ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Συναισθήματος σε Δεδομένα Κοινωνικών Δικτύων με χρήση Γράφων ν-γραμμάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Μ. Τζαννέτος

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Συναισθήματος σε Δεδομένα Κοινωνικών Δικτύων με χρήση Γράφων ν-γραμμάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Μ. Τζαννέτος

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Οκτωβρίου 2014.

.....	.....	.....
Θεοδώρα Βαρβαρίγου Καθηγήτρια Ε.Μ.Π.	Βασίλειος Λούμος Καθηγητής Ε.Μ.Π.	Ελευθέριος Καγιάφας Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

Δημήτριος Μ. Τζαννέτος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η έλευση του Διαδικτύου δημιούργησε νέα κανάλια επικοινωνίας, ενημέρωσης και ανταλλαγής απόψεων. Η ανάγκη εξαγωγής χρήσιμων συμπερασμάτων αναλύοντας με αυτοματοποιημένο τρόπο τον τεράστιο όγκο του παραγόμενου από χρήστες περιεχομένου οδήγησε στην εμφάνιση της Ανάλυσης Συναισθήματος. Ειδικότερα, τα δεδομένα των Κοινωνικών Μέσων Δικτύωσης λόγω των εγγενών χαρακτηριστικών τους δημιουργούν σημαντικές δυσκολίες με αποτέλεσμα να αναζητούνται νέες μέθοδοι επεξεργασίας.

Σε αυτό το πλαίσιο, η παρούσα εργασία μελετά την εφαρμογή του μοντέλου γράφων ν-γραμμάτων - μίας επιβλεπόμενης προσέγγισης ανεξάρτητη γλώσσας που έχει προταθεί στη βιβλιογραφία - σε ένα πολυγλωσσικό και πολυθεματικό περιβάλλον. Διερευνά τον τρόπο με τον οποίο ανταποκρίνεται στη μεταβολή των βασικών παραμέτρων του μοντέλου και προτείνει τους πλέον κατάλληλους για εφαρμογή αλγορίθμους Μηχανικής Μάθησης. Έπειτα, εξετάζει τροποποιήσεις της αρχικής σύνθεσης και επιτυγχάνει να αυξήσει την ακρίβεια ταξινομητών με μέτρια επίδοση. Τέλος, μέσω σχημάτων ensemble, προσδιορίζει τους συνδυασμούς ταξινομητών οι οποίοι εμφανίζουν συστηματικά βελτιωμένα ποσοστά ακρίβειας.

Λέξεις κλειδιά

ανάλυση συναισθήματος, γράφοι ν-γραμμάτων, αλγόριθμοι Επιβλεπόμενης Μηχανικής Μάθησης, κατηγοριοποίηση πολικότητας, τεχνικές ensemble

Abstract

The advent of the Internet created new channels of communication, information and opinion exchange. The need to extract useful insight through automated analysis of the vast amount of user-generated content gave rise to Sentiment Analysis. Social Media content, in particular, due to its inherent characteristics poses serious challenges that call for new processing techniques.

In this context, this thesis focuses on the application of n-gram graphs model - a language-independent supervised approach presented in the bibliography - to a multilingual and multi-topic environment. Explores how the model responds to basic parameter changes and proposes the most applicable machine learning algorithms. Furthermore, examines modifications of the initial structure and achieves to enhance the accuracy of low-performance classifiers. Finally, using ensemble techniques, specifies classifier combinations that consistently show higher accuracy rates.

Key words

sentiment analysis, n-gram graphs, supervised machine learning algorithms, polarity classification, ensemble techniques

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Κατανεμημένης Γνώσης και Συστημάτων Πληροφορικής της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών και επισφραγίζει τις σπουδές μου στο Εθνικό Μετσόβιο Πολυτεχνείο.

Θα ήθελα να ευχαριστήσω ιδιαιτέρως την καθηγήτριά μου κα. Θεοδώρα Βαρβαρίγου για την εμπιστοσύνη που μου έδειξε και την ευκαιρία που μου παρείχε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Επίσης, θα ήθελα να ευχαριστήσω τον Μεταδιδακτορικό Ερευνητή Φώτη Αίσωπο και τον Υποψήφιο Διδάκτορα Θάνο Παπαοικονόμου για το χρόνο που μου αφιέρωσαν και τις συμβουλές τους καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Τέλος, ευχαριστώ την οικογένειά μου για τη στήριξη και βοήθεια κατά τη διάρκεια των σπουδών μου.

Δημήτριος Μ. Τζαννέτος

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

1

Εισαγωγή

Τα τελευταία χρόνια, η ανάπτυξη του Διαδικτύου έχει αλλάξει σε μεγάλο βαθμό την καθημερινότητά μας προσφέροντας νέους τρόπους επικοινωνίας, ενημέρωσης και αλληλεπίδρασης μεταξύ των ανθρώπων. Οι χρήστες του Internet δεν είναι πλέον παθητικοί αποδέκτες πληροφοριών αλλά συμμετέχοντας σε κοινωνικά δίκτυα έχουν τη δυνατότητα να συζητήσουν με άλλους χρήστες,να ανταλλάξουν απόψεις και ιδέες, απομακρύνονται δε με τον τρόπο αυτό από τις πιο παραδοσιακές υπηρεσίες όπως τα e-mails. Η άποψη της κοινής γνώμης γύρω από ποικίλα θέματα απασχολεί την ανθρώπινη δραστηριότητα η οποία προσπαθεί να την αφουγκραστεί μέσω ερωτηματολογίων και δημοσκοπήσεων. Καθώς ολοένα και περισσότεροι χρήστες αναρτούν κριτικές σχετικά με τα προϊόντα ή υπηρεσίες που χρησιμοποιούν, οι πλατφόρμες κοινωνικής δικτύωσης αποτελούν σημαντικές πηγές πληροφοριών όσον αφορά την άποψη και τα συναισθήματα των ανθρώπων - οι οποίες για πρώτη φορά στην ιστορία είναι καταγεγραμμένες απευθείας σε ηλεκτρονική μορφή. Η ανάγκη ανάλυσης και αξιοποίησης του παραγόμενου όγκου πληροφορίας με αυτοματοποιημένο τρόπο οδήγησε στην εμφάνιση της *Ανάλυσης Συναισθήματος* (*Sentiment Analysis*).

Η Ανάλυση Συναισθήματος σε δεδομένα κοινωνικών δικτύων αποκτά ολοένα και περισσότερο έδαφος στον ακαδημαϊκό χώρο λόγω των τεχνικών προκλήσεων που θέτει αλλά και στον επιχειρηματικό χώρο χάρη στις πολλά υποσχόμενες προοπτικές της. Πρωτοπόρες εταιρείες επενδύουν στην εξόρυξη γνώμης από τα κοινωνικά μέσα δικτύωσης, χρησιμοποιώντας τεχνικές ανάλυσης συναισθήματος. Είναι ιδιαίτερα κρίσιμη η έγκαιρη ανάλυση της γνώμης των καταναλωτών στα κοινωνικά δίκτυα και η κατάλληλη προσαρμογή στις ανάγκες τους καθώς αυξάνεται συνεχώς ο αριθμός των ανθρώπων που στηρίζονται σε αξιολογήσεις άλλων καταναλωτών πριν λάβουν την τελική απόφαση αγοράς.

1.1 Διατύπωση Προβλήματος

Η Ανάλυση Συναισθήματος μπορεί να εφαρμοσθεί σε διάφορα επίπεδα ανάλυσης ανάλογα με το μέγεθος του κειμένου και τη ζητούμενη λεπτομέρεια της εξαγόμενης πληροφορίας:

- Επίπεδο Εγγράφου : γίνεται η παραδοχή ότι σε κάθε έγγραφο διατυπώνεται μία άποψη για ένα συγκεκριμένο αντικείμενο ή θέμα.

- Επίπεδο Πρότασης : διαχωρίζει κάθε έγγραφο σε προτάσεις υποθέτοντας ότι κάθε μία πρόταση εκφράζει μία μόνο άποψη.

- Επίπεδο Χαρακτηριστικού : διαχωρίζει κάθε έγγραφο ή πρόταση σε φράσεις οι οποίες αναφέρονται σε μία οντότητα συνολικά ή ξεχωριστά για κάθε ένα από τα χαρακτηριστικά της με διαφορετικό συναίσθημα.

Ο βαθμός λεπτομέρειας που εξετάζουμε και συναντάται κυρίως στη βιβλιογραφία εστιάζει στην πολικότητα του μηνύματος. Προσπαθεί να κατατάξει την άποψη που διατυπώνεται σε ένα κείμενο σε θετική,αρνητική ή ουδέτερη και για το λόγο αυτό η Ανάλυση Συναισθήματος αναφέρεται συχνά και με τον όρο Εξόρυξη Γνώμης. Πιο προχωρημένες προσεγγίσεις ασχολούνται με τον προσδιορισμό του συγκεκριμένου συναισθήματος που εκφράζει ο δημιουργός π.χ. χαρά, ευχαρίστηση, λύπη, αγανάκτηση.

Στην παρούσα εργασία μελετάμε την Ανάλυση Συναισθήματος σε επίπεδο εγγράφου αναλύοντας πολυγλωσσικά μηνύματα από το κοινωνικό δίκτυο Twitter. Σκοπός είναι ο προσδιορισμός της πολικότητας των κειμένων τα οποία αποτελούνται από μερικές προτάσεις. Ωστόσο, λόγω του περιορισμού του μήκους των μηνυμάτων που επιβάλλει το μέσο, το εύρος της ανάλυσης προσεγγίζει σε μεγάλο βαθμό το επίπεδο πρότασης.

Το πρόβλημα που εξετάζουμε είναι η ταξινόμηση πολυγλωσσικών κειμένων σε τρεις κατηγορίες (θετική, αρνητική ή ουδέτερη). Θεωρούμε πως κάθε μήνυμα ανήκει σε μόνο μία κατηγορία δηλαδή η πολικότητα είναι ενιαία σε όλο το σώμα του εγγράφου (single-label classification). Παράλληλα, εστιάζουμε σε ένα σύνολο γλωσσών χωρίς ωστόσο να γνωρίζουμε εκ των προτέρων τη γλώσσα στην οποία κάθε μήνυμα έχει γραφτεί.

1.2 Προσέγγιση

Τα περισσότερα συστήματα Ανάλυσης Συναισθήματος ανιχνεύουν εκφραστικά μοτίβα και εφαρμόζουν τεχνικές που αξιολογούν τη σημασία και συναισθηματική χροιά συγκεκριμένων λέξεων και φράσεων (π.χ. SentiWordNet). Αν και αυτές οι προσεγγίσεις επιτυγχάνουν αρκετά υψηλή ακρίβεια όταν εφαρμοσθούν σε ένα συγκεκριμένο περιβάλλον με γνωστή θεματολογία, έχουν κατασκευασθεί με την παραδοχή ότι τα προς εξέταση δεδομένα είναι γραμμένα σε μία μόνο εκ των προτέρων γνωστή γλώσσα και δεν περιλαμβάνουν θορυβώδες περιεχόμενο (νεολογισμούς, ορθογραφικά λάθη,αργκό), στοιχεία τα οποία αποτελούν εγγενή χαρακτηριστικά των δεδομένων από κοινωνικά δίκτυα. Σε αυτή την εργασία αντιμετωπίζουμε τους συγκεκριμένου περιορισμούς μελετώντας εκτενέστερα την εφαρμογή της μεθόδου γράφων ν-γραμμάτων σε μηνύματα του Twitter. Η εν λόγω προσέγγιση επιβλεπόμενης μάθησης που προτάθηκε από τους Aisopos, Papadakis, Tserpes και Varvarigou εμφανίζει ανεξαρτησία από τη γλώσσα (language neutrality) και υψηλή ανοχή στο θόρυβο (noise-tolerant) και περιγράφεται αναλυτικά στο Κεφάλαιο 3. Συνοπτικά, δημιουργεί ένα γράφο του οποίου οι κορυφές αντιστοιχούν σε χαρακτήρες ν-γραμμάτων ενός μηνύματος και τα βάρη των ακμών του αναφέρονται στη μέση απόσταση μεταξύ τους. Μηνύματα της ίδιας κατηγορίας πολικότητας συγχωνεύονται στον αντίστοιχο γράφο κλάσης και στη συνέχεια κάθε γράφος μηνύματος συγκρίνεται με τους γράφους κλάσεων για να προσδιορισθεί η πολικότητα του μηνύματος. Το μοντέλο

γράφων ν-γραμμάτων εμφάνισε υψηλά ποσοστά ακρίβειας κατηγοριοποίησης εφαρμοζόμενο σε μεγάλο όγκο δεδομένων που παράχθηκε με αυτόματο τρόπο με την τεχνική της εξ αποστάσεως επίβλεψης (distant supervision).

Στόχος της παρούσας εργασίας είναι : η μελέτη της συμπεριφοράς της μεθόδου σε ένα σύνολο χειροκίνητα ταξινομημένων (manually annotated) πολυγλωσσικών δεδομένων, η διερεύνηση και προσπάθεια βελτίωσης των αποτελεσμάτων των αλγορίθμων Μηχανικής Μάθησης και τέλος η διενέργεια παρεμβάσεων σε στάδια της μεθόδου προκειμένου να διαπιστωθεί η ανταπόκρισή της σε αυτές.

1.3 Οργάνωση Κειμένου

Το υπόλοιπο κείμενο διαρθρώνεται ως εξής :

Κεφάλαιο 2 : Παρουσιάζεται εκτενέστερα το πρόβλημα της Ανάλυσης Συναισθήματος, οι δυσκολίες και προκλήσεις που καλείται να αντιμετωπίσει, οι διάφορες προσεγγίσεις που χρησιμοποιούνται και πραγματοποιείται μία ανασκόπηση της σχετικής βιβλιογραφίας.

Κεφάλαιο 3 : Περιγράφεται αναλυτικά η μέθοδος κατηγοριοποίησης με γράφους ν-γραμμάτων και παρουσιάζονται συνοπτικά τα χαρακτηριστικά των αλγορίθμων Μηχανικής Μάθησης που χρησιμοποιούνται στην εργασία.

Κεφάλαιο 4 : Παρουσιάζονται και αξιολογούνται τα πειραματικά αποτελέσματα της εφαρμογής της μεθόδου και αναλύονται η συμπεριφορά της ως προς τη μεταβολή βασικών παραμέτρων, η συμβατότητα της με διάφορους αλγορίθμους Μηχανικής Μάθησης καθώς και ο τρόπος με τον οποίο ανταποκρίνεται στις τροποποιήσεις της αρχικής αρχιτεκτονικής

Κεφάλαιο 5 : Συνοψίζονται τα συμπεράσματα που προέκυψαν από την παραπάνω μελέτη και αναφέρονται ανοικτά θέματα και μελλοντικοί προσανατολισμοί έρευνας όσον αφορά στην εφαρμογή του μοντέλου γράφων ν-γραμμάτων σε δεδομένα κοινωνικών δικτύων

2

Ανάλυση Συναισθήματος

Τα τελευταία χρόνια η Ανάλυση Συναισθήματος προσελκύει όλο και περισσότερο το ενδιαφέρον της ακαδημαϊκής κοινότητας αλλά και της βιομηχανίας χάρη στις πιθανές εφαρμογές της, κυρίως στον τομέα της Επιχειρηματικής Ευφυίας (Business Intelligence). Στην προσπάθεια να αξιοποιήσουμε αποτελεσματικά τον τεράστιο όγκο δεδομένων που παράγονται καθημερινά από απλούς χρήστες (user-generated content) στα μέσα κοινωνικής δικτύωσης, έχουν πραγματοποιηθεί σημαντικές έρευνες εφαρμόζοντας διαφορετικές τεχνικές και προσεγγίσεις.

2.1 Το Πρόβλημα της Ανάλυσης Συναισθήματος

Από τις πρώτες μελέτες στην περιοχή της Ανάλυσης Συναισθήματος [51] είχε ήδη γίνει αντιληπτό ότι η κατηγοριοποίηση συναισθήματος (sentiment classification) διαφέρει από το κλασικό πρόβλημα κατηγοριοποίησης κειμένου :

"*Η κατηγοριοποίηση κείμενου - γνωστή και ως ταξινόμηση κειμένου ή ανίχνευση θέματος - αναφέρεται στην αντιστοίχηση κειμένων φυσικής γλώσσας σε θεματικές κατηγορίες ή κλάσεις οι οποίες ανήκουν σε ένα προκαθορισμένο σύνολο*" [44]

Οι κατηγορίες καθορίζονται με βάση τα θέματα στόχους του εκάστοτε προβλήματος. Επομένως, διαφορετικά προβλήματα ταξινόμησης κειμένου βασίζονται σε διαφορετικά σύνολα κατηγοριών. Το πλήθος των κατηγοριών σε ένα σύνολο ποικίλει : μπορεί να εκτείνεται από ένα μικρό σύνολο δύο μόνο κατηγοριών έως σύνολα με δεκάδες κατηγορίες π.χ. οι κατηγορίες που απαιτούνται για την ταξινόμηση ενός άρθρου εφημερίδας με βάση τη θεματολογία που καλύπτει. Παράλληλα, ανάλογα με το πρόβλημα και το σύνολο κατηγοριών, ένα κείμενο μπορεί να ανήκει σε μία ή περισσότερες επικαλυπτόμενες κατηγορίες π.χ. ένα άρθρο να αντιστοιχηθεί με τις κατηγορίες "πολιτική", "οικονομία" και "επικαιρότητα".

Αντίθετα, η Ανάλυση Συναισθήματος αναφέρεται σε ένα μικρό σύνολο κατηγοριών (π.χ. θετικό, αρνητικό, ουδέτερο - "1 αστέρι", ...,"5 αστέρια"). Επειδή επικεντρώνεται στην κατάταξη ενός κειμένου ως προς την πολικότητα του, οι κατηγορίες είναι ανεξάρτητες της θεματολογίας του προβλήματος και μεταξύ τους αμοιβαία αποκλειόμενες.

Το πρόβλημα που προσπαθεί να επιλύσει η Ανάλυση Συναισθήματος είναι ένα από τα πιο απλά προβλήματα με τα οποία ασχολείται η Επεξεργασία Φυσικής Γλώσσας [22]. Ο υπολογιστής δε χρειάζεται να αντιλαμβάνεται πλήρως τη σημασιολογία της κάθε πρότασης αλλά θα πρέπει να εντοπίζει τη συνολική στάση του συγγραφέα και να την ταξινομεί ως προς την πολικότητά της. Οι απαιτήσεις απλοποιούν σε μεγάλο βαθμό το πρόβλημα της κατανόησης της φυσικής γλώσσας από τον υπολογιστή αλλά δεν παύει το πρόβλημα της ανίχνευσης της πολικότητας - στο οποίο εξειδικεύεται - να είναι αρκετές φορές δύσκολο ακόμη και για τον άνθρωπο [1].

2.2 Δυσκολίες & Προκλήσεις

Το εννοιολογικό πλαίσιο στο οποίο κινείται η Ανάλυση Συναισθήματος εξασφαλίζει την εφαρμογή των τεχνικών και μεθόδων της σε ένα μεγάλο εύρος θεμάτων χωρίς ιδιαίτερες τροποποιήσεις, επιτυγχάνοντας αρκετά ικανοποιητικά ποσοστά ακρίβειας. Στηριζόμενοι στην ανεξαρτησία του προβλήματος από τη θεματολογία, θα μπορούσαμε να ισχυριστούμε ότι η πολικότητα ενός κειμένου προκύπτει από την πολικότητα των μεμονωμένων λέξεων από τις οποίες απαρτίζεται. Συνεπώς, αναγνωρίζοντας ένα συγκεκριμένο σύνολο λέξεων-κλειδιών (keywords) θα μπορούσαμε να προσδιορίσουμε τη συνολική πολικότητα της άποψης που εκφράζεται στο κείμενο.

Η παραπάνω διαδικασία είναι μία από τις πρώτες μεθόδους που χρησιμοποιήθηκαν και υιοθετεί μία από τις πιο δημοφιλείς και αποτελεσματικές τεχνικές της ανίχνευσης θεματολογίας. Ωστόσο, η προσέγγιση μέσω λέξεων κλειδιών στο συγκεκριμένο πρόβλημα δεν εμφανίζει υψηλά ποσοστά ακρίβειας και έχει αποδεχθεί ελλιπής σε ορισμένες περιπτώσεις ("thwarted expectations" [34]).

Στο σημείο αυτό ανακύπτει το εξής ερώτημα : για ποιο λόγο το πρόβλημα της κατηγοριοποίησης συναισθήματος είναι πιο δύσκολο σε σχέση με την ανίχνευση θεματολογίας, αν λάβουμε υπόψη ότι οι κατηγορίες "θετικό", "αρνητικό" και "ουδέτερο" είναι εννοιολογικά ξένες μεταξύ τους ;

Μία από τις πιο σημαντικές διαφορές με την κατηγοριοποίηση ως προς τη θεματολογία και τις δυσκολίες στην περιοχή της Ανάλυσης Συναισθήματος είναι ότι "το συναίσθημα/άποψη μπορεί πολλές φορές να εκφρασθεί με πιο λεπτό τρόπο χωρίς τη χρήση συναισθηματικά φορτισμένων (θετικά ή αρνητικά) λέξεων με αποτέλεσμα να είναι δύσκολο να αναγνωρισθεί από τους επιμέρους όρους του κειμένου όταν αυτοί εξετάζονται μεμονωμένα" [33].

Παράλληλα, πέρα από τον προσδιορισμό της πολικότητας όταν απουσιάζουν συναισθηματικά φορτισμένες λέξεις, ιδιαίτερα απαιτητικός είναι και ο διαχωρισμός των υποκειμενικών και αντικειμενικών λέξεων και φράσεων ενός κειμένου. Όπως αναφέρεται από τους Kim και Hovy στο [19] "πολλές φορές ακόμη και άνθρωποι διαφωνούν για το αν μία δήλωση αποτελεί ή όχι άποψη".

Ένα άλλο ζήτημα που απασχολεί ιδιαίτερα την Ανάλυση Συναισθήματος είναι ο προσδιορισμός του κατόχου - εκφραστή της άποψης (opinion holder) που διατυπώνεται στο κείμενο. Το συγκεκριμένο θέμα έχει μελετηθεί εκτενώς στη βιβλιογραφία, κυρίως σε

[1] http://mashable.com/2010/04/19/sentiment-analysis/

αναλύσεις σε πολιτικά debates εξετάζοντας αν η γνώμη ανήκει στο συγγραφέα/δημιουργό ή στον σχολιαστή.

Όπως αναφέρθηκε στην Ενότητα 2.1 , η γενικότερη αντίληψη της θετικής ή αρνητικής άποψης δεν εξαρτάται άμεσα από το εκάστοτε θέμα συζήτησης. Ωστόσο, το συναίσθημα και η υποκειμενικότητα ενός κειμένου εξαρτώνται από το σημασιολογικό πλαίσιο στο οποίο τοποθετείται [33]. Χαρακτηριστικό παράδειγμα : "πήγαινε διάβασε το βιβλίο". Η πρόταση εκφράζει θετική άποψη όταν αναφέρεται σε κριτική βιβλίου. Η ίδια πρόταση, όμως, εκφράζει εντελώς διαφορετική άποψη όταν χρησιμοποιείται σε κριτική ταινίας.

Άλλος ένας παράγοντας που επηρεάζει την πολικότητα είναι η σειρά των λέξεων και φράσεων στο κείμενο [33]. Οι ίδιες λέξεις με διαφορετική σειρά μπορεί να οδηγήσουν σε τελείως διαφορετική συνολική πολικότητα.

Τέλος, στις δυσκολίες που συναντά η Ανάλυση Συναισθήματος πρέπει να συμπεριληφθούν και οι προκλήσεις της ευρύτερης περιοχής της Επεξεργασίας Φυσικής Γλώσσας: αμφισημία, χειρισμός της άρνησης, ειρωνεία και σαρκασμός.

2.3 Ανάλυση Συναισθήματος σε Μικρο-Ιστολόγια

Η Ανάλυση Συναισθήματος όταν εφαρμόζεται σε δεδομένα από μικρο-ιστολόγια και κοινωνικά δίκτυα καλείται να αντιμετωπίσει περαιτέρω δυσκολίες οι οποίες οφείλονται στην ιδιαίτερη φύση των κειμένων :

- Μήκος Κειμένου : τα μηνύματα είναι συνήθως σύντομα (π.χ. μέγιστο όριο 140 χαρακτήρες στο Twitter). Αν και ο περιορισμός μήκους μπορεί να οδηγήσει σε περιεκτικές και επί του θέματος τοποθετήσεις, πολλές φορές απουσιάζει το ευρύτερο εννοιολογικό πλαίσιο με αποτέλεσμα να μην είναι σαφής η πολικότητα του κειμένου [6].

- Λεξιλόγιο : τα περισσότερα κείμενα διατυπώνονται σε ανεπίσημη, καθομιλούμενη γλώσσα και εμφανίζουν πολύ μεγαλύτερη ποικιλομορφία σε σχέση με άλλα είδη κειμένου. Περιλαμβάνουν αργκό, νεολογισμούς, εσκεμμένες παραλλαγές λέξεων για έμφαση (επιμήκυνση φθόγγων, χρήση κεφαλαίων γραμμάτων), συντομογραφίες (π.χ. "gr8"-"great") με αποτέλεσμα να μην είναι δυνατή η εφαρμογή λεκτικών αναλυτών ή άλλων εργαλείων που στηρίζονται στη γραπτή και πιο επίσημη μορφή της γλώσσας.

- Θόρυβος : οι πλατφόρμες κοινωνικής δικτύωσης επιτρέπουν μία αυθόρμητη επικοινωνία σε πραγματικό χρόνο όπου πολλές φορές οι χρήστες αναρτούν μηνύματα χωρίς να ελέγχουν για συντακτικά ή γραμματικά λάθη. Ένα μεγάλο ποσοστό από τα δεδομένα που παράγονται περιέχει ακούσια ορθογραφικά λάθη και ακατανόητες εκφράσεις τα οποία συνιστούν ουσιαστικά θόρυβο. Η αναγνώριση και αποκλεισμός τους αποτελεί ιδιαίτερη πρόκληση για τα σύγχρονα συστήματα ανίχνευσης συναισθήματος.

