ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ - ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧ. Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ & ΔΙΚΤΥΩΝ (ΤΜΗΥΤΔ)

Θέμα εργασίας:

Ανάκληση σχολίων από το διαδίκτυο και ανάλυση συναισθήματος με τη μέθοδο της μηχανικής μάθησης.

Μεταπτυχιακή εργασία
της
Φώτη Μαγδαληνής

Εκπονήθηκε υπό την επίβλεψη των:
Βάβαλη Εμμανουήλ
Μποζάνη Παναγιώτη
Κατσαρό Δημήτριο

Ευχαριστίες

Αισθάνομαι βαθιά την ανάγκη να ευχαριστήσω πρώτα από όλους τον κ. Παπαδάκη Νικόλαο συμβασιούχο 407 στο Τμήμα Μηχ. Η/Υ Τηλ. και Δικτύων για την επιστημονική υποστήριξη, τις πολύτιμες συμβουλές, το χρόνο που διέθεσε για τις εποικοδομητικές συζητήσεις καθώς και την ηθική στήριξη που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της μεταπτυχιακής μου εργασίας. Κατά τη διάρκεια της συνεργασίας μας αποκόμισα σημαντική εμπειρία και γνώσεις που μου παρέχουν ένα πολύτιμο εφόδιο και στήριγμα για το ξεκίνημα της επαγγελματικής μου καριέρας.

Θέλω επίσης να ευχαριστήσω και τους επιβλέποντες καθηγητές κ. Βάβαλη Εμμανουήλ, κ. Μποζάνη Παναγιώτη και κ. Κατσαρό Δημήτριο για τη άψογη συνεργασία και υποστήριξη τους σε όλα τα στάδια της εργασίας, για την απαραίτητη καθοδήγηση που μου παρείχαν, καθώς και για τις καθοριστικές συμβουλές τους σε όλα τα κρίσιμα ζητήματα.

Πίνακας Περιεχομένων

1 Γενικά

Στόχος της παρούσας εργασίας είναι η δημιουργία ενός συστήματος το οποίο θα εξάγει κριτικές ταινιών από το διαδίκτυο και θα προχωρά σε κατηγοριοποίησή τους ως θετικές ή αρνητικές. Το σύστημα αυτό προέκυψε από την συνένωση των εργαλείων Rome και LingPipe.

Για την κατανόηση του εργαλείου Rome ήταν απαραίτητη η μελέτη των δύο βασικών προτύπων για τη δημιουργία web feeds, το πρότυπο του RSS 2.0 και Atom 1.0. Πιο συγκεκριμένα μελετήθηκε η δομή τους και διαφορές τους.

Για την βαθύτερη κατανόηση του εργαλείου LingPipe κρίθηκε απαραίτητη η μελέτη του τομέα της ανάλυσης συναισθήματος. Η ανάλυση συναισθήματος αποτελεί τομέα της επεξεργασίας φυσικής γλώσσας (NLP) και στόχο έχει την κατηγοριοποίηση κειμένων με βάση την πολικότητά τους. Συνεπώς μελετήθηκαν οι δύο κύριοι οδοί έρευνας που έχουν αναπτυχθεί στα πλαίσια της ανάλυσης συναισθήματος:

- Η λεξιλογική προσέγγιση
- Η προσέγγιση μηχανικής μάθησης

Τέλος θα γίνει η παρουσίαση του συστήματος που αναπτύχθηκε, το οποίο χρησιμοποιεί το εργαλείο Rome. Σε αυτό παρέχοντας μία διεύθυνση ενός RSS ή Atom feed γίνεται αυτόματη εξαγωγή κριτικών ταινιών από το διαδίκτυο. Στη συνέχεια οι κριτικές αυτές διοχετεύονται στο εργαλείο του LingPipe το οποίο χαρακτηρίζει τις κριτικές αυτές ως θετικές ή αρνητικές χρησιμοποιώντας τη μέθοδο της μηχανικής μάθησης.

Με την ανάπτυξη του Web 2.0 υπάρχουν πολλά κείμενα στο διαδίκτυο τα οποία μέχρι σήμερα έπρεπε να εξαχθούν χειροκίνητα για να διατεθούν σε ένα σύστημα ανάλυσης συναισθήματος όπως το LingPipe. Με το σύστημα το οποίο αναπτύχθηκε στην εργασία αυτή η ανάκληση των κειμένων γίνεται χωρίς την παρέμβαση ανθρώπου, απλά παρέχοντας στο σύστημα τη διεύθυνση ενός RSS ή Atom feed. Το σύστημα εξάγει τα σχόλια και τα παρέχει στη συνέχεια στο υποσύστημα που είναι υπεύθυνο για την ανάλυση συναισθήματος.

2 Ανάλυση Συναισθήματος

Η ανάλυση συναισθήματος ή εξόρυξη γνώμης αναφέρεται σε μία ευρεία περιοχή της επεξεργασίας φυσικής γλώσσας, της υπολογιστικής γλωσσολογίας και της εξόρυξης κειμένου. Σε γενικές γραμμές έχει ως στόχο να καθορίσει την στάση του ομιλητή ή συγγραφέα σε σχέση με κάποιο θέμα. Η στάση μπορεί να είναι η αξιολόγηση ή η αποτίμησή του, η συναισθηματική του κατάσταση κατά τη σύνταξη του κειμένου ή ο συναισθηματικός αντίκτυπος που ο συντάκτης επιθυμεί να δημιουργήσει στον αναγνώστη.

Η βασική αποστολή της ανάλυσης συναισθήματος είναι η κατηγοριοποίηση της πολικότητας ενός συγκεκριμένου κειμένου, εάν η εκφραζόμενη γνώμη είναι θετική ή αρνητική. Πρόωρη εργασία στον τομέα αυτό έγινε από τους Turney[1] και Pang[2] οι οποίοι εφάρμοσαν διαφορετικές μεθόδους για την ανίχνευση πολικότητας σε αξιολογήσεις προϊόντων και κριτικές ταινιών αντίστοιχα.

Κατηγοριοποίηση της πολικότητας ενός εγγράφου μπορεί να γίνει και σε μία κλίμακα πολλών κατηγοριών και όχι μόνο χαρακτηρισμός του ως θετικό ή αρνητικό. Ο Pang[3] επέκτεινε το βασικό στόχο της κατηγοριοποίησης κριτικών ως θετικές ή αρνητικές στην κατάταξή τους σε μία κλίμακα τριών ή τεσσάρων αστέρων. Ο Snyder[4] έκανε μία ανάλυση σε βάθος αξιολογήσεων εστιατορίων προβλέποντας τη βαθμολογία για διάφορες πτυχές του συγκεκριμένου εστιατορίου, όπως το φαγητό ή η ατμόσφαιρα, σε μία κλίμακα πέντε αστέρων.

Τυπικά, στην ανάλυση συναισθήματος, υπάρχουν δύο κύριοι οδοί έρευνας

- οι λεξιλογικές προσεγγίσεις, που εστιάζουν στην οικοδόμηση επιτυχημένων λεξικών, και
- οι προσεγγίσεις μηχανικής μάθησης, οι οποίες πρωτίστως εστιάζουν στα διανύσματα χαρακτηριστικών γνωρισμάτων.

2.1 Σύντομη Ιστορική αναδρομή

Παρόλο που ο τομέας της ανάλυσης συναισθήματος και της εξόρυξη γνώμης έχει πρόσφατα απολαύσει ένα ξέσπασμα ερευνητικής δραστηριότητας, υπήρχε για αρκετό καιρό ένα σταθερό υποβόσκον ενδιαφέρον. Θα μπορούσαμε να θεωρήσουμε έρευνες πάνω στις απόψεις και τις πεποιθήσεις σαν προάγγελους αυτού του τομέα [5], [6]. Μεταγενέστερη δουλειά εστίασε κυριότερα στη ερμηνεία των μεταφορικών εκφράσεων, των αφηγήσεων, της οπτικής γωνίας και συναφών περιοχών [7], [8], [9], [10], [11], [12], [13], [14], [15].

Η χρονιά 2001 φαίνεται να σημαδεύει την αρχή της ευρύτατης διάδοσης της επίγνωσης των ερευνητικών προβλημάτων αλλά και ευκαιριών που γεννιούνται από την ανάλυση συναισθήματος και την εξόρυξη γνώμης[16], [17], [18], [19], [20], [21], [22], [23], [24],

[25], [1], [26], [27] και ακολούθησε η δημοσίευση κυριολεκτικά εκατοντάδων paper πάνω στο αντικείμενο.
Οι λόγοι πίσω από αυτήν την έκρηξη της ερευνητικής δραστηριότητας περιλαμβάνουν:

- την ανάπτυξη μεθόδων μηχανικής μάθησης στην επεξεργασία φυσικής γλώσσας και της ανάκλησης πληροφορίας.
- την διαθεσιμότητα πληθώρας κειμένων για την εκπαίδευση των αλγορίθμων μηχανικής μάθησης, χάρη στην άνθηση του διαδικτύου και ειδικότερα στην ανάπτυξη ιστοτόπων συγκέντρωσης κριτικών. Και φυσικά
- στην συνειδητοποίηση των συναρπαστικών διανοητικών προκλήσεων που προσφέρει αυτός ο τομέας. [28]

2.2 Λεξιλογικές προσεγγίσεις

Μία λεξιλογική προσέγγιση τυπικά χρησιμοποιεί ένα λεξικό με προχαρακτηρισμένες λέξεις. Κάθε λέξη που συναντάται στο κείμενο συγκρίνεται με το λεξικό, εάν είναι παρούσα στο λεξικό, η τιμή πολικότητάς της προστίθεται στη συνολική τιμή πολικότητας του κειμένου. Για παράδειγμα αν βρεθεί ένα ταίριασμα με τη λέξη «υπέροχο», η οποία είναι χαρακτηρισμένη στο λεξικό ως θετική, τότε η συνολική τιμή πολικότητας του κειμένου αυξάνεται. Εάν τελικά η συνολική τιμή πολικότητας του κειμένου είναι θετική τότε το κείμενο κατηγοριοποιείται ως θετικό, στην αντίθετη περίπτωση κατηγοριοποιείται ως αρνητικό.
Παρόλο που είναι πολύ απλή από τη φύση της αυτή η προσέγγιση, πολλές παραλλαγές αυτής της λεξικολογικής προσέγγισης έχει παρατηρηθεί ότι έχουν καλύτερη επίδοση από την τύχη. [1],[29],[30]

Επειδή η κατηγοριοποίηση μίας δήλωσης εξαρτάται από τη βαθμολογία την οποία λαμβάνει, έχει αφιερωθεί πολλή δουλειά για να ανακαλυφθεί ποιες λεξιλογικές πληροφορίες φέρνουν τα καλύτερα αποτελέσματα.

- Μία αρχική έρευνα έγινε από τους Χατζηβασίλογλου και Wiebe [31], οι οποίοι έδειξαν ότι η υποκειμενικότητα μίας πρότασης μπορεί να προσδιοριστεί με τη χρήση ενός λεξικού το οποίο περιέχει μόνο επίθετα στα οποία προστέθηκε χειρονακτικά μια ετικέτα ενδεικτική της πολικότητάς τους.

  Με μέθοδο αυτή επιτεύχθηκε ακρίβεια σε ποσοστό πάνω από 80% κατά την κατηγοριοποίηση ανεξάρτητων προτάσεων.

- Οι Kennedy και Inkpen[29] τροποποιώντας την παραπάνω απλή μέθοδο καταμέτρησης όρων, χρησιμοποίησαν το λεξικό επιθέτων αλλά έλαβαν υπόψη τους και τους «μετατοπιστές σθένους». Χρησιμοποιήθηκαν δύο διαφορετικοί τύποι μετατόπισης σθένους για την βελτίωση της αρχικής μεθόδου. Αρχικά συμπεριέλαβαν τις αρνήσεις, οι οποίες μπορούν να αντιστρέψουν τη γνώμη που

εκφράζει ένας όρος του κειμένου από θετική σε αρνητική και το αντίθετο. Ως τέτοιοι όροι μπορούν να χαρακτηριστούν οι λέξεις: δεν, καθόλου, ούτε κτλ. Ακόμη, χρησιμοποιήθηκαν και οι ενισχυτές. Ως ενισχυτές χαρακτηρίζονται εκείνοι οι όροι οι οποίοι μεταβάλουν την ένταση με την οποία μία λέξη λαμβάνεται υπόψη ως θετική ή αρνητική.

Σε αντίθεση με την απλή μέθοδο καταμέτρησης των όρων, η οποία εφαρμόστηκε σε ανεξάρτητες προτάσεις, η μέθοδος αυτή δοκιμάστηκε σε κριτικές ταινιών και απέδωσε σωστά αποτελέσματα κατηγοριοποίησης σε ποσοστό 62,7%.

- Ο Turney[1] στην μέθοδο που ανέπτυξε χρησιμοποίησε μία αναζήτηση σε μία διαδικτυακή μηχανή αναζήτησης για να καθορίσει την πολικότητα των όρων που θα συμπεριλάμβανε στο λεξικό του.

  Το πρώτο βήμα του αλγορίθμου που ανέπτυξε είναι να εξάγει από το κείμενο προς κατηγοριοποίηση τις προτάσεις εκείνες οι οποίες περιλαμβάνουν τουλάχιστον ένα επίθετο ή επίρρημα. Παρόλο που ένα απομονωμένο επίθετο μπορεί να δηλώνει υποκειμενικότητα μπορεί να μην υπάρχουν επαρκή συμφραζόμενα από τα οποία να μπορεί να βγει ένα ασφαλές συμπέρασμα για τον σημασιολογικό προσανατολισμό του. Για παράδειγμα το επίθετο «απρόβλεπτο» μπορεί έχει είτε αρνητικό προσανατολισμό αν βρεθεί σε μία φράση η οποία περιλαμβάνεται σε μια κριτική για ένα αυτοκίνητο, π.χ. «απρόβλεπτη συμπεριφορά» , είτε θετική εάν βρεθεί σε μια φράση για την κριτική μιας ταινίας, για παράδειγμα μέσα στη φράση «απρόβλεπτη πλοκή». Για το λόγο αυτό ο αλγόριθμος εξάγει δύο διαδοχικές λέξεις, από τις οποίες η μία είναι ένα επίθετο ή επίρρημα και η δεύτερη είναι αντιπροσωπευτική των συμφραζομένων.

  Το δεύτερο βήμα του αλγορίθμου περιλαμβάνει την αποτίμηση του σημασιολογικού προσανατολισμού των φράσεων που εξήχθησαν στο προηγούμενο βήμα χρησιμοποιώντας τον PMI-IR αλγόριθμο. Ο αλγόριθμος αυτός χρησιμοποιεί την κοινή πληροφορία σαν ένα μέτρο της έντασης της εννοιολογικής συσχέτισης των δύο λέξεων. Η πληροφορία αυτή(PMI-Pointwise Mutual Information) μεταξύ των δύο λέξεων, λέξη1 και λέξη2 ορίζεται ως εξής:

$$\text{PMI}(\text{λέξη1}, \text{λέξη2}) = \log_2\left[\frac{p(\text{λέξη1 \& λέξη2})}{p(\text{λέξη1})p(\text{λέξη2})}\right] \quad (1)$$

  p(λέξη1 & λέξη2) είναι η πιθανότητα οι δύο λέξεις να εμφανίζονται μαζί. Εάν οι λέξεις αυτές είναι στατικά ανεξάρτητες, τότε η πιθανότητα να εμφανιστούν μαζί δίνεται από το γινόμενο p(λέξη1)p(λέξη2). Το αποτέλεσμα της διαίρεσης αυτών

των δύο πιθανοτήτων είναι ένα μέτρο της στατικής ανεξαρτησίας των δύο λέξεων. Ο λογάριθμος του αποτελέσματος αυτού μας δείχνει την ποσότητα της πληροφορίας που αποκτάμε για την παρουσία μίας λέξης όταν παρατηρούμε τη δεύτερη.

Ο σημασιολογικός προσανατολισμός(SO- Semantic Orientation) μίας φράσης υπολογίζεται ως εξής:

SO(φράση)=PMI(φράση, «τέλειο»)-PMI(φράση, «κακό») (2)

Ο σημασιολογικός προσανατολισμός είναι θετικός όταν η φράση είναι πιο στενά συνδεδεμένη με τη λέξη «τέλειο» και αρνητικός όταν η φράση είναι στενότερα συνδεδεμένη με τη λέξη «κακό»

Ο αλγόριθμος PMI-IR υπολογίζει την τιμή του PMI χρησιμοποιώντας ερωτήματα τα οποία εφαρμόζει σε μία μηχανή αναζήτησης και καταγράφοντας τον αριθμό των αποτελεσμάτων που επιστρέφει κάθε ερώτημα. Στη μέθοδο αυτή χρησιμοποιήθηκε η μηχανή αναζήτησης AltaVista επειδή διαθέτει τον τελεστή NEAR. Ο τελεστής NEAR περιορίζει την αναζήτηση στα έγγραφα τα οποία περιέχουν τις δύο λέξεις με μέγιστη απόσταση 10 λέξεων μεταξύ τους, ανεξαρτήτου σειράς εμφανίσεως. Συνδυάζοντας τις συναρτήσεις (1) και (2) προκύπτει η παρακάτω συνάρτηση

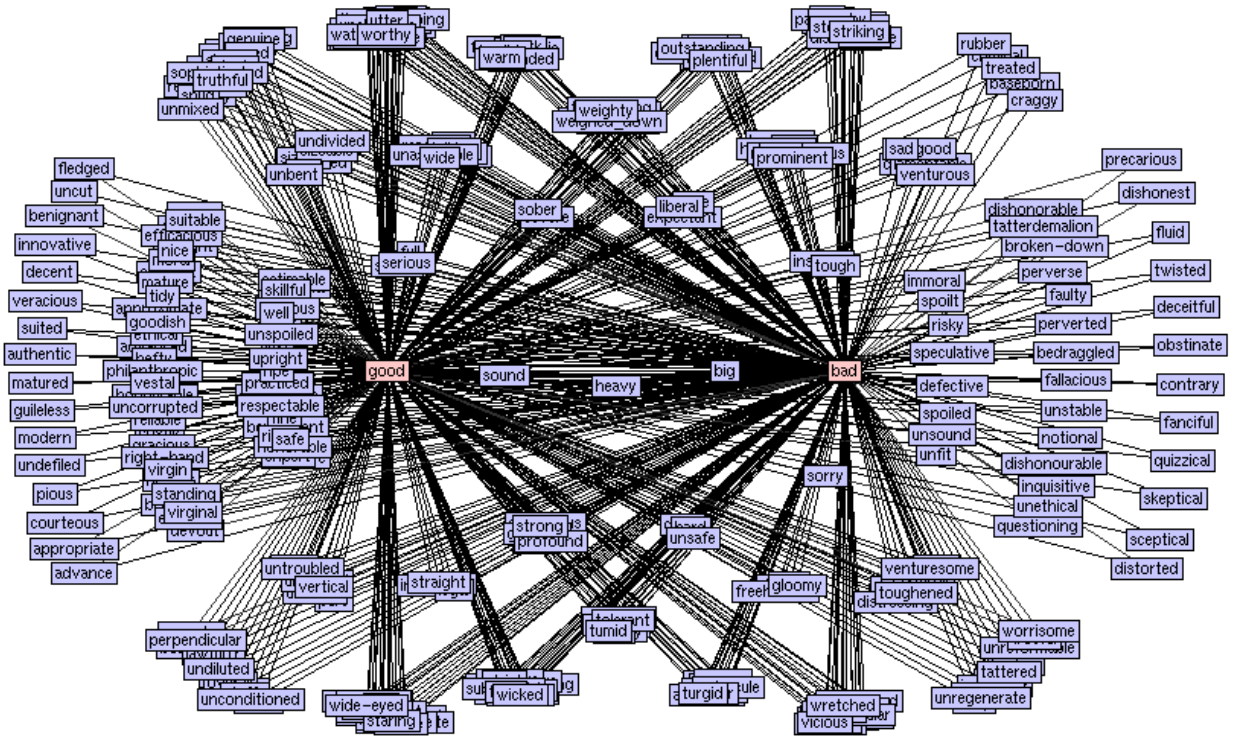
$$\text{SO(φράση)} = \log_2 \frac{hits(\varphi\rho\acute{\alpha}\sigma\eta\ NEAR\ \text{"τέλειο"})hits(\text{"κακό"})}{hits(\varphi\rho\acute{\alpha}\sigma\eta\ NEAR\ \text{κακό})hits(\text{"τέλειο"})}$$

Το τρίτο βήμα του αλγορίθμου περιλαμβάνει τον υπολογισμό του σημασιολογικού προσανατολισμού όλων τον φράσεων του κειμένου και την τελική κατηγοριοποίηση του κειμένου. Αν τελικά ο μέσο όρος του σημασιολογικού προσανατολισμού των φράσεων του κειμένου είναι θετικός αριθμός το κείμενο χαρακτηρίζεται ως μια θετική κριτική ενώ εάν είναι αρνητικός χαρακτηρίζεται ως αρνητική κριτική.

Με τη μέθοδο αυτή οι δημιουργοί της κατάφεραν ακρίβεια σε ποσοστό 65,8%

- Μια άλλη προσέγγιση, αυτή των Kamps et al.[30] και Andreevskaia et al.[32] , περιλαμβάνει τον προσδιορισμό της πολικότητας των λέξεων με τη χρήση της βάσης δεδομένων του WordNet. Στην προσέγγιση αυτή χρησιμοποιήθηκαν δύο λέξεις κλειδιά, οι λέξεις «καλό» και «κακό». Η λέξη στόχος, της οποίας η πολικότητα πρέπει να προσδιοριστεί συγκρίνεται με τις δύο λέξεις κλειδιά για να βρεθεί το ελάχιστο μονοπάτι μεταξύ της λέξης στόχου και των δύο λέξεων κλειδιών στην ιεραρχία του WordNet. Το ελάχιστο αυτό μονοπάτι μετατρέπεται σε ένα αυξανόμενο σκορ και το σκορ αυτό αποθηκεύεται με τη λέξη στόχο στο λεξικό. Με τον τρόπο αυτό το λεξικό είναι δυναμικό και μπορεί να προστεθεί οποιαδήποτε λέξει χρειαστεί με τρόπο αυτόματο.

Η μέθοδος αυτή έδωσε σωστά αποτελέσματα σε ποσοστό 64% [32]



Εικόνα 1: Μήκος μονοπατιών μεταξύ των λέξεων και των δύο επιθέτων «καλό» και «κακό»

- Μια εναλλακτική μέθοδος στη μέθοδο της μετρικής του WordNet προτάθηκε από τους Turney και Littman, στόχος της οποίας είναι ο προσδιορισμός του σημασιολογικού προσανατολισμού των λέξεων[33]. Αφαιρώντας την τιμή της έντασης με την οποία μία λέξη συνδέεται με ένα σετ αρνητικών λέξεων από την τιμή της έντασης που η ίδια λέξη συνδέεται με ένα σετ θετικών λέξεων οι δημιουργοί της μεθόδου αυτής κατάφεραν να πετύχουν ένα ποσοστό επιτυχίας της τάξης του 82% χρησιμοποιώντας δύο στατιστικές μετρικές σημασιολογικού προσανατολισμού.

Positiv		Negativ	
abide	absolve	abandon	abhor
ability	absorbent	abandonment	abject
able	absorption	abate	abnormal
abound	abundance	abdicate	abolish

Εικόνα 2: Παραδείγματα θετικών και αρνητικών λέξεων

2.3 Προσέγγιση μηχανικής μάθησης

Στις εφαρμογές μηχανικής μάθησης μία σειρά από διανύσματα χαρακτηριστικών επιλέγονται και μια συλλογή προχαρακτηρισμένων κειμένων εκπαίδευσης παρέχεται στον κατηγοριοποιητή για την εκπαίδευσή του ο οποίος μπορεί μετά να εφαρμοστεί σε μη χαρακτηρισμένα δεδομένα. Σε μία προσέγγιση μηχανικής μάθησης είναι πολύ κρίσιμη η επιλογή των χαρακτηριστικών για το ποσοστό επιτυχίας του κατηγοριοποιητή. Συνήθως, μια πληθώρα μονογραμμάτων (ανεξάρτητες λέξεις του κειμένου) ή ν-γραμμάτων (δύο ή περισσότερες λέξεις προερχόμενες από ένα έγγραφο στο οποίο ήταν διαδοχικά τοποθετημένες) επιλέγονται ως διανύσματα χαρακτηριστικών.

2.3.1 Μέθοδοι επιλογής χαρακτηριστικών [34]

Μερικές από τις ποιο ευρέως γνωστές μεθόδους επιλογής χαρακτηριστικών είναι οι παρακάτω:

- Document Frequency (Συχνότητα εγγράφου)
- Information Gain (Κέρδος Πληροφορίας)
- Mutual Information (Αμοιβαία πληροφορία)
- CHI

Όλες οι παραπάνω μέθοδοι υπολογίζουν μία βαθμολογία για κάθε ένα χαρακτηριστικό ξεχωριστά και στη συνέχεια επιλέγουν ένα προκαθορισμένο αριθμό χαρακτηριστικών.
Το αρχικό σύνολο χαρακτηριστικών αποτελείται από όλους τους όρους (λέξεις ή φράσεις) που εμφανίζονται στα κείμενα, οι οποίοι μπορεί να είναι δεκάδες χιλιάδες όροι ακόμη και για μεσαίου μεγέθους συλλογή κειμένων. Ο αριθμός αυτός είναι απαγορευτικά μεγάλος για πολλούς αλγορίθμους μάθησης. Μερικά νευρωνικά δίκτυα για παράδειγμα μπορούν να χειριστούν ένα τέτοιο νούμερο κόμβων εισόδου. Είναι ιδιαίτερα επιθυμητό να μειωθεί το αρχικό σύνολο των χαρακτηριστικών χωρίς όμως να θυσιαστεί ακρίβεια στην κατηγοριοποίηση. Είναι ακόμη επιθυμητό να επιτευχθεί ένας τέτοιος στόχος αυτόματα, να μην χρειάζεται χειρωνακτικός ορισμός ή κατασκευή των χαρακτηριστικών.

Η αυτοματοποίηση των μεθόδων επιλογής χαρακτηριστικών περιλαμβάνει την απομάκρυνση των όρων που δεν προσφέρουν πληροφορία σύμφωνα με τα στατιστικά στοιχεία του συνόλου των κειμένων και την κατασκευή νέων χαρακτηριστικών συνδυάζοντας χαρακτηριστικά χαμηλού επιπέδου (π.χ. όρους που αποτελούνται από μία λέξη). Οι Lewis και Ringuette[35] χρησιμοποίησαν ένα μέτρο κέρδους πληροφορίας για να μειώσουν δραστικά το λεξιλόγιο του κειμένου σε ένα μοντέλο Naïve Bayes και μία προσέγγιση δέντρου απόφασης για την δυαδική κατηγοριοποίηση. Οι Wiener et al[39, 38] χρησιμοποίησαν αμοιβαία πληροφορία και μία x^2 στατιστική για την επιλογή χαρακτηριστικών τα οποία τροφοδότησαν σε νευρωνικά δίκτυα. Οι Yang[40] και

Schutze et al[38, 39] χρησιμοποίησαν μία ανάλυση βασικών συστατικών για να βρουν ορθογώνιες στο διανυσματικό χώρο των εγγράφων. Οι Yang και Wilbur[41] χρησιμοποίησαν μία μέθοδο ομαδοποίησης εγγράφων για να υπολογίσουν πιθανολογικά την «ισχύ των όρων», την οποία χρησιμοποίησαν για την μείωση των μεταβλητών στην γραμμική παλινδρόμηση και την κατηγοριοποίηση πλησιέστερου γείτονα. Οι Moulinier et al[37] χρησιμοποίησαν έναν επαγωγικό αλγόριθμο μάθησης για την απόκτηση χαρακτηριστικών σε μία διαζευκτική μορφή για την κατηγοριοποίηση ειδήσεων. Ο Lang[36] χρησιμοποίησε μία αρχή ελάχιστου μήκους περιγραφής για να επιλέξει όρους για την κατηγοριοποίηση ειδήσεων.

