ListCreator: Κατάταξη Οντοτήτων στο Διαδίκτυο

Αλέξανδρος Κομνηνός
Τμήμα Ηλεκτρολόγων Μηχ. και Μηχ. Υπολογιστών
Δημοκρίτειο Πανεπιστήμιο Θράκης
Ξάνθη 67100, Ελλάδα
alexkomn@ee.duth.gr

Αυγερινός Αραμπατζής
Τμήμα Ηλεκτρολόγων Μηχ. και Μηχ. Υπολογιστών
Δημοκρίτειο Πανεπιστήμιο Θράκης
Ξάνθη 67100, Ελλάδα
avi@ee.duth.gr

***Περίληψη* - Η παρακάτω εργασία αφορά την ανάπτυξη μίας διαδικτυακής εφαρμογής κατάταξης οντοτήτων. Ο χρήστης εισάγει στο σύστημα ένα ερώτημα σε φυσική γλώσσα και παίρνει μία λίστα που περιέχει τις δέκα πιο σχετικές οντότητες. Η εφαρμογή χρησιμοποιεί κείμενα από το Διαδίκτυο ως δεδομένα και πραγματοποιεί την εξαγωγή, μορφοποίηση και κατάταξη των οντοτήτων σε πραγματικό χρόνο. Για τον καθορισμό του αλγορίθμου κατάταξης πραγματοποιήθηκε ένα πείραμα όπου δοκιμάστηκαν έξι διαφορετικές εξισώσεις βαθμολόγησης.**

I. ΕΙΣΑΓΩΓΗ

Οι μηχανές αναζήτησης απαντούν στα ερωτήματα των χρηστών επιστρέφοντας μία λίστα από σχετικά κείμενα. Πολλές φορές όμως, οι χρήστες δεν ψάχνουν για κείμενα, αλλά για κάποια συγκεκριμένη πληροφορία που περιέχεται σε αυτά. Αυτή η πληροφορία αποτελεί συχνά ονόματα από οντότητες (για λόγους απλούστευσης αναφέρονται απλά ως οντότητες). Με τον όρο οντότητα εννοούμε οτιδήποτε έχει ξεχωριστή ύπαρξη όπως για παράδειγμα ένα πρόσωπο, μία εταιρεία ή ένα προϊόν μιας εταιρείας. Η ανάγκη αυτή έχει οδηγήσει στην έρευνα για ανάπτυξη συστημάτων που μπορούν να αναγνωρίζουν και να επιστρέφουν ως απάντηση σε ένα ερώτημα οντότητες αντί για ολόκληρα κείμενα.

Το συγκεκριμένο σύστημα αποτελεί μία διαδικτυακή εφαρμογή που δίνει τη δυνατότητα στο χρήστη να πραγματοποιήσει αναζητήσεις για οντότητες που ανήκουν στις κατηγορίες πρόσωπο, τοποθεσία και οργανισμός. Το σύστημα χρησιμοποιεί ως δεδομένα κείμενα από το Διαδίκτυο που έχουν κοινό θέμα το ερώτημα που υποβάλλεται. Η κατάταξη των οντοτήτων πραγματοποιείται χρησιμοποιώντας στατιστικές μεθόδους και αξιοποιώντας την κοινή πληροφορία που περιέχεται σε αυτά τα κείμενα. Το αποτέλεσμα που επιστρέφει είναι μία λίστα από τις δέκα πιο σχετικές οντότητες με το ερώτημα που εισήγαγε ο χρήστης. Η εφαρμογή φιλοξενείται στην ιστοσελίδα http://listcreator.nonrelevant.net/. Για το σκοπό της κατάταξης ήταν υποψήφιες έξι εξισώσεις βασισμένες σε διαφορετικές στατιστικές ποσότητες. Η επιλογή της βέλτιστης εξίσωσης κατάταξης έγινε με τη διεξαγωγή ενός πειράματος.

II. ΣΧΕΤΙΚΈΣ ΕΡΓΑΣΙΕΣ

Το πρόβλημα της κατάταξης οντοτήτων έχει κοινά στοιχεία με την αυτόματη απάντηση ερωτήσεων, όπου μπορεί η απάντηση σε πολλές ερωτήσεις να είναι κάποιο όνομα ή ακόμα και μία λίστα από ονόματα. Τεχνικές που χρησιμοποιούν τμήματα κειμένων από το Διαδίκτυο και εκμεταλλεύονται την κοινή πληροφορία για να εντοπίσουν την απάντηση έχουν δοκιμαστεί και έδωσαν ενθαρρυντικά αποτελέσματα [1]. Το INEX (Initiative for the Evaluation of XML retrieval) εισήγαγε στη θεματολογία του το 2007 την κατάταξη οντοτήτων, και συνέχισε την αξιολόγηση τέτοιων συστημάτων ως το 2009. Στόχος ήταν να γίνει κατάταξη σύμφωνα με ένα ερώτημα σε οντότητες που έχουν δική τους σελίδα στη Wikipedia. Το TREC (Text Retrieval Conference) είχε από το 2009 ως το 2011 κατηγορία με θέμα την κατάταξη οντοτήτων στο Διαδίκτυο. Η προσέγγιση που ακολούθησαν οι περισσότερες ομάδες ήταν να πάρουν ένα σχετικό κείμενο για κάθε υποψήφια οντότητα και ύστερα να εξετάσουν τη συνάφεια του κειμένου με το ερώτημα χρησιμοποιώντας διάφορες μεθόδους ανάκτησης πληροφορίας [2][3].

