι πληροφορίες που υπάρχουν στον κρυμμένο ιστό υπολογίζονται ότι είναι περίπου 500 φορές μεγαλύτερες σε όγκο από αυτές που είναι διαθέσιμες στον Παγκόσμιο Ιστό. Επίσης η ποιότητα του περιεχομένου του είναι 1.000 με 2.000 φορές καλύτερη σε σχέση με τον επιφανειακό ιστό [4].

Στο Σχήμα 2, όπως αυτό παρουσιάζεται στο [3] απεικονίζεται ο ρυθμός ανάπτυξης του επιφανειακού και του κρυμμένου ιστού σε σχέση με το χρόνο.



Εικόνα 4 - Σύγκριση του ρυθμού ανάπτυξης επιφανειακού και κρυμμένου ιστού

Συμπερασματικά, μπορεί να ειπωθεί ότι όσο αυξάνεται ο όγκος πληροφοριών στον επιφανειακό ιστό, τόσο θα υπάρχει μια αύξηση στον όγκο των πληροφοριών που υπάρχουν στον κρυμμένο ιστό.

3.2. Λόγοι χρήσης του κρυμμένου Ιστού

Υπάρχουν διάφορες αιτίες για τις οποίες ο χρήστης θα επέλεγε τη χρήση του κρυμμένου ιστού για την εύρεση των πληροφοριών που επιθυμεί, ωστόσο όπως συμβαίνει και στα περισσότερα πράγματα, έτσι και η χρήση του κρυμμένου ιστού έχει κάποια πλεονεκτήματα και μειονεκτήματα.

3.2.1. Περιορισμοί των μηχανών αναζήτησης

Οι μηχανές αναζήτησης συνέβαλαν τα μέγιστα στην δυνατότητα των ανθρώπων να έχουν πρόσβαση στο περιεχόμενο του διαδικτύου και είναι ένα σημαντικό συστατικό μιας επιτυχούς έρευνας. Ωστόσο υπάρχουν αρκετοί λόγοι για τους οποίους μπορεί να μην είναι η βέλτιστη επιλογή. Ακολούθως παρατίθενται κάποιοι από τους λόγους:

- *Ποιότητα αποτελεσμάτων*: Πολλές μηχανές αναζήτησης επιστρέφουν αποτελέσματα βάση της συνάφειας ταξινόμησης σύμφωνα με τις λέξεις κλειδιά που πληκτρολογεί ο χρήστης. Η ποιότητα των αποτελεσμάτων δεν λαμβάνεται υπόψη με αποτέλεσμα να εμφανίζονται πολλές σελίδες που το περιεχόμενό τους είναι αμφιβόλου ποιότητας.
- *Ποσότητα αποτελεσμάτων*: Οι μηχανές αναζήτησης επιστρέφουν πληθώρα αποτελεσμάτων, χωρίς ωστόσο να υπάρχει διακριτοποίηση ανάμεσα στις ποιοτικές σελίδες και σε εκείνες που παρέχουν επισφαλείς πληροφορίες. Επίσης, ένα άλλο σημαντικό θέμα που προκύπτει, είναι το γεγονός ότι ένα υψηλό ποσοστό των αποτελεσμάτων που προκύπτουν δεν είναι προσπελάσιμα από τον

χρήστη, δηλαδή ενώ οι μηχανές αναζήτησης εμφανίζουν στον αριθμό των αποτελεσμάτων τους κάποιες σελίδες, αυτές στην πράξη δεν είναι προσβάσιμες.

- *Επιφανειακή αναζήτηση*: Είναι σύνηθες το φαινόμενο στα αποτελέσματα που προκύπτουν από τις μηχανές αναζήτησης να εμφανίζονται μόνο 2-3 σελίδες από έναν ιστότοπο κι όχι σελίδες που βρίσκονται σε "βαθύτερο επίπεδο". Για παράδειγμα σ' ένα μεγάλο ιστότοπο με πολλές σελίδες, μπορεί να επιστραφούν μόνο η αρχική σελίδα και 1-2 ακόμα. Αυτό συμβαίνει λόγω της επιθυμίας των σχεδιαστών των μηχανών αναζήτησης να μειώσουν το υπολογιστικό κόστος, καθώς είναι αρκετά χρονοβόρο για τα "spiders" να ανιχνεύσουν τον ιστό. Αυτό όμως έχει σαν ενδεχόμενο αποτέλεσμα να αποκλείονται από τα αποτελέσματα τα περιεχόμενα ενός ποιοτικού ιστότοπου με πολλά επίπεδα σελίδων.
- *Προτιμήσεις των μηχανών αναζήτησης*: Ανάλογα με το πώς ο χρήστης έχει ρυθμίσει τις προτιμήσεις του, μια μηχανή αναζήτησης ενώ μπορεί να βρει και να εμφανίσει κάποια διαθέσιμα έγγραφα, τελικά τα αποκλείει λόγω των ρυθμίσεων που έχουν καθοριστεί από τον χρήστη. Επίσης, είναι σύνηθες το φαινόμενο εάν ένα αποτέλεσμα δεν έχει εμφανιστεί στα πρώτα 20 της λίστας ο χρήστης να αποφεύγει να προχωρήσει στα επόμενα.
- *Όριο εμφάνισης μηχανών αναζήτησης*: Κάποιες μηχανές αναζήτησης μπορεί να έχουν κάποιο όριο στον αριθμό των αποτελεσμάτων που μπορούν να εμφανίσουν από έναν ιστότοπο. Για παράδειγμα, η μηχανή αναζήτησης της Google επιτρέπει 1-2 αποτελέσματα στο ευρετήριο του από έναν ιστότοπο για ένα θέμα αναζήτησης. Πρακτικά αυτό σημαίνει ότι για να προβάλει κάποιος τις υπόλοιπες σχετικές σελίδες από τον ιστότοπο, πρέπει να επιλέξει την αντίστοιχη επιλογή.
- *Επιχειρήσεις και δημοτικότητα*: Οι περισσότερες μηχανές αναζήτησης ταξινομούν τα αποτελέσματά τους σύμφωνα με το πόσο δημοφιλής είναι η κάθε σελίδα. Για παράδειγμα, η μηχανή αναζήτησης της Google επιστρέφει αποτελέσματα βάση του ποια είναι τα πιο δημοφιλή και ευρέως γνωστά sites για το θέμα αναζήτησης. Ένα σημαντικό κριτήριο που λαμβάνεται υπόψη είναι το πόσες σελίδες έχουν ως σύνδεσμο το "δημοφιλές site". Αυτό ωστόσο δεν σημαίνει απαραίτητα ότι αυτές είναι καλύτερες σελίδες για την κάλυψη των αναγκών μας. Επίσης, πρέπει να

τονισθεί ότι πίσω από τις περισσότερες μηχανές αναζητήσεις είναι επιχειρήσεις που ως βασική επιδίωξη έχουν το κέρδος. Ένα μεγάλο λοιπόν κέρδος προσφέρεται σε αυτές, μέσω των εσόδων από διαφημίσεις και ως γνωστό πολλές επιχειρήσεις επενδύουν πολλά χρήματα ώστε να καταφέρουν να εμφανίζεται ο ιστότοπός τους στην κορυφή των αποτελεσμάτων.

3.2.2. **Οφέλη από τη χρήση του κρυμμένου ιστού**

Τα οφέλη που έχει ο χρήστης από την αναζήτηση πληροφοριών στο κρυμμένο ιστό, είναι σε άμεση συσχέτιση με εκείνους τους λόγους, για τους οποίους οι μηχανές αναζήτησης πολλές φορές δεν είναι σε θέση να μας επιστρέψουν τις πληροφορίες που χρειαζόμαστε τόσο ποσοτικά, όσο και ποιοτικά.

Οι κυριότεροι λόγοι για τους οποίους ο χρήστης θα επέλεγε τον κρυμμένο ιστό παρουσιάζονται ακολούθως:

- *Μη διαθεσιμότητα στο web*: Ο κυριότερος λόγος για τη χρήση του κρυμμένου ιστού είναι η αδυναμία εύρεσης πολλών πληροφοριών στον επιφανειακό ιστό. Όπως αναφέρθηκε, οι μηχανές αναζήτησης γενικού σκοπού εκτελούν ειδικά προγράμματα (spiders ή crawlers) τα οποία ανιχνεύουν τα περιεχόμενα των σελίδων, ακολουθώντας τους συνδέσμους που βρίσκονται σε κάθε σελίδα. Τα περισσότερα όμως περιεχόμενα που βρίσκονται στον κρυμμένο ιστό, υπάρχουν μέσα σε βάσεις δεδομένων άρα παραμένουν κρυμμένα από τις μηχανές αναζήτησης. Έτσι, τα προγράμματα που χρησιμοποιούνται για την αναζήτηση πληροφοριών στον κρυμμένο ιστό, παρέχουν πληθώρα αποτελεσμάτων που οι παραδοσιακές μηχανές αναζήτησης δεν δύναται να τις εμφανίσουν.
- *Εξειδίκευση*: Συνήθως οι πόροι-δεδομένα του κρυμμένου ιστού εστιάζουν σε συγκεκριμένα θέματα, έτσι δύνεται η δυνατότητα στους χρήστες να ανακτήσουν πολύ πιο συγκεκριμένα και περιεκτικά αποτελέσματα.
- *Εστίαση*: Οι διεπαφές αναζήτησης στον κρυμμένο ιστό σχεδιάζονται με βάση τον τύπο αναζήτησης που γίνεται.

- *Ποιότητα - αξιοπιστία*: Τα περιεχόμενα του κρυμμένου ιστού πολλές φορές δημιουργούνται από οργανισμούς και ιδρύματα που έχουν την εποπτεία των θεμάτων που καλύπτουν, γι' αυτό το λόγο οι αντίστοιχες πηγές είναι και πιο αξιόπιστες.

4. ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ

Στην ενότητα αυτή, γίνεται μια αναφορά στην έννοια της αναζήτησης πληροφοριών στον παγκόσμιο ιστό, καθώς και στους διάφορους αλγόριθμους που αναπτύχθηκαν για την ανάκτηση πληροφορίας. Μέσω της παρουσίασης που θα γίνει στη συνέχεια, καθίσταται ευκολότερα κατανοητή η διαδικασία της ανάκτησης πληροφοριών από το κρυμμένο ιστό και η δυσκολία που αντιμετωπίζουν οι υπάρχουσες τεχνικές για την ανάκτηση των δεδομένων από αυτόν. Χαρακτηριστικό παράδειγμα εφαρμογής, είναι οι "κρυφές" (για μη εγγεγραμμένους χρήστες) βάσεις δεδομένων.

Η πρακτική της αποθήκευσης και ανάκτησης πληροφορίας ήταν από τα θέματα που απασχόλησε τον κόσμο από την αρχαιότητα. Με την πάροδο των χρόνων, η ανάγκη αποθήκευσης και ανάκτησης πληροφορίας αυξήθηκε ιδιαίτερα. Επίσης, με την εισαγωγή των ηλεκτρονικών υπολογιστών στη ζωή του σύγχρονου κόσμου, διαπιστώθηκε η μεγάλη χρησιμότητα που θα είχαν στον τομέα της αποθήκευσης και ανάκτησης πληροφορίας. Το 1945 ο Vannevar Bush δημοσίευσε το άρθρο με τίτλο «As We May Think» [6], στο οποίο εισήγαγε την ιδέα της αυτόματης πρόσβασης σε μεγάλες ποσότητες αποθηκευμένων δεδομένων.