Παρόλο που έχουν δοκιμαστεί πολλές τεχνικές για την επιλογή χαρακτηριστικών, εκτενείς αποτιμήσεις τους σε μεγάλα προβλήματα κατηγοριοποίησης κειμένου γίνονται σπάνια. Αυτό συμβαίνει, εν μέρει, γιατί πολλοί αλγόριθμοι μάθησης δεν κλιμακώνουν σωστά για υψηλών διαστάσεων σύνολα χαρακτηριστικών. Αυτό σημαίνει ότι όταν ένας κατηγοριοποιητής μπορεί να ελεγχθεί μόνο σε ένα μικρό υποσύνολο του αρχικού συνόλου, ο κατηγοριοποιητής δεν μπορεί να χρησιμοποιηθεί για την αποτίμηση της πλήρους εμβέλειας των δυνατοτήτων των μεθόδων επιλογής χαρακτηριστικών.

Οι Yang και Pedersen[34] σε μία έρευνα που έκαναν εστίασαν στην αποτίμηση και την σύγκριση των μεθόδων επιλογής χαρακτηριστικών ως προς την ικανότητά τους να μειώνουν το υψηλών διαστάσεων σύνολο χαρακτηριστικών σε προβλήματα κατηγοριοποίησης κειμένου. Χρησιμοποίησαν δύο κατηγοριοποιητές οι οποίοι είχαν ήδη δοκιμαστεί με χιλιάδες ή δεκάδες χιλιάδες κατηγορίες. Έψαξαν για απαντήσεις στις παρακάτω ερωτήσεις:

- Ποια είναι τα δυνατά σημεία και ποιες οι αδυναμίες των υπαρχουσών μεθόδων επιλογής χαρακτηριστικών που έχουν εφαρμοστεί στην κατηγοριοποίηση κειμένου
- Τι δυνατότητες βελτίωσης μπορεί να παρέχει η επιλογή χαρακτηριστικών στους κατηγοριοποιητές; Πόσο μπορεί να μειωθεί το λεξικό των κειμένων χωρίς να χαθεί χρήσιμη πληροφορία για την πρόβλεψη κατηγορίας;

2.3.1.1 **Document Frequency**

Η συχνότητα εγγράφου είναι ο αριθμός των εγγράφων στα οποία εμφανίζεται ο κάθε όρος. Υπολογίζεται η συχνότητα εγγράφων για κάθε όρο που εμφανίζεται στο σύνολο εκπαίδευσης και απομακρύνονται οι όροι εκείνοι των οποίων η συχνότητα είναι μικρότερη από ένα κάτω όριο το ποίο έχει προαποφασιστεί. Η βασική υπόθεση είναι ότι οι όροι οι οποίοι εμφανίζονται σπάνια είτε δεν προσφέρουν πληροφορία για την πρόβλεψη της κατηγορίας των κειμένων, είτε δεν έχουν μεγάλη επίδραση στη συνολική επίδοση. Και στις δύο περιπτώσεις η απομάκρυνση των σπάνιων όρων μειώνει τις διαστάσεις του χώρου των χαρακτηριστικών. Βελτίωση στην ακρίβεια της κατηγοριοποίησης είναι επίσης πιθανή εάν οι σπάνιοι όροι είναι τελικά θόρυβος.

Το κατώφλι συχνότητας εγγράφου είναι μία απλή τεχνική για τη μείωση του λεξιλογίου. Κλιμακώνει εύκολα για πολύ μεγάλο αριθμό εγγράφων, με υπολογιστική πολυπλοκότητα σχεδόν γραμμική σε σχέση με τον αριθμό των εγγράφων εκπαίδευσης. Παρόλα αυτά, συνήθως χρησιμοποιείται ως μία ειδική μέθοδος για την βελτίωση της αποτελεσματικότητας και όχι ως ένα κριτήριο για την επιλογή χαρακτηριστικών πρόβλεψης. Ακόμη η τεχνική DF τυπικά δεν χρησιμοποιείται για μεγάλης έκτασης απομάκρυνση όρων.

2.3.1.2 **Information Gain**

Το κέρδος πληροφορίας συχνά χρησιμοποιείται σαν ένα κριτήριο ωφελιμότητας όρου στο πεδίο της μηχανικής μάθησης [43], [42]. Μετράει τον αριθμό των bit πληροφορίας που αποκτάται για την πρόβλεψη κατηγορίας όταν είναι γνωστή η ύπαρξη ή η απουσία του όρου στο έγγραφο. Έστω $\{c_i\}_{i=1}^{m}$ το σύνολο των κατηγοριών. Το κέρδος πληροφορίας του όρου t ορίζεται ως εξής:

$$G(t) = -\sum_{i=1}^{m} P_r(c_i) \log P_r(c_i) + P_r(t) \sum_{i=1}^{m} P_r(c_i|t) \log P_r(c_i|t) + P_r(\bar{t}) \sum_{i=1}^{m} P_r(c_i|\bar{t}) \log P_r(c_i|\bar{t})$$

Ο ορισμός αυτός είναι γενικότερος σε σύγκριση με αυτόν που χρησιμοποιείται στο μοντέλο δυαδικής κατηγοριοποίησης [35], [37]. Στην σύγκριση μεταξύ των μεθόδων επιλογής χαρακτηριστικών χρησιμοποιήθηκε αυτός ο ορισμός γιατί τα προβλήματα κατηγοριοποίησης κειμένου έχουν συνήθως ένα σύνολο M κατηγοριών (όπου M μπορεί να είναι μέχρι και δεκάδες χιλιάδες κατηγορίες) και χρειάζεται ο υπολογισμός της ωφελιμότητας ενός όρου συνολικά λαμβάνοντας υπόψη όλες τις κατηγορίες.

Δεδομένου ενός συνόλου εγγράφων εκπαίδευσης, για κάθε διαφορετικό όρο υπολογίζεται το κέρδος πληροφορίας και απομακρύνονται από το σύνολο των χαρακτηριστικών οι όροι εκείνοι, οι οποίοι δεν συγκέντρωσαν κέρδος μεγαλύτερο του κατωφλίου που είχε προαποφασιστεί. Ο υπολογισμός περιλαμβάνει την αποτίμηση της δεσμευμένης πιθανότητας μιας κατηγορίας δοσμένου ενός όρου και τους υπολογισμούς της εντροπίας στον ορισμό. Η εκτίμηση της πιθανότητας έχει πολυπλοκότητα χρόνου O(N) και πολυπλοκότητα χώρου O(VN) όπου N είναι ο αριθμός των εγγράφων εκπαίδευσης και V το μέγεθος του λεξιλογίου. Οι υπολογισμοί της εντροπίας έχουν πολυπλοκότητα χρόνου O(Vm).

2.3.1.3 **Mutual Information**

Εάν κάποιος θεωρήσει τον πίνακα συμπτώσεων δύο εισόδων ενός όρου t και μίας κατηγορίας c, όπου A είναι ο αριθμός των φορών που το t και το c εμφανίζονται μαζί, B είναι οι φορές όπου το t εμφανίζεται χωρίς το c, C είναι οι φορές που το c εμφανίζεται

χωρίς το t, και N είναι ο συνολικός αριθμός των εγγράφων, τότε η αμοιβαία πληροφορία μεταξύ του t και του c ορίζεται ως:

$$I(t, c) = \log \frac{P_r(t \cap c)}{P_r(t) \times P_r(c)}$$

και υπολογίζεται χρησιμοποιώντας :

$$I(t, c) \approx \log \frac{A \times N}{(A + C) \times (A + B)}$$

Το I(t, c) ισούται με μηδέν εάν τα t και c είναι ανεξάρτητα μεταξύ τους. Για τον υπολογισμό της ωφελιμότητας ενός όρου σε μία γενική επιλογή χαρακτηριστικών, συνδυάζονται οι βαθμολογίες του όρου για κάθε κατηγορία με δύο διαφορετικούς τρόπους.

$$I_{avg}(t) = \sum_{i=1}^{m} P_r(c_i)\, I(t, c_i)$$

$$I_{max}(t) = max_{i=1}^{m}\{I(t, c_i)\}$$

Ο υπολογισμός της αμοιβαίας πληροφορίας έχει πολυπλοκότητα χρόνου O(Vm) όμοια με αυτήν του υπολογισμού του κέρδους πληροφορίας.
Μία αδυναμία της αμοιβαίας πληροφορίας είναι το γεγονός ότι η βαθμολογία επηρεάζεται σημαντικά από τις οριακές πιθανότητες των όρων, όπως φαίνεται και στον παρακάτω τύπο:

$$I(t, c) = \log P_r(t|c) - \log P_r(t)$$

Για όρους με ίδια τιμή δεσμευμένης πιθανότητας $P_r(t|c)$ σπάνιοι όροι θα έχουν υψηλότερη βαθμολογία από τους συχνούς όρους. Συνεπώς η βαθμολογία δεν είναι συγκρίσιμο μέγεθος μεταξύ όρων οι οποίοι έχουν μεγάλη απόκλιση στις συχνότητες εμφάνισής τους.

2.3.1.4 **CHI**

Η x^2 στατιστική υπολογίζει την έλλειψη ανεξαρτησίας μεταξύ των t και c και μπορεί να συγκριθεί με την x^2 κατανομή με ένα βαθμό ελευθερίας για να εκτιμηθεί η ακρότητα. Χρησιμοποιώντας τον πίνακα συμπτώσεων δύο εισόδων του όρου t και της κατηγορίας c, όπου A είναι ο αριθμός των φορών που το t και το c εμφανίζονται μαζί, B είναι οι φορές όπου το t εμφανίζεται χωρίς το c, C είναι οι φορές που το c εμφανίζεται χωρίς το t, και N είναι ο συνολικός αριθμός των εγγράφων, η βαθμολογία της ωφελιμότητας του κάθε όρου υπολογίζεται με τον παρακάτω τύπο.

$$x^2(t,c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

Η x^2 στατιστική παίρνει την τιμή μηδέν αν τα t και c είναι ανεξάρτητα μεταξύ τους. Υπολογίστηκε για κάθε κατηγορία η x^2 στατιστική μεταξύ κάθε ξεχωριστού όρου των κειμένων εκπαίδευσης και της κατηγορίας αυτής, και στη συνέχεια συνδυάστηκε η βαθμολογία δεδομένης κατηγορίας του κάθε όρου στις δύο επόμενες βαθμολογίες:

$$x^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) x^2(t, c_i)$$

$$x^2_{max}(t) = max^m_{i=1}\{x^2(t, c_i)\}$$

Ο υπολογισμός των CHI βαθμολογιών έχει τετραγωνική πολυπλοκότητα όμοια με τις τεχνικές MI και IG.

Μία σημαντική διαφορά μεταξύ των CHI και MI είναι ότι το x^2 είναι μία κανονικοποιημένη τιμή και για το λόγο αυτό οι x^2 τιμές είναι συγκρίσιμες μεταξύ όρων της ίδιας κατηγορίας. Παρόλα αυτά, η κανονικοποίηση σταματάει να λειτουργεί αν κάποιο κελί στον πίνακα συμπτώσεων έχει χαμηλή τιμή, που είναι η περίπτωση όρων που εμφανίζονται με πολύ μικρή συχνότητα. Παρόλα αυτά, η x^2 στατιστική είναι γνωστή ως μη αξιόπιστη για όρους μικρής συχνότητας.

2.3.2 Μέθοδοι μηχανικής μάθησης [44]

2.3.2.1 **Centroid classifier**

Η ιδέα στην οποία βασίστηκε η ανάπτυξη του αλγορίθμου του κεντροειδούς κατηγοριοποιητή είναι εξαιρετικά απλή. Αρχικά υπολογίζουμε το πρωτότυπο διάνυσμα ή το κεντροειδές διάνυσμα για κάθε κλάση εκπαίδευσης. Στη συνέχεια υπολογίζουμε την ομοιότητα μεταξύ ενός δοκιμαστικού εγγράφου d και όλων των κεντροειδών και τελικά, βασιζόμενοι σε αυτές τις ομοιότητες, αναθέτουμε το d στην κλάση εκείνη η οποία αντιστοιχεί στο πιο όμοιο κεντροειδές.
Στη φάση της εκπαίδευσης υπολογίζουμε k κεντροειδή, $\{C_1, C_2, C_3, \ldots, C_K\}$ για τις k κλάσεις χρησιμοποιώντας τον παρακάτω τύπο:

$$C_i = \frac{1}{|c_i|} \sum_{d \in C_i} d$$

Όπου $|z|$ δηλώνει τον αριθμό των στοιχείων του συνόλου z και το d δηλώνει το έγγραφο στην κλάση c_i .
Για κάθε δοκιμαστικό έγγραφο d υπολογίζουμε την ομοιότητά του με κάθε κεντροειδές c_i χρησιμοποιώντας το μέτρο του συνημίτονου ως εξής:

$$\mathrm{sim}(\mathrm{d}, \mathrm{c_i}) = \frac{\mathrm{d} \times \mathrm{c_i}}{\|\mathrm{d}\|_2 \, \|\mathrm{c_i}\|_2}$$

2.3.2.2 **k-Nearest Neighbor classifier**

Η κατηγοριοποίηση κ-πλησιέστερων γειτόνων είναι μία ευρέως γνωστή στατιστική προσέγγιση η οποία έχει μελετηθεί εντατικά στον τομέα της αναγνώρισης προτύπων τις τελευταίες τέσσερις δεκαετίες[49]. Ο kNN αλγόριθμος εφαρμόστηκε στην κατηγοριοποίηση κειμένου από τα πρώτα στάδια της έρευνας[50,51,52]. Ο kNN αλγόριθμος είναι αρκετά απλός: δεδομένου ενός δοκιμαστικού εγγράφου, το σύστημα βρίσκει τους κ πλησιέστερους γείτονες μεταξύ των εγγράφων εκπαίδευσης και χρησιμοποιεί τις κατηγορίες των κ γειτόνων για να αναθέσει βάρη στις υποψήφιες κατηγορίες. Ο βαθμός ομοιότητας του κάθε γειτονικού εγγράφου με το δοκιμαστικό έγγραφο χρησιμοποιείται ως το βάρος των κατηγοριών του γειτονικού εγγράφου. Εάν διάφοροι από τους κ πλησιέστερους γείτονες μοιράζονται την ίδια κατηγορία τότε τα βάρη των γειτόνων για αυτήν την κατηγορία προστίθενται και το βεβαρημένο άθροισμα που προκύπτει χρησιμοποιείται ως η βαθμολογία της πιθανότητας το δοκιμαστικό έγγραφο να ανήκει σε αυτήν την κατηγορία. Ταξινομώντας τις βαθμολογίες των υποψηφίων κατηγοριών δημιουργείται μία λίστα. Εφαρμόζοντας ένα κατώφλι βαθμολογίας στη λίστα αυτή εξασφαλίζεται μία δυαδική εκχώρηση κατηγορίας. Ο κανόνας απόφασης για τον κNN μπορεί να γραφτεί ως εξής:

$$y(\vec{x}, \mathrm{c_j}) = \sum_{\vec{di} \in kNN} sim(\vec{x}, \vec{d_\iota}) y(\vec{d_\iota}, c_j) - b_j$$

Όπου $y(\vec{d_\iota}, c_j) \in \{0,1\}$ είναι η κατηγοριοποίηση του εγγράφου $\vec{d_\iota}$ σε σχέση με την κατηγορία $\mathrm{c_j}$ (y=1 για ΝΑΙ, y=0 για ΟΧΙ) $sim(\vec{x}, \vec{d_\iota})$ είναι η ομοιότητα μεταξύ του δοκιμαστικού εγγράφου $\vec{x}$ και του εγγράφου εκπαίδευσης $\vec{d_\iota}$ και $\mathrm{b_j}$ είναι το κατώφλι της κάθε κατηγορίας για δυαδικές αποφάσεις. Για λόγους ευκολίας μπορεί να χρησιμοποιηθεί η τιμή του συνημίτονου των δύο διανυσμάτων ως βαθμολογία ομοιότητας μεταξύ των δύο εγγράφων, παρόλο που μπορούν να χρησιμοποιηθούν και άλλες μετρικές ομοιότητας.

2.3.2.3 **Naïve Bayes**

Ο Naïve Bayes αλγόριθμος είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος στην κατηγοριοποίηση κειμένου. Δεδομένου ενός πίνακα διανυσμάτων χαρακτηριστικών, ο αλγόριθμος υπολογίζει την μεταγενέστερη πιθανότητα ένα έγγραφο να ανήκει σε διαφορετικές κλάσεις και το αναθέτει στην κλάση με τη μεγαλύτερη μεταγενέστερη πιθανότητα. Υπάρχουν δύο συχνά χρησιμοποιούμενα μοντέλα, το πολυωνυμικό μοντέλο και το multi-variate Bernoulli μοντέλο.

Το πολυωνυμικό μοντέλο του Naïve Bayes αλγορίθμου υπολογίζει την πιθανότητα της λέξης w_t δοσμένης της κατηγορίας c_j σύμφωνα με τον παρακάτω τύπο:

$$p(w_t|c_j) = \frac{\sum_{i=1}^{N_j} n_{it}}{\sum_{s=1}^{W}\sum_{i=1}^{N_j} n_{is}}$$

Όπου n_{it} είναι ο αριθμός των φορών που η λέξη t εμφανίζεται στο έγγραφο i, το N_j αναφέρεται στον αριθμό των εγγράφων εκπαίδευσης στην κατηγορία c_j και W είναι το μέγεθος του λεξιλογίου.
Η μεταγενέστερη πιθανότητα μπορεί να υπολογιστεί ως ακολούθως:

$$p(c_j|d_i) = \frac{p(c_j)p(d_i|c_j)}{p(d_i)}$$

2.3.2.4 **Winnow Classifier**

Η winnow είναι μία online ευρέως γνωστή μέθοδος οδηγούμενη από σφάλματα. Δουλεύει ανανεώνοντας τα βάρη της σε μία σειρά από δοκιμές. Σε κάθε δοκιμή, αρχικά κάνει μία πρόβλεψη για ένα έγγραφο d και στη συνέχεια λαμβάνει μία ανάδραση, εάν έχει συμβεί κάποιο σφάλμα, ανανεώνει το διάνυσμα βαρών του χρησιμοποιώντας το έγγραφο d. Κατά τη διάρκεια της φάσης εκπαίδευσης, με μία συλλογή δεδομένων εκπαίδευσης, η διαδικασία αυτή επαναλαμβάνεται αρκετές φορές επαναχρησιμοποιώντας τα δεδομένα. Μέχρι σήμερα, υπάρχουν αρκετές εκδοχές του winnow, όπως ο θετικός winnow, ο ισορροπημένος winnow, ο μεγάλων περιθωρίων winnow. Εδώ θα μελετήσουμε τον ισορροπημένο winnow λόγω της πολύ καλής του απόδοσης.

Ο ισορροπημένος winnow αλγόριθμος διατηρεί δύο βάρη για κάθε χαρακτηριστικό w_{kt}^{+} και w_{kt}^{-} . Για ένα δεδομένο στιγμιότυπο (d_{k1}, d_{k2}, d_{k3},....,d_{kw}) ο αλγόριθμος θεωρεί ένα έγγραφο σχετικό αν και μόνο αν:

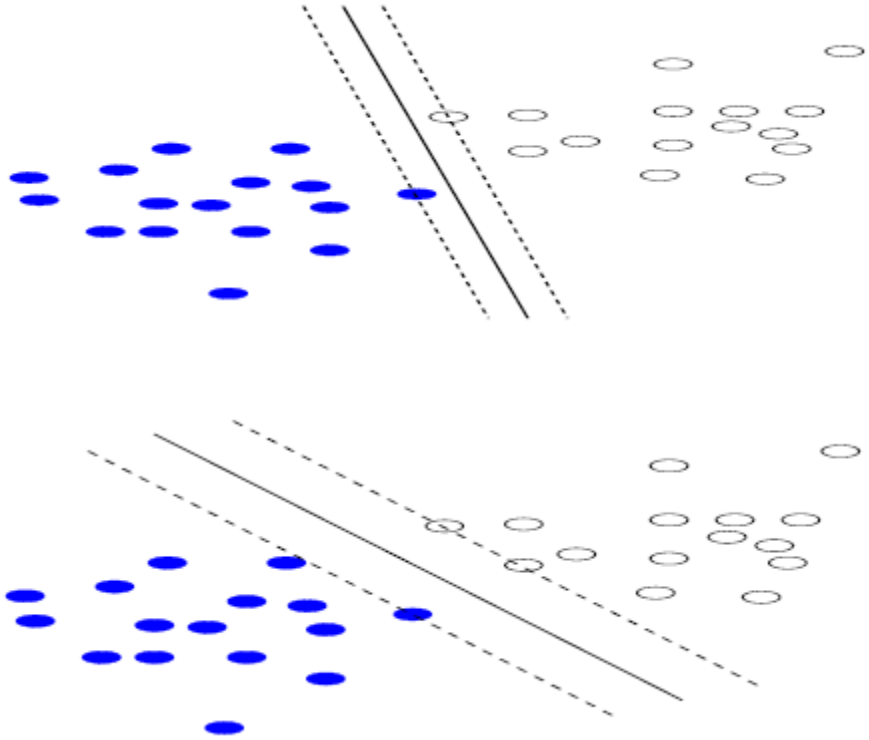
$$\sum_{t=1}^{w}(w_{kt}^{+} - w_{kt}^{-})d_{kt} \geq \tau$$

Όπου το τ είναι ένα δοσμένο κατώφλι και το κ είναι η ετικέτα της κλάσης.

Τα βάρη των ενεργών χαρακτηριστικών ενημερώνονται μόνο όταν συμβεί ένα λάθος. Στην φάση της προαγωγής, που ακολουθεί ένα λάθος σε ένα θετικό παράδειγμα, το θετικό μέρος του βάρους προάγεται $w_{kt}^{+} = w_{kt}^{+} \times \alpha$ (α>1) ενώ το αρνητικό μέρος του βάρους υποβιβάζεται $w_{kt}^{-} = w_{kt}^{-} \times \beta$ (0<β<1). Ο συντελεστής d_{kt} στην παραπάνω εξίσωση αυξάνεται μετά από μία προαγωγή. Αντίθετα στη φάση του υποβιβασμού, το θετικό κομμάτι του βάρους υποβιβάζεται ενώ το αρνητικό προάγεται.

2.3.2.5 **SVM Classifier**

Η SVM είναι μία σχετικά νέα προσέγγιση μάθησης, η οποία παρουσιάστηκε από τον Vapnik το 1995[45] για την επίλυση προβλημάτων αναγνώρισης προτύπων δύο κλάσεων. Βασίζεται στην αρχή ελαχιστοποίησης του δομικού κινδύνου (Structural Risk Minimization Principal). Η μέθοδος αυτή ορίζεται σε ένα διανυσματικό χώρο, όπου το πρόβλημα είναι η εύρεση μίας επιφάνειας απόφασης η οποία χωρίζει βέλτιστα τα σημεία των δεδομένων σε δύο κλάσεις. Για να καθοριστεί ο βέλτιστος διαχωρισμός πρέπει να εισάγουμε τον όρο «περιθώριο» μεταξύ των δύο κλάσεων. Η παρακάτω εικόνα αποτυπώνει την ιδέα.



Εικόνα 3: Επιφάνεια απόφασης [48]

Για λόγους απλότητας, βλέπουμε μία περίπτωση σε χώρο δύο διαστάσεων με γραμμικά διαχωριζόμενα σημεία δεδομένων, η ιδέα μπορεί να γενικευτεί για χώρο πολλών διαστάσεων και για μη-γραμμικά διαχωριζόμενα στοιχεία.

Η επιφάνεια απόφασης σε γραμμικά διαχωριζόμενο χώρο είναι ένα υπερεπίπεδο. Οι συμπαγείς γραμμές στις παραπάνω εικόνες δείχνουν δύο πιθανές επιφάνειες απόφασης, κάθε μία από τις οποίες διαχωρίζει ορθά τις δύο ομάδες δεδομένων. Οι διακεκομμένες γραμμές, οι οποίες είναι παράλληλες προς τις συμπαγείς δείχνουν πόσο μπορεί να μετακινηθεί η επιφάνεια απόφασης χωρίς να προκαλέσει λανθασμένη κατηγοριοποίηση των δεδομένων. Η απόσταση μεταξύ κάθε συνόλου παράλληλων γραμμών ονομάζεται περιθώριο. Το SVM πρόβλημα είναι η εύρεση της επιφάνειας απόφασης που μεγιστοποιεί το περιθώριο μεταξύ των σημείων των δεδομένων σε ένα σύνολο εκπαίδευσης.

Ακριβέστερα, η επιφάνεια απόφασης που προκύπτει από SVM για γραμμικά διαχωριζόμενο χώρο είναι ένα υπερεπίπεδο το οποίο μπορεί να γραφτεί ως εξής:

$$\vec{w}\vec{x} - b = 0$$

$\vec{x}$ είναι ένα αυθαίρετο σημείο ενός δεδομένου (προς κατηγοριοποίηση) και το διάνυσμα $\vec{w}$ και η σταθερά b προκύπτουν από το σύνολο εκπαίδευσης των γραμμικά διαχωριζόμενων δεδομένων. $D = \{(y_i, \vec{x_i})\}$ συμβολίζει το σύνολο εκπαίδευσης και $y_i \in \{\pm 1\}$ είναι η κατηγοριοποίηση για το $\vec{x}$ (+1 εάν είναι ένα θετικό παράδειγμα και -1 εάν είναι ένα αρνητικό παράδειγμα για μία δεδομένη κλάση), το SVM πρόβλημα είναι ο προσδιορισμός των $\vec{w}$ και b που ικανοποιούν τους παρακάτω περιορισμούς:

$\vec{w} \times \vec{x_i} - b \geq +1$ για $y_i = +1$
$\vec{w} \times \vec{x_i} + b \leq -1$ για $y_i = -1$

Το SVM πρόβλημα μπορεί να λυθεί χρησιμοποιώντας quadratic τεχνικές προγραμματισμού [45,46,47]. Οι αλγόριθμοι για την επίλυση γραμμικά διαχωριζόμενων περιπτώσεων μπορούν να επεκταθούν και για τη λύση γραμμικών μη-διαχωριζόμενων περιπτώσεων είτε εισάγοντας τα υπερεπίπεδα εύκαμπτου περιθωρίου είτε σχεδιάζοντας τα αρχικά διανύσματα δεδομένων σε ένα χώρο υψηλότερων διαστάσεων όπου τα νέα χαρακτηριστικά διαθέτουν όρους αλληλεπίδρασης με τα αρχικά χαρακτηριστικά και τα σημεία των δεδομένων γίνονται γραμμικά διαχωριζόμενα[45],[46],[47].

Μία ενδιαφέρουσα ιδιότητα του SVM αλγορίθμου είναι το γεγονός ότι η επιφάνεια απόφασης καθορίζεται μόνο από τα σημεία των δεδομένων τα οποία απέχουν $\frac{1}{\|\vec{w}\|}$ από το επίπεδο απόφασης. Τα σημεία αυτά καλούνται διανύσματα υποστήριξης (support vectors), τα οποία είναι τα μόνα αποτελεσματικά στοιχεία του συνόλου εκπαίδευσης, εάν απομακρυνόταν όλα τα υπόλοιπα σημεία, ο αλγόριθμος θα κατέληγε στην ίδια συνάρτηση απόφασης. Η ιδιότητα αυτή κάνει τον SVM θεωρητικά μοναδικό και διαφορετικό από πολλές άλλες μεθόδους όπως οι kNN και NB όπου όλα τα σημεία των δεδομένων του συνόλου εκπαίδευσης χρησιμοποιούνται για τη βελτιστοποίηση της συνάρτησης απόφασης.