- Πολυγλωσσικό Περιεχόμενο : τα μέσα κοινωνικής δικτύωσης εξαπλώνονται σε μη αγγλόφωνες χώρες, αποκτώντας χρήστες που χρησιμοποιούν και γράφουν σε διαφορετικές γλώσσες, αρκετές φορές ακόμη και σε επίπεδο πρότασης ή μηνύματος. Το φαινόμενο αυτό έχει ως αποτέλεσμα ιδιαίτερα διαδεδομένες τεχνικές στοχευμένες σε συγκεκριμένες γλώσσες (language-specific) να καθίστανται πρακτικά μη εφαρμόσιμες.

2.4 Προσεγγίσεις

Έχουν προταθεί διάφορες τεχνικές που εξετάζουν την αναπαράσταση του κειμένου υπό διαφορετική οπτική γωνία, έχοντας, ωστόσο, κοινό στόχο : τον προσδιορισμό της πολικότητας.Οι πιο βασικές προσεγγίσεις συνοψίζονται στις εξής κατηγορίες :

- Λεξικό Συναισθήματος : βασίζεται στον ισχυρισμό ότι ο προσδιορισμός της πολικότητας ενός κειμένου βασίζεται στον σημασιολογικό προσανατολισμό (semantic orientation) των επιμέρους λέξεων και φράσεών του [51]. Δημιουργούνται, λοιπόν, λεξικά συναισθημάτων στα οποία περιέχονται λέξεις και φράσεις με την αντίστοιχη σημασιολογική πολικότητα και ισχύ τους (βασικά λήμματα). Στη συνέχεια, εμπλουτίζονται είτε αξιοποιώντας πληροφορίες από μεγάλα σώματα κειμένου (ανίχνευση συντακτικών μοτίβων, συχνότητα εμφάνισης λέξεων) (text-corpus based) ή χρησιμοποιώντας εξωτερικούς γλωσσολογικούς πόρους (θησαυρούς λέξεων,ερμηνευτικά λεξικά) (dictionary-based) για την επέκτασή τους με συνώνυμα, αντώνυμα και επιπλέον συντακτικές και σημασιολογικές πληροφορίες [16].

- Σχέσεις και Συνδέσεις : εξετάζει τις πιθανές σχέσεις και εξαρτήσεις μεταξύ των διάφορων χαρακτηριστικών του κειμένου. Μελετά τον τρόπο με τον οποίο συνδέονται μεταξύ τους οι παράγραφοι, οι προτάσεις και τα διάφορα μέρη του λόγου έτσι ώστε να ανιχνευτούν νοηματικές αντιθέσεις ή επικαλύψεις στις συνιστώσες του κειμένου και να προσδιορισθεί με μεγαλύτερη ακρίβεια η συνολική πολικότητα [36].

- Δομή του Λόγου : μελετά τη συντακτική δομή του κειμένου : κάθε λέξη εξετάζεται αν ανήκει στους κύριους όρους της πρότασης (υποκείμενο-ρήμα-αντικείμενο) και εντοπίζεται η θέση της μέσα στο κείμενο με σκοπό να προσδιορισθεί η τοπική της σημασιολογία (ενεργή ή παθητική συμμετοχή) και η βαρύτητά της στο συνολικό συναίσθημα. π.χ. στις κριτικές η συνολική στάση διατυπώνεται συνήθως προς το τέλος του κειμένου [34].

- Γλωσσικά Μοντέλα : δανείζεται αρκετά στοιχεία από την περιοχή της Αναγνώρισης Φωνής. Στηρίζεται στη στατιστική επεξεργασία του κειμένου με στόχο την κατασκευή ενός διανύσματος χαρακτηριστικών (feature vector) το οποίο θα χρησιμοποιηθεί στη συνέχεια για την κατηγοριοποίηση του συναισθήματος [32]. Ένα γλωσσικό μοντέλο αποτελεί την υπό συνθήκη κατανομή πιθανότητας της i-ιοστής λεκτικής μονάδας σε μία πρόταση , δηλαδή υποδηλώνει την πιθανότητα εμφάνισης της συγκεκριμένης λεκτικής μονάδας, γνωρίζοντας την κατηγορία όλων των προηγούμενων όρων της πρότασης.Τα πιο δημοφιλή μοντέλα βασίζονται στην αναπαράσταση του κειμένου με λέξεις ή χαρακτήρες ν-γραμμάτων. Η προσέγγιση αυτή

διαφέρει σε σχέση με τις προαναφερθείσες καθώς απαιτεί ένα σύνολο από ήδη ταξινομημένα εκπαιδευτικά πρότυπα (training set) - τα οποία πρέπει να είναι αντιπροσωπευτικά των κειμένων προς εξέταση (test set). Εφαρμόζοντας το μοντέλο στο σύνολο εκπαίδευσης, επιλέγουμε τα χαρακτηριστικά εκείνα που καθιστούν τα κείμενα διαφορετικών κατηγοριών διαχωρίσιμα.

2.5 Κατηγοριοποίηση

Η οπτική γωνία με την οποία προσεγγίζεται η δομή και αναπαράσταση ενός κειμένου καθορίζει και τον τρόπο κατηγοριοποίησης του. Οι τεχνικές που εφαρμόζονται χωρίζονται ανάλογα με το βαθμό παρέμβασης του ανθρώπου στη διαδικασία της μάθησης σε δύο βασικές κατηγορίες :

2.5.1 Μέθοδοι Επιβλεπόμενης Μάθησης

Αποτελεί τη πιο δημοφιλή τεχνική κατηγοριοποίησης συναισθήματος. Στόχος είναι η δημιουργία ενός ταξινομητή (classifier) ο οποίος θα αντιστοιχίζει κείμενα με κατηγορίες (θετικά, αρνητικά, ουδέτερα) εφαρμόζοντας κάποιον αλγόριθμο.

Στην επιβλεπόμενη μάθηση κάθε κείμενο αναπαρίσταται με ένα διάνυσμα χαρακτηριστικών έτσι ώστε ο ταξινομητής να αναγνωρίσει και να μάθει τις πιο αντιπροσωπευτικές διαφορές ανάμεσα σε κείμενα που ανήκουν σε διαφορετικές κατηγορίες και γι'αυτό απαιτείται ένα σύνολο εκπαίδευσης. Οι αλγόριθμοι μηχανικής μάθησης εστιάζουν στη βελτιστοποίηση των εσωτερικών τους παραμέτρων ανάλογα με τη δυαδική τιμή ή βάρος ορισμένων χαρακτηριστικών ή στην κατασκευή επαγώμενων κανόνων ανάλογα με το ζεύγος χαρακτηριστικό-τιμή στο σύνολο εκπαίδευσης. Οι πιο γνωστοί αλγόριθμοι είναι: Naive Bayes, Multinomial Naive Bayes, C4.5 και Support Vector Machines (SVM).

Οι τεχνικές επιβλεπόμενης μάθησης επιτυγχάνουν υψηλά ποσοστά ακρίβειας και υπερτερούν των μη-επιβλεπόμενων τεχνικών [34]. Ωστόσο, εμφανίζουν κάποια μειονεκτήματα. Απαιτείται αρκετός χρόνος και προσπάθεια για την κατασκευή ενός συνόλου εκπαίδευσης αλλά και για την εκπαίδευση του ταξινομητή μέχρις ότου βρεθούν οι βέλτιστες τιμές των παραμέτρων ή εξαχθούν οι απαραίτητοι κανόνες. Παράλληλα, η ακρίβεια ενός ταξινομητή εξαρτάται άμεσα από το σύνολο εκπαίδευσης. Συνεπώς, τα εκπαιδευτικά πρότυπα θα πρέπει να έχουν επιλεγεί κατάλληλα έτσι ώστε να είναι αντιπροσωπευτικά του συνολικού πληθυσμού των κειμένων.

2.5.2 Μέθοδοι Μη-Επιβλεπόμενης Μάθησης

Η κατηγοριοποίηση συναισθήματος γίνεται με βάση το σημασιολογικό προσανατολισμό των λέξεων και φράσεων του κειμένου. Δεν απαιτείται σύνολο εκπαίδευσης για την εξαγωγή διανύσματος χαρακτηριστικών. Αντίθετα, χρησιμοποιώντας προ-κατασκευασμένα λεξικά συναισθήματος, χαρακτηρίζονται οι διάφοροι όροι του κειμένου και προκύπτει η συνολική πολικότητα.

Αρκετές από τις μη-επιβλεπόμενες μεθόδους επιτυγχάνουν ικανοποιητικά ποσοστά ακρίβειας σε συστηματική βάση όταν εφαρμόζονται σε γνωστά θεματικά πεδία όπου το λεξιλόγιο των κειμένων τους καλύπτεται από τα λεξικά συναισθήματος. Αποκτούν ιδιαίτερη δημοτικότητα καθώς δεν απαιτείται σύνολο εκπαίδευσης με αποτέλεσμα να μπορούν να εφαρμοστούν σε μεγαλύτερο εύρος θεμάτων σε σχέση με τις μεθόδους επιβλεπόμενης μάθησης [47]. Ωστόσο, παρουσιάζουν δύο σημαντικούς περιορισμούς [39]. Αρχικά, το πλήθος των λέξεων στα λεξικά είναι πεπερασμένο με αποτέλεσμα η τεχνική να μην μπορεί να εφαρμοστεί σε πολύ δυναμικά περιβάλλοντα όπως το Twitter όπου νεολογισμοί και συντομογραφίες συνεχώς εμφανίζονται. Επιπλέον, τα λεξικά συναισθήματος αναθέτουν συνήθως ένα σταθερό συναισθηματικό προσανατολισμό στις λέξεις χωρίς να εξετάζουν το ευρύτερο πλαίσιο στο οποίο χρησιμοποιούνται.

2.6 Σχετικές Εργασίες

Οι πρώτες μελέτες στην περιοχή της Ανάλυσης Συναισθήματος εξέταζαν κυρίως άρθρα εφημερίδων (κριτικές ταινιών και προϊόντων,πολιτικές και οικονομικές αναλύσεις). Τα τελευταία χρόνια, χάρη στην ανάπτυξη των μέσων κοινωνικής δικτύωσης, το ενδιαφέρον της ακαδημαϊκής κοινότητας αλλά και της βιομηχανίας στράφηκε προς την επεξεργασία και ανάλυση των παραγόμεων δεδομένων. Αυτό είχε ως αποτέλεσμα να παραχθεί σημαντικό ερευνητικό έργο που εστιάζει αποκλειστικά σε δεδομένα από κοινωνικά δίκτυα. Ακολουθεί μία επισκόπηση των κύριων εργασιών σε δεδομένα από το Twitter, χωρισμένες σε ενότητες ανάλογα με το είδος της τεχνικής που εφαρμόζουν.

2.6.1 Εργασίες με χρήση Επιβλεπόμενης Μηχανικής Μάθησης

Οι **Go et al. (2009)** [14] υπήρξαν από τους πρώτους που μελέτησαν την ανάλυση συναισθήματος σε δεδομένα από το Twitter. Στη μελέτη τους ασχολούνται με τη δυαδική εκδοχή του προβλήματος κατηγοριοποίησης συναισθήματος , χαρακτηρίζοντας τα tweets ως θετικά ή αρνητικά. Λόγω της έλλειψης σε εκπαιδευτικά πρότυπα ήδη κατηγοριοποιημένα χειροκίνητα από άνθρωπο (manually annotated), εφαρμόζουν την τεχνική της εξ αποστάσεως επίβλεψης (distant supervision) για να εκπαιδεύσουν ένα ταξινομητή επιβλεπόμενης μηχανικής μάθησης. Μέσω του Twitter API, συλλέγουν ένα μεγάλο σύνολο από tweets τα οποία ταξινομούν αυτόματα σε κατηγορίες ανάλογα με τα emoticons (noisy labels) , διαγράφοντας tweets που περιέχουν emoticons και από τις δύο κατηγορίες. Το τελικό training set αποτελείται από 1.6 εκατομμύρια tweets, 800 χιλιάδες tweets από κάθε κατηγορία. Εφαρμόζουν στάδιο προεπεξεργασίας του αρχικού κειμένου των tweets όπου αφαιρούνται τα emoticons ενώ αναφορές σε χρήστες (@username) και υπερσύνδεσμοι αντικαθίστανται με κατάλληλες λέξεις-κλειδιά (placeholders). Για την κατηγοριοποίηση χρησιμοποιούν ως χαρακτηριστικά μονογράμματα, διγράμματα, συνδυασμό μονογραμμάτων και διγραμμάτων καθώς και επισημειώσεις για την ιδιότητα της κάθε λέξης (μέρος του λόγου), γνώρισμα που συναντάται στην βιβλιογραφία με τον όρο POS (part-of-speech) tags. Συγκρίνουν τους αλγορίθμους Naive Bayes, Maximum Entropy και Support Vector Machines (SVM). Η χρήση των SVM με μοναδικό χαρακτηριστικό τα μονογράμματα αποφέρει το καλύτερο αποτέλεσμα (82.9 %). Παρατηρούν

πως η προσθήκη των διγραμμάτων στο διάνυσμα χαρακτηριστικών βελτιώνει την επίδοση των NaiveBayes και Maximum Entropy αλλά όχι των SVM. Τέλος, καταλήγουν στο συμπέρασμα ότι προσθέτοντας την άρνηση (negation) ως ξεχωριστό χαρακτηριστικό καθώς και τα POS tags δεν παρατηρείται βελτίωση ενώ η χρήση μόνο των διγραμμάτων οδηγεί σε χειρότερα αποτελέσματα εξαιτίας του αραιού χώρου χαρακτηριστικών (feature space).

Οι **Pak & Paroubek (2010)** [31] χρησιμοποιούν επίσης θετικά και αρνητικά emoticons για να δημιουργήσουν ένα σύνολο εκπαίδευσης με 300 χιλιάδες tweets. Ωστόσο, συλλέγουν παράλληλα tweets από λογαριασμούς εφημερίδων στο Twitter για να τα χρησιμοποιήσουν ως ουδέτερα πρότυπα και να μελετήσουν το γενικότερο πρόβλημα κατηγοριοποίησης με τις τρεις κλάσεις. Στο στάδιο της προεπεξεργασίας τους, αφαιρούν τα ονόματα χρηστών, τα emoticons, τους υπερσυνδέσμους και τα άρθρα (a, an, the) (stopwords), οι λέξεις άρνησης (no, not) συνενώνονται με την προηγούμενη ή επόμενη λέξη και το κείμενο κατακερματίζεται στα κενά και τα σημεία στίξης (tokenization). Πειραματίζονται με μονογράμματα, διγράμματα και τριγράμματα. Κατασκευάζουν δύο εκδοχές του ταξινομητή Naive Bayes χρησιμοποιώντας διαφορετικά χαρακτηριστικά. Ο ένας στηρίζεται στην παρουσία ενός ν-γράμματος στο κείμενο (χαρακτηριστικό με δυαδική τιμή). Ο άλλος βασίζεται στην πληροφορία κατανομής των μερών του λόγου (POS) για να εκτιμήσει την παρουσία των POS tags και να υπολογίσει την εκ των υστέρων πιθανότητα (posterior probability) του μοντέλου Naive Bayes. Θεωρώντας πως τα δύο χαρακτηριστικά είναι υπό συνθήκη ανεξάρτητα και κατ' επέκταση και οι δύο ταξινομητές, η τελική ταξινόμηση γίνεται με βάση το λογάριθμο πιθανοφάνειας (log-likelihood). Παρατηρούν ότι το συγκεκριμένο μοντέλο υπερτερεί των Support Vector Machines (SVM) και των Conditional Random Fields (CRF) έχοντας καλύτερο αποτέλεσμα στον όχι και τόσο γνωστό δείκτη $F_{0.5} = 0.63$. Συμπεραίνουν πως τα διγράμματα πετυχαίνουν τη καλύτερη ακρίβεια γιατί "αποτελούν μία καλή ισορροπία ανάμεσα στην κάλυψη των εύρους (μονογράμματα) και στην ικανότητα αναγνώρισης συναισθηματικών μοτίβων έκφρασης (τριγράμματα)" [31].

Οι **Barbosa & Feng (2010)** [5] προτείνουν ένα ταξινομητή δύο φάσεων. Στην πρώτη φάση, τα tweets κατηγοριοποιούνται σε υποκειμενικά ή αντικειμενικά και στη συνέχεια τα υποκειμενικά διακρίνονται σε θετικά ή αρνητικά tweets. Ακολουθούν μία διαφορετική προσέγγιση για την κατασκευή του συνόλου εκπαίδευσης : χρησιμοποιούν ως noisy labels όχι τα emoticons αλλά τη "γνώμη" τριών εργαλείων ανίχνευσης συναισθήματος : Twendz[2], Twitter Sentiment[3] και TweetFeel[2], διαγράφοντας τα tweets στα οποία δεν υπάρχει ομόφωνη απόφαση. Το τελικό σύνολο εκπαίδευσης περιλαμβάνει 200 χιλιάδες tweets για ανίχνευση υποκειμενικότητας και 71.046 θετικά και 79.628 αρνητικά tweets για κατηγοριοποίηση πολικότητας. Διαχωρίζουν τα χαρακτηριστικά τους σε δύο κατηγορίες : τα μέτα-χαρακτηριστικά (meta-features) και τα χαρακτηριστικά σύνταξης του tweets (tweet-syntax). Η πρώτη κατηγορία περιλαμβάνει χαρακτηριστικά όπως POS tags, ο βαθμός υποκειμενικότητας και πολικότητας της εκάστοτε λέξης όπως αυτά προσδιορίζονται στο λεξικό MPQA [54]. Η δοθείσα πολικότητα αντιστρέφεται από θετική σε αρνητική και αντίστροφα όταν μία άρνηση προηγείται της λέξης. Η δεύτερη κατηγορία περιλαμβάνει πιο ειδικά για το Twitter χαρακτηριστικά όπως η ύπαρξης retweets, hashtags, υπερσυνδέσμων, θαυμαστικών, ερωτηματικών, emoticons και κεφαλαίων γραμμάτων. Η συχνότητα κάθε χαρακτηριστικού κανονικοποιείται διαιρώντας με το πλήθος

[2] Η υπηρεσία δεν είναι πλεόν διαθέσιμη
[3] http://www.sentiment140.com/

των όρων του κάθε tweet. Συνολικά και από τις 2 κατηγορίες προκύπτουν 20 χαρακτηριστικά. Τα καλύτερα αποτελέσματα προκύπτουν χρησιμοποιώντας ως ταξινομητή τον SVM και στις δύο φάσεις επιτυγχάνουν δε ακρίβεια 81.9 % στην αναγνώριση υποκειμενικότητας και 81.3 % στην αναγνώριση πολικότητας ενώ ως βάση αναφοράς θεωρούνται τα μονογράμματα με ακρίβεια 72.4 % και 79.1 % αντίστοιχα στις δύο φάσεις. Παρατηρούν ότι τα μέτα-χαρακτηριστικά είναι πιο σημαντικά στη φάση προσδιορισμού της πολικότητας ενώ τα χαρακτηριστικά σύνταξης κατά την φάση αναγνώρισης της υποκειμενικότητας. Οι συγγραφείς καταλήγουν στο συμπέρασμα ότι επειδή χρησιμοποιούν μια πιο αφηρημένη αναπαράσταση των δεδομένων και όχι μεμονωμένους όρους του, η προσέγγισή τους εμφανίζει μεγαλύτερη ανοχή στο θόρυβο και μεροληψία (bias) του συνόλου εκπαίδευσης σε σχέση με άλλες μεθόδους ενώ εμφανίζει καλύτερη συμπεριφορά ως προς την ικανότητας γενίκευσης όταν χρησιμοποιούνται σχετικά λίγα εκπαιδευτικά πρότυπα.

Οι **Bermingham & Smeaton (2010)** [6] εξετάζουν την επίδραση του μικρού μήκους των tweets στις συνήθεις τεχνικές επιβλεπόμενης μάθησης. Συλλέγουν tweets από τα δέκα πιο δημοφιλή θέματα (trending) σε πέντε κατηγορίες (ψυχαγωγία, προϊόντα & υπηρεσίες, αθλητικά, επικαιρότητα και εταιρείες) δημιουργώντας ένα σύνολο από χειροκίνητα κατηγοριοποιημένα πρότυπα (1.410 θετικά, 1.040 αρνητικά και 2.597 ουδέτερα tweets). Κατά την προεπεξεργασία των δεδομένων αντικαθιστούν τα ονόματα χρηστών, υπερσυνδέσμους και hashtags με προκατασκευασμένες λέξεις-κλειδιά. Ως χαρακτηριστικά αναπαράστασης του κειμένου χρησιμοποιούνται μονογράμματα, διγράμματα, τριγράμματα, POS tags και POS ν-γράμματα. Συγκρίνουν τα αποτελέσματα κατηγοριοποίησης από την εφαρμογή της μεθόδου σε tweets, κριτικές ταινιών και αναρτήσεις ιστολογίων. Παρατηρούν ότι ο Naive Bayes εμφανίζει υψηλότερα ποσοστά ακρίβειας σε σχέση με τα Support Vector Machines στη περίπτωση των tweets αλλά όχι σε μεγαλύτερου μήκους κείμενα (κριτικές και αναρτήσεις ιστολογίων). Χρησιμοποιώντας Naive Bayes και μονογράμματα επιτυγχάνουν ακρίβεια 74.85 % στο πρόβλημα της δυαδικής κατηγοριοποίησης και 61.3 % στο γενικότερο πρόβλημα των τριών κλάσεων. Η χρήση ν-γραμμάτων και POS tags βελτιώνει την ακρίβεια μόνο στην περίπτωση των μεγάλων κειμένων ενώ τα POS ν-γράμματα, η επίλυση συνωνύμων (stemming) και η αφαίρεση κοινών λέξεων (stopwording) δεν οδηγούν σε καλύτερα αποτελέσματα. Συμπεραίνουν ότι η ανάλυση συναισθήματος σε κείμενα μικρού μήκους όπως τα tweets είναι εν γένει πιο εύκολο πρόβλημα.

Οι **Bifet & Frank (2010)** [7] ασχολούνται με την ανάλυση συναισθήματος σε μεγάλη ροή δεδομένων (data stream) από το Twitter. Προτείνουν ένα Kappa στατιστικό δείκτη κυλιόμενου παραθύρου για να αξιολογήσουν την επίδοση κατηγοριοποίησης σε χρονομεταβλητές ροές δεδομένων. Χρησιμοποιούν το σύνολο δεδομένων (dataset) Stanford Twitter Sentiment των Go et al. [14] και το Edinburgh Twitter Corpus των Petrovic et al. [35], χρησιμοποιώντας τα emoticons ως δείκτες αυτόματης ταξινόμησης (noisy labels). Κατά την προεπεξεργασία των δεδομένων, αντικαθιστούν τα ονόματα χρηστών και τους υπερσυνδέσμους με κατάλληλες λέξεις-κλειδιά ενώ ως χαρακτηριστικά χρησιμοποιούν μόνο τα μονογράμματα. Παράλληλα με την αξιολόγηση μέσω του προτεινόμενου δείκτη, αναφέρουν ως καλύτερα αποτελέσματα για το πρόβλημα της δυαδικής ταξινόμησης 82.45 % στο πρώτο dataset με χρήση Naive Bayes και 86.26 % στο δεύτερο σύνολο tweets με χρήση Στοχαστικής Κλίσης Καθόδου (Stochastic Gradient Descent - SGD). Παρατηρούν πως οι Naive Bayes και SGD παρουσιάζουν παρόμοια ποσοστά ακρίβειας σε αντίθεση με τα δένδρα Hoeffing τα οποία υστερούν συστηματικά. Επομένως, συνιστούν

να αποφεύγονται γενικά ταξινομητές - δένδρα στην περίπτωση μεγάλης ροής δεδομένων και προτείνουν έναντι τη χρήση SGD καθώς προσαρμόζονται καλύτερα στις αλλαγές με τη πάροδο του χρόνου και παράλληλα οι αλλαγές στα βάρη των χαρακτηριστικών μπορούν να χρησιμοποιηθούν για να παρακολουθούνται αλλαγές στο συναίσθημα και απόψεις γύρω από συγκεκριμένα θέματα.

Οι **Davidov et al. (2010)** [9] ταξινομούν αυτόματα το σύνολο δεδομένων των O'Connor et al. [29] χρησιμοποιώντας ως δείκτες κατηγοριοποίησης (noisy labels) 50 hashtags και 15 emoticons. Το σύνολο χαρακτηριστικών περιλαμβάνει : λέξεις, ν-γράμματα (2-5), μήκος του κάθε tweet, πλήθος σημείων στίξης, θαυμαστικών, ερωτηματικών, εισαγωγικών, κεφαλαίων γραμμάτων και λέξεων καθώς και την ύπαρξη λέξεων με υψηλή συχνότητα εμφάνισης. Εφαρμόζοντας μία τεχνική παρόμοια με αυτή των k-Κοντινότερων Γειτόνων (k-Nearest Neighbours - kNN) επιτυγχάνουν βέλτιστη ακρίβεια στο μέσο αρμονικό δείκτη $F_1 = 0.86$ στην περίπτωση των emoticons και $F_1 = 0.8$ στην περίπτωση των hashtags για τη δυαδική εκδοχή του προβλήματος μέσω 10-πλης σταυρωτής επικύρωσης (10-fold cross-validation). Στο γενικότερο πρόβλημα των τριών κλάσεων, η επίδοση ήταν αισθητά χαμηλότερη (0.64 και 0.31 αντίστοιχα). Παράλληλα, προτείνουν δύο διαφορετικές μεθόδους για την αυτόματη ανίχνευση της επικάλυψης συναισθήματος και των αλληλεξαρτήσεων ανάμεσα στις λέξεις του κειμένου. Παρατηρούν ότι οι λέξεις, τα σημεία στίξης και τα εκφραστικά μοτίβα είναι τα πιο σημαντικά χαρακτηριστικά ενώ τα ν-γράμματα οδηγούν σε οριακή βελτίωση.