Στη δεκαετία του 50 η ιδέα του Bush άρχισε να υλοποιείται σε πιο συγκεκριμένες περιγραφές για το πώς είναι δυνατόν αποθηκευμένα κείμενα να ανιχνευτούν με αυτοματοποιημένο τρόπο. Στα μέσα της δεκαετίας του 50, πολλοί ερευνητές ασχολήθηκαν με την αναζήτηση κειμένου μέσω υπολογιστή. Μια από τις πιο γνωστές μεθόδους αναπτύχθηκε από τον Luhn το 1957 [7]. Στην εργασία αυτή, ο Luhn προτείνει τη χρήση λέξεων ως μονάδων ευρετηριοποίησης για τα κείμενα και μέτρησης της επικάλυψης της λέξης σαν κριτήριο για την ανάκτηση.

Στη διάρκεια της δεκαετίας του 60, σημειώθηκαν κάποια σημαντικά επιτεύγματα στο πεδίο της ανάκτησης δεδομένων. Οι πιο σημαντικές, ήταν η ανάπτυξη του "SMART Retrieval System" από τον Gerald Salton [8] και οι εκτιμήσεις που πραγματοποίησε ο Cyril Cleverdon [9]. Η μεθοδολογία αξιολόγησης των συστημάτων ανάκτησης που

ανέπτυξε ο Cyril Cleverdon, χρησιμοποιείται μέχρι και σήμερα από τα συστήματα ανάκτησης πληροφορίας (Information Retrieval Systems).

Στις δύο δεκαετίες που ακολούθησαν, αναπτύχθηκαν διάφορα μοντέλα για την ανάκτηση κειμένων. Το κύριο χαρακτηριστικό που είχαν τα περισσότερα από τα μοντέλα αυτά, ήταν η αποδοτικότητά τους στη συλλογή μικρών κειμένων. Ωστόσο, λόγω της έλλειψης διαθεσιμότητας στη συλλογή μεγάλων κειμένων οι τεχνικές αυτές παρουσίαζαν σημαντικά κενά στην εφαρμογή τους. Το πρόβλημα αυτό ουσιαστικά λύθηκε το 1992 με την εισαγωγή του "Text Retrieval Conference" (TREC) [10]. Το TREC στην ουσία ήταν μια σειρά από συνέδρια αξιολογήσεων επιδοτούμενα από διάφορες υπηρεσίες της αμερικάνικης κυβέρνησης, υπό την επίβλεψη του National Institute of Standards and Technology (NIST), που σκοπός του ήταν η ενθάρρυνση της έρευνας στο πεδίο της ανάκτησης πληροφορίας από συλλογές μεγάλων κειμένων.

Οι αλγόριθμοι που αναπτύχθηκαν στο πεδίο έρευνας της ανάκτησης πληροφορίας, ήταν οι πρώτοι που χρησιμοποιήθηκαν για την ανίχνευση πληροφορίας στον Παγκόσμιο Ιστό μεταξύ των ετών 1996 και 1998. Αργότερα ωστόσο, για την ανίχνευση πληροφορίας στον ιστό, δόθηκε έμφαση σε συστήματα που αξιοποιούν την τεχνολογία του ιστού και συγκεκριμένα του πλεονεκτήματος που δίνει η διασταυρούμενη σύνδεση.

4.1. Έννοια και ορισμός της ανάκτησης πληροφορίας

Η έννοια της ανάκτηση πληροφορίας είναι κάτι περισσότερο από έναν απλό ορισμό. Στην ουσία είναι ένα ολόκληρο πεδίο έρευνας που ασχολείται με την αναζήτηση:

- Κειμένων
- Πληροφορίας που εντοπίζεται μέσα στα κείμενα
- Μεταδεδομένα σχετικά με τα κείμενα
- Βάσεων Δεδομένων
- Παγκόσμιου Ιστού.

Οι όροι ανάκτηση δεδομένων, κειμένου και πληροφορίας πολλές φορές συγχέονται, ωστόσο ο κάθε όρος έχει τη δική του θεωρητική προσέγγιση, τη δική του εφαρμογή καθώς και τις δικές του τεχνολογίες.

Η ανάκτηση πληροφορίας συνδυάζει πολλά πεδία επιστημών και βασίζεται στην επιστήμη των υπολογιστών, στα μαθηματικά, στην αρχιτεκτονική πληροφορίας και σε άλλες. Ως σύντομο ορισμό για την ανάκτηση πληροφορίας μπορεί να δοθεί ο ακόλουθος:

Ορισμός: Η ανάκτηση πληροφορίας σχετίζεται με τις τεχνικές της αποθήκευσης, της ανάκτησης και συχνά της διάδοσης καταγεγραμμένων δεδομένων, ειδικά μέσω της χρήσης ηλεκτρονικών συστημάτων.

4.2. **Διαδικασία Ανάκτησης Πληροφορίας**

Στο σημείο αυτό θα δοθεί μια περιγραφή της διαδικασίας που εκτελείται για την ανάκτηση πληροφορίας από τον χρήστη. Η εκκίνηση της διαδικασίας γίνεται με την εισαγωγή ενός ερωτήματος (query) από τον χρήστη στο σύστημα. Τα ερωτήματα (queries), θέτονται σε μια βάση δεδομένων (πχ αναζήτηση στον Παγκόσμιο Ιστό μέσω μιας μηχανής αναζήτησης). Κατά την ανάκτηση της/των πληροφορίας/ών, στο ερώτημα μπορεί να ταιριάζουν πολλά αντικείμενα με διαφορετικούς βαθμούς σχετικότητας, με άλλα λόγια ένα αντικείμενο δεν προσδιορίζεται μονοσήμαντα από το ερώτημα.

Με τον όρο αντικείμενο, ορίζεται μια οντότητα που χαρακτηρίζεται από κάποιες πληροφορίες που υπάρχουν σε μια βάση δεδομένων. Για την απάντηση των ερωτημάτων που θέτονται από τους χρήστες, πραγματοποιείται αναζήτηση των στοιχείων που ενυπάρχουν στη βάση δεδομένων για απαντήσεις που πληρούν τα κριτήρια. Το πλήθος των συστημάτων ανάκτησης πληροφορίας, ταξινομεί τα αντικείμενα με βάση κάποιον αλγόριθμο που αναλαμβάνει να αξιολογήσει το βαθμό συσχέτισης του ερωτήματος με τα αντικείμενα στη βάση δεδομένων. Όσο πιο υψηλή θέση κατέχουν τα αντικείμενα στη λίστα, τόσο μεγαλύτερος (θεωρητικά) ο βαθμός συσχέτισης. Τα αντικείμενα που

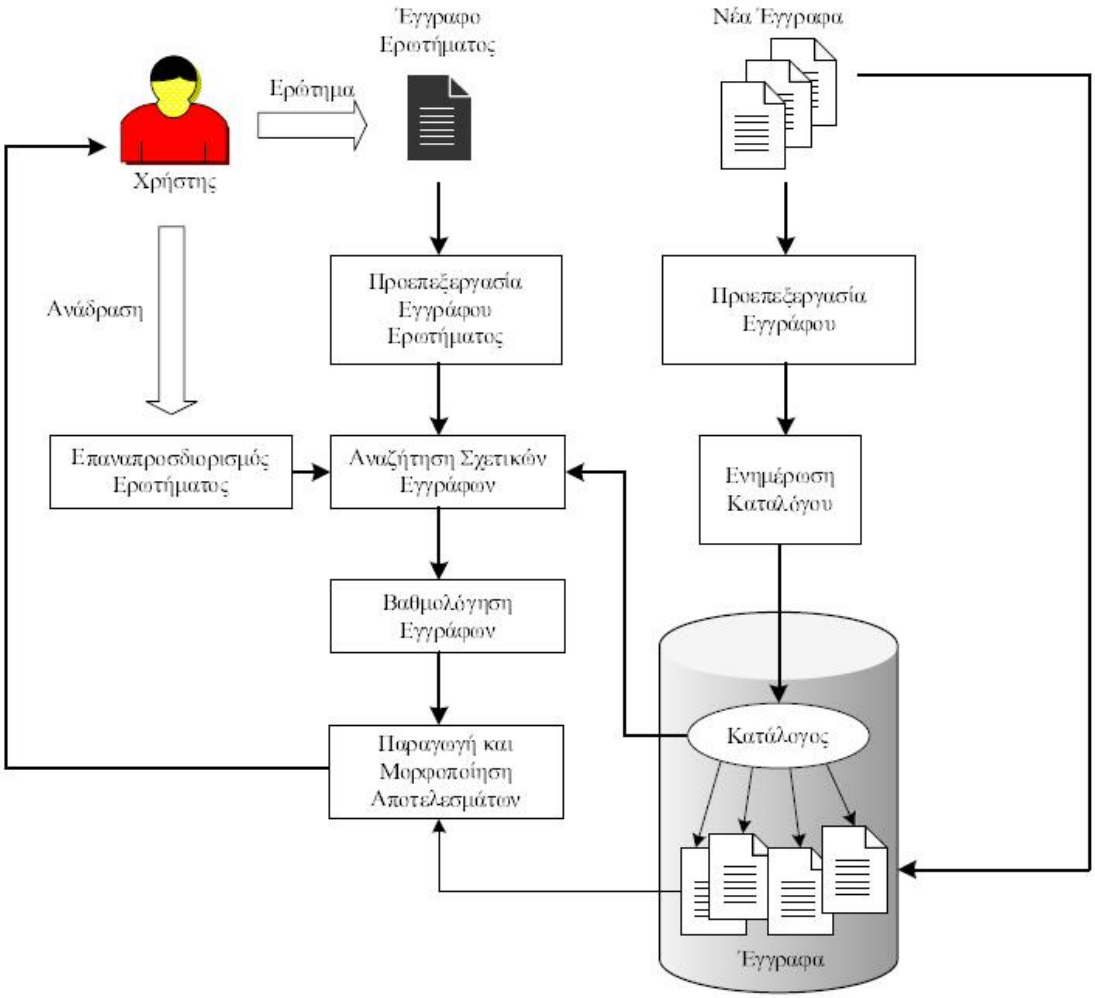
κατέχουν τις θέσεις στις πρώτες θέσεις της λίστας είναι και τα αποτελέσματα τα οποία εμφανίζονται στον χρήστη.

Στο σχήμα που ακολουθεί βλέπουμε τα στάδια της διαδικασίας, με τη σειρά που εκτελούνται, για την ανάκτηση της ζητούμενης πληροφορίας. Επιπροσθέτως, ακολουθεί μια συνοπτική περιγραφή του κάθε σταδίου και της αντίστοιχης διεργασίας που εκτελείται.

Στο σημείο αυτό και με βάση το ακόλουθο διάγραμμα, πρέπει να τονιστεί ότι ο χρήστης μπορεί να εκτελέσει δυο ενέργειες:

α) την υποβολή ενός νέου κειμένου προς αποθήκευση (για να εκτελεστεί αυτή η ενέργεια θα πρέπει ο χρήστης να κατέχει το αντίστοιχο δικαίωμα).

β) την υποβολή ενός ερωτήματος



Εικόνα 5 - Διαδικασία Ανάκτησης Πληροφορίας[20]

4.2.1. **Υποβολή νέου κειμένου προς αποθήκευση**

Στη περίπτωση όπου μελετάται η πρώτη ενέργεια, δηλαδή η υποβολή ενός νέου κειμένου από τον χρήστη τα στάδια είναι τα ακόλουθα:

1ο Στάδιο: Το πρώτο στάδιο αφορά την υποβολή του νέου κειμένου από τον χρήστη. Όπως αναφέρθηκε και νωρίτερα, για να είναι κάτι τέτοιο εφικτό θα πρέπει ο χρήστης να είναι κάτοχος ανάλογων δικαιωμάτων. Είθισται, δικαιώματα καταχώρισης νέων

[20] Πηγή: http://delab.csd.auth.gr/courses/c_ir/irbook.pdf

κειμένων να έχουν μόνο οι εξουσιοδοτημένοι χρήστες που είναι και υπεύθυνοι για το περιεχόμενο των συστημάτων ανάκτησης πληροφορίας.