2.3.3 Αποτελέσματα Πειραμάτων

Στα πειράματα που διεξήχθησαν από τους Songo Tan και Jin Zhang[44], χρησιμοποιήθηκε το μέτρο F1, το οποίο προτάθηκε από τον van Rijsbergen[53], για την αξιολόγηση ενός συστήματος σημασιολογικής κατηγοριοποίησης. Αυτό το μέτρο αξιολόγησης συνδυάζει την ανάκληση πληροφορίας (recall) και την ακρίβεια προσέγγισης (precision) με τον ακόλουθο τρόπο:

$$Recall = \frac{αριθμός\ αληθώς\ θετικών\ προβλέψεων}{αριθμός\ θετικών\ παραδειγμάτων}$$

$$Precision = \frac{αριθμός\ αληθώς\ θετικών\ προβλέψεων}{αριθμός\ θετικών\ προβλέψεων}$$

$$F1 = \frac{2 \times Recall \times Precision}{(Recall + Precision)}$$

Για λόγους ευκολίας της σύγκρισης, συνοψίζονται οι F1 βαθμολογίες των διαφορετικών κατηγοριών χρησιμοποιώντας τις micro και macro μέσες τιμές των F1 βαθμολογιών.

$$Micro - F1 = F1 κατηγοριών\ και\ εγγράφων$$
$$Macro - F1 = μέση\ τιμή\ των\ F1\ τιμών\ μίας\ κατηγορίας$$

Οι MicroF1 και MacroF1 τιμές δίνουν έμφαση στην απόδοση του συστήματος για τις συχνά εμφανιζόμενες και τις σπάνιες κατηγορίες αντίστοιχα. Χρησιμοποιώντας αυτές τις μέσες τιμές μπορεί να παρατηρηθεί η επίδραση που έχουν διαφορετικά είδη δεδομένων σε ένα σύστημα κατηγοριοποίησης.

Οι παρακάτω πίνακες δείχνουν τη βέλτιστη επίδοση των τεσσάρων μεθόδων επιλογής χαρακτηριστικών όταν συνδυαστούν με πέντε μεθόδους μάθησης.

Για τον Winnow αλγόριθμο η αρχική τιμή βάρους $w_{il}^{+}(w_{il}^{-})$ έχει οριστεί στο 2.0(1.0) και η τιμή κατωφλίου αρχικοποιήθηκε στο 1.0. Η παράμετρος προαγωγής α και υποβιβασμού β έχουν οριστεί στο 1.2 και 0.8 αντίστοιχα.

Για το αλγόριθμο KNN ο αριθμός κ των γειτόνων έχει οριστεί στο 13.

	Centroid	KNN	Winnow	NB	SVM	Average
MI	0,8129	0,7943	0,8090	0,8012	0,8257	0,8086
IG	0,8736	0,8756	0,8981	0,8883	0,9060	**0,8883**
CHI	0,8658	0,8433	0,8776	0,8913	0,8903	0,8737
DF	0,8668	0,8511	0,8805	0,8717	0,8521	0,8644
Average	0,8548	0,8411	0,8663	0,8631	**0,8685**	

Βέλτιστες MicroF1 των τεσσάρων μεθόδων επιλογής χαρακτηριστικών συνδυασμένες με πέντε μεθόδους μάθησης

	Centroid	KNN	Winnow	NB	SVM	Average
MI	0,8084	0,7841	0,8049	0,7866	0,8244	0.8017
IG	0,8681	0,8730	0,8996	0,8840	0,9043	**0,8858**
CHI	0,8602	0,8404	0,8739	0,8882	0,8888	0,8703
DF	0,8612	0,8468	0,8777	0,8644	0,8480	0,8596
Average	0,8495	0,8361	0,8640	0,8558	**0,8664**	

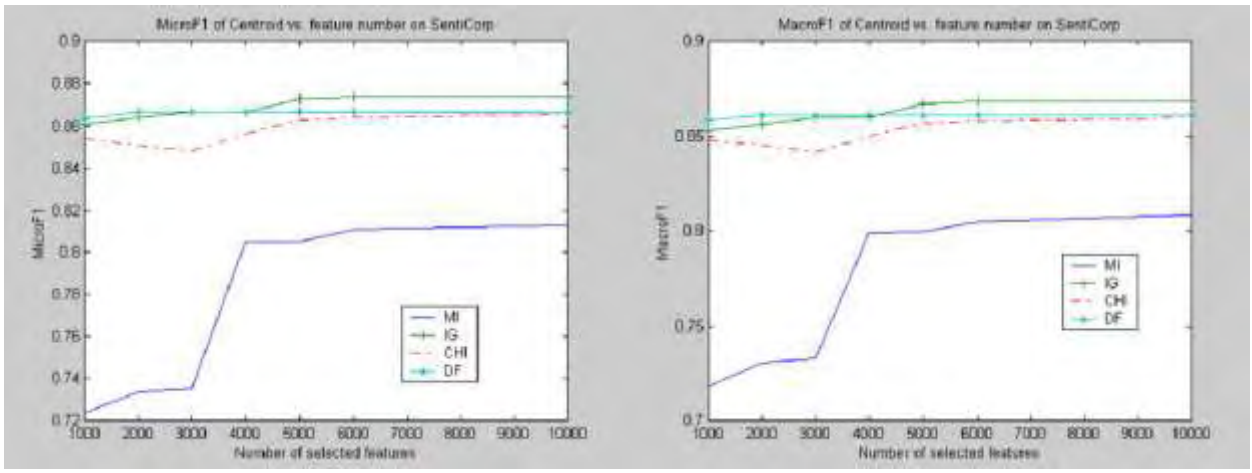
Βέλτιστες MacroF1 των τεσσάρων μεθόδων επιλογής χαρακτηριστικών συνδυασμένες με πέντε μεθόδους μάθησης

Από τους δύο αυτούς πίνακες προκύπτουν τα επόμενα συμπεράσματα:

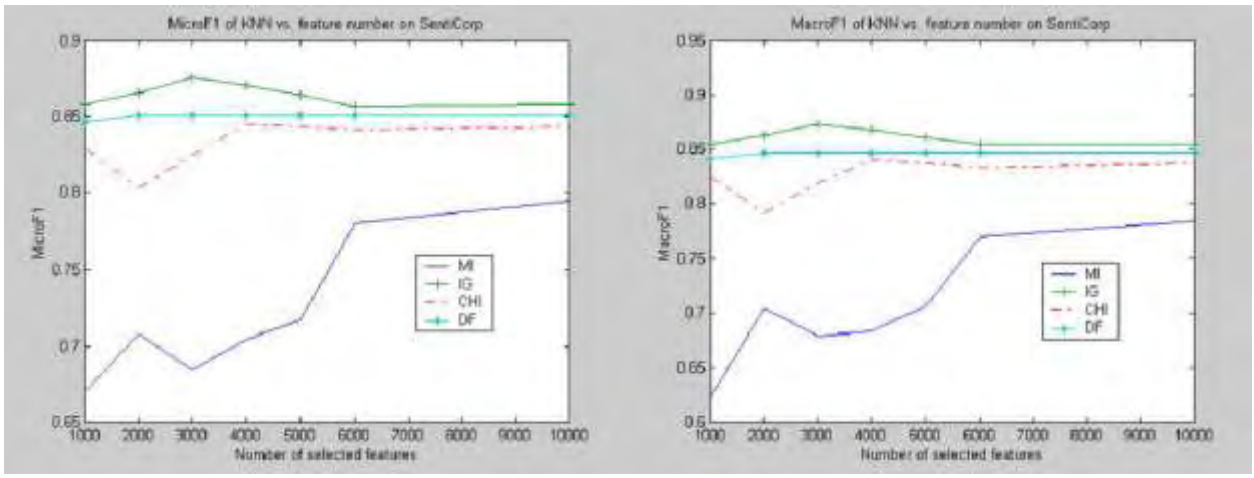
- Κατά πρώτων, όσων αφορά τις μεθόδους επιλογής χαρακτηριστικών, η μέθοδος IG έχει τη βέλτιστη απόδοση συνδυαζόμενη με όλες σχεδόν τις μεθόδους μάθησης. Η μέση MicroF1 τιμή της IG μεθόδου είναι 0.8883, η οποία είναι κατά μία εκατοστιαία μονάδα μεγαλύτερη από τη μέση τιμή της μεθόδου CHI (0.8737), δύο εκατοστιαίες μονάδες μεγαλύτερη της DF μεθόδου(0.8644) και κατά οχτώ εκατοστιαίες μονάδες μεγαλύτερη της MI μεθόδου (0.8086). Επομένως η IG μέθοδος είναι η βέλτιστη μέθοδος επιλογής σημασιολογικών όρων.

  Ανάμεσα στις μεθόδους μάθησης, η SVM μέθοδος είναι αυτή που παράγει το βέλτιστο μέσο όρο MicroF1 βαθμολογιών (0.8685) ο οποίος είναι ελάχιστα μεγαλύτερος του αντίστοιχου της μεθόδου Winnow (0.8663) και της μεθόδου NB (0.8631). Αυτή η παρατήρηση δηλώνει ότι οι SVM, NB, και Winnow μέθοδοι είναι κατάλληλες για την ανάλυση συναισθήματος.

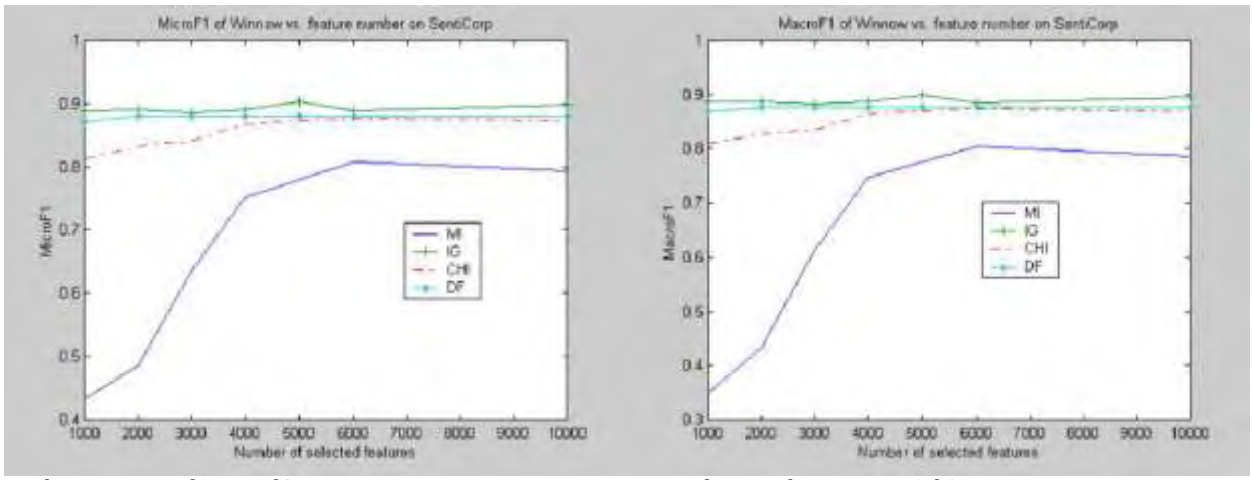
  Στις παρακάτω εικόνες βλέπουμε τις καμπύλες επίδοσης των πέντε μεθόδων μάθησης σε σχέση με το πλήθος των χαρακτηριστικών που επιλέγονται. Από τις εικόνες αυτές παρατηρούμε ότι όταν το πλήθος των χαρακτηριστικών ξεπερνά τις 6.000, όλες οι μέθοδοι μάθησης έχουν λογική και ικανοποιητική επίδοση. Για παράδειγμα χρησιμοποιώντας ένα σύνολο μεγαλύτερο των 6.000 χαρακτηριστικών οι καμπύλες επίδοσης της SVM μεθόδου συνδυαζόμενης με τις μεθόδους IG, CHI και DF παραμένουν σχεδόν αμετάβλητες. Συνεπώς ένα σύνολο 6.000 χαρακτηριστικών ή μεγαλύτερο είναι επαρκές για την κατηγοριοποίηση συναισθήματος.



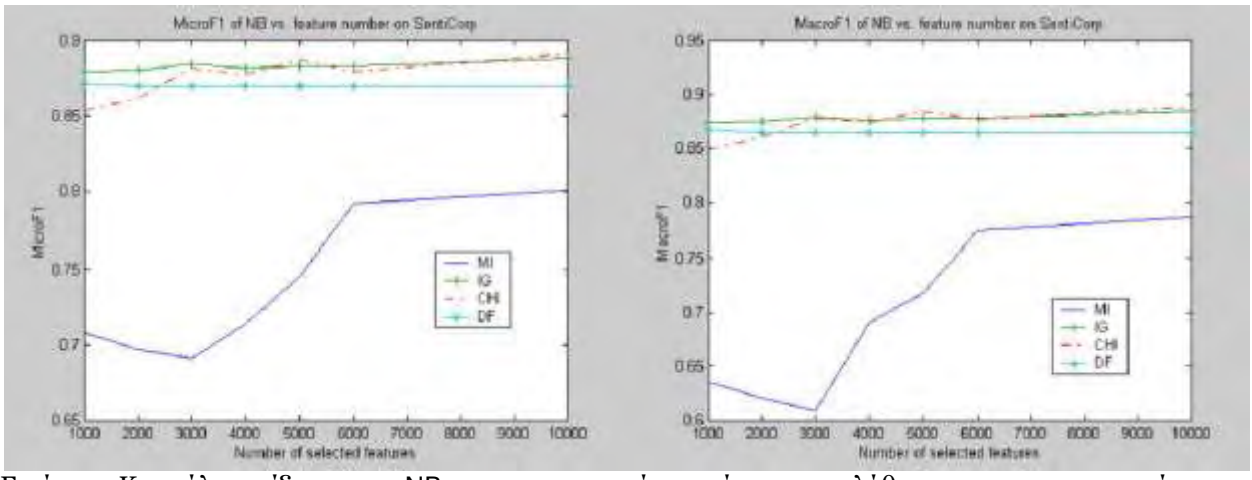
Εικόνα 4: Καμπύλες επίδοσης του κεντροειδούς κατηγοριοποιητή σε σχέση με το πλήθος των χαρακτηριστικών που έχουν επιλεγεί.



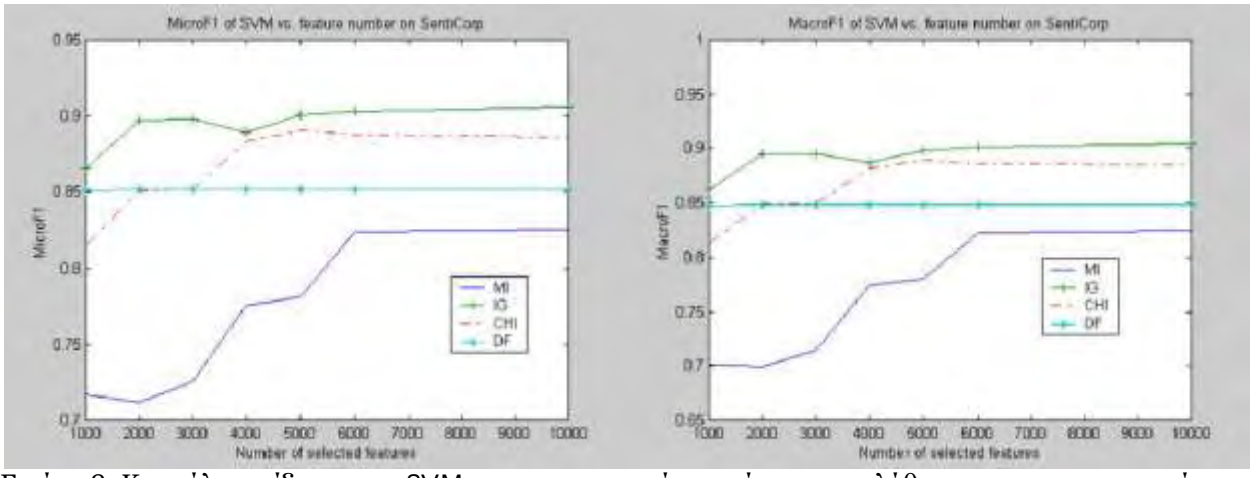
Εικόνα 5: Καμπύλες επίδοσης του kNN κατηγοριοποιητή σε σχέση με το πλήθος των χαρακτηριστικών που έχουν επιλεγεί.



Εικόνα 6: Καμπύλες επίδοσης του Winnow κατηγοριοποιητή σε σχέση με το πλήθος των χαρακτηριστικών που έχουν επιλεγεί.



Εικόνα 7: Καμπύλες επίδοσης του NB κατηγοριοποιητή σε σχέση με το πλήθος των χαρακτηριστικών που έχουν επιλεγεί.



Εικόνα 8: Καμπύλες επίδοσης του SVM κατηγοριοποιητή σε σχέση με το πλήθος των χαρακτηριστικών που έχουν επιλεγεί.

- Η δεύτερη παρατήρηση είναι ότι οι τρείς μέθοδοι επιλογής χαρακτηριστικών (IG, CHI και DF) συνδυαζόμενες με τέσσερις μεθόδους μάθησης (Centroid, KNN, Winnow, και NB) έχουν παρόμοια και ικανοποιητική επίδοση. Σε κάθε περίπτωση η μέθοδος MI δεν έχει συγκρίσιμη επίδοση με καμία από τις άλλες μεθόδους.

3 Ανάλυση Συναισθήματος στην Ελληνική Γλώσσα

Μέχρι στιγμής μεγάλος όγκος έρευνας έχει αφιερωθεί για την συναισθηματική κατηγοριοποίηση εγγράφων γραμμένων στην αγγλική γλώσσα. Η έρευνα αυτή χωρίζεται σε δύο κατηγορίες. Στην πρώτη κατηγορία ανήκουν οι προσεγγίσεις αυτές που εκπαιδεύουν έναν κατηγοριοποιητή συναισθήματος βασιζόμενες στη συχνότητα εμφάνισης διάφορων όρων στα έγγραφα. Στη δεύτερη κατηγορία ανήκουν οι προσεγγίσεις οι οποίες βασίζονται σε ένα λεξικό το οποίο περιέχει χαρακτηρισμένες λέξεις ως θετικές ή αρνητικές και το οποίο χρησιμοποιείται για τον υπολογισμό της συνολικής θετικής/αρνητικής βαθμολογίας ενός κειμένου.

Παρόλα αυτά για κείμενα γραμμένα στην ελληνική γλώσσα έχουν γίνει λίγες προσπάθειες με σημαντικότερη αυτή του BalkaNet project συντονιστής της οποίας ήταν το εργαστήριο βάσεων δεδομένων του πανεπιστημίου Πατρών. Το BalkaNet ανήκει στην δεύτερη κατηγορία ερευνών που παρουσιάστηκαν παραπάνω. Το κύριο αντικείμενου του BalkaNet αφορά τη δημιουργία ενός πολύγλωσσου σημασιολογικού δικτύου για τις βαλκανικές γλώσσες. Ένα τέτοιο δίκτυο περιλαμβάνει έννοιες όλων αυτών των γλωσσών συνδεδεμένες με προκαθορισμένες λεξιλογικές και σημασιολογικές σχέσεις. Η βασική λεξιλογική μονάδα του BalkaNet είναι το σύνολο συνώνυμων όρων (synset). Κάθε σύνολο συνώνυμων όρων περιλαμβάνει όρους όλων των γλωσσών που μοιράζονται την ίδια σημασιολογική έννοια. Οι όροι που συμμετέχουν στο ίδιο synset ενώνονται με βάση τις εσωτερικές λεξιλογικές σχέσεις μίας γλώσσας ενώ τα synsets συνδέονται μεταξύ τους μέσω προκαθορισμένων σημασιολογικών σχέσεων (hyponymy, meronymy).

Κάθε μονόγλωσσο WordNet αναπτύχθηκε χρησιμοποιώντας διάφορες διαθέσιμες λεξιλογικές πηγές, όπως επεξηγηματικά λεξικά, γλωσσάρια, λεξικά συνωνύμων κ.α. Όροι οι οποίοι εξήχθησαν από τις παραπάνω πηγές επεξεργάστηκαν και εισήχθησαν σε κάθε WordNet. Οι όροι αυτοί αποτελούν τις βασικές έννοιες του WordNet κάθε γλώσσας.

Όμως σήμερα, με την ανάπτυξη του διαδικτύου και την πληθώρα κείμενων που μπορούμε να βρούμε αναρτημένα σε blogs, forums και άλλες σελίδες δικτύωσης γραμμένα στα ελληνικά καθίσταται δυνατή και η ανάπτυξη ενός συστήματος κατηγοριοποίησης συναισθήματος και με τη πρώτη προσέγγιση, αυτή δηλαδή της μηχανικής μάθησης. Αυτό γίνεται από την i-sieve Technologies η οποία είναι συνδεδεμένη με το Εθνικό Κέντρο Έρευνας «Δημόκριτος». Η i-sieve Technologies αναλαμβάνει να κάνει ανάλυση συναισθήματος των σχολίων που ανακτά από το διαδίκτυο για να απαντήσει στις ερωτήσεις των πελατών της, οι οποίες μπορεί να είναι για τη γενική άποψη του κοινού για το προϊόν που παράγουν ή την υπηρεσία που προσφέρουν. Το σύστημα που χρησιμοποιεί η i-sieve Technologies είναι κλειστού κώδικα αλλά κάνει γνωστό μέσο του ιστοτόπου της ότι λειτουργεί με τη μέθοδο της μηχανικής μάθησης.

4 Δημοσίευση Περιεχομένου στον Ιστό (Web Syndication)

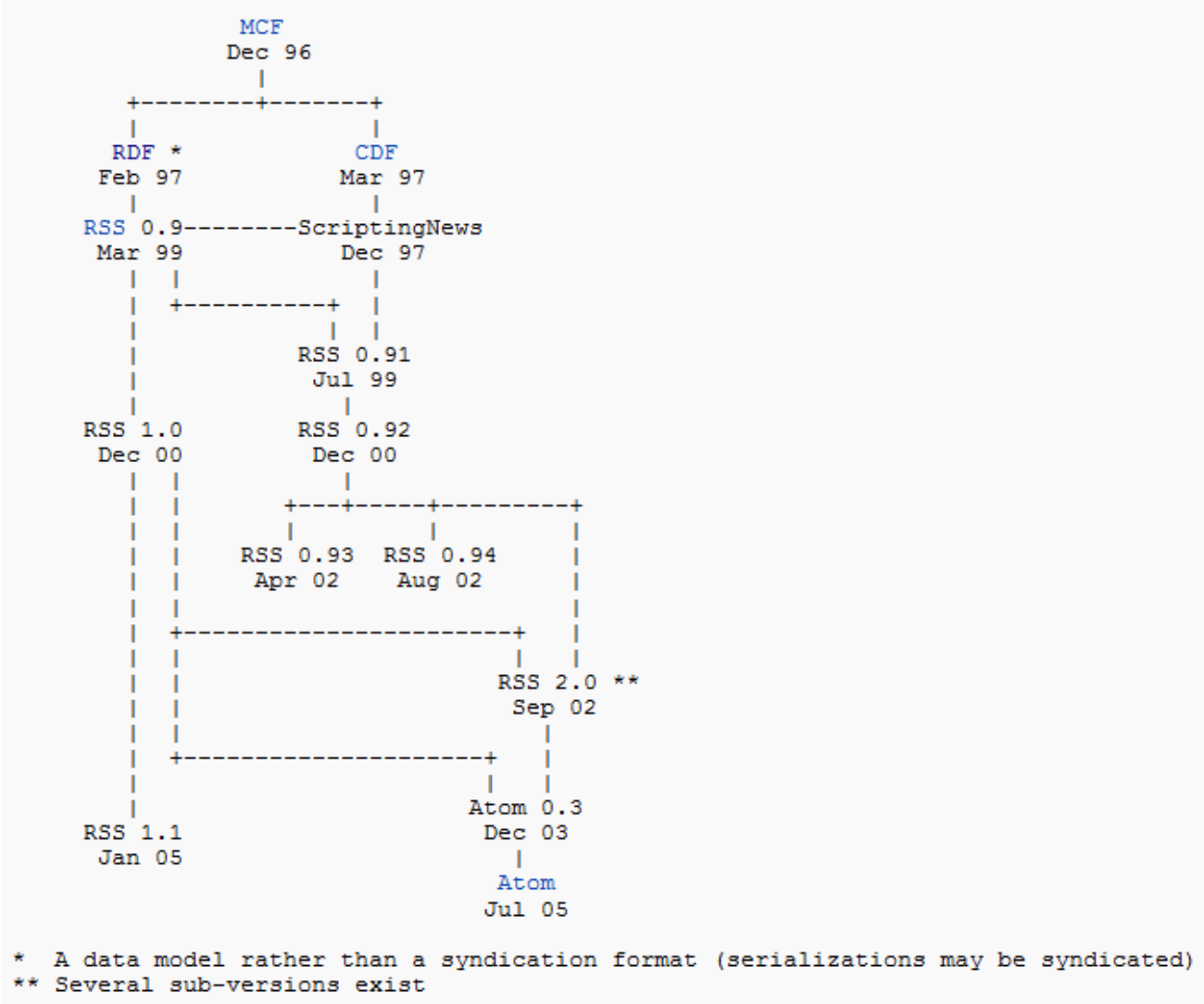
Η δημοσίευση περιεχομένου στον ιστό είναι ένα είδος δημοσίευσης κατά την οποία το υλικό μίας ιστοσελίδας γίνεται διαθέσιμο σε πολλαπλούς χρήστες. Συχνά η δημοσίευση περιεχομένου στον ιστό αναφέρεται στη δημιουργία web feeds (τροφοδότες ιστού) τα οποία γίνονται διαθέσιμα από μία ιστοσελίδα με σκοπό να παρέχει στον κόσμο μία περίληψη του ανανεωμένου περιεχόμενού της, για παράδειγμα τα τελευταία νέα, αλλά και να μπορεί να αναδημοσιευτεί σε οποιαδήποτε άλλη πλατφόρμα εύκολα και γρήγορα.

Ένα web feed (τροφοδότης ιστού) είναι ένα format δεδομένων το οποίο χρησιμοποιείται για να παρέχει στους χρήστες περιεχόμενο το οποίο μεταβάλλεται συχνά. Οι διανομείς περιεχομένου δημοσιεύουν ένα web feed, επιτρέποντας στους χρήστες να γίνουν συνδρομητές σε αυτό. Η δημιουργία μία συλλογής από web feeds σε ένα σημείο είναι γνωστή ως aggregation (συγκέντρωση), η οποία εφαρμόζεται από έναν aggregator (συσσωρευτή).

Ένα τυπικό σενάριο χρήσης ενός web feed είναι το παρακάτω: ένας πάροχος περιεχομένου (content provider) δημοσιεύει ένα feed link σε μια ιστοσελίδα στο οποίο οι χρήστες μπορούν να εγγραφούν χρησιμοποιώντας ένα πρόγραμμα συσσωρευτή (aggregator program) γνωστό και ως feed reader, το οποίο τρέχει τοπικά στον υπολογιστή τους. Η εγγραφή αυτή συνήθως είναι τόσο απλή όσο ένα drag and drop του link από τον browser στον aggregator. Όταν του ζητηθεί από το χρήστη ο aggregator αναζητά σε όλους τους servers που έχει καταχωρημένους στη λίστα με τα feeds για νέο περιεχόμενο. Εάν υπάρχει κάτι νέο ο aggregator είτε δημιουργεί ένα σημείωμα για το χρήστη με το οποίο τον ενημερώνει είτε κατεβάζει το νέο περιεχόμενο. Οι aggregators μπορούν να ρυθμιστούν να ελέγχουν περιοδικά για νέο περιεχόμενο χωρίς την παρεμβολή του χρήστη.

Η δημοσίευση περιεχομένου ωφελεί και τους ιστότοπους οι οποίοι παρέχουν τις πληροφορίες αλλά και τους ιστότοπους οι οποίοι τις προβάλουν. Για τους ιστότοπους οι οποίοι προβάλουν το περιεχόμενο το όφελος προέρχεται από το γεγονός ότι εμπλουτίζουν το περιεχόμενό τους και δίνουν αμεσότητα στη πληροφορία που περιέχουν, καθιστώντας τους πιο ελκυστικούς για τους χρήστες. Οι ιστότοποι οι οποίοι παρέχουν τις πληροφορίες τους μέσω web syndication ωφελούνται από το γεγονός ότι το περιεχόμενό τους προβάλλεται απευθείας σε πολλές πλατφόρμες, κάτι που οδηγεί στην αύξηση της κίνησής τους.

Οι δύο κύριες οικογένειες των format για τη δημιουργία web feeds είναι το RSS και το ATOM.



Εικόνα 9: Εξέλιξη προτύπων για τη δημοσίευση περιεχομένου στον ιστό

Στην παραπάνω εικόνα βλέπουμε ένα δέντρο στο οποίο φαίνεται η εξέλιξη των προτύπων για τη δημοσίευση περιεχομένου στον ιστό. Για κάθε πρότυπο σημειώνεται η ημερομηνία κατά οποία την έγινε η πρώτη μεγάλη δημοσίευση για το πρότυπο αυτό.