Σε αντίθεση με τις περισσότερες μελέτες, οι **Agarwal et al. (2011)** [1] δεν περιορίζουν τη συλλογή tweets μόνο σε αυτά της αγγλικής γλώσσας μέσω του Twitter API αλλά χρησιμοποιούν την υπηρεσία Google Translate για μετάφρασή τους. Δημιουργούν ένα σύνολο από 8753 χειροκίνητα ταξινομημένα tweets στις τρεις κατηγορίες (θετικά, αρνητικά, ουδέτερα) αφού πρώτα διέγραψαν όσα περιείχαν λάθη λόγω μετάφρασης. Για την αξιολόγηση των μεθόδων τους, δημιουργούν ένα ισοζυγισμένο σύνολο δεδομένων περιλαμβάνοντας 1709 πρότυπα από κάθε κατηγορία, 5127 tweets συνολικά. Προτείνουν δύο νέες τεχνικές στο στάδιο της προεπεξεργασίας : δημιουργούν ένα λεξικό emoticons το οποίο περιέχει 170 emoticons όπως καταγράφονται στη Wikipedia χωρισμένα σε πέντε κατηγορίες ανάλογα με το συναίσθημα που εκφράζουν (υπερβολικά θετικό, θετικό, ουδέτερο, αρνητικό, υπερβολικά αρνητικό) και κατασκευάζουν ένα λεξικό με τη μετάφραση 5184 ακρωνύμιων . Έπειτα, αντικαθιστούν τα emoticons με την συναισθηματική πολικότητα στο λεξικό ,τα ακρωνύμια με την κανονική τους μορφή καθώς και τα ονόματα χρηστών, τους υπερσυνδέσμους, τα hashtags και τις αρνήσεις με γνωστές λέξεις-κλειδιά και περιορίζουν τους επιμηκυμένους χαρακτήρες σε δύο π.χ. το *cooooooooool* σε *coool*. Αρκετά από τα χαρακτηριστικά που χρησιμοποιούν βασίζονται στη πρότερη πολικότητα των λέξεων την οποία προσδιορίζουν χρησιμοποιώντας το Dictionary of Affect in Language (DAL) [53] και το επεκτείνουν με συνώνυμα από το WordNet [10]. Στο σύνολο των χαρακτηριστικών περιλαμβάνονται το πλήθος, η συχνότητα και το ποσοστό των λέξεων, άρθρων (stopwords), αγγλικών λέξεων, σημείων στίξης, θαυμαστικών, tags, αρνήσεων και κεφαλαίων τα οποία υπολογίζονται για όλο και για το τελευταίο τρίτο του tweet. Συγκρίνουν πέντε διαφορετικά μοντέλα τα οποία στηρίζονται στα Support Vector Machines (SVM) και θέτουν ως βάση αναφοράς τα μονογράμματα. Παρατηρούν ότι ταξινομητές οι οποίοι στηρίζονται μόνο σε αφηρημένα γλωσσικά χαρακτηριστικά αποδίδουν εξίσου καλά με τα μονογράμματα τα οποία χρησιμοποιούν πολύ περισσότερα χαρακτηριστικά. Στο πρόβλημα της δυαδικής κατηγοριοποίησης επιτυγχάνουν βέλτιστη ακρίβεια 75.39 % με μοντέλο που συνδυάζει μονογράμματα και αφηρημένα γλωσσικά

χαρακτηριστικά. Στο πρόβλημα των τριών κλάσεων το μοντέλο με τα καλύτερα ποσοστά ακρίβειας (60.83 %) συνδυάζει αφηρημένα γλωσσικά χαρακτηριστικά μαζί με μία ειδική δενδρική αναπαράσταση του κάθε όρου σε SVM με partial tree kernel [26]. Παρατηρούν ότι τα αφηρημένα γλωσσικά χαρακτηριστικά με τη περισσότερη πληροφορία είναι εκείνα τα οποία συνδυάζουν την πρότερη πολικότητα των λέξεων μαζί με τα POS tags. Καταλήγουν στο συμπέρασμα ότι η ανάλυση συναισθήματος σε δεδομένα από το Twitter δε διαφέρει από την ανάλυση συναισθήματος σε άλλα είδη κείμενου.

Οι **Jiang et al. (2011)** [18] εξετάζουν την ανάλυση συναισθήματος σε συγκεκριμένα θέματα στόχους (target-dependent) εφαρμόζοντας μία τεχνική τριών σταδίων. Όμοια με τους Barbosa και Feng [5], πρώτα ταξινομούν τα tweets σε υποκειμενικά και αντικειμενικά και στη συνέχεια ($2^{η}$ φάση) τα υποκειμενικά tweets σε θετικά και αρνητικά χρησιμοποιώντας δύο ξεχωριστούς ταξινομητές Support Vector Machines (SVM) με γραμμική συνάρτηση πυρήνα. Υποστηρίζουν ότι συνήθεις τεχνικές ([5, 14]) δεν επαρκούν καθώς όλα τα χαρακτηριστικά είναι ανεξάρτητα του στόχου. Στο τρίτο στάδιο προτείνουν μία μέθοδο η οποία βασίζεται σε γράφους με σκοπό να αυξήσουν την ακρίβεια : εξετάζουν το ευρύτερο εννοιολογικό πλαίσιο που τοποθετείται το κάθε tweet μέσω των συσχετιζόμενων με αυτό tweets όπως retweets, tweets που περιέχουν την ίδια οντότητα-στόχο και προέρχονται από τον ίδιο χρήστη καθώς και τις πιθανές απαντήσεις από ή στο εκάστοτε tweet. Μέσω του Twitter API συλλέγουν tweets που περιέχουν 5 δημοφιλείς οντότητες : Obama, Google, iPad, Lakers, Lady Gaga και τα ταξινομούν χειροκίνητα στις 3 κατηγορίες δημιουργώντας τελικά ένα σύνολο από 459 θετικά, 268 αρνητικά και 1212 ουδέτερα tweets. Κατά το στάδιο της προεπεξεργασίας με τη βοήθεια εξωτερικών εργαλείων εφαρμόζουν τεχνικές κανονικοποίησης των κειμένων (διόρθωση απλών ορθογραφικών λαθών και εμφατικής επιμήκυνσης λέξεων), επίλυσης συνωνύμων (stemming), γραμματικής αναγνώρισης (POS tagging) και συντακτικής ανάλυσης έτσι ώστε να κατασκευάσουν χαρακτηριστικά ειδικά για τις εξεταζόμενες οντότητες. Παράλληλα, υπολογίζουν και χαρακτηριστικά ανεξάρτητα του στόχου μέσω μονογραμμάτων και του λεξικού General Inquirer [4]. Τέλος, επιλύουν τις έμμεσες αναφορές αναζητώντας τις K πιο ισχυρά συσχετιζόμενες με τους στόχους λέξεις και φράσεις μέσω του δείκτη PMI (Pointwise Mutual Inforation). Παρατηρούν πως ο συνδυασμός των χαρακτηριστικών ειδικών του στόχου μαζί με άλλα χαρακτηριστικά οδηγεί σε καλύτερη επίδοση 68.2 % ξεπερνώντας κατά 7.9 % την δική τους υλοποίηση της εκδοχής των Barbosa και Feng. Αναφέρουν ως πιθανή αιτία το γεγονός ότι οι Barbosa και Feng χρησιμοποιούν πιο αφηρημένα χαρακτηριστικά ενώ η δική τους προσέγγιση στηρίζεται περισσότερο σε λεξιλογικά χαρακτηριστικά. Συμπεραίνουν πως τα χαρακτηριστικά εξαρτώμενα από τους στόχους συντελούν καθοριστικά ιδίως σε περιπτώσεις όπου το συναίσθημα δεν αναφέρεται στην πραγματικότητα στην οντότητα-στόχο.

Οι **Kouloumpis et al. (2011)** [20] ερευνούν την συμβολή των γλωσσολογικών χαρακτηριστικών στην αναγνώριση πολικότητας των tweets. Χρησιμοποιούν δύο γνωστά datasets και επιλέγουν διαφορετικό δείκτη αυτόματης κατηγοριοποίησης (noisy labels) για κάθε ένα. Εξετάζουν το Endinburgh Twitter Corpus των Petrovic et al. [35] μέσω hashtags ενδεικτικών του συναισθήματος (π.χ. *#imthankfulfor, #ihate, #news*) και το Stanford Twitter Sentiment Corpus των Go et al. [14] μέσω emoticons. Όμοια με τις περισσότερες μελέτες, στο στάδιο της προεπεξεργασίας αντικαθιστούν τα ονόματα χρηστών, τους υπερσυνδέσμους και τα hashtags με κατάλληλες λέξεις-κλειδιά, τις συντομογραφίες με την κανονικής τους μορφή και διορθώνουν την ορθογραφία των λέξεων από

[4] http://www.wjh.harvard.edu/~inquirer/

εμφατική επιμήκυνση και χρήση κεφαλαίων γραμμάτων. Αφαιρούν επίσης κοινές λέξεις και άρθρα (stopwording) και αναγνωρίζουν γραμματικά την κάθε λέξη (POS tagging). Χρησιμοποιούν ένα αρκετά μεγάλο σύνολο χαρακτηριστικών : μονογράμματα, διγράμματα, τα πρώτα 1000 μονογράμματα και διγράμματα με βάση το κέρδος πληροφορία κατά το δείκτη Chi-squared, τη πρότερη πολικότητα των λέξεων κατά το MPQA λεξικό [54], το πλήθος και ποσοστό των κυριότερων POS tags και δυαδικά χαρακτηριστικά μικροιστολογίων για την ύπαρξη εξειδικευμένων όρων (hashtags, emoticons). Παρατηρούν ότι η χρήση του ταξινομητή AdaBoost υπερτερεί των Support Vector Machines (SVM) έχοντας βέλτιστη επίδοση 75% στο πρόβλημα των τριών κλάσεων με χρήση όλων των χαρακτηριστικών εκτός του πλήθους των POS tags. Συμπεραίνουν ότι σε αντίθεση με τα χαρακτηριστικά μικροιστολογίων που ήταν τα πιο χρήσιμα, τα POS tags οδηγούν σε μείωση της ακρίβειας και δεν είναι μάλλον κατάλληλα για χρήση σε κείμενα από μικροιστολόγια.

Οι **Saif et al. (2011)** [41] μελετούν το πρόβλημα της αραιότητας των δεδομένων λόγω του μικρού μήκους των μηνυμάτων του Twitter. Προτείνουν δύο διαφορετικές προσεγγίσεις της σημασιολογικής εξομάλυνσης (semantic smoothing) με σκοπό να εξάγουν σημασιολογικά κρυμμένες έννοιες από τα κείμενα και να τις χρησιμοποιήσουν ως επιπρόσθετα χαρακτηριστικά για την εκπαίδευση των ταξινομητών. Εξετάζουν ένα ισοζυγισμένο υποσύνολο 60 χιλιάδων tweets από το Standford Twitter Sentiment Corpus των Go et al. [14] μαζί με το σύνολο εξέτασης με 177 αρνητικά και 182 θετικά χειροκίνητα ταξινομημένα tweets. Για την εξαγωγή των εννοιών χρησιμοποιούν την υπηρεσία AlchemyAPI[5] όπου αναγνωρίζουν γνωστές οντότητες στα κείμενα των tweets. Στη πρώτη μέθοδο (shallow semantic smoothing) οι λέξεις αντικαθίστανται με τις αντίστοιχες σημασιολογικές τους έννοιες ενώ στη δεύτερη (interpolation) το γλωσσικό μοντέλο μονογράμματος παρεμβάλλεται μαζί ένα παραγωγικό μοντέλο λέξεων δοθέντων των σημασιολογικών εννοιών στον ταξινομητή Naive Bayes. Παρατηρούν πως ενώ η πρώτη μέθοδος οδηγεί σε μείωση της ακρίβειας κατά 5% σε σχέση με ένα ταξινομητή Naive Bayes με μόνο χαρακτηριστικό τα μονογράμματα, η μέθοδος της παρεμβολής οδηγεί σε οριακή βελτίωση επιτυγχάνοντας ακρίβεια 81.3% στη δυαδική κατηγοριοποίηση. Οι παραπάνω προσεγγίσεις βελτιώνονται στο [42] όπου προστίθεται προεπεξεργασία κειμένου: αντικατάσταση ονομάτων χρηστών, υπερσυνδέσμων με λέξεις-κλειδιά, διόρθωση της επιμήκυνσης φθόγγων, αφαίρεση hashtags, emoticons, μονών χαρακτήρων, ψηφίων και άλλων μη αλφαριθμητικών χαρακτήρων. Επεκτείνουν το αρχικό σύνολο εξέτασης σε 100 tweets και επιτυγχάνουν ακρίβεια 84% με βελτιωμένη έκδοση της μεθόδου παρεμβολής. Στο [43] οι συγγραφείς εξετάζουν τις μεθόδους σε δύο ακόμη σύνολα δεδομένων: HealthCare Reform (HCR) [46] και Obama McCain Debate [45] με καλύτερα ποσοστά ακρίβειας 79% και 69,15% αντίστοιχα. Καταλήγουν στο συμπέρασμα ότι η τεχνική της σημασιολογικής εξομάλυνσης εμφανίζει καλύτερα αποτελέσματα σε μεγάλα σύνολα δεδομένων που περιέχουν ποικίλη θεματολογία.

Οι **Liu et al. (2012)** [23] προτείνουν μία νέα προσέγγιση για τη συνένωση χειροκίνητα και αυτόματα μέσω noisy labels ταξινομημένων tweets χρησιμοποιώντας το γλωσσικό μοντέλο ESLM (Emoticon Smoothed Language Model). Αρχικά, εκπαιδεύουν ένα γλωσσικό μοντέλο με χειροκίνητα ταξινομημένα πρότυπα από το Sanders Corpus[6] (570 θετικά, 654 αρνητικά, 2503 ουδέτερα tweets). Στη συνέχεια, μέσω του Twitter API συλλέγουν tweets που περιέχουν emoticons με σκοπό να εξομαλύνουν το γλωσσικό

[5] http://www.alchemyapi.com
[6] http://www.sananalytics.com/lab/twitter-sentiment/

μοντέλο. Στο στάδιο της προεπεξεργασίας αντικαθιστούν ονόματα χρηστών, υπερσυνδέσμους και ψηφία με λέξεις-κλειδιά, διαγράφουν κοινές λέξεις (stopwording), αντικαθιστούν συνώνυμα (stemming) και κεφαλαία με πεζά γράμματα ενώ διαγράφουν retweets και διπλότυπα από το αρχικό dataset. Επίσης, ξεχωρίζουν υπερσυνδέσμους που αναφέρονται σε εικόνες/video από τα υπόλοιπα URLs. Παρατηρούν ότι το προτεινόμενο μοντέλο εμφανίζει πολύ καλύτερη συμπεριφορά συγκρινόμενο με ένα γλωσσικό μοντέλο πλήρως επιβλεπόμενο επιτυγχάνοντας ακρίβεια 82.5% και 79.5% στην αναγνώριση πολικότητας και υποκειμενικότητας αντίστοιχα. Τονίζουν τη σημασία των χειροκίνητα ταξινομημένων tweets επισημαίνοντας την προσθήκη τους ως κύριο λόγο βελτίωσης των αποτελεσμάτων.

Οι **Mohammand et al. (2013)** [24] ασχολούνται με την ανάλυση συναισθήματος σε επίπεδο μηνύματος και οντότητας σχεδιάζοντας μία εκδοχή ταξινομητή Support Vector Machines (SVM) για κάθε επίπεδο ανάλυσης. Εξετάζουν το πρόβλημα των τριών κλάσεων και αξιολογούν το μοντέλο τους με δεδομένα από το διαγωνισμό SemEval2013 [27], καταλαμβάνοντας την $1^{η}$ θέση και στα δύο υποπροβλήματα υποκειμενικότητας και πολικότητας. Το σύνολο εκπαίδευση περιέχει 3855 θετικά, 1624 αρνητικά και 4889 ουδέτερα χειροκίνητα ταξινομημένα tweets. Κατά την προεπεξεργασία των δεδομένων, αντικαθιστούν τα ονόματα χρηστών και τους υπερσυνδέσμους με κατάλληλες λέξεις-κλειδιά και κατακερματίζουν το κείμενο στους επιμέρους όρους, τους οποίους στη συνέχεια αναγνωρίζουν γραμματικά (POS tagging). Κάθε tweet αναπαρίσταται με ένα σύνολο χαρακτηριστικών : λέξεις και χαρακτήρες ν-γραμμάτων (3, 4, 5), πλήθος κεφαλαίων, POS tags, hashtags, emoticons, αρνήσεων και σημείων στίξης καθώς και λεξικογραφικών ιδιοτήτων που προσδιορίζονται με τη βοήθεια λεξικών. Παρατηρούν ότι ένας ταξινομητής Support Vector Machines (SVM) με όλα τα παραπάνω χαρακτηριστικά εμφανίζει πολύ καλύτερη συμπεριφορά σε σχέση με έναν απλό SVM ταξινομητή που χρησιμοποιεί μόνο μονογράμματα : F-score 69.02% έναντι 39.61% για το πρόβλημα της υποκειμενικότητας και 88.93% έναντι 80.28% στο πρόβλημα της πολικότητας. Συμπεραίνουν πως τα συναισθηματικά χαρακτηριστικά που προκύπτουν μέσω των λεξικών σε συνδυασμό με τα ν-γράμματα συνεισφέρουν το περισσότερο κέρδος στην αύξηση της ακρίβειας.

Οι **Günther & Furrer (2013)** [15] χρησιμοποιούν επίσης το SemEval2013 dataset [27] αλλά μελετούν την ανάλυση συναισθήματος μόνο σε επίπεδο μηνύματος-πρότασης. Κατά την προεπεξεργασία των δεδομένων, κανονικοποιούν τα κείμενα αντικαθιστώντας κεφαλαία με πεζά γράμματα, αφαιρώντας ψηφία και επαναλαμβανόμενους χαρακτήρες που προσδίδουν έμφαση. Εκτός από την ύπαρξη ή απουσία κανονικοποιημένων και συνώνυμων λέξεων, στο διάνυσμα των χαρακτηριστικών περιλαμβάνονται η χρήση άρνησης, η πρότερη πολικότητα κάθε όρου μέσω του SentiWordNet καθώς και η ύπαρξη/απουσία του εκάστοτε όρου σε clusters με λέξεις από το Twitter. Παρατηρούν ότι η κατασκευή ενός γραμμικού μοντέλου με συνάρτηση εκπαίδευσης Στοχαστική Κλίση Καθόδου (Stochastic Gradient Descent) υπερτερεί έναντι άλλων μεθόδων καταλαμβάνοντας τη $2^{η}$ θέση στο διαγωνισμό με $F_1 = 0.65$ και τονίζουν ότι η επιλογή ενός αλγορίθμου εκπαίδευσης είναι πιο σημαντική από την επιλογή των χαρακτηριστικών αυξανόμενου του πλήθους των προτύπων εκπαίδευσης.

Οι **Aston et al. (2014)** [4] μελετούν την ανάλυση συναισθήματος σε ροή δεδομένων από το Twitter - όπου συνήθεις τεχνικής μάθησης δέσμης (batch learning) είναι αναποτελεσματικές. Εξετάζουν εναλλακτικούς αλγορίθμους μάθησης με περιορισμούς ως προς το χρόνο επεξεργασίας και χωρητικότητας διατηρώντας υψηλά ποσοστά ακρίβειας. Ασχολούνται με τα προβλήματα υποκειμενικότητας και πολικότητας ξεχωριστά,

κατασκευάζοντας δύο εκδοχές από το σύνολο Sanders Corpus[6]. Αναπαριστούν το κείμενο μέσω ν-γραμμάτων αλλά παράλληλα επιτρέπουν την ύπαρξη ν-γραμμάτων διαφόρων μεγεθών στο σύνολο αναπαράστασης - το οποίο αποκαλούν 1-ν γράμματα. Ο αριθμός των πιθανών ν-γραμμάτων αυξάνεται εκθετικά με την αύξηση του μεγέθους ν οπότε ο υπολογισμός όλων των πιθανών χαρακτηριστικών είναι πρακτικά ανέφικτος σε περιορισμένο χρόνο. Επιλέγουν, λοιπόν, τα N πρώτα χαρακτηριστικά ν-γραμμάτων όπως προκύπτουν μέσω 6 διαφορετικών αλγορίθμων αξιολόγησης της περιεχόμενης πληροφορίας (Chi-squared, Filtered Feature, Gain Ratio, Info Gain, OneR και Relief). Έπειτα, εξετάζουν 3 εκδοχές του ταξινομητή Perceptron (Simple, Best Learning Rate, Voted) καθώς και συνδυασμούς τους. Παρατηρούν ότι οι εκδοχές Best Learning Rate και Voted εμφανίζουν σταθερά και όμοια αποτελέσματα με την πρώτη, ωστόσο, να απαιτεί πολύ μεγαλύτερο χρόνο εκπαίδευσης. Συνδυάζοντας τις δύο αυτές τεχνικές επιτυγχάνουν καλύτερη ακρίβεια και στα δύο προβλήματα με F-score 85% και 78% στην ανίχνευση υποκειμενικότητας και πολικότητας αντίστοιχα. Συμπεραίνουν πως δε συντελούν με τον ίδιο βαθμό όλα τα χαρακτηριστικά στην κατηγοριοποίηση καθώς εξαιρώντας κάποια από αυτά, μειώνεται ο χρόνος εκτέλεσης χωρίς όμως να επηρεάζεται αρνητικά η επίδοση του ταξινομητή.

2.6.2 Εργασίες με χρήση Μη-Επιβλεπόμενης Μηχανικής Μάθησης

Οι **O'Connor et al. (2010)** [29] εξετάζουν τη σύνδεση των δημοσκοπήσεων με την ανάλυση συναισθήματος σε tweets που αναφέρονται στον Πρόεδρο των ΗΠΑ Barrack Obama. Συλλέγουν μέσω του TwitterAPI 1 δις tweets αναρτήθηκαν στο διάστημα 2008 - 2009 χωρίς να ελέγχουν τα δημογραφικά χαρακτηριστικά των δημιουργών και τη γλώσσα γραφής. Κατηγοριοποιούν κάθε tweet μετρώντας αν περιέχει περισσότερες θετικές ή αρνητικές λέξεις, αναζητώντας την πολικότητα του κάθε όρου στο λεξικό συναισθημάτων MPQA [54]. Παρατηρούν πως αν και πρόκειται για μία απλή μέθοδο ανίχνευσης συναισθήματος, επιτυγχάνει να συλλέξει το συνολικό συναίσθημα της κοινής γνώμης και εμφανίζει υψηλή συσχέτιση με τα αποτελέσματα των δημοσκοπήσεων σε βαθμό μέχρι και 80%. Συμπεραίνουν πως οι απαιτητικές και χρονοβόρες τεχνικές εξόρυξης της κοινής γνώμης μέσω δημοσκοπήσεων μπορούν να ενισχυθούν και να συμπληρωθούν από την ανάλυση συναισθήματος στον τεράστιο όγκο εύκολα συλλέξιμων δεδομένων από τα κοινωνικά δίκτυα.

Οι **Gayo-Avello et al. (2011)** [11] ασχολούνται επίσης με την εξόρυξη κοινής γνώμης σε θέματα πολιτικής από tweets. Εξετάζουν την ικανότητα πρόβλεψης του τελικού αποτελέσματος εγκλογικών αναμετρήσεων εφαρμόζοντας δημοφιλείς τεχνικές σε tweets από τις εκλογές της Βουλής των Αντιπροσώπων των ΗΠΑ το 2010. Στηρίζονται στη μέθοδο των O'Connor et al.[29], χρησιμοποιούν το λεξικό MPQA [54] και εισάγουν κάποιες τροποποιήσεις έτσι ώστε το συνολικό μοντέλο να προσαρμόζεται στη φύση και τα ιδιαίτερα χαρακτηριστικά του κάθε εκλογικού συστήματος. Λαμβάνουν υπόψη tweets τα οποία περιέχουν ονόματα υποψηφίων από αντίπαλες παρατάξεις και δεν επιτρέπουν ένα tweet να έχει ταυτόχρονα δύο αντίθετες πολικότητες. Σε αντίθεση με άλλες μελέτες, δεν παρατηρούν άμεση συσχέτιση με τα αποτελέσματα δημοσκοπήσεων, εμφανίζοντας μέσο όρο σφάλματος 7.6% εκτός του αποδεκτού ορίου 2-3%. Καταλήγουν στο συμπέρασμα ότι η ακρίβεια των λεξικών συναισθήματος όταν εφαρμόζονται σε πολιτικές συζητήσεις

είναι αρκετά χαμηλή και ως προσέγγιση ανεπαρκής και δηλώνουν ότι απαιτούνται πιο εξελιγμένες τεχνικές για να συλλάβουμε τη δυναμική του πολιτικού λόγου στα κοινωνικά δίκτυα. Θεωρούν πως η αποτυχία της ανάλυσης συναισθήματος μέσω λεξικών είναι ως ένα βαθμό αναμενόμενη καθώς οι ακριβείς δημογραφικές πληροφορίες των χρηστών που συζητούν για τις εκλογές είναι ελάχιστες, η φύση και ποιότητα των online πολιτικών συζητήσεων αδιευκρίνιστη όπως επίσης και ο τρόπος με τον οποίο ομάδες με διαφορετικές ιδεολογίες συμμετέχουν και ασκούν επιρροή μέσω των κοινωνικών δικτύων.

Οι **Thelwall et al. (2010)** [49] προτείνουν ένα νέο αλγόριθμο βασισμένο σε λεξικό συναισθήματος τον οποίο αποκαλούν SentiStrength χρησιμοποιώντας επίσης μη λεξικογραφικές γλωσσολογικές πληροφορίες και κανόνες. Εκτός από την πολικότητα κάθε κειμένου (θετικό/αρνητικό) υπολογίζουν και την αντίστοιχη ισχύ του συναισθήματος με εύρος τιμών 1 έως 5. Αρχικά, χρησιμοποιούν ένα σύνολο από 2600 σχόλια από το MySpace και κατασκευάζουν μία λίστα με 298 θετικούς και 465 αρνητικούς όρους ταξινομημένους ως προς την πολικότητά τους μαζί με την αντίστοιχη ισχύ τους. Έπειτα, επεκτείνουν το μοντέλο με λίστες από emoticons, όρους άρνησης, λέξεις που αυξάνουν ή μειώνουν τη ισχύ του συναισθήματος των συμφραζόμενων όρων (booster words). Παράλληλα, στο στάδιο της προεπεξεργασίας, διορθώνονται απλά ορθογραφικά λάθη και αντιμετωπίζονται φαινόμενα εμφατικής επιμήκυνσης (επαναλαμβανόμενα γράμματα, φθόγγοι και σημεία στίξης). Συγκρίνοντας το μοντέλο σε σχέση με διάφορους ταξινομητές επιβλεπόμενης μηχανικής μάθησης παρατηρούν ότι συμπεριφέρεται καλύτερα στην αναγνώριση των αρνητικών αλλά όχι των θετικών σχολίων. Βελτιωμένη έκδοση του αλγορίθμου παρουσιάζεται στο [48] όπου οι συγγραφείς αυξάνουν τους όρους από 693 σε 2310, εισάγουν μία λίστα με ιδιώματα καθώς και την έννοια της ενίσχυσης της πολικότητας λόγω εμφατικής επιμήκυνσης. Συγκρίνουν πάλι το μοντέλο με διαφορετικούς αλγορίθμους μάθησης σε διαφορετικά σύνολα δεδομένων - και από το Twitter- και παρατηρούν πως σε γενικές γραμμές εμφανίζει ικανοποιητικά ποσοστά ακρίβειας και μόνο ο ταξινομήτης Linear Regression υπερτερεί συστηματικά. Καταλήγουν ότι η ανάλυση συναισθήματος που βασίζεται σε λεξικά συναισθήματος και κανόνες έχει σταθερή συμπεριφορά και είναι ανεξάρτητη του πεδίου εφαρμογής.