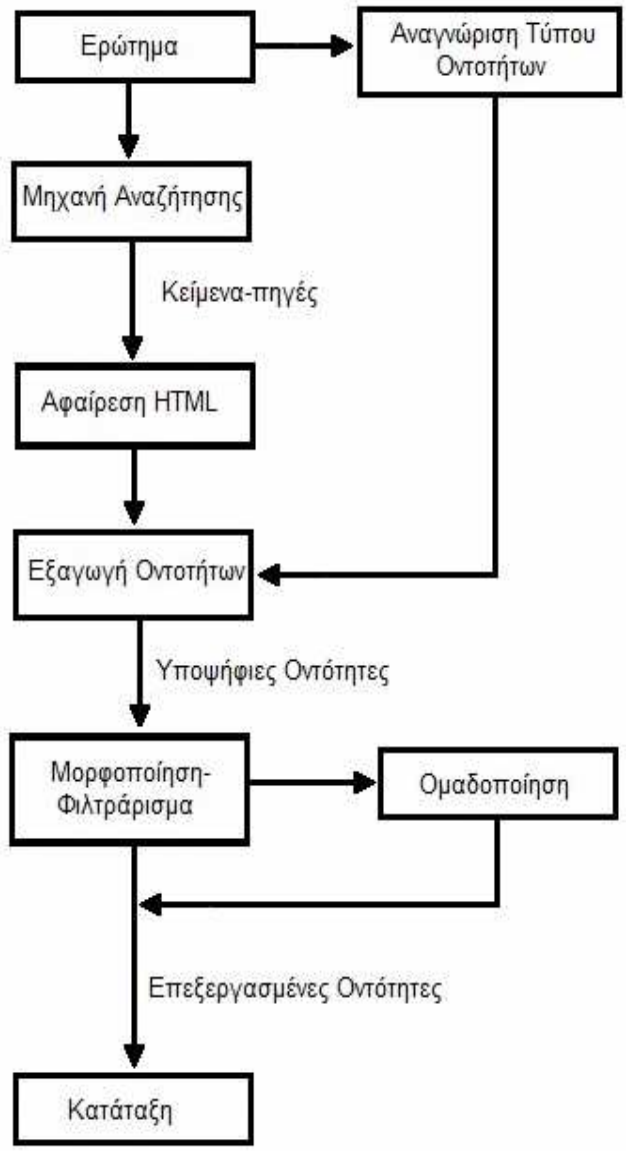
Για την υλοποίηση του παρόντος συστήματος, όπου η κατάταξη πρέπει να γίνεται σε πραγματικό χρόνο, κρίθηκε σκόπιμο να περιοριστεί η χρήση τεχνικών επεξεργασίας φυσικής γλώσσας που έχουν μεγάλο υπολογιστικό κόστος. Έτσι, χρησιμοποιήθηκε ένας αντίστοιχος αλγόριθμος με αυτούς για την αυτόματη απάντηση ερωτήσεων που εκμεταλλεύεται την κοινή πληροφορία μεταξύ πολλών κειμένων. Η κατάταξη σε αυτά τα συστήματα έγινε χρησιμοποιώντας μόνο τη συχνότητα εμφάνισης κάθε υποψήφιας απάντησης. Για τον αλγόριθμο κατάταξης του ListCreator δοκιμάστηκαν επιπλέον κριτήρια που βασίζονται σε μεθόδους ανάκτησης πληροφορίας, ώστε να προσδιοριστεί ο βέλτιστος τρόπος κατάταξης στην περίπτωση των ονομάτων.

III. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η δομή του συστήματος φαίνεται στην Εικόνα 1. Ο κώδικας για τη διασύνδεση των υποσυστημάτων και τα υποσυστήματα για τη μορφοποίηση, το φιλτράρισμα, την ομαδοποίηση και την κατάταξη των οντοτήτων έχουν γραφεί στην αντικειμενοστραφή γλώσσα προγραμματισμού JAVA.

