ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Πρόγραμμα Προπτυχιακών Σπουδών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ

ΓΕΩΡΓΙΟΣ ΦΑΚΟΥΚΑΚΗΣ

Επιβλέπων καθηγητής: Μιχάλης Σαλαμπάσης

Θεσσαλονίκη, Ιανουάριος 2022

Υπεύθυνος Καθηγητής:

Μιχάλης Σαλαμπάσης

Διεθνές Πανεπιστήμιο της Ελλάδος

Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ

ΓΕΩΡΓΙΟΣ ΦΑΚΟΥΚΑΚΗΣ

Υπεύθυνη Δήλωση Συγγραφέα:

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γεώργιου Φακουκάκη που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Ευχαριστώ ιδιαιτέρως τον καθηγητή μου Μιχάλη Σαλαμπάση,
για την ευκαιρία που μου έδωσε να εργαστώ σε ένα τόσο μεγάλο
project αλλά και για την εμπιστοσύνη που μου έδειξε για την διεκπεραίωση
αναβάθμισης και εργασίας πάνω σε μηχάνημα Linux Debian που συντηρεί.

Πρόλογος

Η συγκεκριμένη πτυχιακή εργασία επιλέχθηκε γιατί ήθελα να εργαστώ σε μία μεγάλη εφαρμογή όπως το PerFedPat, το οποίο επεκτείνει ένα ήδη μεγάλο σύστημα (ezDL) αλλά και γιατί βρίσκω τον τρόπο λειτουργίας των μηχανών αναζήτησης συναρπαστικό. Αποκτήθηκε πολύ μεγαλύτερη αντίληψη για τον τρόπο λειτουργίας των μηχανών αναζήτησης, αλλά και για το πως χρησιμοποιούνται τα εκάστοτε συστήματα για τη δημιουργία ευρετηρίων και επιστροφή αποτελεσμάτων με βάση τα ερωτήματα και φυσικά την ταξινόμηση τους. Αποκτήθηκε γνώση για το Solr και Lucene, που ήταν η πρώτη μου επαφή και με τα δύο. Το Solr έχει πάρα πολλές λειτουργίες αξιοποιώντας το Lucene και επεκτείνοντάς το, κάνοντας το μία πολύ δυνατή μηχανή αναζήτησης. Ο χρόνος που αφιερώθηκε στη δημιουργία του ευρετηρίου, μου έδειξε πως μπορούν να λυθούν προκλήσεις ασυμβατότητας σε πολύ μεγάλες συλλογές. Το PerFedPat με βοήθησε να μάθω πως να εργαστώ σε ένα τόσο μεγάλο έργο και πως γίνεται η σωστή δημιουργία συστημάτων που αλληλεπιδρούν χωρίς το ένα να επηρεάζει άμεσα το άλλο. Μέσα από την πτυχιακή εργασία, μου δόθηκε η μοναδική ευκαιρία να μάθω πολλούς διαφορετικούς τομείς των μηχανών αναζήτησεις, καθώς και πως να δημιουργήσω μία δική μου.

Περίληψη

Η παρούσα πτυχιακή εργασία στοχεύει την επέκταση συστήματος αναζήτησης πατεντών, καθώς και την ανάπτυξη ενός νέου ευρετηρίου βασισμένο σε Lucene/Solr με 1.768.493 έγγραφα. Αξιοποιείται το σύστημα PerFedPat για την δημιουργία ερωτημάτων προς το ευρετήριο. Το PerFedPat είναι ένα framework που παρέχει πολλαπλές διεπαφές χρήστη και εργαλεία, δίνοντας τη δυνατότητα στον χρήστη να ψάξει ταυτόχρονα σε πολλές ανεξάρτητες πηγές, κάνοντας μία μόνο ερώτηση, επιστρέφοντας ένα ενιαίο σύνολο αποτελεσμάτων. Το PerFedPat βασίζεται σε ένα λογισμικό ανοιχτού κώδικα που ονομάζεται ezDL. Πληροφορίες για το ezDL και τον τρόπο λειτουργίας του θα δοθούν αργότερα. Μετά τη δημιουργία του ευρετηρίου και την αξιοποίηση του συστήματος PerFedPat, δημιουργήθηκε μια διεπαφή ιστού για την προβολή των πατεντών του ευρετηρίου αλλά και με επιπρόσθετες λειτουργίες όπως: αναζήτηση, κατηγοριοποίηση, επισήμανση και αυτόματη συμπλήρωση. Τα συγκεκριμένα συστήματα ανέβηκαν σε μηχάνημα Linux Debian όπου και είναι προσβάσιμα ανά πάσα στιγμή (http://perfedpat.salampasis.gr:5000/solr/clefip/browse).

Λέξεις – Κλειδιά

Πατέντες, Ενοποιημένη αναζήτηση, ezDL, PerFedPat, Lucene, Solr

ADVANCED PATENT SEARCH SYSTEMS

GEORGIOS FAKOUKAKIS

Abstract

This thesis aims to extend a patent search system, as well as to develop a new Lucene / Solr based index with 1.768.493 documents. Utilizing the PerFedPat system to generate index queries. PerFedPat is a framework that provides multiple user interfaces and tools, enabling the user to search multiple independent sources simultaneously, asking a single question, returning a single set of results. PerFedPat is based on an open source software called ezDL. Information about ezDL and how it works will be provided later. After the index was created and the PerFedPat system was utilized, a web interface was created to display the index patents but also with additional functions such as: search, categorization, highlighting and auto-completion. These systems have been uploaded to a Linux Debian machine where they are accessible at any time (http://perfedpat.salampasis.gr:5000/solr/clefip/browse).

Keywords

Patents, Federated search, ezDL, PerFedPat, Lucene, Solr

Περιεχόμενα

Κατάλογος Εικόνων / Πινάκων

1. Εισαγωγή

1.1 Πατέντες

Πατέντα (ή δίπλωμα ευρεσιτεχνίας) είναι η παραχώρηση ενός δικαιώματος ιδιοκτησίας από μία κυρίαρχη αρχή σε έναν εφευρέτη. Αυτή η επιχορήγηση, παρέχει στον εφευρέτη αποκλειστικά δικαιώματα στην κατοχυρωμένη με πατέντα διαδικασία, σχέδιο ή εφεύρεση για καθορισμένη περίοδο σε αντάλλαγμα με ολοκληρωμένη αποκάλυψη της εφεύρεσης. Για να μπορεί να κατοχυρωθεί ως πατέντα, πρέπει να είναι νέα, βιομηχανικά εφαρμόσιμη και να περιλαμβάνει εφευρετικότητα. Οι πατέντες προστατεύουν τις τεχνικές εφευρέσεις και αποτρέπουν τρίτους από την εκμετάλλευση μιας εφεύρεσης για εμπορικούς σκοπούς χωρίς εξουσιοδότηση. Οι αιτήσεις για πατέντες και τα χορηγούμενα διπλώματα ευρεσιτεχνίας δημοσιεύονται, γεγονός που τα καθιστά βασική πηγή τεχνικών πληροφοριών.

1.1.1 Σκοπός πατεντών

Ιστορικά, η προστασία των πατεντών σχεδιάστηκε για να ενθαρρύνει την καινοτομία και την αποκάλυψη των λεπτομερειών νέων εφευρέσεων. Επειδή οι εφευρέτες μπορεί να διστάζουν να δημοσιοποιήσουν τη δημιουργία τους μήπως κάποιος την αντιγράψει, η προστασία των διπλωμάτων ευρεσιτεχνίας παρέχει ένα κίνητρο για να μοιραστούν ιδέες με ένα προσωρινό μονοπώλιο στη χρήση τους. Η πλήρης εφεύρεση αποκαλύπτεται στη δημόσια διαθέσιμη αίτηση.

1.1.2 Τύποι πατεντών

Υπάρχουν δύο τύποι διπλωμάτων ευρεσιτεχνίας: πατέντες χρησιμότητας (utility patents) και πατέντες σχεδιασμού (design patents). Κάθε τύπος διπλώματος ευρεσιτεχνίας έχει τις δικές του απαιτήσεις καταλληλότητας και προστατεύει έναν συγκεκριμένο τύπο εφεύρεσης ή ανακάλυψης. Ωστόσο, είναι πιθανό για μία εφεύρεση ή ανακάλυψη να διαθέτει δυνητικά περισσότερους από έναν τύπους ευρεσιτεχνιών για αυτήν. Για παράδειγμα, εάν ένα άτομο εφεύρει ένα αντικείμενο και επιθυμεί να κατοχυρώσει με πατέντα τόσο τα λειτουργικά χαρακτηριστικά όσο και το σχέδιο του αντικειμένου, ο εφευρέτης θα πρέπει να υποβάλει αίτηση για δύο ξεχωριστά διπλώματα ευρεσιτεχνίας.

1.1.3 Αξία των πατεντών

Αν και κάποτε τα διπλώματα ευρεσιτεχνίας έφταναν σπάνια στα δικαστήρια, σήμερα πολλοί εφευρέτες έχουν λάβει διακανονισμούς παραβάσεων πολλών εκατομμυρίων. Σε περιπτώσεις εκούσιας παράβασης, το δικαστήριο μπορεί να επιδικάσει στον εφευρέτη τριπλή

αποζημίωση. Αυτό μπορεί να αποφευχθεί με τη λήψη γραπτής γνωμάτευσης μη παραβίασης από δικηγόρο ευρεσιτεχνιών. Τα διπλώματα ευρεσιτεχνίας αντικατοπτρίζουν επίσης την αξία των εταιρειών στον κλάδο της τεχνολογίας. Οι επιχειρηματίες κεφαλαίου είναι πιο πιθανό να επενδύσουν σε εταιρείες που κατέχουν τουλάχιστον μία πατέντα.

1.2 Συστήματα αναζήτησης διαδικτύου

1.2.1 Εισαγωγή στις μηχανές αναζήτησης

Μια μηχανή αναζήτησης είναι ένα σύστημα λογισμικού που έχει σχεδιαστεί για να πραγματοποιεί αναζητήσεις στο διαδίκτυο. Πραγματοποιούν αναζήτηση στον Παγκόσμιο Ιστό με συστηματικό τρόπο, για συγκεκριμένες πληροφορίες που καθορίζονται σε ένα ερώτημα. Τα αποτελέσματα αναζήτησης παρουσιάζονται γενικά σε μια λίστα αποτελεσμάτων, που συχνά αναφέρονται ως σελίδες αποτελεσμάτων μηχανών αναζήτησης (SERP). Ορισμένες μηχανές αναζήτησης εξορύσσουν επίσης δεδομένα που είναι διαθέσιμα σε βάσεις δεδομένων ή ανοιχτούς καταλόγους. Σε αντίθεση με τους καταλόγους ιστού, οι οποίοι διατηρούνται μόνο από ανθρώπινους συντάκτες, οι μηχανές αναζήτησης διατηρούν επίσης πληροφορίες σε πραγματικό χρόνο εκτελώντας έναν αλγόριθμο σε έναν ανιχνευτή ιστού (web crawler). Περιεχόμενο διαδικτύου που δεν μπορεί να αναζητηθεί από μια μηχανή αναζήτησης περιγράφεται γενικά ως ο βαθύς ιστός (deep web).

1.2.2 Βασική λειτουργία μηχανών αναζήτησης

Οι μηχανές αναζήτησης είναι σε θέση να επιστρέφουν γρήγορα αποτελέσματα, ακόμη και με εκατομμύρια ιστότοπους στο διαδίκτυο, σαρώνοντας συνεχώς το διαδίκτυο και ευρετηριάζοντας (indexing) κάθε σελίδα που βρίσκουν. Όταν ένας χρήστης εισάγει έναν όρο αναζήτησης, η μηχανή αναζήτησης εξετάζει τους τίτλους, τα περιεχόμενα και τις λέξεις κλειδιά της ιστοσελίδας που έχει καταχωρήσει στο ευρετήριο και χρησιμοποιεί αλγόριθμους (λειτουργίες βήμα προς βήμα) για να δημιουργήσει μια λίστα ιστοτόπων, με τους πιο σχετικούς ιστότοπους στην κορυφή της λίστας. Οι εταιρείες χρησιμοποιούν τη βελτιστοποίηση μηχανών αναζήτησης (SEO) για να βοηθήσουν τις μηχανές αναζήτησης να αναγνωρίσουν τους ιστότοπούς τους ως εξαιρετικά σχετικούς με συγκεκριμένες αναζητήσεις.

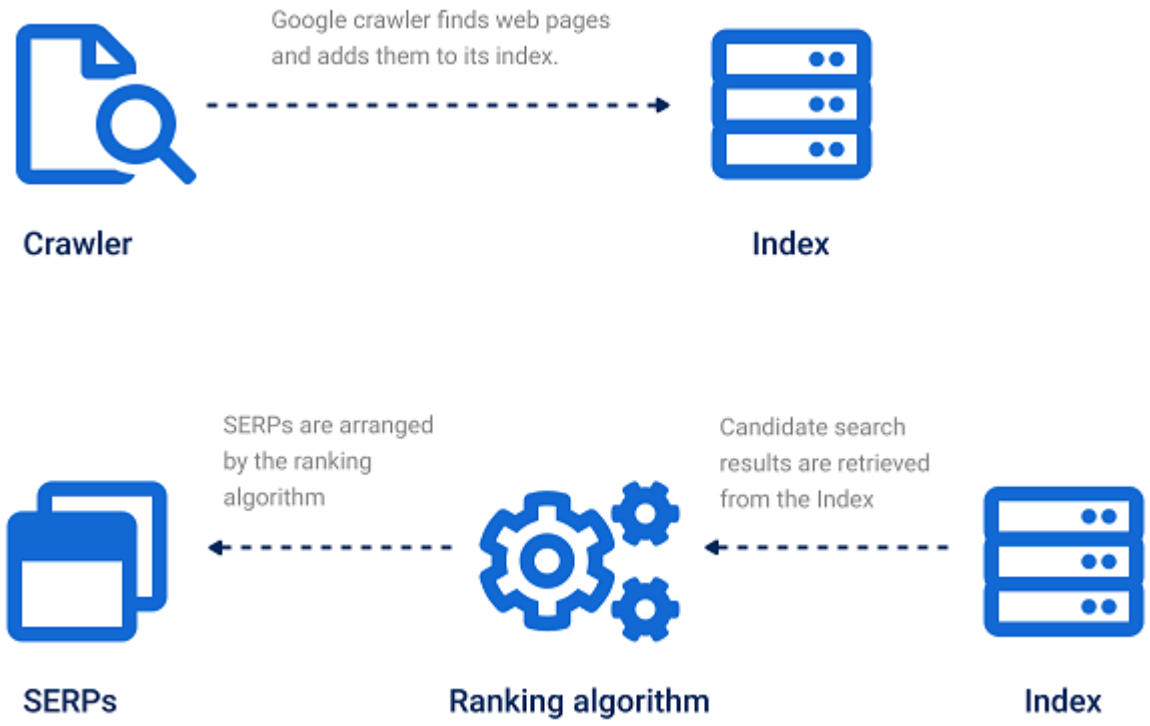
1.2.3 Τρόποι λειτουργίας μηχανών αναζήτησης

Ανίχνευση (Crawling): οι μηχανές αναζήτησης χρησιμοποιούν προγράμματα, που ονομάζονται spiders, bots ή crawlers, για να σαρώσουν το διαδίκτυο. Μπορεί να το κάνουν αυτό κάθε λίγες μέρες, επομένως είναι πιθανό το περιεχόμενο να είναι ξεπερασμένο μέχρι να ανιχνεύσουν ξανά έναν ιστότοπο.

Ευρετηρίαση (Indexing): η μηχανή αναζήτησης θα προσπαθήσει να κατανοήσει και να κατηγοριοποιήσει το περιεχόμενο μιας ιστοσελίδας μέσω «λέξεων-κλειδιών». Η τήρηση των βέλτιστων πρακτικών SEO θα βοηθήσει τη μηχανή αναζήτησης να κατανοήσει το περιεχόμενό της, ώστε να είναι εφικτή η κατάταξη για τα σωστά ερωτήματα αναζήτησης.

Κατάταξη (Ranking): τα αποτελέσματα αναζήτησης ταξινομούνται με βάση διάφορους παράγοντες. Αυτά μπορεί να περιλαμβάνουν πυκνότητα λέξεων κλειδιών, ταχύτητα και συνδέσμους. Στόχος της μηχανής αναζήτησης είναι να παρέχει στον χρήστη το πιο σχετικό αποτέλεσμα.

Αν και οι περισσότερες μηχανές αναζήτησης παρέχουν συμβουλές για το πως να βελτιωθεί η κατάταξη μιας σελίδας, οι ακριβείς αλγόριθμοι που χρησιμοποιούνται είναι καλά προστατευμένοι και αλλάζουν συχνά για να αποφευχθεί η κακή χρήση.