4.1 RSS 2.0

Το ακρώνυμο RSS προέρχεται από τις λέξεις Really Simple Syndication. Το RSS αποτελεί μια διάλεκτο της XML. Όλα τα RSS αρχεία πρέπει να υπακούν στους κανόνες σύνταξης της XML 1.0 .

Ένα RSS έγγραφο είναι ένα στοιχείο <rss>, με ένα υποχρεωτικό χαρακτηριστικό το οποίο ονομάζεται version και σαν τιμή παίρνει την έκδοση RSS στην οποία κατατάσσεται το συγκεκριμένο έγγραφο.

Υποκείμενο στο <rss> στοιχείο βρίσκεται ένα μοναδικό στοιχείο <channel> το οποίο περιέχει πληροφορία, μεταδεδομένα, για το κανάλι και το περιεχόμενό του.

Παρακάτω βλέπουμε ένα τυπικό αρχείο RSS:

```
<rss version="2.0">
   <channel>
      <title>Liftoff News</title>
      <link>http://liftoff.msfc.nasa.gov/</link>
      <description>Liftoff to Space Exploration.</description>
      <language>en-us</language>
      <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
      <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
      <docs>http://blogs.law.harvard.edu/tech/rss</docs>
      <generator>Weblog Editor 2.0</generator>
      <managingEditor>editor@example.com</managingEditor>
      <webMaster>webmaster@example.com</webMaster>
      <item>
         <description>Sky watchers in Europe, Asia, and parts of Alaska and
Canada will experience a &lt;a
href="http://science.nasa.gov/headlines/y2003/30may_solareclipse.htm"&gt;part
ial eclipse of the Sun&lt;/a&gt; on Saturday, May 31st.</description>
         <pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
         <guid>http://liftoff.msfc.nasa.gov/2003/05/30.html#item572</guid>
      </item>
      <item>
         <title>Astronauts' Dirty Laundry</title>
         <link>http://liftoff.msfc.nasa.gov/news/2003/news-laundry.asp</link>
         <description>Compared to earlier spacecraft, the International Space
Station has many luxuries, but laundry facilities are not one of them.
Instead, astronauts have other options.</description>
         <pubDate>Tue, 20 May 2003 08:56:02 GMT</pubDate>
         <guid>http://liftoff.msfc.nasa.gov/2003/05/20.html#item570</guid>
      </item>
   </channel>
</rss>
```

Παρακάτω βλέπουμε μία λίστα με τα στοιχεία του καναλιού τα οποία πρέπει υποχρεωτικά να υπάρχουν στο RSS έγγραφο και δίπλα μία σύντομη περιγραφή τους.

Στοιχείο	**Περιγραφή**
title	Το όνομα του καναλιού. Είναι το όνομα με το οποίο ο κόσμος αναφέρεται στην υπηρεσία αυτή. Εάν υπάρχει μία HTML σελίδα η οποία περιέχει την ίδια πληροφορία με το RSS αρχείο, ο τίτλος του καναλιού πρέπει να είναι ο ίδιος με τον τίτλο της σελίδας.
link	Η διεύθυνση της HTML σελίδας στην οποία αναφέρεται το κανάλι
description	Φράση ή πρόταση η οποία περιγράφει το κανάλι

Προαιρετικά στοιχεία καναλιού:

Στοιχείο	**Περιγραφή**
language	Η γλώσσα στην οποία το κανάλι έχει γραφτεί. Αυτό επιτρέπει την συγκέντρωση, για παράδειγμα, όλων των ελληνικών ιστοσελίδων σε μία και μόνο σελίδα
copyright	Ένδειξη για τα πνευματικά δικαιώματα που διέπουν το περιεχόμενο του καναλιού.
managingEditor	Η διεύθυνση e-mail του υπεύθυνου για το περιεχόμενο που δημοσιεύεται από το RSS
webMaster	Η διεύθυνση e-mail του υπεύθυνου για τεχνικά θέματα τα οποία συνδέονται με το κανάλι
pubDate	Η ημερομηνία δημοσίευσης του περιεχομένου στο κανάλι. Για παράδειγμα, οι New York Times κάνουν δημοσιεύσεις καθημερινά, έτσι η ημερομηνία αλλάζει κάθε 24 ώρες, εκείνη τη στιγμή αλλάζει και το περιεχόμενο του στοιχείου αυτού. Οι ημερομηνίες στο RSS έγγραφο ακολουθούν τις προδιαγραφές που ορίζονται από το RFC 822, με την μόνη εξαίρεση ότι οι χρονιές μπορούν να αναπαρασταθούν είτε με 2 είτε με 4 ψηφία.

lastBuildDate	Η τελευταία στιγμή κατά την οποία αλλάχθηκε το περιεχόμενο του καναλιού.
category	Ορίζει μία ή περισσότερες κατηγορίες στις οποίες ανήκει το κανάλι
generator	Ένα αλφαριθμητικό το οποίο υποδεικνύει το πρόγραμμα που χρησιμοποιήθηκε για τη δημιουργία του καναλιού.
docs	Η διεύθυνση URL η οποία στοχεύει στην τεκμηρίωση του format που χρησιμοποιήθηκε για τη δημιουργία του RSS αρχείου
cloud	Επιτρέπει σε διεργασίες να εγγράφονται σε ένα σύννεφο για να ενημερώνονται για τυχόν αναπροσαρμογές του καναλιού εφαρμόζοντας ένα ελαφρύ πρωτόκολλο δημοσιεύσεων – εγγραφών στο RSS feed
ttl	Ακρώνυμο για το time to live. Είναι ο αριθμός των λεπτών που δηλώνει για πόσο ένα κανάλι μπορεί να παραμείνει στη μνήμη πριν ανανεωθεί από την πηγή
image	Ορίζει μία GIF, JPEG ή PNG εικόνα η οποία μπορεί να εμφανίζεται μαζί με το κανάλι
rating	Η βαθμολογία PICS του καναλιού
textInput	Ορίζει ένα input box το οποίο μπορεί να εμφανίζεται με το κανάλι.
skipHours	Μία ένδειξη για τους συλλέκτες η οποία τους ενημερώνει για το ποιες ώρες μπορούν να παραλείψουν χωρίς να κάνουν συλλογή περιεχομένων
skipDays	Μία ένδειξη για τους συλλέκτες η οποία τους ενημερώνει για το ποιες μέρες μπορούν να παραλείψουν χωρίς να κάνουν συλλογή περιεχομένων

Ένα κανάλι μπορεί να περιέχει έναν αριθμό από αντικείμενα (items). Ένα αντικείμενο μπορεί να αντιπροσωπεύει μία ιστορία, όπως για παράδειγμα μία ιστορία σε μία εφημερίδα ή ένα περιοδικό. Εάν συμβαίνει αυτό η περιγραφή του αποτελεί μία περίληψη της συνολικής ιστορίας και το στοιχείο link στοχεύει στη συνολική ιστορία. Ένα αντικείμενο μπορεί ακόμη να είναι πλήρες και ανεξάρτητο, στην περίπτωση αυτή το στοιχείο description περιέχει το σύνολο του κειμένου και τα στοιχεία link και title

μπορούν να παραληφθούν. Όλα τα στοιχεία ενός αντικειμένου είναι προαιρετικά αλλά τουλάχιστον ένα στοιχείο μεταξύ του τίτλου και της περιγραφής πρέπει να υπάρχει.

Στοιχείο	**Περιγραφή**
title	Ο τίτλος του αντικειμένου
link	Η διεύθυνση URL του αντικειμένου
description	Η περίληψη του αντικειμένου
author	Η διεύθυνση e-mail του συντάκτη του αντικειμένου
category	Κατατάσσει το αντικείμενο σε μία ή περισσότερες κατηγορίες
comments	URL της σελίδας με τα σχόλια τα οποία σχετίζονται με το αντικείμενο
enclosure	Περιγραφή αντικειμένου media το οποίο επισυνάπτεται στο συγκεκριμένο αντικείμενο
guid	Ένα αλφαριθμητικό το οποίο προσδιορίζει μοναδικά το αντικείμενο
pubDate	Ημερομηνία δημοσίευσης του αντικειμένου
source	Το RSS κανάλι από το οποίο προέρχεται το αντικείμενο

4.2 ATOM 1.0

Τα αρχεία τροφοδοσίας ATOM είναι XML έγγραφα. Τα αρχεία ATOM είναι καλοσχηματισμένα XML αρχεία τα οποία ακολουθούν τους κανόνες σύνταξης της XML 1.0.

Υπάρχουν δύο τύποι εγγράφων ορισμένα για τα ATOM αρχεία.

- ATOM feed documents – αναπαριστά ένα ATOM feed εμπεριέχοντας μεταδεδομένα που αφορούν το feed και μερικές ή όλες τις καταχωρίσεις (entries) που σχετίζονται με αυτό. Το <root> στοιχείο του είναι το <feed>

- ATOM entry documents – αναπαριστά μόνο μία ATOM καταχώρηση. Το root στοιχείο του είναι το <entry>

Δομή των ATOM Feed Εγγράφων

Ένα ATOM feed έγγραφο πρέπει να έχει ένα root στοιχείο το οποίο ονομάζεται “feed”. Μέσα στο feed στοιχείο εσωκλείονται ένα ή περισσότερα στοιχεία “entry” (καταχώρηση).

Ένα ATOM feed έγγραφο έχει την παρακάτω δομή:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
 <title>The Most Popular FAQ Entries for Webmasters</title>
 <subtitle>The top 3 popular
  entries of this week on our comprehensive collection
  of Webmaster FAQs.</subtitle>
 <link rel="self"
  href="http://dev.fyicenter.com/faq/top_3_atom.xml"/>
 <id>http://dev.fyicenter.com/faq/top_3_atom.xml</id>
 <updated>2005-07-13T18:30:02Z</updated>
 <author>
  <name>FYIcenter.com</name>
  <email>noreply@fyicenter.com</email>
 </author>
<entry>
  <title>Atom Feed Introduction and File Generation</title>
  <link rel="alternate"
   href="http://dev.fyicenter.com/faq/rss/index.html"/>
  <id>href="http://dev.fyicenter.com/faq/rss/index.html</id>
  <updated>2005-07-13T18:30:02Z</updated>
  <summary>A collection of
  16 FAQs on Atom feed file standard. Clear answers are
  provided with tutorial samples on introduction to Atom
  feed file standard; various ways to generate Atom feeds;
  linking Atom feeds to Web pages.</summary>
```