A. Κεντρική Σελίδα της Εφαρμογής

Η κεντρική σελίδα περιέχει μία φόρμα για την εισαγωγή του ερωτήματος και δίνει τη δυνατότητα να ορίσει ο χρήστης την κατηγορία των οντοτήτων (άνθρωποι, τοποθεσίες, οργανισμοί) που αναζητά. Η προεπιλεγμένη επιλογή είναι το auto που αντιστοιχεί σε αυτόματη αναγνώριση της κατηγορίας από το σύστημα. Για την αυτόματη αναγνώριση χρησιμοποιείται μία λίστα από περίπου εκατόν πενήντα λέξεις κλειδιά που υποδηλώνουν αναζήτηση τοποθεσίας ή οργανισμού. Το σύστημα αποφασίζει την κατηγορία των οντοτήτων ανάλογα με τις λέξεις και τη σειρά που εμφανίζονται στο ερώτημα. Αν δεν εμφανιστεί καμία λέξη κλειδί υποθέτει ότι η κατηγορία είναι πρόσωπα Για τη



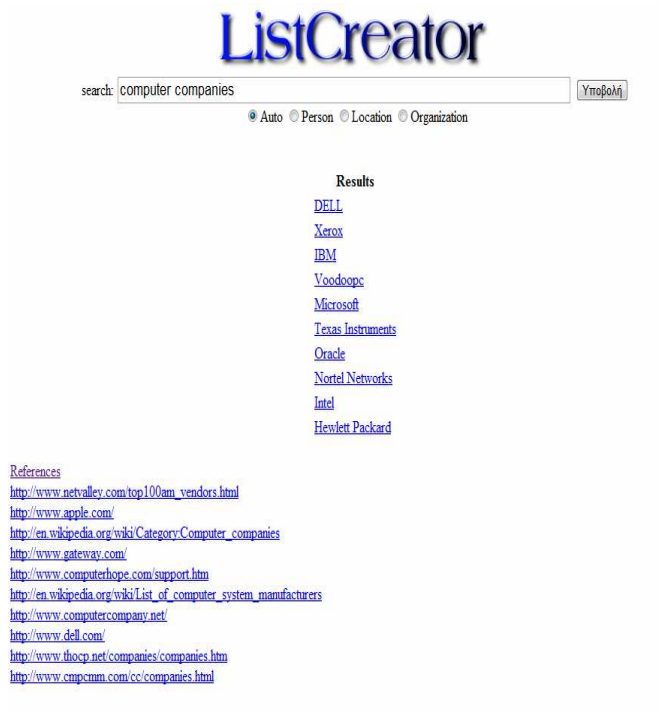
Εικόνα 1

συλλογή των λέξεων χρησιμοποιήθηκαν οι λίστες συνωνύμων του WordNet [4].

Η σελίδα με τα αποτελέσματα δημιουργείται δυναμικά με χρήση της γλώσσας προγραμματισμού PHP. Μόλις γίνει υποβολή ενός ερωτήματος, καλείται η βασική εφαρμογή με την επιθυμητή είσοδο, και η έξοδός της εμφανίζεται στη σελίδα ύστερα από μία επεξεργασία που αφορά την εμφάνιση των αποτελεσμάτων. Το κάθε αποτέλεσμα εμφανίζεται με έναν υπερσύνδεσμο στην αντίστοιχη σελίδα της Wikipedia για να δώσει τη δυνατότητα στο χρήστη να πάρει περισσότερες πληροφορίες. Ο χρήστης μπορεί επίσης να δει τα κείμενα-πηγές. Μία σελίδα αποτελεσμάτων φαίνεται στην Εικόνα 2.

Β. Μηχανή Αναζήτησης

Η μηχανή αναζήτησης αποτελεί πολύ σημαντικό τμήμα της εφαρμογής, αφού αυτή παρέχει όλα τα δεδομένα σε μορφή κειμένων για την εύρεση και κατάταξη των οντοτήτων. Η



Εικόνα 2

εφαρμογή λειτουργεί δηλαδή ως εμπρόσθιο τμήμα σε κάποια μηχανή αναζήτησης. Στη συγκεκριμένη έκδοση χρησιμοποιείται το Yahoo! BOSS API [5]. Άλλες μηχανές αναζήτησης που δοκιμάστηκαν ήταν το Google και Bing με παρόμοια αποτελέσματα. Το yahoo! κρίθηκε καταλληλότερο λόγω της υψηλής ποιότητας αποτελεσμάτων σε συνδυασμό με την εύχρηστη διεπαφή για επικοινωνία με άλλα προγράμματα.

Το ερώτημα αποστέλλεται στο Yahoo! API και αυτό επιστρέφει μία λίστα με τα URLs των κειμένων σε μορφή JSON (JavaScript Object Notation). Το σύστημα ζητάει τα δέκα πρώτα αποτελέσματα της αναζήτησης. Ο αριθμός των κειμένων έχει επιλεχθεί εμπειρικά ώστε να υπάρχει η επιθυμητή ποσότητα πληροφορίας και ταυτόχρονα να διατηρείται το υπολογιστικό κόστος σε λογικά επίπεδα για μία εφαρμογή πραγματικού χρόνου.