1.2.4 Apache Lucene

Η Lucene είναι μία βιβλιοθήκη μηχανών αναζήτησης κειμένου υψηλής απόδοσης, με πλήρεις δυνατότητες, γραμμένη εξ ολοκλήρου σε Java. Είναι μια τεχνολογία κατάλληλη για σχεδόν οποιαδήποτε εφαρμογή που απαιτεί αναζήτηση πλήρους κειμένου, ειδικά διαπλατφορμική. Θα πρέπει να σημειωθεί ότι η Lucene δεν είναι μια πλήρης εφαρμογή αναζήτησης που μπορεί κανείς να αρχίσει να τη χρησιμοποιεί «ως έχει». Είναι μια βιβλιοθήκη λογισμικού, με

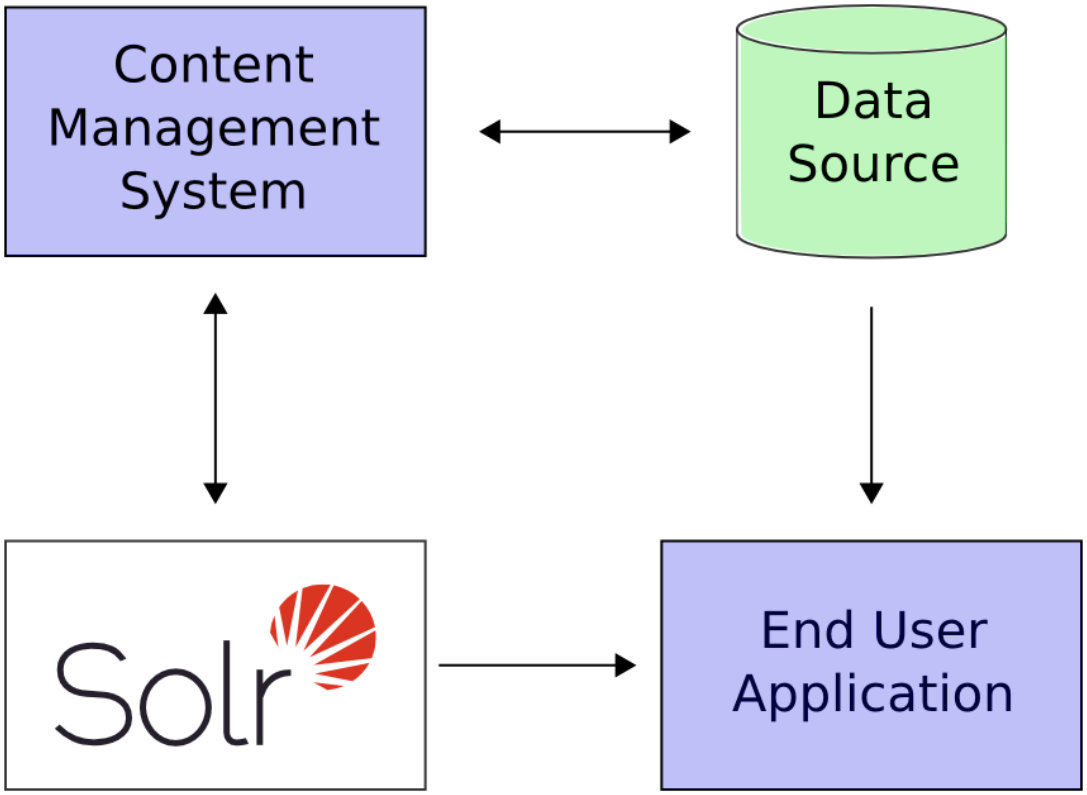
δυνατότητες ευρετηρίασης και αναζήτησης που μπορεί να ενσωματωθεί με διάφορες εφαρμογές. Η Lucene, ως βιβλιοθήκη Java, είναι πολύ ευέλικτη σε σύγκριση με άλλες εφαρμογές αναζήτησης. Προσφέρει ικανοποιητικές δυνατότητες ως βιβλιοθήκη αναζήτησης:

- Boolean λογική
- Αναζήτηση φράσεων και εγγύτητας
- Κατάταξη σχετικότητας
- Περιήγηση σε ευρετήρια
- Περικοπή
- Αναζήτηση πεδίου
- Αναζήτηση πεζών-κεφαλαίων
- Ελεγχόμενο λεξιλόγιο
- Μετάφραση γλώσσας
- Αναζήτηση ημερομηνίας/περιοχής
- Διύλιση της αρχικής αναζήτησης
- Σχετικά αντικείμενα
- Αναζήτηση πολυμέσων
- Προηγμένες και βασικές δυνατότητες αναζήτησης
- Μορφές εμφάνισης
- Πληροφορίες βοήθειας και τεκμηρίωσης

1.2.5 Apache Solr

Το Solr είναι μια πλατφόρμα εταιρικής αναζήτησης ανοιχτού κώδικα, γραμμένη σε Java. Τα κύρια χαρακτηριστικά του περιλαμβάνουν αναζήτηση πλήρους κειμένου, επισήμανση αποτελεσμάτων, πολύπλευρη αναζήτηση (faceted search), ευρετηρίαση σε πραγματικό χρόνο, δυναμική ομαδοποίηση, ενοποίηση βάσεων δεδομένων, λειτουργίες NoSQL και διαχείριση εμπλουτισμένων εγγράφων (π.χ. Word, PDF). Το Solr παρέχει κατανεμημένη αναζήτηση, αντιγραφή ευρετηρίου (index replication) και έχει σχεδιαστεί για επεκτασιμότητα και ανοχή σφαλμάτων. Χρησιμοποιείται ευρέως για περιπτώσεις χρήσης εταιρικής αναζήτησης και ανάλυσης και έχει μια ενεργή κοινότητα ανάπτυξης με τακτικές εκδόσεις. Το Solr εκτελείται ως αυτόνομος διακομιστής (server) αναζήτησης πλήρους κειμένου. Χρησιμοποιεί τη βιβλιοθήκη αναζήτησης Lucene στον πυρήνα του για ευρετηρίαση και αναζήτηση πλήρους κειμένου και διαθέτει API τύπου REST HTTP/XML και JSON που το καθιστούν χρησιμοποιήσιμο από τις πιο δημοφιλείς γλώσσες προγραμματισμού. Η εξωτερική διαμόρφωση του Solr, του επιτρέπει να προσαρμόζεται σε πολλούς τύπους εφαρμογών χωρίς κωδικοποίηση Java και διαθέτει αρχιτεκτονική πρόσθετων για την υποστήριξη πιο προηγμένης προσαρμογής. Για την αναζήτηση ενός εγγράφου, εκτελεί τις ακόλουθες λειτουργίες με τη σειρά:

1. Ευρετηρίαση: μετατρέπει τα έγγραφα σε μορφή αναγνώσιμη από μηχανή.
2. Ερώτημα: κατανόηση των όρων ενός ερωτήματος που τίθεται από τον χρήστη. Αυτοί οι όροι μπορεί να είναι εικόνες ή λέξεις-κλειδιά, για παράδειγμα.
3. Αντιστοίχιση: Το Solr αντιστοιχίζει το ερώτημα χρήστη στα έγγραφα που είναι αποθηκευμένα στη βάση δεδομένων για να βρει το κατάλληλο αποτέλεσμα.
4. Κατάταξη: μόλις η μηχανή αναζητήσει τα ευρετηριασμένα έγγραφα, ταξινομεί τα αποτελέσματα με βάση τη συνάφειά τους.



2. Ενοποιημένη αναζήτηση

2.1 Βασική λειτουργία ενοποιημένης αναζήτησης

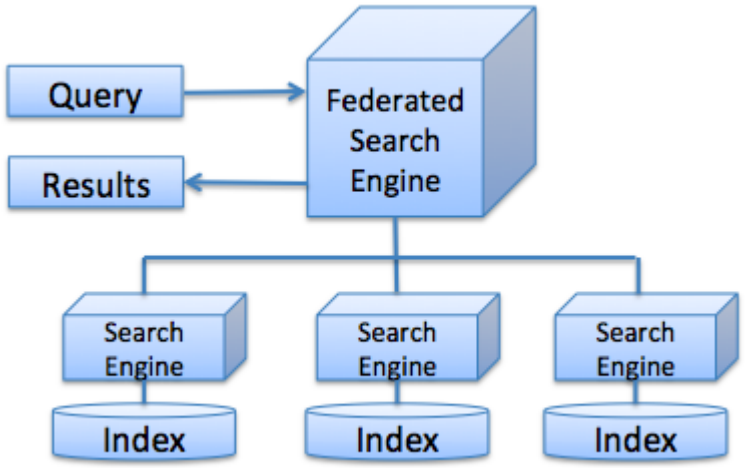
Η ενοποιημένη αναζήτηση ανακτά πληροφορίες από διάφορες πηγές μέσω μιας εφαρμογής αναζήτησης που έχει δημιουργηθεί πάνω σε μία ή περισσότερες μηχανές αναζήτησης. Ένας χρήστης υποβάλλει ένα μόνο αίτημα ερωτήματος το οποίο διανέμεται στις μηχανές αναζήτησης, τις βάσεις δεδομένων ή άλλες μηχανές ερωτημάτων. Στη συνέχεια, η ενοποιημένη αναζήτηση συγκεντρώνει τα αποτελέσματα που λαμβάνονται από τις μηχανές αναζήτησης για παρουσίαση στον χρήστη. Η ενοποιημένη αναζήτηση μπορεί να χρησιμοποιηθεί για την ενσωμάτωση διαφορετικών πόρων πληροφοριών σε έναν μόνο μεγάλο οργανισμό (επιχείρηση) ή για ολόκληρο τον ιστό. Η ενοποιημένη αναζήτηση, σε αντίθεση με την κατανεμημένη αναζήτηση, απαιτεί κεντρικό συντονισμό των πόρων με δυνατότητα αναζήτησης. Αυτό περιλαμβάνει τόσο τον συντονισμό των ερωτημάτων που μεταδίδονται στις μεμονωμένες μηχανές αναζήτησης, όσο και τη συγχώνευση των αποτελεσμάτων αναζήτησης που επιστρέφονται από καθεμία από αυτές.

2.1.1 Διαδικασία ενοποιημένης αναζήτησης

Η ενοποιημένη αναζήτηση αποτελείται από τη μετατροπή ενός ερωτήματος και τη μετάδοσή του σε μια ομάδα διαφορετικών βάσεων δεδομένων ή άλλων πόρων ιστού, με την κατάλληλη σύνταξη, τη συγχώνευση των αποτελεσμάτων που συλλέγονται από τις βάσεις δεδομένων, την παρουσίασή τους σε συνοπτική και ενοποιημένη μορφή με ελάχιστη αντιγραφή και παροχή ενός μέσου, που εκτελείται είτε αυτόματα είτε από τον χρήστη, για την ταξινόμηση του συγχωνευμένου συνόλου αποτελεσμάτων. Οι μεμονωμένες πηγές πληροφοριών στέλνουν πίσω στη διεπαφή του προγράμματος μια λίστα αποτελεσμάτων από το ερώτημα αναζήτησης. Τα πιο ανεπτυγμένα προγράμματα ενοποιημένης αναζήτησης θα συγχωνεύσουν ή διαγράψουν διπλότυπα αποτελέσματα. Υπάρχουν κι άλλες ανεπτυγμένες λειτουργίες, αλλά η βασική ιδέα είναι η ίδια, να βελτιωθεί η ακρίβεια και η συνάφεια των μεμονωμένων αναζητήσεων καθώς και να μειωθεί ο χρόνος που απαιτείται για την αναζήτηση πόρων. Αυτή η διαδικασία επιτρέπει στην ενοποιημένη αναζήτηση ορισμένα βασικά πλεονεκτήματα σε σύγκριση με τις υπάρχουσες μηχανές αναζήτησης που βασίζονται σε ανιχνευτή (crawler). Η ενοποιημένη αναζήτηση δεν χρειάζεται να επιβάλλει απαιτήσεις ή βάρη στους κατόχους των μεμονωμένων πηγών πληροφοριών, εκτός από το χειρισμό αυξημένης επισκεψιμότητας. Οι ενοποιημένες αναζητήσεις είναι εγγενώς τόσο επίκαιρες όσο και οι μεμονωμένες πηγές πληροφοριών, καθώς αναζητούνται σε πραγματικό χρόνο.

2.1.2 Εφαρμογή ενοποιημένης αναζήτησης

Μια εφαρμογή της ενοποιημένες αναζήτησης είναι η μηχανή μετα-αναζήτησης (metasearch). Ωστόσο, η προσέγγιση μετα-αναζήτησης δεν ξεπερνά τα μειονεκτήματα των μηχανών αναζήτησης στοιχείων (components), όπως τα ελλιπή ευρετήρια. Έγγραφα που δεν ευρετηριάζονται από τις μηχανές αναζήτησης δημιουργούν αυτό που είναι γνωστό ως βαθύς ιστός ή αόρατος ιστός. Η προσέγγιση μετα-αναζήτησης, όπως και η υποκείμενη τεχνολογία μηχανών αναζήτησης, λειτουργεί μόνο με πηγές πληροφοριών που είναι αποθηκευμένες σε ηλεκτρονική μορφή. Μία από τις κύριες προκλήσεις της μετα-αναζήτησης, είναι η διασφάλιση ότι το ερώτημα αναζήτησης είναι συμβατό με τις μηχανές αναζήτησης στοιχείων που συνενώνονται και συνδυάζονται. Όταν το λεξιλόγιο αναζήτησης ή το μοντέλο δεδομένων του συστήματος αναζήτησης είναι διαφορετικό από το μοντέλο δεδομένων ενός ή περισσότερων από τα στοχευμένα συστήματα, το ερώτημα πρέπει να μεταφραστεί σε καθένα από τα στοχευμένα συστήματα. Αυτό μπορεί να γίνει χρησιμοποιώντας απλή μετάφραση στοιχείων ή μπορεί να απαιτεί σημασιολογική μετάφραση. Για παράδειγμα, εάν μια μηχανή αναζήτησης επιτρέπει την παράθεση συμβολοσειρών και μια άλλη όχι, το ερώτημα πρέπει να μεταφραστεί ώστε να είναι συμβατό με κάθε μηχανή αναζήτησης. Μια άλλη πρόκληση που αντιμετωπίζεται στην εφαρμογή των ενοποιημένων μηχανών αναζήτησης είναι η επεκτασιμότητα. Είναι δύσκολο να διατηρηθεί η απόδοση και η ταχύτητα απόκρισης μιας ενοποιημένης μηχανής αναζήτησης, καθώς συνδυάζει όλο και περισσότερες πηγές πληροφοριών μαζί.



2.1.3 Προκλήσεις ενοποιημένης αναζήτησης

Όταν η ενοποιημένη αναζήτηση εκτελείται έναντι ασφαλών πηγών δεδομένων, τα διαπιστευτήρια των χρηστών πρέπει να διαβιβάζονται σε κάθε υποκείμενη μηχανή αναζήτησης, έτσι ώστε να διατηρείται η κατάλληλη ασφάλεια. Εάν ο χρήστης έχει διαφορετικά διαπιστευτήρια σύνδεσης για διαφορετικά συστήματα, πρέπει να υπάρχει ένα μέσο για να αντιστοιχιστεί το αναγνωριστικό σύνδεσής του στον τομέα ασφαλείας κάθε μηχανής αναζήτησης. Μια άλλη πρόκληση είναι η ταξινόμηση και η βαθμολογία των αποτελεσμάτων. Κάθε πόρος ιστού έχει τη δική του έννοια βαθμολογίας συνάφειας και μπορεί να υποστηρίζει ορισμένες ταξινομήσεις αποτελεσμάτων. Η συνάφεια ποικίλλει πολύ μεταξύ των αποτελεσμάτων στην αναζήτηση, επομένως είναι δύσκολο ή αδύνατο να γνωρίζει πως να παρεμβάλλει τα αποτελέσματα για να δείξει τα πιο σχετικά. Επίσης, ο συνδυασμός των αποτελεσμάτων σε μια κοινή φόρμα, αποτελεί συχνό πρόβλημα. Μια άλλη πρόκληση είναι η διαθεσιμότητα και το χρονικό περιθώριο. Καθώς ο αριθμός των πηγών αυξάνεται, η πιθανότητα μιας ή περισσότερων αργών ή εκτός σύνδεσης πηγών γίνεται όλο και υψηλότερη. Η ενοποιημένη αναζήτηση πρέπει να αποφασίσει πότε θα εξετάσει μια αναζήτηση εκτός σύνδεσης ή να περιμένει μια αργή απόκριση. Οι χρόνοι απόκρισης θα υπαγορεύονται από τον πιο αργή πηγή της ομάδας. Μια άλλη πρόκληση είναι η ανάπτυξη και οι δοκιμές σε μια επιχείρηση (έναντι στο δημόσιο διαδίκτυο). Οι ομάδες ανάπτυξης δεν πρέπει συνήθως να χτυπούν ζωντανά συστήματα παραγωγής καθώς εκτελούν κανονική εργασία. Επίσης, ορισμένοι πόροι είναι ασφαλείς και δεν θα πρέπει να υποβάλλονται σε αυθαίρετα ερωτήματα και να εκτίθενται στην ανάπτυξη, λόγω ανησυχιών σχετικά με το απόρρητο και την ασφάλεια. Ως εκ τούτου, τα περιβάλλοντα ανάπτυξης, δοκιμών και δοκιμών απόδοσης πρέπει να περιλαμβάνουν εγκατάσταση και διαμόρφωση για πολλά υποσυστήματα ώστε να επιτρέπεται η ασφαλής δοκιμή.

2.2 Τύποι ενοποιημένης αναζήτησης

2.2.1 Συγχώνευση χρόνου αναζήτησης

Μερικές φορές αποκαλούμενη συγχώνευση χρόνου ερωτήματος, περιλαμβάνει τη διατήρηση ενός ξεχωριστού ευρετηρίου για κάθε πηγή δεδομένων που χρειάζεται να συμπεριληφθεί στην ενοποιημένη αναζήτησή. Στη συνέχεια, για να πραγματοποιηθεί μια αναζήτηση, πραγματοποιείται αναζήτηση σε κάθε ευρετήριο ξεχωριστά για έναν δεδομένο όρο αναζήτησης. Μπορεί επίσης να χρειαστεί να αφαιρεθούν τα αντίγραφα των δεδομένων, προσδιορίζοντας αποτελέσματα που προέρχονται από περιττές πηγές δεδομένων. Τέλος, τα αποτελέσματα αναζήτησης συγκεντρώνονται για να παράγουν τη λίστα των τελικών αποτελεσμάτων. Το κύριο πλεονέκτημα της συγχώνευσης χρόνου αναζήτησης είναι ότι είναι η απλούστερη μέθοδος ενοποιημένης αναζήτησης προς εφαρμογή. Επειδή δεν απαιτείται να

δημιουργηθεί ένα κεντρικό ευρετήριο για όλες τις πηγές δεδομένων, είναι γρήγορα εφικτή η ρύθμιση μιας λύσης χρόνου αναζήτησης, χρησιμοποιώντας τους δείκτες που υπάρχουν ήδη σε κάθε πηγή δεδομένων. Επιπλέον, η συγχώνευση χρόνου αναζήτησης μπορεί να είναι πιο απλή στη ρύθμιση, επειδή δεν χρειάζεται να τυποποιηθούν οι δείκτες. Οι δομές δεδομένων για ένα ευρετήριο μπορεί να είναι διαφορετικές από εκείνες για ένα άλλο, αλλά η συγχώνευση χρόνου αναζήτησης θα λειτουργήσει και με τα δύο. Από την άλλη πλευρά, ο ρυθμός απόδοσης των αναζητήσεων που πραγματοποιούνται με τη χρήση συγχώνευσης χρόνου αναζήτησης τείνει να είναι πιο αργός από αυτόν των άλλων ενοποιημένων μεθόδων αναζήτησης. Είναι λιγότερο αποτελεσματική η ανεξάρτητη αναζήτηση πολλών δεικτών. Εάν ένα ευρετήριο αργεί ιδιαίτερα να ανταποκριθεί, ολόκληρη η αναζήτηση θα καθυστερήσει. Τέλος, η δημιουργία μιας ικανοποιητικής συνάφειας για τη λίστα συγκεντρωτικών αποτελεσμάτων μπορεί να είναι πολύ δύσκολη.

