ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Ηλεκτρονικών μηχανικών και μηχανικών Η/Υ

Σύστημα Αποφώνησης και Αναζήτησης Ελληνικών Δελτίων Ειδήσεων

Διπλωματική Εργασία

του

μπακαρού Αλέξανδρου μιχαήλ

ΤΟμΕΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Χανιά, Οκτώβριος 2008

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή κ. Αλ. Ποταμιάνο για όλη την υποστήριξη και καθοδήγηση που μου πρόσεφερε κατά την εκπόνηση της διπλωματικής αυτής εργασίας καθώς και τον μεταπτυχιακό φοιτητή Ο. Τσεργούλα για τη συνεργασία, τις συμβουλές και την βοήθεια που μου παρείχε. Επίσης όλους τους φοιτητές του εργαστηρίου τηλεπικοινωνιών για την άψογη συνεργασία που είχαμε στις ατελείωτες ώρες που περάσαμε μαζί στο εργαστήριο. Τέλος θα ήθελα να ευχαριστήσω την μητέρα μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφερε όλα αυτά τα χρόνια.

Περίληψη

Στον υπολογιστικό κόσμο των δεδομένων, έχουμε από την μία, μία πληθώρα μαθηματικών εργαλείων που μας βοηθούν στην καλύτερη οργάνωση αλλά και την αποτελεσματική αναζήτηση των δεδομένων, και από την άλλη μεγάλες συλλογές οπτικοακουστικών αρχείων με πολύ σπουδαίο κοινωνικό αλλά και επιστημονικό ενδιαφέρον οι οποίες παραμένουν 'ανεξιχνίαστες'. Έτσι τα τελευταία χρόνια εστιάζεται μεγάλη προσοχή στην αυτόματη ευρετηρίαση βίντεο, την αναζήτηση, ανάκτηση και απεικόνιση τους σε συστήματα ψυχαγωγικά αλλά και εκπαιδευτικά, μέσα από εφαρμογές στο διαδίκτυο μιάς και η τεχνολογική πρόοδος πάνω στο streaming media δίνει πλέον αυτήν την δυνατότητα.

Αυτό επιτυγχάνεται συνδυάζοντας όλες τις υπάρχουσες αποτελεσματικές λειτουργίες και ικανότητες για κείμενη ανάκτηση πληροφοριών, με τις νέες ηχητικές και οπτικές πληροφορίες που μας παρέχονται μέσω του αρχείου βίντεο, προσφέροντας μας πλουσιότερα αποτελέσματα. Η προσέγγιση αυτή συνδυάζει την αναγνώριση φωνής και την τεχνολογία φυσικής επεξεργασίας λόγου για την απομαγνητοφώνηση (transcription) και τον τεμαχισμό (segmentation) των δεδομένων. Τα ίδια εργαλεία χρησιμοποιούνται για την περιήγηση και αναζήτηση των οπτικοακουστικών δεδομένων.

Στόχος της διπλωματικής αυτής είναι η δημιουργία ενός συστήματος στο οποίο θα μπορεί να γίνεται αναζήτηση μέσα σε δελτία ειδήσεων της Ελληνικής τηλεόρασης. Αυτό το σύστημα θα συνδυάσει τα κείμενα δεδομένα που δημιουργούνται από ένα σύστημα αναγνώρισης φωνής (ASR) αλλά και τα ίδια τα αρχεία βίντεο για την περιήγηση μέσα στα δελτία και την εύρεση αλλά και παρουσίαση των καλύτερων δυνατόν αποτελεσμάτων.

Τα αρχεία βίντεο αναλύονται, μετατρέπονται σε μία υψηλά συμπιεσμένη streaming μορφή, και χωρίζονται σε θεματικές ενότητες μέσα από μία σειρά τεχνικών. Τα επεξεργασμένα βίντεο χρησιμοποιούνται για την ανάκτηση των κατάλληλων αποτελεσμάτων από την αναζήτηση ενώ η αλληλεπίδραση με τον χρήστη γίνεται μέσω ενός διαδικτυακού περιβάλλοντος.

Abstract

In the world of computer data, on the one hand we have, a plethora of mathematical tools that help us organize and search data effectively, and on the other we have major collections of audiovisual files with very high social and scientific interest that remain "traceless". So in recent years great attention has been focused on the automatic indexing of video, searching, retrieval and display systems for entertaining and educational reasons through web applications.

This is being achieved by combining all the existing effective operations and faculties for current recuperation of information with the new visual information that is provided with the video file, offering us richer results. This approach combines the speech recognition and the technology of natural language processing for the transcription and the segmentation of the data. The same tools are used for surfing and searching audiovisual data.

In this project, a system has been developed for the web-based retrieval of broadcast news material in order to demonstrate the potential for video search systems in the Internet context. The video is parsed off line to segment it into manageable chunks via one of a number of algorithms that have been implemented. The processed video is in a highly compressed streaming media format for retrieval via a keyword search through a web interface.

This system will combine the textual data that are created by a speech recognition system (ASR) but also from the video files themself in order to surf in the bulletins, to search and to represent the best possible results.

Περιεχόμενα

Κεφάλαιο 1

ΕΙΣΑΓΩΓΗ

1.1 ΓΕΝΙΚΑ

Περιμένουμε πολλά από μία μηχανή αναζήτησης. Κάνουμε περίεργες ερωτήσεις πάνω σε θέματα τα οποία δεν ξέρουμε ούτε οι ίδιοι αν υπάρχουν. Περιμένουμε από τον υπολογιστή να μας παρέχει τις πληροφορίες που θέλουμε αντί αυτές που πραγματικά ζητάμε. Βέβαια αυτά τα προβλήματα δεν είναι τίποτα καινούριο για κάποιο βιβλιοθηκονόμο που εργάζεται σε κάποια βιβλιοθήκη. Κάποιος έμπειρος βιβλιοθηκονόμος μπορεί ,κάνοντας μόνο λίγες σύντομες ερωτήσεις να οδηγήσει τον ενδιαφερόμενο στις πληροφορίες εκείνες που θα εκπληρώσουν τις προσδοκίες του.

Στον υπολογιστικό κόσμο των βάσεων δεδομένων η ίδια τεχνική αναπτύσσεται ενώ χρησιμοποιούμε σαν δεδομένα όχι μόνο κείμενα αλλά και αρχεία ήχου, βίντεο κλπ. Από την μία έχουμε πληθώρα μαθηματικών εργαλείων που μας βοηθούν στην καλύτερη οργάνωση αλλά και στην αποτελεσματική αναζήτηση των δεδομένων και από την άλλη μεγάλες συλλογές οπτικοακουστικών αρχείων με πολύ μεγάλο κοινωνικό αλλά και επιστημονικό ενδιαφέρον που παραμένουν όμως "ανεξιχνίαστες". Έτσι τα τελευταία χρόνια εστιάζεται μεγάλη προσοχή στην αυτόματη ευρετηρίαση βίντεο, την αναζήτηση, ανάκτηση και απεικόνιση τους σε συστήματα ψυχαγωγικά αλλά και εκπαιδευτικά.

Αυτό επιτυγχάνεται συνδυάζοντας όλες τις υπάρχουσες αποτελεσματικές λειτουργίες και ικανότητες για κείμενη ανάκτηση πληροφοριών με τις νέες χρονικές αλλά και οπτικές πληροφορίες που μας παρέχονται με το βίντεο, παρέχοντας μας πλουσιότερα αποτελέσματα, τα οποία πλέον είναι αρχεία πολυμέσων. Η προσέγγιση αυτή συνδυάζει αναγνώριση φωνής και την τεχνολογία φυσικής επεξεργασίας λόγου για την απομαγνητοφώνηση (transcription) και τον τεμαχισμό (segmentation) των δεδομένων. Τα ίδια εργαλεία χρησιμοποιούνται για την περιήγηση και αναζήτηση των οπτικοακουστικών δεδομένων. Λέγον-

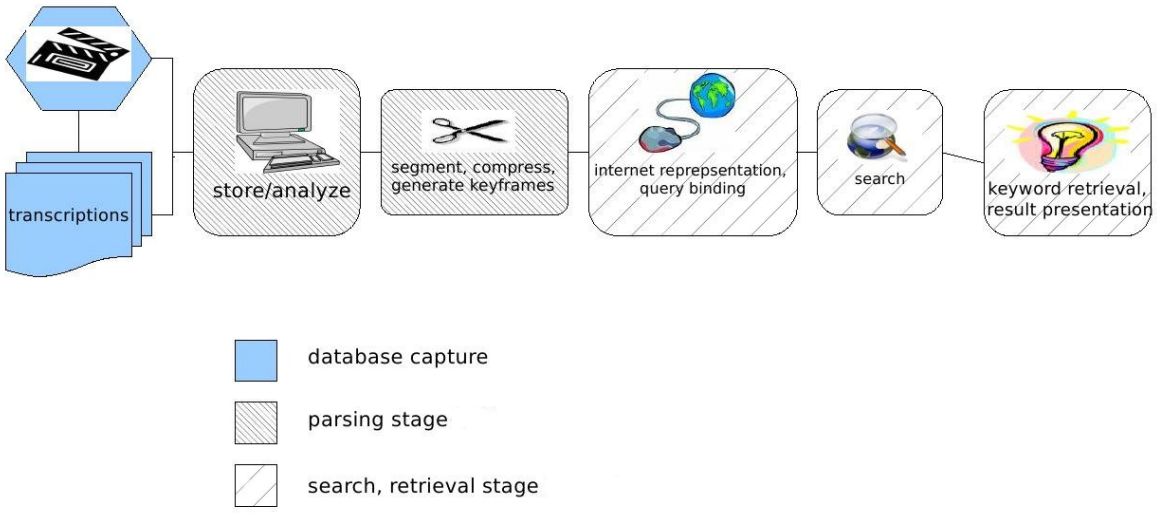
τας αναγνώριση φωνής εννοούμε την διαδικασία μετατροπής ενός ακουστικού ή φωνητικού σήματος που μπορεί να ληφθεί μέσω ποικίλων τρόπων σε μία ακολουθία λέξεων μέσω ενός αλγορίθμου. Οι αναγνωρισμένες λέξεις αποτελούν τα κείμενα δεδομένα (textual data) ενός βίντεο τα οποία μετά μπορούμε να επεξεργαστούμε περαιτέρω (αναζήτηση, τεμαχισμό).

Ένας τρόπος επεξεργασίας των κείμενων εγγραφών είναι ο τεμαχισμός (segmentation). Έτσι ονομάζουμε τον διαχωρισμό των εγγράφων σε κομμάτια καθένα από τα οποία αποτελεί μία ξεχωριστή θεματική ενότητα. μία πολύ γνωστή προσέγγιση είναι αυτή που χρησιμοποιεί λεξικολογική ανάλυση βασισμένη στην βαρύτητα όρων προκειμένου να χωριστούν τα έγγραφα σε θεματικές ενότητες, όπως θα έκρινε ένας αναγνώστης του κειμένου.

Ακόμα τα έγγραφα αυτά από την απομαγνητοφώνηση των βίντεο μπορούμε να τα χρησιμοποιήσουμε για την αναζήτηση όρων που βρίσκονται μέσα στα αρχεία βίντεο. Όταν ο χρήστης διατυπώσει μία ερώτηση με χρήση λέξεων-κλειδιών ή αυτοτελών φράσεων τότε η μηχανή αναζήτησης βρίσκει ποια αρχεία περιέχουν αυτές τις λέξεις και με βάση την βαρύτητα των όρων αλλά και κάποια άλλα κριτήρια (απόσταση λέξεων-κλειδιών, στατιστικά μοντέλα κλπ) το σύστημα εκτιμά ποια από τα αρχεία έχουν περιεχόμενο σχετικό με περιεχόμενο των λέξεων-κλειδιών που αναζητά ο χρήστης.

Τέλος πολύ σημαντικό ρόλο παίζει και η αναπαράσταση και παρουσίαση των αποτελεσμάτων επεξεργασίας των δεδομένων προς των χρήστη. μεγάλη πρόοδος έχει γίνει τα τελευταία χρόνια , με την αναβάθμιση του ρόλου του διαδικτύου στην καθημερινή μας ζωή, στην παρουσίαση της πληροφορίας προς έναν χρήστη μέσω του διαδικτύου. Σαν αποτέλεσμα παρατηρείται μεγάλη αύξηση των εφαρμογών που λειτουργούν μέσω ενός φυλλομετρητή (Web Broswer) και αναμένεται ακόμα μεγαλύτερη ανάπτυξη αυτού του κλάδου. Ένας σημαντικός παράγοντας στην επιτυχία μίας εφαρμογής είναι η φιλικότητα προς τον χρήστη . μία προσέγγιση για την βελτίωση της ποιότητας των διεπαφών χρήσης (graphic user interfaces) είναι η χρησιμοποίηση τεχνικών προσαρμογής. Αυτό σημαίνει οτι το γραφικό περιβάλλον δεν είναι στατικό αλλά ο χρήστης μπορεί να συμμετέχει ενεργά σε αυτό το περιβάλλον με τρόπο που να αλληλεπιδρά με αυτό. Στην περίπτωση των εφαρμογών που λειτουργούν μέσω ενός φυλλομετρητή η πραγματοποίηση αυτού του εγχειρήματος είναι δύσκολη, αφού η εφαρμογή τρέχει στον φυλλομετρητή του χρήστη ενώ δημιουργείται σε έναν εξυπηρετητή ιστού (Web Sever).

Συνοψίζοντας είναι πολύ σημαντικό η πληροφορία να είναι προσιτή σε όσο το δυνατό μεγαλύτερο όγκο ενδιαφερομένων. Έτσι το διαδίκτυο αποτελεί ιδανικό τρόπο παρουσίασης της πληροφορίας αφού όλοι οι χρήστες του διαδικτύου αποτελούν εν δυνάμει "πληροφοριολήπτες".



Σχήμα 1.1: σχεδιάγραμμα συστήματος

1.2 ΤΟ ΘΕΜΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ

Όπως ήδη αναφέραμε τα τελευταία χρόνια ο τομέας της ανάκτησης δεδομένων και ειδικότερα της αναζήτησης αρχείων πολυμέσων εξελίσσεται ραγδαίως. Η παρούσα διπλωματική εργασία ασχολείται με την υλοποίηση ενός συστήματος αποφώνησης και αναζήτησης τηλεοπτικών δελτίων ειδήσεων καθώς και παρουσίασης των αποτελεσμάτων στο διαδίκτυο (σχήμα 1.1).

Αποτελεί συνέχεια του συστήματος αναγνώρισης ηχητικών σημάτων από τηλεοπτικά δελτία ειδήσεων που υλοποιήθηκε από τον Τσεργούλα Ορφέα και χρησιμοποιεί την βάση που δημιουργήθηκε από την υλοποίηση του συστήματος αυτού. Χρησιμοποιήσαμε τα δελτία ειδήσεων του παραπάνω συστήματος σε μορφή οπτικοακουστικών αρχείων αλλά και των απομαγνητοφωνημένων δεδομένων που επιτεύχθηκε με την χρήση του προγράμματος/εργαλείου Transcriber Tool. Τα απομαγνητοφωνημένα αυτά δεδομένα αποτελούν την βάση δεδομένων που χρησιμοποιήσαμε για την υλοποίηση της μηχανής αναζήτησης μας. Θα αναφερθούμε διεξοδικά σε επόμενο κεφάλαιο για τα δεδομένα περιέχει η βάση μας και πως ακριβώς γίνεται η χρήση της.

Αφού επεξεργαστήκαμε κατάλληλα τα αρχεία της βάσης δεδομένων υλοποιήσαμε μέσω του κατάλληλου αλγορίθμου, την αναζήτηση των αιτημάτων. Τα κριτήρια βαρύτητας που χρησιμοποιεί αυτός ο αλγόριθμός αποτελεί τόσο η συχνότητα των όρων σε έναν τηλεοπτικό δελτίο ειδήσεων όσο η εμφάνιση τους σε όλη την βάση αλλά και η χρονική απόσταση των όρων του αιτήμα-

τος αναζήτησης που εμφανίζονται μέσα στα δελτία.

Κατόπιν υλοποιήσαμε τον τεμαχισμό των τηλεοπτικών δελτίων ειδήσεων σε θεματικές ενότητες. Ο τεμαχισμός υλοποιήθηκε σε επίπεδο κειμένου μέσω του αλγορίθμου TextTiling αλλά και σε επίπεδο αρχείων ήχου μέσω του αλγορίθμου Bic.

Σε επόμενο κεφάλαιο θα αναφερθούμε διεξοδικά στον τρόπο τεμαχισμού. Τέλος ανεβάσαμε όλα τα αρχεία της βάσης σε έναν εξυπηρετητή ιστού (Web Server) και υλοποιήσαμε την κατάλληλη σχεδιαστική διεπαφή (Graphical Interface) όπου ο χρήστης μπορεί να εισάγει το αίτημα και να βλέπει τα αποτελέσματα της αναζήτησης μέσα από κατάλληλη εφαρμογή στο διαδίκτυο.

1.3 ΟΡΓΑΝΩΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ

Η ύλη που παρουσιάζεται στην διπλωματική εργασία έχει ως εξής:

Στο δεύτερο κεφάλαιο γίνεται μία εισαγωγή στην σχεδιάση μηχανών αναζήτησης και περιγράφονται οι τρόποι δημιουργίας τους αλλά και τα μέρη από τα οποία αποτελείται, από τον "καθαρισμό" των αρχείων δεδομένων που υλοποιείται η αναζήτηση έως την σχεδίαση του γραφικού περιβάλλοντος. Ακόμα περιγράφονται κάποιοι αλγόριθμοι αναζήτησης οι οποίοι θα μας χρειαστούν στο να κατανοήσουμε καλυτερα την τεχνική που χρησιμοποιούμε εμείς.

Στο τρίτο κεφάλαιο όπου δίνονται αναλυτικά η χρονική διάρκεια και ο τύπος των δεδομένων που χρησιμοποιήθηκαν, οι κατηγορίες δεδομένων που σχηματίστηκαν (ακουστικά, βίντεο, κείμενα), όπως και ο τρόπος επεξεργασίας τους. Τέλος βλέπουμε αναλυτικά τον τρόπο δημιουργίας των απομαγνητοφωνημένων κειμένων που χεησιμοποιούμε σαν πηγή αναζήτησης.

Στο τέταρτο κεφάλαιο γίνεται αναλυτική περιγραφή του σχεδιασμού του αλγόριθμου αναζήτησης του συστήματος μας, καθώς και οι βασικές αρχές σχεδιασμού και υλοποίησης ενός αλγόριθμου εξόρυξης δεδομένων.

Στο πέμπτο κεφάλαιο περιγράφεται ο τρόπος τεμαχισμού των δεδομένων ανα θεματική ενότητα των κειμένων των δελτίων ειδήσεων. Ακόμα περιγράφεται ο τρόπος τεμαχισμού των ακουστικών σημάτων καθώς και ο συνδιασμός των 2 τρόπων προκειμένου να επιτύχουμε καλύτερα αποτελέσματα.

Στο έκτο κεφάλαιο γίνεται αναλυτική περιγραφή της σχεδίασης του interface, από το back-end στο front-end, περιγραφή του πρωτοκόλλου cgi που χρησιμοποιήσαμε καθώς και του apache server αλλά και των άλλων προγραμ-

μάτων που χρησιμοποιήσαμε για την απεικόνιση των αποτελεσμάτων (mplayer, imagemagick ,flash player κλπ) και τέλος παρουσίαση των αποτελεσμάτων όπως φάινονται κατά την αναζήτηση από κάποιο χρήστη.

Στο έβδομο και τελευταίο κεφάλαιο γίνεται αναφορά σε θέματα σχετικά με την εξέλιξη και τις μελλοντικές επεκτάσεις του συστήματος αλλά και ανακεφαλαίωση της συμπεριφοράς του συστήματος με βάση την αξιολόγηση διαφόρων χρηστών οι οποίοι το δοκίμασαν.

Κεφάλαιο 2

ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ

Ένα μεγάλο εμπόδιο στην εύρεση ικανοποιητικών αποτελεσμάτων για κάποιο σχετικό αίτημα είναι ο μεγάλος όγκος των δεδομένων ο οποίος μας αναγκάζει να εστιάζουμε την προσοχή μας στην επεξεργασία τους, είτε αυτά πρόκειται να είναι κείμενα είτε εικόνες, βίντεο ή αρχεία ήχου. Προκειμένου να αντιμετωπιστεί αυτός ο μεγάλος όγκος δεδομένων οι σχεδιαστές έχουν αναπτύξει ένα σετ από μαθηματικά εργαλεία τα οποία μας βοηθούν να βελτιώσουμε την απόδοση της μηχανής αναζήτησης.

Διάφοροι μέθοδοι βαρύτητας όρων (term weighting methods) χρησιμοποιούνται για να δώσουν διαφορετική έμφαση μεταξύ όρων ενός κειμένου ή διαφόρων κειμένων σε μία συλλογή. Το πιο συνηθισμένο εργαλείο που χρησιμοποιούμε στην Ανάκτηση Πληροφορίας (Information Retrieval) με τα πλεονεκτήματα αλλά και τα μειονεκτήματα του είναι το μοντέλο Διανυσματικού Χώρου (Vector Space Model).

2.1 ΠΡΟΕΤΟΙΜΑΣΙΑ ΑΡΧΕΙΩΝ ΚΕΙΜΕΝΟΥ

Είναι πολύ αναγκαία η οργάνωση και η προετοιμασία των δεδομένων που πρόκειται να επεξεργαστούμε. Προκειμένου τα δεδομένα μας να είναι ποιο εύκολα αναζητή-σιμα πρέπει πρώτα να "καθαριστούν". με άλλα λόγια μόνο εάν κάποιο έγγραφο έχει τίτλο, συγκεκριμένη αρχή και τέλος και συγκεκριμένη δομή, τότε μόνο, μία μηχανή αναζήτησης μπορεί να δώσει στον χρήστη ικανοποιητικά αποτελέσματα. Ακόμα ο σχεδιαστής θα πρέπει να λάβει υπ' όψιν του οτι προκειμένου να απεικονιστεί η πληροφορία στον χρήστη θα πρέπει να χρησιμοποιήσει βάσεις δεδομένων με διαφορετικούς τύπους δεδομένων (οπτικοακουστικά αρχεία, κείμενα, αρχεία ήχου κλπ) και να τα συνδυάσει αυτά για να τα παρουσιάσει στον χρήστη.

Η προετοιμασία των δεδομένων περιλαμβάνει δύο στάδια:

α)Ανάλυση Εγγράφου: Αρχικά όλα τα έγγραφα θα πρέπει να έρθουν σε μία πρότυπη μορφή που θα περιλαμβάνει τις σημαντικές πληροφορίες για κάθε έγγραφο (τίτλος, συγγραφέας, κείμενο) ανάλογα με το είδος του κειμένου ή του τύπου δεδομένων.

β) Κατόπιν θα πρέπει ο σχεδιαστής να ορίσει ποιοι όροι έχουν σημασιολογική βαρύτητα και ποιοι όχι. Αυτοί που δεν έχουν σημασιολογική βαρύτητα θα πρέπει να αποκλειστούν από την βάση. Έτσι δημιουργούμε την λίστα με τους όρους που θέλουμε να αποκλείσουμε (stop list). Αυτοί οι όροι είναι συνήθως άρθρα, προθέσεις κάποια πολύ συνηθισμένα επίθετα αλλά και κάποια ουσιαστικά. Συνήθως οι όροι με μεγαλύτερη σημασιολογική βαρύτητα είναι τα ρήματα.

2.2 ΕΞΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

Αφού πλέον έχει "καθαριστεί" η βάση μας από περιττούς όρους και έχει αποκτήσει μία συγκεκριμένη δομή το επόμενο βήμα είναι η κανονικοποίηση των όρων.

Οι όροι που βρίσκονται σε ένα αρχείο στο οποίο πρόκειται να γίνει αναζήτηση αντιμετωπίζονται σαν λεκτικές μονάδες και είναι μέρος της σχεδίασης ο χαρακτηρισμός τους(λέξεις, αριθμοί, σημεία στίξης κλπ) αλλά και η αφαίρεση του επιθέματος τους (stemming). Για παράδειγμα ο όρος $\boldsymbol{\epsilon\pi\acute{\iota}\sigma\kappa\epsilon\psi\eta}$ δεν έχει σημασιολογική διαφορά από τον όρο $\boldsymbol{\epsilon\pi\iota\sigma\kappa\acute{\epsilon}\psi\epsilon\iota\varsigma}$ επομένως κρατάμε μόνο το θέμα του όρου (στο συγκεκριμένο παράδειγμα $\boldsymbol{\epsilon\pi\acute{\iota}\sigma\kappa\epsilon\psi}$) και αφαιρούμε το επίθεμα.

2.3 VECTOR SPACE MODELING

Το μοντέλο διανυσματικού χώρου (Vector Space Model) αποτελεί μία μέθοδο αναπαράστασης των όρων και των αρχείων σε μία συλλογή. Σε ένα τέτοιο μοντέλο κάθε στοιχείο του διανύσματος (αρχείου) μπορεί να χρησιμοποιηθεί για να αναπαραστήσει λέξεις, φράσεις ή θέμα ενός κειμένου. Κάθε τιμή που παίρνει αυτό το στοιχείο αντανακλά στην σπουδαιότητα του όρου και αναπαριστά την σημαντικότητα (semantic) του κειμένου, αλλά σε αυτό θα αναφερθούμε αναλυτικότερα παρακάτω.

Ένα από τα πρώτα μοντέλα διανυσματικού χώρου στον τομέα της ανάκτησης πληροφοριών (Information Retrieval) είναι το SMART (System for the Mechanical Analysis and Retrieval of Text) όπου δημιουργήθηκε από τον Gerald Salton στο Πανεπιστήμιο του Cornell [1].

μια συλλογή κειμένων που αποτελείται από n κείμενα τα οποία περιέχουν m όρους μπορούν να αναπαρασταθούν σαν ένας mxn πίνακας A. Τα διανύσματα (στήλες) αναπαριστούν τα n κείμενα . Επομένως το στοιχείο a_{ij} αναπαριστά την συχνότητα (weighted frequency) στην οποία ο όρος i βρίσκεται στο κείμενο j. Χρησιμοποιώντας το VSM, οι στήλες του πίνακα A αποτελούν τα διανύσματα του κειμένου (document vectors) καθώς οι γραμμές του πίνακα A αποτελούν τα διανύσματα όρου (term vectors). Το VSM χρησιμοποιείται για να εξάγει μία γεωμετρική σχέση μεταξύ των κειμένων (διανυσμάτων) προκειμένου να καταλάβουμε πόσο κοντά ή μακριά σημασιολογικά βρίσκονται δύο κείμενα.