Γ. Εξαγωγή Οντοτήτων

Σε αυτό το στάδιο γίνεται η αναγνώριση των οντοτήτων μέσα στα κείμενα και προσδιορίζεται η κατηγορία τους. Αυτό επιτυγχάνεται με τη χρήση του Stanford NER(Named Entity Recognizer) [6]. Το Stanford NER αποτελεί ειδικό λογισμικό για την εξαγωγή οντοτήτων γραμμένο σε γλώσσα JAVA και διατίθεται με άδεια GNU general public license για ερευνητικούς και εκπαιδευτικούς σκοπούς. Η αναγνώριση των οντοτήτων γίνεται με τη χρήση ενός ταξινομητή, δηλαδή έναν κανόνα που κατατάσσει τις παρατηρήσεις σε κατηγορίες. Στη συγκεκριμένη περίπτωση οι κατηγορίες είναι οντότητες του τύπου πρόσωπο, τοποθεσία ή οργανισμός.

Ο ταξινομητής αποτελεί μια μέθοδο επιβλεπόμενης μηχανικής εκμάθησης. Χρησιμοποιώντας παραδείγματα όπου η ταξινόμηση έχει γίνει από άνθρωπο, κατασκευάζει μία

συνάρτηση που μπορεί να πάρει αποφάσεις για την ταξινόμηση άγνωστων λέξεων με βάση το κείμενο που τις περιβάλλει. Το Stanford NER χρησιμοποιεί ταξινομητή βασισμένο στο στατιστικό μοντέλο CRF (Conditional Random Field) και παραδείγματα από Αμερικάνικα και Βρετανικά δημοσιογραφικά άρθρα.

Για να γίνει η εξαγωγή πρέπει να αφαιρεθούν από το κείμενο οι ετικέτες HTML. Για το σκοπό αυτό χρησιμοποιήθηκε ο JSOUP HTML parser [7]. Ένα πρόβλημα που προέκυψε ήταν ότι με την αφαίρεση του HTML το κείμενο χάνει την αρχική του μορφή, με αποτέλεσμα το Stanford NER να δυσκολεύεται να αναγνωρίσει ονόματα σε λίστες ή πίνακες. Το πρόβλημα ξεπεράστηκε προσθέτοντας μία τελεία στη θέση των ετικετών κλεισίματος της HTML. Με αυτή την προσθήκη το Stanford NER πετυχαίνει ικανοποιητικά αποτελέσματα παρά τις ιδιαιτερότητες που παρουσιάζουν τα κείμενα στο Διαδίκτυο.

Δ. Μορφοποίηση/Φιλτράρισμα

Κάθε οντότητα μπορεί να εμφανίζεται σε ένα κείμενο με πολλούς διαφορετικούς τρόπους. Ένα όνομα προσώπου για παράδειγμα, μπορεί να εμφανίζεται αρχικά με το ονοματεπώνυμο και στη συνέχεια να αναφέρεται μόνο με το επώνυμο. Για να γίνει αποτελεσματική η κατάταξη των οντοτήτων στην επόμενη βαθμίδα, πρέπει πρώτα το σύστημα να αναγνωρίζει ποια ονόματα αντιστοιχούν στην ίδια οντότητα και στη συνέχεια να γραφούν όλα με τον ίδιο ακριβώς τρόπο. Τα αποτελέσματα αυτής της βαθμίδας είναι σημαντικά και για την τελική παρουσίαση των αποτελεσμάτων, όπου θέλουμε κάθε οντότητα να εμφανίζεται με την πιο σωστή και πλήρη ονομασία της και να μην έχουμε εμφάνιση του ίδιου ονόματος με πολλούς διαφορετικούς τρόπους. Η επεξεργασία των ονομάτων γίνεται σε δύο στάδια. Στο πρώτο στάδιο γίνονται μετατροπές βλέποντας κάθε όνομα μεμονωμένα, ενώ στο δεύτερο γίνεται ομαδοποίηση των ονομάτων που αντιστοιχούν στην ίδια οντότητα βλέποντας τα συνολικά.

Η βασική μετατροπή στο πρώτο στάδιο είναι κάθε όνομα να γραφεί σε μια τυποποιημένη μορφή. Η μορφή αυτή είναι κάθε λέξη να αποτελείται από το πρώτο γράμμα κεφαλαίο και τα υπόλοιπα μικρά, εκτός από ονόματα οργανισμών με λιγότερα από τέσσερα γράμματα, όπου όλα γράφονται με κεφαλαία. Στη συνέχεια απομακρύνονται συγκεκριμένα ονόματα από κάθε κατηγορία που παρατηρήθηκε ότι κατατάσσονται συχνά λανθασμένα από το Stanford NER. Για παράδειγμα ονόματα από δημοφιλείς ιστοτόπους που κατατάσσονται ως πραγματικές τοποθεσίες ή ακρωνύμια όπως FAQ, ISBN που κατατάσσονται ως οργανισμοί. Επίσης, διαγράφονται όσα ονόματα αποτελούνται αποκλειστικά από όρους του ερωτήματος. Για τις τοποθεσίες διαγράφονται ακόμα όλα τα ονόματα χωρών σε περίπτωση που δεν αποτελούν το ζητούμενο του χρήστη. Ο λόγος είναι ότι εμφανίζονται με πολύ μεγάλη συχνότητα σε κείμενα που αφορούν τοποθεσίες και διαστρεβλώνουν τα τελικά αποτελέσματα.