2.2.2 Συγχώνευση χρόνου ευρετηρίου

Μια εναλλακτική προσέγγιση για την ενοποιημένη αναζήτηση είναι η συγχώνευση χρόνου ευρετηρίου. Με αυτήν την προσέγγιση, δημιουργείται ένα κεντρικό ευρετήριο για όλες τις πηγές δεδομένων και στη συνέχεια, αναλύεται αυτό το ευρετήριο για να πραγματοποιηθεί αναζήτηση. Επειδή πρέπει να αναζητηθεί μόνο ένα ευρετήριο, η συγχώνευση χρόνου ευρετηρίου συνήθως οδηγεί σε ταχύτερες αναζητήσεις από τη συγχώνευση χρόνου αναζήτησης. Αυτό είναι το κύριο πλεονέκτημα της συγχώνευσης χρόνου ευρετηρίου. Επιτρέπει επίσης τη συμπερίληψη πηγών δεδομένων που δεν έχουν τη δική τους λειτουργία αναζήτησης και επομένως δεν μπορούν να χρησιμοποιηθούν με τη συγχώνευση χρόνου αναζήτησης. Το κύριο μειονέκτημα της συγχώνευσης χρόνου ευρετηρίου είναι ότι απαιτεί περισσότερη προσπάθεια για να εφαρμοστεί. Αντί να μπορεί να γίνει ανάλυση μιας συλλογής δεικτών, πρέπει να δημιουργηθεί ένα κεντρικό ευρετήριο για όλες τις πηγές δεδομένων και να ενημερώνεται το ευρετήριο κάθε φορά που αλλάζουν οι πηγές δεδομένων. Επιπλέον, εάν ορισμένες από τις πηγές δεδομένων έχουν διαφορετική μορφή από άλλες, πρέπει να τυποποιηθούν όλα τα δεδομένα ώστε να έχουν την ίδια μορφή. Ομοίως με τη συγχώνευση χρόνου αναζήτησης, εξακολουθεί να απαιτεί μια μοναδική στρατηγική συνάφειας για όλους τους διαφορετικούς τύπους περιεχομένου, πράγμα το οποίο δεν είναι βέλτιστο.

2.2.3 Υβριδική ενοποιημένη αναζήτηση

Υβριδική προσέγγιση για την ενοποιημένη αναζήτηση συνδυάζοντας ορισμένες από τις μεθόδους τόσο από τη συγχώνευση χρόνου αναζήτησης όσο και από τη συγχώνευση χρόνου ευρετηρίου. Για μια υβριδική ενοποιημένη αναζήτηση, δημιουργείται ένα κεντρικό ευρετήριο για όσο το δυνατόν περισσότερες πηγές δεδομένων, όπως ακριβώς θα γινόταν για τη

συγχώνευση χρόνου ευρετηρίου. Ωστόσο, εάν οι πηγές δεδομένων δε μπορούν εύκολα να αναπαρασταθούν στο κεντρικό ευρετήριο, διατηρούνται ξεχωριστοί δείκτες για αυτές. Όταν εκτελείται μια αναζήτηση, πραγματοποιείται αναζήτηση σε όλους τους δείκτες στο κεντρικό ευρετήριο, καθώς και στους πρόσθετους δείκτες που υπάρχουν για οποιεσδήποτε άλλες πηγές δεδομένων που δεν αντιπροσωπεύονται στο κεντρικό ευρετήριο. Τα αποτελέσματα αναζήτησης που βασίζονται σε όλους τους δείκτες, συγκεντρώνονται για να δημιουργηθεί μια τελική λίστα, όπως θα γινόταν με τη συγχώνευση χρόνου αναζήτησης. Με τη μείωση του αριθμού των δεικτών που πρέπει να αναζητηθούν, η υβριδική ενοποιημένη αναζήτηση παρέχει καλύτερη απόδοση από ό,τι θα πετυχαινόταν με τη συγχώνευση χρόνου αναζήτησης. Ταυτόχρονα, ωστόσο, δεν απαιτεί τη δημιουργία ενός ενιαίου ευρετηρίου για όλες τις πηγές δεδομένων. Το κύριο μειονέκτημα της τεχνικής υβριδικής ενοποιημένης αναζήτησης είναι ότι, επειδή υπάρχουν ακόμα περισσότερα από ένα ευρετήρια για αναζήτηση, η απόδοση είναι συνήθως πιο αργή από ό,τι θα ήταν αν υπήρχε ένα μόνο ευρετήριο.

2.2.4 Ενοποιημένη διεπαφή αναζήτησης

Αυτή η μέθοδος ξεκινά παρόμοια με τη μέθοδο συγχώνευσης χρόνου αναζήτησης, αλλά αντί να συγκεντρώνει τα αποτελέσματα σε μία λίστα αποτελεσμάτων, παρουσιάζει μία λίστα αποτελεσμάτων για κάθε τύπο περιεχομένου στον οποίο εκτελείται η αναζήτηση, σε μια ενοποιημένη διεπαφή. Όχι μόνο η ενοποιημένη διεπαφή αναζήτησης προσφέρει ανώτερη απόδοση, αλλά επιτρέπει επίσης στους κατόχους ιστότοπων να ρυθμίζουν ανεξάρτητα τη συνάφεια για κάθε τύπο περιεχομένου. Ωστόσο, η επίτευξη αυτών των πλεονεκτημάτων απαιτεί λίγη προνοητικότητα. Ο σχεδιασμός της τελικής διεπαφής πρέπει να αντικατοπτρίζει την εμπειρία που θέλει ένας ιδιοκτήτης ιστότοπου, να έχουν οι επισκέπτες. Επομένως απαιτείται κάποιος στρατηγικός σχεδιασμός. Επιπλέον, δεν είναι εξοπλισμένα όλα τα εργαλεία ιστοτόπων αναζήτησης για να εμφανίζουν μια ενοποιημένη διεπαφή αναζήτησης. Έτσι, ένας κάτοχος ιστότοπου θα πρέπει να διασφαλίσει ότι η λύση αναζήτησης του ιστότοπού του είναι ικανή τόσο να ευρετηριάζει διαφορετικούς τύπους περιεχομένου σε διαφορετικούς δείκτες όσο και να παρουσιάζει αυτές τις πληροφορίες με τον πιο φιλικό προς τον χρήστη τρόπο.

2.3 Κριτήρια επιλογής ενοποιημένης αναζήτησης

Θα πρέπει να ληφθούν υπόψη οι τύποι δεδομένων και ποια εργαλεία είναι διαθέσιμα για χειρισμό, δημιουργία ευρετηρίου και αναζήτησης. Εάν οι πηγές δεδομένων περιλαμβάνουν πολλές διαφορετικές μορφές, μια προσέγγιση χρόνου αναζήτησης θα έχει συνήθως τη μεγαλύτερη λογική. Ο χρόνος αναζήτησης είναι επίσης πιο βιώσιμος εάν κάθε μία από τις πηγές δεδομένων μπορεί εύκολα να αναζητηθεί ανεξάρτητα, κάτι που συμβαίνει εάν τα

δεδομένα είναι δομημένα με συνέπεια. Εάν, από την άλλη πλευρά, όλα τα δεδομένα μπορούν εύκολα να τυποποιηθούν σε μια ενιαία βάση δεδομένων, η συγχώνευση χρόνου ευρετηρίου είναι μια καλύτερη λύση. Ωστόσο, εάν υπάρχει μια σειρά διαφορετικών μορφών περιεχομένου και η λύση αναζήτησής υποστηρίζει ενοποιημένες διεπαφές αναζήτησης, θα πρέπει να είναι η προτιμώμενη προσέγγιση. Οι προγραμματιστές είναι επίσης ένας σημαντικός παράγοντας για να αποφασιστεί ποια προσέγγιση ενοποιημένης αναζήτησης θα χρησιμοποιηθεί. Εάν υπάρχει μια μεγάλη ομάδα ανάπτυξης και οι απαραίτητοι πόροι για τη δημιουργία ενός κεντρικού ευρετηρίου, η συγχώνευση χρόνου ευρετηρίου μπορεί να είναι κατάλληλη. Αλλά για μικρότερες ομάδες ανάπτυξης, η συγχώνευση χρόνου αναζήτησης μπορεί να είναι μια πιο πρακτική επιλογή, καθώς απαιτεί λιγότερη προσπάθεια για την εφαρμογή της. Εάν η ομάδα προγραμματισμού δεν έχει μεγάλη εμπειρία στη δημιουργία εφαρμογών αναζήτησης, μια ενοποιημένη λύση αναζήτησης τρίτων μπορεί επίσης να είναι μια ελκυστική επιλογή.

3. Ευρετήρια

3.1 Ευρετήρια μηχανών αναζήτησης

Η ευρετηρίαση μηχανών αναζήτησης είναι η συλλογή, η ανάλυση και η αποθήκευση δεδομένων για τη διευκόλυνση της γρήγορης και ακριβούς ανάκτησης πληροφοριών. Ο σχεδιασμός ευρετηρίου ενσωματώνει διεπιστημονικές έννοιες από τη γλωσσολογία, τη γνωστική ψυχολογία, τα μαθηματικά, την πληροφορική και την επιστήμη των υπολογιστών. Ένα εναλλακτικό όνομα για τη διαδικασία, στο πλαίσιο των μηχανών αναζήτησης που έχουν σχεδιαστεί για την εύρεση ιστοσελίδων στο διαδίκτυο, είναι η ευρετηρίαση ιστού. Χωρίς ευρετήριο, η μηχανή αναζήτησης θα σάρωνε κάθε έγγραφο, κάτι που θα απαιτούσε σημαντικό χρόνο και υπολογιστική ισχύ. Για παράδειγμα, ενώ ένα ευρετήριο 10.000 εγγράφων μπορεί να αναζητηθεί μέσα σε κλάσματα του δευτερολέπτου, μια διαδοχική σάρωση κάθε λέξης σε 10.000 μεγάλα έγγραφα μπορεί να διαρκέσει ώρες. Ο πρόσθετος χώρος αποθήκευσης υπολογιστή που απαιτείται για την αποθήκευση του ευρετηρίου, καθώς και η σημαντική αύξηση του χρόνου που απαιτείται για να πραγματοποιηθεί μία ενημέρωση, ανταλλάσσονται με τον χρόνο που εξοικονομείται κατά την ανάκτηση πληροφοριών.

3.2 Δημιουργία ευρετηρίου με Solr

3.2.1 Προετοιμασία των εγγράφων

Στο πλαίσιο της παρούσας πτυχιακής εργασίας χρειάστηκε να δημιουργηθεί ένα νέο ευρετήριο μέσα από μία Clef-IP συλλογή που περιέχει 1.768.493 έγγραφα. Αρχικά, τα έγγραφα έπρεπε να μετατραπούν από την αρχική τους SGML μορφή σε XML, ώστε να είναι εφικτή η δημιουργία ευρετηρίου χρησιμοποιώντας το Solr, καθώς τα SGML είναι μη συμβατά αρχεία. Τα XML έγγραφα δεν είναι η μόνη επιλογή για δημιουργία ευρετηρίου με Solr, αλλά είναι η πιο συνηθισμένη. CSV και PDF είναι δύο άλλοι διαδεδομένοι τύποι αρχείων που μπορούν να χρησιμοποιηθούν. Ανεξάρτητα από τη μέθοδο που χρησιμοποιείται για την απορρόφηση δεδομένων, υπάρχει μια κοινή βασική δομή δεδομένων για τα δεδομένα που τροφοδοτούνται σε ένα ευρετήριο Solr: ένα έγγραφο που περιέχει πολλά πεδία, το καθένα με ένα όνομα και περιεχόμενο. Η διαδικασία έγινε σε δύο βήματα, το πρώτο πολύ γρηγορότερο από το δεύτερο. Ως πρώτο βήμα, χρησιμοποιήθηκε ένα batch script, το οποίο άλλαξε τον τύπο των εγγράφων από SGML σε XML, μια διαδικασία που για το μέγεθος της συγκεκριμένης συλλογής κρατάει κατά μέσο όρο 30 λεπτά. Ως δεύτερο βήμα, γράφτηκε ένα script στην εφαρμογή UltraEdit, το οποίο μετέτρεψε το περιεχόμενο των αρχείων από γραφή SGML σε XML. Παρόμοια scripts μπορούν να γραφούν σε Python ή άλλες γλώσσες προγραμματισμού. Στην προκειμένη περίπτωση χρησιμοποιήθηκε το UltraEdit καθώς είναι

μια εφαρμογή με πολύ γρήγορη αναζήτηση-αντικατάσταση και ο σκοπός ήταν η εξοικονόμηση χρόνου. Παρόλα αυτά, η διαδικασία κράτησε πολλές ώρες καθώς το script χρειαζόταν να ψάξει γραμμή-γραμμή, όλα τα έγγραφα της συλλογής και να τα τροποποιήσει σε μορφή που θα ήταν δεκτή από το Solr (δηλαδή σε μορφή XML). Οι ειδικοί χαρακτήρες που περιείχε η συλλογή, σε συνδυασμό με τα μοτίβα αναζήτησης που έπρεπε να χρησιμοποιηθούν, αλλά και τις διαφορετικές περιπτώσεις χαρακτήρων που είχε το εκάστοτε έγγραφο έκαναν αυτή τη διαδικασία ακόμα πιο δύσκολη και ακόμα πιο χρονοβόρα. Παρακάτω είναι κάποια παραδείγματα αναζήτησης-αντικατάστασης που έγιναν από το script στα 1,7 εκατομμύρια έγγραφα.

Αναζήτηση	Αντικατάσταση
<DOC>\xd?\xa	<add><doc>
\xd?\xa</DOC>	</doc></add>
<DOCNO>\xd?\xa	<field name="id">
\xd?\xa</DOCNO>	</field>
<TEXT>\xd?\xa	
\xd?\xa</TEXT>	

Η ακολουθία χαρακτήρων “\xd?\xa” στα παραδείγματα, είναι ένα regular expression το οποίο ορίζει την εφαρμογή να αντικαταστήσει την κενή γραμμή μετά τα πεδία αναζήτησης, ήταν απαραίτητο να διαγραφούν οι κενές γραμμές από όλα τα έγγραφα, διαφορετικά το Solr δε θα τα δεχόταν και θα έβγαζε σφάλματα. Η ετικέτα <TEXT> διαγράφηκε εντελώς από το περιεχόμενο καθώς δε χρησίμευε κάπου αλλά υπήρχε σε όλα τα έγγραφα για να διαφοροποιήσει το head από το body. Οι υπόλοιπες αντικαταστάσεις, είναι απαραίτητες για να δεχτεί τα έγγραφα το Solr και φυσικά πρέπει να οριστούν πεδία με ονόματα που θα βγάζουν νόημα για τον χρήστη που θα χρησιμοποιήσει το ευρετήριο, αλλά και για την ομάδα ανάπτυξης της μηχανής αναζήτησης. Παρακάτω, μερικοί άλλοι χαρακτήρες που χρειάστηκε να γίνει ειδικός χειρισμός από το script:

Αναζήτηση	Αντικατάσταση
<	&lt;
>	&gt;

&	&amp;
#	#
@	@
γ	γ

Η αντικατάσταση αυτών των χαρακτήρων ήταν απαιτητική, εφόσον η XML χρησιμοποιεί τους χαρακτήρες "<" ">" για τις ετικέτες της, άρα δε μπορούν να βρίσκονται μέσα σε ένα XML αρχείο χωρίς να έχουν αυτή τη συγκεκριμένη ιδιότητα. Στην HTML οι ειδικοί χαρακτήρες < και > θα τυπώσουν ακριβώς το ίδιο αποτέλεσμα σα να τους γράφαμε κανονικά. Το ίδιο ισχύει και για τον χαρακτήρα "&", είναι ειδικός χαρακτήρας της XML και χρησιμοποιείται για να τυπωθούν άλλοι ειδικοί χαρακτήρες όπως το "&gt". Αυτά είναι μόνο κάποια παραδείγματα από τις αντικαταστάσεις που έπρεπε να λάβουν μέρος. Μετά την πολύωρη εκτέλεση του script, έχουμε ως αποτέλεσμα έγγραφα που είναι πλέον αποδεκτά από το Solr και έτοιμα προς ευρετηριοποίηση.