Στο παρακάτω σχήμα[1] [σχήμα 2.1] μπορούμε να δούμε πώς ένας $9x7$ πίνακας έχει κατασκευαστεί από μία μικρή συλλογή τίτλων βιβλίων. Σε αυτό το μικρό παράδειγμα έχουν υπογραμμιστεί οι όροι που χρησιμοποιούνται στον πίνακα και αποτελούν διανύσματα αφού οι υπόλοιποι διαγράφονται κατά την διάρκεια "καθαρίσματος" όπως αναφέραμε προηγουμένως (αφού δεν έχουν ερμηνευτική αξία). Συνήθως για μεγάλα κείμενα οι συχνότητες θα είναι μεγαλύτερες από 1 για κάποιο κείμενο. Πολλές φορές για την αναπαράσταση της συχνότητας του κάθε όρου στα κείμενα χρησιμοποιούμε την ευκλείδεια νόρμα η οποία ορίζεται ως εξής:

$$\|x\|_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{m} x_i^2}$$

Όπως φαίνεται από το παραπάνω σχήμα δεν χρησιμοποιούνται όλες οι λέξεις για να περιγράψουν τους τίτλους των βιβλίων. Κατασκευάζοντας τον πίνακα όρων-κειμένων, οι όροι συνήθως χαρακτηρίζονται από το στέμμα τους (όπως

[1] το παραπάνω σχήμα απεικονίζει τίτλους αγγλικών βιβλίων

κείμενα
D1: **infant** and **Toddler** First Aid
D2: **Babies** and **Childrens** Room (For yor **Home**)
D3: **Child** safety at **Home**
D4: Your **Babys Health** and **Safety**: From **infant** to **Toddler**
D5:**Baby proofing** Basics
D6: Your **Guide** to easy Rust **proofing**
D7: Beannie **Babies** Collectors **Guide**

T1: Bab(y,ies,ys)
T2: Child(ren)
T3: Guide
T4: Health
T5:Home
T6: Infant
T7: Proofing
T8: Safety
T9: Toodler

κάθε στοιχείο του πίνακα A αναπαριστά την συχνότητα εμφάνισης του όρου i στο κείμενο j.

$$\hat{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

ο πινακας A αφου κανονικοποιηθηκε:

$$\hat{A} = \begin{pmatrix} 0 & 0.5774 & 0 & 0.4472 & 0.7071 & 0 & 0.7071 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7071 & 0.7071 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \end{pmatrix}$$

Σχήμα 2.1: η κατασκευή του πίνακα A

είδαμε στην ενότητα 2.2). Στο παράδειγμα μας οι λέξεις child, children και childrens θεωρούνται σαν ένας όρος. Η μείωση του όγκου των δεδομένων μέσω της αφαίρεσης του επιθέματος είναι πολύ σημαντική για μεγάλες συλλογές κειμένων.

Ακόμα, στο παραπάνω μικρό παράδειγμα βλέπουμε δύο κύρια χαρακτηριστικά εμπόδια στην ανάκτηση δεδομένων. Και αυτά είναι η συνωνυμία και της πολυσημία. Η συνωνυμία αναφέρεται στην χρήση συνώνυμων ή διαφορετικών όρων που όμως έχουν το ίδιο νόημα, και η πολυσημία αναφέρεται σε όρους που έχουν διαφορετική έννοια ανάλογα με τον περιεχόμενο του κειμένου που βρίσκονται. Παρακάτω θα αναφέρουμε διάφορες μεθόδους προκειμένου να ξεπεράσουμε το πρόβλημα της συνωνυμίας και της πολυσημίας.

Εάν τώρα θέλουμε να βρούμε όλα τα βιβλία που να περιέχουν τους όρους child και proofing τότε το αίτημα μας θα προρεί να αναπαρασταθεί με τον παρακάτω τρόπο σαν διάνυσμα:

$$q = (0\,1\,0\,0\,0\,0\,0\,1\,0\,0)^T$$

Το ταίριασμα τους αιτήματος με τα υπόλοιπα διανύσματα στο VSM μπορεί να θεωρηθεί σαν αναζήτηση στον χώρο στηλών του πίνακα A για την εύρεση του πιο κοντινού στο αίτημα διανύσματος. μία από τις πιο γνωστές μετρικές ομοιότητας (similarity metrics) για ταίριασμα αιτημάτων είναι αυτή του συνημιτόνου της γωνίας μεταξύ του διανύσματος-αιτήματος (query vector) και των διανυσμάτων κειμένων (document vectors). Εαν θεωρήσουμε σαν α_j το j-οστό διανύσμα-κειμένου τότε το συνημίτονο μεταξύ του διανύσματος αιτήματος $q = (q_1, q_2, q_3, ..., q_m)^T$ και των διανυσμάτων κειμένων θα είναι ώς εξής :

$$cos\,(i,j) = \frac{a_j^T q}{\|a_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{m} a_{ij} q_i}{\sqrt{\sum_{i=1}^{m} a_{ij}^2}\sqrt{\sum_{i=1}^{m} q_i^2}}$$

για $j = 1, 2...., n$. Αφού τα περισσότερα στοιχεία τόσο του διανύσματος κειμένου όσο και του διανύσματος αιτήματος είναι μηδενικά τότε ο υπολογισμός του εσωτερικού γινομένου τους δεν είναι και τόσο ακριβή υπολογιστικά διαδικασία. Οσο μεγαλύτερο είναι το συνημίτονο για κάποιο διάνυσμα κειμένου τόσο πιο σχετικό είναι αυτό με το διάνυσμα αιτήματος.

Η αναπαράσταση των κειμένων βασισμένα αποκλειστικά στην συχνότητα εμφάνισης των όρων (tf) δεν προσφέρει ικανοποιητικά αποτελέσματα κυρίως όσον αφορά το σημασιολογικό ταίριασμα εγγράφων. Υπάρχουν διάφορες τεχνικές που μπορούν να βελτιώσουν την απόδοση του VSM. μία από αυτές είναι βασισμένη στην προσθήκη ειδικών βαρών στους όρους (term weights) όπως θα δούμε και αργότερα ενώ μία άλλη βασίζεται στην low-rank εκτίμηση του

Πίνακας 2.1: λίστα με τοπικά βάρη όρων (l_{ij})

σύμβολο	όνομα	σχέση
b	binary	$x(f_{ij})$
l	logarithmic	log(1+fij)
n	augmented normalized term frequency	$(x(f_{ij}) + f_{ij}/max_{kj}f_{ij})/2$
t	term frequency	f_{ij}

όπου

$$x\left(r\right) = \begin{cases} 1\, if\, r > 0 \quad 0\, if\, r = 0 \end{cases}$$

αρχικού πίνακα A. Η μέθοδος αυτή βασίζεται στην μείωση του θορύβου (λόγω των προβλημάτων συνωνυμίας και πολυσημίας).

μία συλλογή από κείμενα που αποτελούνται από m όρους (ακριβέστερα λεκτικές μονάδες) όπως είπαμε και παραπάνω μπορεί να αναπαρασταθεί από ένα mxn πίνακα $A = [a_{ij}]$. Κάθε a_{ij} αποτελεί αποτελεί και ένα βάρος το οποίο αντιστοιχεί στον όρο i που βρίσκεται στο κείμενο j. Ο κύριος σκοπός χρήσης τέτοιων βαρών είναι για να βελτιώσουν την απόδοση της μηχανής αναζήτησης. Η απόδοση σε αυτή την περίπτωση βασίζεται στην ικανότητα να λαμβάνουμε σχετικές πληροφορίες (recall) και να απορρίπτουμε μη σχετικές πληροφορίες (presicion). Η ανάκληση (recall) υπολογίζεται ως ο αριθμός σχετικών κειμένων για κάποιο αίτημα σε σχέση με τον συνολικό αριθμό των σχετικών όρων που υπάρχουν στην συλλογή ενώ η ακρίβεια (presicion) υπολογίζεται ως ο αριθμός των σχετικών κειμένων σε σχέση με τον συνολικό αριθμό κειμένων της συλλογής. Οσο μεγαλύτερη είναι η ακρίβεια (presicion) τόσο καλύτερη θεωρείται η απόδοση του συστήματος.

Χρησιμοποιώντας την ίδια φόρμα που έχουμε χρησιμοποιήσει έως τώρα κάθε στοιχείο aij θα μπορεί να προσδιοριστεί ώς $a_{ij} = l_{ij}g_{ij}d_j$ όπου το l_{ij} είναι το τοπικό βάρος για τον όρο i όταν αυτός βρίσκεται στο κείμενο j, g_i είναι το ολικό βάρος και d_j είναι παράγοντας κανονικοποίησης του κειμένου που προσδιορίζει το εάν κάθε στήλη του Πίνακα A έχει κανονικοποιηθεί ή όχι.

Οι πίνακες 2.1 έως 2.3 περιέχουν κάποια βασικά βάρη που χρησιμοποιούνται σε τέτοιου είδους συστήματα[3].

Οπως αναφέραμε και παραπάνω ο αριθμός των μη μηδενικών στοιχείων σε έναν πίνακα είναι αρκετά μικρός σε σχέση με τον αριθμό των μηδενικών στοιχείων. Γενικότερα αυτοί οι πίνακες δεν έχουν κάποια συγκεκριμένη δομή όπως ο παρακάτω διαγώνιος πίνακας:

Πίνακας 2.2: λίστα με ολικά βάρη όρων (g_{ij})

σύμβολο	όνομα	σχέση
x	none	1
e	entropy	1+(Σ(plog(p))/log n)
f	inverse document frequency	log(n/Σχ(fij))
g	GfIdf	$\sum_j f_{ij} / \sum_j x(f_{ij})$
n	Normal	$1/\sqrt{\sum_j f_{ij}^2}$
p	Probalistic Inverse	log((n-$\sum_j x(f_{ij})$)/$\sum_j x(f_{ij})$)

όπου

$$x(r) = \begin{cases} 1 \, if \, r > 0 \\ 0 \, if \, r = 0 \end{cases}$$

Πίνακας 2.3: λίστα με τρόπους κανονικοποίησης των κειμένων (d_j)

σύμβολο	όνομα	σχέση
χ	none	1
c	cosine	$\left(\sum_j (g_i l_{ij})^2\right)^{-\frac{1}{2}}$

$$A = \begin{pmatrix} q & q & & & & \\ q & q & & & & \\ & q & q & & & \\ & & q & q & & \\ & & q & q & & \\ & & & q & q & \\ & & & q & q & \\ & & & q & q & \\ & & & & q & q \\ & & & & q & q \end{pmatrix}$$

Εάν ο πίνακας όρων-κειμένων έχει την παραπάνω μορφή τότε ο τρόπος να βρούμε όμοια κείμενα είναι αρκετά πιο απλός. Παρόλα αυτά είναι δύσκολο να βρούμε τέτοιου είδους πίνακες, εντούτοις υπάρχουν διάφοροι τρόποι ανακατάταξης των γραμμών προκειμένου να αποκτήσουν την παραπάνω μορφή. Προκειμένου να αποφύγουμε την αποθήκευση και επεξεργασία των μηδενικών στοιχείων έχει δημιουργηθεί μια σειρά μεθόδων "συμπίεσης" αραιών πινάκων. Δύο κατάλληλοι τρόποι για την "συμπίεση" των πινάκων όρων-κειμένων είναι ο CSS (Compressed Column Storage) και ο CRS (Computed Row Storage). Με τον πρώτο όρο αποθηκεύονται οι μη μηδενικές στήλες σε ειδικούς πίνακες ενώ με

τον δεύτερο τρόπο αποθηκεύονται οι μη-μηδενικές γειτονικές γραμμές σε πίνακες. Ο τρόπος δημιουργίας αλλά και τα περιεχόμενα τέτοιων πινάκων είναι πέρα από τους σκοπούς αυτής της εργασίας γι αυτό δεν αναλύονται περαιτέρω.

Πιο πρόσφατες μελέτες έχουν εστιάσει το ενδιαφέρον τους στην μέθοδο εύρεσης συναφών σημασιολογικά κειμένων χωρίς το λεκτικό ταίριασμα όρων αλλά μέσω της rank-k προσέγγιση. Το Latent Semantic Indexing αποτελεί μία τέτοια μέθοδο όπου οι m όροι και τα n κείμενα κωδικοποιούνται στον k-χώρο όπου k<<min(m,n).

2.4 ΑΙΤΗΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ

Διάφορες μηχανές αναζήτησης μπορούν να δέχονται διαφορετικά είδη αιτημάτων, ενώ ο χρήστης της μηχανής αναζήτησης μπορεί να έχει διαφορετικό τύπου αιτήματος κατά νου από αυτό που η μηχανή αναζήτησης μπορεί να δεχτεί.

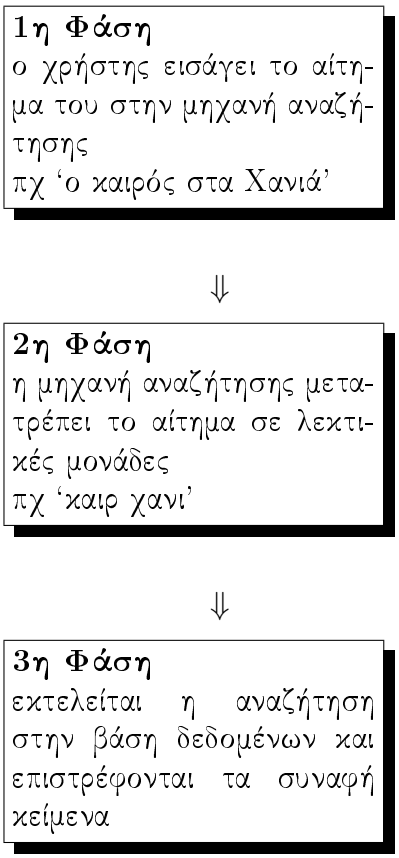
Σαν quey binding εννοούμε την ανάλυση του αιτήματος της μηχανής αναζήτησης από τον χρήστη. Αυτή η ανάλυση χωρίζεται σε τρεις φάσεις όπως βλέπουμε στο σχήμα 2.2.

Η πρώτη φάση περιλαμβάνει την απάντηση από τον χρήστη μίας ερώτησης ή η εισαγωγή κάποιων όρων που θα χρησιμοποιηθούν για την εύρεση των αποτελεσμάτων. Στην επόμενη φάση η μηχανή αναζήτησης μεταφράζει αυτούς τους όρους και τους μετατρέπει σε λεκτικές μονάδες και τέλος η τρίτη φάση περιλαμβάνει η μηχανή αναζήτησης χρησιμοποιεί τις επεξεργασμένες αυτές λεκτικές μονάδες για να εκτελέσει αναζήτηση στην βάση δεδομένων και να λάβει τα αντίστοιχα αποτελέσματα.

Ένας χρήστης μπορεί να θέσει ένα αίτημα μέσα από ποικίλους τρόπους (Boolean statement, ερώτηση, λίστα από όρους ή ακόμα με την χρήση θησαυρού). Εδώ όμως αρχίζουν να προκύπτουν προβλήματα ανάλογα με το είδος της αναζήτησης. Για παράδειγμα εάν ένα αίτημα χρησιμοποιεί τον Boolean τελεστή AND η OR πρέπει να αντιμετωπιστεί διαφορετικά από ένα απλό αίτημα. Η σχεδίαση μίας μηχανής αναζήτησης είναι άρρηκτα συνδεδεμένη με τον τύπο των αιτημάτων που μπορούν να δεχτούν.

Υπάρχουν, όπως είπαμε και παραπάνω, διάφοροι τύποι αιτημάτων. Είναι πολύ συνηθισμένο μία μηχανή αναζήτησης να παρέχει συνδυασμό αιτημάτων που θα μας προσφέρουν στον χρήστη περισσότερες επιλογές και καλύτερα επιθυμητά αποτελέσματα. Παρακάτω θα περιγράψουμε κάποιους τύπους αιτημάτων που συναντάμε συνήθως[3]:

- **Boolean Queries**



Σχήμα 2.2: οι τρεις φάσεις συσχετισμού του αιτήματος

Τα λογικά αιτήματα συνδέουν τους όρους σε μία αναζήτηση χρησιμοποιώντας τελεστές όπως ΚΑΙ, Η ή ΌΧΙ. Όπως ήδη πολύ πιθανόν να γνωρίζουμε ο τελεστής ΚΑΙ επιστρέφει αποτελέσματα από την αναζήτηση που περιέχουν όλους τους όρους του αιτήματος, ο τελεστής Η επιστρέφει αποτελέσματα που περιέχουν κάποιους από τους όρους του αιτήματος (τουλάχιστον έναν) ενώ ο τελεστής ΌΧΙ επιστρέφει αποτελέσματα που δεν περιέχουν τους όρους του αιτήματος. Η μεγαλύτερη αδυναμία των λογικών αιτημάτων είναι ότι οι περισσότεροι χρήστες συστημάτων εξόρυξης πληροφοριών δεν είναι καλά εκπαιδευμένοι στο να χρησιμοποιούν τέτοιου είδους αιτήματα και τα αποφεύγουν.

- **Natural Language Queries**

Τα αιτήματα φυσικού λόγου είναι αιτήματα όπου ο χρήστης θέτει στο σύστημα ερωτήσεις ή δηλώσεις. Για παράδειγμα ένα αίτημα φυσικού λόγου θα μπορούσε να είναι:

Ποια κείμενα αναφέρονται στο μάννα·
ή
Βρες μου για την χρήση του όρου μάννα ως πνευματική έννοια .

Για την επεξεργασία των αιτημάτων η μηχανή αναζήτησης θα πρέπει να εξάγει αρχικά όλες τις λεκτικές μονάδες προκειμένου να ξεκινήσει η αναζήτηση. Προφανώς κάποιοι όροι του αιτήματος χαρακτηρίζονται σαν stop words και θα αφαιρεθούν από το αίτημα. Αυτή όμως η προσέγγιση έχει το μειονέκτημα ότι το αίτημα χάνει την σημασιολογική του ερμηνεία απομονώνοντας μόνο κάποιες λέξεις του. Ειδικότερα αφαιρώντας κάποιους όρους από το αίτημα τότε μπορεί να μην είναι σαφές με τι ερμηνευτική σκοπιμότητα χρησιμοποιούνται κάποιοι όροι (πολυσημία).

- **Thesaurus Queries**

Ενας θησαυρός χρησιμοποιείται στην περίπτωση που ο χρήστης μπορεί να επιλέξει από μία λίστα από προκαθορισμένους όρους που έχουν δημιουργηθεί από το σύστημα και γνωρίζουμε ήδη οτι υπάρχουν σαν έννοιες μέσα στην βάση δεδομένων. Το πλεονέκτημα αυτού του τρόπου είναι οτι η πρώτη φάση της ανάλυσης του αιτήματος έχει γίνει ήδη από το σύστημα αυτόματα και δεν χρειάζεται να επεξεργαστούμε το αίτημα που εισαγάγει ο χρήστης. Ακόμα μέσω του θησαυρού μπορούμε να αναζητήσουμε και συναφή με αυτό που ψάχνουμε αντικείμενα. Παρ όλα αυτά με αυτόν τον τρόπο ο χρήστης είναι εγκλωβισμένος σε μία στατική λίστα από όρους και είναι αναγκασμένος να επιλέξει ανάμεσα σε αυτούς ακόμα και στην περίπτωση που θεωρεί οτι δεν τον καλύπτουν οι υπάρχοντες όροι.

- **Fuzzy Queries**

Τα ασαφή αιτήματα είναι αυτά που μπορεί το σύστημα να τα χειριστεί ακόμα και στην περίπτωση που οι όροι του αιτήματος δεν υπάροχυν (πχ όροι που περιέχουν ορθογραφικά λάθη). Τότε από τους όρους αυτούς αφαιρείται το επίθεμα και γίνεται αυτόματος ορθογραφικός έλεγχος (μέσω λεξικού) και έτσι επιστρέφονται στον χρήστη τα επιθυμητά αποτελέσματα.

- **Term Searches**

Ο ποιο δημοφιλης τρόπος εισαγωγής αιτημάτων (ειδικά στο διαδίκτυο) είναι όταν ο χρήστης παρέχει κάποιες λέξεις ή κάποιες μικρές φράσεις. Κάποιοι πιο έμπειροι χρήστες μπορεί να εισάγουν συνεχόμενες φράσεις και να περιμένουν αποτελέσματα από αυτή την αλληλουχία των λέξεων (πχ να εισάγουν το αίτημα: **ΕΘΝΙΚΗ ΕΛΛΑΔΟΣ**). Εάν η μηχανή αναζήτησης δεν αντιμετωπίσει το αίτημα σαν μία συνεχόμενη φράση τότε μπορεί να δώσει στον χρήστη αποτελέσματα διαφορετικά των προσδοκιών του. (στο παράδειγμα μας μπορεί να δώσει κείμενα που να έχουν σχέση πχ με μία **ΕΘΝΙΚΗ** εορτή και κείμενα που αναφέρονται στην εκκλησία της **ΕΛΛΑΔΟΣ**). Από την άλλη εάν χρησιμοποιήσουμε αποκλειστικά τους όρους σαν μία πολύ στενή συνεχόμενη φράση ίσως χάσουμε κάποια αποτελέσματα που ο χρήστης όμως θα ήθελε να δεί (πχ θα χάσουμε κάποιο κείμενο που αναφέρεται στην ...**ΕΘΝΙΚΗ** ομάδα της **ΕΛΛΑΔΟΣ**...).

Ετσι θα πρέπει να λανβάνεται υπ όψιν και η απόσταση των όρων μεταξύ τους και σε περίπτωση που εξετάζουμε ένα σύστημα όπως το παραπάνω θα πρέπει να υπάρχει μία ανοχή μεταξύ των ενδιάμεσων όρων (πχ πέντε ή δέκα λέξεις ανάμεσα στους όρους του αιτήματος). Τελος ένα συνηθισμένο δίλημμα του χρήστη είναι να καθορίσει πόσους όρους πρέπει να παρέχει στην μηχανή αναζήτησης έτσι ώστε να του δώσει ικανοποιητικά αποτελέσματα. Ειδικά στην περίπτωση που τα δεδομένα αναπαριστώνται με VSM τότε όσες περισσότερες λέξεις εισάγει ο χρήστης τόσο καλύτερα για το σύστημα (λιγότεροι μη μηδενικοί όροι).

- **Probabilistic Queries**

Ενας άλλος τρόπος διαχείρησης των αιτημάτων είναι με βάση στατιστικές μεθόδους. Είναι δυνατό να υπολογιστεί η δεσμευμένη πιθανότητα ,δεδομένου του αιτήματος, αυτό το αίτημα να έιναι συναφές με κάποιο κείμενο [2].

2.5 ΟΜΟΙΟΤΗΤΑ

Κάποιον χρήστη δεν τον ενδιαφέρει πώς μια μηχανή αναζήτησης δουλεύει ή τι χαρακτηριστικά έχει. Αρκεί να μπορεί να κάνει την δουλειά του και να λάβει τα αποτελέσματα που επιθυμεί. Αν η μηχανή αναζήτησης δεν το κάνει αυτό τότε και αυτός δεν την χρησιμοποιεί ξανά. μόνο ο ίδιος ο χρήστης μπορεί να κρίνει εάν τα αποτελέσματα που πήρε είναι ικανοποιητικά. Στον τομέα ανάκτησης πληροφοριών αυτό ονομάζεται συνάφεια (relevance), δηλαδή το πόσο καλά είναι τα ληφθέντα αποτελέσματα σε σχέση με το δοθέν αίτημα. Προκειμένου να μετρήσουμε την απόδοση του συστήματος αναζήτησης δεδομένων χρησιμοποιούμε δύο μέτρα: την ακρίβεια (presicion) και την ανάκληση (recall).

Η ακρίβεια συμβολίζεται με P ορίζεται ως:

$$P = \frac{D_r}{D_t}$$

όπου Dr είναι ο αριθμός των συναφών λεφθήνετων αποτελεσμάτων και Dt είναι ο συνολικός αριθμός των ληφθέντων αποτελεσμάτων. Είναι σημαντικό να καταλάβουμε οτι η συνάφεια ενός κειμένου δεδομένου κάποιου αιτήματος είναι σχετική και εξαρτάται από την κρίση του χρήστη.

Η ανάκληση συμβολίζεται με R και ορίζεται ως:

$$R = \frac{D_r}{N_r}$$

όπου Dr είναι ο αριθμός συναφών ληφθέντων κειμένων ,όπως και παραπάνω, ενώ ο Nr είναι ο συνολικός αριθμός συναφών κειμένων στην συλλογή (δηλαδή Dr + τα συναφή κείμενα που όμως δεν λήφθηκαν).

Η μέση ακρίβεια είναι μία μετρική που χρησιμοποιείται στην ανάκτηση δεδομένων για τον υπολογισμό της απόδοσης. Θέτουμε σαν r_i τον μέγιστο αριθμό των συναφών κειμένων και i την θέση της ταξινομημένης ,κατά συνάφεια , λίστας των κειμένων. Η ανάκληση του i-οστού κειμένου στην λίστα είναι ανάλογο των συναφών κειμένων που έχουμε δει έως τώρα. Ομοίως και το $P_i = r_i/i$.

Ακόμα για την μέτρηση της απόδοσης χρησιμοποιείται το Φ-μεασυρε το οποίο ορίζεται ώς:

$$F = \frac{2 \star P \star R}{P + R}$$

όπου P και R είναι το precision και το recall όπως αναφέραμε και παραπάνω.

2.6 ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Ένα πολύ σημαντικό μέρος της κατασκευής μίας μηχανής αναζήτησης είναι η δημιουργία του εργαλείου εκείνου που θα χρησιμοποιούν οι χρήστες για να αλληλεπιδρούν με την μηχανή αναζήτησης, ή αλλιώς το γραφικό περιβάλλον εργασίας του χρήστη (Graphical User Interface ή απλά GUI). Η σημασία του γραφικού περιβάλλοντος έγκειται στο γεγονός οτι πολλές φορές ο χρήστης θα κρίνει το αποτέλεσμα της αναζήτησης όχι μέσω των αποτελεσμάτων που επιστρέφουν από τον αλγόριθμο αναζήτησης αλλά από τα κουμπιά που θα πατήσει και την προσβασιμότητα που θα έχει σε αυτά τα αποτελέσματα. Ακόμα τρόπος σχεδίασης ενός γραφικού περιβάλλοντος μπορεί να διαφέρει ανάλογα για το τί είδους χρήστη προορίζεται.