2^{ο} Στάδιο: Το νέο κείμενο που είναι προς υποβολή από το χρήστη, υποβάλλεται στο στάδιο της προ-επεξεργασίας κειμένου. Στο στάδιο αυτό η κύρια διαδικασία που εκτελείται είναι η μετατροπή – προσαρμογή του κειμένου σε κατάλληλη μορφή ούτως ώστε να μπορεί να αναπαρασταθεί στο σύστημα ανάκτησης πληροφορίας. Η προ-επεξεργασία αυτή μπορεί να αφορά την απάλειψη λέξεων ή φράσεων που θεωρείται ότι δεν περιέχουν κάποια ιδιαίτερη ποσότητα πληροφορίας (αυτή η επεξεργασία ονομάζεται και Stop words removal).

3^{ο} Στάδιο: Από τη στιγμή που καταχωρείται το νέο κείμενο από το χρήστη και αφού διαμορφωθεί κατάλληλα ώστε να μπορεί να αναπαρασταθεί από το πληροφοριακό σύστημα, ενημερώνεται ο κατάλογος (index). Ο κατάλογος (ή ευρετήριο) είναι το μέρος εκείνο του συστήματος που έχει την ευθύνη για τη γρήγορη αναζήτηση των λέξεων. Στόχος του είναι ο προσδιορισμός των κειμένων που είναι σχετικά με το ερώτημα. Ανάμεσα στα περιεχόμενα του καταλόγου και στα περιεχόμενα των κειμένων, υπάρχει γραμμική εξάρτηση, κάτι το οποίο σημαίνει ότι σε κάθε μεταβολή των περιεχομένων των κειμένων υπάρχει αντίστοιχη μεταβολή στα περιεχόμενα του καταλόγου μέσω της αντίστοιχης ενημέρωσης που εκτελείται.

4.2.2. **Υποβολή ερωτήματος από χρήστη**

Στη περίπτωση τώρα όπου μελετάται η δεύτερη ενέργεια, δηλαδή η υποβολή ενός ερωτήματος από τον χρήστη τα στάδια είναι τα ακόλουθα:

1^{ο} Στάδιο: Είναι το στάδιο κατά το οποίο ο χρήστης υποβάλει το ερώτημα προς το σύστημα ανάκτησης πληροφορίας. Το ζητούμενο στο στάδιο αυτό είναι ο προσδιορισμός των κατάλληλων λέξεων για την ανάκτηση της πληροφορίας που θα αναζητηθεί. Πολλές φορές κρίνεται σκόπιμη και χρήση κάποιων τελεστών για τον καλύτερο προσδιορισμό των αποτελεσμάτων (όπως γίνεται για παράδειγμα στη σύνθετη αναζήτηση).

2^{ο} Στάδιο: Από τη στιγμή που ο χρήστης επιλέξει τις λέξεις κλειδιά και υποβάλει το ερώτημα, αυτό τίθεται στη διάθεση του συστήματος για προ-επεξεργασία. Στο στάδιο αυτό, με βάση το ερώτημα που έχει θέσει ο χρήστης, αποκλείονται ή όχι κάποιες πληροφορίες.

3^{ο} Στάδιο: Από τη στιγμή που το ερώτημα που έχει θέσει ο χρήστης έχει προ-επεξεργαστεί, μετατίθεται προς εκτέλεση. Αυτό είναι το στάδιο, που στόχο έχει την αναζήτηση και ανεύρεση των σχετικών με το ερώτημα, κειμένων. Στο στάδιο αυτό εμπλέκεται και ο κατάλογος, ο οποίος προσδιορίζει εκείνα τα κείμενα τα οποία περιλαμβάνουν τις λέξεις κλειδιά που τίθενται σαν ερώτημα από τον χρήστη. Τα συστήματα ανάκτησης πληροφορίας συνήθως χρησιμοποιούν τον ανεστραμμένο κατάλογο (inverted index), που ως κύρια λειτουργία έχει την αντιστοίχιση της κάθε λέξης του ερωτήματος στα κείμενα που τη περιέχουν, καθώς και τις θέσεις των λέξεων μέσα στα κείμενα. Ο ανεστραμμένος κατάλογος χωρίζεται σε δύο τμήματα:

α) το λεξικό (lexicon), που περιλαμβάνει όλες τις λέξεις που υπάρχουν στα κείμενα και

β) τις λίστες εμφανίσεων (occurrence lists), που περιλαμβάνουν τη συχνότητα εμφάνισης των λέξεων μέσα στα κείμενα.

4^{ο} Στάδιο: Αφού γίνει ο προσδιορισμός των σχετικών κειμένων (με τη χρήση του καταλόγου), η διαδικασία οδηγείται στο στάδιο της βαθμολόγησης των κειμένων. Σε αυτό το σημείο, τα κείμενα βαθμολογούνται, λαμβάνοντας μια τιμή με βάση τη σχετικότητα που παρουσιάζουν με το ερώτημα που έχει θέσει ο χρήστης. Ο βαθμός σχετικότητας κυμαίνεται μεταξύ του 0 και του 1 (όπου 0 = μη σχετικότητα και 1 = πλήρης σχετικότητα) και μπορεί να εκφραστεί και σε ποσοστό. Η μέθοδος της βαθμολόγησης προσδιορίζεται από το μοντέλο ανάκτησης που χρησιμοποιείται από το σύστημα, καθώς δεν επιτρέπουν όλα τα μοντέλα τον προσδιορισμό του βαθμού σχετικότητας.

$5^{ο}$ Στάδιο: Στο τελευταίο στάδιο της διαδικασίας έχουμε τη παραγωγή των αποτελεσμάτων. Στο σημείο αυτό, τα κείμενα που έχουν βαθμολογηθεί, επιστρέφονται στον χρήστη ταξινομημένα με φθίνουσα διάταξη.

Ένα πολύ σημαντικό σημείο που πρέπει να τονιστεί, είναι το γεγονός ότι συχνά παρατηρείται κάποια από τα κείμενα που επιστρέφονται από το σύστημα ανάκτησης πληροφορίας να μην σχετίζονται σε ικανοποιητικό βαθμό με το ερώτημα του χρήστη. Μια δημοφιλής μέθοδος που χρησιμοποιείται για την βελτιστοποίηση της ποιότητας των αποτελεσμάτων είναι η ανάδραση σχετικότητας (relevance feedback). Με αυτή τη μέθοδο, ο χρήστης είναι σε θέση να προσδιορίζει κάποια από τα κείμενα που επιστρέφονται ως πιο συναφή με το ερώτημα (σε σχέση με τα υπόλοιπα κείμενα) και το σύστημα να επαναπροσδιορίσει τις απαντήσεις που θα επιστραφούν με βάση πλέον το επαναπροσδιορισμένο ερώτημα του χρήστη.

4.3. Μετρικές για την αξιολόγηση της αποδοτικότητας των συστημάτων ανάκτησης πληροφορίας

4.3.1. Μη κατανεμημένη ανάκτηση πληροφορίας

Για την αξιολόγηση της αποδοτικότητας των συστημάτων ανάκτησης πληροφορίας, έχουν προταθεί πολλές διαφορετικές τεχνικές μετρήσεων. Οι τεχνικές αυτές χρειάζονται μια συλλογή αρχείων κι ένα ερώτημα. Οι μετρικές που περιγράφονται ακολούθως θεωρούν ως βασική υπόθεση ότι κάθε κείμενο μπορεί να είναι σχετικό ή μη σχετικό προς ένα συγκεκριμένο ερώτημα.

Οι δύο πιο βασικές και γνωστές μετρικές που χρησιμοποιούνται για την αξιολόγηση της αποδοτικότητας της ανάκτησης πληροφορίας είναι η ακρίβεια (precision) και η ανάκληση (recall). Για την απόδοση των ακόλουθων αλγορίθμων θεωρείται η υπόθεση ότι ένα σύστημα ανάκτησης πληροφορίας επιστρέφει ένα σύνολο κειμένων για ένα ερώτημα (στη συνέχεια η θεωρία θα επεκταθεί στην ταξινόμηση των ανακτημένων αντικειμένων).

Ακρίβεια ***(Precision)***

Ως ακρίβεια ορίζεται το κλάσμα των ανακτημένων αντικειμένων που είναι σχετικά με το ερώτημα, προς το συνολικό αριθμό των ανακτημένων αντικειμένων

$$Precision = \frac{relevant\ items\ retrieved}{retrieved\ items}$$

Η ακρίβεια υπολογίζεται βάση του συνολικού πλήθους των ανακτημένων αρχείων, αλλά μπορεί επίσης να υπολογιστεί και για μια προκαθορισμένη κατάταξη, λαμβάνοντας υπόψη μόνο εκείνα τα αποτελέσματα που επιστρέφονται από το σύστημα στη κορυφή της λίστας. Αυτή η μετρική καλείται «ακρίβεια στο n».

Ανάκληση ***(Recall)***

Ως ανάκληση ορίζεται το κλάσμα των σχετικών αντικειμένων με το ερώτημα που επιτυχώς ανακτώνται, προς τον συνολικό αριθμό των σχετικών αντικειμένων

$$Recall = \frac{relevant\ items\ retrieved}{relevant\ items}$$

Οι ανωτέρω έννοιες, μπορούν να γίνουν καλύτερα κατανοητές με τη βοήθεια του ακόλουθου πίνακα όπως αυτός παρουσιάζεται στο [12]:

	Relevant	Non relevant
Retrieved	True positives (tp)	False positives (fp)
Non retrieved	False negatives (fn)	True negatives (tn)

Με βάση τον ανωτέρω πίνακα, η ακρίβεια και η ανάκληση μπορούν να προσδιοριστούν από τις ακόλουθες σχέσεις:

$$Ακρίβεια: P = \frac{tp}{tp + fp}$$

$$Ανάκληση: R = \frac{tp}{tp + fn}$$

Μια εναλλακτική που προκύπτει με βάση τον παραπάνω πίνακα για την αξιολόγηση της ακρίβειας της ανάκτησης πληροφορίας του συστήματος, είναι:

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

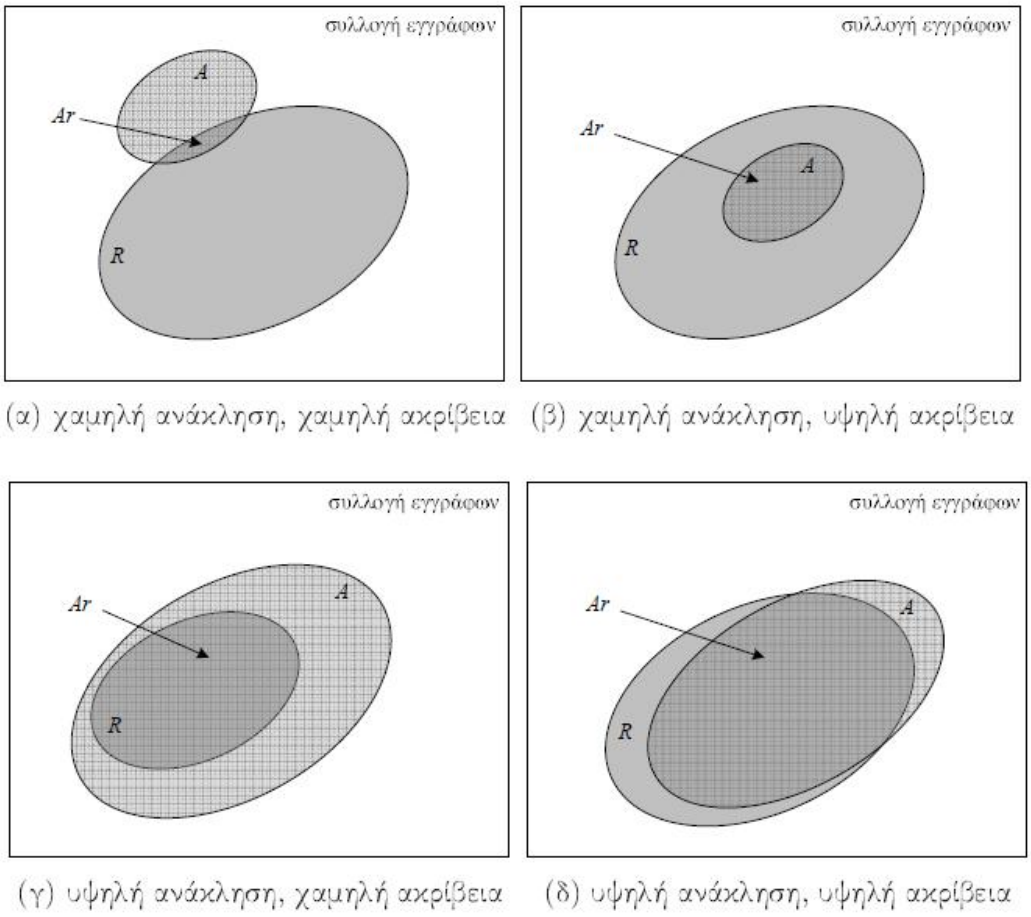
Αυτό είναι εύκολα κατανοητό, αν ληφθεί υπόψη ότι υπάρχουν δύο κλάσεις (η σχετικότητα και η μη σχετικότητα), και υπάρχει ένα σύστημα ανάκτησης πληροφορίας το οποίο μπορεί να θεωρηθεί ως ταξινομητής δύο κλάσεων που στόχο έχει τη κατηγοριοποίηση των αντικειμένων με βάση αυτές τις κλάσεις.

Από τα όσα αναφέρθηκαν παραπάνω, είναι εύκολο να συμπεράνει κανείς ότι όσο μεγαλύτερο είναι το ποσοστό στα δύο αυτά μεγέθη, τόσο αποτελεσματικότερο είναι το σύστημα ανάκτησης πληροφορίας για ένα ερώτημα. Στα διαγράμματα Venn που ακολουθούν, απεικονίζονται τέσσερεις διαφορετικές καταστάσεις, αντιπροσωπευτικές για τον συνδυασμό μεταξύ ανάκλησης και ακρίβειας και των μετρήσεων τους σ' ένα σύστημα, όπου:

A: Το σύνολο των κειμένων που ανακτώνται

R: Το σύνολο των σχετικών κειμένων

Ar: Το σύνολο των σχετικών κειμένων που ανακτώνται, δηλαδή η τομή του A με το R ($A \cap R$)



Εικόνα 6 - Συνδυασμοί ανάκλησης-ακρίβειας[21]

Ωστόσο η ακρίβεια (accuracy) δεν είναι η βέλτιστη μετρική για προβλήματα ανάκτησης πληροφορίας, καθώς συνήθως η κατανομή των δεδομένων είναι ασύμμετρη (συνήθως το 99.9% των κειμένων ανήκουν στη κατηγορία των μη σχετικών δεδομένων).

Μια άλλη τεχνική που συνδυάζει την ακρίβεια (precision) και την ανάκληση (recall) είναι η μετρική F (F measure), η οποία είναι ο σταθμισμένος αρμονικός μέσος (weighted harmonic mean) της ακρίβειας και της ανάκλησης.

[21] Πηγή: http://delab.csd.auth.gr/courses/c_ir/irbook.pdf

$$F = \frac{1}{a\frac{1}{P} + (1-a)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

όπου

$$\beta^2 = \frac{1-\alpha}{\alpha}, \qquad \alpha \in [0,1], \qquad \beta^2 \in [0,\infty]$$

Η προκαθορισμένη ισόρροπη μετρική F (η οποία είναι ευρέως γνωστή ως F_1), σταθμίζει εξίσου την ακρίβεια και την ανάκληση. Αυτό πρακτικά σημαίνει ότι $\alpha = 1/2$ και $\beta = 1$. Έτσι λοιπόν η ανωτέρω εξίσωση που προσδιορίζει την F παίρνει τη μορφή:

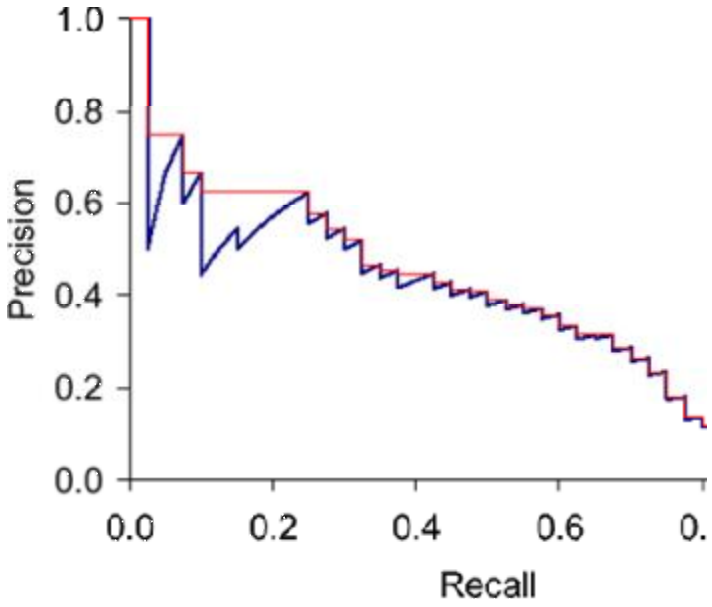
$$F_1 = \frac{2PR}{P+R}$$

Στην περίπτωση που $\beta < 1$ δίνεται έμφαση στην ακρίβεια (precision), ενώ όταν $\beta > 1$ δίνεται έμφαση στην ανάκληση.

4.3.2. **Κατανεμημένη ανάκτηση πληροφορίας**

Οι μετρικές που εξετάστηκαν νωρίτερα (precision, recall, F) υπολογίζονται χρησιμοποιώντας μη ταξινομημένα σύνολα κειμένων. Για τον υπολογισμό των κατανεμημένων αποτελεσμάτων που ανακτώνται από τις μηχανές αναζήτησης πρέπει να επεκταθεί η υπάρχουσα θεωρία των μετρικών ή να αναπτυχθούν νέες.

Σε ένα κατανεμημένο πλαίσιο ανάκτησης, τα κατάλληλα σύνολα ανακτημένων κειμένων δίνονται, υπό φυσιολογικές συνθήκες, από τα *k* ανακτημένα κείμενα. Για κάθε τέτοιο σύνολο, οι τιμές της ακρίβειας και της ανάκλησης μπορούν να αναπαρασταθούν σε ένα γράφημα που θα έχει για παράδειγμα την ακόλουθη μορφή [12]:



Εικόνα 7 - *Γράφημα ακρίβειας/ανάκλησης.*

Αν το $(k+1)^{th}$ ανακτημένο κείμενο δεν είναι σχετικό, τότε η ανάκληση είναι η ίδια όπως για τα k υψηλότερα στη λίστα κείμενα, αλλά η ακρίβεια είναι μειωμένη. Από την άλλη, αν είναι σχετικό, τότε τόσο η ακρίβεια όσο και η ανάκληση αυξάνονται.

Πολλές φορές είναι χρήσιμο να απαλείψουμε αυτές τις αυξομειώσεις που παρατηρούνται στη καμπύλη του σχήματος 3.3. Για να γίνει αυτό, χρησιμοποιείται η τεχνική της παρεμβολής (interpolation).

Η ακρίβεια με παρεμβολή (P_{interp}) σε ένα συγκεκριμένο επίπεδο ανάκλησης r ορίζεται ως η υψηλότερη ακρίβεια που υπολογίζεται για οποιοδήποτε επίπεδο ανάκλησης $r' \geq r$. Ισχύει δηλαδή η σχέση:

$$P_{interp}(r) = \max_{r' \geq r} p(r')$$

Στο πρόσφατο παρελθόν αναπτύχθηκαν και άλλες μετρικές. Μια από τις πιο γνωστές είναι η MAP (Mean Average Precision), η οποία μετράει τη ποιότητα στα διάφορα επίπεδα ανάκλησης. Επιπλέον, μεταξύ των μετρικών αξιολόγησης, η MAP

χαρακτηρίζεται από ιδιαίτερα μεγάλη ευστάθεια. Η Μέση Ακρίβεια (Average Precision) είναι η μέση τιμή της ακριβούς τιμής που προκύπτει για το σύνολο των k καλύτερων κειμένων μετά από κάθε σχετικό κείμενο που ανακτάται. Αυτό ισχύει εάν το σύνολο των σχετικών κειμένων για μια πληροφοριακή ανάγκη $q_j \in Q$ είναι $\{d_1, d_2, \dots, d_{m_j}\}$ και R_{jk} είναι το σύνολο των κατανεμημένων ανακτημένων αποτελεσμάτων από τα καλύτερα αποτελέσματα μέχρι να επιστραφεί το κείμενο d_k.

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk})$$

Στη περίπτωση που ένα σχετικό κείμενο δεν ανακτάται, η τιμή της ακρίβειας στην ανωτέρω εξίσωση είναι μηδενική. Για μια πληροφοριακή ανάγκη, η μέση ακρίβεια προσεγγίζει τη περιοχή κάτω από τη καμπύλη ακρίβειας-ανάκλησης και έτσι η μέθοδος MAP είναι περίπου η μέση περιοχή κάτω από τη καμπύλη ακρίβειας-ανάκλησης για ένα σύνολο ερωτημάτων.

Η μετρική MAP έχει το πλεονέκτημα ότι δεν απαιτεί καμία εκτίμηση για το μέγεθος του συνόλου των σχετικών κειμένων, αλλά έχει και τα μειονεκτήματα ότι παρουσιάζει τη μικρότερη ευστάθεια σε σχέση με τις περισσότερο χρησιμοποιούμενες μετρικές, καθώς και ότι δεν εκφράζει ορθά το μέσο όρο, αφού ο συνολικός αριθμός των σχετικών κειμένων για ένα ερώτημα έχει δυνατή επιρροή στην ακρίβεια στο *k*.

Μια εναλλακτική μετρική που απαλείφει το πρόβλημα που παρουσιάσει η μετρική MAP, είναι η ακρίβεια-R. Αυτό που χρειάζεται, είναι ένα σύνολο γνωστών σχετικών κειμένων Rel, από τα οποία υπολογίζεται η ακρίβεια στα ανώτερα ***Rel*** κείμενα που επιστρέφονται. Η ακρίβεια-R, προσαρμόζεται στο μέγεθος των συνόλων των σχετικών κειμένων. Το τέλειο σύστημα μπορεί να βαθμολογηθεί με 1 για κάθε ερώτημα.