Στο δεύτερο στάδιο γίνεται η ομαδοποίηση συγκρίνοντας όλα τα ονόματα μεταξύ τους. Για κάθε οντότητα εξετάζεται αν αυτή αποτελεί υποσύνολο μιας άλλης σε επίπεδο λέξεων, και ύστερα αντικαθίσταται από την πληρέστερη ονομασία της. Για παράδειγμα τα ονόματα John Kennedy, Kennedy, John F. Kennedy και John Fitzgerald Kennedy θα ομαδοποιηθούν και θα γραφούν όλα με τον τελευταίο τρόπο. Για την αποφυγή ομαδοποίησης σε ονόματα που περιέχουν ορθογραφικά λάθη ή σε κάποια λανθασμένη συγχώνευση ονομάτων από λίστες, η μετατροπή αυτή γίνεται σε ονόματα που εμφανίζονται παραπάνω από μία φορά συνολικά. Η ομαδοποίηση δεν εφαρμόζεται καθόλου σε ονόματα χωρών, πόλεων και οργανισμών. Οι χώρες και οι πόλεις εμφανίζονται σπάνια με εναλλακτικούς τρόπους γραφής, ενώ τα ονόματα οργανισμών παρουσιάζουν μεγάλη ποικιλία ώστε να ομαδοποιηθούν από απλούς κανόνες.

Η παραπάνω μέθοδος ομαδοποίησης παρέχει ικανοποιητικά αποτελέσματα, υπάρχουν όμως και περιπτώσεις όπου δεν μπορεί να καθορίσει ποια ονόματα πρέπει να ομαδοποιηθούν. Για παράδειγμα, δεν μπορεί να αποφανθεί σε ποιο πρόσωπο ανήκει η εμφάνιση ενός επωνύμου, αν στις υποψήφιες οντότητες υπάρχουν δύο διαφορετικά πρόσωπα με αυτό το επώνυμο. Μια πιθανή βελτίωση θα ήταν η χρήση ενός συστήματος μηχανικής εκμάθησης, όπου η ομαδοποίηση θα γινόταν λαμβάνοντας υπόψη και τα συμφραζόμενα για κάθε όνομα, αλλά θα υπήρχε η επιβάρυνση του υπολογιστικού κόστους.

Ε. Κατάταξη Οντοτήτων

Ο αλγόριθμος της κατάταξης βασίστηκε σε στατιστικές μεθόδους χωρίς να χρησιμοποιεί τεχνικές επεξεργασίας φυσικής γλώσσας. Η είσοδος στη βαθμίδα αυτή είναι δέκα λίστες ονομάτων, μία για κάθε κείμενο-πηγή. Κάθε οντότητα βαθμολογείται με βάση την εξίσωση:

$$score = \sum_{j=1}^{df} (N + 1 - r_j)$$

όπου df είναι ο αριθμός των κειμένων που εμφανίζεται κάθε όνομα, r είναι η κατάταξη του κειμένου σύμφωνα με τη μηχανή αναζήτησης και παίρνει τιμές από ένα ως δέκα, N είναι ο συνολικός αριθμός των κειμένων και έχει την τιμή δέκα. Η εξίσωση είναι βασισμένη στη μέθοδο εκλογής Borda Count. Σύμφωνα με αυτή, αν μία οντότητα εμφανίζεται στο πρώτο κείμενο θα πάρει δέκα βαθμούς, αν εμφανίζεται στο δεύτερο θα πάρει εννέα κλπ. Οι τιμές αυτές αθροίζονται για κάθε οντότητα και το τελικό αποτέλεσμα που επιστρέφεται είναι οι δέκα οντότητες με το μεγαλύτερο βαθμό. Η επιλογή της εξίσωσης βαθμολόγησης έγινε μετά τη διεξαγωγή του πειράματος που περιγράφεται παρακάτω.