3.2.2 Δημιουργία σχήματος Solr

Πλέον, έχοντας τα έγγραφα της συλλογής σε μορφή που είναι αποδεκτή από το Solr, πρέπει να δημιουργηθεί ένα σχήμα ή managed-schema όπως ονομάζεται στο Solr, ώστε να γνωρίζει τι ονόματα/τύποι δεδομένων των πεδίων μπορούν να γίνουν αποδεκτοί απο τα XML έγγραφα αλλά και για να είναι εφικτή η αναζήτησή τους στην πορεία. Προτού γίνει αυτό, απαιτείται η δημιουργία ενός πυρήνα (core). Χρειάζεται πρόσβαση στο διαχειριστικό του Solr και εφόσον δεν έχει γίνει καμία αλλαγή στις προεπιλογές του, η πρόσβαση αποκτάται μέσω ενός οποιουδήποτε φυλλομετρητή (browser) στο σύνδεσμο: http://localhost:8983/solr. Έπειτα, πηγαίνοντας στην κατηγορία Core Admin και στην πορεία πατώντας το κουμπί "Add Core", το Solr θα ζητήσει κάποια στοιχεία, όπου αρκεί η εισαγωγή ενός ονόματος στα πεδία "name" και "instanceDir", τα υπόλοιπα μπορούν να μείνουν όπως τα έχει συμπληρώσει αυτόματα το Solr. Το όνομα είναι συνηθισμένο, αλλά και καλή πρακτική να σχετίζεται με τη συλλογή που θα ευρετηριοποιηθεί. Μετά τη δημιουργία του πυρήνα, μένει η δημιουργία του σχήματος, η οποία μπορεί να επιταχυνθεί χρησιμοποιώντας ένα από τα έτοιμα παραδείγματα μέσα στον φάκελο "configsets" του Solr. Τα παραδείγματα περιέχουν αναλυτικούς σχολιασμούς, διευκολύνοντας την ομάδα ανάπτυξης από το να πρέπει να κατανοήσει όλη τη δομή διαβάζοντας documentation στο διαδίκτυο και τη χρονοβόρα διαδικασία δημιουργίας του σχήματος από το μηδέν. Παρόλα αυτά, χρειάζεται βασική αντίληψη του πως λειτουργούν οι τύποι δεδομένων, αλλα και γνώση για το περιεχόμενο των εγγράφων. Εφόσον όμως τροποποιήθηκαν τα έγγραφα σε προηγούμενο στάδιο, υπάρχει ήδη η γνώση για τι

ονόματα/τύπους δεδομένων θα χρειαστούν για τα πεδία του σχήματος. Ενδείκνυται η χρήση ενός σχήματος από παράδειγμα του Solr, καθώς έχει έτοιμους τύπους δεδομένων που υπό διαφορετικές συνθήκες θα έπρεπε να αναπτυχθούν από την ομάδα ανάπτυξης. Από τους πιο συνηθισμένους τύπους είναι ο αλφαριθμητικός "string", για να δημιουργηθεί και να είναι εφικτή η χρήση του πρέπει να δηλωθεί στο σχήμα καλώντας την κλάση "solr.StrField":
<fieldType name="string" class="solr.StrField" sortMissingLast="true" />
Μετά τη δήλωσή του μπορεί να χρησιμοποιηθεί ως τύπος δεδομένων σε οποιοδήποτε πεδίο του σχήματος. Η ιδιότητα "sortMissingLast" ελέγχει την τοποθέτηση των εγγράφων όταν δεν υπάρχει πεδίο ταξινόμησης. Παρακάτω γίνεται ανάλυση του πεδίου id, το οποίο υπάρχει στα περισσότερα ευρετήρια Solr και θα πρέπει να υπάρχει συγκεκριμένος λόγος για την αφαίρεσή του.
<field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />

- name: το όνομα του πεδίου
- type: ο τύπος του πεδίου
- indexed: ορίζεται σε αληθής/ψευδής ανάλογα αν χρειάζεται το πεδίο για να πραγματοποιηθεί αναζήτηση με αυτό
- stored: ορίζεται σε αληθής/ψευδής ανάλογα αν χρειάζεται να διαβαστούν οι τιμές/περιεχόμενο του πεδίου από το ευρετήριο
- required: εάν δηλωθεί true, δίνει εντολή στο Solr να απορρίψει κάθε προσπάθεια προσθήκης εγγράφου που δεν έχει τιμή για αυτό το πεδίο
- multiValued: εάν δηλωθεί true, υποδηλώνει ότι ένα μεμονωμένο έγγραφο μπορεί να περιέχει πολλές τιμές για αυτόν τον τύπο πεδίου. Στην περίπτωση του id, όπου χρειάζεται να είναι μοναδικό ανά έγγραφο δηλώνεται false

Η ίδια λογική όπως το id πρέπει να οριστεί σε όλα τα πεδία των εγγράφων. Αν κάποιο από τα πεδία στα έγγραφα δεν υπάρχει στο σχήμα, το Solr θα αγνοήσει εντελώς το έγγραφο και δε θα το ευρετηριοποιήσει καθόλου. Άλλοι τύποι δεδομένων που μπορούν να χρησιμοποιηθούν στο σχήμα είναι: boolean, χρησιμοποιώντας την κλάση "solr.BoolField" για να δηλωθεί στο σχήμα, pint με την κλάση "solr.IntPointField", pdate με την κλάση solr.DatePointField και φυσικά πολλά άλλα, διαβάζοντας το documentation του Solr ή χρησιμοποιώντας ένα από τα παραδείγματα του είναι εφικτή η κατανόηση όλων των τύπων. Φυσικά τα ονόματα είναι ενδεικτικά, αλλά είναι πάντα καλή ιδέα να χρησιμοποιούνται ονόματα που θα είναι κατανοητά από διαφορετικές ομάδες ανάπτυξης, με την πρώτη ματιά. Επίσης, στο Solr μπορεί να χρησιμοποιηθεί πεδίο για ψευδο-τυχαία ταξινόμηση των εγγράφων, καλώντας την κλάση "solr.RandomSortField". Σημειώνεται ότι δε μπορεί να χρησιμοποιηθεί σε πεδίο για αποθήκευση δεδομένων. Μια μοναδική ικανότητα που παρέχει το Solr είναι η προαιρετική ομαδοποίηση τύπων δεδομένων. Μια τέτοια συνηθισμένη ομαδοποίηση είναι η

"text_general". Δηλώνεται η κλάση του και μπορούν να χρησιμοποιηθούν φίλτρα πεζών γραμμάτων, κοινών λέξεων, συνώνυμων και ότι άλλο θεωρείται απαραίτητο από την ομάδα ανάπτυξης της μηχανής αναζήτησης. Τα παραπάνω μπορούν να χρησιμοποιηθούν είτε στο ευρετήριο είτε στα ερωτήματα. Στην παρακάτω εικόνα γίνεται χρήση φίλτρων και στο ευρετήριο και στο ερώτημα.

```
<fieldType name="text_general" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />
    <filter class="solr.SynonymGraphFilterFactory" synonyms="index_synonyms.txt" ignoreCase="true" expand="false"/>
    <filter class="solr.FlattenGraphFilterFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />
    <filter class="solr.SynonymGraphFilterFactory" synonyms="synonyms.txt" ignoreCase="true" expand="true"/>
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
</fieldType>
```

Στο αρχείο stopwords.txt ορίζονται λέξεις που είναι τόσο κοινές σε ένα συγκεκριμένο σύνολο δεδομένων που το σύστημα δεν μπορεί να τις χειριστεί με κανέναν χρήσιμο τρόπο, άρα μπορούν να αγνοηθούν, δηλώνοντας την κάθε λέξη ανά γραμμή στο έγγραφο. Στο αρχείο synonyms.txt ορίζονται συνώνυμες λέξεις, πάλι ανά γραμμή. Τα συνώνυμα χωρίζονται μεταξύ τους με κόμμα, πχ "foo,bar,baz", μπορεί να οριστεί και μία λέξη με πολλά συνώνυμα, χωρίς τη χρήση κόμματος πχ "cccfoo => cccbar cccbaz". Μία ακόμα δυνατότητα του Solr είναι αυτή των δυναμικών πεδίων, τα οποία είναι χρήσιμα εάν ανακαλύψει η ομάδα ανάπτυξης ότι παρέλειψε να ορίσει ένα ή περισσότερα πεδία. Τα δυναμικά πεδία μπορούν να κάνουν την εφαρμογή λιγότερο εύθραυστη παρέχοντας κάποια ευελιξία στα έγγραφα που μπορούν να προστεθούν στο Solr. Ένα δυναμικό πεδίο είναι ακριβώς όπως ένα κανονικό πεδίο, εκτός από το ότι έχει ένα όνομα με ένα μπαλαντέρ μέσα. Όταν γίνεται ευρετηρίαση εγγράφων, ένα πεδίο που δεν ταιριάζει με κανένα ρητά καθορισμένο πεδίο μπορεί να αντιστοιχιστεί με ένα δυναμικό πεδίο. Όπως τα κανονικά πεδία, τα δυναμικά πεδία έχουν όνομα, τύπο πεδίου και επιλογές. Οι δυνατότητες του Solr δεν τελειώνουν, υπάρχει η επιλογή αντιγραφής πεδίων και μια κοινή χρήση αυτής της λειτουργικότητας είναι η δημιουργία ενός ενιαίου πεδίου αναζήτησης που θα χρησιμεύει ως το προεπιλεγμένο πεδίο ερωτήματος όταν οι χρήστες δεν καθορίζουν ένα πεδίο για ερώτημα. Για παράδειγμα, ο τίτλος, ο συγγραφέας, οι λέξεις-κλειδιά και το σώμα μπορεί να είναι όλα πεδία που θα πρέπει να αναζητηθούν από προεπιλογή, με κανόνες πεδίου αντιγραφής για κάθε πεδίο για αντιγραφή σε ένα πεδίο "catchall" που θα μπορούσε να πάρει οποιοδήποτε όνομα. Αργότερα, μπορεί να οριστεί ένας κανόνας στο solrconfig.xml για αναζήτηση στο πεδίο catchall από προεπιλογή. Το μόνο μειονέκτημα σε αυτό είναι ότι το ευρετήριό θα αυξάνεται όταν χρησιμοποιούνται πεδία αντιγραφής. Ωστόσο, εάν αυτό γίνει προβληματικό θα εξαρτηθεί από τον αριθμό των πεδίων που αντιγράφονται, τον αριθμό των πεδίων προορισμού που αντιγράφονται, την ανάλυση

που χρησιμοποιείται και τον διαθέσιμο χώρο στο δίσκο. Στο παρακάτω παράδειγμα γίνεται ένωση τεσσάρων πεδίων σε ένα πεδίο με όνομα "text":

```
<!-- Text fields from SolrCell to search by default in our catch-all field -->
<copyField source="title" dest="text"/>
<copyField source="inventor" dest="text"/>
<copyField source="applicant" dest="text"/>
<copyField source="description" dest="text"/>
```

Πράγματι, το Solr είναι ένα εργαλείο που κάνει την μηχανή αναζήτησης όσο καλή όσο είναι η φαντασία ή ο χρόνος που έχει διαθέσιμο η ομάδα ανάπτυξης. Οι περισσότερες από από αυτές τις τεχνικές έχουν αξιοποιηθεί για τη δημιουργία του ευρετηρίου της συλλογής Clef-IP.

3.2.3 Διαμόρφωση του solrconfig.xml

Το αρχείο solrconfig.xml είναι το αρχείο διαμόρφωσης με τις περισσότερες παραμέτρους που επηρεάζουν το ίδιο το Solr. Στο solrconfig.xml, ρυθμίζονται σημαντικές δυνατότητες όπως:

- Χειριστές αιτημάτων (request handlers), οι οποίοι επεξεργάζονται τα αιτήματα προς Solr, όπως αιτήματα για προσθήκη εγγράφων στο ευρετήριο ή αιτήματα για επιστροφή αποτελεσμάτων για ένα ερώτημα.
- Ακροατές (listeners), διαδικασίες που "ακούν" για συγκεκριμένα συμβάντα που σχετίζονται με ερωτήματα. οι ακροατές μπορούν να χρησιμοποιηθούν για την ενεργοποίηση της εκτέλεσης ειδικού κώδικα, όπως η επίκληση ορισμένων κοινών ερωτημάτων σε προσωρινές μνήμες προθέρμανσης (warm-up caches).
- Το Request Dispatcher για τη διαχείριση των επικοινωνιών HTTP.
- Τη διεπαφή ιστού διαχειριστή.
- Παράμετροι που σχετίζονται με την αναπαραγωγή και την αντιγραφή.

Ξανά, ενδείκνυται η χρήση του solrconfig.xml από παράδειγμα του Solr προς αποφυγή χρονοβόρας διαδικασίας και χρήση έτοιμων λειτουργιών που θα χρησιμοποιούνταν έτσι κι αλλιώς. Οι προεπιλογές συνήθως αρκούν για πολλές από τις λειτουργίες του Solr, όπως τα όρια προσωρινής μνήμης ή μέγιστοι χρόνοι αναμονής (πχ για ένα κλείδωμα εγγραφής). Το μέγεθος της cache ορίζεται επίσης στο solrconfig.xml για τα φίλτρα, ερωτήματα και τα έγγραφα. Το μέγεθος μπορεί να οριστεί ανάλογα με τη διαθεσιμότητα της RAM στο μηχάνημα που θα εκτελείται το Solr. Αν η ομάδα ανάπτυξης μπορεί να διαθέσει πολύ από τη διαθέσιμη RAM για το Solr, τότε η μηχανή αναζήτησης επιστρέφει αποτελέσματα ερωτημάτων πολύ πιο γρήγορα καθώς δεν χρειάζεται να πραγματοποιήσει τα ίδια ερωτήματα ξανά και ξανά. Ο πιο τυπικός τρόπος με τον οποίο χρησιμοποιεί το Solr το filterCache είναι να αποθηκεύει προσωρινά τα αποτελέσματα κάθε παραμέτρου αναζήτησης fq, αν και υπάρχουν και κάποιες άλλες περιπτώσεις. Τα επόμενα ερωτήματα που χρησιμοποιούν το ίδιο ερώτημα φίλτρου παραμέτρων έχουν ως αποτέλεσμα επισκέψεις στην cache και γρήγορες επιστροφές αποτελεσμάτων. Η cache queryResultCache περιέχει τα αποτελέσματα

προηγούμενων αναζητήσεων: ταξινομημένες λίστες αναγνωριστικών εγγράφων (ID) με βάση ένα ερώτημα, μια ταξινόμηση και το εύρος των εγγράφων που ζητήθηκαν. Και τέλος, η cache εγγράφων documentCache περιέχει αντικείμενα Lucene Document (τα αποθηκευμένα πεδία για κάθε έγγραφο). Μπορεί επίσης να οριστεί cache από την ομάδα ανάπτυξης της μηχανής αναζήτησης για τον δικό της κώδικα εφαρμογής.

```
<cache name="myUserCache" class="solr.LRUCache"
       size="4096"
       initialSize="1024"
       autowarmCount="1024"
       regenerator="org.mycompany.mypackage.MyRegenerator" />
```

Ένας χειριστής αιτημάτων επεξεργάζεται αιτήματα που έρχονται στο Solr. Αυτά μπορεί να είναι αιτήματα ερωτημάτων ή αιτήματα ενημέρωσης ευρετηρίου. Πιθανότατα θα χρειαστούν πολλά από αυτά καθορισμένα, ανάλογα με τον τρόπο με τον οποίο υπάρχει ανάγκη να χειρίζεται το Solr τα διάφορα αιτήματα. Ένα στοιχείο αναζήτησης (search component) είναι ένα χαρακτηριστικό της αναζήτησης, όπως η επισήμανση ή η κατηγοριοποίηση (faceting). Το στοιχείο αναζήτησης ορίζεται στο solrconfig.xml χωριστά από τους χειριστές αιτημάτων και στη συνέχεια, καταχωρείται σε ένα χειριστή αιτημάτων, όπως απαιτείται. Κάθε χειριστής αιτημάτων ορίζεται με ένα όνομα και μια κλάση. Το όνομα του χειριστή αιτημάτων αναφέρεται με το αίτημα προς Solr, συνήθως ως διαδρομή. Για παράδειγμα, εάν το Solr είναι εγκατεστημένο στη διεύθυνση http://localhost:8983/solr/ και υπάρχει μια συλλογή με το όνομα "clefip", μπορεί να υποβληθεί ένα αίτημα που μοιάζει με αυτό: http://localhost:8983/solr/clefip/select?id:EP-0924729

Αυτό το ερώτημα θα υποβληθεί σε επεξεργασία από τον χειριστή αιτημάτων με το όνομα /select. Χρησιμοποιήθηκε μόνο η παράμετρος "q" εδώ, η οποία περιλαμβάνει τον όρο του ερωτήματός, μια απλή λέξη-κλειδί "EP-0924729". Εάν ο χειριστής αιτημάτων έχει καθορισμένες περισσότερες παραμέτρους, αυτές θα χρησιμοποιηθούν με οποιοδήποτε ερώτημα σταλθεί σε αυτόν τον χειριστή αιτημάτων, εκτός εάν παρακαμφθούν από το client (ή τον χρήστη) στο ίδιο το ερώτημα. Εάν έχει οριστεί κι άλλος χειριστή αιτημάτων, μπορεί να σταλεί αίτημα με αυτό το όνομα. Για παράδειγμα, το /update είναι ένας χειριστής αιτημάτων που χειρίζεται ενημερώσεις ευρετηρίου (δηλαδή, αποστολή νέων εγγράφων στο ευρετήριο). Από προεπιλογή, το /select είναι ένας χειριστής αιτημάτων που χειρίζεται αιτήματα ερωτημάτων. Ο κύριος χειριστής αιτημάτων που ορίζεται με το Solr από προεπιλογή είναι το "SearchHandler", το οποίο χειρίζεται τα ερωτήματα αναζήτησης. Ο χειριστής αιτημάτων ορίζεται και, στη συνέχεια, ορίζεται μια λίστα προεπιλογών.

```
<requestHandler name="/select" class="solr.SearchHandler">
   <lst name="defaults">
     <str name="echoParams">explicit</str>
     <int name="rows">10</int>
   </lst>
  </requestHandler>
```