Μια άλλη τεχνική που χρησιμοποιείται για την αξιολόγηση των συστημάτων ανάκτησης πληροφορίας είναι η καμπύλη ROC (Receiver Operating Characteristics). Η καμπύλη ROC αναπαριστά το αληθές θετικό ρυθμό (ευαισθησία) προς τον ψευδή θετικό ρυθμό (1

- εξειδίκευση). Στο συγκεκριμένο υπόδειγμα η ευαισθησία έχει την έννοια της ανάκλησης και δίνεται από τη σχέση:

$$sensitivity - \frac{fp}{fp + tn}$$

Αντίστοιχα, η εξειδίκευση για ένα σύνολο μη κατανεμημένων αποτελεσμάτων δίνεται από τη σχέση:

$$specificity = \frac{tn}{fp + tn}$$

Τέλος μια άλλη μετρική που δείχνει να υιοθετείται από όλο και περισσότερους ερευνητές είναι η NDCG (Normalized Discounted Cumulative Gain). Υπολογίζεται βάση κάποιων k ανώτερων αποτελεσμάτων αναζήτησης. Για ένα σύνολο ερωτημάτων Q, όπου R(j,d) είναι ο βαθμός σχετικότητας που δίνεται στο κείμενο d για το ερώτημα j, τότε:

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)} - 1}{\log_2(1+m)}$$

5. ΜΕΘΟΔΟΙ ΙΣΤΟΣΥΛΛΟΓΗΣ ΤΟΥ ΚΡΥΜΜΕΝΟΥ ΙΣΤΟΥ

Στις ενότητες που προηγήθηκαν, αναλύθηκε η έννοια του κρυμμένου ιστού καθώς και τα ιδιαίτερα χαρακτηριστικά του γνωρίσματα. Η πρόσβαση στον μεγάλο όγκο πληροφοριών του παγκόσμιου ιστού είναι συνήθως εφικτή μέσω διεπαφών αναζήτησης, στις οποίες ο χρήστης εισάγει ένα σύνολο λέξεων κλειδιών. Όπως έχει ήδη αναφερθεί, δεν υπάρχουν στατικοί σύνδεσμοι για τις σελίδες στον κρυμμένο ιστό, γεγονός που καθιστά τις μηχανές αναζήτησης ανίσχυρες στη προσπάθεια τους να εντοπίσουν και να επιστρέψουν ως αποτέλεσμα τις σελίδες αυτές. Το πρόβλημα αυτό, έχει απασχολήσει πλήθος ερευνητών, καθώς οι σελίδες αυτές περιέχουν πολλές φορές υψηλής ποιότητας περιεχόμενο.

Στο κεφάλαιο αυτό, διαπραγματεύεται η μεθοδολογία συλλογής πληροφοριών για τις σελίδες του κρυμμένου ιστού. Η ανάπτυξη της μεθοδολογίας αυτής ανήκει στους A. Ntoulas, P. Zerfos και J. Cho, και αφορά τη κατασκευή ενός αποδοτικού και λειτουργικού ιστοσυλλέκτη κρυμμένου ιστού, ο οποίος θα είναι σε θέση να εντοπίζει και να «κατεβάζει» αυτόνομα σελίδες από τον κρυμμένο ιστό.

5.1. Το Πλαίσιο

5.1.1. Το μοντέλο βάσης δεδομένων του κρυμμένου ιστού

Ένας τρόπος κατηγοριοποίησης που θα μπορούσε να γίνει σε μια ιστοσελίδα του κρυμμένου ιστού, είναι σύμφωνα με τον τύπο πληροφοριών που διαθέτει. Με βάση αυτή τη διακριτοποίηση, ο δικτυακός τόπος μπορεί να χαρακτηριστεί είτε ως βάση δεδομένων κειμένου του κρυμμένου ιστού (textual database), είτε ως μια δομημένη βάση δεδομένων (structured database).

Μια βάση δεδομένων κειμένου είναι ένας δικτυακός τόπος το οποίο κυρίως περιέχει έγγραφα απλού κειμένου (plain-text). Καθώς τα έγγραφα απλού κειμένου δεν έχουν συνήθως ορθή δομή, οι περισσότερες βάσεις δεδομένων κειμένου, παρέχουν στους

χρήστες τους ένα απλό περιβάλλον αναζήτησης, στο οποίο οι χρήστες μπορούν απλά να πληκτρολογήσουν τις λέξεις κλειδιά που επιθυμούν σε ένα απλό πεδίο αναζήτησης.

5.1.2. **Ο γενικός αλγόριθμος** crawling **του κρυμμένου ιστού**

Με δεδομένο, ότι ο μόνος τρόπος για να προσπελαστούν οι σελίδες ενός δικτυακού τόπου του κρυμμένου ιστού είναι η φόρμα αναζήτησης που παρέχει, ένας ιστοσυλλέκτης κρυμμένου ιστού πρέπει να ακολουθήσει τα επόμενα τρία βήματα:

i. Να παράγει ένα ερώτημα και να το θέσει στον δικτυακό τόπο.
ii. Να λάβει τη σελίδα με τον πίνακα αποτελεσμάτων.
iii. Να επιλέξει της συνδέσεις ώστε να λάβει τις αντίστοιχες σελίδες.

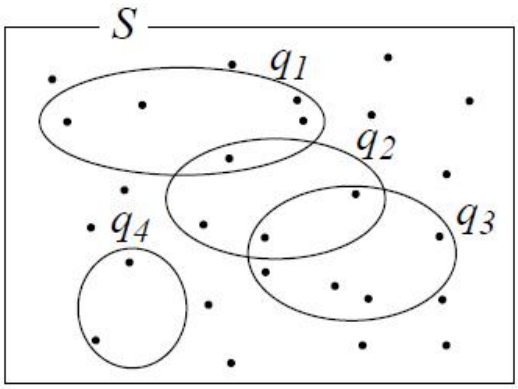
Ακολούθως, βλέπουμε τον γενικό αλγόριθμο crawling του κρυμμένου ιστού

```
(1) while ( there are available resources ) do
      // select a term to send to the site
(2)     q_i = SelectTerm()
      // send query and acquire result index page
(3)     R(q_i) = QueryWebSite( q_i )
      // download the pages of interest
(4)     Download( R(q_i) )
(5) done
```

Με βάση τον αλγόριθμο που παρουσιάζεται ανωτέρω, είναι εμφανές ότι η πιο κρίσιμη επιλογή που έχει να κάνει ο αλγόριθμος είναι πιο ερώτημα θα θέσει. Όπως είναι προφανές όσο πιο επιτυχημένα είναι τα ερωτήματα που θα θέσει ο ιστοσυλλέκτης, τόσο πιο πολλές σελίδες που πληρούν τα κριτήρια που επιθυμεί ο χρήστης θα επιστραφούν.

5.1.3. **Γενική περιγραφή του προβλήματος επιλογής**

Για να αποδοθεί καλύτερα η έννοια του προβλήματος της επιλογής (που σχολιάστηκε νωρίτερα), θα χρησιμοποιηθεί η ακόλουθη εικόνα.



Εικόνα 8 - Ο φορμαλισμός του προβλήματος της βέλτιστης επιλογής ερωτήματος με βάση τη θεωρία συνόλων

Υποθέτουμε ότι ο ιστοσυλλέκτης λαμβάνει σελίδες από έναν δικτυακό τόπο, το οποίο έχει ένα σύνολο σελίδων S. Η κάθε σελίδα μέσα στον δικτυακό τόπο αναπαριστάται με μια τελεία. Επίσης, κάθε δυνητικό ερώτημα q_i στον δικτυακό τόπο αναπαριστάται ως υποσύνολο του S, όπου το κάθε ερώτημα q_i επιστρέφει ένα σύνολο σελίδων. Κάθε υποσύνολο χαρακτηρίζεται από ένα «βάρος» το οποίο υποδηλώνει το κόστος από την χρησιμοποίηση του κάθε ερωτήματος.

Με βάση τα παραπάνω, η βέλτιστη λύση του προβλήματος είναι να βρεθούν εκείνα τα ερωτήματα q_i (τα υποσύνολα στο S), τα οποία επιστρέφουν όσες το δυνατόν περισσότερες σελίδες που ενδιαφέρουν τον χρήστη (τελείες στα υποσύνολα), ενώ ταυτόχρονα επιτυγχάνεται το μικρότερο δυνατό κόστος.

Το κόστος που αναφέρθηκε στις ανωτέρω παραγράφους, μπορεί να μετριέται σε χρόνο, στην ευρυζωνικότητα του δικτύου, στον αριθμό των αλληλεπιδράσεων με τον δικτυακό τόπο ή ακόμα μπορεί να είναι μια συνάρτηση όλων αυτών. Το συνολικό κόστος που προκύπτει από ένα ερώτημα q_i, δίνεται από τη σχέση:

$$Cost(q_i) = c_q + c_r P(q_i) + c_d P(q_i)$$

όπου:

$P(q_i)$: το σύνολο των σελίδων που θα επιστραφούν ένα τεθεί το ερώτημα q_i,

c_q : το καθορισμένο κόστος από την υποβολή του ερωτήματος q_i,

c_r : το κόστος για τη λήψη της σελίδας που περιέχει τα αποτελέσματα της αναζήτησης και

c_d το καθορισμένο κόστος από τη λήψη ενός εγγράφου που πληρεί τα κριτήρια αναζήτησης.

Σε κάποιες περιπτώσεις, κάποια από τα έγγραφα που επιστρέφονται από το ερώτημα q_i, μπορεί να έχουν ήδη ληφθεί από προηγούμενα ερωτήματα. Σε αυτές τις περιπτώσεις ο ιστοσυλλέκτης μπορεί να παραλείψει τη λήψη των εγγράφων αυτών. Κάτι τέτοιο σημαίνει μεταβολή της συνάρτησης κόστους, της οποίας πλέον η έκφραση είναι:

$$Cost(q_i) = c_q + c_r P(q_i) + c_d P_{new}(q_i)$$

όπου το $P_{new}(q_i)$ αναφέρεται στον αριθμό των νέων σελίδων που επιστρέφονται από την υποβολή του ερωτήματος και οι οποίες δεν είχαν συμπεριληφθεί σε προηγούμενες αναζητήσεις.

Σύμφωνα με τα όσα αναφέρθηκαν νωρίτερα ο φορμαλισμός του στόχου μπορεί να διατυπωθεί ως εξής:

Να βρεθεί το σύνολο των ερωτημάτων $q_1, \ldots, q_n$ *που μεγιστοποιούν τη συνάρτηση* $P(q_1 \; V \; \ldots \; V \; q_{i-1} \; V \; q_{i-1})$*, σύμφωνα με τον περιορισμό:*

$$\sum_{i=1}^{n} Cost(q_i) \leq t$$

όπου t είναι ο μέγιστος αριθμός πηγών λήψης που έχει ο ιστοσυλλέκτης.

5.2. **Επιλογή των λέξεων-κλειδιών**

Με βάση ότι ο κύριος στόχος είναι η λήψη όσο το δυνατόν περισσότερων μοναδικών εγγράφων από μια βάση δεδομένων κειμένων, κάποιος μπορεί να ακολουθηθεί κάποια εκ των τριών ακόλουθων επιλογών:

Τυχαία (Random): Επιλέγονται τυχαίες λέξεις κλειδιά για να χρησιμοποιηθούν στη βάση δεδομένων με την ελπίδα ότι ένα τυχαίο ερώτημα θα επιστρέψει ένα λογικό αριθμό εγγράφων που πληρούν τα κριτήρια αναζήτησης.

Γενική Συχνότητα (Generic-frequency): Ανάλυση ενός γενικού εγγράφου που συλλέγεται από κάπου αλλού (πχ από τον ιστό) και εύρεση της γενικής συχνότητας κατανομής της κάθε λέξης κλειδί. Βάση αυτής της γενικής κατανομής, ξεκινάμε από τη λέξη κλειδί με τη μεγαλύτερη συχνότητα εμφάνισης, την εφαρμόζουμε στη βάση δεδομένων του κρυμμένου ιστού και εξάγουμε τα αποτελέσματα. Έπειτα, συνεχίζουμε τη διαδικασία με τη δεύτερη λέξη κλειδί με τη μεγαλύτερη συχνότητα εμφάνισης και ακλουθούμε την ίδια διαδικασία έως ότου εξαντληθούν όλες οι πηγές λήψης.