```
 </entry>
 <entry>
  <title>Understanding and Using Sessions in PHP Scripts
  </title>
  <link rel="alternate"
   href="http://dev.fyicenter.com/faq/php/index.html"/>
  <id>href="http://dev.fyicenter.com/faq/php/index.html</id>
  <updated>2005-07-13T18:30:02Z</updated>
  <summary>A collection of 19 tips on understanding and
  using sessions in PHP. Clear explanations and tutorial
  exercises are provided on starting and closing sessions,
  saving and retrieving values in sessions, deciding how
  session IDs to be transferred, deciding where to store
  session files, deciding when to expire session values,
  etc.</summary>
 </entry>
</feed>
```

4.2.1.1 **Υπο-στοιχεία του στοιχείου feed:**

Το στοιχείο feed έχει τα ακόλουθα υπο-στοιχεία

Στοιχείο	**Περιγραφή**
author	Περιέχει προσωπικές πληροφορίες του συγγραφέα των περιεχομένων που παρέχονται από το συγκεκριμένο feed. Ένα στοιχείο feed μπορεί να έχει κανένα, ένα ή περισσότερα στοιχεία author
category	Ορίζει την κατηγορία των περιεχομένων του feed. Ένα feed μπορεί να έχει από ο έως N στοιχεία category
contributor	Παρέχει προσωπικές πληροφορίες για έναν συντελεστή των περιεχομένων του feed. Ένα feed μπορεί να έχει από κανένα μέχρι πολλά στοιχεία contributor
generator	Περιέχει πληροφορίες για το λογισμικό που χρησιμοποιήθηκε για να δημιουργηθεί αυτό το feed. Ένα στοιχείο feed μπορεί να περιέχει είτε ένα είτε κανένα στοιχείο generator
icon	Περιέχει την τοποθεσία URL μίας εικόνας που προσδιορίζει το feed. Ένα feed μπορεί να περιέχει το πολύ ένα στοιχείο icon

logo	Περιέχει την τοποθεσία URL του λογότυπου του feed
id	Ορίζει ένα URI το οποίο προσδιορίζει μοναδικά το συγκεκριμένο feed. Ένα feed πρέπει να περιέχει ακριβώς ένα id
link	Αποτελείται από μία URL διεύθυνση μίας ιστοσελίδας η οποία λειτουργεί σαν αναφορά στο feed. Ένα feed μπορεί να περιέχει από 0 έως N στοιχεία link, παρόλα αυτά, τουλάχιστον ένα στοιχείο link προτείνεται να έχει το χαρακτηριστικό rel με τιμή “self”
rights	Δίνει πληροφορίες για τα δικαιώματα του feed. Ένα feed μπορεί να περιέχει κανένα ή ένα στοιχείο rights
subtitle	Περιέχει μία σύντομη περιγραφή ή τον υπότιτλο του feed. Ένα feed περιέχει το πολύ ένα στοιχείο subtitle
title	Ορίζει τον τίτλο του feed. Κάθε feed περιέχει υποχρεωτικά έναν τίτλο συνεπώς και ένα στοιχείο title.
updated	Περιέχει μία χρονοσφραγίδα η οποία προσδιορίζει τη χρονική στιγμή κατά την οποία το feed ενημερώθηκε για τελευταία φορά. Ένα feed έχει ακριβώς ένα στοιχείο updated.
entry	Περιέχει μία καταχώρηση του feed. Ένα στοιχείο feed μπορεί να έχει 1 έως N στοιχεία entry.

4.2.1.2 **Υπο-στοιχεία του στοιχείου entry**

Το στοιχείο entry έχει τα παρακάτω υπο-στοιχεία ορισμένα:

Στοιχείο	**Περιγραφή**
author	Περιέχει προσωπικές πληροφορίες του συγγραφέα των περιεχομένων που παρέχονται από το συγκεκριμένο feed entry. Ένα στοιχείο entry μπορεί να έχει κανένα, ένα ή περισσότερα στοιχεία author

category	Ορίζει την κατηγορία των περιεχομένων του feed entry. Ένα feed entry μπορεί να έχει από 0 έως N στοιχεία category
content	Ορίζει το περιεχόμενο του συγκεκριμένου feed entry. Ένα feed entry μπορεί να έχει το πολύ ένα content υπο-στοιχείο.
contributor	Παρέχει προσωπικές πληροφορίες για έναν συντελεστή των περιεχομένων του feed entry. Ένα feed entry μπορεί να έχει από κανένα μέχρι πολλά στοιχεία contributor
id	Ορίζει ένα URI το οποίο προσδιορίζει μοναδικά το συγκεκριμένο feed entry. Ένα feed entry πρέπει να περιέχει ακριβώς ένα id
link	URL διεύθυνση μίας ιστοσελίδας η οποία λειτουργεί σαν αναφορά σε αυτό το feed entry. Ένα feed entry μπορεί να έχει από 0 έως N link στοιχεία. Εάν όμως δεν υπάρχει κανένα στοιχείο content στο entry, τότε τουλάχιστον ένα στοιχείο link με χαρακτηριστικό rel και τιμή “alternate” είναι απαραίτητο.
published	Χρονοσφραγίδα η οποία καθορίζει τη στιγμή που δημοσιεύτηκε η καταχώρηση. Κάθε entry μπορεί να περιέχει το πολύ μία χρονοσφραγίδα published
rights	Δίνει πληροφορίες για τα πνευματικά δικαιώματα του feed entry. Ένα feed entry μπορεί να περιέχει κανένα ή ένα στοιχείο rights
source	Προσδιορίζει μία καταχώρηση ενός άλλου feed entry εγγράφου εάν η συγκεκριμένη καταχώρηση είναι αντίγραφο εκείνης. Μία καταχώρηση feed entry μπορεί να έχει το πολύ ένα στοιχείο source
summary	Περιέχει μία σύνοψη της καταχώρησης. Κάθε καταχώρηση μπορεί να έχει είτε μία είτε καμία σύνοψη. Παρόλα αυτά προτείνεται κάθε καταχώρηση να περιέχει μία περίληψη.
title	Ορίζει τον τίτλο του feed entry. Κάθε feed entry περιέχει υποχρεωτικά έναν τίτλο συνεπώς και ένα στοιχείο title.

updated	Περιέχει μία χρονοσφραγίδα η οποία προσδιορίζει τη χρονική στιγμή κατά την οποία το feed entry ενημερώθηκε για τελευταία φορά. Ένα feed entry έχει ακριβώς ένα στοιχείο updated.

4.3 Διαφορές μεταξύ RSS και ATOM

Το RSS έχει υπάρξει το βασικό πρότυπο για τη δημιουργία web feeds για ένα μεγάλο χρονικό διάστημα. Τα web feeds περιέχουν είτε μία σύνοψη είτε το σύνολο του κειμένου μίας ιστοσελίδας. Το πρόβλημα με το RSS είναι οι συχνά συγκεχυμένες και μη τυποποιημένες συμβάσεις που χρησιμοποιεί που οφείλονται εν μέρει στην διασκορπισμένη ανάπτυξή του. Η εμφάνιση του προτύπου ATOM ήρθε ως απάντηση στα σχεδιαστικά λάθη του προτύπου RSS. Το κύριο πλεονέκτημα του ATOM είναι η αναγνώρισή του ως ένα IETF(Internet Engineering Task Force) πρότυπο.

Όντας ένα IETF πρότυπο, το ATOM έπρεπε να ενσωματώσει ορισμένα γνωρίσματα τα οποία το κατέστησαν ευκολότερο στην εφαρμογή του. Κάθε ATOM feed διαθέτει μία ρητή δήλωση του format του περιεχομένου του καθώς και της γλώσσας την οποία χρησιμοποιεί. Τα RSS feeds δεν δηλώνουν τον τύπο των περιεχομένων τους, όμως δεδομένου ότι περιέχουν μόνο είτε κείμενο είτε escaped HTML είναι εύκολο για το browser να διακρίνει τον τύπο των δεδομένων. Ακόμη, το RSS 2.0 χρησιμοποιεί χρονοσφραγίδες οι οποίες διαμορφώνονται σύμφωνα με το format που ορίζεται από το RFC 822, σε αντίθεση με το ATOM για το οποίο επιλέχθηκε η χρήση χρονοσφραγίδων οι οποίες είναι διαμορφωμένες σύμφωνα με τους κανόνες οι οποίοι ορίζονται στο RCF 3393.

Ένα σημαντικό μειονέκτημα του RSS βρίσκεται στον κώδικά του. Ο κώδικας του RSS δεν είναι ιδιαίτερα λειτουργικός σε άλλα λεξιλόγια της XML καθώς δεν προοριζόταν αρχικά να έχει την ιδιότητα αυτή. Ο κώδικας του ATOM έχει χτιστεί από την αρχή έχοντας την συμβατότητα στο νου. Επομένως, ένα μεγάλο τμήμα του κώδικά του είναι επαναχρησιμοποιήσιμο και από άλλα λεξιλόγια της XML όπως αυτό του RSS.

Το γεγονός ότι το RSS υπήρξε το πρώτο πρότυπο για τη δημοσίευση περιεχομένου στον ιστό αποτέλεσε τον πρωταρχικό παράγοντα για την γρήγορη ανάπτυξή του και τη δημοσιότητα που πήρε. Το podcasting προήλθε επίσης από το πρότυπο του RSS όταν προστέθηκε σε αυτό και η υποστήριξη επισύναψης στην έκδοση 2.0 . Παρόλο που και το ATOM έχει προσαρμοστεί στην εξυπηρέτηση του podcasting το RSS κρατά ακόμη ένα μεγάλο ποσοστό αυτής της αγοράς.

Στην παρακάτω εικόνα βλέπουμε τα στοιχεία που υποστηρίζει το πρότυπο ATOM 1.0 και το πρότυπο RSS 2.0. Για το ATOM 1.0 ορίζονται 21 στοιχεία, ενώ για το RSS 2.0 ορίζονται 30. Τα περισσότερα στοιχεία του RSS 2.0 που λείπουν από το ATOM 1.0 είναι

είτε τα στοιχεία εκείνα που στην πράξη δεν χρησιμοποιούταν είτε οι δυνατότητές τους παρέχονται με άλλο τρόπο

RSS 2.0	Atom 1.0	Comments
rss	-	Vestigial in RSS
channel	feed	
title	title	
link	link	Atom defines an extensible family of rel values
description	subtitle	
language	-	Atom uses standard xml:lang attribute
copyright	rights	
webMaster	-	
managingEditor	author or contributor	
pubDate	published (in entry)	Atom has no feed-level equivalent
lastBuildDate (in channel)	updated	RSS has no item-level equivalent
category	category	
generator	generator	
docs	-	
cloud	-	
ttl	-	<ttl> is problematic, prefer HTTP 1.1 cache control
image	logo	Atom recommends 2:1 aspect ratio
-	icon	As in favicon.ico
rating	-	
textInput	-	
skipHours	-	
skipDays	-	
item	entry	
author	author	
-	contributor	
description	summary and/or content	depending on whether full content is provided
comments	-	
enclosure	-	rel="enclosure" on <link> in Atom
guid	id	
source	-	rel="via" on <link> in Atom
-	source	Container for feed-level metadata to support aggregation

Εικόνα 10: Διαφορές μεταξύ RSS και Atom

5 LingPipe [63]

Η μέθοδος που προτείνεται από τους Bo Pang και Lillian Lee[54] αποτελείται από τα ακόλουθα βήματα:

- Χαρακτηρισμός των προτάσεων του κειμένου ως αντικειμενικές ή υποκειμενικές και απόρριψη των πρώτων.

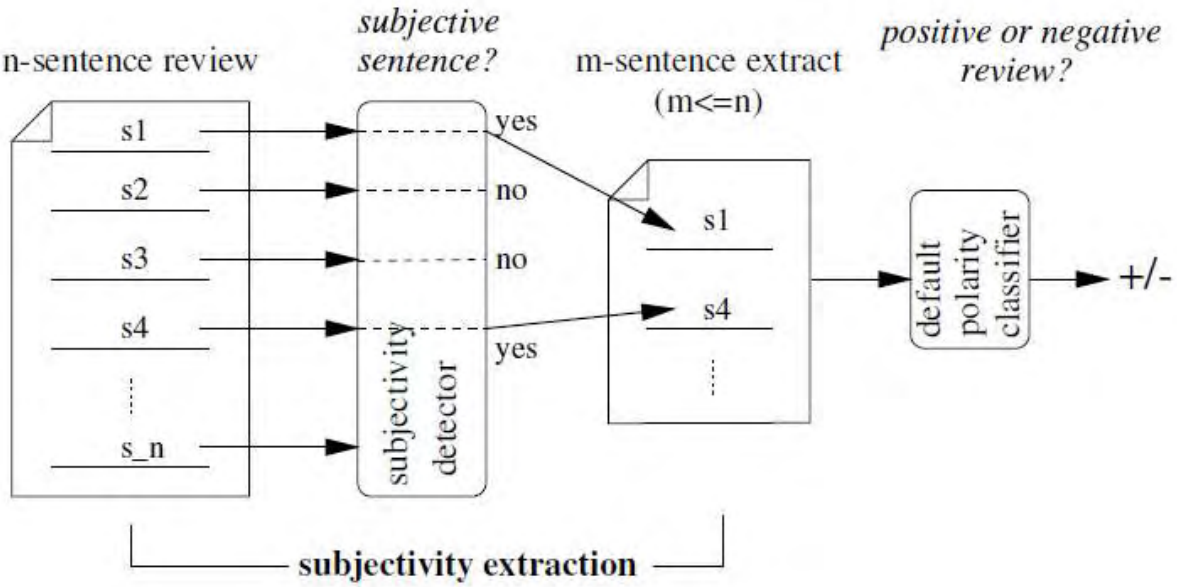
- Εφαρμογή ενός τυπικού κατηγοριοποιητή μηχανικής μάθησης στο αποτέλεσμα που εξάγεται από το πρώτο βήμα.

Αυτή η διαδικασία προστατεύει τον κατηγοριοποιητή από το να μελετά κείμενο που δεν εκφράζει άποψη ή και ακόμα και παραπλανητικό κείμενο. Για παράδειγμα η πρόταση «Ο πρωταγωνιστής προσπαθεί να διατηρήσει το καλό του όνομα» παρόλο που περιέχει τη λέξη «καλό» δεν προσφέρει πληροφορία για την άποψη του συγγραφέα και στην πραγματικότητα θα μπορούσε να είναι μέρος μία αρνητικής κριτικής για μία ταινία. Ακόμη, το υποκειμενικό απόσπασμα που προκύπτει από το πρώτο βήμα της μεθόδου μπορεί να χρησιμοποιηθεί ως μία περίληψη του εγγράφου.

Τα αποτελέσματα δείχνουν ότι το υποκειμενικό απόσπασμα του κειμένου αντιπροσωπεύει με ακρίβεια την πληροφορία συναισθήματος του αρχικού κειμένου σε μία πιο συμπαγή μορφή: ανάλογα με την επιλογή του κατηγοριοποιητή μπορεί να επιτευχθεί σημαντική βελτίωση στα ποσοστά επιτυχίας της κατηγοριοποίησης (από 82,8% σε 86,4%) ή να διατηρηθεί η ίδια επίδοση κατηγοριοποίησης διατηρώντας μόνο ένα 60% των λέξεων του αρχικού κειμένου.

5.1 Αρχιτεκτονική Συστήματος

Αρχικά χρησιμοποιείται ένας ανιχνευτής υποκειμενικότητας ο οποίος αποφαίνεται εάν μία πρόταση είναι υποκειμενική ή όχι. Στη συνέχεια απορρίπτοντας τις αντικειμενικές προτάσεις δημιουργείται ένα απόσπασμα του κειμένου το οποίο αντιπροσωπεύει το υποκειμενικό περιεχόμενο μίας κριτικής. Αυτό το απόσπασμα διοχετεύεται σε έναν κατηγοριοποιητή πολικότητας. Οι Bo Pang και Lillian Lee[54] συνδυάζουν την ανίχνευση υποκειμενικότητας σε επίπεδο προτάσεων με την πολικότητα συναισθήματος σε επίπεδο εγγράφου.



Εικόνα 11: Αρχιτεκτονική συστήματος

Όπως με την κατηγοριοποίηση πολικότητας σε επίπεδο εγγράφου, θα μπορούσε να ανιχνευτεί η υποκειμενικότητα σε επίπεδο προτάσεων εφαρμόζοντας έναν αλγόριθμο κατηγοριοποίησης σε κάθε πρόταση ξεχωριστά. Παρόλα αυτά, μοντελοποιώντας την εγγύτητα μεταξύ των προτάσεων θα μπορούσε να ποσοτικοποιηθεί η συνάφεια: κομμάτια κειμένου τα οποία βρίσκονται κοντά μπορεί να έχουν την ίδια τιμή υποκειμενικότητας. Για το λόγο αυτό οι Bo Pang και Lillian Lee επιθυμούν να συμπεριλάβουν στον αλγόριθμό τους πληροφορία για την αλληλεπίδραση ζευγών π.χ. η διαπίστωση ότι δύο προτάσεις πρέπει να έχουν την ίδια τιμή υποκειμενικότητας χωρίς να καθορίζεται ποια είναι αυτή. Χρήση τέτοιου είδους πληροφορίας είναι αφύσικη για κατηγοριοποιητές των οποίων η είσοδος αποτελείται απλά από ανεξάρτητα διανύσματα γνωρισμάτων, όπως ο NB ή ο SVM ακριβώς γιατί αυτοί οι κατηγοριοποιητές χαρακτηρίζουν κάθε αντικείμενο ξεχωριστά. Θα μπορούσε να οριστεί ένα σύνθετο χαρακτηριστικό ή ένα διάνυσμα χαρακτηριστικών για να ξεπεραστεί το εμπόδιο αυτό. Παρόλα αυτά οι Bo Pang και Lillian Lee προτείνουν μια εναλλακτική η οποία δεν χρειάζεται τη δημιουργία τέτοιων χαρακτηριστικών. Χρησιμοποιούν μία αποτελεσματική και διαισθητική μέθοδο βασισμένη σε γράφους, η οποία βασίζεται στην εύρεση ελάχιστων τομών. Η προσέγγιση αυτή εμπνεύστηκε από τους Blum και Chawla[55], παρόλο που αυτοί εστίασαν στην ομοιότητα μεταξύ των αντικειμένων (ήθελαν να συνδυάσουν χαρακτηρισμένα και μη-χαρακτηρισμένα δεδομένα) ενώ οι Bo Pang και Lillian Lee[54] ενδιαφέρονται για την φυσική εγγύτητα μεταξύ των αντικειμένων προς κατηγοριοποίηση.

Αν υποθέσουμε ότι έχουμε n αντικείμενα x_1, x_2, x_3,..... x_n τα οποία πρέπει να χωριστούν σε δύο κλάσεις C_1 και C_2 και έχουμε πρόσβαση σε δύο τύπους πληροφοριών:

1. Ατομικές βαθμολογίες $ind_j(x_i)$: μη αρνητική εκτίμηση της προτίμησης του κάθε x_i να βρίσκεται στην κλάση C_j βασιζόμενοι μόνο στα χαρακτηριστικά του x_i

2. Βαθμολογίες συσχέτισης assoc(x_i , x_k): μη αρνητική εκτίμηση του πόσο σημαντικό είναι τα x_i και x_k να ανήκουν στην ίδια κλάση.

Στόχος είναι η μεγιστοποίηση της «ικανοποίησης δικτύου» για κάθε αντικείμενο: η ατομική του βαθμολογία για την κλάση στην οποία έχει καταχωρηθεί μείον την ατομική του βαθμολογία για την άλλη κλάση. Αλλά πρέπει να επιβληθεί βαθμολογική τιμωρία και στην περίπτωση όπου τοποθετούνται στενά συνδεδεμένα αντικείμενα σε διαφορετικές κλάσεις. Για το λόγο αυτό, τα x_i ανατίθενται στις κλάσεις C_1 και C_2 ελαχιστοποιώντας το κόστος διαχωρισμού

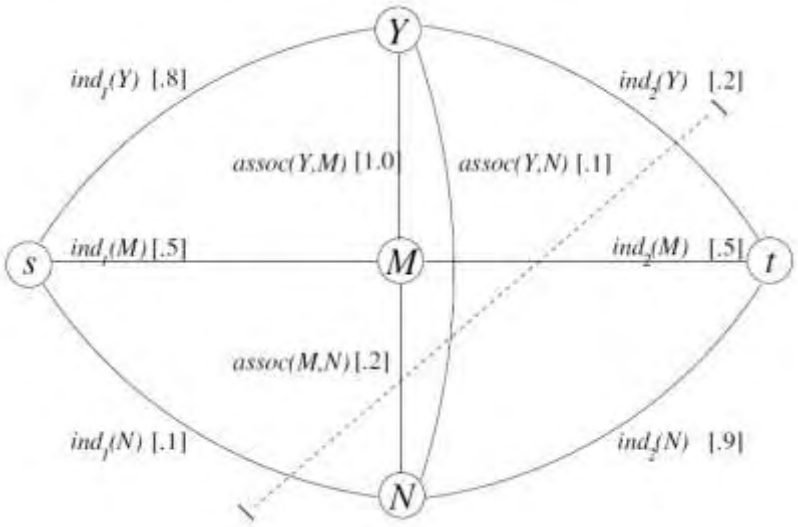
$$\sum_{x \in C_1} ind_2(x) + \sum_{x \in C_2} ind_2(x) + \sum_{\substack{x_i \in C_1 , \\ x_k \in C_2}} assoc(x_i, x_k)$$

Το πρόβλημα αυτό φαίνεται δυσεπίλυτο από τη στιγμή που υπάρχουν 2^n πιθανοί διαχωρισμοί των x_i. Για το λόγο αυτό επιλέχθηκε η αναπαράσταση του προβλήματος με τον ακόλουθο τρόπο: δημιουργούμε ένα μη κατευθυνόμενο γράφο G με κόμβους τους {u_1, u_2, ... , u_n , s , t} όπου οι δύο τελευταίοι είναι η πηγή και η καταβόθρα αντίστοιχα. Προσθέτουμε n ακμές (s , u_i) κάθε μία με βάρος $ind_1(x_i)$, και n ακμές (u_i, t) κάθε μία με βάρος $ind_2(x_i)$. Τέλος, προσθέτουμε $\binom{n}{2}$ ακμές (u_i, u_k) κάθε μία με βάρος assoc(x_i , x_k).

Οι τομές του G ορίζονται ως εξής:

Η τομή (S , T) του G είναι ένας διαχωρισμός των κόμβων του στα σύνολα S = {s} ∪ S' και T = {t} ∪ T' όπου s ∉ S' και t ∉ T'. Το κόστος της τομής cost(S, T) ισούται με το άθροισμα των βαρών όλων των ακμών που διασχίζονται από το S στο T. Μία ελάχιστη τομή του G είναι μία τομή με ελάχιστο κόστος.

Στην παρακάτω εικόνα βλέπουμε ένα γράφο για την κατηγοριοποίηση τριών αντικειμένων.



Εικόνα 12: Γράφος κατηγοριοποίησης τριών αντικειμένων

Στις αγκύλες εσωκλείονται οι τιμές του παραδείγματος. Στο συγκεκριμένο παράδειγμα οι ατομικές βαθμολογίες τυχαίνει να είναι πιθανότητες. Βασιζόμενοι στις ατομικές βαθμολογίες μπορούμε να τοποθετήσουμε το Y στην κλάση C_1 και το N στην C_2 ενώ να μείνουμε αναποφάσιστοι για το M. Αλλά οι τιμές συσχέτισης ευνοούν την τοποθέτηση του M στην ίδια κλάση με το Y όπως φαίνεται και στον παρακάτω πίνακα. Έτσι η ελάχιστη τομή που ορίζεται από τη διακεκομμένη γραμμή τοποθετεί το M μαζί με το Y στην κλάση C_1.

C_1	Individual penalties	Association penalties	Cost
{Y,M}	.2 + .5 + .1	.1 + .2	1.1
(none)	.8 + .5 + .1	0	1.4
{Y,M,N}	.2 + .5 + .9	0	1.6
{Y}	.2 + .5 + .1	1.0 + .1	1.9
{N}	.8 + .5 + .9	.1 + .2	2.5
{M}	.8 + .5 + .1	1.0 + .2	2.6
{Y,N}	.2 + .5 + .9	1.0 + .2	2.8
{M,N}	.8 + .5 + .9	1.0 + .1	3.3

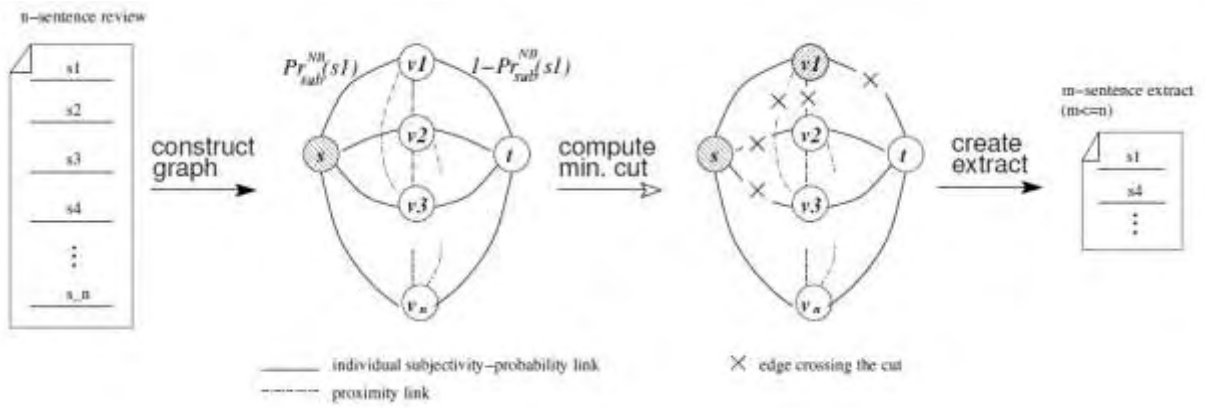
5.2 Αποτίμηση Συστήματος

Τα πειράματα που διεξήχθησαν από τους Bo Pang και Lillian Lee περιλαμβάνουν την κατηγοριοποίηση κριτικών ταινιών είτε ως θετικές είτε ως αρνητικές. Τα δεδομένα που χρησιμοποιήθηκαν περιλαμβάνουν 1000 θετικές και 1000 αρνητικές κριτικές όλες γραμμένες πριν το 2002, με μέγιστο 20 κριτικές ανά συγγραφέα (312 συγγραφείς συνολικά). Τα δεδομένα αυτά αναφέρονται ως δεδομένα πολικότητας.

Για την επιλογή του βασικού κατηγοριοποιητή πολικότητας δοκιμάστηκαν οι SVM και NB. Οι Bo Pang και Lillian Lee ακολουθώντας τους Pang et al.[2] χρησιμοποίησαν διανύσματα παρουσίας χαρακτηριστικών: το ι-οστό στοιχείο ενός διανύσματος χαρακτηριστικών είναι 1 εάν το χαρακτηριστικό αυτό εμφανίζεται στο κείμενο εισόδου και 0 σε αντίθετη περίπτωση. Κάθε βασικός κατηγοριοποιητής πολικότητας σε επίπεδο εγγράφου εκπαιδεύεται και δοκιμάζεται πάνω στο αποτέλεσμα που προκύπτει εάν εφαρμόσουμε έναν από τους ανιχνευτές υποκειμενικότητας σε επίπεδο πρότασης στα δεδομένα πολικότητας.

Για να εκπαιδευτούν οι ανιχνευτές υποκειμενικότητας είναι απαραίτητη μία συλλογή χαρακτηρισμένων προτάσεων. Οι Riloff και Wiebe[56] είχαν δηλώσει ότι «είναι πολύ δύσκολο να εξασφαλίσει κανείς ένα σύνολο προτάσεων που μπορούν εύκολα να χαρακτηριστούν ως υποκειμενικές ή αντικειμενικές». Οι Bo Pang και Lillian Lee για να

λύσουν αυτό το πρόβλημα χρησιμοποίησαν το διαδίκτυο για την δημιουργία ενός μεγάλου συνόλου αυτόματα χαρακτηρισμένων προτάσεων. Για τη συλλογή υποκειμενικών προτάσεων, συνέλεξαν 5000 σύντομες προτάσεις από τα σχόλια που αφήνουν οι χρήστες του www.rottentomatoes.com . Ενώ για τη συλλογή των αντικειμενικών δεδομένων επέλεξαν 5000 προτάσεις από περιλήψεις ταινιών που βρήκαν διαθέσιμες στο www.imdb.com . Επέλεξαν προτάσεις ή αποσπάσματα μήκους τουλάχιστον δέκα λέξεων τα οποία είχαν δημοσιευτεί μετά το 2001 για την αποφυγή της επικάλυψης των συνόλων των δεδομένων πολικότητας και των δεδομένων υποκειμενικότητας.
Ως ανιχνευτής υποκειμενικότητας μπορεί να χρησιμοποιηθεί ένας κατηγοριοποιητής πολικότητας εκπαιδευμένος με τα δεδομένα υποκειμενικότητας για την εξαγωγή του υποκειμενικού αποσπάσματος του αρχικού κειμένου. Οι Bo Pang και Lillian Lee ακόμη δημιούργησαν μία οικογένεια ανιχνευτών υποκειμενικότητας οι οποίοι βασίζονται στις ελάχιστες τομές. Οι ανιχνευτές αυτοί παίρνουν ως είσοδο το σύνολο των προτάσεων που εμφανίζονται σε ένα κείμενο και προσδιορίζουν την κατάσταση υποκειμενικότητάς τους χρησιμοποιώντας και την ατομική πληροφορία και την πληροφορία συσχέτισης. Ειδικότερα, για ένα δεδομένο κείμενο, χρησιμοποιούμαι την τεχνική που παρουσιάστηκε παραπάνω για την δημιουργία ενός γράφου όπου η πηγή s αντιστοιχεί στην κλάση των υποκειμενικών προτάσεων ενώ η καταβόθρα t αντιστοιχεί στην κλάση των αντικειμενικών προτάσεων και κάθε εσωτερικός κόμβος u_i αντιστοιχεί στην i-οστή πρόταση του κειμένου. Οι ατομικές βαθμολογίες $ind_1(s_i)$ παίρνουν την τιμή $Pr_{sub}^{NB}(s_i)$ και οι $ind_2(s_i)$ την τιμή $1 - Pr_{sub}^{NB}(s_i)$ όπως φαίνεται και στην παρακάτω εικόνα, όπου $Pr_{sub}^{NB}(s_i)$ είναι η Naïve Bayes τιμή της πιθανότητας η πρόταση s να είναι υποκειμενική, ή μπορεί να χρησιμοποιηθεί η τιμή που προκύπτει από τον SVM κατηγοριοποιητή.



Εικόνα 13: Εξαγωγή υποκειμενικότητας χρησιμοποιώντας την μέθοδο των ελάχιστων τομών

Εάν οι βαθμολογίες συσχέτισης τεθούν όλες στο μηδέν τότε η κατηγοριοποίηση ελάχιστων τομών των προτάσεων είναι ίδια με εκείνη ενός βασικού ανιχνευτή υποκειμενικότητας. Εναλλακτικά, ενσωματώνεται ο βαθμός εγγύτητας μεταξύ ζευγών προτάσεων και προσδιορίζεται από τρεις παραμέτρους.

1. Το κατώφλι T ορίζει την μέγιστη απόσταση που μπορεί να χωρίζει δύο προτάσεις και να θεωρούνται ακόμη κοντινές.

2. Η μη αύξουσα συνάρτηση f(d) καθορίζει πως η επιρροή των εγγείων προτάσεων εξασθενεί σε συνάρτηση με την απόσταση d.
3. Η σταθερά c ελέγχει την επιρροή των βαθμολογιών συσχέτισης. Μία μεγάλη τιμή στην σταθερά c κάνει τον αλγόριθμο ελάχιστων τομών αυστηρό στην τοποθέτηση συσχετιζόμενων προτάσεων σε διαφορετικές κλάσεις.

Με αυτά τα δεδομένα έχει οριστεί ο παρακάτω τύπος:

$$\mathrm{assoc}(s_i, s_j) \stackrel{\text{def}}{=} \begin{cases} f(j-i) \times c & if\ (j-i) \le T \\ 0 & \text{otherwise} \end{cases}$$

5.3 Αποτελέσματα

Παρακάτω παρουσιάζονται οι μέσες τιμές ακριβείας τις οποίες υπολόγισαν οι Bo Pang και Lillian Lee μετά από επαναλαμβανόμενες επιβεβαιώσεις με τη χρήση των δεδομένων πολικότητας. Στις επόμενες δύο ενότητες εξετάζονται αρχικά οι βασικοί αλγόριθμοι εξαγωγής υποκειμενικότητας, οι οποίοι βασίζονται αποκλειστικά στις προβλέψεις μεμονωμένων προτάσεων, και στη συνέχεια αξιολογείται η πιο εξεζητημένη μορφή εξαγωγής υποκειμενικότητας η οποία χρησιμοποιεί την πληροφορία των συμφραζόμενων μέσω του παραδείγματος των ελάχιστων τομών.

Όπως θα δούμε η χρήση των υποκειμενικών αποσπασμάτων μπορεί στην καλύτερη περίπτωση να προσφέρει ικανοποιητική βελτίωση στην κατηγοριοποίηση πολικότητας, ενώ μπορεί τουλάχιστον να αποφέρει ακρίβεια ίδια με αυτή που προκύπτει από την κατηγοριοποίηση με την χρήση του κειμένου της κριτικής. Την ίδια στιγμή, τα αποσπάσματα που δημιουργούνται είναι κατά κύριο λόγο μικρότερα του αρχικού κειμένου και συγχρόνως πιο αποτελεσματικά σαν είσοδος σε ένα κατηγοριοποιητή πολικότητας από τα αποσπάσματα που προκύπτουν από κλασικές μεθόδους δημιουργίας περιλήψεων.

Βασική εξαγωγή υποκειμενικότητας

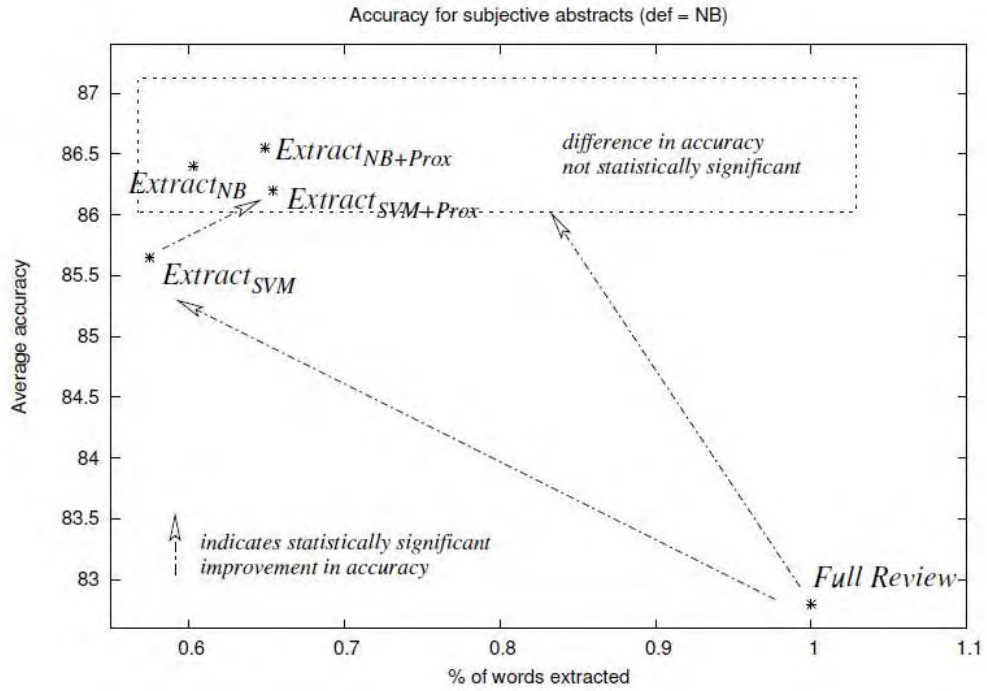
Όπως σημειώθηκε και παραπάνω και ένας NB και ένας SVM κατηγοριοποιητής μπορεί να εκπαιδευτεί με τα δεδομένα υποκειμενικότητας και στη συνέχεια να χρησιμοποιηθεί σαν βασικός ανιχνευτής υποκειμενικότητας. Ο NB κατηγοριοποιητής παρουσιάζει καλύτερη απόδοση όταν εκπαιδευτεί με τα δεδομένα υποκειμενικότητας (92% όταν ο SVM παρουσιάζει 90% ακρίβεια) για το λόγο αυτό η συζήτηση περιορίζεται στα αποτελέσματα του NB ανιχνευτή υποκειμενικότητας.

Χρησιμοποιώντας τον Naïve Bayes ως ανιχνευτή υποκειμενικότητας ($\mathrm{Extract}_{\mathrm{NB}}$) σε συνδυασμό με ένα Naïve Bayes κατηγοριοποιητή πολικότητας επιπέδου εγγράφου πετυχαίνουμε 86,4% ακρίβεια. Αυτή είναι μία καθαρή βελτίωση του 82,8% το οποίο

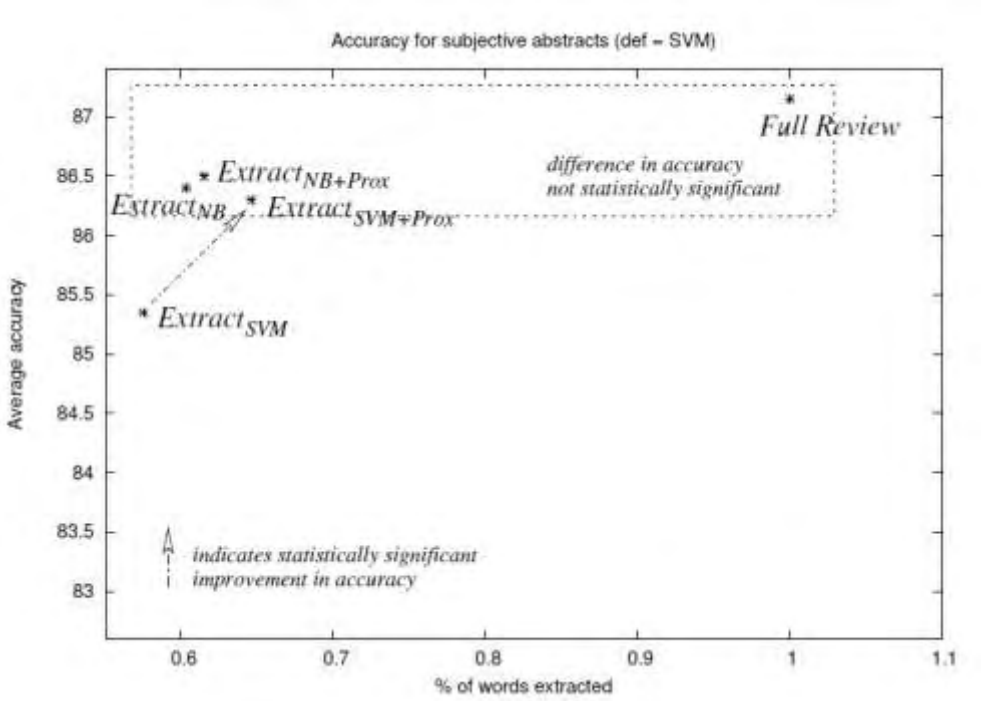
επιτυγχάνεται όταν δεν εφαρμοστεί ανίχνευση υποκειμενικότητας στο αρχικό κείμενο (full review). Αντίθετα με ένα SVM κατηγοριοποιητή πολικότητας η επίδοση πλήρους κειμένου (full review) ανέρχεται στο 87,15%, ενώ όταν στον κατηγοριοποιητή αυτό εφαρμοστεί το απόσπασμα που προκύπτει από τον NB ανιχνευτή υποκειμενικότητας το ποσοστό αυτό γίνεται 86,4%. Η διαφορά των δύο ποσοστών είναι στατιστικά δυσδιάκριτη.

Τα ευρήματα αυτά δείχνουν πως τα αποσπάσματα των κειμένων διατηρούν (και στην περίπτωση του NB κατηγοριοποιητή πολικότητας αποσαφηνίζουν) την πληροφορία συναισθήματος του κειμένου από το οποίο προέρχονται, συνεπώς είναι καλές περιλήψεις από την οπτική της κατηγοριοποίησης συναισθήματος. Περαιτέρω υποστήριξη προκύπτει από ένα αντίστροφο πείραμα: εάν δοθεί ως είσοδος, στον βασικό κατηγοριοποιητή πολικότητας ένα απόσπασμα το οποίο αποτελείται από προτάσεις οι οποίες έχουν χαρακτηριστεί ως αντικειμενικές, η ακρίβεια πέφτει δραματικά στο 71% για τον NB και στο 67% για τον SVM. Το γεγονός αυτό επιβεβαιώνει την υπόθεση των Bo Pang και Lillian Lee ότι οι προτάσεις που έχουν απορριφθεί από την διαδικασία της εξαγωγής υποκειμενικότητας είναι λιγότερο ενδεικτικές της πολικότητας συναισθήματος.

Ακόμη, τα υποκειμενικά αποσπάσματα είναι πολύ πιο συμπαγή από τα αρχικά έγγραφα: περιέχουν κατά μέσο όρο το 60% των λέξεων της αρχικής κριτικής (το ποσοστό διατήρησης των λέξεων του αρχικού κειμένου φαίνεται στον άξονα x των δύο παρακάτω διαγραμμάτων). Το γεγονός αυτό ώθησε τους Bo Pang και Lillian Lee να μελετήσουν πόση μείωση μπορούν να υποστούν τα αρχικά έγγραφα από τους ανιχνευτές υποκειμενικότητας και ακόμη να αναπαριστούν με ακρίβεια την πληροφορία συναισθήματος του κειμένου.



Εικόνα 14: Ποσοστό διατήρησης λέξεων σε συνάρτηση με την ακρίβεια, χρησιμοποιώντας έναν NB κατηγοριοποιητή πολικότητας



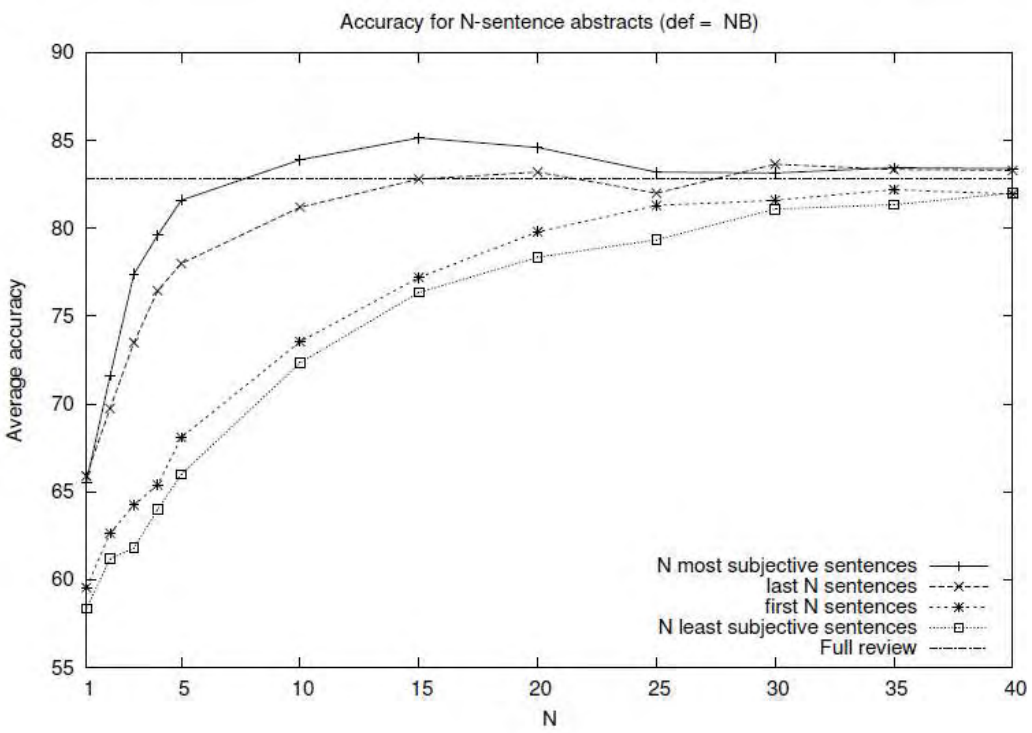
Εικόνα 15: Ποσοστό διατήρησης λέξεων σε συνάρτηση με την ακρίβεια, χρησιμοποιώντας έναν SVM κατηγοριοποιητή πολικότητας.

Υπάρχουν διάφοροι τρόποι για τη δημιουργία αποσπασμάτων με διαφορετικά μήκη.

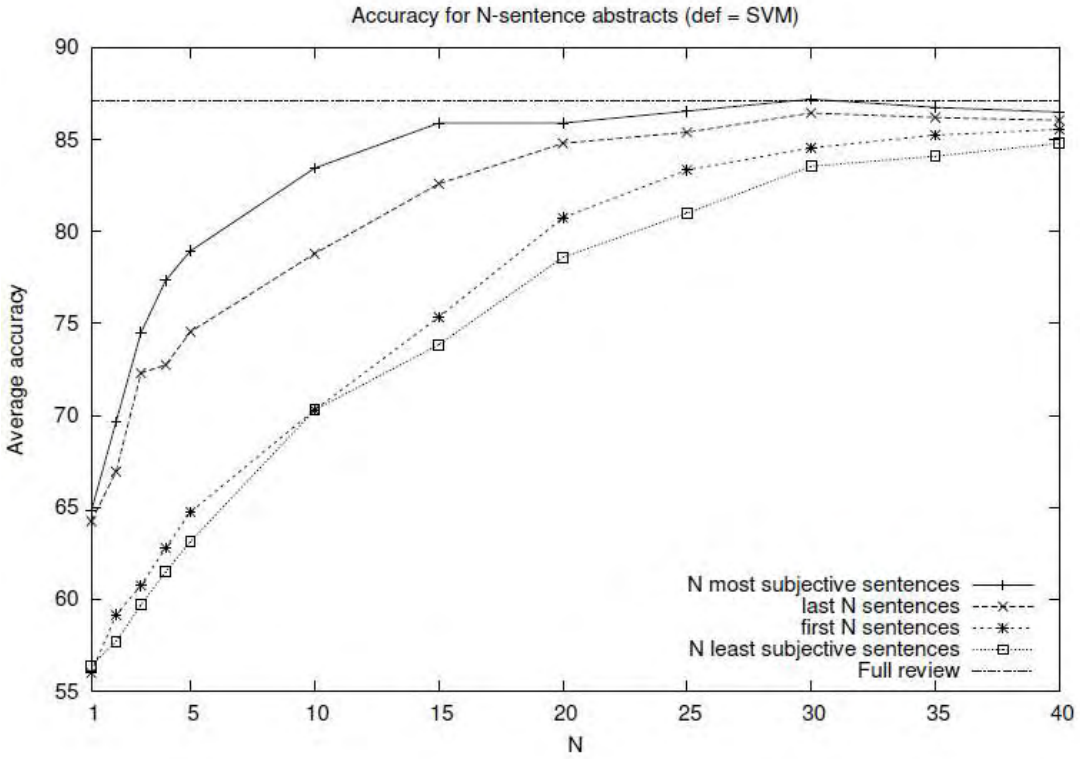
- Επιλέγοντας τις N ποιο υποκειμενικές προτάσεις
- Επιλέγοντας τις N πρώτες προτάσεις της κριτικής. Οι συγγραφείς συνήθως ξεκινούν την κριτική τους με μία περίληψη αυτής.
- Επιλέγοντας τις N τελευταίες προτάσεις. Σε πολλά έγγραφα ο επίλογος μπορεί να είναι μία καλή περίληψη.
- Επιλέγοντας τις N λιγότερο υποκειμενικές προτάσεις.

Για κριτικές που αποτελούνται από λιγότερες από N προτάσεις επιστρέφεται ολόκληρη η κριτική.

Στα παρακάτω διαγράμματα συγκρίνονται οι τέσσερεις μέθοδοι δημιουργίας περιλήψεων, ενώ σαν βάση σύγκρισης χρησιμοποιείται η ακρίβεια που προσφέρει το σύνολο της κριτικής. Τα διαγράμματα δείχνουν τα αποτελέσματα του κατηγοριοποιητή πολικότητας καθώς το N κυμαίνεται μεταξύ του 1 και του 40.



Εικόνα 16: Ακρίβεια χρησιμοποιώντας αποσπάσματα N προτάσεων για έναν NB κατηγοριοποιητή πολικότητας



Εικόνα 17: Ακρίβεια χρησιμοποιώντας αποσπάσματα N προτάσεων για έναν SVM κατηγοριοποιητή πολικότητας

Η πρώτη παρατήρηση είναι ότι ο NB ανιχνευτής έχει πολύ καλή απόδοση. Με το υποκειμενικό απόσπασμα να αποτελείται από 15 προτάσεις, η ακρίβεια είναι πολύ κοντά σε αυτή που επιτυγχάνεται όταν χρησιμοποιείται το σύνολο του κειμένου. Για τον NB κατηγοριοποιητή πολικότητας η χρήση των 5 πιο υποκειμενικών προτάσεων είναι σχεδόν το ίδιο διαφωτιστική όσο η χρήση ολόκληρης της κριτικής ενώ περιέχει κατά μέσο όρο μόνο το 22% των λέξεων των αρχικών κειμένων. Ακόμη, όταν N=30 η επίδοση είναι ελαφρώς καλύτερη από εκείνη που επιτυγχάνεται με τη χρήση ολόκληρου του κειμένου (full review) ακόμη και όταν χρησιμοποιείται ο SVM κατηγοριοποιητής πολικότητας (87,2% έναντι 87,15%).

Όπως φαίνεται στα δύο παραπάνω γραφήματα η μέθοδος των N πιο υποκειμενικών προτάσεων γενικά έχει καλύτερες επιδόσεις από τις λοιπές μεθόδους εξαγωγής περιλήψεων. Είναι ακόμη ενδιαφέρον να παρατηρήσουμε πόσο καλύτερα αποτελέσματα φέρουν οι N τελευταίες προτάσεις από τις N πρώτες προτάσεις. Το γεγονός αυτό αντανακλά την τάση των συγγραφέων κριτικών για ταινίες να τοποθετούν μία μικρή περίληψη της πλοκής των ταινιών στην αρχή των κειμένων τους αντί για το τέλος και να καταλήγουν με δηλώσεις που δείχνουν φανερά την άποψη τους.

Ενσωματώνοντας την πληροφορία συμφραζομένων

Στην προηγούμενη ενότητα μελετήθηκε η αξία της ανίχνευσης υποκειμενικότητας. Τώρα θα εξετάσουμε κατά πόσο η πληροφορία των συμφραζομένων, πιο συγκεκριμένα η εγγύτητα των προτάσεων, μπορεί να βελτιώσει περαιτέρω την εξαγωγή υποκειμενικότητας. Όπως συζητήθηκε και παραπάνω, οι περιορισμοί που τίθενται από τα συμφραζόμενα μπορούν εύκολα να ενσωματωθούν μέσο της μεθόδου των ελάχιστων τομών αλλά δεν είναι φυσικές είσοδοι για τους βασικούς κατηγοριοποιητές Naïve Bayes και SVM.

Οι εικόνες 14 έως 17 δείχνουν την επίδραση που έχει η ενσωμάτωση της πληροφορίας των συμφραζομένων. $\text{Extract}_{\text{NB+Prox}}$ και $\text{Extract}_{\text{SVM+Prox}}$ είναι οι ανιχνευτές υποκειμενικότητας οι οποίοι βασίζονται στη μέθοδο των ελάχιστων τομών και χρησιμοποιούν τον NB και SVM κατηγοριοποιητή αντίστοιχα για τις ατομικές βαθμολογίες. Οι Bo Pang και Lillian Lee εστιάζουν τη μελέτη τους στη σύγκριση μεταξύ των $\text{Extract}_{\text{NB+Prox}}$ και $\text{Extract}_{\text{NB}}$ και μεταξύ των $\text{Extract}_{\text{SVM+Prox}}$ και $\text{Extract}_{\text{SVM}}$.

Παρατηρούμε στις εικόνες 4 και 5 ότι οι ανιχνευτές υποκειμενικότητας που λαμβάνουν υπ' όψιν τα συμφραζόμενα και βασίζονται σε γράφους τείνουν να δημιουργούν εξαγόμενα τα οποία περιέχουν περισσότερη πληροφορία, παρόλο που τα εξαγόμενα αυτά είναι μεγαλύτερα από αυτά που προκύπτουν από μεθόδους που δεν λαμβάνουν υπ' όψιν τα συμφραζόμενα. Σημειώνεται πως η βελτίωση της απόδοσης δεν μπορεί να αποδοθεί αποκλειστικά στην απλή ενσωμάτωση περισσότερων προτάσεων ανεξαρτήτως εάν αυτές είναι υποκειμενικές ή όχι, (ένα αντεπιχείρημα είναι το γεγονός ότι το σύνολο της κριτικής απέδωσε αρκετά χειρότερα αποτελέσματα στον NB βασικό κατηγοριοποιητή) και σε κάθε περίπτωση, τα εξαγόμενα που προέκυψαν από γράφους είναι αρκετά πιο συνοπτικά από το σύνολο του κειμένου.

Παρόλο που χρησιμοποιούμε μία τακτική για την ανάθεση κοντινών προτάσεων στην ίδια κατηγορία, στους NB και SVM ανιχνευτές υποκειμενικότητας φαίνεται ότι είναι απαραίτητη μία όχι προφανής μέθοδος. Ακόμη, οι Bo Pang και Lillian Lee επιθυμούσαν να εξετάσουν κατά πόσο το παράδειγμα που είναι βασισμένο στο γράφο κάνει καλύτερη χρήση των συμφραζομένων από μία εύκολη μέθοδο η οποία κωδικοποιεί την πληροφορία αυτή στην είσοδο του κατηγοριοποιητή. Για το λόγο αυτό μελέτησαν την πληροφορία αλλαγής παραγράφου την οποία συνδύασαν μόνο με τον SVM ανιχνευτή υποκειμενικότητας για λόγους ευκολίας.

Φαίνεται λογικό ότι τα όρια των παραγράφων χαλαρώνουν τους περιορισμούς συνάφειας μεταξύ των κοντινών προτάσεων. Για να συμπεριληφθεί αυτή η έννοια στην κατηγοριοποίηση μέσω ελάχιστων τομών, μπορούμε απλά να μειώσουμε τις βαθμολογίες συσχέτισης για όλα τα ζεύγη προτάσεων οι οποίες βρίσκονται σε διαφορετικές παραγράφους πολλαπλασιάζοντας αυτές με ένα βάρους $w \in [0,1]$. Για τους βασικούς κατηγοριοποιητές, μπορούμε να χρησιμοποιήσουμε το τέχνασμα ο ανιχνευτής να διαχειρίζεται παραγράφους αντί για προτάσεις σαν τη βασική ενότητα για χαρακτηρισμό. Αυτό δίνει τη δυνατότητα στους βασικούς κατηγοριοποιητές να χρησιμοποιήσουν τη συνάφεια μεταξύ προτάσεων της ίδιας παραγράφου, από την άλλη τίθεται ένας σοβαρός περιορισμός, όλες οι προτάσεις μίας παραγράφου λαμβάνουν τον ίδιο χαρακτηρισμό το οποίο αυξάνει σημαντικά το θόρυβο. Τα πειράματα των Bo Pang και Lillian Lee αποκαλύπτουν ότι η μέθοδος τομών βασισμένη σε γράφους είναι η βέλτιστη προσέγγιση και για τους δύο βασικούς κατηγοριοποιητές πολικότητας (NB και SVM) κάποια επιλογή παραμέτρων (συμπεριλαμβανομένου και του w) για το $\text{Extract}_{\text{SVM+Prox}}$ αποφέρει σημαντική στατιστική βελτίωση έναντι της μεθόδου που διαχειρίζεται κάθε παράγραφο σαν μία μονάδα χωρίς τη χρήση γράφου (NB: 86,4% έναντι 85,2% , SVM: 86,15% έναντι 85,45%).

6 Rome

6.1 Πως ξεκίνησε η ανάπτυξη του Rome

Το 2004 είχαν αναπτυχθεί πολλές εκδόσεις των RSS και Atom formats. Στην Sun ξεκίνησαν διάφορα projects τα οποία περιλάμβαναν αυτά τα Syndication formats, αλλά δεν υπήρχαν ικανοποιητικές βιβλιοθήκες στη Java για το χειρισμό και δημιουργία RSS. Ο στόχος των ανθρώπων της Sun ήταν να ξεπεράσουν (ESCAPE) αυτό το πρόβλημα. Για να επιτευχθεί αυτό η βιβλιοθήκη έπρεπε να είναι:

- **E**asy to use
- **S**imple
- **C**omplete
- **A**bstract
- **P**owerful
- **E**xtensible

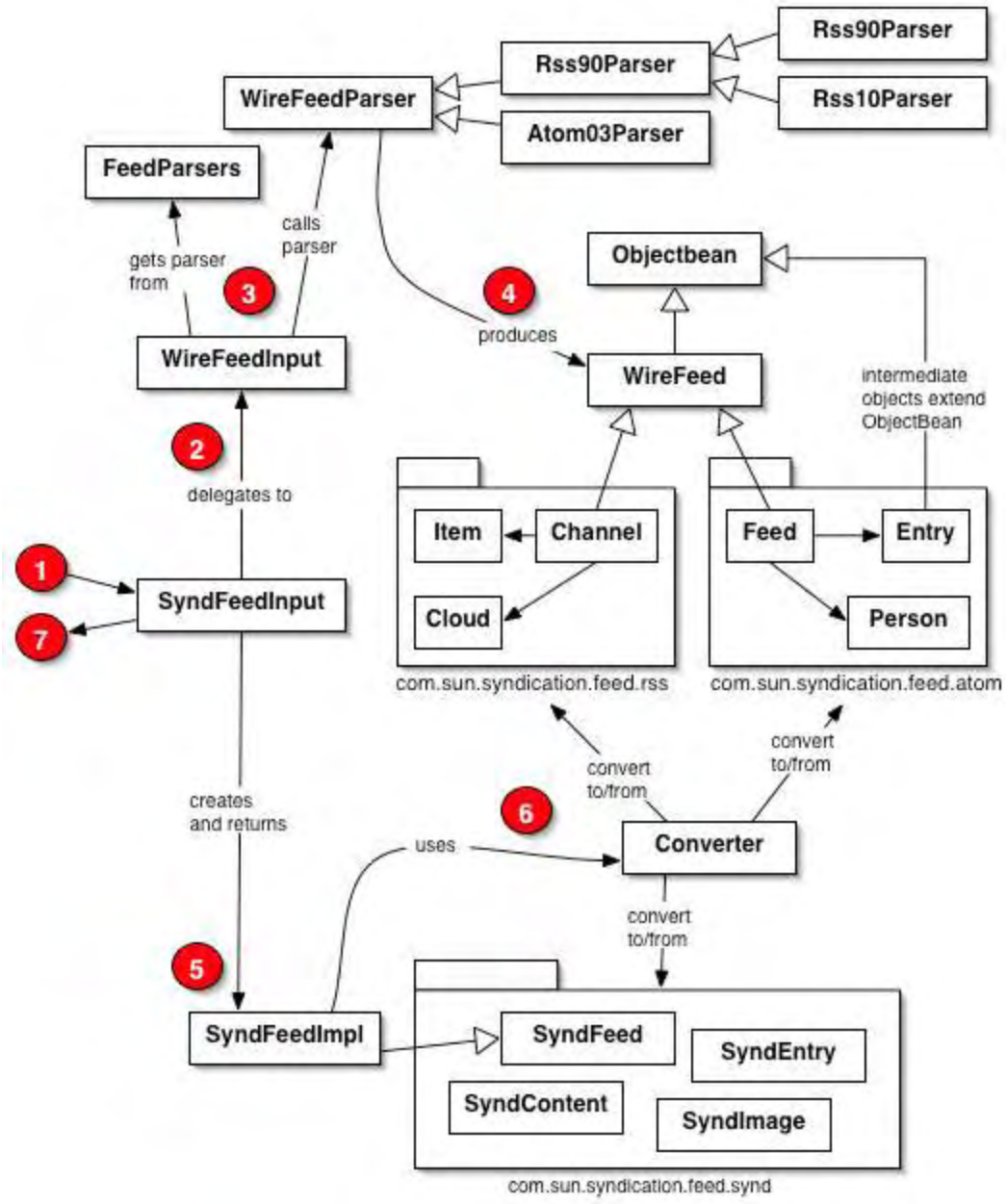
6.2 Πως λειτουργεί το Rome

Το Rome βασίζεται σε ένα εξιδανικευμένο και αφηρημένο μοντέλο τροφοδότη νέων (Newsfeed) ή τροφοδότη δημοσίευσης περιεχομένου (Syndication Feed). Το Rome μπορεί να προσαρμόσει στο μοντέλο αυτό κάθε format τροφοδότη νέων συμπεριλαμβανομένων όλων των εκδόσεων του RSS και του Atom. Το Rome μπορεί να μετατρέψει μία αναπαράσταση αυτού του μοντέλου σε οποιοδήποτε άλλο format τροφοδότη νέων.

Εσωτερικά το Rome ορίζει ενδιάμεσα μοντέλα αντικειμένων (Wirefeed formats) ένα για κάθε ξεχωριστό Newsfeed format, συμπεριλαμβανομένων όλων των εκδόσεων του RSS και του Atom. Για κάθε format υπάρχει μία ξεχωριστή JDOM κλάση η οποία μετατρέπει ένα XML σε ένα ενδιάμεσο μοντέλο. Το Rome προσφέρει «μετατροπείς» για τη μετατροπή μεταξύ των ενδιάμεσων Wirefeed μοντέλων και του εξιδανικευμένου μοντέλου Syndication feed.

Το Rome δεν προσπαθεί να διορθώσει τα XML αρχεία. Εάν ένα Newsfeed είναι ένα μη έγκυρο XML τότε το Rome θα αποτύχει.

Στην παρακάτω εικόνα βλέπουμε τι συμβαίνει κατά την επεξεργασία ενός Newsfeed από το Rome.



Εικόνα 18: Βήματα επεξεργασίας ενός Newsfeed από το Rome 0.4

1. Ο κώδικας καλεί την κλάση SyndFeedInput για να χειριστεί το Newsfeed, για παράδειγμα:

```
URL feedUrl = new URL("file:blogging-roller.rss");
SyndFeedInput input = new SyndFeedInput();
SyndFeed feed = input.build(new
InputStreamReader(feedUrl.openStream()));
```

2. Η SyndFeedInput καλεί την WireFeedInput να κάνει τον πραγματικό χειρισμό του feed

3. Η WireFeedInput χρησιμοποιεί έναν PluginManager της κλάσης FeedParsers για την επιλογή του σωστού parser για το χειρισμό του feed και στη συνέχεια καλείται αυτός ο parser για να επεξεργαστεί το Newsfeed.

4. Ο κατάλληλος parser μετατρέπει το Newsfeed, χρησιμοποιώντας το JDom, σε ένα WireFeed. Εάν το Newsfeed είναι σε RSS format το WireFeed αντικείμενο είναι της κλάσης Channel και περιέχει Items, Clouds και άλλα RSS στοιχεία από το πακέτο com.sun.syndication.feed.rss. Στην άλλη περίπτωση, εάν το Newsfeed είναι σε Atom format, τότε το WireFeed είναι αντικείμενο της κλάσης Feed του πακέτου com.sun.syndication.atom. Τελικά το αντικείμενο WireFeedInput επιστρέφει ένα WireFeed.

5. Το SyndFeedInput αντικείμενο χρησιμοποιεί το επιστρεφόμενο WireFeedInput αντικείμενο για να δημιουργήσει ένα SyndFeedImpl το οποίο υλοποιεί ένα SyndFeed. SyndFeed είναι μία διασύνδεση, η ρίζα μίας αφαίρεσης η οποία αποτελεί ένα Newsfeed ανεξάρτητου format.

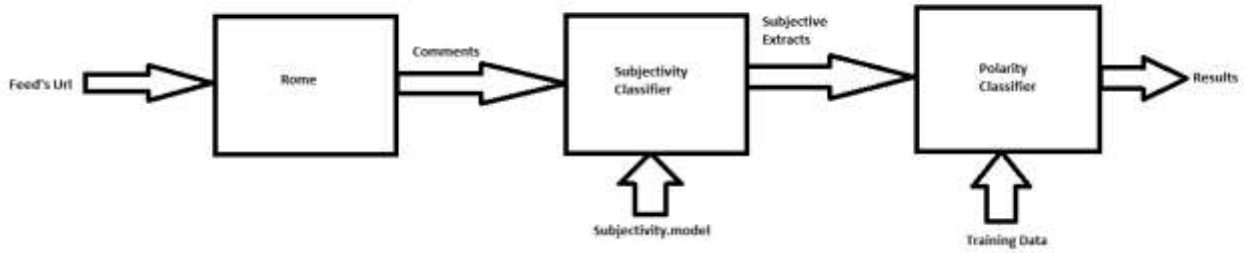
6. Το SyndFeedImpl χρησιμοποιεί έναν Converter για τη μετατροπή μεταξύ του WireFeed το οποίο έχει ένα συγκεκριμένο format και του SyndFeed που είναι ανεξάρτητο format.

7. Το SyndFeedInput επιστρέφει ένα SyndFeed το οποίο περιέχει το επεξεργασμένο Newsfeed.

Το Rome ακόμη υποστηρίζει και τη δημιουργία Newsfeed και για κάθε format προσφέρει μία “generator” κλάση η οποία λαμβάνει ένα μοντέλο Syndication feed και παράγει από αυτό ένα Newsfeed XML.[58]

7 Περιγραφή Συστήματος

7.1 Αρχιτεκτονική Συστήματος



Αρχικά παρέχεται στο σύστημά μας η διεύθυνση Url ενός RSS ή ATOM feed. Η διεύθυνση αυτή παρέχεται στον κώδικα για τη χρήση του Rome. Το Rome εξάγει τις κριτικές που βρίσκονται στο feed. Τα σχόλια αυτά, τα οποία έχουν τοποθετηθεί σε μία λίστα, επεξεργάζονται από τον κατηγοριοποιητή υποκειμενικότητας και το υποκειμενικό κομμάτι τους τροφοδοτείται ως είσοδος στον κατηγοριοποιητή πολικότητας. Ο κατηγοριοποιητής υποκειμενικότητας δεν χρειάζεται να περάσει την φάση της εκπαίδευσης γιατί ο κατηγοριοποιητής διαβάζει το μοντέλο κατηγοριοποίησης από το αρχείο subjectivity.model. Μόλις ξεκινήσει η εκτέλεση του κώδικα του κατηγοριοποιητή πολικότητας γίνεται η εκπαίδευσή του με τα δεδομένα εκπαίδευσης και στη συνέχεια κατηγοριοποιούνται τα υποκειμενικά αποσπάσματα που έχει εξάγει ο κατηγοριοποιητής υποκειμενικότητας. Μόλις ολοκληρωθεί η διαδικασία αυτή επιστρέφονται τα αποτελέσματά της, δηλαδή ο αριθμός των θετικών και αρνητικών σχολίων.

7.2 Κώδικας για τη χρήση του LingPipe

SubjectivityBasic.java

Αρχικά τρέχουμε την κλάση SubjectivityBasic.java η οποία δημιουργεί έναν κατηγοριοποιητή υποκειμενικότητας τον οποίο και αποθηκεύει σε ένα αρχείο για μεταγενέστερη χρήση.

Ο κώδικας για τη δημιουργία του κατηγοριοποιητή υποκειμενικότητας φαίνεται παρακάτω:

```
public class SubjectivityBasic {

    File mPolarityDir;
    String[] mCategories;
    DynamicLMClassifier<NGramProcessLM> mClassifier;

    SubjectivityBasic(String[] args) {
        System.out.println("\nBASIC SUBJECTIVITY DEMO");
        mPolarityDir = new File("polarity_dir");
        System.out.println("\nData Directory=" + mPolarityDir);
        mCategories = new String[] { "plot", "quote" };
        int nGram = 8;
        mClassifier =
            DynamicLMClassifier
            .createNGramProcess(mCategories,nGram);
    }

    void run() throws ClassNotFoundException, IOException {
        train();
        evaluate();
    }

    void train() throws IOException {
        int numTrainingChars = 0;
        System.out.println("\nTraining.");
        for (int i = 0; i < mCategories.length; ++i) {
            String category = mCategories[i];
            Classification classification
                = new Classification(category);
            File file = new File(mPolarityDir,
                                 mCategories[i] + ".tok.gt9.5000");
            String data = Files.readFromFile(file,"ISO-8859-1");
            String[] sentences = data.split("\n");
            System.out.println("# Sentences " + category + "=" +
sentences.length);
            int numTraining = (sentences.length * 9) / 10;
            for (int j = 0; j < numTraining; ++j) {
                String sentence = sentences[j];
                numTrainingChars += sentence.length();
                Classified<CharSequence> classified
                    = new Classified<CharSequence>(sentence,classification);
                mClassifier.handle(classified);
            }
        }

        System.out.println("\nCompiling.\n  Model file=subjectivity.model");
        FileOutputStream fileOut = new
FileOutputStream("subjectivity.model");
        ObjectOutputStream objOut = new ObjectOutputStream(fileOut);
        mClassifier.compileTo(objOut);
        objOut.close();

        System.out.println("  # Training Cases=" + 9000);
        System.out.println("  # Training Chars=" + numTrainingChars);
    }
```

```
void evaluate() throws IOException {
        // classifier hasn't been compiled, so it'll be slower
        boolean storeInputs = false;
        JointClassifierEvaluator<CharSequence> evaluator
            = new JointClassifierEvaluator<CharSequence>(mClassifier,
mCategories,storeInputs);
        System.out.println("\nEvaluating.");
        for (int i = 0; i < mCategories.length; ++i) {
            String category = mCategories[i];
            Classification classification
                = new Classification(category);
            File file = new File(mPolarityDir,
                                 mCategories[i] + ".tok.gt9.5000");
            String data = Files.readFromFile(file,"ISO-8859-1");
            String[] sentences = data.split("\n");
            int numTraining = (sentences.length * 9) / 10;
            for (int j = numTraining; j < sentences.length; ++j) {
                Classified<CharSequence> classified
                    = new
Classified<CharSequence>(sentences[j],classification);
                evaluator.handle(classified);
            }
        }
        System.out.println();
        System.out.println(evaluator.toString());
    }

    public static void main(String[] args) {
        try {
            new SubjectivityBasic(args).run();
        } catch (Throwable t) {
            System.out.println("Thrown: " + t);
            t.printStackTrace(System.out);
        }
    }
}
```

Το πρόγραμμα αυτό διαβάζει τα δεδομένα εκπαίδευσης τα οποία βρίσκονται στο φάκελο polarity_dir, εκπαιδεύει τον κατηγοριοποιητή και στη συνέχεια τον αξιολογεί. Η μέθοδος main αρχικά κατασκευάζει ένα στιγμιότυπο της κλάσης και στη συνέχεια το τρέχει.

Ο constructor αρχικοποιεί τις μεταβλητές:

```
File mPolarityDir;
String[] mCategories;
DynamicLMClassifier<NGramProcessLM> mClassifier;


SubjectivityBasic() {
        System.out.println("\nBASIC SUBJECTIVITY DEMO");
        mPolarityDir = new File("polarity_dir");
        System.out.println("\nData Directory=" + mPolarityDir);
        mCategories = new String[] { "plot", "quote" };
        int nGram = 8;
```

```
        mClassifier =
            DynamicLMClassifier
            .createNGramProcess(mCategories,nGram);
    }
```

Ο φάκελος mPolarityDir αρχικοποιείται στη θέση polarity_dir σε σχέση με τη θέση στην οποία έχει τοποθετηθεί ο κώδικας στο σύστημα αρχείων. Ο πίνακας mCategories αρχικοποιείται με τις τιμές “plot” και “quote” οι οποίες είναι για τις αντικειμενικές και υποκειμενικές κριτικές αντίστοιχα. Το μήκος του n-gram ορίζεται στο 8 και στη συνέχεια δημιουργείται ο κατηγοριοποιητής με τις κατηγορίες που ορίστηκαν παραπάνω και μήκος διανύσματος ίσο με 8.

Η συνάρτηση train() διατρέχει τις κατηγορίες, οι οποίες στη συγκεκριμένη περίπτωση είναι δύο, και ανοίγει κάθε φορά το αρχείο που περιέχει τις προτάσεις για την συγκεκριμένη κατηγορία. Διαβάζει το σύνολο των δεδομένων του αρχείου και στη συνέχεια διαχωρίζει τις προτάσεις τις οποίες και αποθηκεύει στον πίνακα sentences. Οι προτάσεις αυτές χρησιμοποιούνται για την εκπαίδευση του κατηγοριοποιητή. Από τους δημιουργούς του LingPipe έχει οριστεί ότι μόνο το 90% των προτάσεων θα χρησιμοποιηθεί για την εκπαίδευση του κατηγοριοποιητή.

Ο υπόλοιπος κώδικας της μεθόδου train() μεταγλωττίζει το μοντέλο στο αρχείο subjectivity.model

Με γκρι σημειώνεται ο κώδικας της μεθόδου evaluate() ο οποίος παραμένει ο ίδιος με αυτόν της μεθόδου train() που περιγράψαμε παραπάνω.