Αυτό το παράδειγμα ορίζει την παράμετρο rows, η οποία ορίζει πόσα αποτελέσματα αναζήτησης θα επιστρέψουν, στο 10. Η παράμετρος echoParams ορίζει ότι οι παράμετροι που ορίζονται στο ερώτημα πρέπει να επιστρέφονται όταν επιστρέφονται πληροφορίες εντοπισμού σφαλμάτων. Σημειώνεται επίσης ότι ο τρόπος με τον οποίο ορίζονται οι προεπιλογές στη λίστα ποικίλλει εάν η παράμετρος είναι αλφαριθμητικός, ακέραιος ή άλλος τύπος. Εκτός από τις προεπιλογές, υπάρχουν και άλλες επιλογές για το SearchHandler, οι οποίες είναι: "appends" επιτρέπει τον ορισμό των παραμέτρων που προστίθενται στο ερώτημα χρήστη. Αυτά μπορεί να είναι ερωτήματα φίλτρου ή άλλοι κανόνες ερωτήματος που θα πρέπει να προστεθούν σε κάθε ερώτημα. Δεν υπάρχει μηχανισμός στο Solr που να επιτρέπει σε ένα client να παρακάμπτει αυτές τις προσθήκες, επομένως η ομάδα ανάπτυξης πρέπει να είναι απολύτως βέβαιη ότι πρέπει πάντα να εφαρμόζονται αυτές οι παράμετροι σε ερωτήματα. Η επιλογή "invariants" επιτρέπει τον ορισμό παραμέτρων που δεν μπορούν να παρακαμφθούν από ένα client. Οι τιμές που ορίζονται σε μια ενότητα invariants θα χρησιμοποιούνται πάντα ανεξάρτητα από τις τιμές που καθορίζονται από τον χρήστη, από το client, στις προεπιλογές ή στα appends. Το τελευταίο τμήμα του ορισμού ενός χειριστή αιτημάτων είναι τα στοιχεία, τα οποία καθορίζουν μια λίστα στοιχείων αναζήτησης που μπορούν να χρησιμοποιηθούν με ένα χειριστή αιτημάτων. Η ενότητα <initParams> του solrconfig.xml επιτρέπει να οριστούν παράμετροι των χειριστών αιτημάτων εκτός της διαμόρφωσης του χειριστή. Υπάρχουν μερικές περιπτώσεις χρήσης όπου αυτό μπορεί να είναι επιθυμητό: ορισμένοι χειριστές ορίζονται σιωπηρά (implicitly) στον κώδικα και θα πρέπει να υπάρχει ένας τρόπος να προστεθούν/προσαρτηθούν/παρακάμψουν ορισμένες από τις σιωπηρά καθορισμένες ιδιότητες. Υπάρχουν μερικές ιδιότητες που χρησιμοποιούνται σε όλους τους χειριστές. Αυτό βοηθά να διατηρηθεί μόνο ένας ορισμός αυτών των ιδιοτήτων και να εφαρμοστούν σε πολλούς χειριστές. Για παράδειγμα, εάν υπάρχει η ανάγκη πολλών από τους χειριστές αναζήτησής να επιστρέφουν την ίδια λίστα πεδίων, μπορεί να δημιουργηθεί μια ενότητα <initParams> χωρίς να χρειάζεται να οριστεί το ίδιο σύνολο παραμέτρων σε κάθε ορισμό του χειριστή αιτημάτων. Εάν υπάρχει ένα μόνο πρόγραμμα χειρισμού αιτημάτων που θα πρέπει να επιστρέψει διαφορετικά πεδία, μπορούν να οριστούν παράμετροι που υπερισχύουν σε μεμονωμένες ενότητες <requestHandler> ως συνήθως. Οι ιδιότητες και η διαμόρφωση μιας ενότητας <initParams> αντικατοπτρίζουν τις ιδιότητες και τη διαμόρφωση ενός χειριστή αιτημάτων. Μπορεί να περιλαμβάνει ενότητες για προεπιλογές, appends και invariants, όπως οποιοσδήποτε χειριστής αιτημάτων.

3.2.4 Ευρετηριοποίηση συλλογής

Εν τέλει, η συλλογή είναι έτοιμη προς ευρετηριοποίηση, ώστε τα έγγραφά της να είναι διαθέσιμα προς αναζήτηση χρησιμοποιώντας το Solr και σε αργότερο στάδιο και το PerFedPat. Η διαδικασία στο στάδιο αυτό, απαιτεί μονάχα κάποιο χρόνο εφόσον υπάρχουν

έτοιμα τα έγγραφα, το σχήμα αλλά και τα εργαλεία. Ανεξαρτήτως του λογισμικού που χρησιμοποιεί ο χρήστης, με μία μονάχα εντολή είναι εφικτή η δημιουργία ολόκληρου του ευρετηρίου, δεδομένου ότι δεν έχει υπάρχει κάποιο σφάλμα στα έγγραφα. Στην προκειμένη περίπτωση χρησιμοποιήθηκε το "SimplePostTool" σε λογισμικό Windows, το οποίο περιέχεται με τη λήψη του Solr, στο μονοπάτι "example/exampledocs". Σε λογισμικό Linux, το Solr περιέχει script ικανό να ευρετηριοποιήσει τη συλλογή. Το εργαλείο προαπαιτεί μονάχα εγκατεστημένη Java αλλά και να έχει εκκινηθεί ο Solr server. Για την εκκίνηση του server, μπορεί να χρησιμοποιηθεί οποιοδήποτε τερματικό, εκτελώντας την εντολή "cd file_path/bin" όπου file_path ορίζεται το πλήρες μονοπάτι διαδρομής που είναι αποθηκευμένο το Solr και έπειτα με την εκτέλεση της εντολής "solr start", το Solr ξεκινάει (σε λογισμικό linux "./solr start"). Το μόνο που χρειάζεται η εντολή δημιουργίας ευρετηρίου είναι το όνομα της συλλογής (πυρήνα) που έχει δημιουργηθεί στον Solr server και ο φάκελος όπου βρίσκονται τα έγγραφα. Παράδειγμα:
java -jar -Dc=collection_name example_path\post.jar example_path\exampledocs*
Ο αστερίσκος δρα ως "μπαλαντέρ" και δίνει την οδηγία στο εργαλείο να ευρετηριοποιήσει όλα τα έγγραφα από το μονοπάτι που ορίστηκε στην εντολή. Διαφορετικά, θα μπορούσε να οριστεί και ο τύπος αρχείων ως εξής "*.xml", ή σε περίπτωση που ενδιαφέρει τον χρήστη η δοκιμή ενός εγγράφου, αρκεί να γραφεί το πλήρες όνομα του πχ "example.xml". Μετά από κάποια αναμονή και εφόσον δεν υπάρχει κάποιο σφάλμα στα έγγραφα, τότε όλες οι πατέντες της συλλογής θα ευρετηροποιηθούν, όπου σε αυτό το σημείο είναι εφικτή η δημιουργία οποιουδήποτε ερωτήματος και η λήψη αποτελεσμάτων γίνεται μέσα σε δευτερόλεπτα (κλάσματα δευτερολέπτου σε μικρές συλλογές), με το μόνο μειονέκτημα ότι ως ευρετήριο, τα έγγραφά παίρνουν περισσότερο αποθηκευτικό χώρο συγκριτικά με πριν.

3.3 Ερώτημα προς το ευρετήριο

Το ερώτημα προς το Solr γίνεται πολύ εύκολα και η απάντηση επιστρέφεται αρκετά γρήγορα. Ας υποθέσουμε ένα ερώτημα προς τον server που τρέχει τοπικά στον υπολογιστή και ότι δεν έχει αλλαχτεί η πόρτα του Solr, η οποία από προεπιλογή είναι η 8983 (ενδέχεται να διαφέρει ανάλογα την έκδοση του Solr). Σε οποιονδήποτε περιηγητή ιστού το σύνδεσμο, γράφοντας:
http://localhost:8983/solr/clefip/select?q=id:EP-0924729
Το Solr αναζητά το έγγραφο με id EP-0924729 και αν υπάρχει στη συλλογή, επιστρέφει όλα τα δεδομένα της στο πεδίο response, στην αντίθετη περίπτωση, αν δεν υπάρχει, επιστρέφει το response άδειο.


```
"response":{"numFound":1,"start":0,"numFoundExact":true,"docs":[
    {
      "id":"EP-0924729",
      "release_date":"19990623",
      "content_type":"C22C-5/06 H01H-33/18 H01H-33/04 H01H-33/66 H01H-33/66 H01H-33/664 C22C-38/00 ",
      "title":["electrode arrangement of vacuum circuit breaker with magnetic member for longitudinal magnetization"]
```

Περαιτέρω ανάλυση του συνδέσμου:

- localhost: η διεύθυνση του solr server
- 8983: η πόρτα του solr server
- solr: το όνομα που ορίζεται από προεπιλογή για το solr
- clefip: το όνομα της συλλογής που δημιουργήθηκε σε προηγούμενο στάδιο
- select: ο RequestHandler του Solr, μπορούν να χρησιμοποιηθούν μόνο όσοι είναι δηλωμένοι στο solrconfig.xml
- q: δηλώνει query (ερώτημα), διαφορετικά fq: το οποίο δηλώνει filtered query

Είναι εφικτό να δηλωθούν κι άλλοι παράμετροι, παραθέτονται οι πιο συνηθισμένοι:

Παράμετρος	Περιγραφή
sort	Ταξινομεί την απάντηση σε ένα ερώτημα σε αύξουσα ή φθίνουσα σειρά με βάση τη βαθμολογία της απάντησης ή άλλο καθορισμένο χαρακτηριστικό.
rows	Ελέγχει πόσες σειρές απαντήσεων εμφανίζονται κάθε φορά (προεπιλεγμένη τιμή: 10).
timeAllowed	Καθορίζει τον χρόνο που επιτρέπεται για την επεξεργασία του ερωτήματος. Εάν παρέλθει ο χρόνος πριν ολοκληρωθεί η απάντηση στο ερώτημα, ενδέχεται να επιστραφούν μερικές πληροφορίες.
df	Καθορίζει ένα προεπιλεγμένο πεδίο, παρακάμπτοντας τον ορισμό του προεπιλεγμένου πεδίου του σχήματος.
fl	Περιορίζει τις πληροφορίες που περιλαμβάνονται σε μια απάντηση ερωτήματος σε μια καθορισμένη λίστα πεδίων. Για παράδειγμα προσθέτοντας στο ερώτημα: &fl=id+title δηλώνεται στο Solr ότι δεν υπάρχει ενδιαφέρον για κάποιο άλλο πεδίο εκτός αυτών των δύο.
start	Καθορίζει μια μετατόπιση (από προεπιλογή, 0) στις απαντήσεις στις οποίες το Solr θα πρέπει να αρχίσει να εμφανίζει περιεχόμενο.
debug=query	Επιστρέφει όλες τις πληροφορίες εντοπισμού σφαλμάτων.

Διάφοροι χαρακτήρες μπορούν να αξιοποιηθούν απευθείας στα ερωτήματα, για την επίτευξη διαφορετικών σκοπών από τον χρήστη:

Χαρακτήρας	Χρησιμότητα
?	Αναζήτηση μονού χαρακτήρα. Με το ερώτημα “te?t”, το Solr θα μπορούσε να επιστρέψει και “test” αλλά και “text”.
*	Αναζήτηση πολλαπλών χαρακτήρων. Με το ερώτημα “tes*”, το Solr

	θα μπορούσε να επιστρέψει οποιοδήποτε από: “test”, “testing”, “tester”.
~	Ασαφή αναζήτηση όρων. Με το ερώτημα “roam~” το Solr κάλλιστα θα μπορούσε να επιστρέψει “roams” ή “foam” ή “foams”. Μπορεί να συνδυαστεί με αναζήτηση εγγύτητας: “roam~1”, ορίζοντας τους πιθανούς χαρακτήρες αποτελέσματος της αναζήτησης.
^	Ορισμός βάρους πεδίων αναζήτησης. Ο χρήστης ενδέχεται να ενδιαφέρεται παραπάνω για έναν από τους όρους αναζήτησης, οπότε μπορεί να ορίσει το “βάρος” μιας λέξης κλειδιού απευθείας στο ερώτημα. Παράδειγμα: “vacuum^5 circuit”, ορίζοντας το “vacuum” 5 φορές πιο σημαντικό για το ερώτημα αναζήτησης από ότι το “circuit”.
+	Απαιτεί ο όρος μετά το σύμβολο + να υπάρχει κάπου σε ένα πεδίο σε τουλάχιστον ένα έγγραφο, προκειμένου το ερώτημα να επιστρέψει μια αντιστοίχιση.
-	Ο χειριστής - εξαιρεί έγγραφα που περιέχουν τον όρο μετά το σύμβολο.

Οι πιθανότητες δεν τελειώνουν, μπορεί να οριστεί εύρος στο ερώτημα, πχ year:[1999 TO 2010], το οποίο θα περιορίσει την αναζήτηση σε έγγραφα με χρονολογία από 1999 μέχρι 2010. Επίσης είναι εφικτό να γίνει συνδυασμός όσων όρων απαιτεί ο χρήστης. Ακολουθεί παράδειγμα: “title:"electrode arrangement" AND text:"the present invention"”, συνδυάζοντας δύο πεδία στην αναζήτηση. Μπορεί να χρησιμοποιηθεί και χωρίς τα ονόματα των πεδίων, δηλαδή: “"electrode arrangement" AND "the present invention"”. Ο τελεστής αντιστοιχίζει έγγραφα όπου υπάρχουν και οι δύο όροι οπουδήποτε στο κείμενο ενός μεμονωμένου εγγράφου. Το σύμβολο && μπορεί να χρησιμοποιηθεί στη θέση της λέξης AND. Αντί για το AND, μπορούν να χρησιμοποιηθούν και τα OR ή NOT. Ο χειριστής NOT εξαιρεί έγγραφα που περιέχουν τον όρο μετά το NOT. Το σύμβολο ! μπορεί να χρησιμοποιηθεί στη θέση της λέξης NOT. Ο χειριστής OR απαιτεί να υπάρχει ένας όρος (ή και οι δύο όροι) στο έγγραφο. Το σύμβολο || μπορεί να χρησιμοποιηθεί στη θέση της λέξης OR.

3.4 Χρησιμοποιώντας το Solr Velocity

Το Velocity είναι ένα ισχυρό εργαλείο ενσωματωμένο στο Solr, που αξιοποιεί πολλές από τις λειτουργίες του, όπως αναζήτηση, κατηγοριοποίηση, επισήμανση, αυτόματη συμπλήρωση και γεωχωρική αναζήτηση. Αξιοποιείται πλήρως στην παρούσα πτυχιακή εργασία για τη δημιουργία μιας διεπαφής ιστού, όπου είναι εφικτή η προβολή όλων των πατεντών της συλλογής Clef-IP και γίνεται αξιοποίηση όλων των χαρακτηριστικών του Solr. Για να γίνει προσβάσιμο, πρέπει να δηλωθεί στο solrconfig.xml ως RequestHandler και από προεπιλογή

είναι προσβάσιμο με τον ίδιο τρόπο που έγινε το ερώτημα πρωτίστως, με τη διαφορά ότι αντί για "/select", ορίζεται "/browse" στο σύνδεσμο: http://localhost:8983/solr/clefip/browse
Φυσικά, για να λειτουργήσει σωστά και να μπορούν να προβληθούν όλα τα δεδομένα των εγγράφων της συλλογής, πρέπει να γίνουν αρκετές αλλαγές. Αρχικά, πρέπει να οριστούν τα σωστά πεδία στα αρχεία που είναι υπεύθυνα για την προβολή των εγγράφων του ευρετηρίου: "hit_plain.vm" που είναι υπεύθυνο για την απλή προβολή των εγγράφων και "richtext_doc.vm", που είναι υπεύθυνο για την λεπτομερή προβολή. Για παράδειγμα, εάν πρέπει να προβληθεί η περιγραφή ενός εγγράφου τότε δηλώνεται στην λεπτομερή προβολή χρησιμοποιώντας το όνομα του πεδίου που ορίστηκε στο σχήμα. Μπορούν να γίνουν διάφορες τροποποιήσεις με βάση την κρίση της ομάδα ανάπτυξης. Στην συγκεκριμένη περίπτωση υπήρχε η ανάγκη για πολύ μεγάλες περιγραφές να φαίνεται μόνο ένα κομμάτι, αλλά να δίνεται και η επιλογή στον χρήστη να εμφανίσει ολόκληρη την περιγραφή. Ο τρόπος για να γίνει αυτό είναι εκμεταλλεύοντας μεθόδους όπως "length" και "substring" για να παρθεί το μέγεθος της περιγραφής και να εμφανιστεί μονάχα ένα κομμάτι της:

```
#set($description = $doc.getFieldValue('description'))
#if($description)
  <div>
        <b>Description:</b> #if($page.results_found > 5 && $description.length() > 1010) $description.substring(0, 1000).... <a href="#url_for_home?q=id:#field('id')">Show More</a> #else $description #end
  </div>
#end
```

Αν η αναζήτηση γύρισε πάνω από 5 σελίδες αποτελεσμάτων και το μέγεθος της περιγραφής είναι πάνω από 1010 χαρακτήρες, τότε το Velocity θα εμφανίσει τους 1000 πρώτους χαρακτήρες και ένα κουμπί "Show More", που πατώντας το θα μεταβιβάσει το χρήστη στη σελίδα του εγγράφου με αυτό το ID. Δυστυχώς από προεπιλογή δε δουλεύουν κάποια χαρακτηριστικά του Velocity, όπως αυτόματη συμπλήρωση αναζήτησης και κατηγοριοποίηση (faceting). Η αυτόματη συμπλήρωση χρειάζεται κάποια βασική τεχνογνωσία στη Javascript και συγκεκριμένα στη βιβλιοθήκη jQuery για να διορθωθεί, ενώ τα πεδία προς κατηγοριοποίηση, πρέπει να οριστούν κατάλληλα στο σχήμα του Solr (managed-schema) αλλά και στο Velocity αρχείο "facet_fields.vm". Στους ερευνητές παρουσιάζονται οι ευρετηριασμένοι όροι, μαζί με αριθμητικές μετρήσεις για το πόσα έγγραφα που ταιριάζουν βρέθηκαν για κάθε όρο. Το faceting διευκολύνει τους χρήστες να εξερευνήσουν τα αποτελέσματα αναζήτησης, περιορίζοντας ακριβώς τα αποτελέσματα που αναζητούν. Παραθέτονται οι πιο συνηθισμένοι παράμετροι που μπορούν να χρησιμοποιηθούν για την αξιοποίηση της κατηγοριοποίησης.