Ενα σημείο που απασχολεί ιδιαίτερα τους σχεδιαστές είναι η συμπλήρωση της φόρμας από τους χρήστες. Παρατηρώντας τις ήδη γνωστές μηχανές αναζήτησης (Yahoo!, Google, Altavista, κλπ) παρατηρούμε οτι είναι σχεδιασμένες με το ίδιο μοτίβο μιας και αφήνουν τον χρήστη να αποφασίσει ο ίδιος για το αίτημα (search terms method όπως αναφέραμε και προηγουμένως) αλλά και περιέχει την δυνατότητα εισαγωγής περαιτέρω ρυθμίσεων αναζήτησης (advanced search) για κάποιο πιο προχωρημένο χρήστη χωρίς αυτες οι επιλογές να επηρεάζουν τον αρχάριο.

μπορούμε να κατηγοριοποιήσουμε τους τρόπους αλληλεπίδρασης μεταξύ υπολογιστή χρήστη σε τρεις βασικούς τύπους όσον αφορά τα συστήματα αναζήτησης [4]:

- **Commands:**

Ο χρήστης εισάγει συγκεκριμένες εντολές που είναι καθορισμένες για κάποιο σύστημα. Τα συστήματα που λειτουργούν με εντολές είναι πιο δύσκολο να χρησιμοποιηθούν αφού απαιτείται ειδική γνώση των εντολών.

- **Menu**:

Ο χρήστης έχει να επιλέξει μέσα από ένα μενού από εντολές, και εκτελείται η εντολή εκείνη που θα επιλέξει. Αυτός ο τρόπος επιτρέπει στον χρήστη να χρησιμοποιεί το σύστημα χωρίς να έχει ιδιαίτερες γνώσεις στο πώς λειτουργούν οι εντολές αυτές. Παρολα αυτά με αυτό τον τρόπο είναι περιορισμένες οι ενέργειες που μπορεί να εκτελέσει ο χρήστης.

- **Form Fill-in:**

Αποτελεί τον συνδιασμό των δύο παραπάνω τρόπων (commands και menu).

Ειδικότερα όσον αφορά την σχεδίαση περιβαλλόντων για μηχανές αναζήτησης υπάρχουν κάποιοι κανόνες που πρέπει κάθε σχεδιαστής να ακολουθεί.

Αρχικά θα πρέπει να μπορεί να δεχτεί και νέους αλλά και πιο έμπειρους χρήστες. Ακόμα ο χρήστης θα πρέπει να θεωρεί οτι έχει τον έλεγχο της βάσης δεδομένων και κατά συνέπεια των αποτελεσμάτων που επιστρέφονται από την αναζήτηση. Έτσι θα πρέπει να μπορεί να αλλάζει την διάταξη των αποτελεσμάτων (κατά χρονολογική σειρά, κατά αλφαβητική σειρά , κατά συνάφεια κλπ).

Ακόμα θα πρέπει να είναι ξεκάθαρες και οι πληροφορίες που λαμβάνει ο χρήστης για το συγκεκριμένο αίτημα (ημερομηνία, όροι αιτήματος που συναντώνται κλπ).

Επιπλέον είναι πολύ σημαντικό ο χρήστης να ξέρει με τί κριτήρια υλοποιήθηκε η αναζήτηση (πχ αν υλοποιήθηκε με λογικό τελεστή ΚΑΙ ή Ή) και ο χρήστης να έχει την δυνατότητα να αλλάξει τα κριτήρια χωρίς αυτό όμως να τον μπερδεύει (πιθανή λύση σε αυτό είναι να υπάρχει μία προεπιλεγμένη/προτεινόμενη μορφή που θα χρησιμοποιείται σε περίπτωση που ο χρήστης δεν ξέρει με τί κριτήρια να εκτελέσει την αναζήτηση).

Τέλος όσον αφορά την σχεδίαση του γραφικού περιβάλλοντος πρέπει να πούμε οτι το καλύτερο περιβάλλον είναι αυτό που είναι και το πιο "διάφανο" ή για να χρησιμοποιήσουμε και ένα αθλητικό όρο "οι καλύτεροι διαιτητές είναι αυτοί που δεν φαίνονται..."

Κεφάλαιο 3

ΠΕΡΙΓΡΑΦΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Σε αυτό το κεφάλαιο θα κάνουμε μία πλήρη περιγραφή της βάσης δεδομένων που χρησιμοποιήσαμε πάνω στην οποία εκτελούμε αναζήτηση. Πρέπει να αναφέρουμε οτι αυτή η βάση δεδομένων προέρχεται από την δημιουργία του αναγνωριστή φωνής (ASR) που υλοποιήθηκε από τον Τσεργούλα Ορφέα [5].

Προκειμένου να εκπαιδευτεί το ακουστικό μοντέλο του αναγνωριστή χρειάστηκαν αρκετά ακουστικά δεδομένα στα οποία εμείς μαζί και με κάποια αποτελέσματα του αναγνωριστή εκτελούμε αναζήτηση. Η βάση δεδομένων απαρτίζεται από βιντεοσκοπημένες τηλεοπτικές εκπομπές, και πιο συγκεκριμένα δελτία ειδήσεων. Συνεπώς σε αυτό το κεφάλαιο θα αναλύσουμε διεξοδικά τα τρία είδη αρχείων που επεξεργαζόμαστε (βίντεο, ηχητικά σήματα, κείμενο) και θα αναφερθούμε πιο συγκεκριμένα:

1. στον τρόπο και την πηγή μαγνητοσκόπησης τους
2. τα χαρακτηριστικά τους
3. την απομαγνητοφώνηση τους

Αρχικά πρέπει να ερμηνευθεί η ονομασία του κάθε αρχείου με σκοπό να γίνει πιο εύκολη η επεξεργασία του αργότερα. Τα στοιχεία που έπρεπε να τονιστουν περισσότερο και επομένως να συμπερηληφθούν στον τίτλο του κάθε αρχείου είναι τα εξής:

- ημερομηνία (χρόνος, μήνας, μέρα) που παρουσιάστηκε το δελτίο ειδήσεων.
- το κανάλι από το οποίο γράφτηκε το δελτίο ειδήσεων.
- το είδος της εκπομπής (ειδήσεις, talk skow κλπ).

- η ώρα παρουσίασης της εκπομπής .

Ενδεικτικά μπορούμε σαν παράδειγμα να παρουσιάσουμε ένα από τα ονόματα των αρχείων βίντεο που είναι μέρος της βάσης μας:

060627NETED2100.flv

(1) (2) (3) (4) (5) (6)

όπου:

(1) χρονος (2006)

(2) μήνας (06-Ιούνης)

(3) μέρα (27)

(4) κανάλι (NET)

(5) είδος εκπομπής (ειδήσεις)

(6) ώρα δελτίου (21:00)

3.1 ΑΡΧΕΙΑ ΒΙΝΤΕΟ

Συνολικά έχουμε σαράντα ώρες περίπου τηλεοτικών ειδήσεων. Η συγκεκριμένη κάρτα έχει δυνατότητα εγγραφής εικόνας και ήχουν σε μορφή MPEG 1,2,3 και 4. Επιπλέον έχει δυνατότητα εγγραφής βίντεο που επιτρέπει ,μεγαλύτερη συμπίεση και καλύτερη ποιότητα εικόνας από την κοινή μέθοδο (AVI). Τα αρχεία βίντεο που εγγράφηκαν προέρχονται από τα τηλεοπτικά κανάλια NET (Νέα Ελληνική Τηλεόραση), ET1 (Ελληνική Τηλεόραση 1) και ΣΚΑΪ.

Τα τεχνικά χαρακτηριστικά αυτών των αρχείων βίντεο είναι τα εξής:

- ανάλυση 640x480 1411kbps (πληροφορία ήχου)
- 16 bit audio sample size (πληροφορία ήχου)
- 44 kHz audio sample rate (πληροφορία ήχου)
- PCM audio format (πληροφορία ήχου)

- 25 frames/sec frame rate (πληροφορία video)
- 267kbps data rate (πληροφορία video)
- 16bit video sample size (πληροφορία video)
- MPEG-4 video format (πληροφορία video)

Πρέπει εδώ να αναφέρουμε οτι για τις ανάγκες τις αναζήτησης και επειδή η εφαρμογή θα ανέβαινε στο διαδίκτυο έπρεπε να προβούμε σε κάποιες αλλαγές στα αρχεία βίντεο. Αυτές ήταν αρχικά να μετατρέψουμε αλλάξουμε το bitrate των .avi αρχείων προκειμένου να μειωθεί το μέγεθος των αρχείων και κατά συνέπεια να μειωθεί ο χρόνος απόκρισης όταν αργότερα κάνουμε streaming. Το κόστος βέβαια σε αυτό ήταν να μειωθεί και η ποιότητα του βίντεο. Η αλλαγή του bitrate των αρχείων βίντεο έγινε με τον audio/video converter ffmpeg (http://ffmpeg.mplayerhq.hu) όπου αλλάξαμε το bitrate σε 64kbits/sec από 267kbits/sec που ήταν.

Ακόμα αυτό που η τροποποίηση που κάναμε σε σχέση με τα αρχικά βίντεο που υπάρχαν στην βάση ήταν να μετατρέψουμε τα αρχεία από .avi σε μορφή flash video (.flv) μιας και ο player που υλοποιήσαμε είναι flsash player θα δούμε αναλυτικά στο κεφάλαιο 5. Η μετατροπή από .avi σε flash γίνεται παρομοίως με το παραπάνω πρόγραμμα (ffmpeg).

Η τρίτη και τελευταία αλλαγή ονομάζεται metadata injection είναι η δημιουργία keyframes ανα μερικά χρονικά διαστήματα (στην περίπτωση μας κάθε 0.48 sec) ώστε να μπορεί ο player να εκτελεί αναζήτηση μέσα στο βίντεο (ουσιαστικά όταν εκτελούμε χρονική αναζήτηση μέσα στο βίντεο αναζητούμε το κοντινότερο στον χρόνο που θέλουμε keyframe).Αυτά θα τα δούμε αναλυτικότερα όμως στο κεφάλαιο 6.

3.2 ΑΡΧΕΙΑ ΗΧΟΥ

Για την απομαγνητοφώνηση (trancription)αλλά και τον τεμαχισμό (segmentation) των αρχείων χρησιμοποιήσαμε ακουστικά σήματα. Για να εξαχθεί η ακουστική πληροφορία (audio stream) χρησιμοποιήθηκε το πρόγραμμα virtual dub (http://www.virtualdub.org), έτσι ώστε να μπορέσουμε αργότερα να επεξεργαστούμε τα σήματα αυτά. Οπότε τελικά έχουμε σαράντα αρχεία ήχου (.wav files) διάρκειας περίπου μλιας ώρας το καθένα, με τα εξής χαρακτηριστικα:

- 256kbps bit rate

- 16bit audio sample size
- 1 channel (mono)
- 16 kHz audio sample rate
- PCM audio format

3.3 ΑΠΟΜΑΓΝΗΤΟΦΩΝΗΜΕΝΑ ΚΕΙΜΕΝΑ

Για να μπορέσουμε να επεξεργαστουμε τα αρχεία κειμένου όπως προέκυψαν από την απομαγνητοφώνηση και να εκτελέσουμε αναζήτηση σε αυτά θα πρέπει αρχικά να περιγράψουμε κάποια πράγματα για το πώς έγινε η απομαγνητοφώνηση.

Οπως είπαμε και παραπάνω η βάση δεδομένων πάνω στην οποία εκτελούμε αναζήτηση προκύπτει από κείμενα που προέρχονται από μη αυτόματη απομαγνητοφώνηση των δελτίων ειδήσεων αλλά και από κείμενα που προκύπτουν από αυτόματη απομαγνητοφωνηση των δελτίων ειδήσεων.

3.3.1 Μη αυτόματη Απομαγνητοφώνηση

Η μη αυτόματη απομαγνητοφώνησηη έγινε μέσω του εργαλείου Transcriber Tool (http://trans.sourceforge.net/). Το Transcriber Tool είναι ένα εργαλείο για τον χειροκίνητο χαρακτηρισμό σημάτων φωνής. Διαθέτει γραφικό περιβαλλον για τον τεμαχισμό, την απομαγνητοφώνηση και τον χαρακτηρισμό μεγάλων σε διάρκεια ακουστικών σημάτων.

Η απομαγνητοφώνηση των σημάτων ομιλίας έγινε σε τρία επίπεδα:

- σε επίπεδο συνθήκης ομιλιας
- σε επίπεδο ομιλητή
- σε επίπεδο απομαγνητοφώνησης

Επίπεδο συνθήκης ομιλίας:

Έχουμε ήδη σημειώσει πως μια από τις πολλές δυνατότητες που έχει το Transcriber Tool είναι ο τεμαχισμός ενός μεγάλου ακουστικού σήματος σε μικρότερα. Αυτά τα μικρότερα σήματα φωνής που δημιουργούνται θα πρέπει να περιγράφονται ειδικότερα, θα πρέπει να καταγράφεται κατά κάποιο τρόπο η συνθήκη που επικρατεί κατά την διάρκεια της ομιλίας.

Δηλαδή θα πρέπει να γνωρίζουμε ποια συνθήκη επικρατεί σε κάθε μια από τις προτάσεις, τα μικρά αρχεία ήχου, που θα δημιουργηθούν τελικά.

Οι συνθήκες αυτές είναι οι εξής:

report - καθαρή ομιλία στο στούντιο, χωρίς θόρυβο

music - ομιλία με μουσική στο background

noise - ομιλία με θόρυβο στο background

multi_speakers - ομιλία από πολλούς ομιλητές ταυτόχρονα

non_greek - ομιλία σε άλλη γλώσσα και όχι στα ελληνικά

non_trans - δεν υπάρχει ομιλία

επίπεδο ομιλητή:

Εφόσον τα σήματα φωνής που έχουμε στην κατοχή μας είναι ακουστικά σήματα από δελτία ειδήσεων, οι ομιλητές που περιέχονται σε αυτά είναι παρουσιαστές ειδήσεων, δημοσιογράφοι, απλοί πολίτες κλπ. Οπότε δόκιμο είναι για τον σχεδιασμό μιας αξιόπιστης βάσης δεδομένων να γίνεται περιγραφή του εκάστοτε ομιλητή. Τα χαρακτηριστικά που έχει ο κάθε ομιλητής είναι τα εξής:

- όνομα (name)
- φύλο (sex)
- εθνικότητα (dialect)

Τέλος, θα πρέπει να αναφέρουμε την περίπτωση που το όνομα του ομιλητή δεν είναι γνωστό. Αν ο ομιλητής είναι δημοσιογράφος τότε του δίνουμε το όνομα rep μαζί με έναν αριθμό που συμβολίζει τον αριθμό εμφανίσεων διαφορετικών δημοσιογράφων. Για παράδειγμα, για τον πρώτο δημοσιογράφο που θα συναντήσουμε στο σήμα φωνής, θα δώσουμε το όνομα rep_01, για τον δεύτερο το όνομα rep_02 κ.ο.κ. Αν ο ομιλητής δεν είναι δημοσιογράφος (π.χ. αν είναι ένας πολιτικός του οποίου δεν μπορούμε να αναγνωρίσουμε την ταυτότητα του ή ένας κοινός άνθρωπος), τότε του δίνουμε το όνομα unk μαζί με έναν αριθμό που συμβολίζει τον αριθμό εμφανίσεων τέτοιων προσώπων. Για παράδειγμα, για τον πρώτο που θα συναντήσουμε στο σήμα φωνής, θα δώσουμε το όνομα unk_01, για τον δεύτερο το όνομα unk_02 κ.ο.κ.

Επίπεδο απομανητοφώνησης:

Τώρα, αφού ήδη έχουμε περιγράψει τα δύο ανώτερα επίπεδα της απομαγνητοφώνησης, δηλαδή το επίπεδο της συνθήκης ομιλίας αλλά και το επίπεδο του ομιλητή, θα περιγράψουμε πως έγινε η ίδια η απομαγνητοφώνηση των ακουστικών σημάτων. Έτσι, θα περιγράψουμε τα βασικά βήματα-οδηγίες των απομαγνητοφωνήσεων που διενεργήθηκαν.

1. Ορθογραφία

Η κύρια και βασικότερη ιδέα στην εργασία των απομαγνητοφωνήσεων είναι να αποτυπώνουμε γραπτώς, στα ελληνικά, ότι ακριβώς ακούμε στα σήματα, φωνής. Τη διάρκεια λοιπόν αυτής της διαδικασίας δε γράφουμε με κεφαλαία, σημεία στίξης και επιπλέον γράφουμε τα πάντα με ελληνικούς χαρακτήρες.

Π.χ. Είναι πολύ trendy (Λάθος)
Είναι πολύ τρέντι (Σωστό)

2. Ακρώνυμα, Συντομεύσεις κτλ.

Στην περίπτωση ακρονύμων, συντομεύσεων κτλ. ενεργούμε παρομοίως. Δηλαδή απομαγνητοφωνούμε ότι ακριβώς ακούμε.

Π.χ. Αν στο σήμα φωνής ακούμε Η ΔΕΗ είναι μια κερδοφόρα επιχείρηση
Σωστό : Η δεή είναι μια κερδοφόρα επιχείρηση
Λάθος : Η δημόσια επιχείρηση ηλεκτρισμού είναι μια κερδοφόρα επιχείρηση

3.Αριθμοί

Παρομοίως, όπως στις παραπάνω περιπτώσεις.

Π.χ. Αν στο σήμα φωνής ακούμε '101 νύχτες'
Σωστό : 'εκατόν μία νύχτες'

4.Δισταγμοί ομιλητή

Στις περιπτώσεις που παρατηρείται στην ομιλία του ομιλητή δισταγμός γράφουμε την εξής ακολουθία χαρακτήρων @ε@

Π.χ. Ομιλητής : από την συμπεριφορά των εε καταναλωτών
Απομαγνητοφώνηση : από την συμπεριφορά των @ε@ καταναλωτών

5.Λανθασμένες προφορές λέξεων

Στην περίπτωση που ο ομιλητής προφέρει λανθασμένα μια λέξη, τότε η λέξη αυτή γράφεται με την σωστή της προφορά αλλά περικλείεται από τον χαρακτήρα '*' .

Π.χ. Ομιλητής : από την συμπαριφορά των καταναλωτών
Απομαγνητοφώνηση : από την *συμπεριφορά* των καταναλωτών

6. Ατελείς λέξεις

Λέξεις που έχουν ειπωθεί ατελώς γράφονται με την εξής ακολουθία χαρακτήρων : [FRAGMENT]

Π.χ. Ομιλητής : συμπεριφορά των κατ καταναλωτών
Απομαγνητοφώνηση : συμπεριφορά των [FRAGMENT] καταναλωτών

7. Στιγμιαίοι θόρυβοι

Σε περιπτώσεις όπου στην διάρκεια της ομιλίας έχουμε στιγμιαίους θόρυβους θα αποτυπώνονται σαν [NOISE]. Θα πρέπει να σημειώσουμε πως αυτή η περίπτωση έχει να κάνει μόνο για στιγμιαίους θορύβους και όχι για συνθήκη ομιλίας όπου αναφέραμε παραπάνω.
μερικές περιπτώσεις στιγμιαίων θορύβων είναι οι εξής;

side_speech Σημασία: Ομιλία μικρής διάρκειας από άλλον ομιλητή

phone_ring Σημασία: Κουδούνισμα τηλεφώνου

clear_throat Σημασία: Όταν ο ομιλητής καθαρίζει τον λαιμό του

paper_rustle Σημασία: Θόρυβος από χαρτιά

paff_noise Σημασία: Όταν ο ομιλητής μιλάει πολύ κοντά στο μικρόφωνο

8.Αναπνοή
Στην περίπτωση που ο ομιλητής εισπνεύσει ή εκπνεύσει και γίνει ηχητικά αντιληπτό, τότε στην απομαγνητοφώνηση γράφουμε [BREATH].

9.Άλλες περιπτώσεις

Σε περίπτωση που μια λέξη ή ένα μέρος της πρότασης δεν διακρίνεται καθαρά λόγω κακής ποιότητας του σήματος φωνής ή αδυναμία του ομιλητή τότε γράφουμε με όμοιο τρόπο με τις παραπάνω περιπτώσεις [TAG_BAD_READING].

Επίσης στην περίπτωση που έχουμε συνθήκη ομιλίας non-Greek είναι αυτονόητο πως δεν γίνεται καμία απομαγνητοφώνηση ενώ στο σχήμα 2.3 μπορούμε να δούμε το XML αρχείο που βγάζει σαν έξοδο το Transcriber Tool.

Είναι πολύ σημαντική η μελέτη του τρόπου απομανητοφώνησης μιας και η αναζήτηση εκτελείται ουσιαστικά στα .trs αρχεία και πρέπει να γνωρίζουμε τον τρόπο που είναι γραμμένα προκειμένου να θέσουμε και τους κατάλληλους περιορισμούς και στην αναζήτηση. Για παράδειγμα δεν μπορεί να γίνει αναζήτηση αριθμών αφού δεν υπάρχουν αριθμοί στα κείμενα αλλά μόνο ολογράφως ,πχ 5=πέντε, επομένως θα πρέπει να προτρέψουμε τον χρήστη να γράψει πέντε και όχι 5 στην περίπτωση που θέλει να εκτελέσει αναζήτηση με αυτόν τον αριθμό.