Οι **Zhang et al. (2011)** [56] προτείνουν ένα μοντέλο βασισμένο σε κανόνες για την ανάλυση συναισθήματος σε επίπεδο οντότητας σε δεδομένα που συλλέγουν από το Twitter. Προεπεξεργάζονται το σύνολο δεδομένων : διαγράφουν διπλότυπα, αφαιρούν ονόματα χρηστών και υπερσυνδέσμους, αντικαθιστούν συντομογραφίες με την κανονική τους μορφή και αναγνωρίζουν γραμματικά τους επιμέρους όρους των μηνυμάτων (POS tagging). Έπειτα, υπολογίζουν τη συναισθηματική τιμή κάθε όρου με βάση την ομοιότητά του με λέξεις από το λεξικό συναισθημάτων και επιλύουν τις απλές αναφορές αντιστοιχίζοντας αντωνυμίες με την πιο κοντινή οντότητα του κειμένου. Εφαρμόζοντας το σύνολο κανόνων ο αλγόριθμος διαχωρίζει τις προτάσεις σε δηλωτικές, προστακτικές και ερωτηματικές ενώ παράλληλα μπορεί να αναγνωρίσει συγκρίσεις, αρνήσεις και αντιθετικές περιόδους. Τονίζουν ότι αυτή η μέθοδος εμφανίζει αρκετά καλή ακρίβεια (precision) αλλά χαμηλή ανάκληση (recall). Εκπαιδεύουν, λοιπόν, ένα δυαδικό ταξινομητή Support Vector Machines (SVM) με πρότυπα που προκύπτουν από την παραπάνω μη-επιβλεπόμενη διαδικασία ο οποίος ταξινομεί τα tweets στις τελικές τους κατηγορίες. Παρατηρούν πως η προσθήκη του ταξινομητή βελτιώνει δραματικά την ανάκληση και το F-score και παράλληλα ξεπερνά αρκετούς από τους state-of-the-art τεχνικές-σημεία αναφοράς.

Οι **Kumar & Sebastian (2012)** [21] ασχολούνται με την εξαγωγή γνώμης από tweets

προτείνοντας μία μη επιβλεπόμενη υβριδική μέθοδο που συνδυάζει μεγάλα σώματα κειμένου (corpus) και λεξικά για να προσδιορίσει τον σημασιολογικό προσανατολισμό των όρων του κειμένου. Προτείνουν ένα τρόπο υπολογισμού της τιμής του συναισθήματος ο οποίος εκτός από τις πολικότητες του λεξικού λαμβάνει υπόψη και το πλήθος των emoticons, επαναλαμβανόμενων γραμμάτων, θαυμαστικών και κεφαλαίων, χαρακτηριστικά που χρησιμοποιούνται συνήθως από τις επιβλεπόμενες τεχνικές.

Οι **Hu et al. (2013)** [17] μελετούν τη μη-επιβλεπόμενη ανάλυση συναισθήματος στα κοινωνικά δίκτυα με τη βοήθεια "σημάτων συναισθήματος" , δηλαδή οποιαδήποτε πληροφορία η οποία μπορεί να συσχετισθεί με συναισθηματική πολικότητα. Εξετάζουν την επίδραση των emoticons και των κοινών τους εμφανίσεων στην κατηγοριοποίηση των δεδομένων από τα σύνολα Stanford Twitter Sentiment Corpus [14] και το Obama McCain Debate Corpus [45]. Στο στάδιο της προεπεξεργασίας αναπαριστούν τα κείμενα μέσω μονογραμμάτων και επιλέγουν την εμφάνιση των όρων (term presence) ως χαρακτηριστικό το οποίο σε συνδυασμό με το λεξικό MPQA [54] χρησιμοποιούνται για να υπολογίσουν τους δείκτες ένδειξης (indication) και συσχέτισης (correlation) συναισθήματος των tweets. Παρατηρούν ότι το μοντέλο εμφανίζει καλύτερη συμπεριφορά σε σχέση με άλλες μη-επιβλεπόμενες τεχνικές με βέλτιστη επίδοση 74.2% και 70.97% αντίστοιχα στα 2 σύνολα δεδομένων. Συμπεραίνουν πως η χρήση σημάτων συναισθήματος βελτιώνει την ακρίβεια με το δείκτη ένδειξης να έχει τη μεγαλύτερη συνεισφορά.

Οι **Ortega et al. (2013)** [30] προτείνουν ένα μη-επιβλεπόμενο σύστημα ανάλυσης συναισθήματος για το πρόβλημα της γενικής κατηγοριοποίησης των tweets. Αρχικά, προεπεξεργάζονται τα μηνύματα : κατακερματίζουν τις προτάσεις σε όρους, αφαιρούν retweets, ονόματα χρηστών, υπερσυνδέσμους, τον χαρακτήρα "#" από τα hashtags. Έπειτα, αντικαθιστούν τα emoticons με λέξεις συναισθήματος από ένα χειροκίνητα κατασκευασμένο λεξικό emoticons μέσω της Wikipedia και τις συντομογραφίες με την πλήρη μορφή τους και, τέλος, αφαιρούν συνήθεις λέξεις, λημματοποιούν και αναγνωρίζουν γραμματικά (POS tagging) τους όρους του κειμένου. Στη συνέχεια, υπολογίζουν τη συναισθηματική πολικότητα κάθε λέξης λαμβάνοντας υπόψη και το ευρύτερο εννοιολογικό πλαίσιο στο οποίο εμφανίζεται, αποσαφηνίζοντας την έννοιά της μέσω του WordNet και του SentiWordNet. Η τελική κατηγοριοποίηση του κάθε tweet πραγματοποιείται μέσω ενός μοντέλου κανόνων. Εξετάζουν την ακρίβεια του συστήματος σε δεδομένα από το διαγωνισμό SemEval2013 [27] όπου καταλαμβάνουν τη $25^{η}$ θέση με επίδοση F_1=51.17%. Τονίζουν πως τα αποτελέσματα είναι αρκετά ικανοποιητικά δεδομένης της δυσκολίας του προβλήματος και του γεγονότος ότι δε χρησιμοποιούν κανένα σύνολο εκπαίδευσης, είναι μία καθαρά μη επιβλεπόμενη προσέγγιση.

Οι **Saif et al. (2014)** [40] παρουσιάζουν την μη-επιβλεπόμενη μέθοδο SentiCircle η οποία βασίζεται σε λεξικό συναισθήματος. Θεωρούν πως το συναίσθημα ενός όρου δεν είναι στατικό αλλά εξαρτάται από το εννοιολογικό πλαίσιο στο οποίο ανήκει καθώς και από τα συμφραζόμενα. Προτείνουν, λοιπόν, το δείκτη TDOC (Term Degree of Correlation) για να υπολογίσουν τη σχέση ανάμεσα σε μία λέξη w και τους συμφραζόμενους όρους c_iμε την ίδια σημασιολογική χροιά. Έπειτα, αναπαριστούν τη λέξη w και τους όρους c_i σε πολικό σύστημα συντεταγμένων με κέντρο τη λέξη w , ακτίνα τον δείκτη TDOC και γωνία την πρότερη πολικότητα κάθε όρου όπως προσδιορίζεται από λεξικό συναισθήματος. Αξιοποιώντας τις τριγωνομετρικές ιδιότητες της αναπαράστασης υπολογίζουν τον συναισθηματικό προσανατολισμό και την ισχύ της λέξης w. Εξετάζουν 3 διαφορετικά σύνολα δεδομένων: Stanford Twitter Sentiment Corpus [14], Obama McCain Debate Corpus και Health Care Reform και παρατηρούν ότι η μέθο-

δος SentiCircle συναγωνίζεται την state-of-the-art μέθοδο SentiStrength έχοντας μέση ακρίβεια 72.39% έναντι 71.7%.

3

Ανάλυση Συναισθήματος με Γράφους ν-γραμμάτων

Όπως έγινε εμφανές από την ανασκόπηση της σχετικής βιβλιογραφίας, οι περισσότερες μέθοδοι που εφαρμόζονται στην Ανάλυση Συναισθήματος ανιχνεύουν εκφραστικά μοτίβα σε συγκεκριμένες γλώσσες με αποτέλεσμα να μην είναι αποτελεσματικές σε πολυγλωσσικά σύνολα δεδομένων. Για το λόγο αυτό, στην παρούσα εργασία επιλέγουμε να χρησιμοποιήσουμε την προσέγγιση των Aisopos et al. [3] για την Ανάλυση Συναισθήματος σε δεδομένα από το Twitter με τη βοήθεια γράφων ν-γραμμάτων. Στις υπόλοιπες Ενότητες αναλύεται η εν λόγω μέθοδος και γίνεται μία σύντομη παρουσίαση των αλγορίθμων Μηχανικής Μάθησης που θα χρησιμοποιηθούν.

3.1 Αναπαράσταση Δεδομένων

3.1.1 Βασικές Έννοιες & Ορισμοί

Στην περιοχή της Επεξεργασίας Φυσικής Γλώσσας, η χρήση ν-γραμμάτων είναι ευρύτατα διαδεδομένη με εφαρμογές στη διόρθωση ορθογραφικών λαθών, στο φιλτράρισμα ανεπιθύμητης αλληλογραφίας (spam) και την ανίχνευση λογοκλοπής. Ένα ν-γράμμα είναι ένα - διατεταγμένο συνήθως - σύνολο λέξεων ή χαρακτήρων που αποτελούνται από n στοιχεία (λέξεις ή χαρακτήρες). Στην Ανάλυση Συναισθήματος τα ν-γράμματα χρησιμοποιούνται συνήθως ως γλωσσικό μοντέλο για την εξαγωγή των τιμών του διανύσματος χαρακτηριστικών.

Παράδειγμα 3.1.1. Παραδείγματα ν-γραμμάτων από την πρόταση : *test texts*
Μονογράμματα Λέξεων : *test, texts*
Διγράμματα Χαρακτήρων : *te, es, st, t_ ,_ t, te, ex, xt, ts*
3-γράμματα Χαρακτήρων : *tes, est, st_ ,t _ t, _ te, tex, ext, xts*

Πιο αυστηρά, το ν-γράμμα χαρακτήρων ενός κειμένου ορίζεται ως :

Ορισμός 3.1.1. Εάν $n > 0, n \in \mathbb{Z}$, και c_i είναι ο i-ιοστός χαρακτήρας της ακολουθίας χαρακτήρων μήκους l $T^l = (c_1, c_2, \ldots, c_l)$ τότε ένα ν-γράμμα χαρακτήρων

$S^n = (S_1, S_2, \ldots, S_l)$ είναι μία υπακολουθία μήκους l του $T^l \iff \exists i \in [i, l-n+1] : \forall j \in [1, n] : S_j = C_{i+j-1}$. Θα γράφουμε ένα ν-γράμμα που επεκτείνεται από τον χαρακτήρα c_i έως και τον c_k , $k > i$ ως $S_{i,k}$ ενώ τα ν-γράμματα μήκους n θα απεικονίζονται ως S^n.

Ένα ν-γράμμα $S_{i,i+n-1}$ είναι ουσιαστικά μία υποσυμβολοσειρά μήκους n η οποία επεκτείνεται από το i-ιοστό μέχρι τον $(i-n+1)$-στό χαρακτήρα του αρχικού κειμένου. Το μήκος n ενός ν-γράμματος ονομάζεται και *βαθμός* του ν-γράμματος.

Ένας γράφος ν-γραμμάτων είναι ένας γράφος $G = \{V^G, E^G, L, W\}$ όπου V^G είναι το σύνολο των κορυφών, E^G είναι το σύνολο των ακμών, L είναι μία συνάρτηση που αποδίδει μία τιμή - επισημείωση σε κάθε κορυφή και σε κάθε ακμή και W είναι μία συνάρτηση που αποδίδει τιμή - βάρος σε κάθε ακμή. Ο γράφος έχει ν-γράμματα ως κορυφές $v^G \in V^G$ και οι ακμές $e^G \in E^G$ που συνδέουν τα ν-γράμματα υποδηλώνουν την εγγύτητα των αντίστοιχων κορυφών των ν-γραμμάτων. Τα βάρη των ακμών προκύπτουν είτε από την απόσταση μεταξύ δύο γειτονικών ν-γραμμάτων ή από το ποσοστό συνύπαρξης τους στο αρχικό κείμενο μέσα σε ένα συγκεκριμένο εύρος - παράθυρο. Η έννοια της απόστασης και του μήκους του παραθύρου αλλάζει ανάλογα αν χρησιμοποιούμε ν-γράμματα χαρακτήρων ή λέξεων. Η συνάρτηση επισημείωσης L για τις ακμές εκχωρεί σε κάθε ακμή τη συνένωση των επισημειώσεων των αντίστοιχων κορυφών.

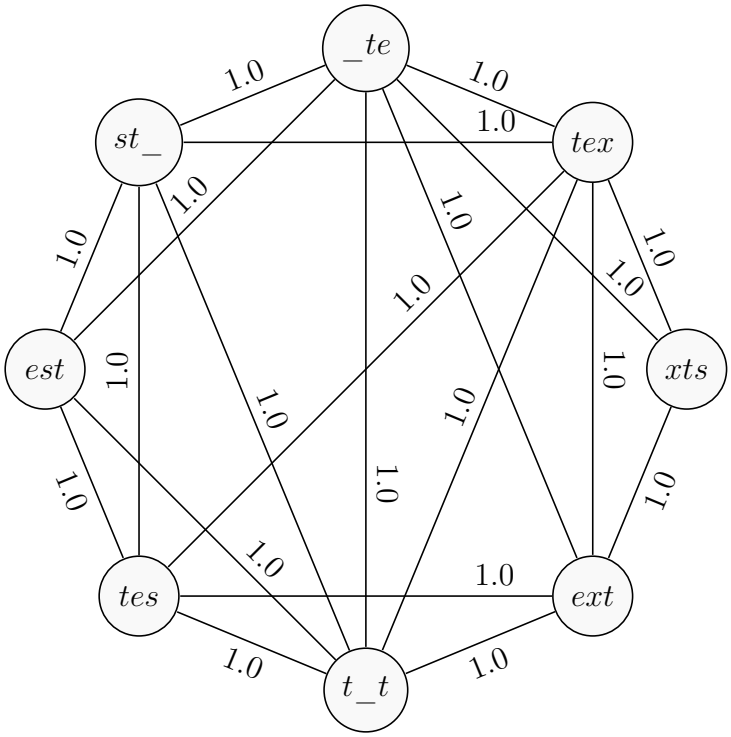
Πιο αυστηρά, ένας γράφος ν-γραμμάτων ενός κειμένου ορίζεται ως :

Ορισμός 3.1.2. Εαν $S = S_1, S_2, \ldots, S_k \neq S_l$, για $k \neq l, k, n \in \mathbb{N}$ είναι το σύνολο των διακριτών ν-γραμμάτων ενός κειμένου T^l και S_i είναι το i-ιστό ν-γράμμα τότε $G = \{V^G, E^G, L, W\}$ είναι ο γράφος όπου $V^G = S$ είναι το σύνολο των κορυφών v, E^G είναι το σύνολο των ακμών e της μορφής $e = \{v_1, v_2\}$, $L : V^G \to \mathbb{L}$ είναι μία συνάρτηση εκχώρησης επισημείωσης $l(v)$ από ένα σύνολο πιθανών επισημειώσεων $\mathbb{L}$ σε κάθε κορυφή v και $W : E^G \to \mathbb{R}$ είναι μία συνάρτηση ανάθεσης βάρους $w(e)$ σε κάθε ακμή.

Στο Σχήμα 3.1 απεικονίζεται ο γράφος τριγραμμάτων χαρακτήρων για τη φράση "test texts". Παρατηρούμε πως αποτυπώνεται περισσότερη πληροφορία σε σχέση με την απεικόνιση τριγραμμάτων χαρακτήρων της ίδιας φράσης στο Παράδειγμα 3.1.1.

Στη συγκεκριμένη προσέγγιση, τα βάρη των ακμών υπολογίζονται σύμφωνα με το πλήθος των φορών όπου ένα δοθέν ζεύγος ν-γραμμάτων S_i, S_j τυχαίνει να γειτνιάζει μέσα σε μία συμβολοακολουθία εντός της μεταξύ τους απόστασης D_{win}, υποδηλώνοντας την εγγύτητα των δύο ν-γραμμάτων.

Γενικά, χρησιμοποιείται ένα σταθερού μήκους παράθυρο χαρακτήρων ή λέξεων D_{win} γύρω από ένα συγκεκριμένο ν-γράμμα $S_0 \equiv S^r, r \in \mathbb{N}^*$ με όλους τους χαρακτήρες ή λέξεις να θεωρούνται γείτονες του S_0. Επομένως, έχει ιδιαίτερη σημασία να επιλεγεί κατάλληλο μήκος παραθύρου καθώς "δεν είναι όλες οι αποστάσεις το ίδιο σημαντικές και επομένως δύο ν-γράμματα σε απόσταση 150 χαρακτήρων δεν έχουν μάλλον ουσιαστική σύνδεση και εξάρτηση."[12]. Παράλληλα, καθοριστικό ρόλο έχει και ο τρόπος με τον οποίο υπολογίζεται η γειτνίαση δύο ν-γραμμάτων λαμβάνοντας υπόψη : (i) μόνο τους προηγούμενους χαρακτήρες του S_0 κατά την κύλιση του παραθύρου D_{win} στο κείμενο (ασύμμετρη προσέγγιση), (ii) και τους επόμενους χαρακτήρες (συμμετρική προσέγγιση) και (iii) και τους επόμενους χαρακτήρες αλλά και την πραγματική απόσταση μεταξύ του S_0 και του εκάστοτε ν-γράμματος S_i (κανονικοποιημένη κατά Gauss συμμετρική προσέγγιση).



Σχήμα 3.1: Πάραδειγμα Γράφου 3-γραμμάτων

3.1.2 Αναπαράσταση Δεδομένων με Γράφους ν-γραμμάτων

Σε αυτή την Ενότητα παρουσιάζεται η εφαρμογή της μεθόδου αναπαράστασης μέσω γράφων ν-γραμμάτων για την ανάλυση συναισθήματος σε δεδομένα από το κοινωνικό δίκτυο Twitter. Η περιγραφή και τα συμπεράσματα προέρχονται από τους Aisopos et al. (2011) στο [3].

Σύμφωνα με το μοντέλο γράφων ν-γραμμάτων, κάθε tweet t_i αναπαρίσταται με ένα γράφο ν-γραμμάτων - ο οποίος ονομάζεται *γράφος μηνύματος* (*tweet graph*). Για την κατασκευή του γράφου χρησιμοποιείται ένα κυλιόμενο παράθυρο D_{win} μήκους όπου το αρχικό κείμενο του tweet αναλύεται σε επικαλυπτόμενες συμβολοακολουθίες μήκους (ν-γράμματα χαρακτήρων). Μία ακμή $e^{G_{t_i}} \in E^{Gt_i}$ που συνδέει ένα ζεύγος ν-γραμμάτων υποδηλώνει ότι αυτές οι συμβολοακολουθίες γειτνιάζουν στο αρχικό κείμενο σε απόσταση το πολύ ν χαρακτήρων.

Το μοντέλο γράφων ν-γραμμάτων μπορεί να χρησιμοποιηθεί και για να αναπαραστήσουμε ομοιόμορφα ένα σύνολο από tweets που εμφανίζουν το ίδιο συναίσθημα. Κάθε κλάση πολικότητας απεικονίζεται με ένα γράφο G_{T_P} ο οποίος αναπαριστά τα tweets του συνόλου εκπαίδευσης από τα οποία έχει δημιουργηθεί. Η κατασκευή του γράφου κλάσης βασίζεται στη *λειτουργικότητα ενημέρωσης (update functionality)* : Δοθέντος ενός συνόλου από tweets με την ίδια πολικότητα T_P, ο *γράφος κλάσης* δημιουργείται από έναν αρχικά κενό γράφο G_{T_P}. Το i-στο tweet $t_i \in T^P$ μετασχηματίζεται σε έναν γράφο μηνύματος G_{t_i} ο οποίος στη συνέχεια συγχωνεύεται με τον G_{T_P} δημιουργώντας έναν νέο γράφο G_u ως αποτέλεσμα της ένωσης των κορυφών και ακμών των δύο αρχικών γράφων. Τα βάρη των ακμών ισούνται με το μέσο όρο των βαρών των G_{t_i} και G_{T_P}. Πιο

αυστηρά, ο νέος γράφος G_u έχει τις εξής ιδιότητες :

$$G_u = \{V^u, E^u, W_i^u\} \text{ όπου } V^u = V^{G_{T_P}} \cup V^{G_{t_i}} \text{ , } E^u = E^{G_{T_P}} \cup E^{G_{t_i}} \text{ και}$$

$$W^{u_i} = W^{G_{T_P(e)}} + \frac{W^{G_{t_i(e)}} - W^{G_{T_P(e)}}}{i} \tag{3.1}$$

Όπως εξηγείται στο [13], η διαίρεση με i εξασφαλίζει ότι το αθροιζόμενο βάρος συγκλίνει στη μέση τιμή των αντίστοιχων βαρών των ακμών ανάμεσα σε όλους τους γράφους μηνύματος G_{t_i} έτσι ώστε η ενημέρωση να είναι ανεξάρτητη της σειράς με την οποία συγχωνεύονται τα tweets. Μετά τη συγχώνευση όλων των tweets του T_P στο γράφο G_{T_P}, οι ακμές $E^{G_{T_P}}$ αποτυπώνουν τα πιο χαρακτηριστικά εκφραστικά μοτίβα που εμφανίζουν τα μηνύματα της εκάστοτε κλάσης όπως επαναλαμβανόμενες και γειτονικές συμβολοακολουθίες,ειδικοί χαρακτήρες και ψηφία.

Για να εκτιμήσουμε την ομοιότητα μεταξύ ενός γράφου μηνύματος G_{t_i} και ενός γράφου κλάσης $E^{G_{T_P}}$ χρησιμοποιούνται τρεις διαφορετικοί δείκτες - μετρικές ομοιότητας [12] :

(i) Ομοιότητα Συνοχής (Containment Similarity - CS) : εκφράζει το ποσοστό των ακμών του γράφου G_{t_i} οι οποίες περιέχονται και στο γράφο G_{T_P}. Αν G είναι ένας γράφος ν-γραμμάτων και e μία ακμή ενός γράφου ν-γραμμάτων τότε ορίζουμε τη συνάρτηση μ όπου $\mu(e, G) = 1$ αν και μόνο αν $e \in G$ και 0 αλλιώς. Συνεπώς,

$$CS(G_{t_i}, G_{T_P}) = \sum_{e \in G_{t_i}} \frac{\mu(e, G_{T_P})}{min(|G_{t_i}|, |G_{T_P}|)}$$

όπου $|G|$ δηλώνει το μέγεθος του γράφου ν-γραμμάτων, δηλαδή το πλήθος των ακμών που περιέχει $|G| \equiv |E^G|$.

(ii) Ομοιότητα Τιμής (Value Similarity - VS) : εκφράζει το πλήθος των ακμών του γράφου G_{t_i} οι οποίες περιέχονται και στο γράφο G_{T_P} λαμβάνοντας υπόψη και τα βάρη τους. Κάθε κοινή ακμή e έχει βάρη $w^{t_i}(e)$ και $w^{T_P}(e)$ στους γράφους G_{t_i} και G_{T_P} συνεισφέροντας $\frac{VR(e)}{max(|G_{t_i}|, |G_{T_P}|)}$ στο δείκτη VS. Ο όρος VR (Value Ratio) είναι ένας συμμετρικός συντελεστής κλίμακας $VR : [0, 1] \to [0, 1]$ με τύπο :

$$VR(e) = \frac{min(w^{t_i}(e), w^{T_P}(e))}{max(w^{t_i}(e), w^{T_P}(e))}$$

οπότε η Ομοιότητα Τιμής υπολογίζεται από τη σχέση :

$$VS(G_{t_i}, G_{T_P}) = \frac{\sum_{e \in G_{t_i}} \frac{min(w^{t_i}(e), w^{T_P}(e))}{max(w^{t_i}(e), w^{T_P}(e))}}{max(|G_{t_i}|, |G_{T_P}|)}$$

Ο δείκτης VS συγκλίνει στο 1 για γράφους G_{t_i} και G_{T_P} που μοιράζονται ακμές και παρόμοια βάρη με τη τιμή VS = 1 να δηλώνει το τέλειο ταίριασμα μεταξύ των δύο συγκρινόμενων γράφων.