Παράμετρος	Περιγραφή
facet	Εάν οριστεί σε true, ενεργοποιεί τις μετρήσεις κατηγοριών στην απόκριση ερωτήματος. Εάν οριστεί σε false, ή λείπει, η κατηγοριοποίηση απενεργοποιείται. Καμία από τις άλλες παραμέτρους που αναφέρονται παρακάτω δεν θα έχει καμία επίδραση εκτός εάν αυτή η παράμετρος οριστεί σε true. Η προεπιλεγμένη τιμή είναι ψευδής.
facet.query	Επιτρέπει να καθοριστεί ένα αυθαίρετο ερώτημα στην προεπιλεγμένη σύνταξη του Lucene για να δημιουργηθεί ένας αριθμός κατηγοριών. Από προεπιλογή, το Solr καθορίζει αυτόματα τους μοναδικούς όρους για ένα πεδίο και επιστρέφει μια καταμέτρηση για κάθε έναν από αυτούς τους όρους. Με τη χρήση της παραμέτρου είναι δυνατή η παράκαμψη αυτής της προεπιλεγμένης συμπεριφοράς και γίνεται επιλογή ακριβώς των όρων ή εκφράσεων προς καταμέτρηση.
facet.field	Προσδιορίζει ένα πεδίο που πρέπει να αντιμετωπίζεται ως κατηγορία. Μπορεί να καθοριστεί πολλές φορές σε ένα ερώτημα για την επιλογή πεδίων πολλαπλών κατηγοριών.
facet.missing	Εάν οριστεί σε true, αυτή η παράμετρος υποδεικνύει ότι, εκτός από τους περιορισμούς που βασίζονται σε όρους ενός πεδίου κατηγοριών, θα πρέπει να υπολογιστεί και να επιστραφεί στην απόκριση ένας αριθμός όλων των αποτελεσμάτων που ταιριάζουν με το ερώτημα, αλλά δεν έχουν τιμή κατηγορίας για το πεδίο. Η προεπιλεγμένη τιμή είναι ψευδής.
facet.mincount	Καθορίζει τις ελάχιστες μετρήσεις που απαιτούνται για να συμπεριληφθεί ένα πεδίο κατηγορίας στην απόκριση. Εάν οι μετρήσεις ενός πεδίου είναι κάτω από το ελάχιστο, η κατηγορία του πεδίου δεν επιστρέφεται. Η προεπιλεγμένη τιμή είναι 0.
facet.range	Ορίζει το πεδίο για το οποίο το Solr πρέπει να δημιουργήσει κατηγορίες εύρους.
facet.range.start	Καθορίζει το κατώτερο όριο των ευρών.
facet.range.end	Καθορίζει το ανώτερο όριο των ευρών.
facet.range.gap	Το κάθε εύρος εκφράζεται ως τιμή που πρέπει να προστεθεί στο κατώτερο όριο.
facet.range.other	Καθορίζει ότι εκτός από τις μετρήσεις για κάθε περιορισμό εύρους μεταξύ facet.range.start και facet.range.end, θα πρέπει να υπολογίζονται και οι μετρήσεις για αυτές τις επιλογές: before, after, between, all, none

Παράδειγμα αξιοποίησης faceting στο Velocity:

```
<str name="facet">on</str>
<str name="facet.missing">true</str>
<str name="facet.field">content_type</str>
<str name="facet.field">author_s</str>
<str name="facet.query">ipod</str>
<str name="facet.query">GB</str>
<str name="facet.mincount">1</str>
<str name="facet.range.other">after</str>
<str name="facet.range">popularity</str>
<int name="f.popularity.facet.range.start">0</int>
<int name="f.popularity.facet.range.end">10</int>
<int name="f.popularity.facet.range.gap">3</int>
```

Το βάρος του κάθε πεδίου για την αναζήτηση ορίζεται ξεχωριστά για το Velocity μαζί με τη δήλωση του RequestHandler. Στο παρακάτω παράδειγμα το μεγαλύτερο βάρος το έχουν τα id και title πεδία, κρίνοντας ότι είναι τα πιο σημαντικά. Ο χρήστης μπορεί να το παρακάμψει ορίζοντας βάρος στο ερώτημα όπως εκείνος θεωρεί απαραίτητο.

```
<requestHandler name="/browse" class="solr.SearchHandler">
  <lst name="defaults">
    <str name="echoParams">explicit</str>

    <!-- VelocityResponseWriter settings -->
    <str name="wt">velocity</str>
    <str name="v.template">browse</str>
    <str name="v.layout">layout</str>
    <str name="title">CLEF-IP Search</str>

    <!-- Query settings -->
    <str name="defType">edismax</str>
    <str name="qf">
      text^0.5 features^1.0 name^1.2 inventor^1.5 id^10.0 includes^1.1 content_type^1.4
      title^10.0 description^5.0 release_date^5.0 applicant^2.0 resourcename^1.0
    </str>
```

Παρά το γεγονός ότι το Velocity είναι μια βιβλιοθήκη χτισμένη στη Java, δίνονται πολλές επιλογές για να στηθεί με τις προϋποθέσεις του χρήστη, πάντα όμως με τη νοοτροπία ότι είναι ένα Solr template και δε μπορεί να ξεφύγει από τη βασική λειτουργικότητα του Solr.

4. ezDL Framework

4.1 Εισαγωγή στο ezDL

Το ezDL, είναι ένα λογισμικό ανοιχτού κώδικα για δημιουργία υψηλής ποιότητας διαδραστικών διεπαφών χρήστη που βασίζονται σε γνωστικά μοντέλα αναζήτησης καθώς και τις πιο σύγχρονες αρχές σχεδιασμού. Η αρχική ιδέα ήταν η ανάπτυξη ενός νέου τύπου ενοποιημένης μηχανής αναζήτησης: Πρώτον, το σύστημα δεν θα έπρεπε να συνδυάζει μόνο ψηφιακές βιβλιοθήκες, αλλά και άλλα είδη υπηρεσιών ιστού (“agents”) που παρέχουν χρήσιμες λειτουργίες όπως π.χ. προτείνοντας σχετικούς όρους ή μετάφραση του ερωτήματος σε άλλες γλώσσες. Δεύτερον, για βελτιωμένη φιλικότητα προς τον χρήστη, το σύστημα στοχεύει στην εφαρμογή υποστήριξης για γνωστικά μοντέλα αναζήτησης, προορατικότητα και συνεργασία. Το ezDL σχεδιάστηκε με τρεις βασικούς στόχους:
1. Το ezDL είναι ένα ισχυρό εργαλείο frontend που μπορεί εύκολα να ρυθμιστεί για αναζήτηση μια ετερογενής συλλογής ψηφιακών βιβλιοθηκών.
2. Το ezDL είναι μια ευέλικτη και επεκτάσιμη πλατφόρμα λογισμικού που παρέχει μια σταθερή βάση για γραφή προσαρμοσμένων εφαρμογών, τόσο σε λειτουργικό όσο και σε παρουσιαστικό επίπεδο.
3. Το ezDL είναι επίσης ένα πλαίσιο αξιολόγησης που παρέχει μια πλούσια λειτουργικότητα για την εκτέλεση μιας ευρείας ποικιλίας αξιολογήσεων χρηστών.

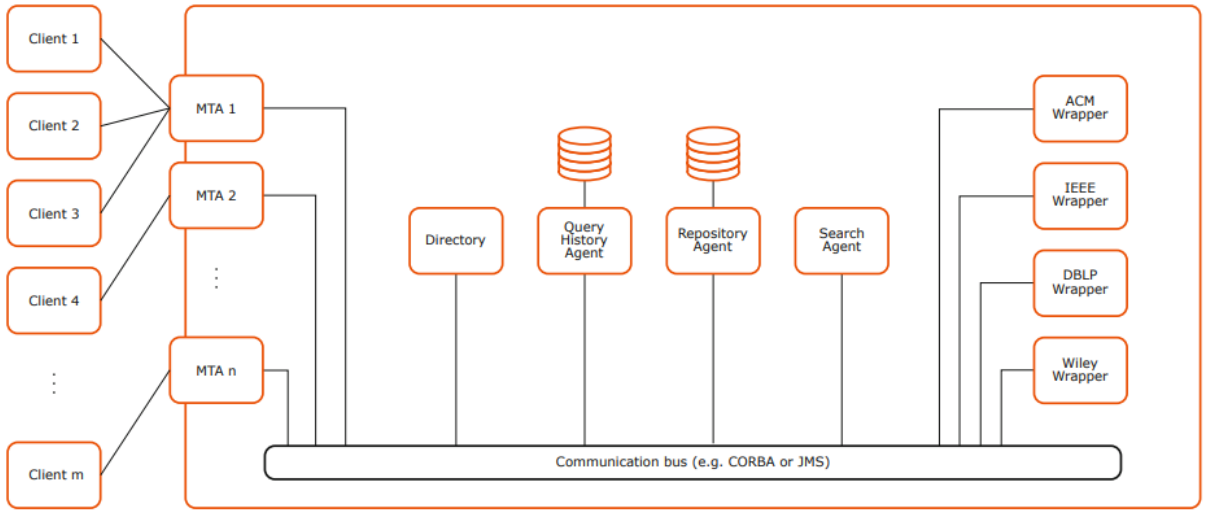
4.2 Αρχιτεκτονική του ezDL

Η αρχιτεκτονική του συστήματος κάνει εκτεταμένη χρήση του διαχωρισμού των ανησυχιών για διατήρηση αλληλεξαρτήσεων στο ελάχιστο και για να κάνουν το σύστημα πιο σταθερό. Αυτό είναι αλήθεια σε επίπεδο συστήματος όπου υπάρχει σαφής διαχωρισμός μεταξύ clients και backend, αλλά και μέσα στο ίδιο το backend, όπου μεμονωμένες διαδικασίες «agent» χειρίζονται συγκεκριμένα μέρη της λειτουργικότητας, ακόμη και εντός αυτών των agents.
Το client, επίσης χωρίζεται σε πολλαπλά ανεξάρτητα στοιχεία που ονομάζονται «εργαλεία».
Το ezDL είναι πλήρως γραμμένο σε Java χρησιμοποιώντας κοινά πλαίσια και βιβλιοθήκες.

4.2.1 Το Backend στο ezDL

Το backend παρέχει ένα μεγάλο μέρος της βασικής λειτουργικότητας του ezDL: τη μετα-αναζήτηση, την εξουσιοδότηση χρήστη, τη βάση γνώσεων σχετικά με τα συλλεγόμενα έγγραφα, καθώς και τους wrappers και τις υπηρεσίες που συνδέονται με εξωτερικές υπηρεσίες. Λειτουργικότητα που παρέχει υποστήριξη συνεργασίας και επιτρέπει την αποθήκευση εγγράφων και ερωτημάτων σε προσωπική βιβλιοθήκη βρίσκεται επίσης εδώ. Τα

στοιχεία του backend είναι agents: ανεξάρτητες διαδικασίες που παρέχουν μια συγκεκριμένη λειτουργικότητα στο σύστημα. Οι agents χρησιμοποιούν έναν κοινό δίαυλο επικοινωνίας για τη μεταφορά μηνυμάτων μεταξύ τους.



Οι agents υποδιαιρούνται σε κύριους (εγγραφή στον Κατάλογο, αποστολή και λήψη μηνυμάτων, διαχείριση πόρων) και σε στοιχεία που ασχολούνται με συγκεκριμένα αιτήματα. Αυτά τα στοιχεία, οι χειριστές αιτημάτων, είναι ανεξάρτητα και επεξεργάζονται αιτήματα ταυτόχρονα.

- MTA (Message Transfer Agent): είναι ένας agent που παρέχει στα clients ένα σημείο σύνδεσης στο backend. Οι MTA είναι υπεύθυνοι για τον έλεγχο ταυτότητας των χρηστών και τη μετάφραση αιτημάτων από clients σε μηνύματα προς ορισμένους agents. Αυτός ο μηχανισμός δημιουργεί έναν σαφή διαχωρισμό μεταξύ της προβολής του client του συστήματος και της εσωτερικής λειτουργικότητας: το client δεν χρειάζεται να γνωρίζει πόσοι agents εξυπηρετούν την αναζήτηση. Τα ερωτήματα και οι νέοι agents αναζήτησης θα μπορούσαν να δημιουργηθούν όπως απαιτείται από τη φόρτωση του συστήματος. Επί του παρόντος, υπάρχει μόνο μία υλοποίηση MTA, η οποία χρησιμοποιεί ένα δυαδικό πρωτόκολλο μέσω μιας σύνδεσης TCP, αλλά είναι δυνατή η παροχή άλλων πρωτοκόλλων με χρήση ξεχωριστών υλοποιήσεων MTA.
- Το Directory είναι ένας ειδικός agent που διατηρεί μια λίστα με τους agents και τις υπηρεσίες που αυτοί παρέχουν. Κατά την έναρξη, κάθε agent εγγράφεται στο Directory και ανακοινώνει τις υπηρεσίες που παρέχει.
- Η διαχείριση της σύνδεσης με απομακρυσμένες ή τοπικές υπηρεσίες αναζήτησης γίνεται από agents που λέγονται wrappers. Μεταφράζουν την εσωτερική αναπαράσταση ερωτήματος του ezDL σε ένα ερώτημα που μπορεί να αναλύσει η

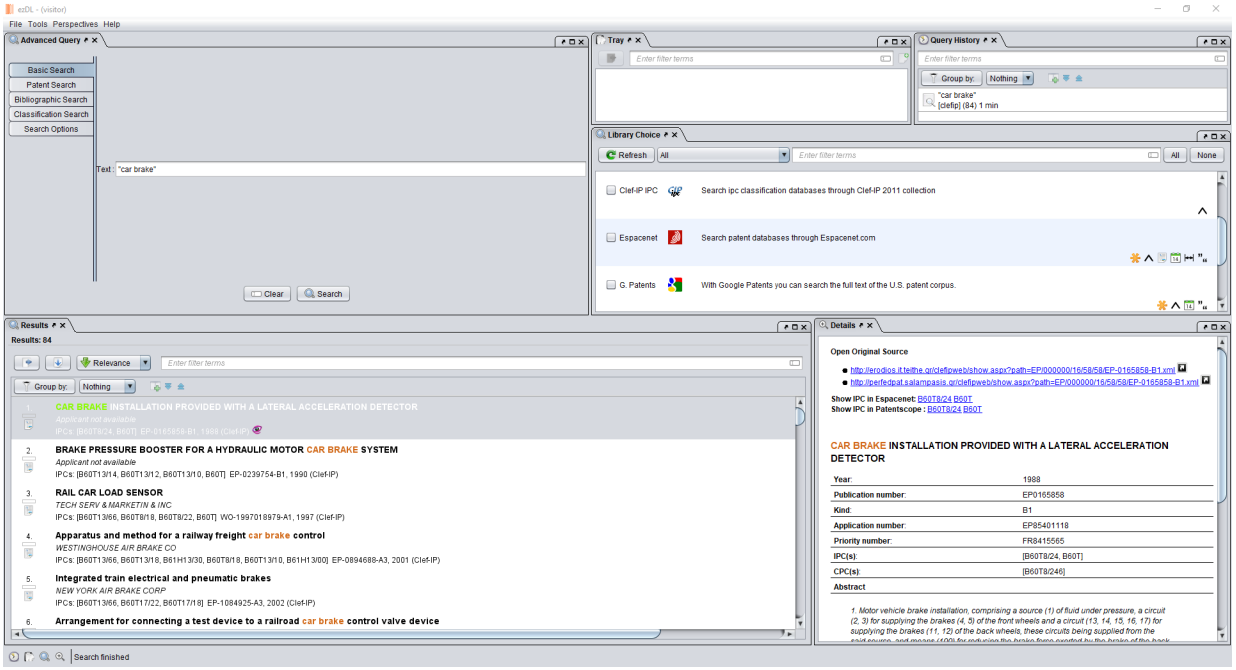
απομακρυσμένη υπηρεσία και μεταφράζει την απάντηση της απομακρυσμένης υπηρεσίας σε ένα κατάλληλο έγγραφο αναπαράστασης για να να το διαχειριστεί το ezDL.

Δεδομένου ότι κάθε είδους λειτουργικότητα αναλαμβάνεται από διαφορετικούς agents, προβλήματα που μπορεί να παρουσιαστούν σε έναν agent, γενικά διαταράσσουν μόνο αυτή τη συγκεκριμένη λειτουργία.

Όταν ένα client κάνει μια αναζήτηση, στέλνει ένα αίτημα στο MTA, με ένα ερώτημα στη σύνταξη ezDL και μια λίστα απομακρυσμένων υπηρεσιών στις οποίες πρέπει να εκτελεστεί το ερώτημα. Το αίτημα διεκπεραιώνεται από το MTA που το προωθεί στον agent αναζήτησης. Ο agent αναζήτησης ζητά από το Directory το όνομα των agents που παρέχουν σύνδεση με τις απομακρυσμένες υπηρεσίες που ζητούνται από το client. Μετά τη λήψη αυτής της λίστας, ο agent αναζήτησης προωθεί το ερώτημα σε καθέναν από αυτούς τους agents. Στη συνέχεια, οι agents μεταφράζουν το ερώτημα σε κάτι που κατανοεί η απομακρυσμένη υπηρεσία και στέλνουν την απάντηση της απομακρυσμένης υπηρεσίας πίσω στον agent αναζήτησης. Ο agent αναζήτησης συλλέγει όλες τις απαντήσεις από όλες τις απομακρυσμένες υπηρεσίες, συγχωνεύει διπλότυπα και τις ανακατατάσσει. Από προεπιλογή, η ανακατάταξη γίνεται χρησιμοποιώντας τις αμοιβαίες αρχικές βαθμίδες από τα αρχικά αποτελέσματα. Στη συνέχεια, το σύνολο απαντήσεων αποστέλλεται πίσω στο MTA που ζήτησε την αναζήτηση. Το MTA αναμεταδίδει την απάντηση στο client. Ο agent αναζήτησης προωθεί επίσης τα συλλεγμένα έγγραφα στον repository agent, ο οποίος είναι υπεύθυνος για την εξυπηρέτηση αιτημάτων για λεπτομέρειες σχετικά με έγγραφα (π.χ. εάν ο χρήστης θέλει να δει το πλήρες κείμενο).