Προσαρμοζόμενη (Adaptive): Ανάλυση των εγγράφων που επεστράφησαν από τα προηγούμενα ερωτήματα στη βάση δεδομένων του κρυμμένου ιστού και εκτίμηση της λέξης κλειδιού που είναι πιο πιθανό να επιστρέψει τα περισσότερα έγγραφα. Βάση αυτής της ανάλυσης εντοπίζεται το πιο ελπιδοφόρο ερώτημα και επαναλαμβάνεται η διαδικασία.

5.2.1. **Υπολογισμός του αριθμού των σελίδων που πληρούν τα κριτήρια αναζήτησης**

Για να βρεθεί το πλέον ελπιδοφόρο (από πλευράς απόδοσης) ερώτημα, χρειάζεται να γίνει μια εκτίμηση για το πόσα νέα έγγραφα θα ληφθούν εάν τεθεί το ερώτημα q_i ως επόμενο ερώτημα. Υποθέτοντας ότι έχουν τεθεί τα ερωτήματα q_1, …, q_{i-1} χρειάζεται να γίνει η εκτίμηση $P(q_1 \text{ V } \dots \text{ V } q_{i-1} \text{ V } q_{i-1})$ για κάθε δυνητικό επόμενο q_i ερώτημα και να γίνει σύγκριση αυτής της τιμής.

Για την εκτίμηση του $P(q_i)$, μπορούν να υπάρξουν διάφοροι τρόποι, συμπεριλαμβανομένων του ακόλουθου:

i. Ανεξάρτητος εκτιμητής (***Independence estimator***): Γίνεται χρήση του τύπου $P(q_i)= P(q_i \mid q_1 \text{ V } \dots \text{ V } q_{i-1} \text{ V } q_{i-1})$, όπου θεωρείται ότι η εμφάνιση του όρου q_i, είναι ανεξάρτητη από τους όρους q_1, …, q_{i-1}.
ii. Εκτιμητής Zipf (***Zipf estimator***): Υπολογίζεται η συχνότητα με την οποία εμφανίζεται ένας όρος μέσα σε μια συλλογή κειμένου και η οποία προκύπτει από την εξίσωση:

$$f = \alpha(r + \beta)^{-\gamma}$$

όπου:

r: η κατάταξη του όρου βάση της συχνότητας (τη θέση 1 θα έχει ο όρος με την μεγαλύτερη συχνότητα)

α, β και γ: σταθερές εξαρτώμενες από τη συλλογή κειμένου, που υπολογίζονται με τη χρήση του (i).

5.2.2. **Αλγόριθμος επιλογής ερωτήματος**

Όπως ήδη έχει αναφερθεί, ο στόχος του ιστοσυλλέκτη κρυμμένου ιστού είναι η λήψη του μέγιστου αριθμού μοναδικών εγγράφων, χρησιμοποιώντας τους περιορισμούς του πηγών λήψης. Με βάση αυτό, ο ιστοσυλλέκτης πρέπει να λάβει υπόψη του δύο συντελεστές:

- τον αριθμό των νέων εγγράφων που θα ανακτηθούν από το ερώτημα q_i, και
- το κόστος από τη χρησιμοποίηση του ερωτήματος q_i.

Όπως είναι προφανές, αν δύο ερωτήματα επιστρέφουν το ίδιο πλήθος νέων εγγράφων αλλά το ένα έχει μεγαλύτερο κόστος από το άλλο, τότε επιλέγεται το ερώτημα με το μικρότερο κόστος.

Με βάση αυτό το σκεπτικό, ο ιστοσυλλέκτης κρυμμένου ιστού μπορεί να ποσοτικοποιήσει την αποδοτικότητα ενός ερωτήματος και κατ' επέκταση την επιλογή του κατάλληλου ερωτήματος q_i, με τη χρήση του ακόλουθου αλγορίθμου αποδοτικότητας:

$$Efficiency(q_i) = \frac{P_{new}(q_i)}{Cost(q_i)}$$

όπου:

$P_{new}(q_i)$: το πλήθος των νέων εγγράφων που επιστρέφονται χρησιμοποιώντας το ερώτημα q_i και

$Cost(q_i)$: το κόστος που προκύπτει από τη χρήση του ερωτήματος q_i.

Αυτό λοιπόν που επιστρέφει ο αλγόριθμος αποδοτικότητας είναι το πλήθος των νέων εγγράφων που επιστρέφονται για κάθε μονάδα κόστους, και μπορεί να χρησιμοποιηθεί για να υποδείξει το πόσο καλά «ξοδεύονται» οι πόροι μας όταν θέτουμε το ερώτημα q_i. Με τη χρήση λοιπόν του εν λόγω αλγορίθμου ο ιστοσυλλέκτης μπορεί να υπολογίσει τον βαθμό αποδοτικότητας κάθε ερωτήματος q_i και να επιλέξει αυτό με την υψηλότερη τιμή (η οποία προκύπτει από την αναλογία). Με τον τρόπο αυτό, ο ιστοσυλλέκτης μπορεί τελικώς να λάβει το μέγιστο αριθμό νέων εγγράφων. Στον αλγόριθμο που ακολουθεί, απεικονίζεται η δομή της συνάρτησης επιλογής ερωτήματος όπου χρησιμοποιεί τη λογική της αποδοτικότητας, που αναλύθηκε νωρίτερα. Ο αλγόριθμος παίρνει μια «άπληστη» προσέγγιση (*greedy approach*) και προσπαθεί να μεγιστοποιήσει το όφελος σε κάθε βήμα.

Parameters:
T: The list of potential query keywords
Procedure
(1) Foreach t_k in T do
(2) Estimate Efficiency$(t_k) = \frac{P_{new}(t_k)}{Cost(t_k)}$
(3) done
(4) return t_k with maximum Efficiency(t_k)

5.2.3. **Βελτιστοποιημένη μέθοδος μέτρησης απόδοσης των ερωτημάτων**

Για τον υπολογισμό της αποδοτικότητας των ερωτημάτων, δείξαμε ότι χρειάζεται να υπολογιστεί το $P(q_i|q_1 \text{ V } \dots \text{ V } q_{i-1})$ για κάθε δυνητικό q_i ερώτημα. Ο υπολογισμός αυτός μπορεί να αποδειχτεί ιδιαίτερα χρονοβόρος, γι' αυτό στο σημείο αυτό παρουσιάζεται ένας αποδοτικός τρόπος μέτρησης του $P(q_i|q_1 \text{ V } \dots \text{ V } q_{i-1})$ με τη χρήση και διατήρηση ενός μικρού πίνακα που καλείται στατιστικός πίνακας ερωτήματος (***query statistics table***).

Η κεντρική ιδέα του στατιστικού πίνακα ερωτήματος είναι ότι το $P(q_i|q_1 \text{ V } \dots \text{ V } q_{i-1})$ μπορεί να υπολογιστεί, μετρώντας πόσες φορές εμφανίζεται η λήξη κλειδί στα έγγραφα που λαμβάνονται από τα $q_1, \dots, q_{i-1}$. Για να γίνει αυτό καλύτερα αντιληπτό, παρουσιάζεται το ακόλουθο παράδειγμα:

Έστω ότι μέχρι στιγμής έχουμε λάβει 50 έγγραφα στα οποία ο όρος «model» εμφανίζεται 10 φορές. Με βάση αυτά τα δεδομένα υπολογίζουμε ότι $P(\text{model} \mid q_1 \text{ V } \dots \text{ V } q_{i-1}) = 10/50 = 0.2$ (Πίνακας 1).

Term t_k	$N(t_k)$
model	10
computer	38
digital	50

Total pages: 50

Πίνακας 1: Αποτελέσματα έπειτα από $q_1, \dots, q_{i-1}$ ερωτήματα

Έστω τώρα ότι αποφασίζουμε να συνεχίσουμε την αναζήτηση χρησιμοποιώντας τον όρο «computer». Από τη καινούρια αναζήτηση λαμβάνονται 20 ακόμα νέες σελίδες. Από αυτές οι 12 περιέχουν τον όρο «model» και οι 18 τον όρο «disk» (Πίνακας 2).

Term t_k	$N(t_k)$
model	12
computer	20
disk	18
New pages: 20	

Πίνακας 2: Αποτελέσματα για το νέο ερώτημα q_i = computer

Με βάση τα αποτελέσματα που παρουσιάστηκαν, μπορούμε να ενημερώσουμε τον Πίνακα 1, προσθέτοντας απλά σε αυτόν τα δεδομένα του Πίνακα 2. Το αποτέλεσμα αυτής της πρόσθεσης παρουσιάζεται στον ακόλουθα πίνακα.

Term t_k	$N(t_k)$
model	10+12 = 22
computer	38+20 = 58
disk	0+18 = 18
digital	50+0 = 50
Total pages: 50 + 20 = 70	

Πίνακας 3: Αποτελέσματα έπειτα από q_1, …, q_i ερωτήματα

Σύμφωνα με το νέο πίνακα που προκύπτει, ο όρος «model» εμφανίζεται σε 22 έγγραφα από τις 70 συνολικά σελίδες που λάβαμε, δίνοντας έτσι $P(\text{model} \mid q_1 \text{ V} \dots \text{V } q_{i-1}) = 22/70 = 0.3$. Με αντίστοιχο τρόπο υπολογίζεται και για τους άλλους όρους.

5.2.4. **Δικτυακοί τόποι που περιορίζουν τον αριθμό των αποτελεσμάτων**

Σε πολλές περιπτώσεις όπου ένα ερώτημα ταιριάζει σε έναν μεγάλο αριθμό σελίδων, ο δικτυακός τόπος του κρυμμένου ιστού επιστρέφει μόνο ένα τμήμα αυτών των σελίδων. Ένα παράδειγμα περιορισμών είναι η αναζήτηση μέσω της ιστοσελίδας του Open Directory Project[22], το οποίο επιτρέπει στους χρήστες του να δουν μέχρι 10000 αποτελέσματα από μια αναζήτηση.

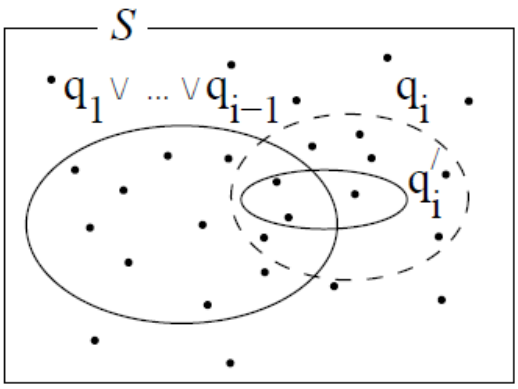
Εικόνα 9 - Αρχική σελίδα του Open Directory Project

Προφανώς, τέτοιου είδους περιορισμοί έχουν άμεση επίδραση στον ιστοσυλλέκτη κρυμμένου ιστού. Αρχικά, αφού γνωρίζουμε ότι μπορούμε να ανακτήσουμε ένα συγκεκριμένο αριθμό σελίδων για κάθε ερώτημα, ο ιστοσυλλέκτης πρέπει να θέσει περισσότερα ερωτήματα προκειμένου να λάβει όλες τις σελίδες. Επιπλέον, η μέθοδος επιλογής ερωτήματος που παρουσιάστηκε νωρίτερα υποθέτει ότι για κάθε δυνητικό ερώτημα q_i, μπορεί να βρει το $P(q_i|q_1 \text{ V } \dots \text{ V } q_{i-1})$, δηλαδή ότι για κάθε ερώτημα q_i

[22] http://www.dmoz.org/

μπορεί να βρεθεί το ποσοστό των εγγράφων σε ολόκληρη τη βάση δεδομένων κειμένου, τα οποία περιέχουν το ερώτημα – όρο q_i, με τουλάχιστον ένα από τα $q_1, \dots, q_{i-1}$. Σε κάθε περίπτωση, αν η βάση δεδομένων κειμένου επιστρέφει μόνο ένα μέρος των αποτελεσμάτων για κάθε $q_1, \dots, q_{i-1}$, τότε η τιμή $P(q_i|q_1 \text{ V } \dots \text{ V } q_{i-1})$ δεν μπορεί να είναι ακριβής και είναι πολύ πιθανό να επηρεάσει τη κρίση μας για την επιλογή του επόμενου ερωτήματος και κατ' επέκταση την αποδοτικότητα του ιστοσυλλέκτη. Παρόλα αυτά υπάρχει ο τρόπος να εκτιμηθεί η ορθή τιμή του $P(q_i|q_1 \text{ V } \dots \text{ V } q_{i-1})$ στη περίπτωση που μια ιστοσελίδα επιστρέφει μόνο ένα μέρος των αποτελεσμάτων. Ο τρόπος αυτός παρουσιάζεται στην επόμενη παράγραφο.