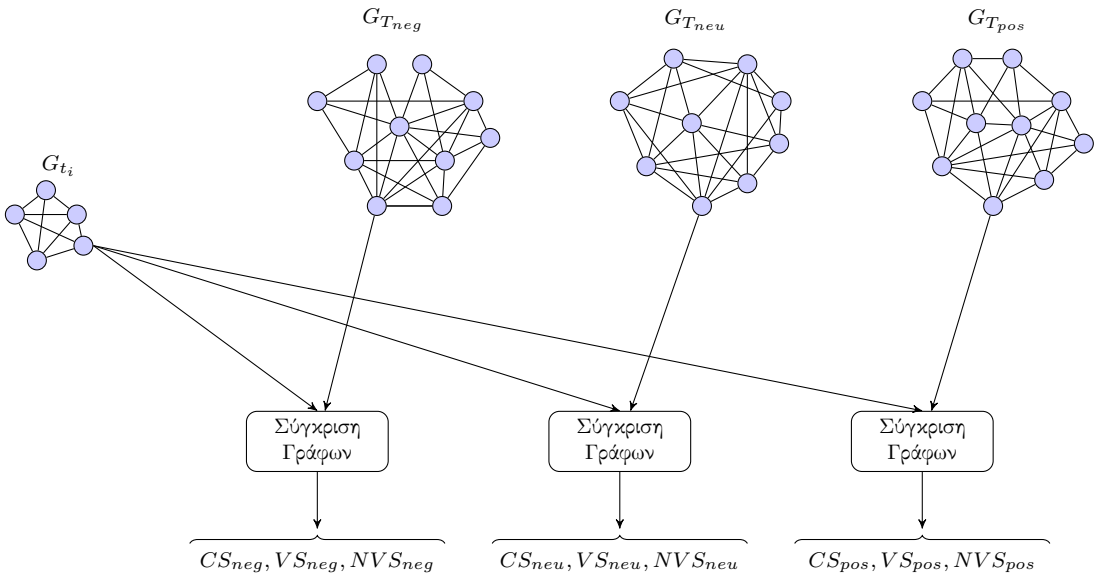
(iii) Κανονικοποιημένη Ομοιότητα Τιμής (Normalized Value Similarity - NVS) : αποσυνδέει το δείκτη Ομοιότητας Τιμής VS από την επίδραση του μεγέθους του μεγαλύτερου γράφου διαιρώντας με το δείκτη Ομοιότητας Μεγέθους (Size Similarity - SS) οπότε :

$$VS(G_{t_i}, G_{T_P}) = \frac{VS(G_{t_i}, G_{T_P})}{SS(G_{t_i}, G_{T_P})}$$

όπου

$$SS(G_{t_i}, G_{T_P}) = \frac{min(|G_{t_i}|, |G_{T_P}|)}{max(|G_{t_i}|, |G_{T_P}|)}$$

Η κατηγοριοποίηση ενός tweet t_i περιλαμβάνει την εξής διαδικασία (Σχήμα 3.2) : ο γράφος μηνύματος G_{t_i} συγκρίνεται με τους γράφους G_{T_neg},G_{T_pos} και G_{T_neut} προσδιορίζεται η εγγύτητά του με κάθε μία κλάση. Προκύπτουν, λοιπόν, 3 δείκτες ομοιότητας (CS, VS, NVS) ανά κλάση οι οποίοι αποτελούν το διάνυσμα χαρακτηριστικών που δοθεί στη συνέχεια ως είσοδος στον ταξινομητή. Εξετάζοντας τα 9 χαρακτηριστικά, ο ταξινομητής θα επιλέξει την πιο πιθανή κατηγορία που ανήκει το tweet.



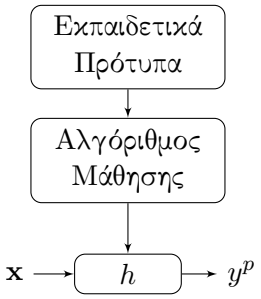
Σχήμα 3.2: Διαδικασία υπολογισμού διανύσματος χαρακτηριστικών με το μοντέλο γράφων ν-γραμμάτων

Οι Aisopos et al. [2] επιλέγουν να χρησιμοποιήσουν γράφους ν-γραμμάτων με χαρακτήρες και όχι λέξεις καθώς η εξαγωγή των λέξεων απαιτεί τη χρήση τεχνικών εξειδικευμένων για συγκεκριμένες γλώσσες με αποτέλεσμα να αίρεται η γλωσσική ανεξαρτησία της προσέγγισης. Παράλληλα, οι μέθοδοι που στηρίζονται στην εξαγωγή όρων και κατασκευή διανύσματος χαρακτηριστικών εμφανίζουν χαμηλή αποτελεσματικότητα σε δεδομένα από κοινωνικά δίκτυα λόγω της πολυγλωσσίας αλλά και χαμηλή αποδοτικότητα εξαιτίας της "κατάρας της διαστατικότητας" (curse of dimensionality) που πλήττει τα συγκεκριμένα μοντέλα αναπαράστασης. Πολλές φορές όροι με την ίδια σημασία αλλά διαφορετική συντακτική μορφή αναγνωρίζονται από τις μεθόδους εσφαλμένα ως ξεχωριστές έννοιες - φαινόμενο γνωστό με τον όρο συνωνυμία - με αποτέλεσμα το μέγεθος

του διανύσματος χαρακτηριστικών να αυξάνεται δραματικά. Επομένως, ο χώρος των χαρακτηριστικών εξαρτάται άμεσα από την ποικιλομορφία του λεξιλογίου των tweets επηρεάζοντας αρνητικά την χρονική και χωρική πολυπλοκότητα των αλγορίθμων και το απαιτούμενο υπολογιστικό κόστος. Αντίθετα, στο μοντέλο γράφων ν-γραμμάτων το πλήθος των χαρακτηριστικών εξαρτάται μόνο από τον αριθμό των κλάσεων κατηγοριοποίησης.

3.2 Αλγόριθμοι Κατηγοριοποίησης

Η κατηγοριοποίηση δεδομένων είναι μία διαδικασία δύο φάσεων με στόχο την πρόβλεψη της μεταβλητής κλάσης y με δυνατές τιμές $c_1, c_2, \cdots, c_k$ χρησιμοποιώντας ένα σύνολο χαρακτηριστικών $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$. Στηρίζεται στη διαδικασία της μάθησης κατά την οποία ένας ταξινομητής ανάλογα με τον αλγόριθμο μάθησης που εφαρμόζει, κατασκευάζει μία συνάρτηση κατηγοριοποίησης $h : \mathcal{X} \to \mathcal{Y}$ από m εκπαιδευτικά πρότυπα (σύνολο εκπαίδευσης) τα οποία βρίσκονται στη μορφή $(\mathbf{x}^{(i)}, y^{(i)}), \quad i = 1, 2, \cdots, m$ όπου y_i η κλάση που ανήκει το κάθε πρότυπα. Έπειτα, στη φάση της κατηγοριοποίησης, ο ταξινομητής αποφασίζει για την πιο πιθανή κλάση y^p των νέων, άγνωστων δεδομένων μέσω της συνάρτησης h. Παρατηρούμε, λοιπόν, πως η επίδοση ενός ταξινομητή είναι άμεσα συνδεδεμένη με τον αλγόριθμο μάθησης που χρησιμοποιεί κατά τη φάση της εκπαίδευσης.



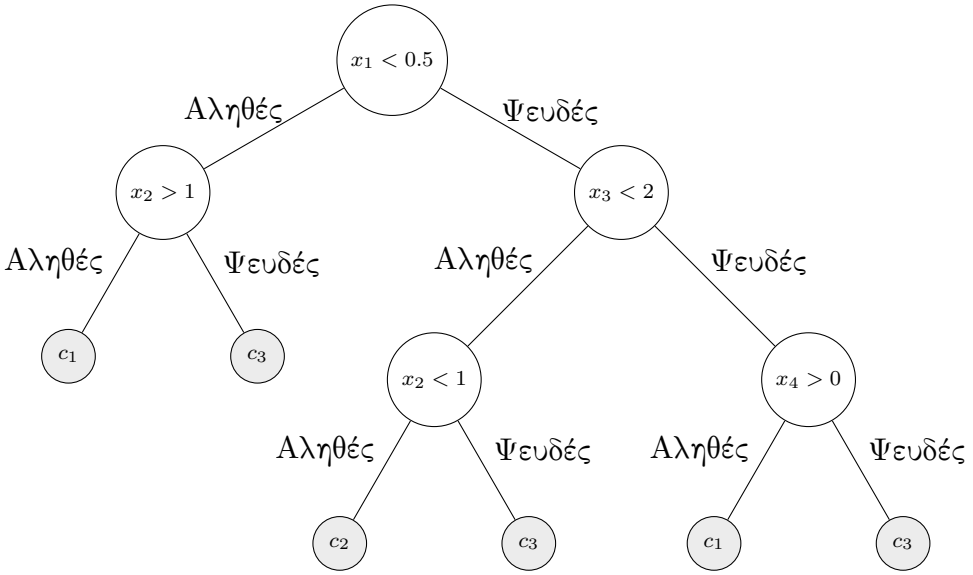
Σχήμα 3.3: Διαδικασία Κατηγοριοποίησης

Στις επόμενες υποενότητες παρουσιάζονται συνοπτικά τα βασικά στοιχεία και ιδιότητες των διαφορετικών αλγορίθμων μάθησης που θα χρησιμοποιηθούν στην εργασία.

3.2.1 Δένδρα Αποφάσεων

Ένα δένδρο απόφασης είναι ένας ταξινομητής που αναπαρίσταται μέσω δενδρικής μορφής. Κάθε εσωτερικός κόμβος του δένδρου αντιστοιχεί σε μία κατάσταση που διαχωρίζει τα δεδομένα σε διαφορετικές ομάδες ανάλογα με κάποιο συγκεκριμένο χαρακτηριστικό. Κάθε κλαδί του δένδρου αναπαριστά μία υποομάδα δεδομένων η οποία εξαρτάται από τον κόμβο - γονέα. Οι κόμβοι-φύλλα αντιστοιχούν σε μία συγκεκριμένη κλάση που προκύπτει από τις τιμές των χαρακτηριστικών ακολουθώντας το μονοπάτι από την κορυφή του δένδρου μέχρι το κόμβο-φύλλο.

Τα δένδρα κατασκευάζονται με αναδρομικές διασπάσεις των υποσυνόλων των δεδομένων ανάλογα με την επιλογή των χαρακτηριστικών και τις συνθήκες ελέγχου. Υπάρχουν



Σχήμα 3.4: Παράδειγμα Δένδρου Αποφάσεων.

αρκετοί αλγόριθμοι μηχανικής μάθησης που εξάγουν δένδρα αποφάσεων από δεδομένα π.χ. ID3, C4.5 και CART επιλέγοντας χαρακτηριστικά μέσω μίας συνάρτησης αξιολόγησης. Συνήθως χρησιμοποιείται το Κέρδος Πληροφορίας (Information Gain) το οποίο βασίζεται στην Εντροπία Πληροφορίας (Information Entropy).

Ορισμός 3.2.1. Έστωσαν δένδρο απόφασης $\mathcal{T}_{tree}$, c το πλήθος των κλάσεων της μεταβλητής y και S το σύνολο των δεδομένων εκπαίδευσης στον κόμβο διαχωρισμού. Η εντροπία του S υπολογίζεται από τη σχέση:

$$\mathcal{E}(S) = -\sum_{i=1}^{c} P(y = c_i|S) \cdot log_2 P(y = c_i|S)$$

όπου $P(y = c_i|S)$ είναι το ποσοστό των προτύπων του S που ανήκουν στην κατηγορία c_i.

Η εντροπία δείχνει την ομογενότητα της μεταβλητής κλάσης y στο χώρο S. Αν ο S αντιστοιχεί στη ρίζα του δένδρου τότε η εντροπία υπολογίζεται για όλο το σύνολο δεδομένων.

Ορισμός 3.2.2. Έστωσαν δένδρο απόφασης $\mathcal{T}_{tree}$, S το σύνολο των δεδομένων εκπαίδευσης στον κόμβο διαχωρισμού και μία ανεξάρτητη μεταβλητή με τιμές $Values(A)$ βάσει της οποίας επιχειρείται ο επόμενος διαχωρισμός. Το κέρδος πληροφορίας αναπαριστά τη μείωση της εντροπίας του συνόλου εκπαίδευσης S αν επιλεγεί ως παράμετρος διαχωρισμού η μεταβλητή και ορίζεται από τη σχέση:

$$\mathcal{G}(S, A) = \mathcal{E}(S) - \sum_{u} \frac{|S_u|}{|S|} \cdot \mathcal{E}(S_u)$$

όπου $\mathcal{E}(S)$ είναι η εντροπία πληροφορίας του υπό εξέταση κόμβου, u μία από τις δυνατές τιμές του A, S_u το πλήθος των δεδομένων με $A = u$ και $\mathcal{E}(S_u)$ είναι η εντροπία πληροφορίας του υπό εξέταση κόμβου ως προς την τιμή $A = u$.

Όταν μειώνεται η εντροπία πληροφορίας, αυξάνεται η πυκνότητα πληροφορίας και η περιγραφή γίνεται πιο συμπαγής.

Τα δένδρα αποφάσεων απαιτούν λίγη προεπεξεργασία δεδομένων, μπορούν να διαχειριστούν μεγάλα σύνολα δεδομένων σε σύντομο χρόνο και γενικά αποτελούν ένα γρήγορο ταξινομητή "ανοικτού τύπου" (white box model) καθώς είναι εμφανή τα χαρακτηριστικά στα οποία δίνεται η μεγαλύτερη βαρύτητα και το τελικό αποτέλεσμα είναι επαληθεύσιμο.

Ωστόσο, η κατασκευή του βέλτιστου δένδρου αποφάσεων θεωρείται NP-Complete πρόβλημα με αποτέλεσμα οι αλγόριθμοι μάθησης εφαρμόζουν ευριστικές μεθόδους π.χ. ο ID3 στηρίζεται στο κριτήριο της άπληστης επιλογής όπου υπολογίζονται τοπικά βέλτιστες αποφάσεις σε κάθε κόμβο χωρίς να εγγυώνται για την εξαγωγή του ολικά βέλτιστου δένδρου απόφασης. Επιπλέον, εμφανίζουν μέτρια συμπεριφορά ως προς την ευστάθεια καθώς αρκετές φορές μικρές διακυμάνσεις στα δεδομένα οδηγούν σε πολύ διαφορετικά δένδρα αποφάσεων.

3.2.2 Λογιστική Παλινδρόμηση

Η Λογιστική Παλινδρόμηση (Logistic Regression) ταξινομεί τα δεδομένα υπολογίζοντας για κάθε μία από τις δυνατές κλάσεις την εκ των υστέρων (posteriori) πιθανότητα το δεδομένο εισόδου να ανήκει στην εκάστοτε κλάση c_i δοθέντος του διανύσματος χαρακτηριστικών $\mathbf{x}$ και επιλέγει ως αναμενόμενη κλάση y^p εκείνη με τη μέγιστη πιθανότητα. Για τον υπολογισμό της πιθανότητας που αντιστοιχεί σε κάθε κλάση p_{c_i} εφαρμόζει στα δεδομένα εκπαίδευσης ένα γραμμικό μοντέλο $f(\mathbf{x}, \mathbf{w})$ όπου $\mathbf{w}$ το διάνυσμα συντελεστών ξεχωριστό για κάθε κλάση. Ωστόσο, η πιθανότητα p_{c_i} εξ' ορισμού έχει πεδίο τιμών το $[0, 1]$ ενώ οι γραμμικές συναρτήσεις είναι μη φραγμένες. Επομένως, για να αντιστοιχίζουμε την p_{c_i} σε μη φραγμένο πεδίο χρησιμοποιούμε τον λογιστικό μετασχηματισμό:

$$logit(p_{c_i}) = log\frac{p_{c_i}}{1 - p_{c_i}}$$

Στην περίπτωση των 2 κλάσεων, οι πιθανότητες p_{c_1} και p_{c_2} είναι συμπληρωματικές καθώς $p_{c_1} + p_{c_2} = 1$ οπότε προκύπτει :

$$logit(p_{c_i}(\mathbf{x})) = log\frac{p_{c_i}(x)}{1 - p_{c_i}(x)} = \beta_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n = \beta_0 + \mathbf{w} \cdot \mathbf{x}$$

όπου β_0 ένας βαθμωτός όρος.

Επιλύοντας ως προς $p_{c_i}(\mathbf{x})$ λαμβάνουμε :

$$p_{c_i}(\mathbf{x}) = \frac{1}{1 + e^{(1-(\beta_0+\mathbf{w}\cdot\mathbf{x}))}}$$

Το σύνορο απόφασης που διαχωρίζει τις δύο κλάσεις προκύπτει από τη λύση της $\beta_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n = \beta_0 + \mathbf{w} \cdot \mathbf{x} = 0$ και δεν απαιτείται ξεχωριστό διάνυσμα συντελεστών και βαθμωτός όρος για την δεύτερη κλάση. Ο βαθμωτός όρος β_0 και το διάνυσμα συντελεστών $\mathbf{w}$ προσδιορίζονται αναζητώντας τις τιμές εκείνες που μεγιστοποιούν την πιθανότητα στο σύνολο εκπαίδευσης.

Στην περίπτωση όπου οι κλάσεις είναι περισσότερες από δύο, έστω $k > 2$ τότε προκύπτει η Πολυωνυμική Λογιστική Παλινδρόμηση (Multinomial Logistic Regression ή Maximum Entropy). Για κάθε κλάση $c \in C^k$ απαιτείται ξεχωριστός βαθμωτός όρος $\beta_0^{(c)}$ και αντίστοιχο διάνυσμα συντελεστών $\mathbf{w}^{(c)}$ και οι υπό συνθήκη πιθανότητες υπολογίζονται από τη σχέση :

$$P(y = c|\bar{x} = \mathbf{x}) = \frac{e^{(\beta_0^{(c)} + \mathbf{w}^{(c)} \cdot \mathbf{x})}}{\sum_c e^{(\beta_0^{(c)} + \mathbf{w}^{(c)} \cdot \mathbf{x})}}$$

3.2.3 Naive Bayes

Ο Naive Bayes είναι ένας πιθανοτικός ταξινομητής ο οποίος χρησιμοποιεί το Θεώρημα του Bayes για να υπολογίσει την εκ των υστέρων (posteriori) πιθανότητα $P(y|\mathbf{x})$ για κάθε κλάση $c \in C^k$ δοθέντος του διανύσματος χαρακτηριστικών :

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y) \cdot P(y)}{P(\mathbf{x})}$$

Όμοια με την περίπτωση της Λογιστικής Παλινδρόμησης, η αναμενόμενη κλάση $y^p = c_i$ είναι εκείνη με τη βέλτιστη πιθανότητα $P(y = c_i|\mathbf{x})$. Ωστόσο, ο Naive Bayes δεν υπολογίζει απευθείας την εκ των υστέρων πιθανότητα αλλά βασίζεται στην πιθανότητα $P(\mathbf{x}|y)$. Η πιθανότητα αυτή - γνωστή ως *likelihood* - αναφέρεται στο πόσο πιθανό είναι να παραχθεί ένα δεδομένο με διάνυσμα χαρακτηριστικών $\mathbf{x}$ θεωρώντας δεδομένη την κλάση y στην οποία ανήκει.

Από τη Σχέση παρατηρούμε ότι η πιθανότητα $P($**x** είναι ανεξάρτητη της μεταβλητής y και σταθερή για όλες τις κλάσεις. Συνεπώς, η εκ των υστέρων πιθανότητα προκύπτει :

$$P(y|\mathbf{x}) \propto P(\mathbf{x}|y) \cdot P(y)$$

Η πιθανότητα εμφάνισης κάθε κλάσης $P(y)$ μπορεί να υπολογιστεί από το σύνολο δεδομένων. Ωστόσο, η $P(\mathbf{x}|y)$ εξαρτάται από τη συνδυασμένη κατανομή πιθανότητα των $\mathbf{x}$ και y και ο υπολογισμός της είναι ιδιαίτερα απαιτητικός ακόμη σε μικρά σύνολα δεδομένων καθώς το $\mathbf{x}$ είναι μία πολυδιάστατη τυχαία μεταβλητή.

Ο ταξινομητής Naive Bayes κάνει την παραδοχή ότι κάθε ζεύγος μεταβλητών (x_i, x_j), $i \neq j$ είναι ανεξάρτητες μεταξύ τους δοθείσης της κλάσης y οπότε : $P(x_i|x_j, y) = P(x_i|y)$ για κάθε ζεύγος $i, j \in [1, n]$. Εφαρμόζοντας τον κανόνα της αλυσίδας σε συνδυασμό με την παραπάνω παραδοχή προκύπτει ότι :

$$P(\mathbf{x}|y) = P(x_1, \cdots, x_n|y) = \prod_{i=1}^{n} P(x_i|y)$$

Επομένως, χρησιμοποιώντας το παραπάνω μοντέλο η εκ των υστέρων πιθανότητα υπολογίζεται ως :

$$P(y|\mathbf{x}) \propto P(y) \cdot \prod_{i=1}^{n} P(x_i|y) = P(y) \cdot \cdots \cdot P(x_1|y) \cdot P(x_n|y)$$

και η αναμενόμενη κλάση y^p προσδιορίζεται από τη σχέση :

$$y^p = classify(\mathbf{f}) = \arg\max_c P(y = c) \cdot \prod_{i=1}^{n} P(x_i = f_i | y = c)$$

Ο Naive Bayes αναφέρεται γενικά στην υπό συνθήκη ανεξαρτησία των μεταβλητών - χαρακτηριστικών x_i αφήνοντας απροσδιόριστη την κατανομή πιθανότητά τους. Αν θεωρήσουμε πως κάθε χαρακτηριστικό x_i ακολουθεί πολυωνυμική κατανομή τότε προκύπτει ο Πολυωνυμικός Naive Bayes (Multinomial Naive Bayes) όπου :

$$P(\mathbf{x}|y) = \frac{(\sum_i x_i)!}{(\prod_i x_i)!} \cdot \prod_{i=1}^{n} p_i^{(x_i)}$$

3.2.4 Πολυεπίπεδο Perceptron

Τα Τεχνητά Νευρωνικά Δίκτυα είναι μαθηματικά μοντέλα επεξεργασίας δεδομένων που αποτελούνται από ένα πλήθος τεχνητών νευρώνων οργανωμένων σε δομές παρόμοιες με αυτές των βιολογικών νευρωνικών δικτύων όπως ο ανθρώπινος εγκέφαλος. Ένας από τους πιο διαδεδομένους τύπους τεχνητών νευρωνικών δικτύων είναι το πολυεπίπεδο νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης (multilayer feed- forward network) ή πολυεπίπεδο perceptron (multilayer perceptron). Στο μοντέλο αυτό οι τεχνητοί νευρώνες είναι οργανωμένοι σε μία σειρά από στρώματα ή επίπεδα (layers). Το πρώτο από αυτά τα επίπεδα ονομάζεται επίπεδο εισόδου (input layer) και χρησιμοποιείται για την εισαγωγή των δεδομένων. Τα στοιχεία του δεν είναι ουσιαστικά νευρώνες γιατί δεν εκτελούν κάποιο υπολογισμό. Στη συνέχεια, μπορεί να ακολουθούν, προαιρετικά, ένα ή περισσότερα ενδιάμεσα ή κρυφά επίπεδα (hidden layers), ενώ στο τέλος υπάρχει το επίπεδο εξόδου(output layer). Το νευρωνικό δίκτυο ονομάζεται εμπρόσθιας τροφοδότησης διότι επιτρέπονται συνδέσεις μόνο μεταξύ νευρώνων διαδοχικών στρωμάτων οπότε η ροή πληροφορίας είναι πρόσθιας πληροφορίας. Παράλληλα, είναι πλήρως συνδεδεμένα καθώς κάθε νευρώνας σε ένα επίπεδο συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου.

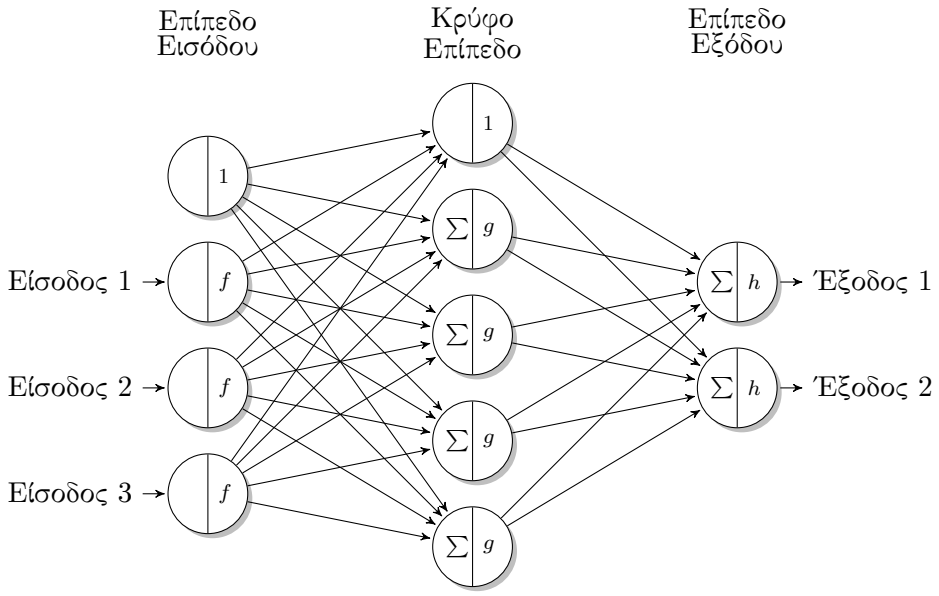
Οι McCulloch and Pitts (1943) πρότειναν την ιδέα ενός τεχνητού νευρώνα j ο οποίος υπολογίζει μία συνάρτηση g ως σταθμισμένο άθροισμα των n εισόδων :

$$y_i(x) = g(\sum_{i=0}^{n} w_i x_i)$$

όπου $(w_0, w_1, \cdots, w_n)$ είναι οι συντελεστές βαρύτητας ή βάρη που εφαρμόζονται στις εισόδους $(x_0, x_1, \cdots, x_n)$. Σε ένα πολυεπίπεδο νευρωνικό δίκτυο η έξοδος y_i δημιουργεί μέρος της εισόδου που θα δοθεί στους τους νευρώνες του επόμενου επιπέδου. Η συνάρτηση ενεργοποίησης g είναι συνήθως μία από τις παρακάτω :

- βηματική συνάρτηση (step function) : ανάλογα με την τιμή του αθροίσματος και την παράμετρο κατωφλίου T_{thres} προκύπτει $y_j \in \{0, 1\}$
- συνάρτηση προσήμου (sign function) : ανάλογα με την τιμή του αθροίσματος και την παράμετρο κατωφλίου T_{thres} προκύπτει $y_j \in [-1, 1]$

- λογιστική συνάρτηση (logistic function) : ανάλογα με την τιμή του αθροίσματος και την μορφή της σιγμοειδούς συνάρτησης h_{sig} προκύπτει $y_j \in [0, 1]$



Σχήμα 3.5: Πολυεπίπεδο Perceptron εμπρόσθιας τροφοδότησης με ένα κρύφο επίπεδο και κόμβους πόλωσης

Το πολυεπίπεδο perceptron ταξινομεί τα δεδομένα υλοποιώντας μία συνάρτηση μεταφοράς $\mathcal{T}$ η οποία συνδέει την είσοδο (διάνυσμα χαρακτηριστικών) με την έξοδο (κλάση στην οποία ανήκει το εκάστοτε δεδομένο).