4.2.2 Το Frontend στο ezDL

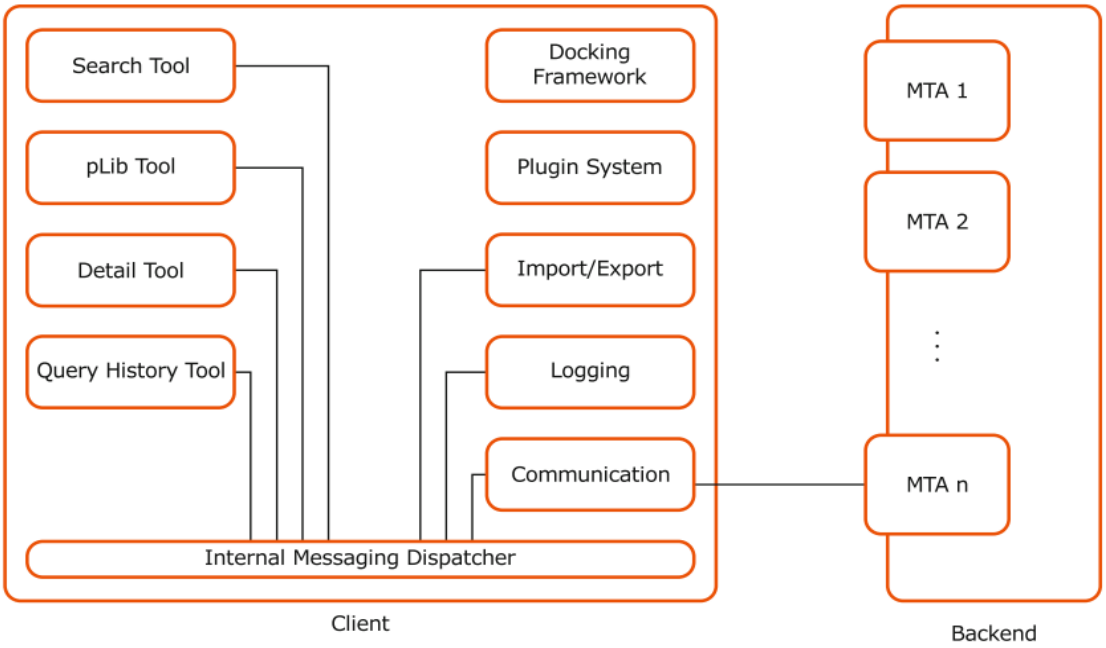
Εργαλεία και Προοπτικές: Ένα εργαλείο περιλαμβάνει ένα σύνολο λογικά συνδεδεμένων λειτουργιών. Κάθε εργαλείο έχει μία ή περισσότερες προβολές εργαλείων, διαδραστικά στοιχεία οθόνης που μπορούν να τοποθετηθούν κάπου στην επιφάνεια εργασίας. Μια διαμόρφωση των διαθέσιμων εργαλείων και η συγκεκριμένη διάταξη των προβολών εργαλείων τους στην επιφάνεια εργασίας ονομάζεται προοπτική. Οι χρήστες μπορούν να τροποποιήσουν υπάρχουσες προκαθορισμένες προοπτικές καθώς και να δημιουργήσουν προσαρμοσμένες προοπτικές.



Το εργαλείο αναζήτησης προσφέρει μια ποικιλία από φόρμες ερωτημάτων για διαφορετικούς σκοπούς και προβολές για την παρουσίαση των αποτελεσμάτων σε μορφή λίστας ή πλέγματος, καθώς και μια προβολή Επιλογής Βιβλιοθήκης για την επιλογή πηγών πληροφοριών. Τα αποτελέσματα μπορούν να ταξινομηθούν ή να ομαδοποιηθούν με διαφορετικά κριτήρια, να φιλτραριστούν και να εξαχθούν. Μια συνάρτηση εξαγωγής μπορεί να χρησιμοποιηθεί για την εξαγωγή συχνών όρων, συγγραφέων ή άλλων χαρακτηριστικών από το αποτέλεσμα και την οπτικοποίηση τους με τη μορφή πίνακα ή "σύννεφου" ετικετών.

Το tray μπορεί να χρησιμοποιηθεί για την προσωρινή συλλογή σχετικών εγγράφων σε μια συνεδρία αναζήτησης. Το Ιστορικό αναζήτησης παραθέτει προηγούμενα ερωτήματα για επαναχρησιμοποίηση και επιτρέπει την ομαδοποίηση κατά ημερομηνία και φιλτράρισμα. Η Προβολή λεπτομερειών εμφανίζει πρόσθετες λεπτομέρειες για μεμονωμένα έγγραφα, όπως μικρογραφίες ή σύντομες περιλήψεις (όπου είναι διαθέσιμες), ή πρόσθετα μεταδεδομένα που δεν περιλαμβάνονται στο υποκατάστατο που εμφανίζεται στη λίστα αποτελεσμάτων. Μπορεί να παρέχεται ένας σύνδεσμος λεπτομερειών για την ανάκτηση του πλήρους κειμένου.

Όπως το backend, το client χρησιμοποιεί μια υποδομή ανταλλαγής μηνυμάτων για επικοινωνία μεταξύ ανεξάρτητων κατά τα άλλα στοιχείων. Για παράδειγμα, εάν ο χρήστης εισάγει ένα ερώτημα στο εργαλείο αναζήτησης και πατήσει το κουμπί «αναζήτηση», αποστέλλεται ένα εσωτερικό μήνυμα το οποίο μεταδίδεται στο backend μέσω του διαύλου επικοινωνίας. Όταν ληφθεί η απάντηση, το μήνυμα δρομολογείται πίσω στο εργαλείο αναζήτησης.


Search Tool
pLib Tool
Detail Tool
Query History Tool
Docking Framework
Plugin System
Import/Export
Logging
Communication
Internal Messaging Dispatcher
Client
MTA 1
MTA 2
MTA n
Backend

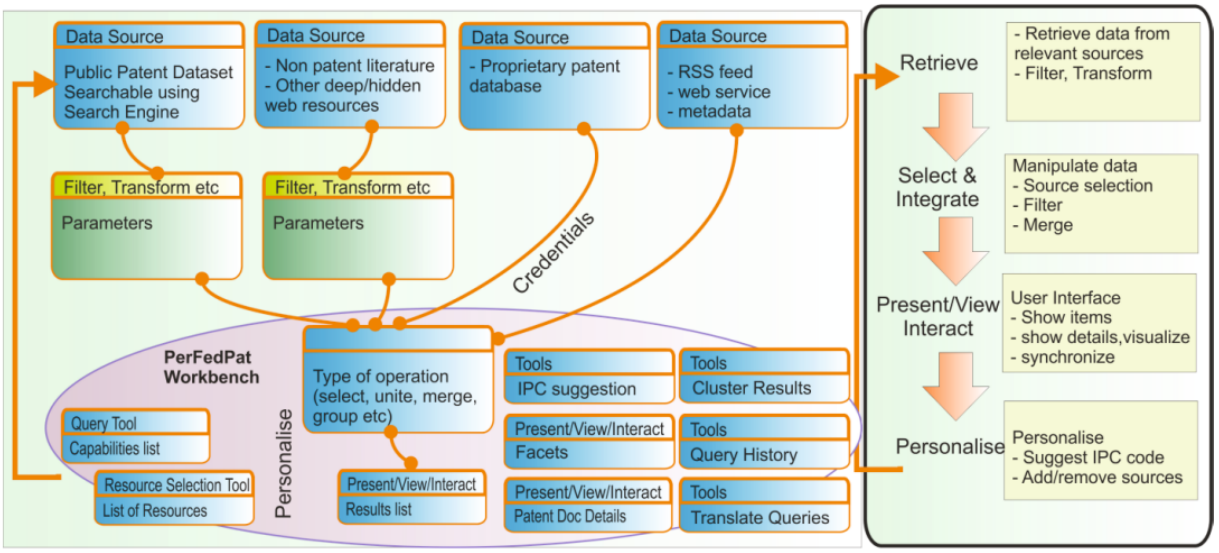
5. PerFedPat

5.1 Εισαγωγή στο PerFedPat

Το PerFedPat, είναι ένα διαδραστικό σύστημα αναζήτησης διπλωμάτων ευρεσιτεχνίας που βασίζεται στην προσέγγιση της ενοποιημένης αναζήτησης και στο πλαίσιο ezDL. Παρέχει βασικές υπηρεσίες αναζήτησης, χρησιμοποιώντας μια ενοποιημένη μέθοδο, πολλαπλών διαδικτυακών πόρων πατεντών, παρέχοντας έτσι παράλληλη πρόσβαση σε πολλαπλές πηγές διπλωμάτων ευρεσιτεχνίας. Το PerFedPat αποκρύπτει την πολυπλοκότητα από τον τελικό χρήστη, που χρησιμοποιεί ένα κοινό εργαλείο μεμονωμένου ερωτήματος για την αναζήτηση όλων των συνόλων δεδομένων πατεντών ταυτοχρόνως. Παρέχει βασικές υπηρεσίες, όπως Boolean και πεδίου αναζήτησης, συγχώνευσης, ομαδοποίησης και φιλτραρίσματος αποτελεσμάτων και προσφέρει υποστήριξη για το ιστορικό ερωτημάτων και τις περιόδους αναζήτησης. Το δεύτερο καινοτόμο χαρακτηριστικό του PerFedPat είναι ότι διαθέτει αρχιτεκτονική με δυνατότητα σύνδεσης και επέκτασης και επιτρέπει τη χρήση πολλαπλών εργαλείων αναζήτησης που είναι ενσωματωμένα στο PerFedPat. Επί του παρόντος, τα ενσωματωμένα εργαλεία είναι για αναζήτηση κατατάξεων IPC, πλοήγηση με πολύπλευρη μορφή, ομαδοποίηση αποτελεσμάτων αναζήτησης, αυτόματη μετάφραση ερωτημάτων, προτάσεις και αποθήκευση συνδέσμων IPC. Ως αποτέλεσμα, το σύστημα είναι σε θέση να παρέχει μια πλούσια, εξατομικευμένη εμπειρία αναζήτησης πληροφοριών για διαφορετικούς τύπους αναζητήσεων πατεντών, αξιοποιώντας πιθανώς τεχνικές από διάφορους τομείς όπως η κατανεμημένη ανάκτηση πληροφοριών, η σημασιολογική αναζήτηση, η μηχανική μάθηση και η αλληλεπίδραση ανθρώπου-υπολογιστή.

5.2 Αρχιτεκτονική του PerFedPat

Χρησιμοποιούνται wrappers που μετατρέπουν το εσωτερικό ερώτημα PerFedPat σε ερωτήματα που μπορεί να επεξεργαστεί κάθε απομακρυσμένο σύστημα. Τα ερωτήματα δρομολογούνται στα συστήματα απομακρυσμένης αναζήτησης και τα επιστρεφόμενα αποτελέσματά τους ανακατατάσσονται εσωτερικά και συγχωνεύονται ως μια ενιαία λίστα που παρουσιάζεται στον ερευνητή διπλωμάτων ευρεσιτεχνίας. Το PerFedPat συνδυάζει τα ακόλουθα στοιχεία: ανάκτηση, επιλογή, ενοποίηση, παρουσίαση και προσαρμογή.



Μερικά από τα βασικά εργαλεία ezDL έχουν προσαρμοστεί ή διαμορφωθεί για την εφαρμογή PerFedPat. Το εργαλείο εξατομικευμένων ερωτημάτων όπως έχει προσαρμοστεί για να ανταποκρίνεται στις ανάγκες πολλών πεδίων στην αναζήτηση διπλωμάτων ευρεσιτεχνίας, το εργαλείο επιλογής βιβλιοθήκης με τα σύνολα δεδομένων διπλωμάτων ευρεσιτεχνίας και τα εργαλεία Αποτελέσματα και Λεπτομέρειες. Επίσης τα Cluster Explorer και τα Entities Explorer εργαλεία που έχουν αναπτυχθεί ειδικά για αναζήτηση διπλωμάτων ευρεσιτεχνίας στο PerFedPat. Τα υπόλοιπα εργαλεία του PerFedPat έχουν δημιουργηθεί για τον πειραματισμο και την βελτίωση αναζήτησης πατεντών.

5.2.1 Εργαλεία σχεδιασμένα για το PerFedPat

- Το εργαλείο προτάσεων IPC στοχεύει, σε ένα ερώτημα, να επιλέξει έναν αριθμό κωδικών IPC, σε διαφορετικά επίπεδα της ιεραρχίας κατάταξης, εάν ζητηθεί, οι οποίοι περιλαμβάνουν πατέντες που σχετίζονται με αυτό το ερώτημα. Η ουσία της μεθόδου είναι ότι προσδιορίζει τους σχετικούς κωδικούς IPC όχι με αναζήτηση της κειμενικής περιγραφής κλάσεων, ομάδων, υποομάδων IPC κλπ., αλλά χρησιμοποιώντας μια έμμεση μέθοδο. Πρώτα ανακτά πατέντες που έχουν ήδη εκχωρηθεί σε κωδικούς IPC και στη συνέχεια δημιουργεί έμμεσα μια εκτίμηση πιθανότητας της συνάφειας των κωδικών IPC που έχουν εκχωρηθεί στο ερώτημα. Στη συνέχεια, το εργαλείο ερωτημάτων ξεκινά μια φιλτραρισμένη αναζήτηση με βάση τους αυτόματα επιλεγμένους κωδικούς IPC. Αυτή η διαδικασία μοιάζει με τον τρόπο με τον οποίο οι επαγγελματίες των διπλωμάτων ευρεσιτεχνίας διεξάγουν διάφορους τύπους αναζήτησης. Επίσης, ο ερευνητής διπλωμάτων ευρεσιτεχνίας μπορεί να χρησιμοποιήσει το εργαλείο όχι μόνο για να παράγει αυτόματα φίλτρα που

βασίζονται σε IPC για να περιορίσει την αναζήτησή του, αλλά και ως αναζήτηση κατάταξης που θα χρησιμοποιηθεί ως αφετηρία για τον εντοπισμό και την πιο προσεκτική εξέταση τεχνικών εννοιών όπως αυτές εκφράζονται σε κωδικούς IPC και με τους οποίους ένα δίπλωμα ευρεσιτεχνίας θα μπορούσε να σχετίζεται.

- Το Cluster Explorer και το Entities Explorer. Το εργαλείο αναζήτησης υποστηρίζει μια διερευνητική στρατηγική για την αναζήτηση διπλωμάτων ευρεσιτεχνίας που εκμεταλλεύεται τα μεταδεδομένα που είναι ήδη διαθέσιμα σε διπλώματα ευρεσιτεχνίας, επιπρόσθετα των αποτελεσμάτων της ομαδοποίησης και της εξόρυξης οντοτήτων που μπορούν να εκτελεστούν τη στιγμή του ερωτήματος. Τα αποτελέσματα μπορούν να συμπληρώσουν τις ταξινομημένες λίστες των διπλωμάτων ευρεσιτεχνίας που παράγονται από τη μηχανή αναζήτησης βασικών διπλωμάτων ευρεσιτεχνίας με πληροφορίες χρήσιμες για τον χρήστη (π.χ. παροχή συνοπτικής επισκόπησης των αποτελεσμάτων αναζήτησης), οι οποίες αξιοποιούνται περαιτέρω σε ένα σχήμα αλληλεπίδρασης που βασίζεται σε συνεδρίες και επιτρέπει στους χρήστες να εστιάζουν τις αναζητήσεις τους σταδιακά και να εναλλάσσουν μεταξύ μεθόδων αναζήτησης, καθώς οι ανάγκες τους σε πληροφορίες καθορίζονται καλύτερα και η κατανόησή τους για το τεχνικό θέμα εξελίσσεται ως απόκριση στις πληροφορίες που βρέθηκαν.
- Το εργαλείο Cluster Explorer παρέχει σε όσους αναζητούν διπλώματα ευρεσιτεχνίας μια επισκόπηση των αποτελεσμάτων που εμφανίζονται στο εργαλείο "Αποτελέσματα". Στοχεύει στην ομαδοποίηση των αποτελεσμάτων σε θέματα (clusters), με προγνωστικά ονόματα (labels), βοηθώντας τον χρήστη να εντοπίσει γρήγορα ένα ή περισσότερα έγγραφα (πατέντες) που διαφορετικά θα ήταν δύσκολο να βρεθούν, ειδικά αν βρίσκονται σε χαμηλή κατάταξη.
- Το εργαλείο Entities Explorer εκτελεί (κατά την ώρα του ερωτήματος) εξόρυξη οντοτήτων στα αποσπάσματα των κορυφαίων αποτελεσμάτων και παρουσιάζει τις αναγνωρισμένες οντότητες ομαδοποιημένες σε κατηγορίες, επιτρέποντας στον χρήστη να περιορίσει τον χώρο αναζήτησης μόνο σε ένα σύνολο των αποτελεσμάτων που περιέχουν μία ή περισσότερες από τις προσδιοριζόμενες οντότητες. Το εργαλείο ομαδοποιεί επίσης τα αποτελέσματα σύμφωνα με τις τιμές των μεταδεδομένων τους. Ο χρήστης ή ο προγραμματιστής μπορεί να καθορίσει τις οντότητες που τον ενδιαφέρουν σε ένα βήμα προεπεξεργασίας, εκμεταλλευόμενος μία ή περισσότερες διαδικτυακές Βάσεις Σημασιολογικών Γνώσεων. Ως εκ τούτου, η διαδικασία εξόρυξης οντοτήτων μπορεί να διαμορφωθεί για διαφορετικά περιβάλλοντα. Οι Γνωσιακές Βάσεις αξιοποιούνται επίσης σε πραγματικό χρόνο για την ανάκτηση περισσότερων σημασιολογικών πληροφοριών σχετικά με μια αναγνωρισμένη

οντότητα που επιτρέπει στον χρήστη να εξερευνήσει τις ιδιότητές της και άλλες σχετικές οντότητες.