Υποθέτουμε ότι ο υπό εξέταση δικτυακός τόπος του κρυμμένου ιστού αναπαριστάται από το ορθογώνιο που εμφανίζεται στην ακόλουθη εικόνα και οι σελίδες του αναπαριστώνται από τις κουκκίδες μέσα σε αυτό.



Εικόνα 10 - Δικτυακός τόπος με περιορισμούς στα αποτελέσματα που επιστρέφει

Έστω ότι ήδη έχουμε θέσει τα ερωτήματα $q_1, \dots, q_{i-1}$, τα οποία επέστρεψαν έναν αριθμό αποτελεσμάτων μικρότερο από τον μέγιστο που επιτρέπει ο δικτυακός τόπος και τις οποίες σελίδες «κατεβάσαμε» (οι σελίδες που ελήφθησαν για τα $q_1, \dots, q_{i-1}$ ερωτήματα αναπαριστώνται από τον αριστερό κύκλο). Έστω ακόμα ότι θέτουμε το ερώτημα q_i στον δικτυακό τόπο, αλλά λόγω του περιορισμού που υπάρχει στην εμφάνιση αποτελεσμάτων, μας επιστρέφεται το σύνολο q_i' (δεξής μικρός κύκλος) αντί του συνόλου q_i (δεξής διακεκομμένος κύκλος).

Στο σημείο αυτό πρέπει να ενημερώσουμε τον στατιστικό πίνακα ερωτήματος, έτσι ώστε να έχει ακριβείς πληροφορίες για το επόμενο βήμα. Αυτό είναι, για κάθε δυνητικό ερώτημα q_{i+1} πρέπει να βρούμε το $P(q_{i+1}|q_1 \text{ V } \dots \text{ V } q_i)$, το οποίο δίνεται από την εξίσωση:

$$P(q_{i+1}|q_1 \vee \dots \vee q_i) = \frac{1}{P(q_1 \vee \dots \vee q_i)} \cdot [P(q_{i+1} \wedge (q_1 \vee \dots \vee q_{i-1})) + P(q_{i+1} \wedge q_i) - P(q_{i+1} \wedge q_i \wedge (q_1 \vee \dots \vee q_{i-1}))]$$

Στην ανωτέρω εξίσωση, η ποσότητα $P(q_1 \text{ V } \dots \text{ V } q_i)$ υπολογίζεται βάση της εξίσωσης για το $P(q_i)$, η οποία παρουσιάστηκε ανωτέρω. Αντιστοίχως, οι ποσότητες $P(q_{i+1} \wedge (q_1 \text{ V } \dots \text{ V } q_{i-1}))$ και $P(q_{i+1} \wedge q_i \wedge (q_1 \text{ V } \dots \text{ V } q_{i-1}))$, υπολογίζονται με τον άμεσο έλεγχο των εγγράφων που έχουν ληφθεί από τα ερωτήματα $q_1 \text{ V } \dots \text{ V } q_{i-1}$. Τέλος, για τον υπολογισμό του $P(q_{i+1} \wedge q_i)$, θεωρείται το q_i' ως ένα τυχαίο δείγμα του q_i και δίνεται η εξίσωση:

$$\frac{P(q_{i+1} \wedge q_i)}{P(q_{i+1} \wedge q_i')} = \frac{P(q_i)}{P(q_i')}$$

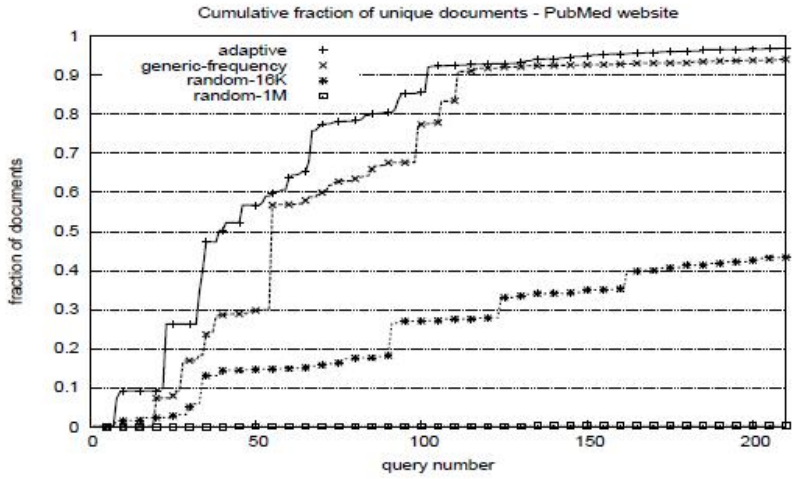
Από τη τελευταία εξίσωση, προκύπτει η ποσότητα $P(q_{i+1} \wedge q_i)$ και εν συνεχεία με αντικατάσταση στη πιο πάνω εξίσωση προκύπτει η ζητούμενη ποσότητα $P(q_{i+1}|q_1 \text{ V } \dots \text{ V } q_i)$.

5.3. **Πειραματική αξιολόγηση μεθοδολογίας**

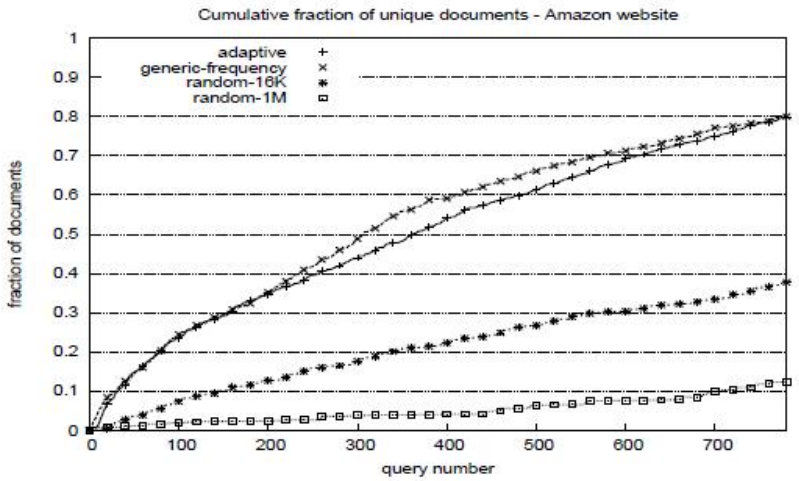
Στην ενότητα αυτή, παρουσιάστηκε ο τρόπος κατασκευής ενός ιστοσυλλέκτης κρυμμένου ιστού που μπορεί αυτόματα να θέτει ερωτήματα σε έναν δικτυακό τόπο του κρυμμένου ιστού και να λαμβάνει σελίδες από αυτό.

Ένα από τα πρώτα πράγματα που μας απασχολεί είναι η εξέλιξη του μέτρου κάλυψης κατά την υποβολή ερωτημάτων σε ένα δικτυακό χώρο. Με άλλα λόγια, αυτό που μας

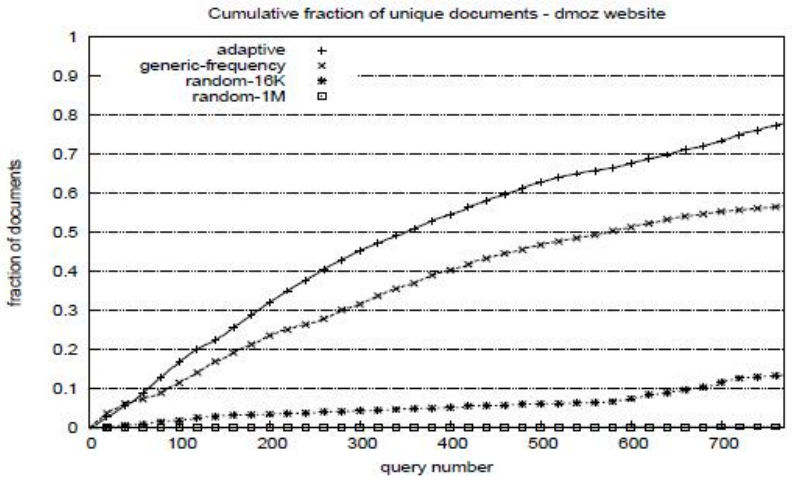
ενδιαφέρει είναι η αναλογία των εγγράφων που είναι αποθηκευμένα στο κρυμμένο ιστό και μπορούμε να τα λάβουμε/ «κατεβάσουμε», συνεχίζοντας παράλληλα την αναζήτηση για νέες λέξεις-κλειδιά, σύμφωνα με τις τρεις μεθοδολογίες που εξετάστηκαν νωρίτερα. Πρακτικά αυτό σημαίνει, ότι σύμφωνα και με τα όσα εξετάστηκαν και νωρίτερα, εξετάζουμε τη ποσότητα $P(q_1 \text{ V } \dots \text{ V } q_i)$, καθώς αυξάνεται το i. Στις εικόνες που ακολουθούν, απεικονίζεται η μετρική κάλυψης για κάθε μια από τις τρεις μεθοδολογίες εύρεσης λέξεων-κλειδιών, ως προς τον αριθμό των ερωτημάτων που τίθενται στις ιστοσελίδες και πιο συγκεκριμένα για τις σελίδες της PubMed (Εικόνα 11), της Amazon (Εικόνα 12) και της dmoz (Εικόνα 13):



Εικόνα 11 - Κάλυψη μεθοδολογιών για τη PubMed



Εικόνα 12 - Κάλυψη μεθοδολογιών για την Amazon

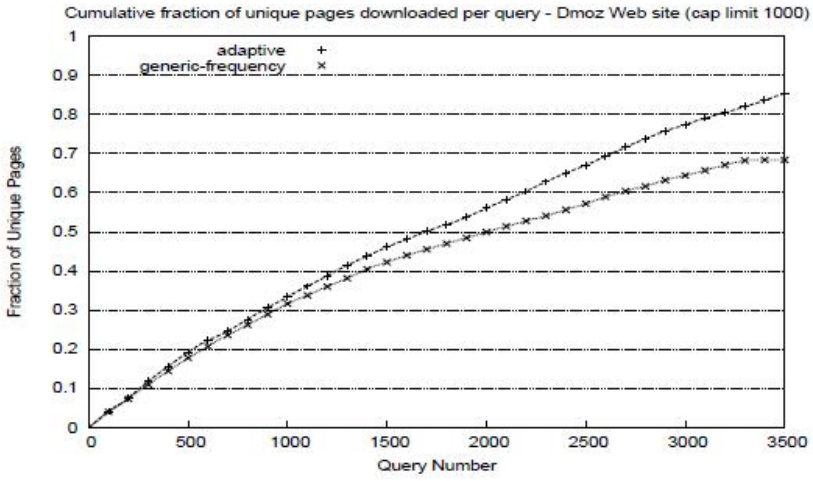


Εικόνα 13 - Κάλυψη μεθοδολογιών για την dmoz

Ένα γρήγορο συμπέρασμα, που εξάγεται από τα ανωτέρω γραφήματα, είναι ότι σε γενικές γραμμές οι μεθοδολογίες γενικής συχνότητας (generic-frequency) και προσαρμογής (adaptive) λειτουργούν αποδοτικότερα από τους τυχαίους αλγόριθμους επιλογής (random).