```
void evaluate() throws IOException {
    BaseClassifierEvaluator<String> evaluator
        = new BaseClassifierEvaluator<String>(mClassifier,
                                                    mCategories);
    for (int i = 0;
         i < mCategories.length; ++i) {
        String category = mCategories[i];
        Classification classification
            = new Classification(category);
        File file = new File(mPolarityDir,
                             mCategories[i]
                             + ".tok.gt9.5000");
        String data = Files.readFromFile(file,"ISO-8859-1");
        String[] sentences = data.split("\n");
        int numTraining = (sentences.length * 9) / 10;
        for (int j = numTraining;
             j < sentences.length; ++j) {
            Classified<CharSequence> classified
                = new Classified<CharSequence>(sentences[j],classification);
            evaluator.handle(classified);
        }
    }
    System.out.println(evaluator.toString());
}
```

Στην πρώτη γραμμή δημιουργείται ένας evaluator από τον κατηγοριοποιητή και τον πίνακα κατηγοριών. Υπολογίζεται το 10% των δεδομένων τα οποία θα χρησιμοποιηθούν για την αξιολόγηση του κατηγοριοποιητή και στη συνέχεια μία-μία οι προτάσεις διοχετεύονται στον κατηγοριοποιητή.

PolarityHierarchical.java

```
public class PolarityHierarchical {

Set<String> comments = new HashSet<String>(0);
    File mPolarityDir;
    String[] mCategories;
    DynamicLMClassifier<NGramProcessLM> mClassifier;
    JointClassifier<CharSequence> mSubjectivityClassifier;
    int numPos = 0;
    int numNeg = 0;

    public PolarityHierarchical()
        throws ClassNotFoundException, IOException {

        System.out.println("\nHIERARCHICAL POLARITY DEMO");
        mPolarityDir = new File("polarity_dir\\txt_sentoken");
        System.out.println("\nData Directory=" + mPolarityDir);
        mCategories = mPolarityDir.list();
        int nGram = 8;
        mClassifier
            = DynamicLMClassifier
              .createNGramProcess(mCategories,nGram);
        File modelFile = new File("subjectivity.model");
        System.out.println("\nReading Compiled Model from file=" +
modelFile);
        FileInputStream fileIn = new FileInputStream(modelFile);
        ObjectInputStream objIn = new ObjectInputStream(fileIn);
        @SuppressWarnings("unchecked")
        JointClassifier<CharSequence> subjectivityClassifier
            = (JointClassifier<CharSequence>) objIn.readObject();
        mSubjectivityClassifier = subjectivityClassifier;
        objIn.close();
    }

    public void run() throws ClassNotFoundException, IOException {
        train();
        evaluate();
    }

    boolean isTrainingFile(File file) {
        return file.getName().charAt(2) != '9';  // test on fold 9
    }

    void train() throws IOException {
        int numTrainingCases = 0;
        int numTrainingChars = 0;
        System.out.println("\nTraining.");
        for (int i = 0; i < mCategories.length; ++i) {
```

```
            String category = mCategories[i];
            Classification classification
                = new Classification(category);
            File file = new File(mPolarityDir,mCategories[i]);
            File[] trainFiles = file.listFiles();
            for (int j = 0; j < trainFiles.length; ++j) {
                File trainFile = trainFiles[j];
                if (isTrainingFile(trainFile)) {
                    ++numTrainingCases;
                    String review = Files.readFromFile(trainFile,"ISO-8859-
1");
                    numTrainingChars += review.length();
                    Classified<CharSequence> classified
                        = new
Classified<CharSequence>(review,classification);
                    mClassifier.handle(classified);
                }
            }
        }
        System.out.println("  # Training Cases=" + numTrainingCases);
        System.out.println("  # Training Chars=" + numTrainingChars);
        // if you want to write the polarity model out for future use,
        // uncomment the following line
        // com.aliasi.util.AbstractExternalizable.compileTo(mClassifier,new
File("polarity.model"));
    }

    void evaluate() throws IOException {

        Iterator<String> it=comments.iterator();

        while(it.hasNext()) {

        String review = it.next();
            String subjReview = subjectiveSentences(review);
            Classification classification
                 = mClassifier.classify(subjReview);

            System.out.println("\n full review: "+review);
            System.out.println("\n subj review: "+subjReview);
            System.out.println(classification.bestCategory());

            if (classification.bestCategory().equals("pos")){
            ++numPos;
            }
            else{
            ++numNeg;
            }

        }

    }



String subjectiveSentences(String review) {
        String[] sentences = review.split("\n");
```

```
        BoundedPriorityQueue<ScoredObject<String>> pQueue
            = new
BoundedPriorityQueue<ScoredObject<String>>(ScoredObject.comparator(),
                                                        MAX_SENTS);
        for (int i = 0; i < sentences.length; ++i) {
            String sentence = sentences[i];
            ConditionalClassification subjClassification
                = (ConditionalClassification)
                mSubjectivityClassifier.classify(sentences[i]);
            double subjProb;
            if (subjClassification.category(0).equals("quote"))
                subjProb = subjClassification.conditionalProbability(0);
            else
                subjProb = subjClassification.conditionalProbability(1);
            pQueue.offer(new ScoredObject<String>(sentence,subjProb));
        }
        StringBuilder reviewBuf = new StringBuilder();
        Iterator<ScoredObject<String>> it = pQueue.iterator();
        for (int i = 0; it.hasNext(); ++i) {
            ScoredObject<String> so = it.next();
            if (so.score() < .5 && i >= MIN_SENTS) break;
            reviewBuf.append(so.getObject() + "\n");
        }
        String result = reviewBuf.toString().trim();
        return result;
    }

    static int MIN_SENTS = 5;
    static int MAX_SENTS = 25;

    public static void main(String[] args) {
        try {
            new PolarityHierarchical().run();
        } catch (Throwable t) {
            System.out.println("Thrown: " + t);
            t.printStackTrace(System.out);
        }
    }

public Set<String> getComments() {
return comments;
}

public void setComments(Set<String> comments) {
this.comments = comments;
}

public int getNumPos() {
return numPos;
}

public void setNumPos(int numPos) {
this.numPos = numPos;
}

public int getNumNeg() {
return numNeg;
```

```
}

public void setNumNeg(int numNeg) {
this.numNeg = numNeg;
}

}
```

Ο constructor της κλάσης αυτής αρχικά ορίζει τον φάκελο mPolarityDir στη θέση "polarity\txt_sentoken" . Ο πίνακας κατηγοριών αρχικοποιείται χρησιμοποιώντας τα ονόματα των φακέλων που βρίσκονται κάτω από τη θέση στην οποία δείχνει η μεταβλητή mPolarityDir, τα οποία στην συγκεκριμένη περίπτωση είναι "pos" και "neg". Ορίζεται το μήκος του n-gram στο 8. Στη συνέχεια κατασκευάζεται ένας κατηγοριοποιητής με τις καθορισμένες κατηγορίες και το συγκεκριμένο μήκος του n-gram. Ακόμη διαβάζει το μοντέλο υποκειμενικότητας από το αρχείο με όνομα subjectivity.model.

Ο κώδικας του constructor αυξάνει την πιθανότητα να προκύψει είτε μία IOException από το I/O είτε μία ClassNotFoundException διαβάζοντας τον κατηγοριοποιητή από ένα αντικείμενο ρεύματος εισόδου.

Η μέθοδος train() διαπερνά τις κατηγορίες, οι οποίες είναι δύο στην περίπτωση μας. Στην συνέχεια δημιουργεί ένα directory χρησιμοποιώντας το directory των δεδομένων εκπαίδευσης πολικότητας και το όνομα της κατηγορίας. Τα πιθανά αρχεία εκπαίδευσης τοποθετούνται στον πίνακα trainFiles. Για κάθε αρχείο γίνεται έλεγχος για να διαπιστωθεί εάν όντως είναι αρχείο εκπαίδευσης. Εάν είναι διαβάζεται το περιεχόμενο του χρησιμοποιώντας τη μέθοδο του LingPipe Files.readFromFile και στη συνέχεια χρησιμοποιείται για την εκπαίδευση του κατηγοριοποιητή για τη συγκεκριμένη κατηγορία.

Τα δεδομένα εκπαίδευσης οργανώθηκαν σε δέκα ίσου μεγέθους κομμάτια τα οποία διακρίνονται από τον τρίτο χαρακτήρα του ονόματος του αρχείου. Για παράδειγμα το αρχείο pos/cv362_15341.txt είναι ένα παράδειγμα θετικής εκπαίδευσης και ανήκει στο 3ο κομμάτι των δεδομένων εκπαίδευσης ενώ το αρχείο pos/cv532_6522.txt είναι ένα παράδειγμα θετικής εκπαίδευσης το οποίο ανήκει στο 5ο κομμάτι των δεδομένων. Ο κατηγοριοποιητής εκπαιδεύεται με τα κομμάτια 0 έως 8 των δεδομένων εκπαίδευσης και με το 9ο κομμάτι να γίνει ο έλεγχος του κατηγοριοποιητή.

Η μέθοδος evaluate() της κλάσης PolarityHierarchical είναι παρόμοια με αυτή της κλάσης SubjectivityBasic ελάχιστες διαφορές.

```
void evaluate() throws IOException {
     BaseClassifierEvaluator<CharSequence> evaluator
        = new BaseClassifierEvaluator<CharSequence>(null,mCategories,false);
    for (int i = 0; i < mCategories.length; ++i) {
        String category = mCategories[i];
        File file = new File(mPolarityDir,mCategories[i]);
        File[] trainFiles = file.listFiles();
        for (int j = 0; j < trainFiles.length; ++j) {
            File trainFile = trainFiles[j];
```

```
            if (!isTrainingFile(trainFile)) {
                String review
                    = Files.readFromFile(trainFile);
                String subjReview
                    = subjectiveSentences(review);
                Classification classification
                    = mClassifier.classify(subjReview);
                evaluator.addClassification(category,
                                            classification);
            }
        }
    }
    System.out.println();
    System.out.println(evaluator.toString());
}
```

Όπως και στο προηγούμενο παράδειγμα ο κώδικας που παραμένει ίδιος γράφεται με γκρι.

Αρχικά δημιουργείται ένας evaluator με null κατηγοριοποιητή. Το subjReview string είναι το αποτέλεσμα που επιστρέφεται από τη μέθοδο subjectiveSentences όταν αυτή εφαρμοστεί στο σύνολο μίας κριτικής (review). Η μέθοδος αυτή εξάγει τις υποκειμενικές προτάσεις της κριτικής και τις επιστρέφει με τη μορφή ενός string. Στη συνέχεια δημιουργείται μία κατηγοριοποίηση (classification) χρησιμοποιώντας την φιλτραρισμένη είσοδο subjReview. Τέλος αυτή προστίθεται σαν μία περίπτωση στον evaluator. Το παράδειγμα αυτό περιγράφει πως μπορεί να χρησιμοποιηθεί ένας evaluator χωρίς τη χρήση ενός κατηγοριοποιητή.

Σημαντικό κομμάτι αυτής της υλοποίησης είναι η εξαγωγή των υποκειμενικών προτάσεων. Στην συγκεκριμένη υλοποίηση εξάγουμε 5 έως 20 προτάσεις. Αυτές θα είναι οι 5 πιο υποκειμενικές προτάσεις σύμφωνα με την κατάταξη που προκύπτει από το μοντέλο υποκειμενικότητας και μέχρι και 20 επιπλέον προτάσεις εάν είναι 50% ή περισσότερο πιθανό αυτές να είναι υποκειμενικές σύμφωνα με το μοντέλο υποκειμενικότητας.

```
static int MIN_SENTS = 5;
static int MAX_SENTS = 25;

String subjectiveSentences(String review) {
        String[] sentences = review.split("\n");
        BoundedPriorityQueue<ScoredObject<String>> pQueue
            = new
BoundedPriorityQueue<ScoredObject<String>>(ScoredObject.comparator(),
                                                        MAX_SENTS);
        for (int i = 0; i < sentences.length; ++i) {
            String sentence = sentences[i];
            ConditionalClassification subjClassification
                = (ConditionalClassification)
                mSubjectivityClassifier.classify(sentences[i]);
            double subjProb;
            if (subjClassification.category(0).equals("quote"))
                subjProb = subjClassification.conditionalProbability(0);
            else
```

```
                subjProb = subjClassification.conditionalProbability(1);
            pQueue.offer(new ScoredObject<String>(sentence,subjProb));
        }
        StringBuilder reviewBuf = new StringBuilder();
        Iterator<ScoredObject<String>> it = pQueue.iterator();
        for (int i = 0; it.hasNext(); ++i) {
            ScoredObject<String> so = it.next();
            if (so.score() < .5 && i >= MIN_SENTS) break;
            reviewBuf.append(so.getObject() + "\n");
        }
        String result = reviewBuf.toString().trim();
        return result;
    }
```

Η πρώτη γραμμή της μεθόδου απλά χωρίζει το κείμενο στις προτάσεις του τις οποίες εισάγει σε έναν πίνακα. Στη συνέχεια δημιουργείται μία ουρά προτεραιότητας τα αντικείμενα της οποίας κατατάσσονται σύμφωνα με τη βαθμολογία που έχουν λάβει με μέγιστο μήκος ουράς τον μέγιστο αριθμό προτάσεων που μπορούν να επιστραφούν. Ελέγχουμε μία-μία τις προτάσεις και τις κατηγοριοποιούμε με τον κατηγοριοποιητή υποκειμενικότητας που δημιουργήσαμε στον constructor. Το αποτέλεσμα μετατρέπεται σε δεσμευμένη κατηγοριοποίηση κάτι που μας επιτρέπει να εξάγουμε δεσμευμένες πιθανότητες. Στη συνέχεια η πιθανότητα η πρόταση να είναι υποκειμενική ανατίθεται στην μεταβλητή subjProb. Ελέγχουμε αν η κατηγορία “quote” , που είναι η κατηγορία των υποκειμενικών προτάσεων, είναι η πρώτη ή η δεύτερη επιλογή του κατηγοριοποιητή. Τέλος προσθέτουμε ένα βαθμολογημένο αντικείμενο στην ουρά το οποίο αποτελείται από την τρέχουσα πρόταση και την πιθανότητά της να είναι υποκειμενική.

Στο επόμενο τμήμα κώδικα δημιουργείται ένας buffer στον οποίο προσαρτώνται οι υποκειμενικές προτάσεις. Δημιουργούμε έναν iterator με τα αντικείμενα της ουράς προτεραιότητας, ο οποίος επιστρέφει τα αντικείμενα με φθίνουσα πιθανότητα υποκειμενικότητας. Εάν έχουμε ήδη 5 προτάσεις και η πιθανότητα της τρέχουσας πρότασης να είναι υποκειμενική είναι χαμηλότερη του 0,5 βγαίνουμε από το loop. Στην αντίθετη περίπτωση προσθέτουμε την πρόταση στον buffer. Τέλος επιστρέφεται το αποτέλεσμα αφού έχουμε αφαιρέσει κάποια κενά που τυχόν έχουν παραμείνει.

7.3 Κώδικας για τη χρήση του ROME

```
public class RetrieveComments {

private Set<String> comments = new HashSet<String>(0);
private URL url;

public RetrieveComments() {
super();
}

public RetrieveComments(URL url) throws Exception {
super();
```

```
this.url = url;
this.setComments();
}

public Set<String> getComments() {
return comments;
}

public void  setComments() throws Exception {

XmlReader reader = null;

  try {

  String delim="=";
  String delim2="\n";
      reader = new XmlReader(url);

      SyndFeed feed = new SyndFeedInput().build(reader);
      //System.out.println("feed type: "+feed.getFeedType());
      Iterator i = feed.getEntries().iterator();

      while ( i.hasNext())
      {
         SyndEntry entry = (SyndEntry) i.next();
         StringTokenizer st = new
StringTokenizer(entry.getContents().get(0).toString(), delim);
         int x=0;
         while (st.hasMoreElements())
         {
              String token = st.nextElement().toString();
              if(x==1){
              StringTokenizer st2 = new StringTokenizer(token, delim2);
              String token2=st2.nextToken();
              comments.add(token2.replace("&#39;", "'"));

              }
              x++;
         }

       }

   }
  finally {
        if (reader != null)
            reader.close();
   }
}

public URL getUrl() {
return url;
}

public void setUrl(URL url) {
this.url = url;
}
```

```
public void print_comments(){
int size=comments.toArray().length;
for(int x=0;x<size;x++){
System.out.println(comments.toArray()[x]);
}
}

}
```

Ο constructor της παραπάνω κλάσης ορίζει τη διεύθυνση RSS ή Atom από την οποία θα λάβουμε τις κριτικές προς κατηγοριοποίηση. Στη συνέχεια καλεί τη μέθοδο setComments η οποία λαμβάνει τα σχόλια από τη διεύθυνση που ορίσαμε και τα τοποθετεί σε μία δομή Set.

Το ROME περιλαμβάνει parsers για τη μετατροπή syndication feeds σε SyndFeed. Η κλάση SyndFeedInput διαχειρίζεται τους parsers χρησιμοποιώντας κάθε φορά το σωστό βασιζόμενη στο syndication feed που διαχειρίζεται. Δεν είναι δουλειά του προγραμματιστή η σωστή επιλογή parser για κάθε syndication feed, η SyndFeedInput θα επιλέξει το σωστό παρατηρώντας τη δομή του syndication feed. Επομένως η ανάγνωση ενός syndication feed υλοποιείται με τις δύο παρακάτω γραμμές κώδικα:

```
SyndFeedInput input = new SyndFeedInput();
SyndFeed feed = input.build (new XmlReader (feedUrl));
```

7.4 Κώδικας για τη δημιουργία του γραφικού περιβάλλοντος

```
public class GuiWork {

private URL url;
private Set<String> comments = new HashSet<String>(0);
    // Policy
    final int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS;
    final int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;

    JPanel leftPane = new JPanel(new GridLayout(0,1));
    JPanel rightPane = new JPanel(new GridLayout(0,1));
    JPanel topPane = new JPanel();
    JPanel bottomPane= new JPanel();
    JScrollPane scrollPaneLeft;// = new JScrollPane();
    // JScrollPane scrollPaneRight = new JScrollPane();

    JButton submit = new JButton("Submit URL");
    JButton evaluate = new JButton("Evaluate reviews");

    JTextField t = new JTextField("", 30);


    public GuiWork(){
```

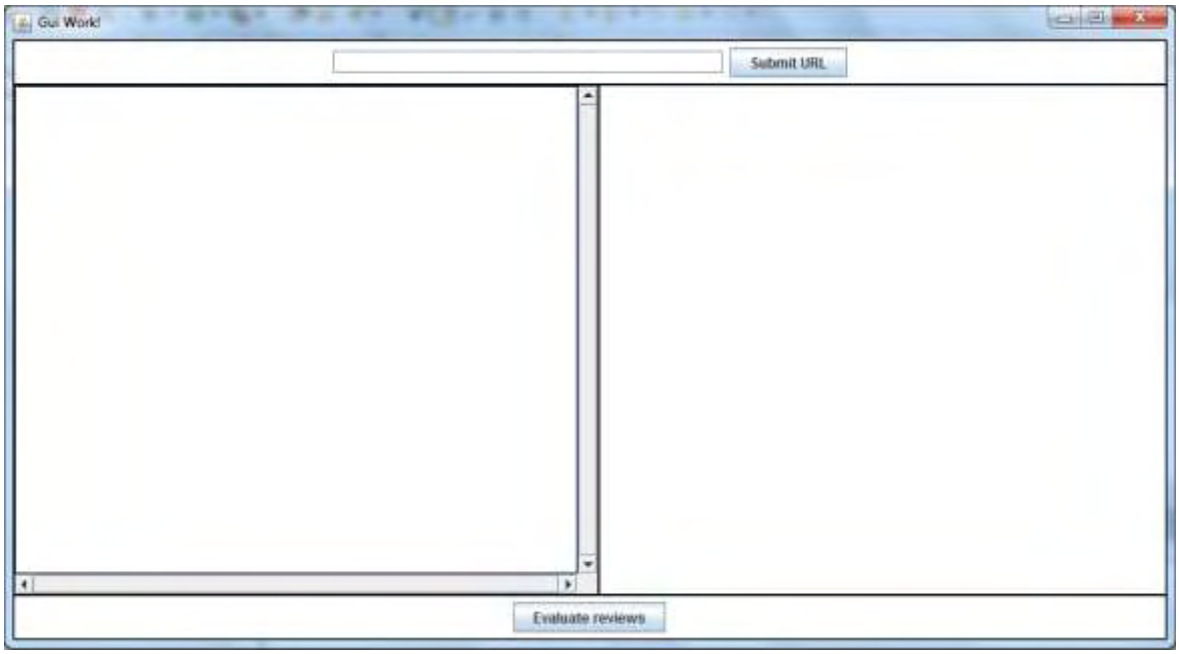
```
    JFrame f = new JFrame("Gui Work!");
        f.setSize(1000, 550);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setLayout(new BorderLayout());

        // Size TopPane on North Border
        topPane.setBackground(Color.white);
        leftPane.setBackground(Color.white);
        rightPane.setBackground(Color.white);
        bottomPane.setBackground(Color.white);

        topPane.setBorder(BorderFactory.createLineBorder(Color.black));
        leftPane.setBorder(BorderFactory.createLineBorder(Color.black));
        rightPane.setBorder(BorderFactory.createLineBorder(Color.black));
        bottomPane.setBorder(BorderFactory.createLineBorder(Color.black));

        rightPane.setPreferredSize(new Dimension(485, 400));


        scrollPaneLeft = new JScrollPane(leftPane, v, h);
        scrollPaneLeft.setPreferredSize(new Dimension(500, 350));


scrollPaneLeft.setBorder(BorderFactory.createLineBorder(Color.black));

        topPane.add(t);
        topPane.add(submit);
        bottomPane.add(evaluate);

        f.add(bottomPane, BorderLayout.SOUTH);
        f.add(rightPane, BorderLayout.EAST);
        f.add(topPane, BorderLayout.NORTH);
        f.add(scrollPaneLeft, BorderLayout.WEST);

        submit.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e)
            {
                //Execute when button is pressed
                System.out.println("You clicked the submit button");
                String text = t.getText();
                System.out.println(text);
                try {
                    url = new URL(text);
                    try{
                    RetrieveComments ret=new RetrieveComments(url);
                    comments=ret.getComments();
                    Iterator<String> it=comments.iterator();
             ret.print_comments();
             while(it.hasNext()){
             leftPane.add(new JLabel(it.next()));
             leftPane.updateUI();
             }

                    }
                    catch(Exception exp){
                    exp.printStackTrace();
```

```
                    }
                } catch (MalformedURLException ex) {
                    ex.printStackTrace();
                }

            }
        });

        evaluate.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e)
            {

                //Execute when button is pressed
                System.out.println("You clicked the train button");
                try {
                PolarityHierarchical polarity=new PolarityHierarchical();
                polarity.setComments(comments);
        System.out.println("  # comments size=" + comments.size());
        try {
        polarity.run();
        rightPane.add(new JLabel("   Positive:
"+Integer.toString(polarity.getNumNeg())));
        rightPane.add(new JLabel("   Negative:
"+Integer.toString(polarity.getNumPos())));
        rightPane.updateUI();
        } catch (Throwable t) {
        System.out.println("Thrown: " + t);
        t.printStackTrace(System.out);
        }

                } catch (Throwable t) {
                    System.out.println("Thrown: " + t);
                    t.printStackTrace(System.out);
                }

            }
        });