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
<Trans scribe="Songokou" audio_filename="060627NETED2100" version="6" version_date="061205">
<Topics> <Topic id="to1" desc="bg_noise"/>
<Topic id="to2" desc="NonGreek"/>
<Topic id="to3" desc="bg_music"/>
</Topics> <Speakers>
<Speaker id="spk1" name="Alabanos" check="no" type="male" dialect="native" accent="" scope="local"/>
<Speaker id="spk2" name="Alogoskoufis" check="no" type="male" dialect="native" accent="" scope="local"/>
<Speaker id="spk3" name="Apostolaki" check="no" type="female" dialect="native" accent="" scope="local"/>
<Speaker id="spk4" name="Athanasakis" check="no" type="male" dialect="native" accent="" scope="local"/>
<Speaker id="spk5" name="Bakogianni" check="no" type="female" dialect="native" accent="" scope="local"/>
.
.
.
</Speakers>
<Episode>
<Section type="report" startTime="0" endTime="29.8">
<Turn startTime="0" endTime="29.8" speaker="spk19" mode="planned" channel="studio" fidelity="high">
<Sync time="0"/> [FRAGMENT] το μεσημέρι στο κέντρο της αθήνας και διαδήλωση των μελών του πάμε στο λαγονήσι όπου συνεδριάζουν οι υπουργοί παιδείας του οοσά [BREATH] συνεχίστηκαν οι κινητοποιήσεις στο χώρο της ανώτατης εκπαίδευσης
<Sync time="10.354"/> κουκουλοφόροι [BREATH] προκάλεσαν πάλι επεισόδια γύρω από τη νομική και τα προπύλαια με αποτέλεσμα να τραυματιστούν τρεις φωτορεπόρτερ δυο δημοσιογράφοι [BREATH] και ένας εικονολήπτης
<Sync time="17.556"/> [BREATH] ένταση επικράτησε και στη βουλή κατά τη συζήτηση του προσχεδίου
<Sync time="20.876"/> [BREATH] από τη συνεδρίαση απείχε το κομμουνιστικό κόμμα ελλάδος ο συνασπισμός αποχώρησε [BREATH] ενώ το πασοκ ζήτησε να αναβληθεί για αύριο η συνεδρίαση που θα είναι παρούσα και η κυρία γιαννάκου [NOISE] </Turn>
</Section>
<Section type="nontrans" startTime="29.8" endTime="34.949">
<Turn startTime="29.8" endTime="34.949"> <Sync time="29.8"/>
</Turn>
</Section>
<Section type="report" topic="to1" startTime="34.949" endTime="47.223">
<Turn speaker="spk21" mode="planned" fidelity="high" channel="telephone" startTime="34.949" endTime="47.223">
<Sync time="34.949"/> έξω από τη νομική στην οδό σόλωνος και στα γύρω στενά κουκουλοφόροι επιδίδονται σε καταστροφές και επιθέσεις με βόμβες μολότωφ πέτρες και ότι βρίσκουν μπροστά τους εναντίον των αστυνομικών και των τηλεοπτικών συνεργείων
</Turn>
</Section>
.
.
.
```

Σχήμα 3.1: έξοδος από τον εργαλέιο μη αυτόματης απομαγνητοφώνησης

3.3.2 Αυτόματη Απομαγνητοφώνηση

Το Σύστημα Αυτόματης Αναγνώρισης των τηελοτικών δελτίων ειδήσεων αποτελεί και την διπλωματική εργασία του Τσεργούλα Ορφέα [5]. Παρακάτω θα περιγράψουμε συνοπτικά τον τρόπο λειτουργίας αυτού του Συστήματος αναγνώρισης. Το σύστημα αποτελείται από δύο μέρη, το γλωσσικό μοντέλο (language model) και το ακουστικό μοντέλο (acoustic model). Η παρατήρηση W μας δίνεται από τον τύπο :

$$\hat{W} = arg_w max P(W \mid X) = arg_w max \frac{P(W)\, P(X \mid W)}{P(X)}$$
$$\approx arg_w max P(W)\, P(X \mid W)$$

όπου το $P(W)$ ονομάζεται γλωσσικό μοντέλο και το $P(X \mid W)$ ονομάζεται ακουστικό μοντέλο.

Το γλωσσικό μοντέλο

Γενικότερα ένα γλωσσικό μοντέλο υπολογίζει την πιθανότητα εμφάνισης μιας λέξης, έχοντας παράλληλα πληροφορία για το περιεχόμενο στο οποίο αναμένεται να βρεθεί η λέξη. Η εργασία του γλωσσικού μοντέλου είναι να υπολογίζει την κατανομή της πιθανότητας $P(w)$ όπου αποτελεί την a-priori πιθανοτητα της παρατήρησης W ανεξάρτητα από το σήμα που παρατηρήθηκε με βάση κάποιο στατιστικό μοντέλο.

Τα στατιστικά γλωσσικά μοντέλα παρέχουν την κατανομή της πιθανότητας βασιζόμενα σε στατιστικά στοιχεία που έχουν συγκεντρωθεί από ένα μεγάλο κείμενο εκπαίδευσης (training set).

Για τον συγκεκριμένο αναγνωριστή προκειμένου να εκπαιδευτεί το γλωσσικό μοντέλο χρησιμοποιήθηκαν κείμενα από τρείς διαφορετικές εφημερίδες και συγκεκριμένα:

215 Mb κείμενο από την εφημερίδα “Ελευθεροτυπία”
170 Mb κείμενο από την εφημερίδα “Τα Νέα”
65 Mb κείμενο από την εφημερίδα “Το Βήμα”

Αφού το κείμενο “καθαρίστηκε” υπακούοντας σε κανόνες :

α)ορθογραφίας (να μήν υπάρχουν κεφαλάιοι χαρακτήρες)

β)συντομεύσεις (αντικατάσταση συντομέσυσεων με ολόκληρες τις λέξεις)

γ)αριθμοί (αντικατάσταση των αριθμητικών ψηφίων με γράμματα)

Κατόπιν κατασκευάστηκε με την βοήθεια εργαλείων της SRILM (http://www.speech.sri.com /projects/srilm/) κατασκευάστηκε το bigram γλωσσικό μοντέλο με λεξικό 60.000 λέξεων.

Το bigram γλωσσικό μοντέλο αποτελέι ένα από τα πιο πετυχημένα στατιστικά γλωσσικά μοντέλα. Σε αυτό το μοντέλο, η πιθανότητα μίας λέξης υπολογίζεται δημιουργώντας ένα μοντέλο Markov, του οποίου η κάθε κατάσταση βασίζεται στην παρουσία της προηγούμενης λέξης. Για παράδειγμα σε ένα απλό bigram μοντέλο, η πιθανότητα μίας λέξης ω δεδομένης της προηγούμενης λέξης δίνεται από τον τύπο:

$$P\left(w \mid w_{-1}\right) = \frac{c\left(w_{-1}w\right)}{c\left(w_{-1}\right)}$$

όπου $c\left(.\right)$είναι ο αριθμός εμφανίσεων μίας συγκεκριμένης ακολουθίας λέξεων στο κείμενο.

Τελος με την βοήθεια του εργαλείου HTK (Hidden Markov Model Toolkit) μετατράπηκε το κατασκευασμένο γλωσσικό μοντέλο στο format του HTK μιας το σύστημα αναγνώρισης υλοποιήθηκε εξ ολοκλήρου με το παραπάνω εργαλείο.

Παρακάτω μπορούμε να δούμε την ποιότητα του γλωσσικού μοντέλου μέσα από το perplexity αλλά και μέσα από τον ρυθμό λέξεων εκτός λεξιλογίου (OOV – Out Of Vocabulary). Το OOV δείχνει δεδομένου ενός αρχείου δοκιμής το ποσοστό των άγνωστων λέξεων που έχει το γλωσσικό μοντέλο.

Το perplexity ορίζεται ως :

$$PPT\left(T\right) = 2^{H\left(\hat{P}(T),P(T)\right)}$$

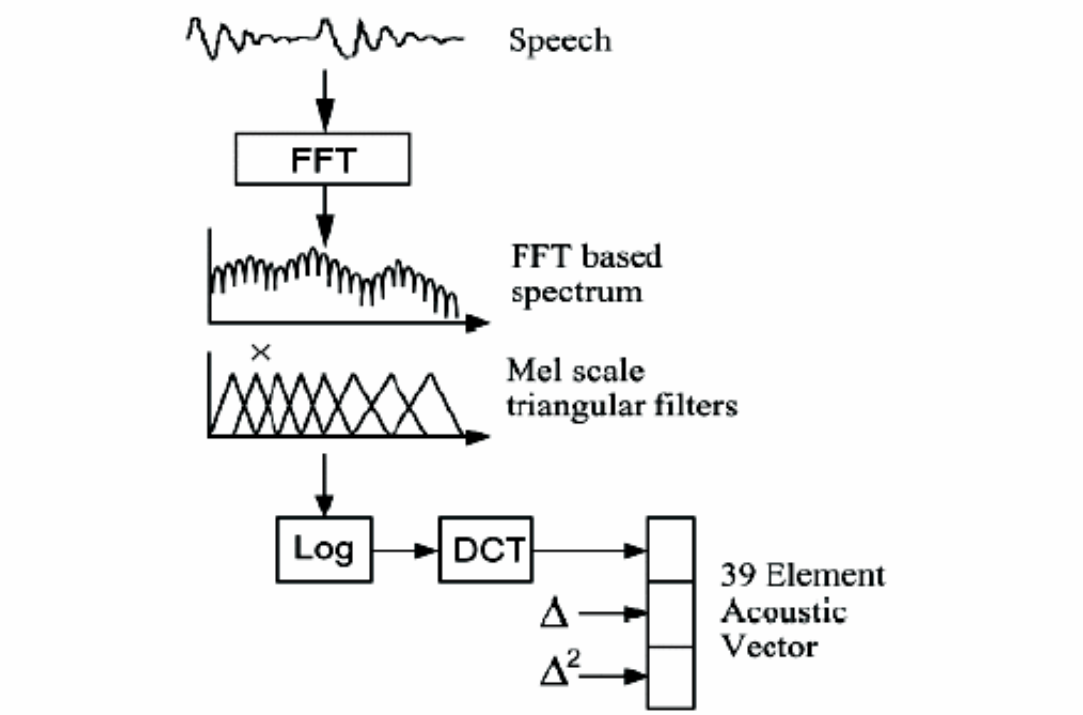
Ο όρος $H\left(\hat{P}\left(T\right),P\left(T\right)\right)$, δηλαδή η εντροπία $\hat{P}\left(T\right)$ που είναι η παρατηρούμενη κατανομή του κειμένου T, σε συνδυασμό με το P(T) που είναι η υπολογιζόμενη κατανομή του κειμένου T από το γλωσσικό μοντέλο, ορίζεται ως:

$$H\left(\hat{P}\left(T\right),P\left(T\right)\right) = -\sum_{x\in T}\hat{P}\left(x\right)logP\left(x\right)$$

Ετσι το perplexity καθώς και το OOV του γλωσσικού μοντέλου της συλλογής κειμένων από τις αντίστοιχες εφημερίδες είναι το εξής:

εφημερίδα	Perplexity	OOV(%)
Ελευθεροτυπία, Τα Νέα, Το Βήμα	211.4075	5.21

Το Ακουστικό μοντέλο



Σχήμα 3.2: διαδικασία εξαγωγής διανυσματικών ακολουθιών

Η δημιουργία του ακουστικού μοντέλου έγινε με την βοήθεια του εργαλείου HTK. Για την μοντελοποίηση της ανθρώπινης ομιλίας και συνεπώς για την υλοποίηση του ακουστικού μοντέλου χρησιμοποιήθηκαν Hidden Markov Models. Το πρώτο τμήμα της αναγνώρισης περιλαμβάνει την προεπεξεργασία του ψηφιοποιημένου σήματος ομιλίας (front-end) η οποία έγινε μέσα από την φασματική επεξεργασία σήματος Mel-Frequency Cepstral Coefficients (MFCC). Σε αυτήν την περίπτωση ανάλυσης χρησιμοποιείται μια μή γραμμική κλιμακα, που ονομάζεται Mel κλίμακα, που μιμείται και αναπαριστά το ακουστικό εύρος της ανθρώπινης ακοής. Η Mel κλίμακα μπορεί να προσεγγιστεί από τον ακόλουθο τύπο:

$$Mel\,(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right)$$

Στο σχήμα 3.2 μπορούμε να δούμε την διαδικασία εξαγωγής διανυσματικών ακολουθιών βασιζόμενα στην Mel-frequency

Αρχικά το σήμα μετασχηματίζεται σε φάσμα μέσω ενός μετασχηματιστού Fourier. Έπειτα, το παραγόμενο φάσμα του σήματος φωνής εξομαλύνεται,

περνώντας τους φασματικούς συντελεστές από τριγωνοειδείς συχνότητες που καθορίζονται από την Mel –frequency.

Στην συνέχεια, η έξοδος αυτού του φίλτρου περνάει από λογαριθμική συμπίεση έτσι ώστε το ενεργειακό φάσμα να γίνεται Γκαουσιανό. Τελικά, στο τελευταίο στάδιο της επεξεργασίας εφαρμόζεται διακριτός μετασχηματισμός συνημιτόνου (discrete consine transform – DCT).

Συνηθίζεται στις τελικές διανυσματικές ακολουθίες που παράγονται να προσθέτονται συντελεστές παραγώγου πρώτης και δεύτερης τάξης και κάποιες φορές να συμπεριλαμβάνεται και μια μέτρηση της ενέργειας του σήματος φωνής. Το παραπάνω υλοποιήθηκε μέσω του HTK και συγκεκριμένα μέσω της εντολής Hcopy όπου μετατράπηκαν τα σήματα φωνής σε διανυσματικές ακολουθίες

Δημιουργήθηκαν MFCCs για κάθε σήμα φωνής. Το κάθε MFCC περιέχει 39 στοιχεία, ανάμεσα σε αυτά συντελεστές παραγώγου πρώτης και δεύτερης τάξης , μετρήσεις της ενέργειας του σήματος φωνής. Τα MFCC υπολογίσθηκαν ανά 10 ms του σήματος φωνής χρησιμοποιώντας παράθυρο πλάτους 25ms.

Επόμενο βήμα για την κατασκευή του ακουστικού μοντέλου είναι επιλογή της γλωσσικής μονάδας. Επειδή το σύστημα πρόκειται για αναγνωριστή μεγάλου λεξιλογίου η καλύτερη λύση επιλογής γλωσσικής μονάδας είναι τα φωνήματα. Ακουστικά μοντέλα που έχουν σαν βάση τους φωνήματα μπορούν να εκπαιδευτούν ικανοποιητικά μόλις με μερικές εκατοντάδες προτάσεις εξασφαλίζοντας έτσι το κριτήριο της εκπαιδευσιμότητας. Αυτά τα μοντέλα είναι εξόρισμού γενικεύσιμα και αυτό γιατί τα φωνήματα είναι η βασική μονάδα από τα οποία προέρχονται οι λέξεις.

Η ακρίβεια είναι ένα σημαντικό θέμα στην περίπτωση των φωνημάτων, διότι κάθε φώνημα εξαρτάται και συγχρόνως επηρεάζεται από το γειτονικό δεξί και αριστερό φώνημα.

μετά ακολουθεί η δημιουργία και η εκπαίδευση των monophones, biphones και triphones. Η εκπαίδευση των μονόφωνων μοντέλων έγινε με τον αλγόριθμο Baum-Welch αφού έτρεξε για δέκα επαναλήψεις. Ομοίως εκπαιδεύτηκαν και τα biphones και triphones αφού πρώτα δημιουργήθηκαν με την εντολή HLED του HTK.

Κεφάλαιο 4

ΤΕΧΝΙΚΗ ΑΝΑΖΗΤΗΣΗΣ ΑΙΤΗΜΑΤΩΝ

Σε αυτό το κεφάλαιο θα περιγράψουμε τον τρόπο αναζήτησης και ανάκτησης δεδομένων που χρησιμοποιήσαμε για την κατασκευή της μηχανής αναζήτησης καθώς και θα αναλύσουμε τους λόγους για τους οποίους δεν χρησιμοποιήσαμε τελικά Vector Space Model για την επεξεργασία των δεδομένων.

4.1 Η ΠΡΟΣΕΓΓΙΣΗ ΜΑΣ

Το Vector Space Model είναι από τα ποιό γνωστά μοντέλα αυτόματης επιλογής κειμένων προς ανάκτηση είδικά για μηχανές αναζήτησης. Ολα τα κείμενα αποκτούν μία συγκεκριμένη δομή και είναι αρκετά ευκολο στην υλοποίηση.

Το μεγαλύτερο όμως μειονέκτημα του είναι οτι οι όροι εξετάζονται ανεξάρτητα ο ένας από τον άλλον. Ετσι για παράδειγμα μπορεί ένα κείμενο που περιέχει έναν όρο (ας πούμε με μεγάλη βαρύτητα) από τους τρείς που περιέχει ένα αίτημα να έχει μεγαλύτερο σκορ από ένα άλλο που περιέχει δύο όρους του αιτήματος. Αυτό όμως είναι μεγάλο μειονέκτημα για μία μηχανή αναζήτησης μιας και όταν ένας χρήστης εισάγει ένα αίτημα δεν θέτει καμμία βαρύτητα στους όρους που ψάχνει απλά θέλει να προηγούνται τα αποτελέσματα με τους περισσότερους όρους.

Ενα ακόμα μειονέκτημα του Vector Space Modeling είναι οτι δεν εξετάζει την απόσταση των όρων του αιτήματος (εφόσον αυτοί υπάρχουν) μέσα σε ένα κείμενο. Για παράδειγμα με την χρήση του VSM ένα κείμενο που περιείχε και τους δύο όρους ενός αιτήματος με απόσταση μεταξύ τους δέκα λέξεις θα είχε το ίδιο σκορ με ένα άλλο κείμενο που περιέχει μεν και τους δύο όρους του αιτήματος αλλά η απόσταση μεταξύ τους είναι 500 λέξεις (και πολύ πιθανόν οι δύο αυτοί όροι να ανήκουν σε άλλο θέμα ο καθένας). Αυτό είναι επίσης ένα ισχυρό μειονέκτημα που μας απέτρεψε από το να χρησιμοποιήσουμε ένα ένα τέτοιο μοντέλο.

Επομένως ψάξαμε να βρούμε έναν αλγόριθμο ο οποίος θα κρατούσε τα βάρη που χρησιμοποιούσε και το VSM (όπως η συχνότητα tf αλλά και η αντίστροφη συχνότητα idf) αλλά και θα χρησιμοποιεί σαν βάρος (με μεγάλη μάλιστα βαρύτητα) την απόσταση των λέξεων μεταξύ τους στο κείμενο αλλά και την ύπαρξη τους ή μη.

4.2 ΚΑΘΑΡΙΣΜΟΣ ΑΡΧΕΙΩΝ

Οπως είπαμε και στο πρώτο κεφάλαιο το πρώτο πράγμα που κάνουμε όταν κατασκευάζουμε μία μηχανή αναζήτησης είναι να φέρουμε τα δεδομένα της βάσης μας σε μία πρότυπη μορφή την οποία θα μπορέσουμε μετά να επεξεργαστούμε. Ακόμα είδαμε στο Κεφάλαιο 3 την μορφή που έχουν τα XML αρχεία με τα απομαγνητοφωνημένα κείμενα. Αυτά τα κείμενα περιέχουν πληροφορίες (όπως πχ ομιλητής, επίπεδο απομαγνητοφώνησης κλπ) τα οποια δεν μας χρειάζονται αλλά και άλλα (προτάσεις, χρόνος εκκίνησης πρότασης) τα οποία μας χρειάζονται.

Ετσι επεξεργαστήκαμε τα .trs αρχείου που έχουμε από τον Trancriber Tool και τα φέραμε σε μία ενιαία μορφή που εξυπηρετούν τις ανάγκες μας, ενώ ο τύπος

δεδομένων που χρησιμοποιήσαμε είναι .txt. Στο σχήμα 4.1 δίνεται ένα δείγμα από την μορφοποίηση των αρχείων.

0 = το μεσημέρι στο κέντρο της αθήνας και διαδήλωση των μελών του πάμε στο λαγονήσι όπου συνεδριάζουν οι υπουργοί παιδείας του οοσά συνεχίστηκαν οι κινητοποιήσεις στο χώρο της ανώτατης εκπαίδευσης
10.354 = κουκουλοφόροι προκάλεσαν πάλι επεισόδια γύρω από τη νομική και τα προπύλαια με αποτέλεσμα να τραυματιστούν τρεις φωτορεπόρτερ δυο δημοσιογράφοι και ένας εικονολήπτης
17.556 = ένταση επικράτησε και στη βουλή κατά τη συζήτηση του προσχεδίου
.
.
.
1384.757 = η νικήτρια του αποψινού αγώνα θα παίξει στα προημητελικά με τη βραζιλία
1388.281 = από μας καλό βράδυ

Σχήμα 4.1: μορφοποίηση αρχείου 060627NETED2100.trs

Επόμενο βήμα σύμφωνα με την σειρά που ορίσαμε στο κεφάλαιο 2 είναι η διαγραφή των σημασιολογικά "φτωχών" όρων από τα κείμενα. Επειδή όμως θέλουμε να κρατήσουμε τα αρχικά κείμενα (για την προβολή τους στον χρήστη εάν εκείνος το επιλέξει) δεν τα πανωγράψαμε αλλά χρησιμοποιήσαμε αντίγραφα τους. Σαν λίστα (stop list) με τους όρους προς διαγραφή χρησιμοποιήσαμε το αρχείο stop_word_list_noCapitals.txt όπου περιέχει περίπου 500 όρους. Παρακάτω μπορούμε να δούμε ένα δείγμα από την stoplist.

...και το να του η της με που την από για τα είναι των σε ο οι
στο θα τη στην τον τους δεν τις ένα μια ότι ή στη στα μας αλλά
στον στις αυτό όπως αν μπορεί μετά σας δύο τι ως κάθε πρέπει
πιο οποία μόνο ενώ ήταν ενός πολύ όμως κατά αυτή όταν μέσα
οποίο πως έτσι στους μέσω όλα καθώς αυτά προς ένας πριν μου
όχι χωρίς επίσης μεταξύ μέχρι έναν μιας αφού ακόμα...

Ετσι τα αρχεία κειμένου πήραν την μορφή που μπορούμε να δούμε στο σχήμα 4.2

0 = μεσημέρι κέντρο αθήνας διαδήλωση μελών πάμε λαγονήσι συνεδριάζουν υπουργοί παιδείας οοσά συνεχίστηκαν κινητοποιήσεις χώρο ανώτατης εκπαίδευσης
10.354 = κουκουλοφόροι προκάλεσαν πάλι επεισόδια γύρω νομική προπύλαια τραυματιστούν τρεις φωτορεπόρτερ δυο δημοσιογράφοι εικονολήπτης
17.556 = ένταση επικράτησε βουλή συζήτηση προσχεδίου...

Σχήμα 4.2: δείγμα 060627NETED2100.trs μετά από την stop list

4.3 ΑΦΑΙΡΕΣΗ ΕΠΙΘΕΜΑΤΩΝ

Επόμενο βήμα είναι η αφαίρεση των επιθεμάτων από τις λέξεις που έχουν μείνει. Πρέπει να πούμε οτι εμείς μετατρέψαμε τις λέξεις σε κεφαλάια και αφαιρέσαμε και όλους τους τόνους προκειμένου να έχουμε μία ενιαία μορφή.

Ενας Ελληνικός Stemmer που θα παρέχει μεγάλη ακρίβεια θα πρέπει να μπορεί να αντιμετωπίσει τα κλιτικά αλλά και τα παράγωγα επιθέματα (Kalamboukis 1995). Ομως η ελληνική γλώσσα είναι πολύ πλούσια στις παράγωγες λέξεις. Αυτό σημαίνει οτι πολλές λέξεις θα προέρχονται από το ίδιο στέμμα. Αν σκεφτούμε οτι κάθε όρος μπορεί να έχει πάνω από δέκα παράγωγα τέλη και περίπου πενήντα κλιτικά τέλη θα προκύψει μία λίστα από 500 περίπου όρους που θα ανήκουν στην ίδια οικογένεια και θα έχουν το ίδιο στέμμα. Αυτός ο τρόπος όμως δεν θα μας βοηθούσε ιδιαίτερα εμάς που θέλουμε να εφαρμόσουμε τον αλγόριθμο στην μηχανή Αναζήτησης αφού θα παράγει πολλά generic αποτελέσματα . Ο αλγόριθμος που χρησιμοποιήσαμε είναι κατάλληλος για μηχανές αναζήτησης αφού αντιμετωπίζει μόνο τα κλιτικά τέλη του όρου.

Για την αφάιρεση των επιθεμάτων (stemming) χρησιμοποιήσαμε τον stemmer του του Γ. Νταή [7]. Ο αλγόριθμός που χρησιμοποιεί βασίζεται στον αλγόριμθο του Porter [6] για αφαίρεση επιθεμάτων ο οποίος βέβαια αναπτύχθηκε για την αγγλική γλώσσα αλλά χρησιμοποιήθηκαν και κάποιοι από τους κανόνες και για την Ελληνική. Για την αφαίρεση των επιθεμάτων χρησιμοποιεί μία λίστα 166 επιθεμάτων και 22 σύνολα κανόνων ενώ στο παρακάτω σχήμα μορούμε να δούμε την αρχή λειτουργίας του Stemmer.

Σχήμα 4.3: αρχή λειτουργίας ενός αλγορίθμου αφαίρεσης επιθεμάτων

Ακόμα αυτός ο αλγόριθμος δεν αφαιρεί τα προθέματα τα οποία παρέχουν σημαντική πληροφορία για την λέξη και μπορούν να της αλλάξουν και την έννοια. Παρόλα αυτά κάποια ρήματα τα οποία έχουν κάποιο πρόθεμα όταν χρησιμοποιούνται σε παρελθοντικό χρόνο ,πχ το “ε”, τότε αυτό αφαιρείται. Γι αυτό τον λόγο γίνεται προσπάθεια να δημιουργηθούν δύο στέμματα για κάθε ρήμα (ένα για τους παρελθοντικούς χρόνους και ένα για τους υπόλοιπους). Στον πίνακα 4.1 μπορούμε να δούμε ένα παράδειγμα αυτού του stemmer .

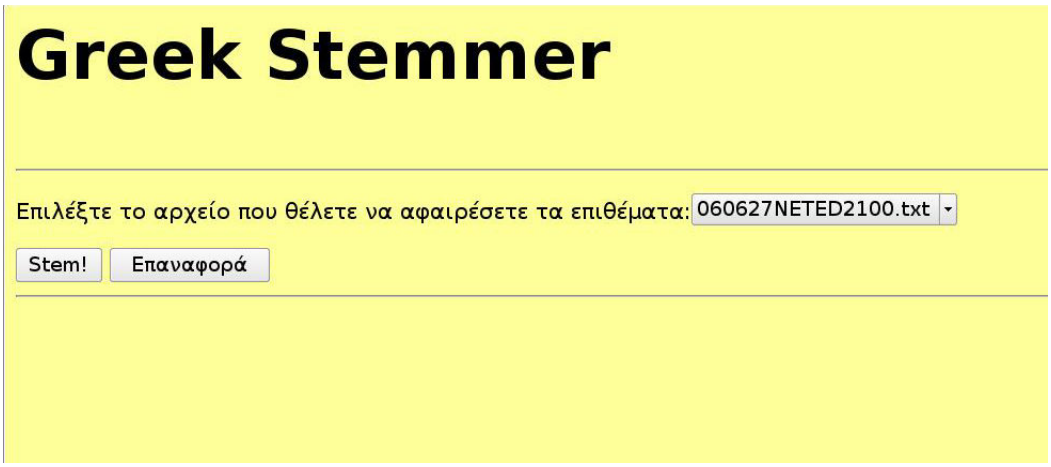
Tenses	Verb	Stems	Suffixes
Present Tenses	ΔΕΝΩ (I tide)	ΔΕΝ	Ω
	ΘΑ ΔΕΝΩ (I will be tiding)	ΔΕΝ	Ω
	ΔΕΝΕ (tide)	ΔΕΝ	Ε
	ΔΕΝΟΝΤΑΣ (tiding)	ΔΕΝ	ΟΝΤΑΣ
Past Tenses	ΕΔΕΣΑ (I tided)	ΔΕΣ	Α
	ΝΑ ΔΕΣΩ (to tide)	ΔΕΣ	Ω
	ΝΑ ΔΕΣΩ (to tide)	ΔΕΣ	Ω
	ΔΕΣΕ (tide)	ΔΕΣ	Ε
	ΔΕΣΕΙ (I have tide)	ΔΕΣ	ΕΙ

Πίνακας 4.1: παράδειγμα από τα αποτελέσματα του αλγόριθμου αφαίρεσης επιθεμάτων