IV. ΠΕΙΡΑΜΑΤΑ

Το πρόβλημα που καλείται να λύσει ο αλγόριθμος της κατάταξης είναι κατά κάποιο τρόπο αντίστροφο από το πρόβλημα της εύρεσης κειμένων που είναι συναφή με ένα ερώτημα. Αντί να ψάχνουμε μια συλλογή κειμένων που σχετίζεται με κάποιους όρους, έχουμε έτοιμη μια συλλογή κειμένων με κοινό θέμα, και ψάχνουμε να βρούμε ποιες λέξεις (στην περίπτωσή μας ποια ονόματα) είναι σημαντικές για αυτά τα κείμενα. Οι ποσότητες που θεωρήθηκαν ενδεικτικές για την κατάταξη με βάση το παραπάνω σκεπτικό είναι οι εξής:

- Ο συνολικός αριθμός εμφάνισης κάθε οντότητας στη συλλογή των κειμένων (f_{tot}). Η ποσότητα αυτή δείχνει πόσο σημαντικό είναι κάθε όνομα βλέποντας τα κείμενα ως μία ενιαία συλλογή.
- Ο αριθμός των ξεχωριστών κειμένων όπου εμφανίζεται κάθε οντότητα (df). Αυτή η ποσότητα δείχνει κατά πόσο ένα όνομα αποτελεί κοινή πληροφορία μεταξύ των κειμένων. Θεωρώντας ότι όλα τα κείμενα είναι το ίδιο συναφή με το ερώτημα που υποβλήθηκε, τα ονόματα που εμφανίζονται σε περισσότερα κείμενα θα είναι και τα πιο σχετικά.
- Η κατάταξη του κειμένου σύμφωνα με τη μηχανή αναζήτησης όπου εμφανίζεται κάθε οντότητα (r). Λαμβάνοντας υπόψη αυτή την ποσότητα, τα κείμενα δεν αντιμετωπίζονται πλέον ως ισοδύναμα.

Για να προσδιοριστεί ποιες ποσότητες ή ποιος συνδυασμός τους είναι καταλληλότερος για την κατάταξη των οντοτήτων στο συγκεκριμένο σύστημα, πραγματοποιήθηκε ένα πείραμα όπου δοκιμάστηκαν οι παρακάτω έξι εξισώσεις βαθμολόγησης

$$score = \log(df) \quad (1)$$

$$score = \log(f_{tot}) \times \log(df) \quad (2)$$

$$score = f_{tot} \times \log(df) \quad (3)$$

$$score = \sum_{j=1}^{df} (N + 1 - r_j) \quad (4)$$

$$score = \sum_{j=1}^{df} \log(1 + f_j)(N + 1 - r_j) \quad (5)$$

$$score = \sum_{j=1}^{df} f_j (N + 1 - r_j) \quad (6)$$

Στις παραπάνω εξισώσεις j είναι ο δείκτης του κειμένου που εμφανίζεται κάθε οντότητα και N είναι ο συνολικός αριθμός των κειμένων που είναι ίσος με δέκα.

Υπάρχουν δύο διαφορετικές υποθέσεις για τη συχνότητα εμφάνισης ενός όρου και τη σημασία που έχει σε ένα κείμενο [8]. Σύμφωνα με την υπόθεση της πολυλογίας (verbosity hypothesis), η πολλαπλή εμφάνιση ενός όρου δεν είναι ιδιαίτερα σημαντική, καθώς υποστηρίζει ότι ο συγγραφέας απλά χρησιμοποίησε περισσότερες φορές αυτή τη λέξη, επειδή είναι φλύαρος. Σύμφωνα όμως με την υπόθεση της έκτασης (scope hypothesis), ο συγγραφέας του κειμένου χρησιμοποιεί περισσότερες φορές έναν όρο επειδή έχει να πει περισσότερα για το συγκεκριμένο θέμα.

Οι εξισώσεις (1), (2) και (3) δε λαμβάνουν υπόψη την κατάταξη των κειμένων σε αντίθεση με τις (4), (5) και (6). Πέρα από την κατάταξη των κειμένων, η διαφορά στις εξισώσεις βαθμολόγησης προκύπτει από τη βαρύτητα που δίνεται στη συχνότητα εμφάνισης κάθε οντότητας, με βάση τις δύο υποθέσεις που αναφέρθηκαν στην προηγούμενη παράγραφο. Έτσι, οι εξισώσεις (1) και (4) αντιπροσωπεύουν την υπόθεση της πολυλογίας, ενώ οι (3) και (6) την υπόθεση της έκτασης. Οι εξισώσεις (2) και (5), όπου χρησιμοποιείται ο λογάριθμος της συχνότητας εμφάνισης, αποτελούν μια ενδιάμεση προσέγγιση. Η χρήση του λογαρίθμου αποσκοπεί στη μείωση της ισχύος που έχει αυτός ο όρος στον υπολογισμό του βαθμού.