Κατά τη διάρκεια της εκπαίδευσης, μέσω της μεθόδου οπισθοδιάδοσης (backpropagation) οι παράμετροι των συναρτήσεων ενεργοποίησης σε συνδυασμό με τους συντελεστές βαρύτητας των νευρώνων αναπροσαρμόζονται επαναληπτικά έτσι ώστε να βελτιστοποιηθεί η συνάρτηση μεταφοράς $\mathcal{T}$. Για να απλοποιηθεί η διαδικασία μπορεί να προστεθούν κόμβοι πόλωσης (bias nodes) με σταθερή τιμή εξόδου 1 σε κάθε επίπεδο πλην της εξόδου έτσι ώστε να αποπλεχθεί η έξοδος από τις παραμέτρους των συναρτήσεων ενεργοποίησης και η αναπροσαρμογή του δικτύου να εξαρτάται μόνο από την ενημέρωση των συντελεστών βαρύτητας.

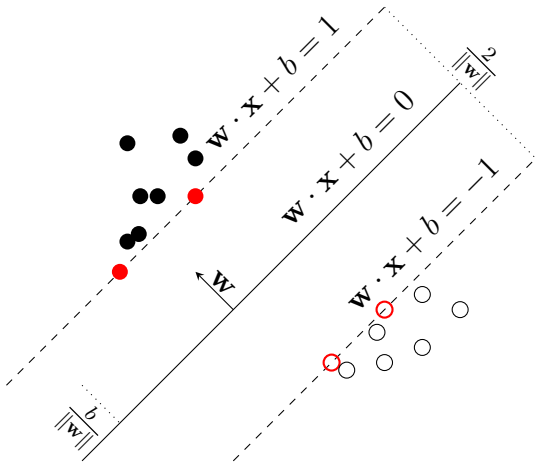
Η έξοδος του ταξινομητή προκύπτει στο στάδιο εκτέλεσης όπου οι τιμές του διανύσματος χαρακτηριστικών διαδίδονται από το επίπεδο εισόδου σε όλο το υπόλοιπο δίκτυο. Αποδεικνύεται πως ένα νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης με ένα κρυφό επίπεδο μπορεί να προσεγγίσει οποιαδήποτε μη γραμμική συνάρτηση.

3.2.5 Μηχανές Διανυσμάτων Υποστήριξης

Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines - SVMs) είναι μηπιθανοτικοί γραμμικοί δυαδικοί ταξινομητές. Προτάθηκαν το 1992 από τους Vapnik et al. ως μία νέα μέθοδος μάθησης, παρόλο που η γενικότερη ιδέα στην οποία στηρίζονται είχε προταθεί από τη δεκατία '60. Συνδυάζουν στοιχεία από τη Θεωρία Στατιστικής Μάθησης και τα Νευρωνικά Δίκτυα τύπου Perceptron.

Έστω σύνολο εκπαίδευσης $\mathcal{D}$ με n πρότυπα της μορφής $\mathcal{D} = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in \mathbb{R}^p \ , y_i \in \{-1, 1\}\}$, $i \in [1, n]$ όπου $\mathbf{x}_i$ το διάνυσμα χαρακτηριστικών p-διαστάσεων του δείγματος i και y_i η κλάση στην οποία ανήκει (-1 ή 1). Μία Μηχανή Διανυσμάτων Υποστήριξης προσπαθεί να προσδιορίσει το βέλτιστο υπερεπίπεδο (hyperplane) $\mathbf{w} \cdot \mathbf{x} - b = 0$ όπου $\mathbf{w}$ το κάθετο διάνυσμα του υπερεπιπέδου το οποίο ορίζεται από τον χώρο χαρακτηριστικών $\mathbb{R}^p$. Το βέλτιστο υπερεπίπεδο είναι αυτό που μεγιστοποιεί την απόσταση μεταξύ των αρνητικών και θετικών δειγμάτων του συνόλου εκπαίδευσης (maximum margin hypersurface).

Η διαδικασία μάθησης ενός SVM μπορεί να μοντελοποιηθεί ως πρόβλημα βελτιστοποίησης : αν τα δεδομένα είναι γραμμικά διαχωρίσιμα τότε τα δύο υπερεπίπεδα περιγράφονται από τις σχέσεις $\mathbf{w} \cdot \mathbf{x} - b = 1$ και $\mathbf{w} \cdot \mathbf{x} - b = -1$ οπότε η μεταξύ τους απόσταση είναι $\frac{2}{\|\mathbf{w}\|}$. Επομένως, για να προσδιορίσουμε το βέλτιστο υπερεπίπεδο θα πρέπει να ελαχιστοποιήσουμε το $\|\mathbf{w}\|$.



Σχήμα 3.6: Μηχανή Διανυσμάτων Υποστήριξης για δεδομένα δύο κλάσεων

Προσθέτοντας περιορισμούς έτσι ώστε να μην απεικονίζονται δεδομένα στο διάστημα μεταξύ των υπερεπιπέδων, το πρόβλημα βελτιστοποίησης λαμβάνει την εξής μορφή με τη βοήθεια των πολλαπλασιαστών Lagrange λ_i :

$$\mathcal{L}(\mathbf{w}, b, \lambda_1, \cdots, \lambda_n) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \lambda_i [y_i(\mathbf{w} \cdot \mathbf{x} - b) - 1]$$

Η αντικειμενική συνάρτηση $\mathcal{L}$ πρέπει να ελαχιστοποιηθεί ως προς τα $\mathbf{w}$ και b και να μεγιστοποιηθεί ως προς τα λ_i. Δηλαδή :

$$\arg \min_{w,b} \max_{\lambda \geq 0} \mathcal{L}$$

Εφαρμόζοντας τις Συνθήκες Karush-Kuhn-Tucker το βέλτιστο υπερεπίπεδο προκύπτει από τη σχέση :

$$g^*(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

Μία Μηχανή Διανυσμάτων υποστήριξης μπορεί να ταξινομεί περιπτώσεις που είναι παρόμοιες αλλά όχι πανομοιότυπες με κάποιο πρότυπο εκπαίδευσης. Το αποτέλεσμα εξόδου

είναι τελικά μία αριθμητική τιμή στο διάστημα $[-1, 1]$ και όχι κάποια πιθανότητα όπως σε άλλους ταξινομητές.

Για το γενικότερο πρόβλημα της κατηγοριοποίησης δεδομένων σε περισσότερες από δύο κλάσεις έχει προταθεί η κατασκευή ενός συνόλου δυαδικών ταξινομητών SVM με δύο εκδοχές :

(i) one-versus-all : κάθε ταξινομητής διαχωρίζει ανάμεσα σε μία κλάση και όλες τις υπόλοιπες. Η τελική κλάση είναι αυτή με το μεγαλύτερη τιμή εξόδου (winner-takes-all strategy).

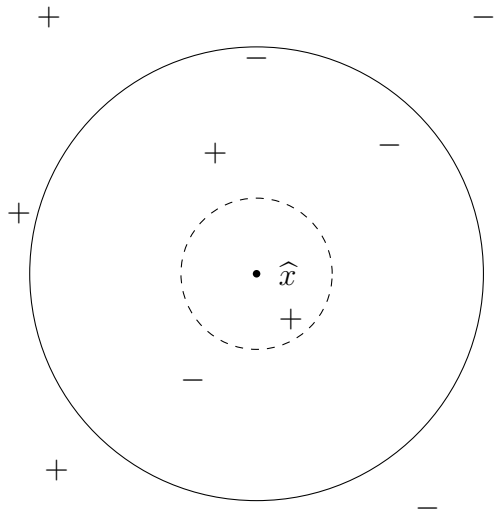
(ii) one-versus-one : κάθε ταξινομητής διαχωρίζει ανάμεσα σε ένα ζεύγος κλάσεων. Για την κατηγοριοποίηση των δεδομένων, πραγματοποιείται μία διαδικασία ψηφοφορίας όπου κάθε ταξινομητής αντιστοιχεί το εκάστοτε δεδομένο εισόδου σε μία από τις δύο κλάσεις και η κλάση με τις περισσότερες ψήφος αναδεικνύεται νικήτρια (max-wins voting strategy).

Το βασικότερο πλεονέκτημα των SVMs έναντι των νευρωνικών δικτύων τύπου Perceptron είναι ότι μπορούν και παράγουν πιο σύνθετες υπερεπιφάνειες, ενσωματώνοντας μετασχηματισμούς και συνδυασμούς των αρχικών μεταβλητών ανάλογα με το πρόβλημα και ξεπερνούν προβλήματα όπως τα τοπικά ελάχιστα και η διασπορά των λύσεων στο χώρο αναζήτησης. Για το σκοπό αυτό, χρησιμοποιούν ένα πεπερασμένο αριθμό υποσυνόλων του συνόλου εκπαίδευσης (τα διανύσματα υποστήριξης) καθώς και συναρτήσεις πυρήνα (kernel functions) προκειμένου να μετασχηματίσουν τον αρχικό χώρο υποθέσεων και να βρουν τη βέλτιστη μη γραμμική υπερεπιφάνεια που ελαχιστοποιεί το σφάλμα αναζήτησης.[57]

3.2.6 k-Κοντινότεροι Γείτονες

Ο ταξινομήτης k-Κοντινότερων Γειτόνων (k-Nearest Neighbors - k-NN) ανήκει στην κατηγορία αλγορίθμων μάθησης κατά περίπτωση (instance-based learning). Σε αντίθεση με τις μεθόδους που αναφέρθηκαν ως τώρα και οι οποίες μετασχηματίζουν τα πρότυπα εκπαίδευσης σε συμπαγή, στη μάθηση κατά περίπτωση τα δεδομένα διατηρούνται αυτούσια. Όταν ένα σύστημα κληθεί να αποφασίσει για την κατηγορία ενός νέου δεδομένου εισόδου, εξετάζει εκείνη τη στιγμή τη σχέση του με τα ήδη αποθηκευμένα παραδείγματα. Η μέθοδος αναβάλλει τη μάθηση έως ότου εμφανιστεί νέο στιγμιότυπο και για το λόγο αυτό ονομάζεται οκνηρή μάθηση (lazy learning) σε αντίθεση με τις υπόλοιπες οι οποίες χαρακτηρίζονται ως πρόθυμες μέθοδοι μάθησης (eager learners) καθώς κατασκευάζουν άμεσα το μοντέλο από το σύνολο εκπαίδευσης χωρίς να περιμένουν για την άφιξη μίας νέας περίπτωσης. [57]

Στον αλγόριθμο k-Κοντινότερων Γειτόνων γίνεται η παραδοχή ότι τα διάφορα πρότυπα μπορεί να αναπαρασταθούν ως σημεία σε κάποιο n-διάστατο Ευκλείδειο χώρο $\mathbb{R}^n$ όπου n ο το πλήθος των χαρακτηριστικών εισόδου. Κάθε νέο στιγμιότυπο τοποθετείται στο χώρο αυτό ως νέο σημείο και η κλάση στην οποία ανήκει προσδιορίζεται σύμφωνα με την πλειοψηφία των αποφάσεων των k πλησιέστερων σημείων που προέρχονται από τα πρότυπα εκπαίδευσης $\mathcal{D}$. Οι κοντινότεροι γείτονες ενός στιγμιότυπου υπολογίζονται με συνήθως με βάση την Ευκλείδεια απόστασή τους.



Σχήμα 3.7: Προσδιορισμός κατηγορίας με βάση τον 1 και τους 5 κοντινότερους γείτονες

Η απόσταση ενός νέου στιγμιότυπου $\widehat{x}$ με σύνολο χαρακτηριστικών $\{a_1(\widehat{x}), a_2(\widehat{x}), \cdots, a_n(\widehat{x})\}$ και ενός αποθηκευμένου πρότυπου x με αντίστοιχο σύνολο χαρακτηριστικών $\{a_1(x), a_2(x), \cdots, a_n(x), y(x)\}$ είναι το άθροισμα των Ευκλείδειων αποστάσεων όλων των χαρακτηριστικών των δύο σημείων του χώρου $\mathbb{R}^n$:

$$dist(\widehat{x}, x) = \sqrt{\sum_{i=1}^{n}(a_i(x) - a_i(\widehat{x}))^2}$$

Στο Σχήμα 3.7 όπου αναπαρίστανται πρότυπα δύο κλάσεων, το νέο στιγμιότυπο χαρακτηρίζεται ως θετικό αν ληφθεί υπόψη μόνο ο πλησιέστερος γείτονας και ως αρνητική αν ληφθούν υπόψη οι πέντε πλησιέστεροι γείτονες καθώς η πλειοψηφία αυτών έχει αρνητικό χαρακτηρισμό (εξωτερικός κύκλος). [57].

4

Αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα της πειραματικής μελέτης που πραγματοποιήσαμε με κύριο στόχο τη διερεύνηση της συμπεριφορά της μεθόδου γράφων ν-γραμμάτων σε ένα πολυγλωσσικό και πολυθεματικό περιβάλλον,τη συμβατότητά της με βασικούς αλγορίθμους Μηχανικής Μάθησης και τέλος την ανταπόκριση της σε σειρά παρεμβάσεων στα διάφορα στάδια της διαδικασίας. Στο πλαίσιο αυτό, εξετάζουμε αλγορίθμους που ανήκουν σε διαφορετικές κατηγορίες μάθησης και αναζητούμε τους πλέον κατάλληλους.

Συγκεκριμένα, συγκρίνουμε τους εξής αλγορίθμους :

(i) C4.5

(ii) Λογιστική Παλινδρόμηση

(iii) Naive Bayes

(iv) Naive Bayes Multinomial

(v) Μηχανή Διανυσμάτων Υποστήριξης

(vi) Πολυεπίπεδο Perceptron

(vii) k-Κοντινότεροι Γείτονες

4.1 Παράμετροι Αξιολόγησης

Η αξιολόγηση των ταξινομητών πραγματοποιείται χρησιμοποιώντας την πλέον διαδεδομένη μετρική απόδοσης , την ακρίβεια κατηγοριοποίησης α :

$$\alpha = \frac{\sum_{c \in \mathcal{C}} |\text{σωστά ταξινομημένα δεδομένα ως c}|}{|\text{δεδομένα εξέτασης}|}$$

όπου $\mathcal{C} = \{c_{neg}, c_{pos}, c_{neu}\}$ το σύνολο των κατηγοριών πολικότητας

Επιπρόσθετα, θέλοντας να μελετήσουμε τη διακριτική ικανότητα των ταξινομητών ως προς τις κατηγορίες, εξετάζουμε επίσης τους δείκτες ακρίβειας (precision) και ανάκλησης (recall) καθώς και του αρμονικού τους μέσου όρου F_1 που ορίζονται ως :

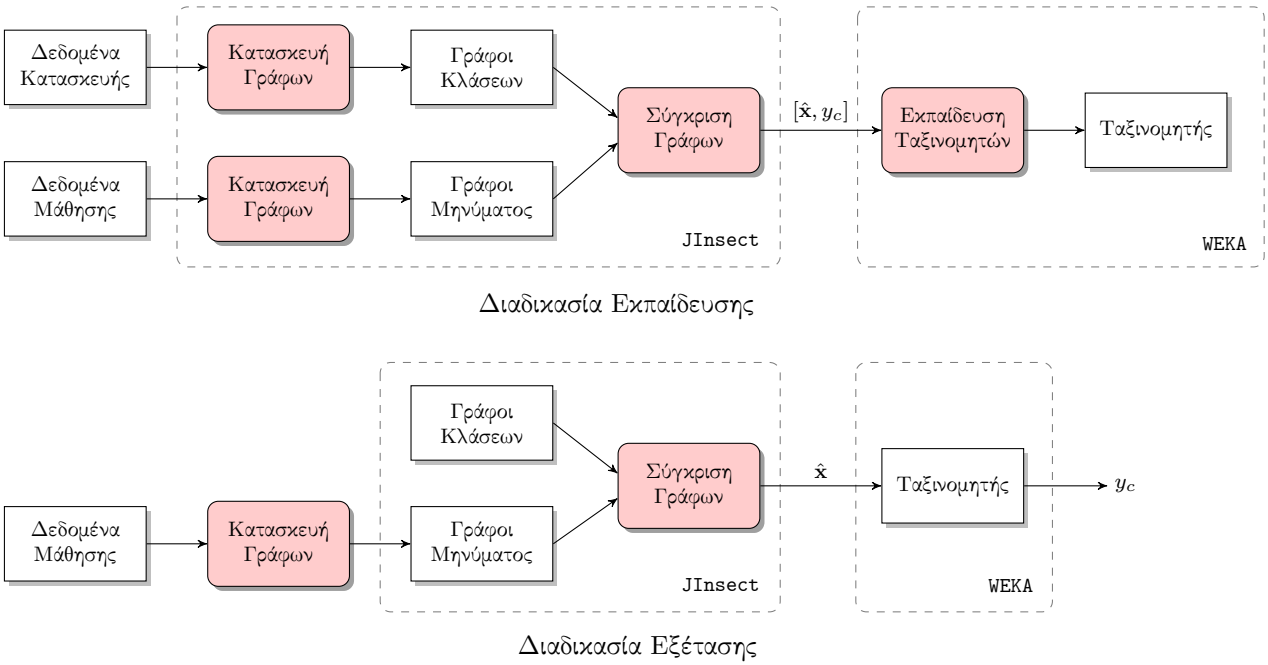
$$Precision = \frac{\sum_{c \in \mathcal{C}} \frac{|\text{σωστά ταξινομημένα δεδομένα ως c}|}{|\text{δεδομένα που ταξινομήθηκαν ως c}|}}{|\mathcal{C}|}$$

$$Recall = \frac{\sum_{c \in \mathcal{C}} \frac{|\text{σωστά ταξινομημένα δεδομένα ως c}|}{|\text{δεδομένα που ανήκουν στην κατηγορία c}|}}{|\mathcal{C}|}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

4.2 Σύστημα Αξιολόγησης

Για την αξιολόγηση των αλγορίθμων κατηγοριοποίησης σχεδιάστηκε και αναπτύχθηκε Σύστημα Ανάλυσης Συναισθήματος στη γλώσσα προγραμματισμού Java, Έκδοση 1.7.55. Η λειτουργία του συστήματος προϋποθέτει τη διάσπαση του συνόλου των δεδομένων σε τρία μη επικαλυπτόμενα υποσύνολα : κατασκευής, μάθησης και εξέτασης. Η ροή εκτέλεσης του συστήματος - όπως απεικονίζεται στο Σχήμα 4.1 - περιλαμβάνει τα εξής διαδοχικά στάδια :



Σχήμα 4.1: Λειτουργικότητες Εκπαίδευσης και Εξέτασης

(i) Κατασκευή Γράφων Κλάσης : κατασκευάζονται οι γράφοι – από τα tweets της αντίστοιχης κλάσης των δεδομένων κατασκευής

(ii) Εκπαίδευση Ταξινομητών : για κάθε tweet του συνόλου μάθησης δημιουργείται ο γράφος μηνύματος ο οποίος στη συνέχεια συγκρίνεται με τους γράφους κλάσης, υπολογίζονται οι δείκτες ομοιότητας και μαζί με την κλάση στην οποία έχει ταξινομηθεί το tweet χρησιμοποιούνται ως διάνυσμα εισόδου για την εκπαίδευση των ταξινομητών .

(iii) Εξέταση Ταξινομητών : η διαδικασία υπολογισμού των δεικτών ομοιότητας επαναλαμβάνεται και για κάθε tweet του συνόλου εξέτασης. Έπειτα, οι εκπαιδευμένοι ταξινομητές βάσει των δεικτών ομοιότητας προβλέπουν την πολικότητα του μηνύματος.

Η πλήρης λειτουργικότητα των γράφων ν-γραμμάτων (κατασκευή γράφων και υπολογισμός των δεικτών ομοιότητας) πραγματοποιείται μέσω της βιβλιοθήκης `JInsect`[1]. Για την εφαρμογή των αλγορίθμων κατηγοριοποίησης χρησιμοποιήθηκε η βιβλιοθήκη `WEKA`[2] [55], Έκδοση 3.6.11. Με στόχο την αξιολόγηση της γενικής συμπεριφοράς του συστήματος, εφαρμόστηκαν οι προκαθορισμένες συνθέσεις των αλγορίθμων κατηγοριοποίησης, χωρίς καμία αναζήτηση βέλτιστων παραμέτρων (fine-tuning). Συγκεκριμένα, χρησιμοποιήθηκαν οι εξής κλάσεις ταξινομητών της βιβλιοθήκης :

(i) `J48` : η υλοποίηση του αλγορίθμου C4.5 με ενεργοποιημένη τη λειτουργία κλαδέματος (pruning)

(ii) `Logistic` : η πολυωνυμική εκδοχή της Λογιστικής Παλινδρόμησης γνωστή και ως Maximum Entropy

(iii) `NaiveBayes` : το μοντέλο NaiveBayes χωρίς τη δυνατότητα ενημέρωσης

(iv) `NaiveBayesMultinomial` : η πολυωνυμική εκδοχή του `NaiveBayes`

(v) `MultilayerPerceptron` : το μοντέλο του Πολυεπίπεδου Perceptron με αριθμό κρυφών επιπέδων $\frac{1}{2} \cdot (\text{σύνολο χαρακτηριστικών} + \text{σύνολο κλάσεων}) = 6$ (προκαθορισμένη τιμή)

(vi) `SMO` : η υλοποίηση της Μηχανής Διανυσμάτων Υποστήριξης χρησιμοποιώντας πολυωνυμική συνάρτηση πυρήνα `PolyKernel` και στρατηγική 1-vs-1.

(vii) `IBk` : η υλοποίηση του αλγορίθμου k-Κοντινότερων Γειτόνων με $k = 3$

Όλα τα πειράματα πραγματοποιήθηκαν σε μηχάνημα με λειτουργικό σύστημα Linux (Έκδοση Πυρήνα 3.13.0-27), επεξεργαστή Intel i7 960 (4-cores) και μνήμη RAM 18 GB.

4.3 Οργάνωση Πειραμάτων

Οι αλγόριθμοι κατηγοριοποίησης αξιολογούνται με τη χρήση ενός συνόλου δεδομένων που απαρτίζεται από χειροκίνητα ταξινομημένα σύνολα δεδομένων διαθέσιμα ελεύθερα

[1] http://sourceforge.net/projects/jinsect/
[2] http://www.cs.waikato.ac.nz/ml/weka/

στην ερευνητική κοινότητα (Πίνακας 4.1) - αρκετά από τα οποία έχουν χρησιμοποιηθεί σε σχετικές εργασίες. Η μελέτη των αλγορίθμων ειδικά σε χειροκίνητα ταξινομημένα δεδομένα υπήρξε από τους βασικούς στόχους της εργασίας επεκτείνοντας την έρευνα των Aisopos et al. [3] και ακολουθώντας παράλληλα τη γενικότερη τάση της ερευνητικής περιοχής. Η συγκεκριμένη απαίτηση - αν και εξαλείφει τους παράγοντες τύχης και προκατάληψης των αυτόματων μεθόδων κατηγοριοποίησης - περιορίζει σημαντικά τον όγκο των εξεταζόμενων δεδομένων.

Λόγω της Πολιτικής Χρήσης και Προστασίας του Προσωπικού Απορρήτου του Twitter, δεν επιτρέπεται η δημοσίευση του περιεχομένου (κειμένου) των μηνυμάτων αλλά μόνο των αναγνωριστικών (tweetIDs) των μηνυμάτων τα οποία μπορούν να χρησιμοποιηθούν για να ανακτηθεί το αρχικό κείμενο μέσω του `TwitterAPI`. Ωστόσο, αρκετοί από τους χρήστες - δημιουργούς των μηνυμάτων επιλέγουν να διαγράψουν ή να κλειδώσουν το λογαριασμό τους με αποτέλεσμα μέρος της συλλογής των δεδομένων να μην είναι πλέον ανακτήσιμο.

Οι βασικές πληροφορίες για τα σύνολα δεδομένων που χρησιμοποιήσαμε αυτούσια συνοψίζονται στον Πίνακα 4.1.

Dataset	**Δημιουργός**	**Γλώσσα**	**Πλήθος**	**Θετικά**	**Αρνητικά**	**Ουδέτερα**
Arabic Twitter Corpus (RRArabic)[1]	Refaee et al.[37]	Αραβικά	5821	767	1670	3384
HealthCare Reform (HCR)[2]	Speriosu et al.[46]	Αγγλικά	2392	541	1381	470
Obama-McCain Debate (OMD)[2]	Shamma et al.[45]	Αγγλικά	1904	709	1195	-
Multi-DaiLabor (MDL)[3]	Narr et al.[28]	Αγγλικά	10594	2334	1486	6774
		Γαλλικά	2155	500	481	1174
		Γερμανικά	2637	496	334	1807
		Πορτογαλικά	2395	923	627	845
Manual Groundtruth (NTUA)[4]	Aisopos et al.	Αγγλικά	500	159	119	222
Sanders Twitter[5]	Sanders	Αγγλικά	3152	473	513	2166
SemEval-2014 Task 9 (SEM)[6]	Rosenthal et al.[38]	Αγγλικά	12838	4855	1986	5997
SentiTuites-PT [7]	Moreira et al.	Πορτογαλικά	10075	1290	5413	3372
Sentiment Strength (SSTweet)[8]	Thelwall et al.[48]	Αγγλικά	4242	1252	1037	1953
Tromp MultiLingual (Tromp)[9]	Tromp [50]	Αγγλικά	12128	4885	3691	3552
		Ολλανδικά	5086	1277	1601	2208
TASS 2013 Corpus (TASS)[10]	Villena et al.[52]	Ισπανικά	45482	24589	18161	2732
Σύνολο			**121401**	**45050**	**39695**	**36656**

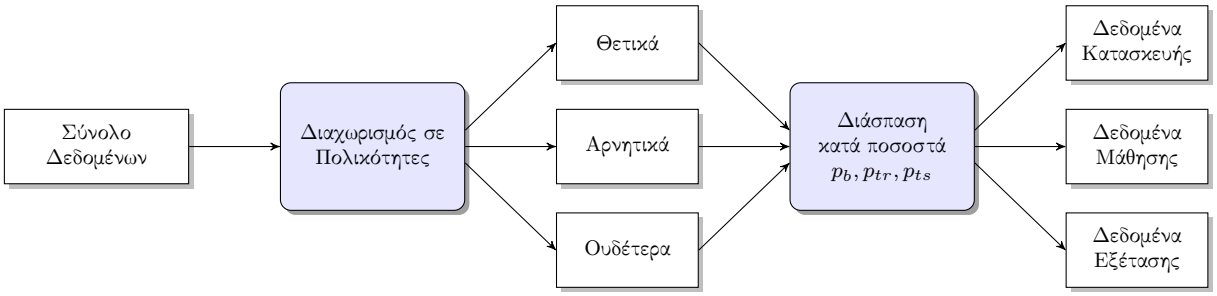
Πίνακας 4.1: Σύνολα χειροκίνητα ταξινομημένων δεδομένων

Για την ανάκτηση του περιεχομένου σε σύνολα δεδομένων που δεν περιείχαν το αρχικό κείμενο χρησιμοποιήθηκε η βιβλιοθήκη της JAVA `twitter4j`.