- Το εργαλείο Μηχανικής Μετάφρασης χρησιμοποιεί υπηρεσίες MT τρίτων (π.χ. το Bing της Microsoft) προκειμένου να μεταφράσει ερωτήματα σε διαφορετικές γλώσσες, ώστε η Ανάκτηση Πληροφοριών μεταξύ Γλωσσών (CLIR) να μπορεί να πραγματοποιηθεί για την ανάκτηση διπλωμάτων ευρεσιτεχνίας σε περισσότερες γλώσσες. Για να ξεκινήσει μια διαδικασία CLIR, ο χρήστης πρέπει να πατήσει τη μετάφραση και να κάνει ερώτημα. Εάν συμβεί αυτό, το εργαλείο αναζήτησης στέλνει ένα μήνυμα στο εργαλείο αυτόματης μετάφρασης το οποίο με τη σειρά του στέλνει τα κατάλληλα αιτήματα σε μία ή περισσότερες υπηρεσίες MT. Η μετάφραση αποστέλλεται πίσω στο εργαλείο ερωτημάτων και το μεταφρασμένο ερώτημα αποστέλλεται στη συνέχεια στους επιλεγμένους πόρους διπλωμάτων ευρεσιτεχνίας (οι οποίοι με τη σειρά τους δημιουργούν το κατάλληλο αίτημα για ανάκτηση αποτελεσμάτων από αυτούς).

5.3 Προσωπικές συλλογές PerFedPat

Τα Lucene και Solr χρησιμοποιήθηκαν για τη δημιουργία προσωπικών συλλογών. Η Lucene για τη δημιουργία ευρετηρίου εγγράφων (σε συνδυασμό με το Solr) και την παροχή ισχυρών δυνατοτήτων αναζήτησης καθώς και ορθογραφικό έλεγχο, επισήμανση αποτελεσμάτων και προηγμένες δυνατότητες ανάλυσης/χαρακτηρισμού. Το Solr για απάντηση ερωτημάτων που δημιουργήθηκαν από το PerFedPat καθώς και για την δημιουργία μιας διεπαφής ιστού από την οποία μπορούν να σταλούν ερωτήματα προς την υλοποίηση του Lucene.

5.3.1 Διασύνδεση PerFedPat και Solr

Αφού χρησιμοποιήθηκαν οι τεχνολογίες Lucene και Solr για την ανάπτυξη προσωπικών συλλογών (συγκεκριμένα χρησιμοποιώντας τη συλλογή Clef-IP), δημιουργήθηκε ένας wrapper agent στο PerFedPat για αποστολή ερωτημάτων απευθείας από το PerFedPat στον Solr server. Στον wrapper δόθηκαν τα στοιχεία του Solr server (όπως Host και Port) και τα πεδία που χρησιμοποιούνται από το ευρετήριο, ώστε να είναι εφικτή η μετάφραση του ερωτήματος του χρήστη, σε μορφή που είναι κατανοητή από το Solr αλλά και για προβολή/ταξινόμηση των αποτελεσμάτων όταν ληφθεί η απάντηση του ερωτήματος. Το Solr μπορεί να αξιοποιήσει τις περισσότερες δυνατότητες αναζήτησης που είναι εφικτές στο PerFedPat, όπως: μονούς ή πολλαπλούς μπαλαντέρ χαρακτήρες, σύνταξη πλήρης boolean (και για χρονολογίες), πολύπλοκες αναζητήσεις ονομάτων, αναζητήσεις εγγύτητας και φράσεων. Όταν μεταφραστεί το ερώτημα από το PerFedPat, αποστέλλεται το ερώτημα στον Solr server. Το Solr δέχεται το ερώτημα και πραγματοποιεί την αναζήτηση στο ευρετήριο που

δημιουργήθηκε πρωτύτερα. Όταν επιστραφούν τα αποτελέσματα σε μορφή Solr εγγράφων, το PerFedPat τα μετατρέπει σε πληροφορία που είναι αντιληπτή από τα εργαλεία του, ώστε ο χρήστης να μπορεί να προβάλει τα αποτελέσματα στη λίστα της εφαρμογής και να μπορεί να τα επεξεργαστεί χρησιμοποιώντας ένα από τα πολλά εργαλεία του PerFedPat. Εν ολίγοις, ο Solr wrapper, μας επιτρέπει να χρησιμοποιηθεί το Solr ως μία ακόμα πηγή δεδομένων για την ενοποιημένη αναζήτηση του PerFedPat.

6. Διαδικτυακή προσβασιμότητα συλλογής

6.1 Μεταφέροντας τη συλλογή στο διαδίκτυο

Δημιουργήθηκε το ευρετήριο της συλλογής, αξιοποιήθηκε το Velocity για τη δημιουργία μιας διεπαφής ιστού ώστε το σύνολο των εγγράφων της συλλογής να είναι διαθέσιμα σε χρήστες χωρίς τις τεχνικές γνώσεις για δημιουργία ερωτημάτων προς το Solr και δημιουργήθηκε η διασύνδεση του PerFedPat με το Solr σε μορφή wrapper agent, ώστε να γίνει αξιοποίηση της ενοποιημένης αναζήτησης του PerFedPat. Όλη η λειτουργικότητα όμως, παραμένει σε τοπικό επίπεδο, πράγμα που σημαίνει ότι δεν υπάρχει πρόσβαση για κανένα χρήστη από εξωτερικό δίκτυο. Στην προκειμένη περίπτωση, χρησιμοποιήθηκε μηχάνημα με λογισμικό Linux Debian, στο οποίο υπήρχε ήδη έκδοση του PerFedPat από το 2015. Υπήρχαν κάποιες προκλήσεις, όπως ότι το PerFedPat είναι κωδικοποιημένο με Java 7, μια έκδοση που δεν υποστηρίζεται από το Solr που χρησιμοποιήθηκε για τη Clef-IP συλλογή (8.10.0). Η λύση ήταν να περαστεί μια πιο ανανεωμένη έκδοση της Java και να οριστεί στο Solr η χρήση της νεότερης έκδοσης Java. Χρησιμοποιώντας την “apt-get install link_to_java” εντολή στο Debian γίνεται εγκατάσταση της Java πολύ εύκολα. Έπειτα, αρκεί ένας ορισμός μονοπατιού της Java που μόλις εγκαταστάθηκε στο εναρκτήριο script του Solr στο φάκελο bin, με όνομα “solr.in.sh”. Ψάχνοντας για “SOLR_JAVA_HOME” και αφαιρώντας από μπροστά τη δίεση, ώστε να μη θεωρείται πλέον σχόλιο και να πάψει να παίρνει την προεπιλεγμένη τιμή του Solr, ορίζεται το μονοπάτι. Στο ίδιο ακριβώς αρχείο αναζητείται το “SOLR_HOST”, ώστε να οριστεί η δημόσια IP διεύθυνση του μηχανήματος και να είναι πλέον προσπελάσιμο από όλα τα δίκτυα. Αν χρειάζεται αλλαγή πόρτας, γίνεται πάλι από το ίδιο αρχείο. Συνήθως η πόρτα επιλέγεται με βάση τις απαιτήσεις του δικτύου κι αν υπάρχει πρόσβαση στο firewall, σε περίπτωση που δεν υπάρχει πρόσβαση στο firewall, ο αριθμός της πόρτας δίνεται από τον διαχειριστή του συστήματος. Η αλλαγή γίνεται με μια πολύ απλή αναζήτηση στο “SOLR_PORT” και αφαιρώντας ξανά τη δίεση/ορίζοντας τη νέα πόρτα. Η τελευταία ρύθμιση για να είναι προσπελάσιμο το Solr από όλα τα δίκτυα, είναι να οριστεί το Host του Jetty. Το Jetty είναι ο server ιστού που χρησιμοποιείται από το Solr. Το host ορίζεται στην εναρκτήρια εντολή της java από το Solr script. Ευτυχώς, το Solr, παρέχει τη δυνατότητα να οριστεί όποια επιπλέον επιλογή θελήσει ο χρήστης, στην εναρκτήρια εντολή java. Αρκεί μια αναζήτηση στο ίδιο αρχείο για “SOLR_OPTS”, έπειτα σε μια καινούρια γραμμή ορίζεται το Jetty Host, ως εξής:
“SOLR_OPTS="$SOLR_OPTS -Djetty.host=0.0.0.0"”, το “0.0.0.0” επιτρέπει όλες τις IPv4 διευθύνσεις να προσπελάσουν το Jetty server και άρα να αποκτήσουν πρόσβαση στο Solr server.

6.2 Ασφαλίζοντας το Solr

Το Solr παρέχει αρκετές μη ασφαλή λειτουργίες για τη χρήση του, με αποτέλεσμα η πρόσβαση στο διαδίκτυο να μην είναι η καλύτερη επιλογή. Υπάρχουν όμως λύσεις να περιοριστεί η χρήση των μη ασφαλή λειτουργιών μόνο στο διαχειριστή του συστήματος, με τη χρήση ονόματος χρήστη και κωδικού. Για να γίνει αυτό, θα οριστούν στο Jetty server, τμήματα της εφαρμογής που δεν πρέπει να είναι προσπελάσιμα χωρίς τη χρήση διαπιστευτηρίων σύνδεσης. Τα πιο σημαντικά τμήματα που πρέπει να ασφαλιστούν είναι το διαχειριστικό του Solr (/admin) καθώς και η λειτουργία που επιτρέπει αλλαγές στο ευρετήριο (/update). Για να γίνουν αλλαγές στο Jetty server, πρέπει να πάμε στο μονοπάτι που περιέχει τις ρυθμίσεις του: solr_path/server/etc. Αρχικά ανοίγοντας το αρχείο “webdefault.xml” και δημιουργόντας έναν περιορισμό ασφαλείας στο τέλος του αρχείου:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Solr authenticated application</web-resource-name>
    <url-pattern>/update/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>update</role-name>
  </auth-constraint>
</security-constraint>
```

Αν ο χρήστης, δεν έχει το ρόλο update όπως ορίστηκε παραπάνω, δε μπορεί να έχει πρόσβαση στο συγκεκριμένο κομμάτι του Solr. Δημιουργώντας έναν ακόμα περιορισμό για τον admin χρησιμοποιώντας την ίδια ακριβώς λογική και απλά αλλάζοντας το “update” σε “admin”, περιορίζεται η πρόσβαση στο τμήμα του “admin” του Solr. Τελευταίο μέρος του “webdefault.xml”, πρέπει να οριστεί το “login-config” ακριβώς κάτω από τους περιορισμούς:

```
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Test Realm</realm-name>
</login-config>
```

Το real-name που ορίζεται εδώ θα χρειαστεί και στην πορεία και θα πρέπει να οριστεί ακριβώς το ίδιο (στην προκειμένη περίπτωση “Test Realm”), διαφορετικά το Jetty δε θα μπορεί να εκκινηθεί. Ώρα να δημιουργηθεί ένα νέο αρχείο, το οποίο θα περιέχει τα διαπιστευτήρια σύνδεσης, κοινώς το όνομα χρήστη και ο κωδικός για τον κάθε ρόλο που δημιουργήθηκε στους περιορισμούς πρόσβασης. Το αρχείο δημιουργείται στο φάκελο που βρίσκεται το “webdefault.xml”, με όνομα “realm.properties” και το περιεχόμενό του: “testuser: testpassword, admin,update”. Το Jetty υποστηρίζει και κρυπτογράφηση σε περίπτωση που ο χρήστης δεν αισθάνεται άνετα με τον κωδικό να βρίσκεται σε απλό κείμενο

μέσα στο αρχείο. Δημιουργείται ο χρήστης “testuser”, με κωδικό “testpassword” για τους ρόλους “admin” και “update”, φυσικά οι ρόλοι θα μπορούσαν να είναι όσοι δημιουργήθηκαν για τους περιορισμούς ασφαλείας πρωτύτερα αλλά και θα μπορούσαν να υπάρχουν διαφορετικά διαπιστευτήρια για τον κάθε ρόλο. Τώρα, μένει να οριστεί στο Jetty το “realm.properties” και κάθε φορά που είναι να προβληθούν συγκεκριμένα τμήματα του Solr, θα βγαίνει ένα παράθυρο ζητώντας τα στοιχεία του χρήστη. Το τελευταίο βήμα γίνεται στο αρχείο “jetty.xml”, στο ίδιο μονοπάτι. Πηγαίνοντας στο τέλος του αρχείου, αλλά πριν το “</Configure>”, ορίζεται η κλάση της υπηρεσίας Jetty, το όνομα που ορίστηκε νωρίτερα και το όνομα του αρχείου που δημιουργήθηκε.

```
<Call name="addBean">
 <Arg>
  <New class="org.eclipse.jetty.security.HashLoginService">
   <Set name="name">Test Realm</Set>
   <Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/realm.properties</Set>
   <Set name="HotReload">false</Set>
  </New>
 </Arg>
</Call>
```

Το Solr είναι πλέον ασφαλής για χρήση από εξωτερικά δίκτυα!

7. Επίλογος

7.1 Συμπεράσματα

Οι μηχανές αναζήτησης είναι ένα εργαλείο ανεκτίμητης αξίας για την εύρεση πληροφοριών στο διαδίκτυο, σε πραγματικά απίστευτους χρόνους δεδομένου το μέγεθος των πληροφοριών που αναζητούν. Εξυπηρετούν πολλούς κλάδους κι έχουν γίνει κομμάτι της καθημερινότητας των ανθρώπων. Μόνο για τη συγγραφή της πτυχιακής εργασίας χρησιμοποιήθηκε η μηχανή αναζήτησης της Google αμέτρητες φορές. Φανταστείτε πως γίνονταν αναζήτησεις παλιότερα σε βιβλιοθήκες, ψάχνοντας μία μία τις ατελείωτες εγκυκλοπαίδειες, ενώ τώρα σε μερικά κλάσματα δευτερολέπτου έχουμε αποτελέσματα που μας αφορούν κι αν δε μας αφορούν είναι πολύ εύκολο να τροποποιήσουμε το ερώτημα για καλύτερη αναζήτηση. Άρα είναι απόλυτα κατανοητό πως η ενοποιημένη αναζήτηση σε διπλώματα ευρεσιτεχνίας εξυπηρετεί ερευνητές σώζοντας τους πολύτιμο χρόνο και κάνοντας τη δουλειά τους πολύ ευκολότερη. Το Solr επιτρέπει τη δημιουργία νέου ευρετηρίου με πλούσια χαρακτηριστικά και επιλογές αξιοποιώντας τη βιβλιοθήκη Lucene στο έπακρο. Η δημιουργία ερωτημάτων προς το Solr δεν απαιτεί μεγάλες τεχνικές γνώσεις, ενώ για κάποιον που δεν διαθέτει καθόλου, το Velocity είναι μια πολύ καλή εναλλακτική αν κι έχει μεγαλύτερες απαιτήσεις για να δουλέψει σωστά συγκριτικά με το Solr. Το PerFedPat είναι ένα πολύ επεκτάσιμο σύστημα με πολλές λειτουργίες, αξιοποιώντας την ενοποιημένη αναζήτηση για ταυτόχρονη αναζήτηση σε πολλαπλές πηγές. Η δημιουργία ενός wrapper agent είναι σχετικά απλή λόγω του documentation που υπάρχει διαθέσιμο και τη χρήση των διαφόρων σχολίων από τους δημιουργούς του. Είναι ένα σύστημα το οποίο αξίζει να ανανεώνεται συνεχώς ώστε να μην παύει η λειτουργία συγκεκριμένων wrappers, όπως είχε γίνει τώρα από την τελευταία του ενημέρωση. Ο λόγος που χρειάζεται σχετικά συχνή ενημέρωση είναι ότι οι πηγές που χρησιμοποιεί για την ενοποιημένη αναζήτηση κάνουν αλλαγές με αποτέλεσμα το PerFedPat να μην επιστρέφει καθόλου αποτελέσματα από τις συγκεκριμένες πηγές ή να είναι ελλιπή, για παράδειγμα να λείπουν πληροφορίες που παλαιότερα θα ήταν διαθέσιμες. Η σχετικά συχνή ενημέρωση που χρειάζεται σε ανταλλαγή με αυτά που παρέχει φαίνεται να αξίζει απόλυτα, καθώς δεν υπάρχει άλλο σύστημα όμοιο του.

Βιβλιογραφία

Internet Site

https://www.investopedia.com/terms/p/patent.asp

https://www.upcounsel.com/what-is-the-purpose-of-a-patent

https://www.findlaw.com/smallbusiness/intellectual-property/types-of-patents.html

https://en.wikipedia.org/wiki/Search_engine

https://en.wikipedia.org/wiki/Federated_search

https://www.algolia.com/blog/product/federated-search-types

https://www.nibusinessinfo.co.uk/content/what-search-engine-and-how-do-they-work

https://www.is.inf.uni-due.de/bib/pdf/ir/Beckers_etal_14.pdf

https://cordis.europa.eu/docs/results/275/275522/final1-executivesummarypictures.pdf

https://en.wikipedia.org/wiki/Apache_Lucene

https://en.wikipedia.org/wiki/Apache_Solr

https://solr.apache.org/guide/8_10