Εξίσωση βαθμολόγησης	**P@10**
$\log(df)$	0,5767
$\log(f_{tot}) \times \log(df)$	0,5633
$f_{tot} \times \log(df)$	0,56
$\sum_{j=1}^{df} (N + 1 - r_j)$	**0,58**
$\sum_{j=1}^{df} \log(1 + f_j)(N + 1 - r_j)$	0,5267
$\sum_{j=1}^{df} f_j (N + 1 - r_j)$	0,48

Πίνακας 1

Η αξιολόγηση των συστημάτων ανάκτησης πληροφορίας γίνεται με τη χρήση ορισμένων δεικτών. Στο συγκεκριμένο πείραμα χρησιμοποιήθηκε ο δείκτης precision-at-k (P@k) , όπου υπολογίζεται ο λόγος των σχετικών απαντήσεων προς τις μη σχετικές στα πρώτα k αποτελέσματα. Άλλοι συνήθεις δείκτες αξιολόγησης είναι ο R-precision (R-prec), ο mean average precision (MAP) και ο normalized discounted cumulative gain (nDCG) [9]. Κάθε δείκτης έχει ορισμένα πλεονεκτήματα και μειονεκτήματα για την αξιολόγηση. Ο P@k δεν παίρνει υπόψη του τη θέση των αποτελεσμάτων στη λίστα και συνεπώς έχει μεγαλύτερο περιθώριο σφάλματος , παρέχει όμως ευκολότερη ερμηνεία των αποτελεσμάτων και δεν απαιτεί τη γνώση των συνολικών σωστών απαντήσεων για ένα ερώτημα, οι οποίες μπορεί να είναι δύσκολο να βρεθούν [10].

Κάθε εξίσωση βαθμολόγησης δοκιμάστηκε σε συνολικά τριάντα ερωτήματα , βασισμένα στα θέματα αξιολόγησης συστημάτων κατάταξης οντοτήτων από το INEX 2009 και TREC 2010. Τα ερωτήματα τροποποιήθηκαν ελαφρά, ώστε να δηλώνουν με ακρίβεια το επιθυμητό αποτέλεσμα, καθώς στην αρχική τους μορφή συνοδεύονταν από ένα επεξηγηματικό κείμενο για διευκρινίσεις. Για κάθε ερώτημα υπολογίστηκε ο δείκτης P@10, καθώς δέκα είναι οι απαντήσεις που παρέχει το σύστημα. Τα περισσότερα ερωτήματα αναζητούν οντότητες που ικανοποιούν δύο συνθήκες. Για να ληφθεί μία οντότητα ως σχετική πρέπει να ικανοποιεί όλες τις επιμέρους συνθήκες. Τα αποτελέσματα του πειράματος για την αξιολόγηση των εξισώσεων φαίνονται στον Πίνακα 1. Τα ερωτήματα του πειράματος με τον αριθμό

Ερωτήματα Αξιολόγησης	Σωστές Απαντήσεις
Pacific navigators Australia explorers	5
List of countries in World War Two	10
Nordic authors known for children's literature	1
Makers of lawn tennis rackets	2
National capitals situated on islands	3
Poets winners of Nobel prize in literature	6
Formula 1 drivers that won the Monaco Grand Prix	6
Formula One World Constructors' Champions	5
Italian Nobel prize winners	9
Musicians who appeared in the Blues Brothers movies	5
Swiss cantons where they speak German	4
US Presidents since 1960	8
Countries which have won the FIFA world cup	8
Toy train manufacturers that are still in business	1
German female politicians	5
Actresses in Bond movies	8
Star Trek Captains characters	5
EU countries	10
Record-breaking sprinters in male 100-meter sprints	4
Professional baseball team in Japan	5
Japanese players in Major League Baseball	10
Airports in Germany	10
Universities in Catalunya	7
German cities that have been part of the hanseatic league	6
Chess world champions	10
Recording companies that now sell the Kingston Trio songs	1
Schools the Supreme Court justices received their undergraduate degrees	5
Axis powers of World War Two	6
State capitals of the United States of America	7
National Parks East Coast Canada US	2

Πίνακας 2

των σωστών απαντήσεων για την καλύτερη εξίσωση κατάταξης φαίνονται στον Πίνακα 2.

Από τα αποτελέσματα φαίνεται ότι η συχνότητα εμφάνισης δεν αποτελεί αξιόλογο κριτήριο, καθώς το ποσοστό της ακρίβειας μικραίνει όσο αυξάνεται η εξάρτηση της βαθμολογίας από αυτό. Φαίνεται δηλαδή, ότι στο συγκεκριμένο πρόβλημα η υπόθεση της πολυλογίας οδηγεί σε καλύτερα αποτελέσματα. Η κατάταξη των κειμένων ως κριτήριο δεν είναι ξεκάθαρο αν βελτιώνει την ακρίβεια των αποτελεσμάτων, αφού η διαφορά από τη χρήση της εξίσωσης (4) είναι ελάχιστη από αυτή της εξίσωσης (1), όπου όλα τα κείμενα θεωρούνται ισοδύναμα. Μία μελλοντική επανάληψη του πειράματος χρησιμοποιώντας και άλλους δείκτες αξιολόγησης θα μπορούσε να ξεκαθαρίσει αυτό το σημείο. Παρά το γεγονός αυτό, η εξίσωση (4) είναι αυτή που χρησιμοποιήθηκε στην υλοποίηση του συστήματος.