Όπως έχει αναλυθεί στην Ενότητα 4.2, για τη λειτουργία του συστήματος Ανάλυσης Συναισθήματος απαιτείται η διάσπαση του συνόλου δεδομένων σε δεδομένα κατασκευής, μάθησης και εξέτασης. Αξίζει να τονισθεί ότι η επίδοση ενός συστήματος επιβλεπόμενης μάθησης στο σύνολο εξέτασης εξαρτάται άμεσα από την αντιπροσωπευτικότητα των

[1] http://www.macs.hw.ac.uk/~eaar1/Eshrag%20Refaee
[2] https://bitbucket.org/speriosu/updown/wiki/Home
[3] http://data.dai-labor.de/corpus/sentiment/
[4] https://github.com/dtz/NTUA-THESIS-Twitter-Sentiment-Analysis
[5] http://www.sananalytics.com/lab/twitter-sentiment/
[6] http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools
[7] http://dmir.inesc-id.pt/project/SentiTuites-PT_01_in_English
[8] http://sentistrength.wlv.ac.uk/documentation/
[9] http://www.win.tue.nl/~mpechen/projects/smm/
[10] http://www.daedalus.es/TASS2013/corpus.php

δεδομένων - στην περίπτωσή μας κατασκευής και μάθησης - που χρησιμοποιήθηκαν κατά τη διαδικασία της εκπαίδευσης. Επιλέγουμε, λοιπόν, το διαχωρισμό των δεδομένων κάθε επιμέρους συνόλου με τυχαία σειρά ποσοστιαία στα σύνολα κατασκευής, μάθησης και εξέτασης. Παρατηρούμε πως αν και στο τελικό υπερσύνολο δεδομένων οι κατηγορίες πολικότητας είναι σχετικά ισοκατανεμημένες με ποσοστά 37.1% (θετικά) 32.7% (αρνητικά) και 30.2% (ουδέτερα), στα περισσότερα σύνολα δεδομένων κυριαρχεί μία συγκεκριμένη κατηγορία (Σχήμα 4.3). Επομένως, επεκτείνουμε τη διαδικασία διάσπασης στα σύνολα κατασκευής, μάθησης και εξέτασης επιλέγοντας ποσοστιαία ανά κατηγορία πολικότητας από κάθε σύνολο δεδομένων.



Σχήμα 4.2: Διαδικασία Διάσπασης Συνόλου Δεδομένων.

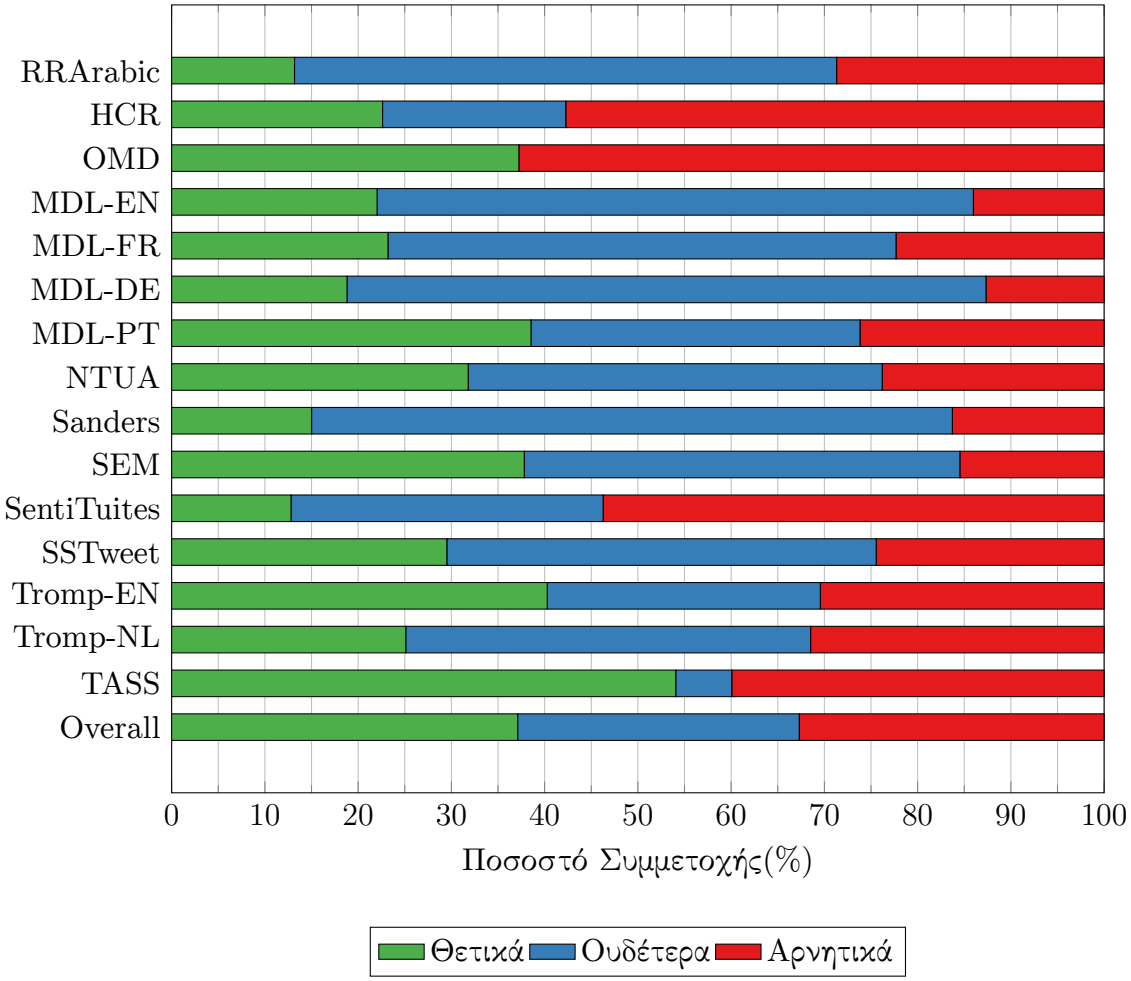
Μέσω αυτής της διαδικασίας του Σχήματος 4.2 κάθε σύνολο δεδομένων συμμετέχει αναλογικά σε κάθε στάδιο λειτουργίας του συστήματος ενώ παράλληλα διατηρείται η κατανομή της πολικότητας του συναισθήματος και εξασφαλίζεται η αντιπροσωπευτικότητα του συνόλου εκπαίδευσης. Με τον τρόπο αυτό αποκλείεται η περίπτωση το σύστημα να υπερ-εξειδικευτεί σε μία γλώσσα ή κατηγορία πολικότητας που δε συναντάται σε τόσο μεγάλο βαθμό στο σύνολο εξέτασης.

4.4 Αποτελέσματα

4.4.1 Επιλογή Ποσοστών Διάσπασης

Μία από τις κυριότερες αποφάσεις κατά τη διαδικασία αξιολόγησης ενός μοντέλου Επιβλεπόμενης Μηχανικής Μάθησης είναι η επιλογή της κατάλληλης αναλογίας μεταξύ των συνόλων εκπαίδευσης και εξέτασης. Για δεδομένο πλήθος προτύπων, χρησιμοποιώντας ένα μεγαλύτερο σύνολο εξέτασης η εκτιμώμενη ακρίβεια εμφανίζει μικρότερη διακύμανση και προκύπτει μία πιο αξιόπιστη εικόνα της συμπεριφοράς του μοντέλου ενώ με τη χρήση ενός μεγαλύτερου συνόλου εκπαίδευσης επιτυγχάνεται πιο αντιπροσωπευτική μάθηση. Στην περίπτωσή μας το σύνολο εκπαίδευσης απαρτίζεται από τα δεδομένα κατασκευής και μάθησης.

Για τον προσδιορισμό της βέλτιστης αναλογίας ανάμεσα στα δεδομένα κατασκευής (**b**uilset), μάθησης (**tr**aining set) και εξέτασης (**t**est **s**et) διερευνήθηκαν διάφοροι συνδυασμοί των ποσοστών διάσπασης. Στο Σχήμα 4.4 απεικονίζεται ο μέσος όρος της ακρίβειας των ταξινομητών στην καλύτερη, μέση και χειρότερη περίπτωση - όπως αυτές προκύπτουν χρησιμοποιώντας μέγεθος ν-γράμματος n από 1 έως και 7 - για τις διάφορες εκδοχές διάσπασης του συνόλου των δεδομένων.

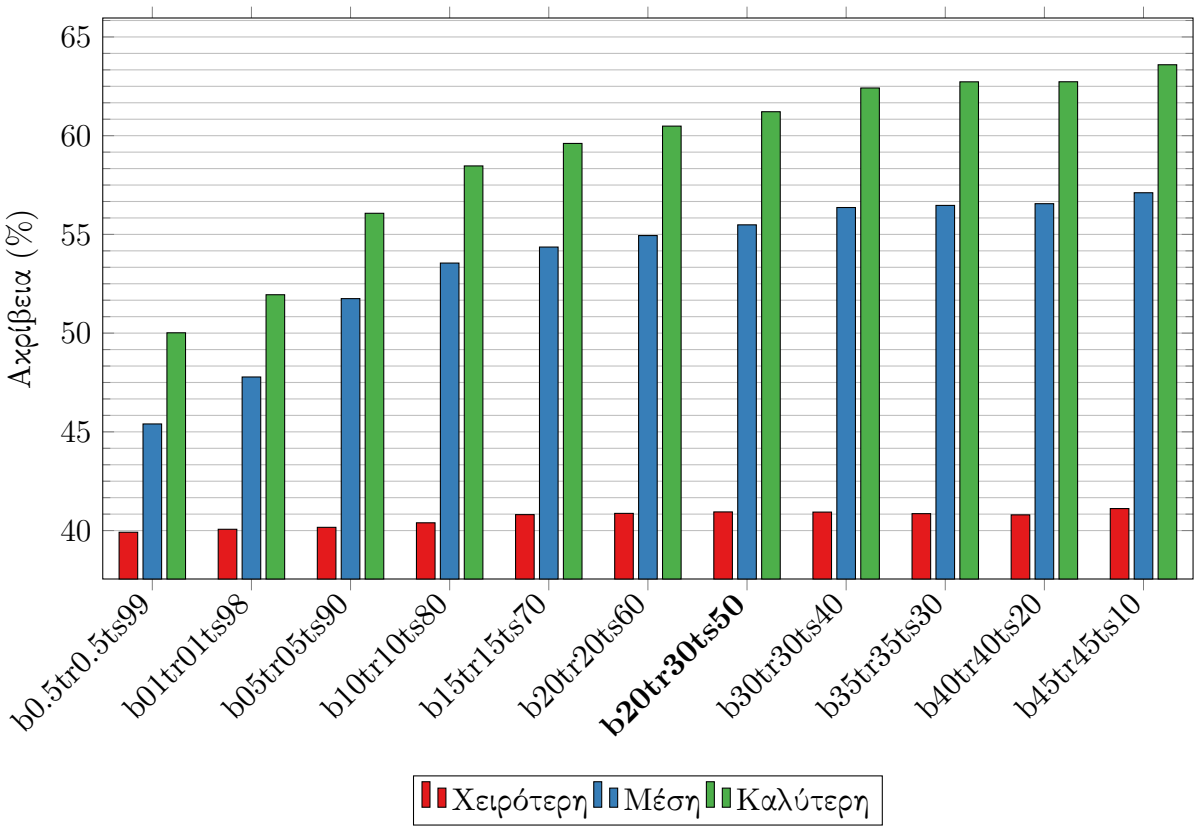


Σχήμα 4.3: Κατανομή Κλάσεων Πολικότητας ανά Υποσύνολο Δεδομένων

Παρατηρούμε πως σε σχέση με άλλες μεθόδους επιβλεπόμενης μάθησης, το εξεταζόμενο μοντέλο Ανάλυσης Συναισθήματος εμφανίζει αρκετά ικανοποιητική συμπεριφορά όταν χρησιμοποιείται πολύ μικρό σύνολο εκπαίδευσης π.χ. στην περίπτωση του συνδυασμού b0.5tr0.5ts99, με αναλογία 1:99 σε πληθυσμό 121401 tweets, επιτυγχάνει μέση ακρίβεια κατηγοριοποίησης 45.9%,αυξημένη κατά 12% σε σχέση με αυτή ενός τυχαίου ταξινομητή (33,33%).

Ωστόσο, διαπιστώνουμε ακόμη πως εκπαιδεύοντας το μοντέλο με περισσότερα πρότυπα, βελτιώνεται η μέγιστη ακρίβεια - η οποία είναι και το βασικό αντικείμενο της έρευνας - αλλά οδηγούμαστε γρήγορα σε κορεσμό : εξετάζοντας τις περιπτώσεις των συνδυασμών b30tr30tr40 έως και b45tr45ts10 παρατηρούμε πως η μέση τιμή της ακρίβειας των ταξινομητών και στις τρεις περιπτώσεις - καλύτερη, μέση, χειρότερη - κυμαίνεται σε παρόμοια επίπεδα καθώς το πλήθος των δεδομένων εκπαίδευσης αυξάνεται και αντίστοιχα το πλήθος των δεδομένων εξέτασης μειώνεται, σημειώνοντας βέλτιστη επίδοση 68% στην περίπτωση b45tr45ts10. Συμπεραίνουμε, λοιπόν, πως για το συγκεκριμένο μέγεθος και εσωτερική διάρθρωση του συνόλου, η μέθοδος Ανάλυσης Συναισθήματος με Γράφους ν-γραμμάτων δεν είναι ισχυρά εξαρτώμενη από το πλήθος των δεδομένων εκπαίδευσης.

Στόχος της εργασίας είναι η μελέτη και αξιολόγηση του μοντέλου όχι σε ακραίες συνθήκες όπως π.χ. στις περιπτώσεις των συνδυασμών b0.5tr0.5ts99 και b45tr45ts10 με



Σχήμα 4.4: Ακριβεία Κατηγοριοποίησης ανά συνδυασμό ποσοστών διάσπασης

ανάλογα ποσοστά ακριβείας αλλά σε κανονικές συνθήκες λειτουργίας έτσι ώστε να εξασφαλίζεται η ευσταθής συμπεριφορά του. Συνεπώς, στο υπόλοιπο μέρος της εργασίας εξετάζουμε την περίπτωση του συνδυασμού b20tr30ts50 όπου επιτυγχάνεται επαρκής αντιστάθμιση ανάμεσα στην ακρίβεια κατηγοριοποίησης και στο μέγεθος του συνόλου εξέτασης.

4.4.2 Επιλογή Μεγέθους ν-γράμματος

Όπως αναφέρεται στο [3], ένας γράφος ν-γραμμάτων χαρακτηρίζεται από τρεις παραμέτρους : (i) τον ελάχιστο βαθμό ν-γράμματος L_{min} (ii) το μέγιστο βαθμό ν-γράμματοςL_{max} και (iii) τη μέγιστη απόσταση γειτνίασης - μήκος κυλιόμενου παράθυρου L_{min}. Ακολουθούμε την προσέγγιση των Aisopos et al. και εξετάζουμε αποκλειστικά την περίπτωση όπου $L_{min} = L_{max} = D_{win}$, η οποία πειραματικά έχει αποδειχθεί ότι οδηγεί σε επίδοση συγκρίσιμη της βέλτιστης, όπως προέκυψε από ενδελεχή αναζήτηση βέλτιστων παραμέτρων (fine-tuning) [12].

Στον Πίνακα 4.2 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα εκτέλεσης των αλγορίθμων ταξινόμησης για μεγέθη ν-γράμματος $n = \{1, 2, \cdots, 7\}$

Όσο αφορά το μέγεθος ν-γράμματος, παρατηρούμε ότι με εξαίρεση τον Πολυωνυμικό Naive Bayes (MNB), οι υπόλοιποι αλγόριθμοι ταξινόμησης εμφανίζουν παρόμοια συμπεριφορά : καθώς αυξάνεται η τιμή του n,βελτιώνονται η ακρίβεια μέχρι την τιμή $n = 4$ (στην περίπτωση του NaiveBayes $n = 5$) όπου επιτυγχάνεται η βέλτιστη επίδοση η οποία στη συνέχεια φθίνει όταν χρησιμοποιούνται γράφοι μεγαλύτερου βαθμού. Συμπε-

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	43.08	39.70	**44.92**	40.25	38.56	36.97	43.52
2	57.78	56.27	**59.19**	48.48	44.75	36.97	55.17
3	64.22	64.47	**64.61**	55.19	51.71	36.98	62.20
4	65.48	**65.62**	65.61	57.25	55.90	40.15	63.88
5	65.15	65.08	**65.31**	56.84	57.51	42.82	63.58
6	64.00	63.75	**64.27**	55.90	55.35	44.94	62.82
7	62.50	62.15	**62.91**	54.62	51.23	46.31	62.20

Πίνακας 4.2: Ποσοστά Ακρίβειας Ταξινομητών για ν-γράμματα 1 έως 7

ραίνουμε, λοιπόν, πως η τιμή $n = 4$ αποτελεί την καλύτερη επιλογή στην πλειονότητα των ταξινομητών όπου αξιοποιούνται στο μέγιστο τα πλεονεκτήματα της μεθόδου : ανεξαρτησία γλώσσας και υψηλή ανοχή στο θόρυβο. Γράφοι ν-γραμμάτων μικρού βαθμού (μονογράμματα, διγράμματα) εξαιτίας του μικρού μεγέθους τους αδυνατούν να συλλάβουν τα πιο χαρακτηριστικά εκφραστικά μοτίβα του πολυγλωσσικού περιεχομένου και οδηγούν σε χαμηλά ποσοστά ακρίβειας. Στους γράφους ν-γραμμάτων μεγάλου βαθμού αυξάνεται εκθετικά το πλήθος των κορυφών και ακμών με αποτέλεσμα η επιπλέον πληροφορία που αποτυπώνεται στους γράφους να μην αντιστοιχεί στην πραγματική εξάρτηση που υπάρχει ανάμεσα στις λέξεις-συμβοακολουθίες και να αποτελεί ουσιαστικά θόρυβο. Αυτό εξηγεί το λόγο για τον οποίο τα ποσοστά ακρίβειας για $n = 6, 7$ υπολείπονται μεν της βέλτιστης επίδοσης, κυμαίνονται δε σε ικανοποιητικά επίπεδα, αρκετά υψηλότερα των αντίστοιχων ποσοστών για $n = 1, 2$.

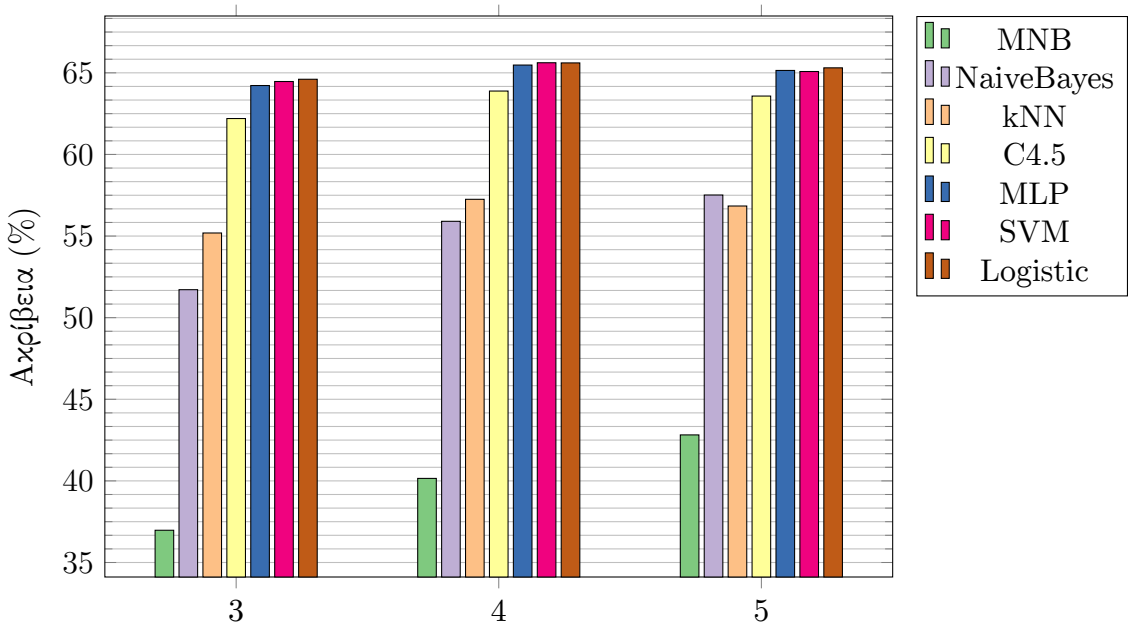
Ο Πολυωνυμικός Naive Bayes ακολουθεί τη συμπεριφορά των υπόλοιπων ταξινομητών, βελτιώνοντας τη ακρίβεια καθώς αυξάνεται το μέγεθος ν-γράμματος αλλά με πολύ πιο αργό ρυθμό οπότε και παρουσιάζει το καλύτερο αποτέλεσμα στην περίπτωση $n = 7$. Διαπιστώνουμε, λοιπόν, πως ο συγκεκριμένος ταξινομητής δεν εμφανίζει την ίδια ευαισθησία ως προς το μέγεθος ν-γράμματος με τους υπόλοιπους. Το φαινόμενο αυτό εξηγεί τα πολύ χαμηλά ποσοστά ακρίβειας στο εύρος $n \in [1, 7]$ και οφείλεται πιθανότατα στην εσφαλμένη υπόθεση ότι οι δείκτες ομοιότητας ακολουθούν πολυωνυμική κατανομή.

4.4.3 Επιλογή Καταλληλότερου Ταξινομητή

Στο Σχήμα 4.5 απεικονίζονται γραφικά τα ποσοστά ακρίβειας των ταξινομητών για μεγέθη ν-γράμματος $n = 3, 4$ και 5.

Η βέλτιστη ακρίβεια κατηγοριοποίησης επιτυγχάνεται στην περίπτωση $n = 4$ από τους ταξινομητές : SVM (65.62%), Λογιστικής Παλινδρόμησης (65.61%) και Πολυεπίπεδου Perceptron (65.48%). Ακολουθούν σε φθίνουσα σειρά επίδοσης οι C4.5 (63.88%), Naive Bayes (57.51%, $n = 5$), k-NN(57.25%) και Πολυωνυμικός Naive Bayes(46.31%, $n = 7$).

Συγκρίνοντας τους τρεις καλύτερους αλγορίθμους ως προς τη γενικότερη επίδοση, παρατηρούμε πως αν και ανήκουν σε διαφορετικές κατηγορίες Μηχανικής Μάθησης, εμφανίζουν παρόμοια αποτελέσματα σε όλες τις τιμές ν-γράμματος που εξετάσαμε. Κρίνεται,λοιπόν , σκόπιμο να μελετήσουμε πιο αναλυτικά τον τρόπο με τον οποίο επιτεύχθηκε το συγκεκριμένο ποσοστό ακρίβειας εξετάζοντας την ικανότητα των ταξινομητών στην



Σχήμα 4.5: Ακρίβεια Κατηγοριοποίησης για μεγέθη ν-γράμματος 3,4 και 5.

κατηγοριοποίηση ανά κατηγορία πολικότητας με τη βοήθεια του πίνακα σφάλματος - σύγχυσης (confusion matrix).

Στον πίνακα παρουσιάζονται οι τιμές της ακρίβειας(precision-PR), ανάκλησης(recall-RC), του συνδυασμού τους F_1 για κάθε κατηγορία όπως επίσης και ο σταθμισμένος μέσος όρος τους για τους τρεις ταξινομητές στην περίπτωση $n = 4$.