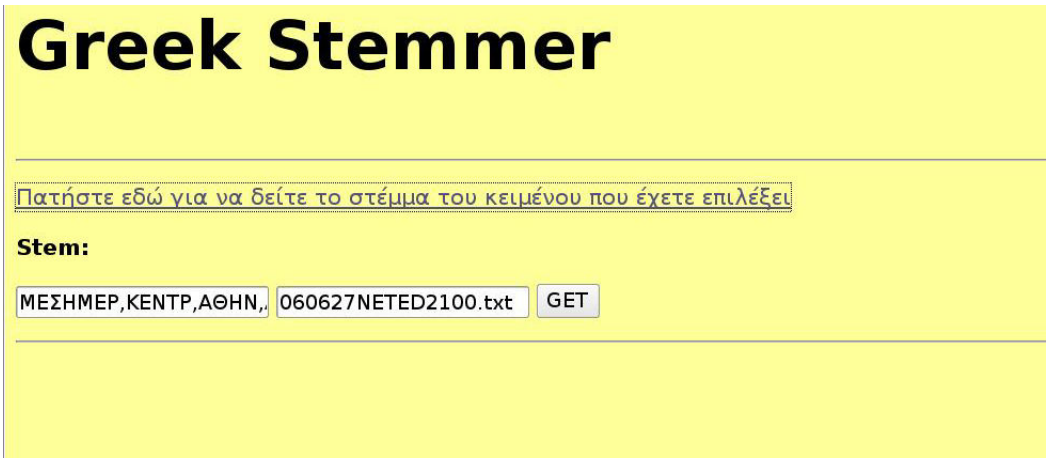
Η υλοποίηση του stemmer έχει γίνει σε γλώσσα προγραμματισμού javascript και δέχεται σαν είσοδο μία λέξη και παράγει έξοδο το στέμμα της λέξης. Επειδή όμως εμείς έχουμε ολόκληρα κείμενα με λέξεις που θέλουμε να βρούμε το στέμμα χρειάστηκε να τροποποιήσουμε λίγο τον κώδικα ώστε να μπορεί να δέχεται σαν είσοδο κείμενα και να παράγει πάλι κείμενα. μπορούμε στα σχήματα 4.4 και 4.5 να δούμε τον γραφικό περιβάλλον του stemmer όπως το τροποποιήσαμε για να εξυπηρετήσει τις ανάγκες μας.

Τέλος μπορούμε παρακάτω μπορούμε να δούμε ένα δείγμα από τα αρχεία που δημιουργούνται :

...μΕΣΗμΕΡ ΚΕΝΤΡ ΑΘΗΝ ΔΙΑΔΗΛΩΣ μΕΛ Π ΛΑΓΟΝΗΣ ΣΥΝΕΔΡΙΑΖ ΥΠΟΥΡΓ ΠΑΙΔΕΙ ΟΟΣ ΣΥΝΕΧΙΣΤ ΚΙΝΗΤΟΠΟΙΗΣ ΧΩΡ ΑΝ ΕΚΠΑΙΔΕΥΣ ΚΟΥΚΟΥΛΟΦΟΡ ΠΡΟΚΑΛΕΣ ΠΑΛ ΕΠΕΙΣΟΔ ΓΥΡ ΝΟμΙΚ ΠΡΟΠΥΛΑΙ ΤΡΑΥμΑΤΙΣΤ ΤΡ ΦΩΤΟΡΕΠΟΡΤΕΡ ΔΥΟ ΔΗμΟΣΙΟΓΡΑΦ ΕΙΚΟΝΟΛΗΠΤ ΕΝΤΑΣ ΕΠΙΚΡΑΤ ΒΟΥΛ ΣΥΖΗΤΗΣ ΠΡΟΣΧΕΔ ΣΥΝΕΔΡΙΑΣ ΑΠΕΙΧ ΚΟμμΟΥΝΙΣΤ ΚΟμμ ΕΛΛΑΔ ΣΥΝΑΣΠΙΣμ ΑΠΟΧΩΡ ΠΑΣΟΚ....



Σχήμα 4.4: επιλογή αρχείου που θέλουμε



Σχήμα 4.5: δημιουργία του αρχείου

4.4 ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ ΤΟΥ Kirsch

Επιστρέφουμε τώρα εκεί που μείναμε στην ενότητα 4.1 όπου εξετάσαμε τί είδους αλγόριθμο θα πρέπει να χρησιμοποιήσουμε. Οπως αναφέραμε και παραπάνω ο αλγόριμθμος που θα χρησιμοποιήσουμε θα πρέπει να λαμβάνει υπ όψιν του την απόσταση των λέξεων του αιτήματος ,εφ όσον εκείνοι υπάρχουν, σε κάποιο κείμενο αλλά και την ύπαρξη τους.

Αναλογιζόμενοι όλα τα παραπάνω χρησιμοποιήσαμε την τεχνική του Kirsch [8] για εύρεση ομοιότητας κειμένων για μηχανές αναζήτησης όπου χρησιμοποιούνται χαρακτηριστικά όπως ο συχνότητα των όρων στο κάθε κείμενο αλλά και σε όλη την βάση δεδομένων. Η μετρική που χρησιμοποιείται λαβάνει υπ όψιν της την συχνότητα των όρων του αιτήματος σε κάθε κείμενο, τον αριθμό των όρων του αιτήματος που βρίσκονται στο κάθε κείμενο αλλά και η εγγύτητα (απόσταση) των όρων του αιτήματος σε κάθε κείμενο.

Ο τύπος που περιγράφουμε είναι ο εξής:

$$R = c_1 N_p + \left(c_2 - \frac{\sum_{i=1}^{N_P - 1} \sum_{J=i+1}^{N_p} min\left(d\left(i,j\right), c_2\right)}{\sum_{k=1}^{N_p - 1} \left(N_P - k\right)} \right) / \frac{c_2}{c_1} + \frac{N_t}{c_3}$$

όπου N_P είναι ο αριθμός των όρων του αιτήματος που βρίσκονται στο κείμενο (κάθε όρος μετριέται μόνο μία φορά), N_t είναι ο συνολικός αριθμός των όρων στο κάθε κείμενο (κάθε όρος μετριέται όσες φορές συναντάται στο αντίστοιχο κείμενο), $d\left(i,j\right)$ είναι η ελάχιστη απόσταση μεταξύ του i-οστού και j-οστού όρου του αιτήματος που βρίσκονται στο κείμενο (σαν απόσταση μετράμε την διαφορά των όρων που βρίσκονται ανάμεσα τους, και πέρνουμε το συνολικό άθροισμα των διαφορων μεταξύ των όρων για να βρούμε την συνολική απόσταση). Ακόμα χρησιμοποιούμε κάποιες σταθερές που αποτελούν τα βάρη του κάθε χαρακτηριστικού του αλγόριθομου. Αυτές οι σταθερές είναι η c_1 όπου αποτελέι το συνολικό βάρος του R , το c_2 που προσδιορίζει την μέγιστη απόσταση μεταξύ των όρων που θεωρείται χρήσιμη (υπολογίσιμη), και η c_3 αναπαριστά μία σταθερά που προσδιορίζει την σημαντικότητα της συχνότητας των όρων (term frequency).

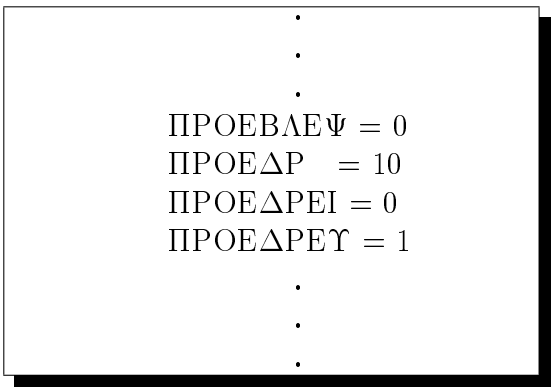
Ο αλγόριθμος του Kirch προτείνει τις παρακάτω τιμές για τις σταθερές:

- C_1=100
- C_2=5000

- C_3=10*C_1

Οταν υπάρχει μόνο ένας όρος στο αίτημα του χρήστη (πράγμα που είναι αρκετά πιθανό στην μηχανή αναζήτησης) τότε σαν απόσταση μεταξύ των όρων χρησιμοποιούμε την απόσταση του όρου από την αρχή του κειμένου, το οποίο και αποτελεί δείκτη ομοιότητας.

Προκειμένου να βρούμε την συχνότητα εμφάνισης των όρων του αιτήματος σε κάποιο κείμενο δγμιουργήσαμε αρχεία που περιέχουν όλους τους όρους της βάσης ,δηλαδή το λεξιλόγιο της βάσης μας, και δίπλα από κάθε όρο την συχνότητα εμφάνισης του στο αντίστοιχο κείμενο. Στο σχήμα 4.6 μπορούμε να δούμε την δομή κάποιου από αυτά τα αρχεία.



Σχήμα 4.6: παράδείγμα αρχείου που περιέχει την συχνότητα όρων

Από τα ίδια αρχεία βρήκαμε και την ύπαρξη ή μη του όρου μέσα σε ένα κείμενο (Nt), με τον απλό συλλογισμό οτι εάν Np>0 τότε προφανώς υπάρχει ενώ εάν Np=0 δεν υπάρχει.

Ακόμα σημαντική πληροφορία για εμάς είναι οχι μόνο να βρούμε τα συναφή με το αίτημα του χρήστη κείμενα αλλά και να βρούμε και το σημείο όπου βρίσκεται ποιο κοντά το θέμα ή η φράση η οποία ψάχνει.
Για να το κάνουμε αυτό κρατήσαμε την θέση του όρου εκείνου που βρίσκεται ποιο κοντά στην αρχή του κειμένου και έχει την μικρότερη απόσταση με τους άλλους όρους του αιτήματος και κατόπιν με χρήση πινάκων (σχήμα 4.7) που φτιάξαμε και περιέχουν την χρονική στιγμή που ακούγεται μία λέξη κάναμε αυτήν την αναγωγή, από θέση σε χρόνο. Την πληροφορία για την χρονική στιγμή που ακούγεται η λέξη (ή καλύτερα η χρονική στιγμή που ακούγεται η πρόταση που περιέχει αυτήν την λέξη) την έχουμε πάρει από τα XML αρχεία που παράγονται από το Transcriber Tool.

Αυτός ο αλγόριθμος μας βοηθάει αρκετά στο να βρίσκουμε αποτελέσματα με τους όρους να βρίσκονται κοντά μεταξύ τους αλλά και με τον τρόπο που τον

10.354 = 16 17 18 19 20 21 22 23 24 25 26 27 28
17.556 = 29 30 31 32 33
20.876 = 34 35 36 37 38 39 40 41 42 43 44 45 46
34.949 = 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

Σχήμα 4.7: παράδειγμα αρχείου που περιέχει τις χρονικές στιγμές που ακούγονται οι αντίστοιχες λέξεις (τις λέξεις τις έχουμε αντικαταστήσει με τις θέσεις τους)

χρησιμοποιούμε αυτή την τεχνική μπορούμε και να βρούμε το καλύτερο δυνατό σημείο μέσα στο δελτίο ειδήσεων που θα ικανοποιήσει τον χρήστη. Μπορούμε στο σχήμα 4.8 να δούμε ένα παράδειγμα από την χρήση της παραπάνω τεχνικής πάνω στην βάση μας.

μπορούμε παρατηρώντας το παραπάνω παράδειγμα να πούμε οτι το καλύτερο σκορ το έφερε το δελτίο ειδήσεων με τίτλο 060627NETED2100 όπου έφερε σκορ 400.013 και περιέχει και τους 3 όρους του αιτήματος (οι όροι η, του, στο θεωρούνται μη σημαντικοί και εξαιρούνται από την αναζήτηση), οι οποίοι εμφανίζονται 19 φορές, με ελάχιστη απόσταση μεταξύ τους 10 λέξεις. Ενώ η πρώτη λέξη λέξη που συναντάται στο κειμενο (στο συγκεκριμένο ειναι η λέξη ΕΠΙΣΚΕΨΗ) βρίσκεται στην θέση 856 που σημαίνει κοιτάζοντας τους πίνακες με τις χρονικές στιγμές οτι βρίσκεται στο 749.445 sec. όπου είναι και η χρονική στιγμή που θα ξεκινήσει το βίντεο εφ όσον το επιθυμεί ο χρήστης (θα τα δούμε αυτό αναλυτικά στο κεφάλαιο 6).

query="Η ΕΠΙΣΚΕΨΗ ΤΟΥ ΠΑΠΑΝΔΡΕΟΥ ΣΤΟ ΙΣΡΑΗΛ"
ΑΠΟΤΕΛΕΣΜΑΤΑ: (11 ΣΥΝΟΛΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ)

file=**060627NETEΔ2100**
Np:3
Nt:19
min distance:10
start word:856
start time:749.445
relevance:400.013
file=**060703NETEΔ2100**
Np:3
Nt:13
min distance:686
start word:1460
start time:1188.744
relevance::399.6014
file=**060628NETEΔ2100**
Np:3
Nt:11
min distance:704
start word:239
start time:189.095
relevance::399.5886
file=**060729NETEΔ2100**
Np:3
Nt:30
min distance:916
start word:1691
start time:1722.08
relevance:399.4804
file=**060730NETEΔ2100**
Np:3
Nt:30
min distance:1472
start word:1332
start time:1379.725
relevance:399.1468

.

.

.

file=**060710ΣΚΑΕΔ2030**
Np:2
Nt:4
min distance:509
start word:1365
start time:1077.905
relevance:299.9022
file=**060801ET1EΔ2300**
Np:1
Nt:19
min distance:0
start word:75
start time:61.03

relevance:200.019

Σχήμα 4.8: παράδειγμα υλοποίησης αλγορίθμου αναζήτησης

Chapter 5

ΤΕΜΑΧΙΣΜΟΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

Σε αυτό το κεφάλαιο θα περιγράψουμε τον τρόπο με τον οποίο τεμαχίσαμε τα δελτία ειδήσεων σε θεματικές ενότητες. Ο τεμαχισμός των δεδομένων σε θεματικές ενότητες έγινε προκειμένου να παρέχουμε στον χρήστη επιπλέον πληροφορίες επεξεργαζόμενοι τα δελτία ειδήσεων. Ο τεμαχισμός (segmentation) χωρίζεται σε δύο μέρη.

Πρώτα κάναμε segmentation στα κείμενα ανά θεματικές ενότητες και κατόπιν προκειμένου να βελτιώσουμε τα άκρα κάθε θεματικής ενότητας (αρχή και τέλος) τα οποία αποτελούν κρίσιμα σημεία κάναμε segmentation στα αρχεία ήχου. Έτσι τα σημεία που αποτελούν αρχή και τέλος των θεματικών ενοτήτων "διορθώνονται" με το κοντινότερο segment που προκύπτει από τα αρχεία ήχου. Πρέπει να πούμε οτι ο τεμαχισμός των αρχείων ήχου γίνεται ανά παύση ή αλλαγή περιβάλλοντος όπως θα περιγράψουμε παρακάτω. Έτσι στο τέλος συνδυάζουμε τα δύο είδη τεμαχισμού προκειμένου καλύτερα αποτελέσματα στις θεματικές ενότητες.

5.1 ΤΕΜΑΧΣΙΜΟΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΚΕΙΜΕΝΟΥ ΑΝΑ ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ

Η μέθοδος που χρησιμοποιήσαμε για τον τεμαχισμό είναι η TextTiling [9]. Η TextTiling αποτελεί μία μέθοδο χωρισμού αρχείων κειμένου σε πολυ μεγάλες παραγραφικές μονάδες. Η προσέγγιση αυτή χρησιμοποιεί λεξικολογική ανάλυση των όρων που περιέχουν τα κέιμενα και μοντελοποίηση τους γίνεται στον δυανυσματικό χώρο (Vector Space Modeling) χρησιμοποιώντας την μετρική $tf.idf$, ένα μέτρο ανάκτησης πληροφοριών. Για την εύρεση της ομοιότητας χρησιμοποιείται cosine similarity metrics η κατανομή της οποίας που καθορίζει το

εύρος της κάθε ενότητας.

Ο αλγόριθμος αυτός που χρησιμοποιούμε προσπαθεί να χωρίσει τα δελτία είδήσεων όπως θα τα χώριζε και κάποιος άνθρωπος, γι αυτό θα πρέπει να συμπεριλάβουμε και το στοιχείο της υποκειμενικότητας στα αποτελέσματα.

Ο αλγόριθμος αυτός περιλαμβάνει δύο βήματα. Στο πρώτο βήμα όλα τα γειτονικά μπλοκ λέξεων συγκρίνονται μεταξύ τους. Κάθε μπλοκ λέξεων σύμφωνα με τον αλγόριθμο περιλαμβάνει 3 με 5 προτάσεις, βέβαια εμείς λόγω έλλειψεις σαφών ορίων προτάσεων (αφού πρώτον δεν είναι γραπτά κείμενα αλλά αποφωνημένοι λόγοι και δεύτερον δεν υπάρχουν σημεία στίξης) οριοθετούμε το κάθε μπλοκ στις 45 λέξεις. Κάθε μπλοκ πρότασης αποτελεί και ένα δυάνυσμα. Αφού συγκριθούν πέρνουν μια τιμή συνάφειας (similarity value) και όταν συγκριθούν όλα μεταξύ τους δημιουργείται μία ακολουθία από τιμές (μεταξύ των γειτονικών μπλοκ).

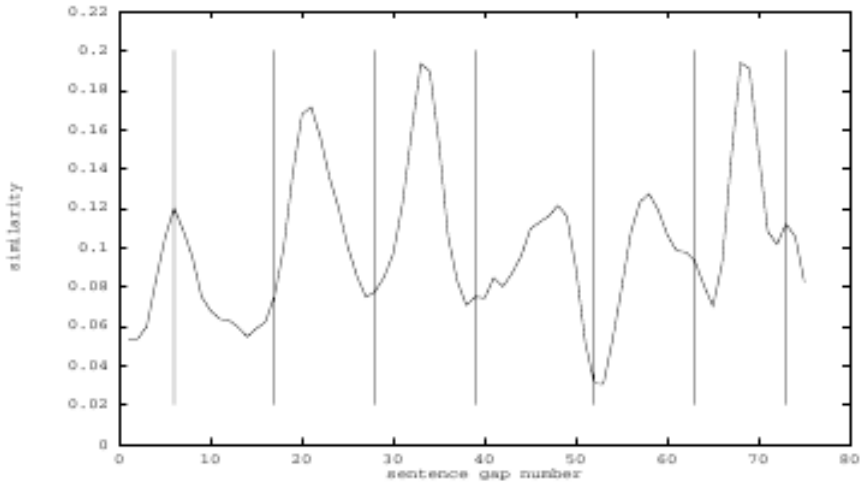
Κατόπιν δημιουργείται η γραφική παράσταση (σχήμα 5.1) που αναπαριστά τα μπλοκ σε σχέση με την συνάφεια τους και αναζητώνται κοιλάδες και κορυφές. Εαν γειτονικά μπλοκ εμφανίζουν μεγάλη ομοιότητα μεταξύ τους (δηλαδή ανήκουν και στην ίδια ενότητα) δημιουργούν κορυφές, ενώ οι κοιλάδες αναπαριστουν τα σημεία αλλαγής θεματικής ενότητας αφού είναι τα σημεία όπου τα γειτονικά μπλοκ εμφανίζουν την μικρότερη ή και καθόλου ομοιότητα. Οι κοιλάδες αποτελούν και τα κρίσιμα σημεία της θεματικής ενότητας γι αυτό και θα προσπαθήσουμε να τα βελτιώσουμε με audio segmentation όπως θα πούμε παρακάτω.

Όπως αναφέραμε και παραπάνω η μετρική με την οποία μετράμε την ομοιότητα των μπλοκ είναι η$tf.idf$. Η περιγραφή αυτών των μετρικών έχει γίνει στο κεφάλαιο 2.

Εμείς δοκιμάσαμε τον παραπάνω αλγόριθμο εξετάζοντας τρείς μετρικές για να επιλέξουμε την καταλληλότερη. Αρχικά χρησιμοποιήσαμε σαν μετρική απλά την συχνότητα των όρων (tf), κατόπιν χρησιμοποιήσαμε σαν μετρική την$tf.idf$ όπου σαν κέιμενο (για τον υπολογισμό του idf) θεωρήσαμε κάθε μπλοκ λέξεων. Και τέλος η τρίτη μετρική που χρησιμοποιήσαμε (η οποία ήταν και αυτή στην οποία καταληξαμε) είναι η $tf.idf$ αλλά σαν κείμενο χρησιμοποιούμε το κάθε segment.

Προφανώς αυτή η υλοποίηση προϋποθέτει δύο περάσματα αφού στο πρώτο πέρασμα δημιουργούνται τα πρώτα segments με χρήση της tf σαν βάρος και στο δεύτερο πέρασμα δημιουργούμε νέα segment χρησιμοποιώντας σαν κείμενα για τον υπολογισμό του idf τα παλιά segments.

Αυτός ο τρόπος μας βοηθάει στο να ξεχωρίσουμε την τοπική από την ολική διάσταση ενός όρου. Και με αυτό εννοούμε οτι εάν ένας όρος εμφανίζεται συχνά μέσα σε ένα μπλοκ αλλά όχι τόσο συχνά μέσα σε ένα μεγαλύτερο πλαίσιο



Σχήμα 5.1: διάγραμμα κατανομής των μλοκ σε σχέση με την ομοιότητα τους

(segment) τότε θεωρείται σημαντική για το μπλοκ. Ετσι συμπαιρένουμε οτι εάν δύο γειτονικά μπλοκ έχουν αρκετούς κοινούς όρους και αυτοί οι όροι έχουν μεγάλη βαρύτητα (από τον υπολογισμό του $tf.idf$τους) τότε υπάρχει μεγάλη πιθανότητα αυτά τα δύο μπλοκ να ανήκουν στην ίδια θεματική ενότητα.

Η ομοιότητα μεταξύ δύο γειτονικών μπλοκ υπολογίζεται με βάση τον συνημίτονο της γωνίας που σχηματίζουν τα δύο διανύσματα μπλοκ (βλέπε κεφάλαιο 2.3).Έτσι δεδομένων δύο μπλοκ b1 και b2 έχουμε:

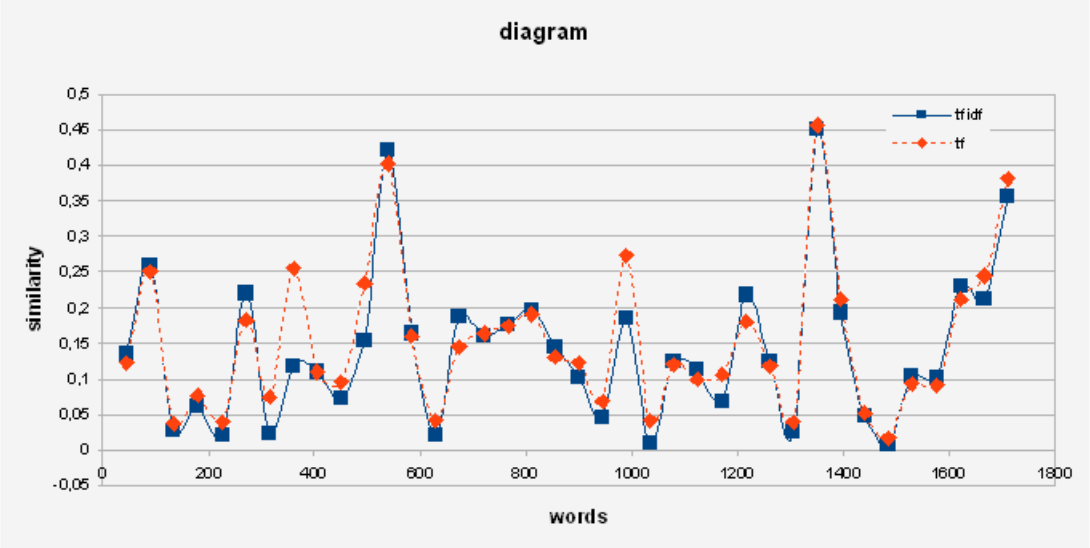
$$cos\,(b_1, b_2) = \frac{\sum_{t=1}^{n} w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_{t=1}^{n} w_{t,b_1}^2 \sum_{t=1}^{n} w_{t,b_2}^2}}$$

όπου το t αποτελεί τον αύξων αριθμό του κάθε όρου και το $wt, b1$ αποτελεί το $tf.idf$ βάρος του όρου t στο μπλοκ $b1$. Ετσι θεωρούμε ότι όσο μεγαλύτερη είναι η ομοιότητα μεταξύ δύο μπλοκ τόσο πιο συναφή είναι επομένως και ανήκουν στο ίδιο θέμα.

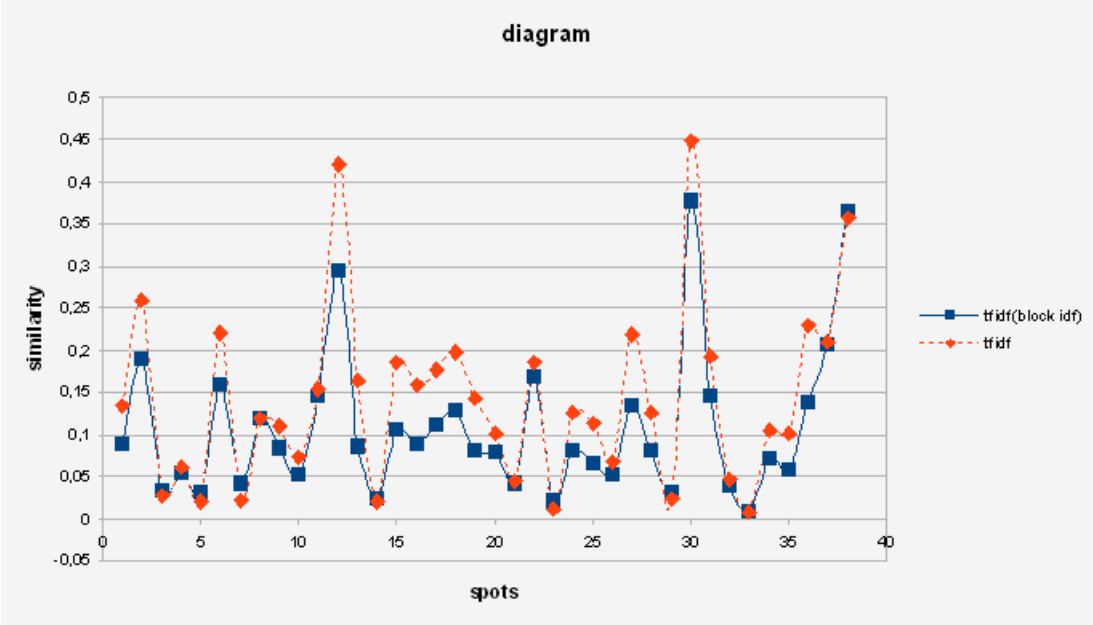
Όπως είπαμε και παραπάνω αφού συγκρίνουμε όλα τα γειτονικά μπλοκ και βρούμε την ομοιότητα μεταξύ τους μετά δημιουργούμε την γραφική παράσταση μεταξύ των μπλοκ λέξεων σε σχεση με την ομοιότητα. Έτσι εάν για παράδειγμα μετρηθεί η ομοιότητα μεταξύ δύο μπλοκ έστω του b και του $b+1$ όπου το b περιέχει λέξεις από το i μέχρι το $i+k-1$ και το $b+1$ λέξεις από το $i+k$ μέχρι το $i+2k-1$ ενώ το k προσδιορίζει τον αριθμό των λέξεων του κάθε μπλοκ τότε το σημείο που θα προσδιορίζει αυτά τα δύο μπλοκ στον χ-άξονα θα είναι στο μεταξύ του σημείου $i+k-1$ και $i+k$. Αυτό το σημείο το ονομάζουμε

gap number. Ο y-άξονας παίρνει τιμές από 0 έως το 1 που είναι και η τιμές που μπορεί να πάρει το συνημίτονο.

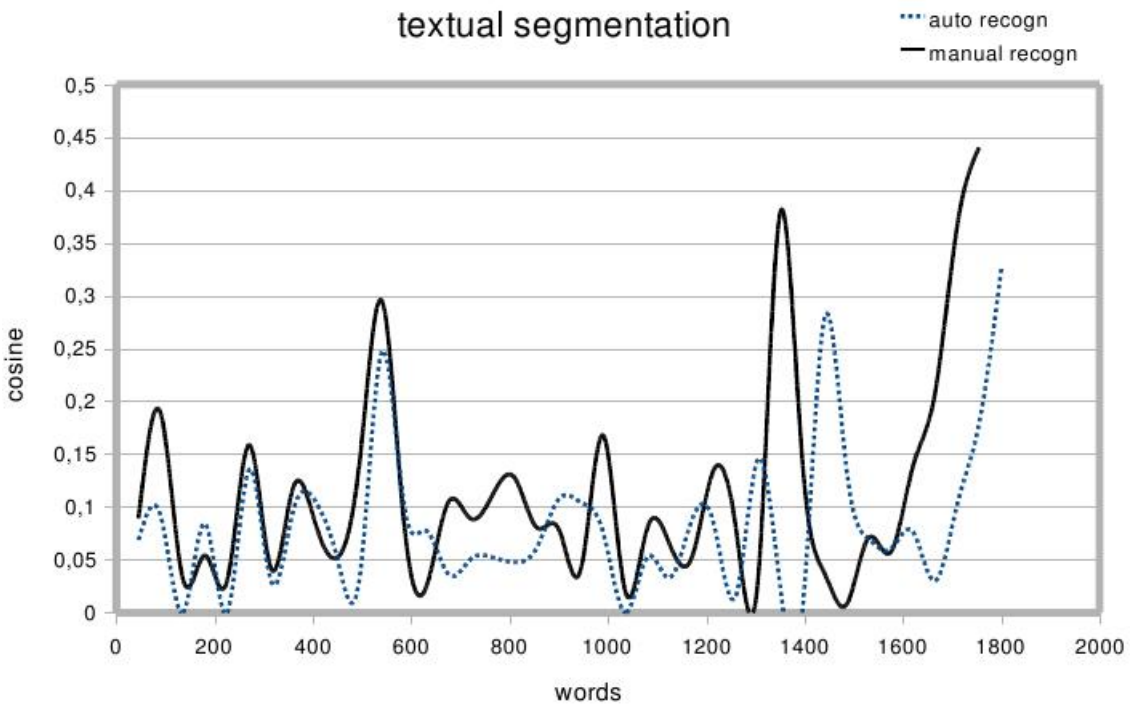
Στο σχήμα 5.2 και 5.3 βλέπουμε τις γραφικες παραστασεις όπως δημιουργήθηκαν από την ομοιότητα των μπλοκ με την χρηση διαφορων βαρων, στο δελτίο ειδήσεων 060627NETED2100. Ακόμα στο σχήμα 5.4 μπορούμε να δούμε τον συνδυασμό αυτού του δελτίου όπως προέκυψε από την μή αυτόματη απομαγνητοφώνηση αλλά και από τον αναγνωριστή. μπορούμε να συμπαιράνουμε οτι η προσέγγιση είναι αρκετά καλή.



Σχήμα 5.2: γραφική παρασταση απ όπου προκύτουν οι ενότητες του 060627NETED2100.trs κάνοντας χρήση των βαρών tf kai tfidf



Σχήμα 5.3: γραφική παρασταση απ όπου προκύτουν οι ενότητες του 060627NETED2100.trs κάνοντας χρήση των βαρών tfidf και tfidf αλλά με διαφορετικό πεδίο document



Σχήμα 5.4: σύγκριση κειμένου από αναγνωριστή αλλά και από μή αυτόματη απομαγνητοφώνηση

Ο αλγόριθμος Texttiling υλοποιήθηκε με χρήση της γλώσσας Perl και προκειμένου να χωρίσουμε μηχανικά τα segments (χωρίς να δημιουργούμε γραφικές παραστάσεις για κάθε κείμενο) θεωρήσαμε οτι ένα σημείο θεωρείται αρχή/τέλος μίας θεματικής ενότητας εφ όσον το προηγούενο του είναι μεγαλύτερο από αυτό όπως και το επόμενο αλλά και έχει τιμή μικρότερη 0.05. Θέσαμε οτι εφόσον έχουν ομοιότητα μεγαλύτερη από 0.05 τότε οι κοιλάδες που σχηματίζονται είναι τοπικές και ανήκουν στο ίδιο θέμα.

5.2 ΤΕΜΑΧΙΣΜΟΣ ΤΩΝ ΑΡΧΕΙΩΝ ΗΧΟΥ

Όπως αναφέραμε και στην προηγούμενη ενότητα, επειδή τα κείμενα προέρχονται από τον προφορικό λόγο, δεν είναι σαφές πότε αρχίζει και πότε τελειώνει μία πρόταση (αφου δεν υπάρχουν σημεία στίξης). Ετσι τα σημεία που ξεκινάει και που τελειώνει κάθε θεματική ενότητα δεν είναι τόσο ακριβή.

Προκειμένου να βελτιώσουμε αυτά τα σημεία θα χρησιμοποιήσουμε έναν αλγόριθμο τεμαχισμού των αρχείων ήχου. Ο τεμαχισμός γίνεται σε επίπεδο

περιβάλλοντος, ομιλητή και παύσης. Έτσι αντί για σημεία στίξης θα ψάξουμε τέτοια σημεία για την έναρξη κάθε θεματικής ενότητας.

Η μέθοδος που ακολουθήσαμε είναι αυτή του Bayes Information Criterion (BIC) [17] και χρησιμοποιήσαμε τον κώδικα του Τσεργούλα Ορφέα. Η τεχνική αυτή προυποθέτει την front-end προεπεξεργασία του σήματος η οποία έγινε μέσα από την φασματική επεξεργασία σήματος Mel-Frequency Cepstral Coefficients (MFCC). Αρχικά το σήμα μετασχηματίζεται σε φάσμα μέσω ενός μετασχηματιστού Fourier. Έπειτα, το παραγόμενο φάσμα του σήματος φωνής εξομαλύνεται, περνώντας τους φασματικούς συντελεστές από τριγωνοειδείς συχνότητες που καθορίζονται από την Mel –frequency. Στην συνέχεια, η έξοδος αυτού του φίλτρου περνάει από λογαριθμική συμπίεση έτσι ώστε το ενεργειακό φάσμα να γίνεται Γκαουσιανό.

Για να κατανοήσουμε την μέθοδο αυτή αρχικά θα αναφερθούμε περιληπτικά στο πως χρησιμοποιείται στην στατιστική το Model Selection Criteria που χρησιμοποιούμε.

Στις στατιστική, προκειμένου να περιγραφεί ένα ιδιαίτερο σύνολο δεδομένων, κάποιος μπορεί να χρησιμοποιήσει τις non-parametric μεθόδους ή τις παραμετρικές μεθόδους. Στις παραμετρικές μεθόδους, υπάρχουν διάφορα μοντέλα με διαφορετικό αριθμό παραμέτρων για να αντιπροσωπεύσουν ένα σύνολο δεδομένων. Ο αριθμός παραμέτρων σε ένα μοντέλο διαδραματίζει έναν σημαντικό ρόλο. Το likelihood αυξάνεται όταν αυξάνεται ο αριθμός παραμέτρων στο μοντέλο αλλά αυτό μπορεί να οδηγήσει σε λάθος (overtraining problem) εάν ο αριθμός παραμέτρων είναι πάρα πολύ μεγάλος. Προκειμένου να καταπολεμηθεί αυτό το πρόβλημα κάποιος μπορεί να χρησιμοποιήσει το Bayes Information Criterion (παραμετρική μέθοδος) που είναι ένα στατιστικό κριτήριο για επιλογή προτυπου και χρησιμοποιεί το likelihood για τον υπολογισμό ενώ επιβαρύνεται με τον αριθμό των παραμέτρων του μοντέλου.