        f.setVisible(true);
    }


public URL getUrl() {
return url;
}


public void setUrl(URL url) {
this.url = url;
}


public Set<String> getComments() {
return comments;
}
```

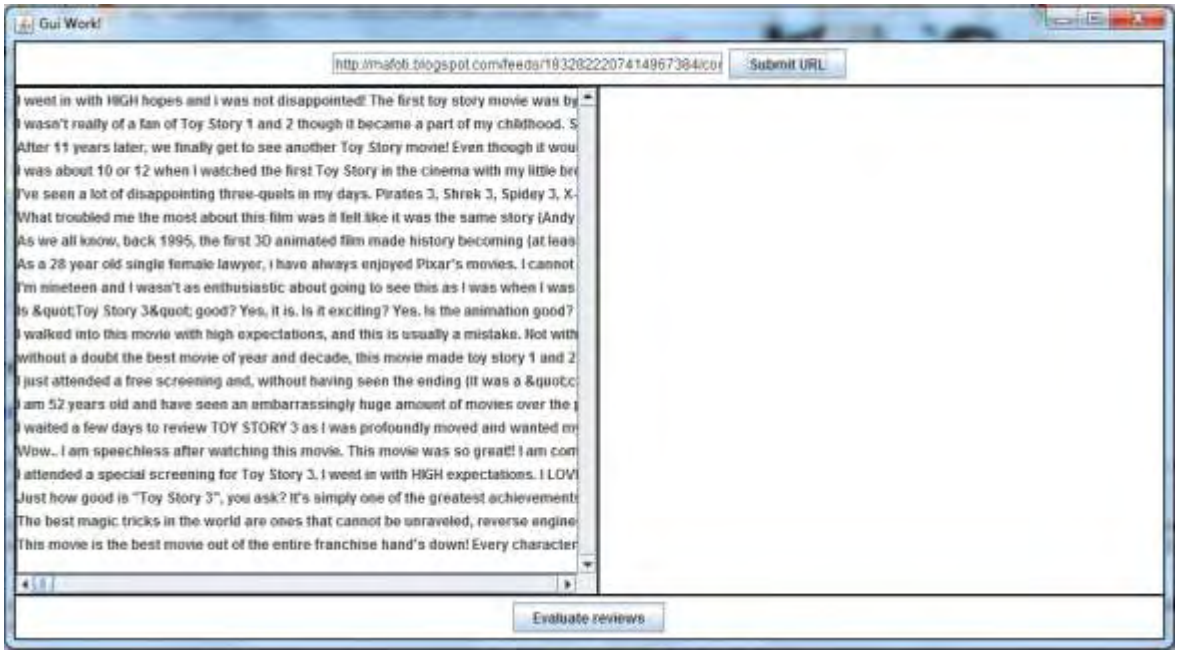
```
public void setComments(Set<String> comments) {
this.comments = comments;
}
}
```

8 Παρουσίαση Συστήματος

Με την εκκίνηση του προγράμματος στην οθόνη μας εμφανίζεται το παρακάτω παράθυρο.



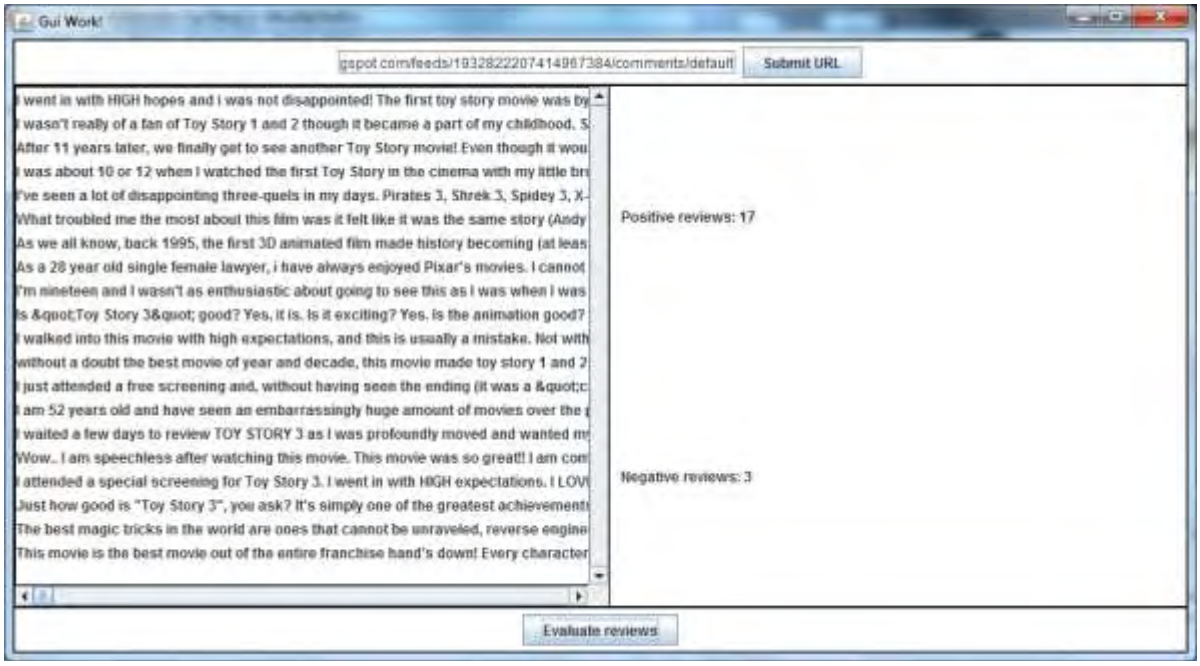
Στο πάνω μέρος του παραθύρου αυτού βρίσκεται ένα text field στο οποίο εισάγουμε την διεύθυνση Url η οποία αντιστοιχεί είτε σε ένα Atom είτε σε ένα RSS feed και την κάνουμε submit.



Μόλις κάνουμε Submit τη διεύθυνση καλείται η μέθοδος για την ανάγνωση των σχολίων και στη συνέχεια αυτά εμφανίζονται στο αριστερό μέρος του παραθύρου μας. Κάθε σχόλιο τοποθετείται σε μία γραμμή.

Στη συνέχεια πατώντας Evaluate reviews γίνεται η εκπαίδευση του κατηγοριοποιητή και κατηγοριοποιούνται οι κριτικές.

Μόλις τελειώσει αυτή η διαδικασία εμφανίζονται τα αποτελέσματα στο δεξί μέρος του παραθύρου.



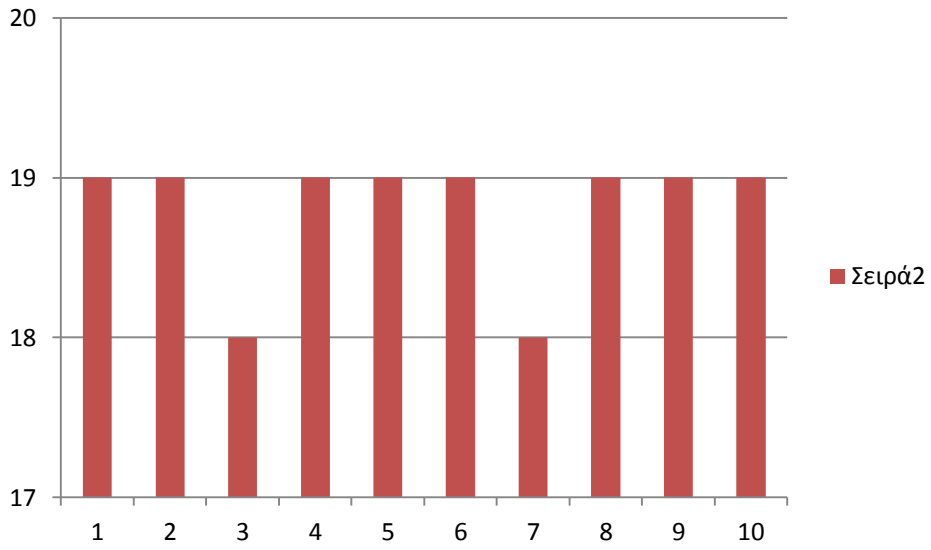
9 Αξιολόγηση Συστήματος

Μετά από αναζήτηση στο διαδίκτυο διαπιστώθηκε ότι οι ιστότοποι οι οποίοι περιέχουν περιλήψεις ταινιών και οι χρήστες μπορούν να αφήσουν σχόλια για τις ταινίες αυτές δεν διαθέτουν RSS ή ATOM feeds μέσω των οποίων να γίνονται διαθέσιμα τα σχόλια αυτά. Για το λόγο αυτό δημιουργήθηκε ένα blog. Στο blog αυτό έγιναν δύο αναρτήσεις που περιείχαν την περίληψη δύο ταινιών και σε κάθε ανάρτηση προστέθηκαν σχόλια για τις ταινίες αυτές. Τα σχόλια αυτά είναι σχόλια τα οποία αντιγράφηκαν από το http://www.imdb.com και αναρτήθηκαν εκεί από διάφορους χρήστες.

Επειδή τα σχόλια στο blog δεν είναι βαθμολογημένα (π.χ. με μία κλίμακα 10 αστέρων) δεν μπορούμε να ξέρουμε εκ των προτέρων πόσα από αυτά είναι θετικά και πόσα αρνητικά, συνεπώς δεν μπορούν να αξιολογηθούν τα αποτελέσματα που επιστρέφει το σύστημά μας με αυτόματο τρόπο. Για να ξεπεραστεί το πρόβλημα αυτό δημιουργήθηκε ένα ερωτηματολόγιο το οποίο περιείχε όλα τα σχόλια που αναρτήθηκαν στο blog. Το ερωτηματολόγιο μοιράστηκε σε 10 άτομα από τα οποία και ζητήθηκε να χαρακτηρίσουν κάθε σχόλιο είτε ως θετικό είτε ως αρνητικό.

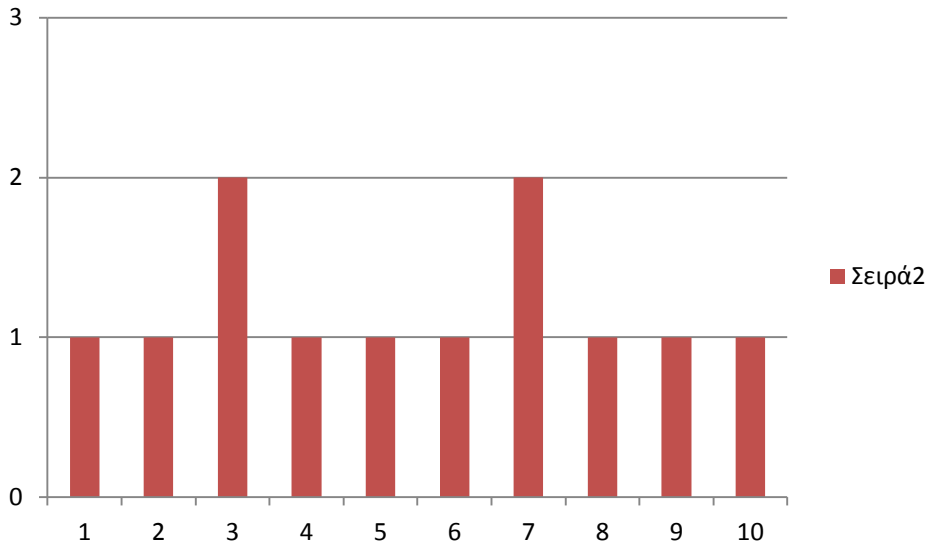
Η πρώτη ανάρτηση περιείχε 20 σχόλια ενώ η δεύτερη 80. Τα αποτελέσματα των ερωτηματολογίων φαίνονται παρακάτω.

Ταινία 1:



Εικόνα 14: πλήθος Θετικών σχολίων για την ταινία 1 σύμφωνα με τα 10 άτομα που απάντησαν το ερωτηματολόγιο.

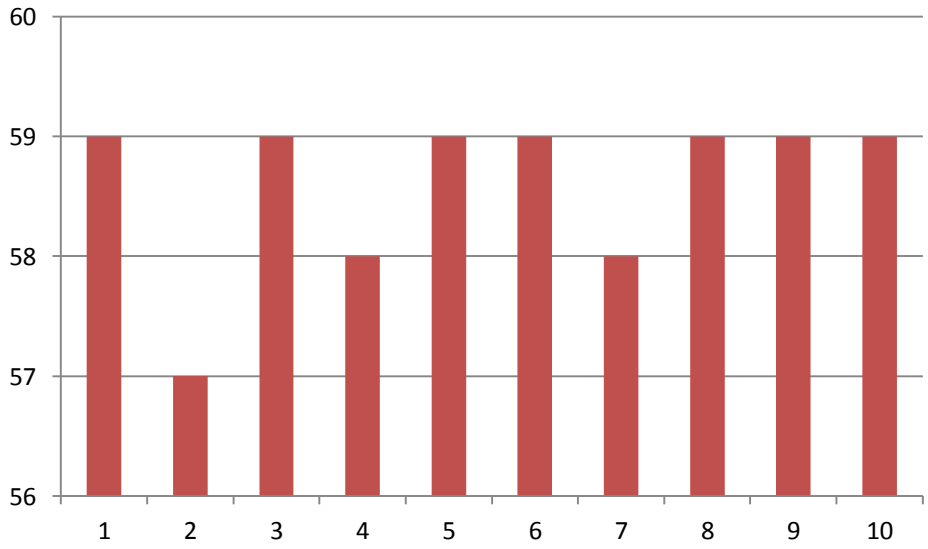
Ο μέσος όρος του πλήθους των θετικών σχολίων είναι : 18,8 σχόλια το οποίο ισούται με το 94% των σχολίων



Εικόνα 15: πλήθος αρνητικών σχολίων για την ταινία 1 σύμφωνα με τα 10 άτομα που απάντησαν το ερωτηματολόγιο.

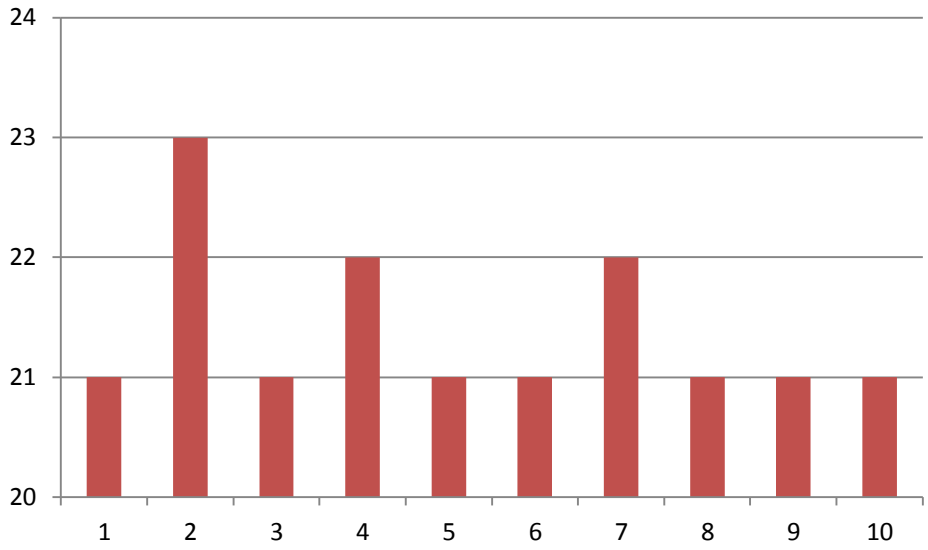
Μέσος όρος του πλήθους των αρνητικών σχολίων είναι: 1,2 σχόλια το οποίο ισούται με το 4% των σχολίων .

Ταινία 2:



Εικόνα 16: πλήθος Θετικών σχολίων για την ταινία 2 σύμφωνα με τα 10 άτομα που απάντησαν το ερωτηματολόγιο.

Ο μέσος όρος του πλήθους των θετικών σχολίων είναι : 58,6 σχόλια το οποίο ισούται με το 73.25% των σχολίων

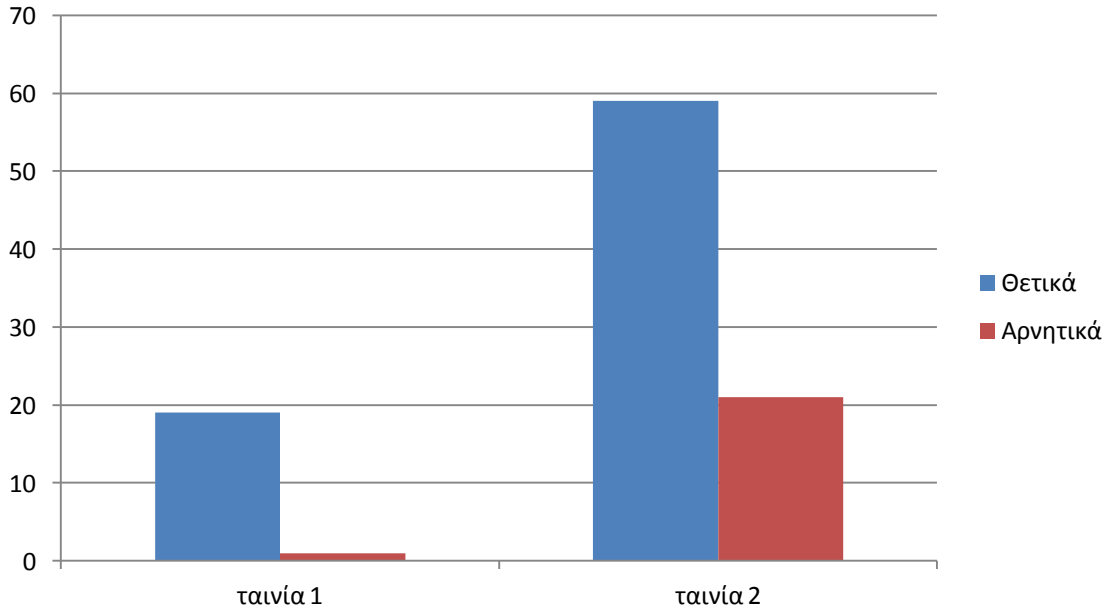


Εικόνα 17: πλήθος αρνητικών σχολίων για την ταινία 2 σύμφωνα με τα 10 άτομα που απάντησαν το ερωτηματολόγιο.

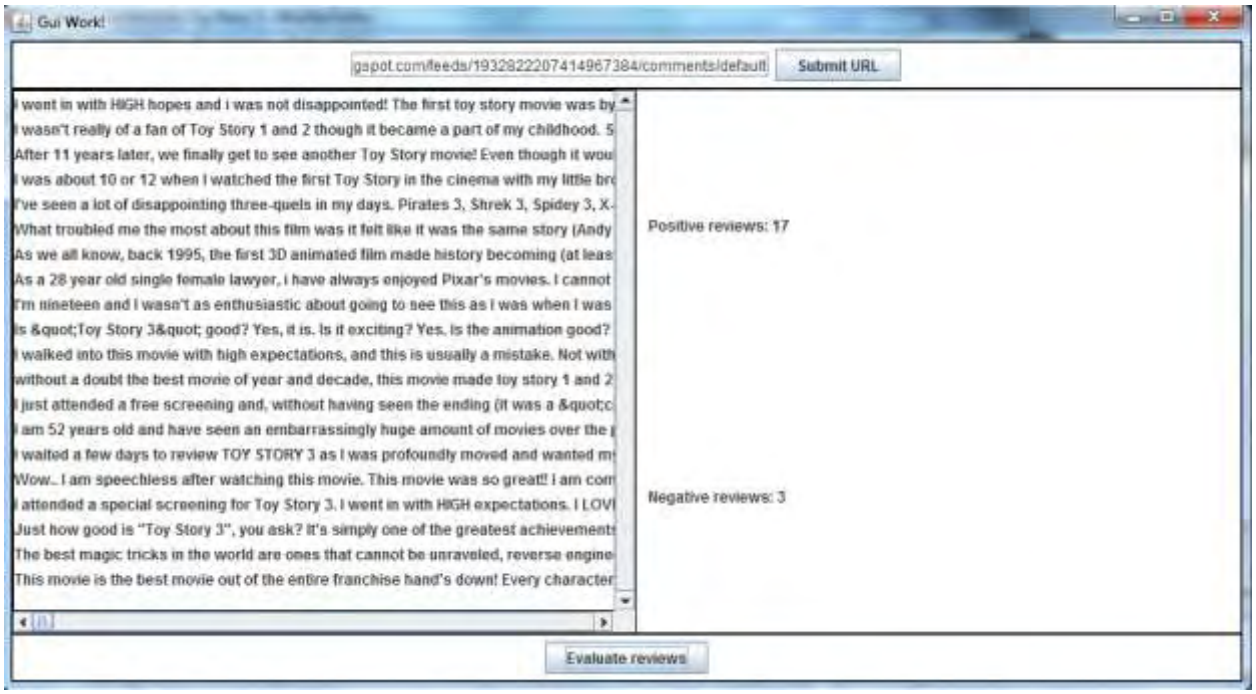
Ο μέσος όρος του πλήθους των αρνητικών σχολίων είναι : 21,4 σχόλια το οποίο ισούται με το 26.75% των σχολίων

Το γεγονός ότι δεν συμφωνούν όλα τα άτομα μεταξύ τους στο χαρακτηρισμό των σχολίων οφείλεται στο γεγονός ότι κάποια από τα σχόλια είναι ουδέτερα. Το σύστημά μας όμως χαρακτηρίζει τα σχόλια είτε ως θετικά είτε ως αρνητικά για το λόγο αυτό στο ερωτηματολόγιο υπήρχε η δυνατότητα χαρακτηρισμού μόνο ως θετικό ή αρνητικό ενός σχολίου.

Συνοψίζοντας τα παραπάνω αποτελέσματα προκύπτει το παρακάτω γράφημα.



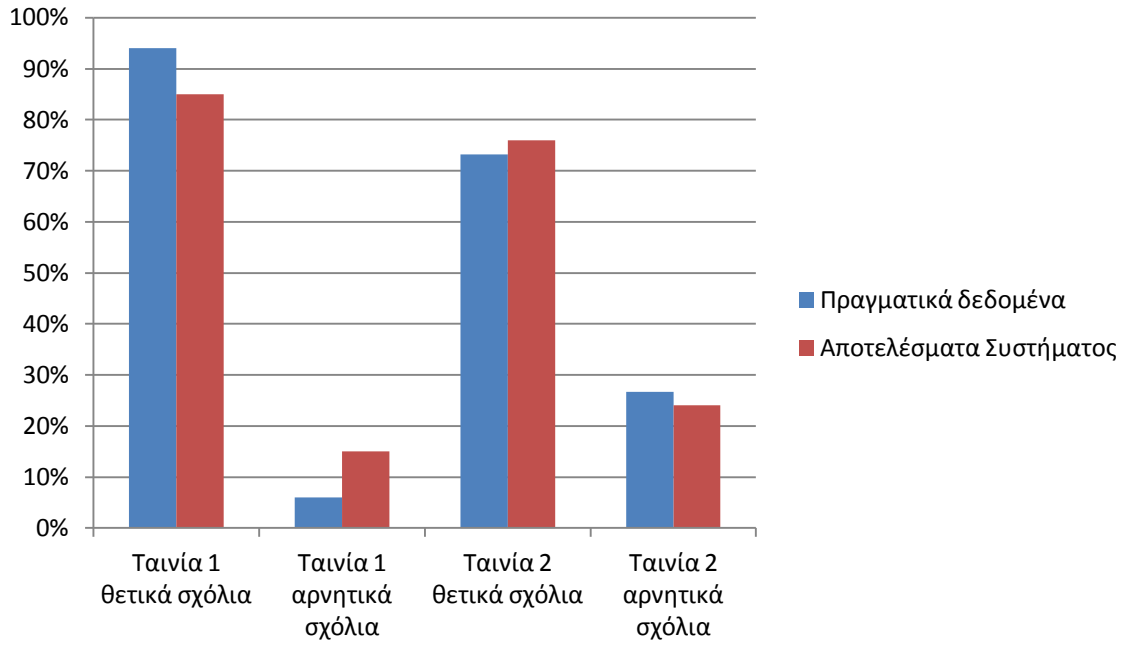
Ταινία 1:



Βλέπουμε ότι το σύστημά μας έχει αποφανθεί ότι το 85% των σχολίων είναι θετικά ενώ το 15% των σχολίων είναι αρνητικά.

Ταινία 2:

Για τη δεύτερη ταινία βλέπουμε ότι τα αποτελέσματα του συστήματός μας λένε ότι το 76% των σχολίων είναι θετικά ενώ το 24% αυτών είναι αρνητικά.



10 Μελλοντική Δουλειά

Σαν μελλοντική επέκταση της εργασίας αυτής, μπορούν να μελετηθούν οι μεταβολές που μπορούν να γίνουν στον κώδικα του LingPipe έτσι ώστε να είναι ένα εργαλείο ανεξάρτητο γλώσσας έτσι ώστε να μπορεί να λειτουργεί και σε άλλες γλώσσες εφόσον γίνει συλλογή των απαραίτητων κειμένων για την εκπαίδευση του κατηγοριοποιητή.

Πέρα από τις αλλαγές που μπορούν να γίνουν στο LingPipe, μεγάλο ενδιαφέρον θα είχε και η ανάπτυξη ενός συστήματος ανάλυσης συναισθήματος κειμένων γραμμένων στην ελληνική γλώσσα με τη λεξιλογική προσέγγιση. Ένα τέτοιο σύστημα θα μπορούσε να βασιστεί στο εννοιολογικό λεξικό του BalkaNet, που έχει υλοποιηθεί από το πανεπιστήμιο Πατρών.

11 Βιβλιογραφία

1. Peter Turney (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". *Proceedings of the Association for Computational Linguistics (ACL)*. pp. 417–424.
2. Bo Pang, Lillian Lee and Shivakumar Vaithyanathan (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques". *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 79–86.
3. Bo Pang; Lillian Lee (2005). "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales". *Proceedings of the Association for Computational Linguistics (ACL)*. pp. 115–124.
4. Benjamin Snyder; Regina Barzilay (2007). "Multiple Aspect Ranking using the Good Grief Algorithm". *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL)*. pp. 300–307.
5. J. Carbonell(1979). "Subjective Understanding: Computer Models of Belief Systems". PhD thesis, Yale.
6. Y. Wilks and J. Bien (1984), "Beliefs, points of view and multiple environments," in *Proceedings of the international NATO symposium on artificial and human intelligence*, pp. 147–171, USA, New York, NY: Elsevier North-Holland, Inc.
7. M. Hearst (1992), "Direction-based text interpretation as an information access refinement," in *Text-Based Intelligent systems*, (P. Jacobs, ed.), pp. 257–274, Lawrence Erlbaum Associates.
8. A. Huettner and P. Subasic (2000), "Fuzzy typing for document management," in *ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes*, pp. 26–27.
9. M. Kantrowitz (2003), "Method and apparatus for analyzing affect and emotion intext," U.S. Patent 6622140, Patent filed in November 2000.
10. W. Sack (1994), "On the computation of point of view," in *Proceedings of AAAI*, p. 1488. (Student abstract).
11. J. Wiebe and R. Bruce (1995), "Probabilistic classifiers for tracking point of view," in *Proceedings of the AAAI Spring Symposium on Empirical Methods in DiscourseInterpretation and Generation*, pp. 181–187.
12. J. M. Wiebe, "Identifying subjective characters in narrative," in *Proceedings of the International Conference on Computational Linguistics (COLING)*,pp. 401–408, 1990.
13. J. M. Wiebe (1994), "Tracking point of view in narrative," *Computational Linguistics*,vol. 20, pp. 233–287
14. J. M. Wiebe, R. F. Bruce, and T. P. O'Hara (1999), "Development and use of a gold standard data set for subjectivity classifications," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 246–253.
15. J. M. Wiebe and W. J. Rapaport (1988), "A computational theory of perspective and reference in narrative," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 131–138.
16. C. Cardie, J. Wiebe, T. Wilson, and D. Litman (2003), "Combining low-level and summary representations of opinions for multi-perspective question answering,"

in *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*, pp. 20–27.
17. S. Das and M. Chen (2001), "Yahoo! for Amazon: Extracting market sentiment from stock message boards," in *Proceedings of the Asia Pacific Finance AssociationAnnual Conference (APFA)*
18. K. Dave, S. Lawrence, and D. M. Pennock (2003), "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of WWW*, pp. 519–528.
19. L. Dini and G. Mazzini (2002), "Opinion classification through information extraction," in *Proceedings of the Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields (Data Mining)*, pp. 299–310.
20. H. Liu, H. Lieberman, and T. Selker (2003), "A model of textual affect sensing using real-world knowledge," in *Proceedings of Intelligent User Interfaces (IUI)*,pp. 125–132.
21. S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima (2002), "Mining product reputations on the Web," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 341–349. (Industry track).
22. T. Nasukawa and J. Yi (2003), "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the Conference on Knowledge Capture (K-CAP)*.
23. B. Pang, L. Lee, and S. Vaithyanathan (2002), "Thumbs up? Sentiment classification using machine learning techniques," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86.
24. K. Tateishi, Y. Ishiguro, and T. Fukushima (2001), "Opinion information retrieval from the internet," *Information Processing Society of Japan (IPSJ) SIGNotes,* 2001, vol. 69, no. 7, pp. 75–82.
25. R. M. Tong (2001), "An operational system for detecting and tracking opinions in on-line discussion," in *Proceedings of the Workshop on Operational Text Classification(OTC)*.
26. J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman,D. Pierce, E. Riloff, T. Wilson, D. Day, and M. Maybury (2003), "Recognizing and organizing opinions expressed in the world press," in *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*.
27. H. Yu and V. Hatzivassiloglou (2003), "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
28. Bo Pang and Lillian Lee (2008), "Opinion mining and sentiment analysis". *Foundations and Trends in Information Retrieval* 2(1-2), pp. 1–135.
29. Kennedy, A., Inkpen, D.(2006) : Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. Computational Intelligence, 110–125
30. Kamps, J., Marx, M., Mokken, R.J.(2004): Using WordNet to Measure Semantic Orientation of Adjectives. In: LREC 2004, vol. IV, pp. 1115–1118

31. Hatzivassiloglou, V., Wiebe, J.(2000): Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In Proceedings of the 18th International Conference on Computational Linguistics, New Brunswick, NJ
32. Andreevskaia, A., Bergler, S., Urseanu, M.(2007): All Blogs Are Not Made Equal: Exploring Genre Differences in Sentiment Tagging of Blogs. In: International Conference on Weblogs and Social Media (ICWSM-2007), Boulder, CO
33. Turney, P.D., Littman, M.L.(2003): Measuring Praise and Criticism: Inference of Semantic Orientation from Association. ACM Transactions on Information Systems, 315–346
34. Yang, Y., & Pedersen, Jan O.(1997): A comparative study on feature selection in text categorization. ICML, 412–420.
35. D.D. Lewis and M.Ringuette (1994): Comparison of two learning algorithms for text categorization. In Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDIAR '94).
36. K. Lang(1995): Newsweeder: Learning to filter netnews. In Proceedings of the Twelfth International Conference on Machine Learning.
37. I. Moulinier, G. Raskinis and J. Ganascia (1996): Text Categorization: a symbolic approach. . In Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval.
38. H. Schutze, D.A. Hull and J.O.Pedersen(1995): A comparison of classifiers and document representations for the routing problem. In 18th Ann Int ACM SIGIR Conference on Research and Development in information Retrieval (SIGIR '95), p. 229-237.
39. E. Wiener, J.O. Pedersen and A.S. Weigend (1995): A neural network approach to topic spotting. In Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR '95).
40. Y. Yang(1995): Noise reduction in a statistical approach to text categorization. In 18th Ann Int ACM SIGIR Conference on Research and Development in information Retrieval (SIGIR '95), p. 256-263.
41. Y. Yang and W.J. Wilbur (1996): Using corpus statistics to remove redundant words in text categorization. In J Amer Soc Inf Sci.
42. Tom Mitchell(1996): Machine Learning. McCraw Hill.
43. J.R. Quinlan (1986): Induction of decision trees. Machine Learning, 1(1):81-106.
44. Tan, S. , & Zhang, J.(2007): An empirical study of sentiment analysis for chinese documents. Expert Systems with Applications (2007), doi:10.1016/j.eswa.2007.05.028
45. Vapnik, Vladmimir N.(1995): The Nature of statistical learning theory. New York: Springer
46. C. Cortes and V. Vapnik (1995): Support vector networks. Machine Learning, 20:273-297.

47. Osuna, R. Freund, and F. Girosi.(1996): Support vector machines: Training and applications. In A.I. Memo. MIT A.I. Lab.
48. Yang, Y., & Lin, X.(1999): A re-examination of text categorization methods. SIGIR, 42–49.
49. Belur V. Dasarathy(1991): Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. McGraw-Hill Computer Science Series. IEEE Computer Society Press, Las Alamitos, California.
50. B. Masand, G. Linoff, and D.Waltz (1992). Classifying news stories using memory based reasoning. In 15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92), p. 59-64.
51. Y. Yang (1994): Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In 17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94), p. 13-22.
52. Makato Iwayama and Takenobu Tokunaga (1995): Cluster-based text categorization: a comparison of category search strategies. In Proceedings of the 18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95), p. 273-281.
53. van Rijsbergen, C.(1979): Information retrieval. London: Butterworth.
54. B. Pang and L. Lee (2004): "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 271–278
55. Blum, Avrim and Shuchi Chawla (2001): Learning from labeled and unlabeled data using graph mincuts.In *Intl. Conf. on Machine Learning (ICML)*, p. 19.26.
56. Riloff, Ellen and Janyce Wiebe. (2003): Learning extraction patterns for subjective expressions. In *EMNLP*.
57. A McCallum, K Nigam (1998): A comparison of event models for Naive Bayes text classification.. AAAI98 Workshop on Learning for Text Categorization.
58. http://wiki.java.net/bin/view/Javawsxml/Rome04HowRomeWorks
59. http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared
60. http://www.rssboard.org/rss-specification/
61. http://cyber.law.harvard.edu/rss/rss.html
62. http://tools.ietf.org/html/rfc4287#section-1.1
63. http://alias-i.com/lingpipe