Όσον αφορά τη μεταξύ τους αξιολόγηση, των αλγορίθμων γενικής συχνότητας και προσαρμογής, παρατηρούμε ότι τα αποτελέσματα ποικίλουν. Ειδικότερα, παρατηρούμε ότι για τις σελίδες της Pubmed και της dmoz ο αλγόριθμος γενικής συχνότητας είναι αποδοτικότερος, ενώ για τη σελίδα της Amazon ο αλγόριθμος προσαρμογής εμφανίζεται αποδοτικότερος. Το συμπέρασμα που θα μπορούσε να εξαχθεί από εδώ είναι ότι σε γενικές γραμμές ο αλγόριθμος προσαρμογής αποδίδει καλύτερα όταν η σελίδα πραγματεύεται συγκεκριμένο θέμα. Παράδειγμα και πειραματική απόδειξη της τελευταίας πρότασης αποτελεί το γεγονός ότι, ο αλγόριθμος προσαρμογής απαιτεί 83 ερωτήματα για να λάβει περίπου το 80% των εγγράφων που είναι αποθηκευμένα στο δικτυακό τόπο της Pubmed, ενώ ο αλγόριθμος γενικής συχνότητας απαιτεί 106 ερωτήματα για να καλύψει το ίδιο ποσοστό.

Σε σχέση με τη τελευταία σύγκριση που έγινε, ιδιαίτερο ενδιαφέρον παρουσιάζει η αποδοτικότητα των δύο επικρατέστερων αλγορίθμων, σε συνάρτηση με τον περιορισμό που έχουν οι δικτυακοί τόποι στον αριθμό των αποτελεσμάτων που επιστρέφουν. Για το λόγο αυτό διερευνήθηκε η κάλυψη που επιφέρουν οι δύο προαναφερθέντες αλγόριθμοι κατά την ιστοσυλλογή από τη σελίδα της dmoz, θέτοντας ως περιορισμό 1.000 αποτελέσματα για κάθε ερώτημα (Εικόνα 14).



Εικόνα 14 - Κάλυψη μεθοδολογιών για την dmoz με περιορισμό αποτελεσμάτων

Σε σχέση με την Εικόνα 13 όπου ο περιορισμός των αποτελεσμάτων ανά ερώτημα ήταν στις 10.000, στην Εικόνα 14 ο περιορισμός αυτός κατέρχεται στα 1.000. Το αποτέλεσμα όσον αφορά το ποιος είναι πιο αποδοτικός αλγόριθμος μπορεί να μην αλλάζει, ωστόσο μια ιδιαίτερη χρήσιμη παρατήρηση που γίνεται, είναι το γεγονός ότι τόσο ο προσαρμοσμένος αλγόριθμος όσο και ο αλγόριθμος γενικής συχνότητας, πρέπει να υποβάλουν μεγαλύτερο αριθμό ερωτημάτων για να επιτύχουν το ίδιο ποσοστό κάλυψης με την περίπτωση άνευ περιορισμού.

5.4. **Συμπεράσματα μεθοδολογίας**

Στην ενότητα αυτή, παρουσιάστηκε ο τρόπος κατασκευής ενός ιστοσυλλέκτης κρυμμένου ιστού που μπορεί αυτόματα να θέτει ερωτήματα σε έναν δικτυακό τόπο του κρυμμένου ιστού και να λαμβάνει σελίδες από αυτό. Με βάση τα πειράματα που πραγματοποιήθηκαν από τους A. Ntoulas, κ.α., σε πραγματικούς δικτυακούς τόπους του κρυμμένου ιστού διαφάνηκε η αποδοτικότητα της ανωτέρω μεθοδολογίας. Αξίζει να αναφερθεί ότι μέσω αυτής της μεθοδολογίας, υπήρξαν πειράματα που επέτρεψαν τη λήψη περίπου του 90% των εγγράφων από δικτυακό τόπο του κρυμμένου ιστού (το οποίο περιέχει 14.000.000 έγγραφα), έπειτα από περίπου 100 ερωτήματα.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη εργασία που παρουσιάστηκε, έγινε κατανοητή η έννοια του κρυμμένου ιστού μέσα από τη παράθεση εννοιών και απόψεων. Επίσης, έγινε μια παρουσίαση των κύριων χαρακτηριστικών του, τα οποία συνοψίζονται ως εξής:

- Ο όγκος πληροφοριών στον κρυμμένο ιστό υπολογίζεται περίπου 500 φορές μεγαλύτερος από αυτόν του επιφανειακού ιστού.
- Ο κρυμμένος ιστός περιλαμβάνει 7.500 terabytes όγκο δεδομένων, ενώ ο επιφανειακός ιστό μόλις 19.
- Ο ρυθμός ανάπτυξης του κρυμμένου ιστού είναι πολύ υψηλότερος σε σχέση με τον ρυθμό ανάπτυξης του επιφανειακού ιστού.
- Υπολογίζεται ότι αυτή τη στιγμή υπάρχουν περισσότερα από 2.000.000 sites στον κρυμμένο ιστό.
- Τα ανεξάρτητα έγγραφα στον κρυμμένο ιστό εκτιμώνται περίπου στα 550.000.000, τη στιγμή που ο επιφανειακός ιστός περιέχει περίπου 1.000.000.
- Οι περισσότερες πληροφορίες στον κρυμμένο ιστό, διατηρούνται από ακαδημαϊκά ιδρύματα και ερευνητικούς οργανισμούς, γι’ αυτό το λόγο άλλωστε πολλοί ερευνητές του αντικειμένου υποστηρίζουν ότι η ποιότητα των πληροφοριών που βρίσκεται στον κρυμμένο ιστό είναι πολύ υψηλότερη από αυτή που βρίσκεται στον επιφανειακό.
- Το 95% των πληροφοριών που βρίσκονται στον κρυμμένο ιστό είναι προσβάσιμες, χωρίς την απαίτηση εγγραφής ή χρηματικού αντιτίμου.
- Το περιεχόμενο του κρυμμένου ιστού έχει μεγαλύτερη συνάφεια με την απαιτούμενη πληροφορία, σε σχέση με τον επιφανειακό ιστό.
- Περίπου το 55% του περιεχομένου του κρυμμένου ιστού είναι αποθηκευμένο σε βάσεις δεδομένων με συγκεκριμένη θεματική ενότητα.

Εν συνεχεία αναλύθηκε το είδος του περιεχομένου του κρυμμένου ιστού, καθώς και το ποσοστό που καταλαμβάνει η κάθε κατηγορία ως προς το συνολικό μέγεθος του κρυμμένου ιστού.

Επίσης, αναδείχθηκαν οι αδυναμίες των παραδοσιακών μηχανών αναζήτησης για την ανάκτηση δεδομένων από τον κρυμμένο ιστό, ενώ τέλος παρουσιάστηκε μια μεθοδολογία ιστοσυλλογής που αφορά τον κρυμμένο ιστό.

Βιβλιογραφία

1. K. Bharat and A. Broder, "A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines," paper presented at the Seventh International World Wide Web Conference, Brisbane, Australia, April 14-18, 1998.
2. S. Lawrence and C.L. Giles, "Searching the World Wide Web," Science 80:98-100, April 3, 1998.
3. Bergman, Michael K. White Paper: The Deep Web: Surfacing Hidden Value, Volume 7, Issue 1, August, 2001
4. G. Rathinasabapathy, Invisible Web and Knowledge discovery tools: A Study, 5th International CALIBER -2007, Panjab University, Chandigarh, 08-10 February, 2007
5. Jie Liang, Estimation Methods for the Size of Deep Web Textural Data Source: A Survey, ACM Transactions on Computational Logic, Vol. V, No. N, August 2008, Pages 1-17.
6. B Pinkerton, Finding What People Want, Experiences with the Web Crawler, Proceedings of the Second International World Wide Web Conference, 1994.
7. S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems, 30, 107-117, 1998.
8. A. Heydon and M. Najork. Mercator, A scalable extensible Web crawler, World Wide Web, 2(4), 219-229, 1999.
9. J. Edwards, K. McCurley and J. Tomlin, An adaptive model for optimizing performance of an incremental web crawler, Proceedings of the 10th international conference on World Wide Web, pp 106-113, 2001.
10. D. Zeinalipour-Yazti and M. Dikaiakos, Design and implementation of a distributed crawler and filtering processor, Proc. Of NGITS 2002, 58-74.
11. P-Boldi, B. Codenotti, M. Santini and S. Vigna, UbiCrawler: a scalable fully distributed Web crawler, Software – Practice and Experience, 34 (8), 711-726, 2004.
12. R.R. Trujillo and A. Ardo, Simulation tool to study focused web crawling strategies, 2006.
13. D. Gibson, J Kleinberg and P. Raghavan, Inferring Web communities from link topology, Proc. of the ninth ACM conference on Hypertext and hypermedia:links, objects, time and space – structure in hypermedia systems, ACM Press New York, pp 225-234, 1998.

14. B.D. Davison, Topical locality in the Web, Proc. of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, ACM New York, pp. 272-279, 2000.
15. E. Amitay, Using common hypertext links to identify the best phrasal description of target web documents, Proc. of the SIGIR, vol 98, 1998.
16. J. Kleinberg, Authoritative Sources in a Heperlinked Environment, Journal of the ACM, 46 (5), 604-632, 1999.
17. M. Najork and J. L. Wiener. Breadth-first crawling yields high-quality pages. Proc. of the 10th international conference on World Wide Web, ACM Press NY, pp. 114-118, 2001.
18. Gary Flake, Steve Lawrence and Lee L. Giles, Efficient identification of web communities, Proc of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 150-160, 2000.
19. D. Bergmark, Collection synthesis, Proc. of the 2nd ACM / IEEE – CS joint conference on Digital libraries, ACM NY, pp. 253-262, 2002.
20. V. Kluev, Compiling document collections from the Internet, ACM SIGIR Forum, ACM Press New York, vol. 34, pp. 9-14, 2000.
21. Andrew McCallum, Kamal Nigam, Jason Rennie and Kristie Seymore, Building domain-specific search engine with machine learning techniques, AAAI Sprng Symposium on Intelligent Agents in Cyberspace, 1999.
22. D. Bergmark, C. Lagoze and Sbityakov, Focused Crawls, Tunneling and Digital Libraries, Lecture notes in computer science, pp. 91-106, 2002.
23. K. Chang, B. He and Z. Zhang, Toward large scale integration: Building a metaquerier over databases on the web, Proc. of CIRD, pp 44-55, 2005.
24. J. Wang, J.R. Wen, F. Lochovsky and W.Y. Ma, Instance-based schema matching for web databases by domain-specific query probing. Proc. of the 13th international conference on Very large data bases, vol 30, pp. 408-419, 2004.
25. A. Maedche and S. Staab, Ontology learning for the Semantic Web, Intelligent Systems, IEEE, 16(2), pp. 72-79, 2001.
26. S. Raghavan and H. Garcia-Molina, Crawling the hidden web, Proc. of the international conference on very large data bases, pp. 129-138, 2001.
27. A. Arasu and H. Garcia-Molina, Extracting structured data from Web pages, Proc. of the 2003 ACM SIGMOD international conference on Management of data, pp. 337-348, 2003.

28. P Wu, J.R. Wen, H. Liu and W.Y. Ma, Query selection techniques for efficient crawling of structured web sources, Proc. of ICDE, 2006.