Εάν για παράδειγμα θέσουμε σαν $\chi = \{x_i : i = 1, 2, ..., N\}$τό σύνολο των δεδομένων το οποίο θέλουμε να μοντελοποιήσουμε και σαν

$M = \{M_i : i = 1, 2, ..., K\}$ το σύνολο των υποψήφιων παραμετροποιημένων μοντέλων τότε το BIC criterion ορίζεται ώς:

$$BIC\,(M) = logL\,(X, M) - \lambda \frac{1}{2} \alpha\,(M) \times log\,(N)$$

όπου το $\alpha\,(M)$ αποτελεί τον αριθμό των παραμέτρων για το μοντέλο M. Ακόμα το λ είναι το penalty weight και ισούται με 1 ($\lambda = 1$). Έτσι για τα συγκεκριμένα δεδομένα θα επιλέξουμε το μοντέλο με το μεγαλύτερο BIC.

Επομένως αφού αναλύσαμε συνοπτικά την αρχή λειτουργίας του BIC μπορούμε να συνχίσουμε περιγράφοντας πως θα εντοπίσουμε ένα από τα κριτήρια αλλαγής segment μέσω του αλγορίθομου BIC.

Έστω ότι $\chi = \left\{x_i \in R^d, i = 1, ..., N\right\}$η ακολουθία των cepstral διανυσμάτων όπως εξήχθησαν από το αρχείο ήχου. Ακόμα υποθέτουμε οτι το Q περιγράφεται από Gaussian ανάλυση:

$$x_i \sim N\left(m_i, \Sigma_i\right)$$

όπου m_i είναι το μέσο δυάνυσμα και το Σ είναι ο πίνακας συνδιακύμανσης.

Αρχικά θα αναλύσουμε την περίπτωση που υπάρχει μόνο έναν σημείο αλλαγής segment και κατόπιν θα γενικέυσουμε τον αλγόριθμο και για παραπάνω σημεία.

Έτσι λοιπόν υποθέτοντας οτι η αλλαγή συμβάινει την χρονική στιγμή i τότε:

$$H_0 = x_1 \ldots x_N \sim N\left(\mu, \Sigma\right)$$

και

$$H_1 = x_1 \ldots x_i \sim N\left(\mu_1, \Sigma_1\right); x_{i+1} ... x_N \sim N\left(\mu_2, \Sigma_2\right)$$

Η maximum likelihood ratio υπολογίζεται ώς:

$$R_i = Nlog \mid \Sigma \mid -N_1 log \mid \Sigma_1 \mid -N_2 log \mid \Sigma_2 \mid$$

όπου το Σ, Σ_1και Σ_2 είναι οι πίνακες συνδιακύμανσης των δειγμάτων από όλα τα δεδομένα, από $\{q_1 ... q_i\}$και από $\{q_{i+1} ... q_N\}$ αντίστοιχα. Επομένως η εκτίμηση της maximum likelihood του σημείου αλλαγής segment είναι:

$$\hat{t} = argmax_i R\left(i\right)$$

Εάν τώρα εξετάσουμε το πρόβλημα από την σκοπιά του model selection τότε έχουμε να συγκρίνουμε δύο μοντέλα. Το ένα μοντελοποιεί τα δεδομένα σαν δύο Gaussian και το άλλο μοντελοποιεί τα δεδομένα σαν ένα Gaussian. Ετσι το BIC μεταξύ αυτών των δύο μοντέλων μπορεί να εκφραστεί ως εξής:

$$BIC\,(i) = R\,(i) - \lambda P$$

όπου το P όρίζεται ώς:

$$P = \frac{1}{2}\left(d + \frac{1}{2}d\,(d+1)\right) logN$$

καιλ = 1,d είναι οι διαστάσεις του χώρου. Εάν το Bic είναι θετικό τότε το μοντέλο των δύο Gaussian είναι καλύτερο για την αναπαράσταση των δεδομένων, και συμπεραίνουμε οτι έχει γίνει αλλαγή. Έτσι γενικότερα ελέγχουμε για το εάν υπάρχει αλλαγή εάν

$$\{max_i BIC\,(i)\} > 0$$

Ακόμα το σημείο που γίνεται αλλαγή του segment μπορεί να εκφραστεί ως (m.l.e):

$$\hat{t} = argmax_i BIC\,(i)$$

Έαν τώρα θέλουμε να ενοπίσουμε τα σημεία αλλαγής μέσα σε μία ακολουθία, τότε ακολουθούμε τα παρακάτω βήματα:

1. αρχικοποιούμε ένα διάστημα $[a, b] : a = 1; b = 2$.
2. εντοπίζουμε μέσω του BIC εάν υπάρχε μέσα στο διάστημα $[a, b]$ κάποιο σημείο αλλαγής.
3. έαν δεν υπάρχει τότε $b = b + 1$ αλλιώς το σημείο αλλαγής είναι το t και θέτουμε $a = t + 1$ και $b = a + 1$.
4. γυρίζουμε στο βήμα 2.

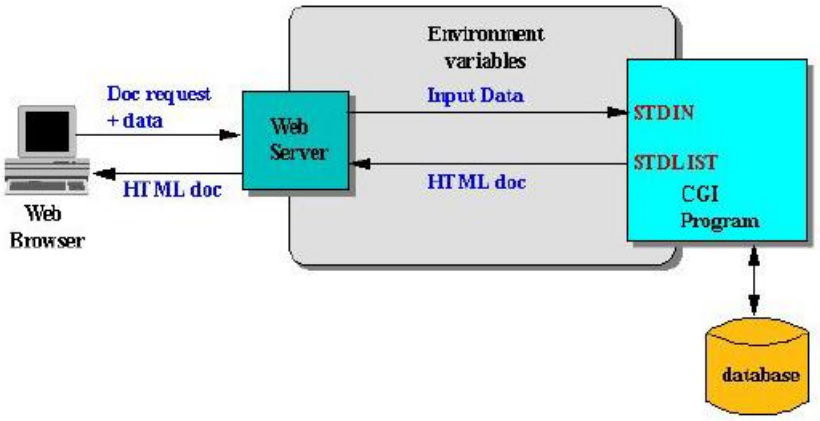
Ο αλγόριθμος υλοποιήθηκε κάνοντας χρήση των εργαλείων της Matlab. Επειδή δεν ήταν δυνατόν να χρησιμοποιήσουμε τόσο μεγάλα δεδομένα αρχικά κόψαμε τα αρχεία ήχου σε μικρότερα μέσω του εργαλείου Julius [16]και κατόπιν κάθε ένα από τα αρχεία ήχου που δημιουργήθηκε το τεμαχίσαμε μεα βάση τον αλγόριθμο BIC. Αυτό που κρατήσαμε από τον τεμαχισμό των αρχείων ήχου δεν ήταν τα ίδια τα αρχεία που δμηιουργούνται αλλά ηπληροφορία για το πότε ξεκινάει ένα segment και πότε τελειώνει.

Κεφάλαιο 6

ΣΧΕΔΙΑΣΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Αφού επεξεργαστήκαμε τα δεδομένα της βάσης μας προκειμένου να μπορέσουμε να τα προβάλουμε και υλοποιήσαμε τον αλγόριθμο αναζήτησης, θα ακολουθήσουμε το επόμενο βήμα στην κατασκευή μίας μηχανής αναζήτησης που είναι η ένωση όλων αυτών των εφαρμογών και η δημιουργία του γραφικού περιβάλλοντος ώστε να μπορεί να το χρησιμοποιεί ο χρήστης.

Σε αυτό το κεφάλαιο θα περιγράψουμε τα εργαλεία και τις τεχνικές που χρησιμοποιήσαμε προκειμένου να δημιουργήσουμε ένα γραφικό περιβάλλον φιλικό προς τον χρήστη αλλά και τον τρόπο με τον οποίο ανεβάσαμε την εφαρμογή μας στο διαδίκτυο. Τέλος θα παρουσιάσουμε συνολικά ένα παράδειγμα χρήσης της μηχανής αναζήτησης.



Σχήμα 6.1: τρόπος λειτουργίας CGI

6.1 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ

- **CGI:**

Η αλληλεπίδραση μεταξύ χρήστη και μηχανής αναζήτησης έγινε μέσω του Common Gateway Interface (CGI) [11]. Το CGI αποτελεί το μέρος εκείνο του εξυπηρετητή ιστού που εκτελεί την επικοινωνία μεταξύ προγραμμάτων που τρέχουν στον εξυπηρετητή. με CGI, ένας εξυπηρετητής μπορεί να καλέσει ένα πρόγραμμα περνώντας στο πρόγραμμα χαρακτηριστικά δεδομένα του χρήστη (όπως για παράδειγμα το IP του χρήστη ή τα δεδομένα εισόδου που έχει εισάγει ο χρήστης μέσω μίας φόρμας HTML). Στο σχήμα 6.1 μπορούμε να δούμε μία απλή εφαρμογή που χρησιμοποιεί CGI προκειμένου να κατανοήσουμε την αρχή λειτουργίας του.

Οι λόγοι που χρησιμοιήθηκε το παρόν πρωτόκολλο για την αλληλεπίδραση μετξύ και χρήστη είναι οι εξής:

- Έχει πολύ καλή ανταπόκριση στην χρήση και επεξεργασία φορμών. Οι φόρμες είναι υποσύνολα γραμμένα σε HTML τα οποία επιτρέπουν στον χρήστη να παρέχει πληροφορίες. Η φόρμα μετατρέπει το διαδίκτυο μία διαδραστική διαδικασία μεταξύ του χρήστη και του παροχέα. Στην περίπτωση μας η χρήση της φόρμας γίνεται προκειμένου ο χρήστης να εισάγει το αίτημα αναζήτησης (search query) αλλά και τα διάφορα κριτήρια αναζήτησης.

- Η δυνατότητα δημιουργίας δυναμικών αρχείων. Τα δυναμικά ή εικονικά αρχεία (virtual documents) δημιουργούνται κατά της διάρκεια επεξερ-

γασίας των δεδομένων ως απάντηση στο αίτημα του χηστη. Στην περίπτωση μας αυτό είναι απολύτως απαραίτητο μιας και τα δεδομένα που προβάλονται στον χρήστη δημιουργούνται εκείνη την στιγμή (εικόνες, βίντεο, κείμενα).

- Τρίτος και ίσως βασικότερος λόγος είναι η πλήρης συμβατότητα του CGI με την γλώσσα Perl (γλώσσα την οποία χρησιμοποιούμε για την δημιουργία των server side εφαρμογών).

- **Apache Server**

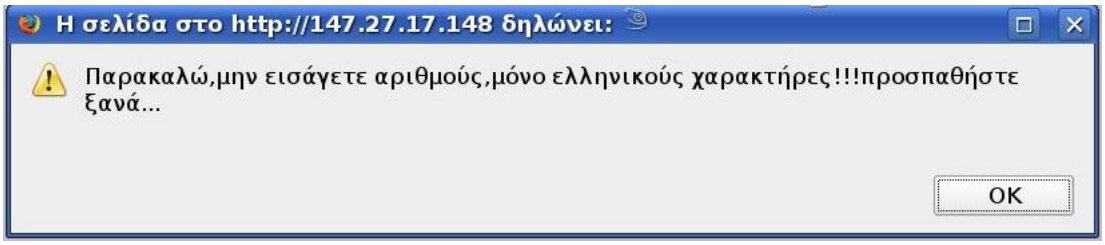
Για την υλοποίηση του γραφικού περιβάλλοντος χρησιμοποιήσαμε τον πολύ γνωστό HTTP Apache Server [10]. Ο εξυπητερητής Apache αποτελεί μία αξιόπιστη πηγή λογισμικού που στοχεύει στην δημιουργία μίας γερής και ελεύθερα διαθέσιμης εφαρμογής εξυπηρετητή ιστού (HTTP Server). Η πρώτη επίσημη έκδοση του Apache Server διανεμήθηκε τον απρίλη του 1995 και βασίζεται στο NCSA httpd 1.3 (ένας εξυπηρετητής που δημιουργήθηκε στο Πανεπιστήμιο του Ιλινόις).

Σήμερα η τελευταία έκδοση την οποία χρησιμοποιούμε και εμείς στην εφαρμογή μας είναι η Apache Server ver 2.2 όπου σε σχέση με τις προηγούμενες εκδόσεις η διαδικασία ρύθμισης των παραμέτρων έχει απλοποιηθεί αρκετά ενώ έχει δυνατότητα να υποστηρίζει μεγάλα αρχεία (μέχρι 2Gbyte) και ακόμα έχουν προστεθεί κάποια επιπλέον χαρακτηριστικά.

Πρέπει να πούμε οτι ο Apache παρέχει πλήρη υποστήριξη στο CGI αλλά και στα αρχεία Perl αρκεί να γίνουν οι κατάλληλες ρυθμίσεις ώστε να επιτρέπεται η εκτέλεση τους. Στο Παράρτημα Α μπορούμε να δούμε πως εγκαθιστούμε τον Apache αλλά και τις κατάλληλες ρυθμίσεις που εκτελούμε για να δέχεται προγράμματα CGI.

Εκτός από τα παραπάνω, χρησιμοποιήσαμε και κάποια προγράμματα που μας βοήθησαν στην απεικόνιση των αποτελεσμάτων.

Αρχικά χρησιμοποιήσαμε τον mplayer [14] για την δημιουργία εικόνων (screen shots) για την περιγραφή του βίντεο. Αναλυτικότερα η εικόνα που δημιουργείται προέρχεται από το σημείο που θα αρχίσει να παίζει το βίντεο με βάση τα αποτελέσματα της αναζήτησης (κεφάλαιο 5). Έτσι δημιουργείται ένα frame την χρονική στιγμή time του βίντεο. Το frame όμως που δημιουργείται είναι σε μορφή Portable Pixel Map, η οποία είναι μορφή που δεν μπορεί να προβληθεί από κάποιον broswer. Προκειμένου να λύσουμε το πρόβλημα του τύπου δεδομένων αλλά και να αλλάξουμε το όνομα του αρχείου χρησιμοποιήσαμε το ImageMagick 6.3.5 (http://www.imagemagick.org).



Σχήμα 6.2: παράθυρο που αναδύεται σε περίπτωση λάθους εισαγωγής αιτήματος

Τέλος για την προβολή του βίντεο (όπως έχουμε πει από το δεύτερο κεφάλαιο χρησιμοποιούμε flash video format) χρησιμοποιούμε τον jwplayer [15]. Ο jwplayer αποτελεί έναν player για flash video ο οποίος υποστηρίζει http streaming over php όπως θα περιγράψουμε αναλυτικότερα αργότερα (ενότητα 6.5).

6.2 ΚΑΝΟΝΕΣ ΣΧΕΔΙΑΣΗΣ WEB ΕΦΑΡμΟΓΗΣ

Όπως συνοπτικά αναφέραμε και στην ενότητα 2.6 πολύ σπουδαίο ρόλο στην επιτυχυμένη λειτουργία μίας μηχανής αναζήτησης είναι η σχεδίαση του *front-end* ώστε να ικανοποιεί τις απαιτήσεις ενός χρήστη [12]. Έναν χρήστη δεν τον ενδιαφέρει τόσο ο τρόπος που λειτουργεί ο αλγόριθμος αναζήτησης όσο η εύκολη και πλήρης πρόσβαση στα αποτελέσματα αλλά και η εύκολη χρήση του εργαλείου.

Κατά την σχεδίαση του γραφικού περιβάλλοντος προσπαθήσαμε να κάνουμε το σύστημα όσο πιο φιλικό προς τον χρήστη γίνεται, ενώ παράλληλα προσπαθήσαμε να δώσουμε στον χρήστη αρκετή πληροφορία με απλό και κατανοητό τρόπο ώστε εκείνος να μείνει ευχαριστημένος με το αποτέλεσμα.

Τα παραπάνω τα πετύχαμε αρχικά ξεχωρίζοντας τον αρχάριο από τον έμπειρο χρήστη. Ένας αρχάριος χρήστης το μόνο που έχει να κάνει προκειμένου να διενεργηθεί μία αναζήτηση είναι να εισάγει ένα αίτημα χωρίς να χρειάζεται να προβληματιστεί για τα κριτήρια αναζήτησης. Ακόμα δεν κάνουμε διάκριση μεταξύ πεζών και κεφαλαίων γραμμάτων ούτε λαμβάνουμε υπόψιν μας εάν οι λέξεις είναι τονισμένες ή οχι.

Σε περίπτωση που ο χρήστης εισάγει χαρακτήρες που δεν είναι δεκτοί από το σύστημα (αριθμοί, ειδικοί ή λατινικοί χαρακτήρες), τότε εμφανίζεται παράθυρο σχήμα 6.2 που του επισημαίνει το λάθος και τον επιστρέφει στην φόρμα συμπλήρωσης.

Όσον αφορά τα κριτήρια αναζήτησης υπάρχει μία προεπιλεγμένη μορφή που

Σχήμα 6.3: παράδειγμα πεδίου εισαγωγής νέου αιτήματος

χρησιμοποιείται σε περίπτωση που ο χρήστης δεν γνωρίζει τί ακριβώς ψάχνει. Σε περίπτωση που υπάρχει ένας πιο έμπειρος χρήστης ισχύουν όλα όσα είπαμε για τον αρχάριο συν του ότι μπορεί ο χρήστης αλλάξει τα κριτήρια αναζήτησης με δικά του.

Ακόμα σε κάθε φύλλο αποτελεσμάτων ο χρήστης μπορεί να δεί το αίτημα με το οποίο διενέργησε αναζήτηση (σχήμα 6.3) (είναι πολύ συνηθισμένο όταν ένας χρήστης ο οποίος διενεργεί μία αναζήτηση να ξεχάσει καθώς ψάχνει τα αποτελέσματα το ακριβές αίτημα αναζήτησης, με αποτέλεσμα να υπάρχει σύγχυση) αλλά και να εκτελέσει μία καινούρια αναζήτηση χωρίς να χρειάζεται να επιστρέψει στην αρχική φόρμα.

Επιπλέον στο φύλλο παρουσίασης αποτελεσμάτων ο χρήστης μπορεί να δεί τα αποτελέσματα με σειρά ομοιότητας (ξεκινώντας προφανώς από το πιο όμοιο με το αίτημα του) και να έχει άμεση πρόσβαση σε όλα τα στοιχεία του δελτίου ειδήσεων (ημερομηνία, κανάλι, θεματική ενότητα στο οποίο ανήκει, φωτογραφία που αναπαριστά το σημείο που θα ξεκινήσει το δελτίο) με τρόπο σαφή και κατανοητό.

Πρέπει εδώ να προσθέσουμε οτι και η γλώσσα που χρησιμοποιούμε είναι απλή. Η μηχανή αναζήτησης από την στιγμή που είναι ανοιχτή στο διαδίκτυο δεν πρέπει να χρησιμοποιεί ορολογία ξένη σε κάποιον απλό χρήστη χωρίς εξειδικευμένες γνώσεις από διαχείριση βάσεων δεδομένων ή λειτουργία τεχνικών αναζήτησης.

Τέλος, βαρύτητα δόθηκε και στην επιλογή των χρωμάτων κατά την σχεδίαση του γραφικού περιβάλλοντος. Σαν φόντο χρησιμοποιήθηκε μία εικόνα με σκούρες αποχρώσεις του μπλέ το οποίο από την μία παραπέμπει στο χρώμα του φίλμ και απο την άλλη δεν προκαλεί σύγχυση ή εκνευρισμό στον χρήστη.

6.3 ΠΕΡΙΓΡΑΦΗ ΦΟΡΜΑΣ ΣΥΜΠΛΗΡΩΣΗΣ ΑΙΤΗΜΑΤΩΝ

Όπως αναφέραμε και στην παραπάνω ενότητα η φόρμα συμπλήρωσης του αιτήματος σχεδιάστηκε έτσι ώστε να προορίζεται και για έναν αρχάριο αλλά και για έναν πιο έμπειρο χρήστη. Αρχικά πρέπει να πούμε οτι υπάρχουν κάποιοι περιορισμοί στην εισαγωγή του αιτήματος. Αυτοί οι περιορισμοί προκύπτουν

από τα απομαγνητοφωνημένα κείμενα.

Έτσι δεν υπάρχουν αριθμητικά ψηφία στα κείμενα και εφόσον θέλουμε ένα αίτημα να περιέχει αριθμούς πρέπει να το γράψουμε ολογράφως.

πχ αίτημα: 5 Ιουλίου λάθος
αίτημα: πέντε Ιουλίου σωστό

Ακόμα δεν υπάρχουν ειδικοί χαρακτήρες (τελείες, κόμματα κλπ)

πχ αίτημα: πόλεμος,λίβανος λάθος
αίτημα: πόλεμος λίβανος σωστό

Τέλος όλες οι λέξεις πρέπει να είναι γραμμένες στα ελληνικά (ακόμα και οι ξένες)

πχ αίτημα: *supermarket* λάθος
αίτημα: σουπερμάρκετ σωστό

Σε περίπτωση που κάποιος χρήστης εισάγει κάτι από τα παραπάνω του επιστρέφεται μήνυμα λάθους και επιστρέφει στην αρχική σελίδα.

Τα φίλτρα που χρησιμοποιούμε προκειμένου να εξειδικεύσουμε περισσότερο την αναζήτηση είναι τα εξής:

1. εμφάνιση αποτελεσμάτων με όλες τις λέξεις του αιτήματος ή με οποιαδήποτε λέξη (η προεπιλεγμένη μορφή είναι με οποιαδήποτε λέξη)

2. η αναζήτηση να διενεργηθεί σε δελτία ειδήσεων από όλα τα διαθέσιμα κανάλια ή ο χρήστης μπορεί να επιλέξει κανάλι αναζήτησης (η προεπιλεγμένη μορφή είναι σε όλα τα κανάλια)

3. η αναζήτηση να διενεργηθεί σε δελτία ειδήσεων από συγκεκριμένες ημερομηνίες (η προεπιλεγμένη μορφή είναι από 01/01/2006 μέχρι 31/12/2010, δηλαδή όλα τα δελτία που περιλαμβάνει η βάση)

6.4 ΄ΕΛΕΓΧΟΣ ΕΦΑΡΜΟΓΗΣ ΑΠΟ ΤΟΝ ΧΡΗΣΤΗ

Ένας χρήστης θα πρέπει κάθε στιγμή να νοιώθει οτι έχει τον έλεγχο της εφαρμογής. Στην δική μας μηχανή αναζήτησης αυτό επιτυγχάνεται για δύο κυρίως

Σχήμα 6.4: μενού ελέγχου ροής δελτίου ειδήσεων

λόγους. Αρχικά ο χρήστης μπορεί να εισάγει ειδικά κριτήρια αναζήτησης θέτοντας έτσι τους δικούς του περιορισμούς στην αναζήτηση. Ακόμα κατά την διάρκεια προβολής του δελτίου ειδήσεων ο χρήστης μπορεί να ελέγχει την ροή του βίντεο, μέσα από ειδικό μενού (σχήμα 6.4).

Το μενού αυτό του παρέχει τις εξής δυνατότητες:

- να σταματάει το βίντεο όποτε θέλει και να το ξαναρχίζει (pause/start button),
- να ρυθμίζει την ένταση του ήχου (volume control),
- να επιλέγει το είδος της οθόνης (εναλλαγή μεταξύ πλήρους ή προεπιλεγμένη οθόνης),
- να μεταπηδά μέσα από scrollbar σε όποιο σημείο του δελτίου ειδήσεων θέλει μή λαμβάνωντας υπόψιν την θεματική ενότητα ή το σημείο εκκίνησης με βάση το αποτέλεσμα αναζήτησης.

6.5 Video Streaming σε HTTP Server

Υπάρχουν δύο τρόποι για να κάνουμε προβάλουμε βίντεο στο διαδίκτυο. Ο πρώτος είναι με την προοδευτική λήψη και προβολή του βίντεο στο διαδίκτυο (progressive download method) ή χρησιμοποιώντας κάποιον stream server.

Η πρώτη μέθοδος είναι κοινή, δεν χρειάζεται κάποια άδεια και λειτουργεί σε κάθε είδος εξυπηρετητή ιστού (Web Servers). Παρόλα αυτά αυτή η μέθοδος δεν επίτρέπει την αναζήτηση σε σημεία του βίντεο που δεν έχουν ακόμα ληφθεί. Επομένως αποτελεί μία σίγουρη μεν περιοριστική όμως λύση.

Από την άλλη οι Stream Servers (όπως ο FMS, RED5 ή ο Wowza) προσφέρουν μεν αυτές τις δυνατότητες αλλά μειονεκτούν στους τύπους αρχείων που μπορούν να δεχτούν αλλά και έχουν και κάποια άλλα μειονεκτήματα (stream server account, ειδικό streaming software) .

Σαν καλύτερη μέθοδος προτείνεται μία ενδιάμεση λύση με χρήση ενός http Server και κάποια αρχεία που επεξεργάζονται στον εξυπηρετητή (Server side scripts). Το http straming δουλέυει μέσω του παρακάτω απλού μηχανισμού.

Κάθε φορά που ζητείται μία λειτουργία χρονικής αναζήτησης μέσα στο βίντεο ο videoplayer κάνει μία αίτηση στον εξυπηρετητή μέσω ενός αρχείου (streamscript) που περλαβάνει ένα σύνολο από GET μεταβλητές.

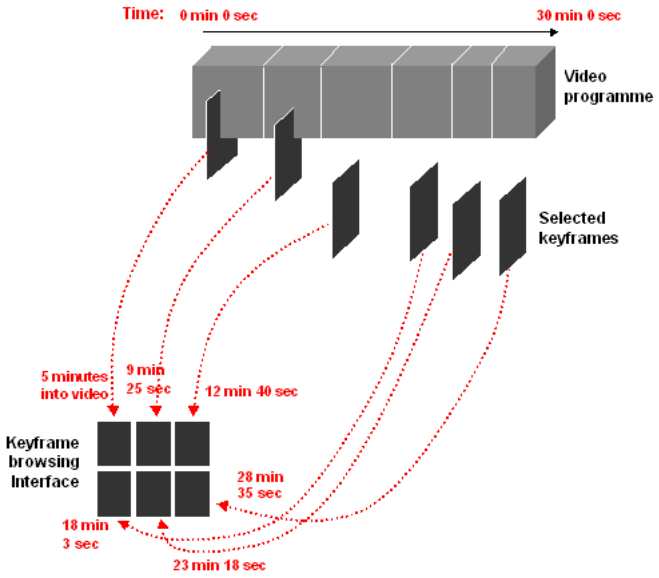
μία από αυτές τις μεταβλητές είναι το file που περιλαμβάνει το αρχείο που θέλουμε να προβληθεί αλλά και η μεταβλητή start που περιλαμβάνει το σημείο που θέλουμε να ξεκινήσει το βίντεο. μόλις ξεκινήσει το βίντεο από το σημείο start που ορίσαμε μπορεί ο χρήστης να μεταπηδήσει σε όποιο σημείο επιθυμεί χωρίς να χρειάζεται να περιμένει μέχρι να φορτωθεί όλο το βίντεο.

Η δημιουργία του streaming περιλαμβάνει 3 στοιχεία:

1. Εναν flash player. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε ο JW PLAYER ver. 4.1 (http://www.jeroenwijering.com) μιας και ο συγκεκριμένος player διαθέτει αρκετά καλή streaming συμπεριφορά. Προσθέσαμε τον player στον κώδικα μας μέσω της κατασκευής ενός SWFobject όπου αποτελέι μία εύκολη μέθοδο για την υλοποίηση flash αρχείων, με βάση την οποία μέθοδο χρησιμοποιείται ένα αρχείο javascript. Ετσι μπορούμε να πούμε στον player οτι θα κανουμε http streaming προσθέτοντας μια μεταβλητή flash (flashvar streamer) όπου θέτει την τοποθεσία που θα βρεθεί το stramscript. μπορούμε παρακάτω να δούμε ένα παράδειγμα υλοποίησης του player με χρήση της μεταβλητής streamer:

```
<p id='example' class="media">Here the video will be shown.</p>

<script type='text/javascript'>

var s1 = new SWFObject('/upload/player.swf','ply','420','250','9','#ffffff');
s1.addParam('allowfullscreen','true');
s1.addParam('flashvars','file=video.flv&streamer=/embed/xmoov.php');

s1.write('example');

</script>
```

2. ένα αρχείο που εκτελείται στον εξυπηρετητή και χρησιμοποιείται για το streaming. Στην περίπτωση μας χρησιμοποιήσαμε το XMOOV-PHP script (http://xmoov.com/xmoov-php/source/). Το xmoov.php είναι ένα pseudo-streaming script που επιχειρεί να „ξεκλειδώσει“ τα πλεονεκτήματα του video streaming ενώ το μόνο που χρειάζεται είναι ένας εξυπηρετητής ιστού που μπορεί να τρέχει προγράμματα php εκδόσεως 4 και πάνω. Υπάρχουν κάποιοι παράμετροι σε αυτό το αρχείο που πρέπει να ρυθμιστούν που θα περιγράψουμε στο παράρτημα Α.



Σχήμα 6.5: keyframes

3. μία βάση με βίντεο που είναι τύπου flash (.flv) και να περιέχουν keyframes (σχήμα 6.5). Τα keyframes είναι μεταδεδομένα που χρησιμοποιούνται για την αναζήτηση σημειών μέσα στο βίντεο. Λόγω του ότι δεν μπορεί να γίνει χρονική αναζήτηση μέσα σε ένα βίντεο τότε εξισώνουμε κάποια frames με προσεγγιστικές χρονικές στιγμές και κατα την διάρκεια της αναζήτησης ουσιαστικά βρίσκουμε το πλησιέστερο frame (το οποίο ονομάζουμε keyframe). Για την εισαγωγή keyframes μπορούμε να χρησιμοποιήσουμε διάφορα εργαλεία (flvtool, flvmdi, yamdi) ενώ για την συγκεκριμένη εργασία χρησιμοποιήθηκε το yamdi (http://yamdi.sourceforge.net/).

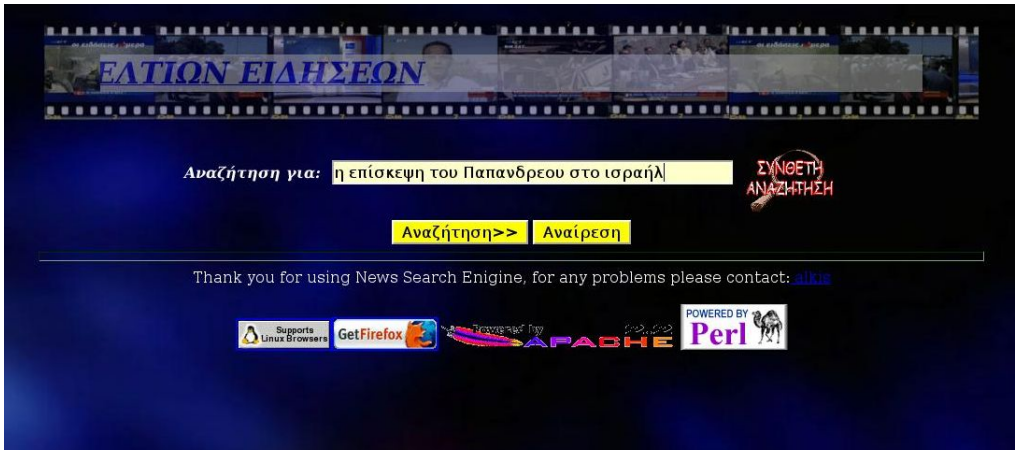
```
<fileset>
<flv name="/srv/www/htdocs/videos/keyframes/060627NETED2100.flv">
<hasKeyframes>true</hasKeyframes>
<hasVideo>true</hasVideo>
.
.
.
.
<lastkeyframelocation>85593946</lastkeyframelocation>
<keyframes>
<times>
<value id="0">0.00</value> //keyframes on xxxx sec
<value id="1">0.48</value>
<value id="2">0.96</value>
<value id="3">1.44</value>
.
.
.
.
<value id="2954">1390.48</value>
<value id="2955">1390.96</value>
</times>
<filepositions>
<value id="0">53839</value> //keyframes on xxxxx Bytes
<value id="1">124956</value>
<value id="2">150651</value>
<value id="3">175381</value>
.
.
.
.
<value id="2954">85563666</value>
<value id="2955">85593946</value>
</filepositions>
</keyframes>
<duration>1391.26</duration>
</flv>
</fileset>
```

Σχήμα 6.6: παράδειγμα αρχείου που XML αρχείου που παράγεται κατά την εισαγωγή keyframes

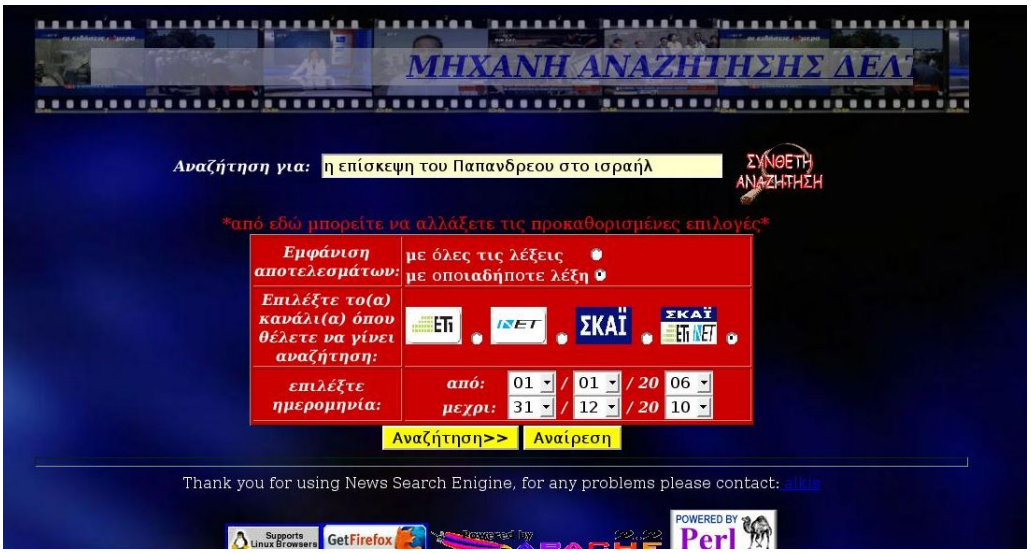
6.6 ΑΠΟΤΕΛΕΣΜΑΤΑ

Η εφαρμογή έχει τοποθετηθεί και σε server στο εργαστήριο τηλεπιοικωνιών του Πολυτεχνείου Κρήτης. Για να μπορεί κάποιος να τρέξει την εφαρμογή σωστά θα πρέπει να έχει εγκατεστημένο στον υπολογιστή του το plug in του flash player (http://www.adobe.com/products/flashplayer/).

Αφού αναλύσαμε την λειτουργία του player μπορούμε να δούμε έναν πλήρες παράδειγμα εισαγωγής ενός αιτήματος στο σύστημα και να δούμε τα αποτελέσματα.



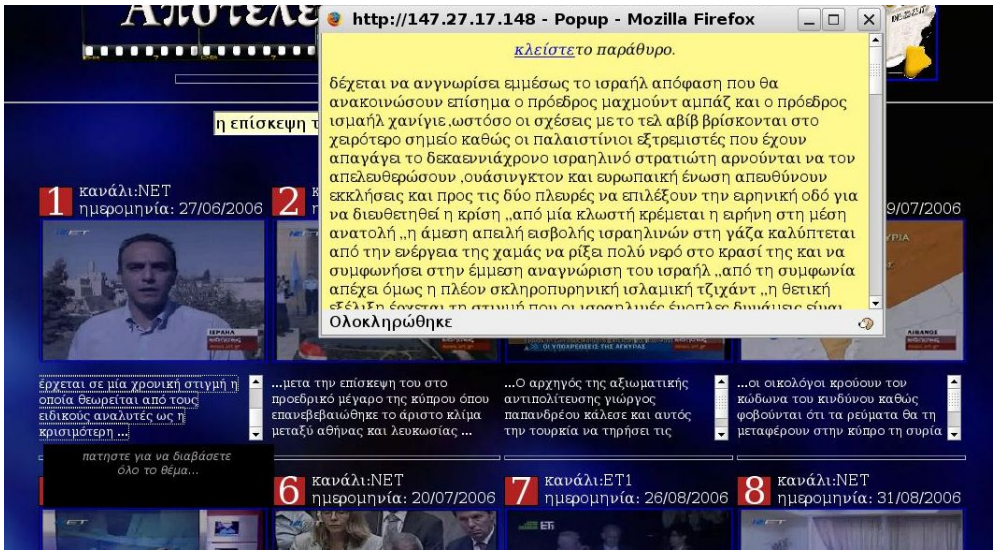
Σχήμα 6.7: εισαγωγή αιτήματος από τον χρήστη (‘η επίσκεψη του παπανδρέου στο ισραήλ’)



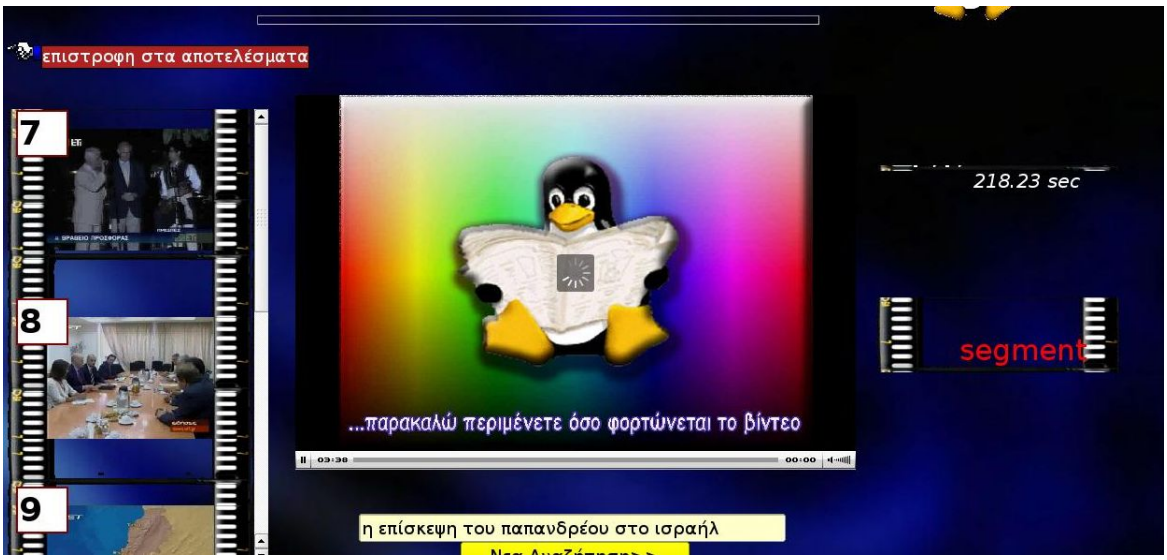
Σχήμα 6.8: ο χρήστης αλλάζει τα κριτήρια αναζήτησης με βάση τις προτιμήσεις του

Σχήμα 6.9: πατώντας Αναζήτηση εμφανίζονται στον χρήστη τα πρώτα 8 αποτελέσαμτα

Σχήμα 6.10: ο χρήστης διατρέχει τις σελίδες των αποτελεσμάτων πατώντας στο δεξί ή αριστερό χέρι



Σχήμα 6.11: πατώντας πάνω σε μία πρόταση ο χρήστης μπορεί να διαβάσει όλο την θεματική ενότητα που ανήκει η πρόταση



Σχήμα 6.12: πατώντας πάνω στην εικόνα ο χρήστης πηγαίνει στην σελίδα προβολής αντίστοιχου δελτίου ειδήσεων

Σχήμα 6.13: με το scrollbar αριστερά του βίντεο ο χρήστης μπορεί να διατρέξει τα αποτελέσματα βλέποντας και την αντίστοιχη σημαντική πρόταση

Σχήμα 6.14: πατώντας δεξιά το segment ο χρήστης μπορεί να παρακολουθήσει το δελιο από την αρχή της θεματικής ενότητας στην οποία ανήκει το αίτημα του

Chapter 7

ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

7.1 ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΤΕΜΑΧΙΣΜΟΥ

Αφού υλοποιήσαμε τον τεμαχισμό των κειμένων των δελτίων ειδήσεων και τον τεμαχισμό των αρχείων ήχου των δελτίο ειδήσεων, επόμενο βήμα ήταν να συνδυάσουμε αυτές τις δύο τεχνικές προκειμένου να βελτιώσουμε τα αποτελέσματα.

Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήσαμε το 20% των δεδομένων της βάσης μας . Συγκεκριμένα τα δεδομένα αυτά περιέχουν περίπου 10000 συνολικά όρους και διαρκεί περίπου 7000 sec.

Στο σχήμα 7.2 μπορούμε να δούμε το διάγραμμα που προκύπτει από ένα από τα δελτία ειδήσεων τα οποία χρησιμοποιήσαμε για την αξιολόγηση με αναγωγή στον χρόνο και όχι στις λέξεις όπως κάναμε προηγουμενως (μας βοηθάει να δούμε ακριβέστερα τα σημεία αλλαγής θέματος). Οι κάθετες γραμμές υποδυκνείουν τα σημεία ώπου ένας χρήστης θα χώριζε μόνος του θεματικά το δελτίο ειδήσεων. Παρατηρούμε από το παραπάνω διάγραμμα οτι καταφέραμε να προσεγγίσουμε τις περισσότερες θεματικές ενότητες και αυτές που δεν εντόπισε καθόλου οφείλεται στο γεγονός οτι ήταν πολύ μικρά θέματα (διάρκειας περίπου 20 δευτερολέπτων) οπότε δεν ήταν δυνατόν να εντοπιστούν αφού "χάνονταν" μέσα στα μπλοκ.

Γενικότερα θεωρούμε σαν ακρίβεια (presicion) τον αριθμό των σωστών συνόρων που δημιουργήθηκαν δια τον συνολικό αριθμό συνόρων και έφτασε

————	presicion	recall	F-measure
test set	85%	89%	86%

Πίνακας 7.2: απόδοση συστήματος τεμαχισμού

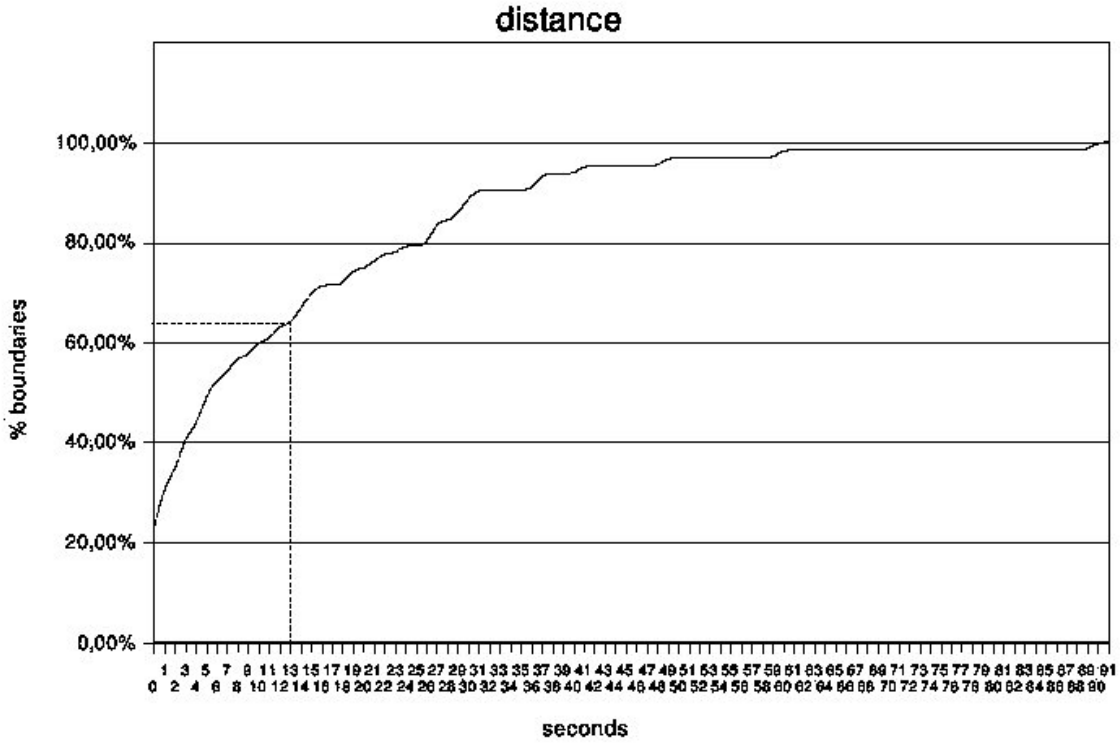
λάθη	αριθμός λαθών(%)
όρια που παραλήφθηκαν	11
όρια που προστέθηκαν λανθασμένα	15
όρια εκτός >12 sec	35

Πίνακας 7.4: αποτελέσματα αξιολόγησης

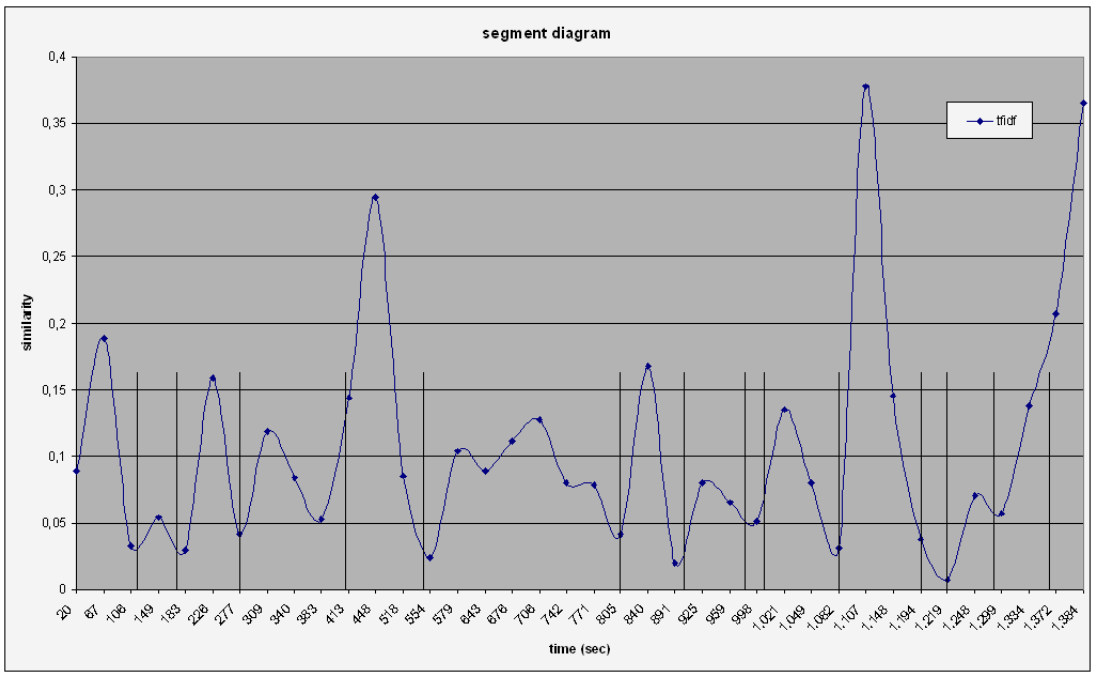
περίπου το 85 % .Σε παρόμοια απόδοση (89%) έφτασε και το recall. Τέλος για την μέτρηση της απόδοσης χρησιμοποιήσαμε το F-measure (ενότητα 2.5) όπου έφτασε το 86%.

Σαν λάθη στην διαδικασία τεμαχισμού (πίνακας 7.4) θεωρούμε είτε την λανθασμένη εισαγωγή ορίων σε μία θεματική ενότητα είτε την παράλειψη εισαγωγής ορίων. Τέλος σαν τρίτη κατηγορία λαθών θεωρούμε και την εύρεση μεν του ορίου αλλά σε απόσταση μεγαλύτερη από δώδεκα δευτερολέπτα από την "πραγματική" θέση του ορίου. Η επιλογή του ορίου των 12 sec δεν έγινε αυθαίρεται (σχημα 7.1) αλλά είναι η μέση απόσταση (μ) που παρατηρείται. Σε αυτό το σημείο ο αλγόριθμος υστερεί μιας και τα μπλοκ μας δεν είναι προτάσεις αλλά αλληλουχία λέξεων και έτσι είναι πιο δύσκολο να γίνει ξεκαθαρος διαχωρισμός μεταξύ των θεματικών ενοτήτων.

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(X_i)$$



Σχήμα 7.1: απόσταση από πραγματική θέση



Σχήμα 7.2: διάγραμμα δελτίου ειδήσεων μαζί με τα 'πραγματικά' σημεία αλλαγής θεματικής ενότητας

με την χρήση των audio segment αυτό που καταφέραμε ήταν όχι να βρούμε θεματικές ενότητες που δεν κατορθώσαμε να βρούμε μέσω του TextTiling αλλά να βελτιώσουμε τα σημεία αλλαγής των θεματικών ενοτήτων, δηλαδή να μειώσουμε τη μέση απόσταση από την "πραγματική" αρχή της θεματικής ενότητας.

Συμπεράινουμε οτι με την χρήση των audio segments η απόσταση μειώνεται σε ποσοστό 18 % κατά μέσο όρο ,και φτάνει στα 10 sec από 12 sec που ήταν ,ποσοστό το οποίο αν και δεν είναι μεγάλο αποτελέι βελτίωση. Βέβαια σε αρκετές περιπτώσεις δεν βελτιώνονται τόσο ώστε να έρθουν σε απόσταση λιγότερο από 12 δευτερόλεπτα από τα πραγματικά όρια. Οστώσο κάθε νέα θεμτική ενότητα ξεκινάει από την αρχή μίας νέας πρότασης σύμφωνα με το audio segmentation.

πραγματική αρχή θεματικής ενότητας	0	111	177	277	413	549	804	901	971	1006	1085	1194	1219	1284	1368
αρχή ενότητας με βάση το TextTiling	0	106	183	277	383	554	805	891	998	—-	1082	—-	1219	1299	—-
αρχή ενόιτητας με συνδιασμό των δύο	0	108	179	279	382	542	810	891	986	——	1085	—-	1219	1297	—-

Πίνακας 7.5: σημεία αλλαγής θεματικής ενότητας (σε δευτερόλεπτα)

ερώτηση	1	2	3	4	5
Η είσοδος των αιτημάτων ήταν εύκολη·					
Ξέρατε κάθε στιγμή τι να κάνετε για να συνεχίσετε·					
Σας άρεσε το γραφικό περιβάλλον·					
Τα αποτελέσματα ήταν σχετικά με το αίτημα σας·					
Οι θεματικές ενότητες σας βοήθησαν να βρείτε αυτό που ψάχνατε·					

Πίνακας 7.6: φόρμα αξιολόγησης συστήματος

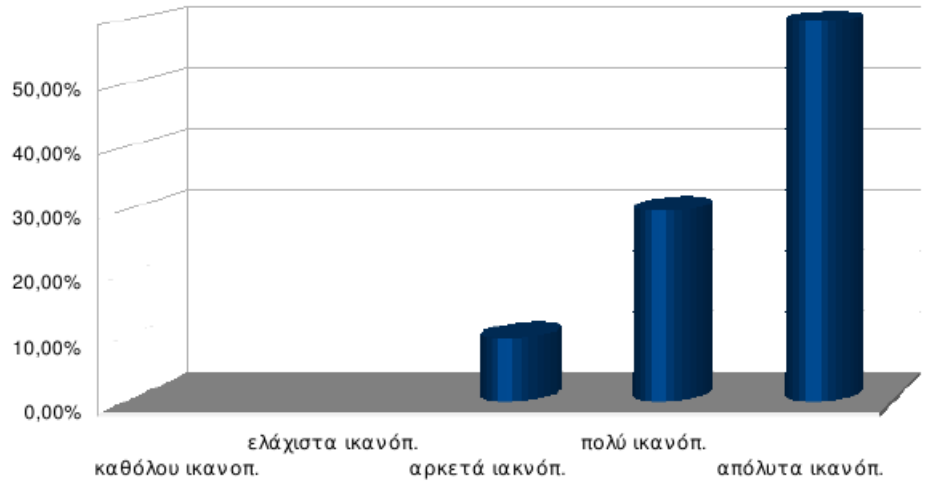
7.2 ΑΞΙΟΛΟΓΗΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΧΡΗΣΤΗ

Το τελευταίο πράγμα που μας μένει για να ολοκληρωθεί η εργασία αυτή, είναι η αξιολόγηση του συστήματος. Η αξιολόγηση ενός συστήματος έχει μεγάλη σημασία, αφού μας δίνει την άποψη τρίτων ατόμων για την εικόνα και την απόδοση του συστήματος που θα βοηθήσουν στην περαιτέρω βελτίωση του.

μέσω της αξιολόγησης μπορούμε να έχουμε μια αρκετά καλή εικόνα για το σύστημα, παρόλα αυτά όμως ο καλύτερος κριτής ενός συστήματος, παραμένει ο ίδιος ο χρήστης. Αντικειμενικά κριτήρια όπως το πόσο ικανοποιημένος μένει ο χρήστης από το σύστημα, η αποτελεσματικότητα, η ευελιξία και η σταθερότητα του συστήματος μπορούν να αποτελέσουν πολύτιμα μετρικά για την αξιολόγηση ενός συστήματος.

Η διαδικασία της αξιολόγησης περιλάμβανε την χρησιμοποίηση της εφαρμογής από ένα σύνολο από δέκα χρήστες και κατόπιν η συμπλήρωση ενός ερωτηματολογίου που περιλάμβαναν ερωτήσεις σχετικά με την αλλληλεπίδραση τους με το σύστημα. Στον πίνακα 7.1 μπορούμε να δούμε την φόρμα που δώσαμε στους χρήστες να συμπληρώσουν μετά από την χρήση της εφαρμογής.

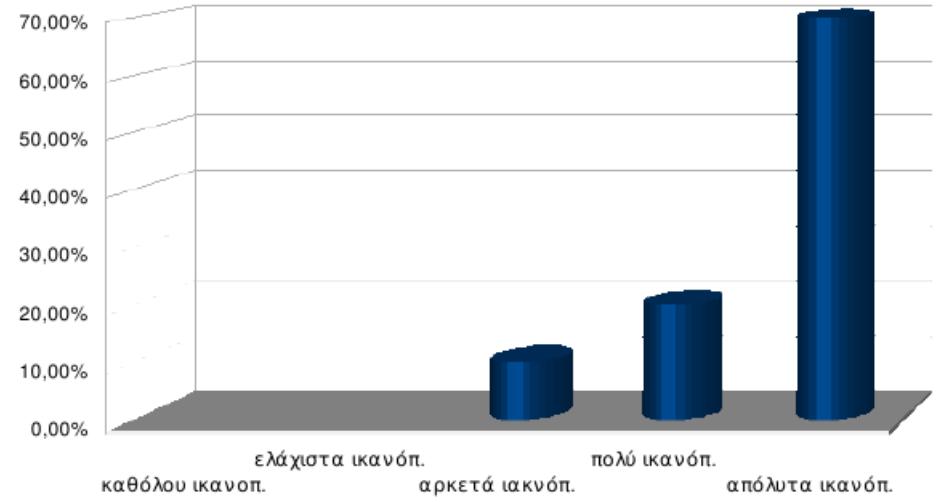
Στην συνέχεια θα αναλύσουμε τα υποκειμενικά αποτελέσματα που προέκυψαν από τα ερωτηματολόγια. Στο πρώτο διάγραμμα (σχήμα 7.3) φαίνονται τα αποτελέσματα στην ερώτηση 'Η είσοδος των αιτημάτων ήταν εύκολη·'.



Σχήμα 7.3: αποτελέσματα στην ερώτηση 'η είσοδος των αιτημάτων ήταν εύκολη'