Από το πείραμα προκύπτουν ορισμένα συμπεράσματα και για τη γενικότερη λειτουργία του συστήματος. Το σημαντικότερο είναι ότι τα κείμενα-πηγές καθορίζουν σε μεγάλο βαθμό την ακρίβεια των αποτελεσμάτων. Στις δοκιμές που τα κείμενα-πηγές ήταν απολύτως συναφή με τα ερωτήματα, τα αποτελέσματα είχαν πολύ μεγαλύτερη ακρίβεια σε σχέση με τα αποτελέσματα των δοκιμών που τα κείμενα ήταν εν μέρει συναφή. Ένα άλλο ζήτημα προκύπτει όταν οι σωστές απαντήσεις σε ένα ερώτημα είναι λιγότερες από δέκα (πχ Axis Powers of World War Two). Ο λόγος είναι ότι δεν είναι εύκολο να προσδιοριστεί ένα όριο στη βαθμολογία που να καθορίζει αν οι οντότητες είναι γενικά σχετικές και στη συνέχεια να αυξομειώνεται ο αριθμός των αποτελεσμάτων.

V. ΣΥΜΠΕΡΑΣΜΑΤΑ

Αυτή η εργασία παρουσιάζει την υλοποίηση ενός διαδικτυακού συστήματος κατάταξης οντοτήτων που χρησιμοποιεί ως πηγές κείμενα από το Διαδίκτυο και εκτελεί την κατάταξη χρησιμοποιώντας στατιστικές μεθόδους. Η ακριβής μέθοδος της κατάταξης καθορίστηκε με πειραματικό τρόπο. Από το πείραμα φαίνεται ότι το σύστημα μπορεί να δώσει ικανοποιητικά αποτελέσματα για μεγάλο αριθμό ερωτημάτων. Ο τεράστιος όγκος δεδομένων στο Διαδίκτυο καθώς και η εξέλιξη των μηχανών αναζήτησης για ανάκτηση κειμένων θέτει πολύ λίγους περιορισμούς στη θεματολογία και τον τρόπο έκφρασης του ερωτήματος. Το σύστημα περιορίζεται στις τρεις γενικές κατηγορίες οντοτήτων (πρόσωπα, τοποθεσίες, οργανισμοί) από τη βαθμίδα εξαγωγής, αλλά μπορεί εύκολα να επεκταθεί και σε άλλες κατηγορίες όπως προϊόντα, τίτλους βιβλίων και ταινιών.

Το γεγονός ότι δε χρησιμοποιούνται τεχνικές επεξεργασίας φυσικής γλώσσας στο στάδιο της κατάταξης το καθιστούν πολύ γρήγορο στην εκτέλεση. Η μόνη χρονοβόρα διαδικασία είναι αυτή της εξαγωγής. Η επεξεργασία που απαιτείται για αυτό το στάδιο μπορεί να γίνει εκ των προτέρων δημιουργώντας μια συλλογή κειμένων από το Διαδίκτυο αντίστοιχη με αυτήν των μηχανών αναζήτησης και εκτελώντας σε αυτά την εξαγωγή οντοτήτων. Μια απλούστερη λύση είναι να αποθηκεύονται σε μια βάση δεδομένων τα ονόματα για κάθε κείμενο που επεξεργάζεται το σύστημα, η οποία θα διευρύνεται σταδιακά με τη χρήση του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] J. Lin. The Web as a Resource for Question Answering: Perspectives and Challenges. In *proceedings of the third International Conference on Language Resources and Evaluation (LREC 2002)*.2002.

[2] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, T. Westerveld. Overview of the TREC 2009 Entity Track. In *Proceedings of TREC 2009*. 2009.

[3] K. Balog, A. P. de Vries, P. Serdyukov. Overview of the Trec 2010 Entity Track. In *Proceedings of TREC 2010*. 2010.

[4] Princeton University "About WordNet". WordNet. Princeton University. 2010. http://wordnet.princeton.edu

[5] Yahoo BOSS API. [*Online*]. http://developer.yahoo.com/search/boss

[6] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*. Pages 363-370. 2005

[7] JSOUP: JAVA HTML Parser. [*Online*]. http://jsoup.org

[8] Robertson, S. E., & Walker, S.. Some Simple Effective Approximations to the 2–Poisson Model for Probabilistic Weighted Retrieval. In W B Croft and C J van Rijsbergen, editors, *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 345–354. Springer-Verlag. 1994.

[9] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze**.** *Introduction to Information Retrieval*, pages 158-163. Cambridge University Press. 2008.

[10] Chris Buckley, Ellen M. Voorhees. Retrieval Evaluation with Incomplete Information. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 2004.