Όπως φαίνεται ποσοστό 60% έμεινε απολύτως ικανοποιημένο, ένα 30% πολύ ικανοποιημένο ενώ ένα 10% έμεινε λίγο ικανοποιημένο (ως λίγο ικανοποιημένο λαμβανουμε την μέση απάντηση, δηλαδή ούτε θετικά αλλά ούτε αρνητικά)

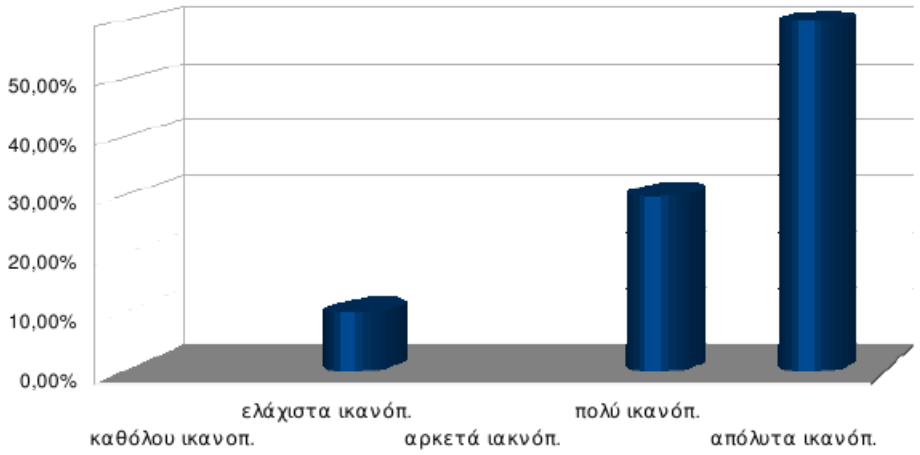
Το επόμενο διάγραμμα (σχήμα 7.4) μας δείχνει την τάση των ερωτηθέντων στην ερώτηση 'Ξέρατε κάθε στιγμή τι να κάνετε για να συνεχίσετε·'.



Σχήμα 7.4: αποτελέσματα στην ερώτηση 'Ξέρατε κάθε στιγμή τι να κάνετε για να συνεχίσετε·'

Όπως φαίνεται ποσοστό 70% έμεινε απολύτως ικανοποιημένο, ένα 20% πολύ ικανοποιημένο ενώ ένα 10% έμεινε λίγο ικανοποιημένο.

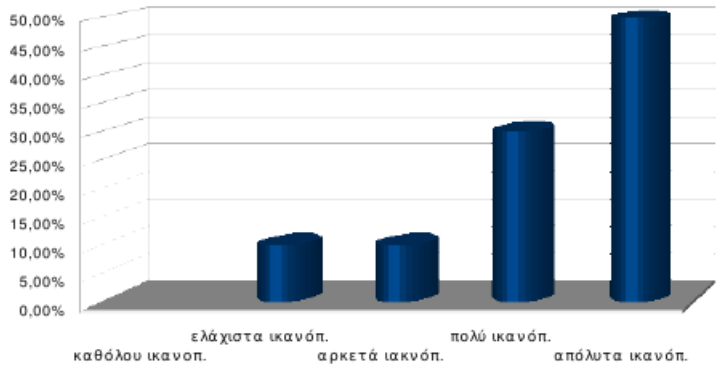
Το επόμενο διάγραμμα (σχήμα 7.5) μας δείχνει την τάση των ερωτηθέντων στην ερώτηση 'Σας άρεσε το γραφικό περιβάλλον·'.



Σχήμα 7.5: αποτελέσματα στην ερώτηση'Σας άρεσε το γραφικό περιβάλλον·'

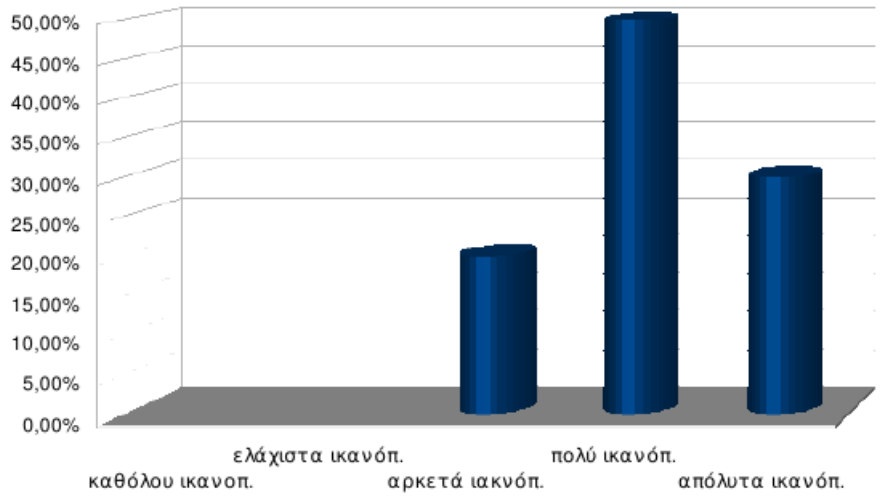
Όπως φαίνεται ποσοστό 60% έμεινε απολύτως ικανοποιημένο, ένα 30% αρκετά ικανοποιημένο ενώ ένα 10% έμεινε ελάχιστα ικανοποιημένο.

Στην ερώτηση 'Τα αποτελέσματα ήταν σχετικά με το αίτημα σας·' ένα 50% έμεινε απολύτως ικανοποιημένο, ένα 30% πολύ ικανοποιημένο ενώ ένα 10% έμεινε λίγο ικανοποιημένο και τέλος ένα 10% έμεινε ελάχιστα ικανοποιημένο. Σε αυτή την ερώτηση βλέπουμε μία μεγαλύτερη διασπορά στις απαντήσεις πράγμα που οφείλεται κυρίως στα υποκειμενικά κριτήρια αναζήτησης του καθενός. Τα παραπάνω αποτελέσματα φαίνονται και στο διάγραμμα 7.6.



Σχήμα 7.6: αποτελέσματα στην ερώτηση 'Τα αποτελέσματα ήταν σχετικά με το αίτημα σας·'

Τέλος στην ερώτηση 'Οι θεματικές ενότητες σας βοήθησαν να βρείτε αυτό που ψάχνατε·' ένα 30% των ερωτηθέντων έμεινε απολύτως ικανοποιημένο, ένα 50% πολύ ικανοποιημένο, και ένα 20% λίγο ικανοποιημένο όπως βλέπουμε και από το διάγραμμα 7.7.



Σχήμα 7.7: Αποτελέσματα στην ερώτηση 'Οι θεματικές ενότητες σας βοήθησαν να βρείτε αυτό που ψάχνατε·'

Το μειωμένο ποσοστό αυτών που δεν έμειναν απολύτως ικανοποιημένοι σε σχέση με τις υπόλοιπες ερωτήσεις οφείλεται κατά την άποψη μου στην απόσταση της πραγματικής αρχής της θεματικής ενότητας από την προσεγιστική. Έτσι ένας χρήστης χωρίς σχετική εμπειρία μπορεί να νομίζει οτι βρίσκεται σε λάθος ενοτητα από αυτήν που ψάχνει εάν ενδεχομένως δεν διαβάσει όλο το κείμενο.

7.3 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Το τελευταίο κομμάτι που μένει για να ολοκληρώσουμε την εργασία αυτή είναι να δούμε την μελλοντική δουλειά που μπορεί να γίνει για να βελτιωθεί το σύστημα. Θα παρουσιάσουμε τα τμήματα που πιστεύουμε ότι χρειάζονται περαιτέρω βελτίωση ώστε το σύστημα να αποδίδει καλύτερα, αλλά και τι μπορεί να προστεθεί ώστε αυτό να γίνει ακόμα καλύτερο.

Αρχικά αυτό που μπορεί να γίνει είναι να ενωθεί το σύστημα αναζήτησης με τον αυτόματο αναγνωριστή και να αποτελέσουν ένα ενιαίο αυτόματο σύστημα αποφώνησης και αναζήτησης δελτίων ειδήσεων. Σε αυτή την περίπτωση θα μπορούν να προστίθενται αυτόματα νέα δελτία ειδήσεων τα οποία θα απομα-

γνητοφωνούνται και στην συνέχεια θα μπορούμε να εκτελούμε αναζήτηση σε αυτά.

Ακόμα πρέπει να γινουν βελτιώσεις στον τρόπο τεμαχισμού των δεδομένων ώστε να μειωθούν τα λάθη του και να βελτιωθεί η απόδοση του. Θα μπορούσαν να δοκιμαστούν και άλλοι τρόποι για το segmentation (περαν του audio και textual) όπως για παράδειγμα το visual segmentation το οποίο ίσως βελτίωνε τα αποτελέσματα.

Τέλος θα μπορούσαν να γίνουν αλλαγές ή βελτιώσεις στη λειτουργικότητα του συστήματος. μπορούν να προστεθούν περαιτέρω επιλογές προς τον χρήστη με σκοπό να έχει μεγαλύτερο έλεγχο της βάσης. Πιο συγκεκριμένα θα μπορούσαν να προστεθούν επιλογές παρουσίασης αποτελεσμάτων αλλά και να γίνονται εξειδικευμένες αναζητήσεις πάνω στα επιστρεφόμενα αποτελέσματα.

Παράρτημα Α΄

ΕΓΚΑΤΑΣΤΑΣΗ ΕΡΓΑΛΕΙΩΝ/ΔΙΑμΟΡΦΩΣΗ

Α.1 ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ xmoov-php

Το xmoov.php (http://xmoov.com/xmoov-php/source/) είναι ένα pseudo-streaming script που επιχειρεί να „ξεκλειδώσει“ τα πλεονεκτήματα του video streaming ενώ το μόνο που χρειάζεται είναι ένας εξυπηρετητής ιστού που μπορεί να τρέχει προγράμματα php εκδόσεως 4 και πάνω. Υπάρχουν κάποιοι παράμετροι σε αυτό το αρχείο που πρέπει να ρυθμιστούν μέσα από τις define() συναρτήσεις. Οι μεταβλητές που θα πρέπει να παραμετροποιήσουμε είναι οι παρακάτω:

XMOOV_PATH_ROOT: το μονοπάτι που βρίσκεται η ιστιοσελίδα στον εξυπηρετητή (πχ. /usr/local/apache2/htdocs/).

XMOOV_PATH_FILES: ο φάκελος που είναι αποθηκευμένα τα αρχεία βίντεο (πχ keyframes/videos/).

XMOOV_GET_FILE: η μεταβλητή αυτή χρησιμοποιείται για το βίντεο που θα παίξει. (αυτό θα πρέπει να είναι file με τον JW Player).

XMOOV_GET_POSITION: η μεταβλητή αυτή χρησιμοποιείται για το σημείο εκκίνησης του βίντεο (αυτό θα πρέπει να είναι start με τον JW Player).

Ακόμα υπάρχουν και κάποιες άλλες μεταβλητές που μπορούμε να παραμετροποιήσουμε μέσα στο xmoov.php με σκοπό να επιταχύνουμε τον χρόνο απόκρισης από τον player. Αυτές οι μεταβλητές είναι:

XMOOV_CONF_LIMIT_BANDWIDTH: η μεταβλητή αυτή ρυθμίζει το bandwidth και του δίνουμε την τιμή TRUE.

XMOOV_BW_PACKET_SIZE: η μεταβλητή αυτή ρυθμίζει το κομμάτι του βίντεο που στέλνουμε στον φυλλομετρητή και του δίνουμε την τιμή 100 (kbytes).

XMOOV_BW_PACKET_INTERVAL: η μεταβλητή αυτή ρυθμίζει τον χρόνο που πρέπει να περιμένουμε για να σταλεί το κάθε πακέτο και του δίνουμε την τιμή 1.(sec) Οι δύο τελευταίες μεταβλητές ορίζουν την ταχύτητα λήψης (στο παράδειγμα μας είναι 100kbytes/sec).

Α.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ Apache HTTP Server ver2.2

Σε αυτό το κεφάλαιο θα περιγράψουμε τις οδηγίες εγκατάστασης του Apache Server [10] καθώς και την διαμόρφωση του προκειμένου να δέχεται αρχεία Perl. Πρέπει να επισημάνουμε οτι η εγκατάσταση αναφέρεται σε υπολογιστή με λειτουργικό σύστημα linux.

Απαιτήσεις Συστήματος:

- Τουλάχιστον 50Mb χώρο στον σκληρό δίσκο
- Κάποιος ANSI-C Compiler (προτείνεται ο GNU gcc Compiler)
- Perl 5

Αφού ελέγξουμε οτι υπάρχουν τα παραπάνω στο σύστημα μας κατεβάζουμε το πακέτο μέσα από το τερματικό με την εξής εντολή:

$ lynx http://httpd.apache.org/download.cgi

Κατόπιν πρέπει να αποσυμπιέσουμε το αρχείο που κατεβάσαμε :

$ gzip -d httpd-NN.tar.gz
$ tar xvf httpd-NN.tar

όπου θα δημιουργηθεί ο φάκελος με το πρόγραμμα στον οποίο θα κάνουμε install. Κατόπιν μπορούμε να διαμοφρώσουμε τα αρχεία που θα εγκαταστήσουμε με τις παρακάτω εντολές (μπορούμε να κάνουμε και αργότερα επιλέον αλλαγές στο Configuration)

***$ CC="pgcc" CFLAGS="-O2" ***

./configure –prefix=/sw/pkg/apache \ –enable-rewrite=shared \ –enable-speling=shared

Κατόπιν είμαστε έτοιμοι να εγκαταστήσουμε την εφαρμογή μέσω των παρακάτω εντολών

$ make

$ make install

μπορούμε να αλλάξουμε τις ρυθμίσεις της εγκατεστημένης έκδοσης μέσω της εντολης:

$ vi PREFIX[1]***/conf/httpd.conf***

Για να ξεκινήσει ο Server θα πρέπει να εκτελέσουμε την παρακάτω εντολή :

$ PREFIX/bin/apachectl -k start

Αντίστοιχα για να κλείσουμε τον Server :

$ PREFIX/bin/apachectl -k stop

Τώρα αφού εγκατασησαμε τον Apache θα πρέπει να τον διαμορφώσουμε έτσι ώστε να δέχεται προγράμματα CGI.

μέσα στο config.h πρέπει να ρυθμισουμε το ΣκριπτΑλιας[2] το οποίο θα πρέπει να είναι κάπως έτσι:

ScriptAlias /cgi-bin/ /usr/local/apache2/cgi-bin/

Όλα τα Perl αρχεία θα πρέπει να τα τρέχουμε μέσα από τον παραπάνω φάκελο που δημιουργείται κατά την εγκατάσταση του Apache.

Επειτα θα πρέπει να ενημερώσουμε τον Apache πως θα αναγνωρίζει τα Perl αρχεία προκειμένου να τα εκτελεί με CGI. Αυτό γίνεται με την εισαγωγή της παρακάτω εντολής στο config.h:

AddHandler cgi-script .cgi .pl

[1]εξαρτάται από το σημείο που έχουμε επιλέξει να γίνει η εγκατάσταση (συνήθως είναι /usr/local/)

[2]ειναι η εντολη που λεει απο που θα τρεχουν τα προγραμματα cgi

Βιβλιογραφία

[1] G. Salton AND M. McGILL, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983

[2] R. R. Korfhage, Information Storage and Retrieval, John Wiley & Sons, Inc , Nw York, 1997

[3] Understanding Search Engines-Mathematical Modeling and Text Retrieval, Michael W. Berry and Murray Browne, University of Tennessee, siam, 1999

[4] B. Shneiderman, Designing the User Interface : Strategies of Effective Human-Computer Interaction, second ed., Addison-Wesley, Reading, MA,1992

[5] Απομανγητοφώνηση ακουστικών σημάτων Ελληνικών Τηλεοπτικών Εκπομπών, Τσέργουλας Ορφέας, Διπλωματική Εργασία, Πολυτεχνείο Κρήτης, 2006

[6] M.F. Porter, 1980, An algorithm for suffix stripping, Program, 14(3) pp 130−137

[7] Development of a Stemmer for the Greek Language, G. Ntais, Master thesis, University of Stockholm, 2006

[8] Kirsch, S.T., Document retrieval over networks wherein ranking and relevance scores are computed at the client for multiple database documents, United States Patent #5,659,732, 1997.

[9] Marti A. Hearst, TextTiling: A Quantitative Approach to Discourse Segmentation ,University of California, Berkeley, 1994

[10] Ralf S. Engelschall, Apache Desktop Reference, electronic version, 2001

[11] Shishir Gundavaran, CGI Progamming on the World Wide Web, O'REILLY, 1996

[12] Jeff Johnson, GUI BLOOPERS, Morgan Kaufmann (an Imprint of Elsevier),2000

[13] Αγγελος Γαλης, Programming in HTML,net Series, 1996

[14] http://www.mplayerhq.hu

[15] http://www.jeroenwijering.com

[16] http://julius.sourceforge.jp/en_index.php

[17] Scott Shaobing Chen, IBM T.J Watson Research Center, Speaker, environment and channel change detection and clustering via Baysian Information Criterion