	MLP			**SVM**			**Logistic**		
Κλάση	**PR**	**RC**	**F_1**	**PR**	**RC**	**F_1**	**PR**	**RC**	**F_1**
Θετικά	0.759	0.587	0.662	0.729	0.620	0.670	0.689	0.672	0.680
Αρνητικά	0.646	0.657	0.651	0.661	0.620	0.640	0.660	0.620	0.639
Ουδέτερα	0.593	0.745	0.660	0.592	0.740	0.658	0.617	0.676	0.645

Πίνακας 4.3: Precision, Recall και F_1-score των MLP, SVM και Logistic

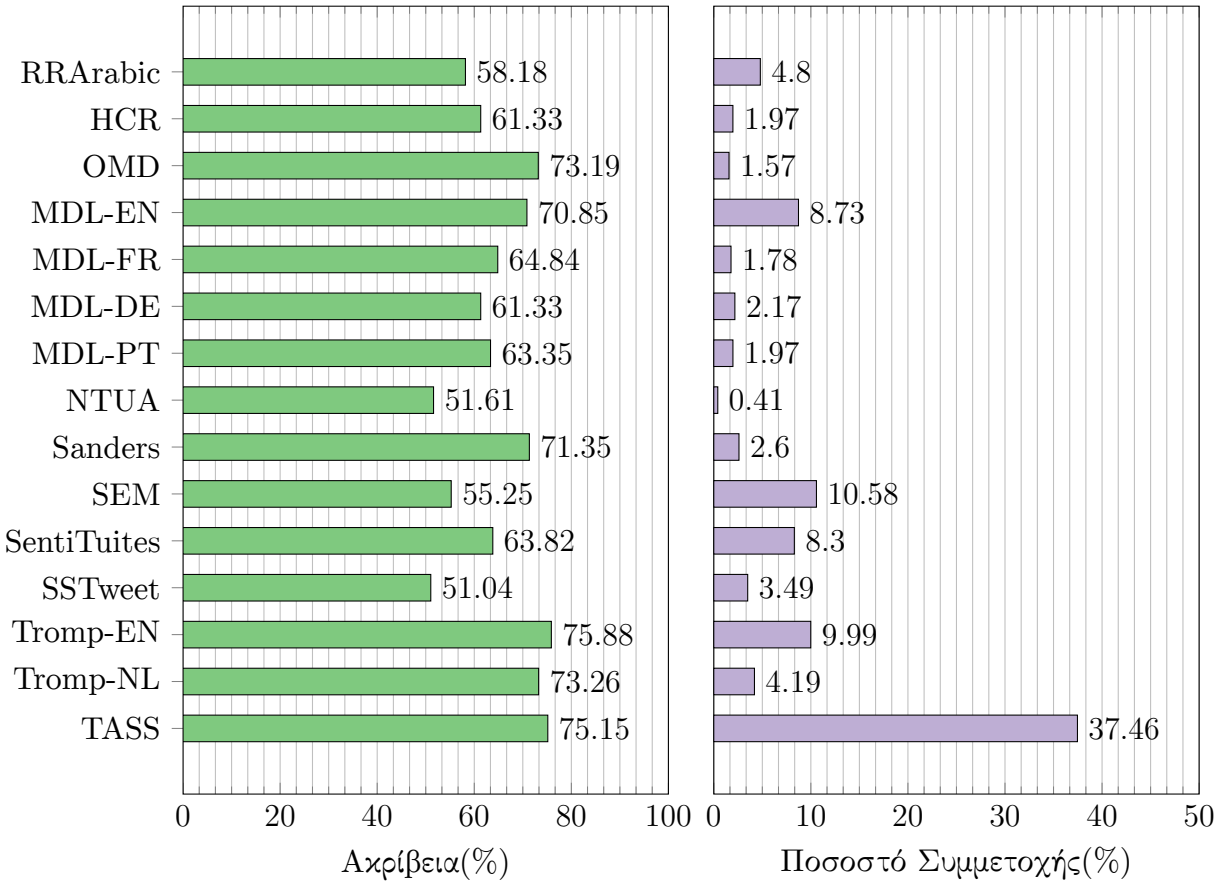
Διαπιστώνουμε πως οι ταξινομητές εμφανίζουν παρεμφερή χαρακτηριστικά και κατά την ανάλυση σε επίπεδο κλάσεων πολικότητας. Σύμφωνα με τους δείκτες ακρίβειας και ανάκλησης προκύπτει ότι και οι τρεις αλγόριθμοι είναι περισσότερο ακριβείς στην ταξινόμηση δεδομένων στη θετική κατηγορία και αναγνωρίζουν επιτυχώς μεγαλύτερο ποσοστό των ουδέτερων tweets χωρίς - ωστόσο - σημαντικές επιπτώσεις στα αντίστοιχα μεγέθη των υπόλοιπων κατηγοριών. Η ισορροπία που επικρατεί μεταξύ στις δύο αντιστρόφως ανάλογες έννοιες αντικατοπτρίζεται στην αρκετά υψηλή τιμή του δείκτη F_1 στις επιμέρους κατηγορίες καθώς και στην περίπτωση του σταθμισμένου μέσου όρου.

Επομένως, χρησιμοποιώντας οποιονδήποτε από τους ταξινομητές SVM, Λογιστική Παλινδρόμηση και Πολυεπίπεδο Perceptron, το σύστημα Ανάλυσης Συναισθήματος ανταποκρίνεται επαρκώς και στις τρεις κατηγορίες των tweets χωρίς αισθητή προκατάληψη - φαινόμενο που υποδηλώνει αντιπροσωπευτική εκπαίδευση και οδηγεί σε ικανοποιητικά ποσοστά ακρίβειας κατηγοριοποίησης.

4.4.4 Συμπεριφορά στα επιμέρους σύνολα δεδομένων

Εξετάζουμε τη συμπεριφορά του μοντέλου στα υποσύνολα των δεδομένων για να διαπιστωθεί αν υπάρχουν διαφοροποιήσεις σε σχέση με το σύνολο και να προσδιορισθεί ο βαθμός τους. Κατά τη πειραματική διερεύνηση σε κάθε υποσύνολο χρησιμοποιήθηκαν τα ίδια σύνολα δεδομένων κατασκευής, μάθησης και εξέτασης όπως αυτά προέκυψαν από τη διαδικασία της διάσπασης (Σχήμα 4.2).

Τα αναλυτικά αποτελέσματα παρατίθενται στο Παράρτημα Α΄. Στο σχήμα 4.6 απεικονίζονται για κάθε υποσύνολο η υψηλότερη ακρίβεια κατηγοριοποίησης ανάμεσα σε όλους τους ταξινομητές που εξετάσαμε και το ποσοστό συμμετοχής του στο σύνολο των δεδομένων.



Σχήμα 4.6: Βέλτιστη Ακρίβεια Κατηγοριοποίησης και Ποσοστό Συμμετοχής των Υποσυνόλων Δεδομένων

Παρατηρούμε πως η ακρίβεια δεν είναι ανάλογη του μεγέθους του κάθε υποσυνόλου. Χαρακτηριστικές περιπτώσεις είναι τα σύνολα OMD και SEM με ποσοστά επί του συνόλου 1.57% και 10.58% και ποσοστά ακρίβειας 73.19% και 55.25%. Συνεπώς, επιβεβαιώνεται πειραματικά ότι παράγοντες που συντελούν στην υψηλή ακρίβεια είναι σε μεγάλο βαθμό η ομοιογένεια και ποιότητα του κάθε συνόλου καθώς και η σωστή και αντικειμενική κατηγοριοποίηση του από τους αξιολογητές και όχι το μέγεθός του.

Όσον αφορά την επίδοση των ταξινομητών, οι τρεις αλγόριθμοι με τα υψηλότερα ποσοστά ακρίβειας στο σύνολο - SVM, Λογιστική Παλινδρόμηση και Πολυεπίπεδο Perceptron - διατηρούνται σε υψηλά επίπεδα σε όλα τα επιμέρους υποσύνολα. Παράλληλα, αξίζει να

σημειωθεί πως αλγόριθμοι με αισθητά χαμηλότερα ποσοστά ακρίβειας στο σύνολο όπως ο Πολυωνυμικός Naive Bayes (MNB) και ο k-NN, εμφανίζονται ιδιαίτερα ανταγωνιστικά σε αρκετά υποσύνολα δεδομένων (MNB : RRArabic, MDL-{DE,EN}, Sanders; k-NN : MDL-{DE,EN,FR,PT}, TASS). Αυτό οφείλεται στο γεγονός ότι σε κάθε υποσύνολο εν γένει επικρατεί μία ομοιομορφία ως προς τη θεματολογία και τη χρήση λέξεων-φράσεων με συγκεκριμένο συναισθηματικό προσανατολισμό η οποία αποτυπώνεται με μεγαλύτερη ευκρίνεια σε σχέση με το σύνολο από τους γράφους ν-γραμμάτων και τους εξαγόμενους δείκτες ομοιότητας.

Η ακρίβεια κατηγοριοποίησης εμφανίζει παραβολική συμπεριφορά ως προς το μέγεθος ν-γράμματος, σημειώνοντας μέγιστο για $n = 4$ στα περισσότερα υποσύνολα σε όλους τους ταξινομητές εκτός του Πολυωνυμικού Naive Bayes, όπως ακριβώς συμβαίνει και στο σύνολο. Υπάρχουν, εντούτοις, εξαιρέσεις με πιο χαρακτηριστική περίπτωση υποσυνόλου αυτή του RRArabic όπου παρατηρούνται αυξομειώσεις της ακρίβειας καθώς αυξάνεται ο βαθμός των γράφων. Το φαινόμενο αυτό μπορεί να αποδοθεί στην ιδιαίτερη φύση της κάθε γλώσσας καθώς και τη συγκεκριμένη εσωτερική δομή των κειμένων του κάθε υποσυνόλου.

Καταλήγουμε στο συμπέρασμα ότι η αποτελεσματικότητα της μεθόδου εξαρτάται σε μεγάλο βαθμό από τα χαρακτηριστικά του συνόλου στο οποίο εξετάζεται. Στα υποσύνολα TASS, Tromp -{EN,NL} και OMD που αποτελούν το 53.2% του σύνολου επιτυγχάνει ακρίβεια 73-75% ενώ στα υποσύνολα NTUA, SEM και SSTweet που συμμετέχουν κατά 14.5% στο σύνολο εμφανίζει ποσοστά της τάξης του 51-55%. Ο σταθμισμένος μέσος όρος της καλύτερης ακρίβειας στα επιμέρους υποσύνολα υπολογίζεται σε 68.8%, συγκρίσιμη επίδοση με το 65.6% της μεθόδου στο σύνολο. Ωστόσο, η εφαρμογή της μεθόδου στο σύνολο των δεδομένων προσπαθεί να επιλύσει ένα πιο δύσκολο πρόβλημα σε σχέση με τα επιμέρους υποσύνολα καθώς η συγχώνευσή τους δημιουργεί ένα πολυγλωσσικό σύνολο με αρκετά έντονη διαφοροποίηση ως προς τη θεματολογία.

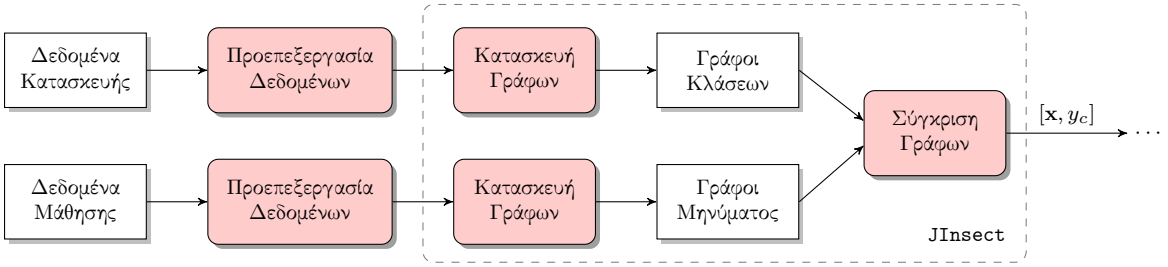
4.4.5 Τροποποιήσεις Αρχιτεκτονικής Συστήματος

Στην Ενότητα αυτή τροποποιούμε διαδοχικά κάθε στάδιο λειτουργίας του μοντέλου και μελετάμε την απόκριση του συστήματος στις μεταβολές της αρχιτεκτονικής του.

4.4.5.1 Προεπεξεργασία Δεδομένων

Για να διατηρήσουμε την ανεξαρτησία της μεθόδου από τη γλώσσα, οι παρεμβάσεις στην αρχική μορφή των δεδομένων περιορίζονται στις ειδικές λέξεις και ακολουθίες χαρακτήρων που χρησιμοποιούνται στο Twitter και στις οποίες θα αναφερόμαστε με τον όρο tokens. Συγκεκριμένα, εξετάζουμε τις αναφορές (mentions) σε λογαριασμούς χρηστών (`@username`), τις αναδημοσιεύσεις (`RT` και `via`), τους υπερσυνδέσμους (`http`) και τα hashtags (`#topicName`).

Κατά τη διεξαγωγή των πειραμάτων, επιχειρήθηκαν τρεις διαφορετικές εκδοχές προεπεξεργασίας : (i) αφαίρεση όλων των Twitter tokens (ii) αφαίρεση μόνο των υπερσυνδέσμων (iii) αντικατάσταση των tokens με συγκεκριμένες λέξεις κλειδιά (keywords : [mention], [retweet], [url], [hashtag]) στα δεδομένα κατασκευής και μάθησης. Τα ποσο-



Σχήμα 4.7: Τροποποίηση Προεπεξεργασίας Δεδομένων

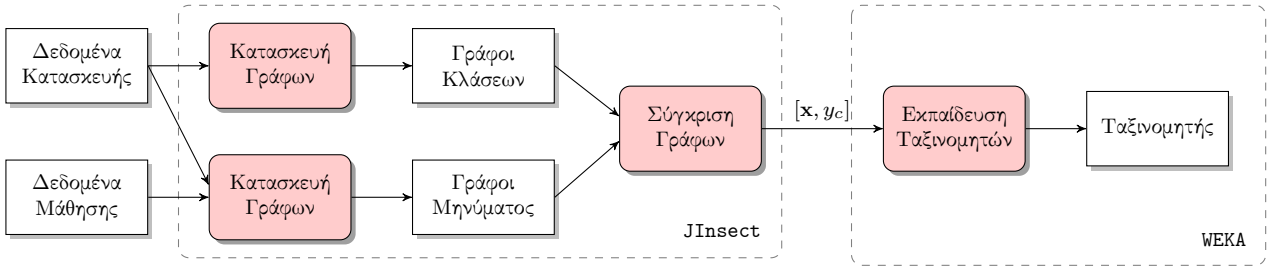
στά ακρίβειας κάθε ταξινομητή και η μεταβολή τους σε σχέση με την αρχική σύνθεση παρουσιάζονται στους Πίνακες για κάθε προσέγγιση ξεχωριστά.

Είναι εμφανές ότι οποιαδήποτε παρέμβαση δεν οδηγεί σε επίτευξη καλύτερου αποτελέσματος. Οι ταξινομητές με την καλύτερη επίδοση επηρεάζονται αρνητικά, εμφανίζοντας μείωση κατά 1-2% στην περίπτωση της βέλτιστης ακρίβειας ($n = 4$). Επιπλέον, όλοι οι ταξινομητές - εκτός του MNB ο οποίος παραμένει στα ίδια επίπεδα - εμφανίζουν αισθητή μείωση στην περίπτωση των διγραμμάτων. Τέλος, μόνο ο αλγόριθμος k-NN ενισχύεται από την αντικατάσταση των tokens και την αφαίρεση των υπερσυνδέσμων.

Συγκρίνοντας τις διαφορετικές προσεγγίσεις, γίνεται αντιληπτό πως - αν και καμία δεν έχει θετική συμβολή - η διαγραφή των υπερσυνδέσμων μόνο, επιβαρύνει το λιγότερο σε σχέση με τις υπόλοιπες τεχνικές τη συμπεριφορά του μοντέλου. Λαμβάνοντας υπόψη πως η συγκεκριμένη εκδοχή πραγματοποιεί και τις λιγότερες αλλαγές, καταλήγουμε στο συμπέρασμα ότι οι γράφοι ν-γραμμάτων δεν αποπροσανατολίζονται από τα Twitter tokens αλλά ως ένα βαθμό επωφελούνται από την ύπαρξή τους.

4.4.5.2 Χρήση Δεδομένων Κατασκευής στην Εκπαίδευση

Όπως απεικονίζεται στο Σχήμα 4.8, κατά την εκπαίδευση των ταξινομητών, μαζί με τα δεδομένα μάθησης χρησιμοποιούμε και τα δεδομένα κατασκευής, ακολουθώντας την προσέγγιση των Aisopos et al.[2].



Σχήμα 4.8: Τροποποίηση Χρήσης Δεδομένων Κατασκευής στη Διαδικασία Εκπαίδευσης

Αναλύοντας τα πειρατικά αποτελέσματα του Πίνακα , αξίζει να αναφερθούν η επίτευξη της καλύτερης επίδοσης με οριακή αύξηση 0.25% και η θεαματική βελτίωση των ποσοστών ακρίβειας του Naive Bayes και ιδιαίτερα της πολυωνυμικής εκδοχής του, MNB.

NGram	MLP		SVM		Logistic		kNN		NaiveBayes		MNB		C4.5	
1	41.52	(-1.55)	38.58	(-1.12)	43.46	(-1.47)	37.65	(-2.60)	36.89	(-1.67)	36.97	(0.00)	38.76	(-4.77)
2	51.76	(-6.02)	50.72	(-5.56)	57.36	(-1.82)	43.61	(-4.87)	41.04	(-3.71)	36.97	(0.00)	48.02	(-7.15)
3	62.45	(-1.77)	63.02	(-1.45)	63.28	(-1.33)	51.47	(-3.72)	47.05	(-4.66)	36.97	(-0.00)	58.18	(-4.02)
4	64.71	(-0.77)	64.58	(-1.04)	64.82	(-0.79)	53.87	(-3.39)	51.95	(-3.96)	39.62	(-0.54)	62.60	(-1.29)
5	63.87	(-1.28)	63.69	(-1.39)	64.48	(-0.83)	53.93	(-2.91)	54.53	(-2.98)	42.52	(-0.30)	62.81	(-0.77)
6	62.77	(-1.22)	62.30	(-1.45)	63.50	(-0.77)	53.07	(-2.83)	53.81	(-1.54)	44.80	(-0.14)	61.76	(-1.06)
7	61.43	(-1.07)	60.49	(-1.66)	62.05	(-0.86)	51.95	(-2.68)	50.15	(-1.08)	46.40	(+0.08)	61.11	(-1.10)

Πίνακας 4.4: Ποσοστά Ακρίβειας μετά την αφαίρεση Twitter tokens

NGram	MLP		SVM		Logistic		kNN		NaiveBayes		MNB		C4.5	
1	42.39	(-0.69)	39.20	(-0.50)	45.07	(+0.15)	38.50	(-1.75)	38.95	(+0.39)	36.97	(0.00)	43.04	(-0.48)
2	54.31	(-3.47)	54.37	(-1.91)	58.37	(-0.82)	47.78	(-0.70)	44.10	(-0.65)	36.97	(0.00)	51.73	(-3.44)
3	63.86	(-0.36)	63.92	(-0.55)	63.89	(-0.72)	56.48	(+1.29)	50.88	(-0.83)	36.98	(+0.00)	61.24	(-0.96)
4	65.53	(+0.05)	65.47	(-0.15)	65.33	(-0.28)	59.06	(+1.81)	55.25	(-0.65)	40.15	(+0.00)	63.93	(+0.05)
5	64.79	(-0.35)	64.93	(-0.15)	65.17	(-0.14)	58.76	(+1.92)	57.14	(-0.37)	42.93	(+0.11)	63.78	(+0.20)
6	63.73	(-0.27)	63.52	(-0.23)	64.35	(+0.08)	58.05	(+2.15)	55.77	(+0.41)	45.28	(+0.34)	62.89	(+0.07)
7	62.21	(-0.29)	61.77	(-0.38)	62.92	(+0.01)	56.58	(+1.95)	51.23	(0.00)	46.88	(+0.57)	61.77	(-0.43)

Πίνακας 4.5: Ποσοστά Ακρίβειας μετά την αφαίρεση υπερσυνδέσμων

NGram	MLP		SVM		Logistic		kNN		NaiveBayes		MNB		C4.5	
1	40.47	(-2.61)	39.62	(-0.08)	43.64	(-1.28)	38.15	(-2.09)	39.47	(+0.91)	36.97	(0.00)	41.28	(-2.24)
2	54.85	(-2.93)	53.44	(-2.84)	56.94	(-2.25)	47.16	(-1.32)	41.71	(-3.04)	36.97	(0.00)	51.38	(-3.79)
3	62.20	(-2.02)	62.32	(-2.14)	62.10	(-2.51)	55.79	(+0.60)	48.63	(-3.08)	36.98	(0.00)	59.27	(-2.93)
4	64.75	(-0.73)	64.23	(-1.39)	64.14	(-1.47)	58.11	(+0.86)	53.15	(-2.75)	39.66	(-0.50)	62.49	(-1.40)
5	64.10	(-1.05)	63.96	(-1.12)	64.30	(-1.00)	57.87	(+1.03)	55.60	(-1.91)	42.48	(-0.34)	62.98	(-0.59)
6	62.96	(-1.04)	62.65	(-1.10)	63.41	(-0.86)	57.34	(+1.44)	54.39	(-0.96)	44.64	(-0.30)	62.42	(-0.40)
7	61.48	(-1.02)	61.05	(-1.10)	62.08	(-0.83)	55.91	(+1.29)	50.49	(-0.75)	46.06	(-0.25)	61.29	(-0.92)

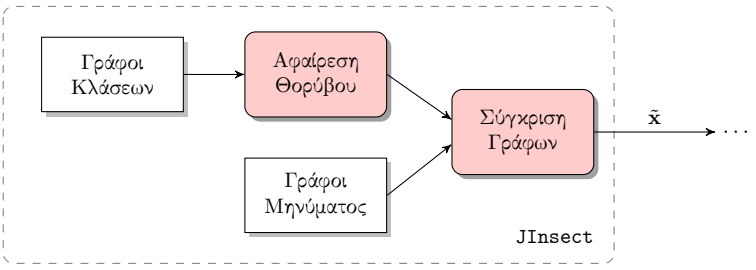
Πίνακας 4.6: Ποσοστά Ακρίβειας μετά την αντικατάσταση Twitter tokens

Η αύξηση του συνόλου εκπαίδευσης παρέχει μία πιο λεπτομερή εικόνα των χαρακτηριστικών της κάθε κατηγορίας πολικότητας η οποία - όπως αποδεικνύεται - ευνοεί τους ταξινομητές που στηρίζονται σε στατιστικά μοντέλα.

Ωστόσο, η συγκεκριμένη τροποποίηση ενέχει τον κίνδυνο της προκατάληψης καθώς τα δεδομένα που χρησιμοποιούνται και στα δύο στάδια λειτουργίας του συστήματος, εμφανίζουν υψηλότερο βαθμό ομοιότητας συγκριτικά με τα δεδομένα μάθησης και αποκτούν με τον τρόπο αυτό - αδίκως - μεγαλύτερη βαρύτητα στη διαδικασία της εκπαίδευσης. Συνεπώς, χρήζει περαιτέρω διερεύνησης το κατά πόσο η συγχώνευση των δύο ξένων υποσυνόλων διατηρεί την αντιπροσωπευτικότητα της εκπαίδευσης και συντελεί σε συστηματική βάση στη βελτίωση της ακρίβειας.

4.4.5.3 Αφαίρεση Θορύβου

Όπως αναλύεται στο [12], κατά την αναπαράσταση κειμένων μέσω γράφων ν-γραμμάτων σε ένα πρόβλημα κατηγοριοποίησης, είναι αναπόφευκτη η εμφάνιση θορύβου στους γράφους. Στο πλαίσιο της Ανάλυσης Συναισθήματος, ο θόρυβος είναι το σύνολο των κοινών υπογράφων ανάμεσα στους γράφους κλάσεων G_{neg}, G_{neu} και G_{pos}. Στις κλασικές τεχνικές κατηγοριοποίησης - όπως περιγράφονται στην Ενότητα 2.6 - το φαινόμενο θορύβου αντιμετωπίζεται σε ένα βαθμό αφαιρώντας κυρίως άρθρα και άλλες λέξεις χωρίς συναισθηματικό προσανατολισμό (stopwords). Ωστόσο, στους γράφους ν-γραμμάτων δεν είναι απαραίτητη κανενός είδους προεπεξεργασία - η οποία άλλωστε υποδηλώνει εξάρτηση από συγκεκριμένα γλωσσικά εργαλεία - καθώς ο "θόρυβος" μπορεί να αφαιρεθεί μέσω των τελεστών που διαθέτει το μοντέλο.



Σχήμα 4.9: Τροποποίηση Αφαίρεση Θορύβου από Γράφους Κλάσεων

Ο μέγιστος κοινός υπογράφος είναι ουσιαστικά η τομή όλων των γράφων που αντιστοιχούν στις κατηγορίες πολικότητας. Στο σημείο αυτό αξίζει να σημειωθεί ότι ο κοινός υπογράφος που προκύπτει δεν είναι μοναδικός. Αν και ο τελεστής της τομής είναι εν γένει αντιμεταθετικός, η χρήση της μέση τιμής κατά την ανανέωση των βαρών των κοινών ακμών σε κάθε εφαρμογή του άρει την αντιμεταθετικότητα του τελικού αποτελέσματος. Δηλαδή : $(G_{neg} \cap G_{neu}) \cap G_{pos} \neq G_{neg} \cap (G_{neu} \cap G_{pos})$ καθώς τα βάρη των ακμών είναι διαφορετικά. Επομένως, η διαδικασία υπολογισμού του κοινού υπογράφου πρέπει να εφαρμοσθεί επαναληπτικά μέχρι ότου οδηγηθούμε σε σύγκλιση : $|G_{subgr}| \to 0$. Πολλές φορές ο κοινός υπογράφος είναι ιδιαίτερα μεγάλου βαθμού με αποτέλεσμα ο ακριβής υπολογισμός του να αποτελεί μία ιδιαίτερα απαιτητική διαδικασία από πλευράς χρόνου και χωρητικότητας οπότε συνήθως χρησιμοποιείται ένα προκαθορισμένος αριθμός επαναλήψεων.

Σε επίπεδο υλοποίησης η συγκεκριμένη λειτουργικότητα αναπτύσσεται χρησιμοποιώντας τις μεθόδους `allNotIn` (τελεστής Δέλτα $\bigtriangledown$) `intersectGraph` (τελεστής Τομής $\cap$) της κλάσης `DocumentNGramGraph` της βιβλιοθήκης `JInsect` όπως παρουσιάζεται στο τμήμα κώδικα που ακολουθεί :

```
/* Calculates the common subgraph and removes it from all class ngrams graphs
 * @param The ngram graphs of all classes (order : negative,neutral,positive)
 * @return A DocumentNGramGraph array containing the noise-free versions of the
 * specified ngram graphs
 * NOTE : Iterations LIMIT must be pre-defined
 */
public DocumentNGramGraph [] removeNoiseFromGraphs(DocumentNGramGraph [] graphs){

 DocumentNGramGraph [] noiseFreeGraphs = new DocumentNGramGraph[3];
 DocumentNGramGraph tempGraph ;
 DocumentNGramGraph commonSubgraph;

 noiseFreeGraphs[0] = graphs[0].clone(); // negative class
 noiseFreeGraphs[1] = graphs[1].clone(); // neutral  class
 noiseFreeGraphs[2] = graphs[2].clone(); // positive class

 int i = 0;
 do{
  // intersection of negative and neutral class graphs
  tempGraph = noiseFreeGraphs[0].intersectGraph(noiseFreeGraphs[1]);
  // intersection of tempGraph and positive class graphs
  commonSubgraph = noiseFreeGraphs[2].intersectGraph(tempGraph);
  //remove calculated commonSubgraph from class graphs
  noiseFreeGraphs[0] = noiseFreeGraphs[0].allNotIn(commonSubgraph);
  noiseFreeGraphs[1] = noiseFreeGraphs[1].allNotIn(commonSubgraph);
  noiseFreeGraphs[2] = noiseFreeGraphs[2].allNotIn(commonSubgraph);
  i++;
  // iterate until convergance condition is satisfied or limit has been reached
 }while (!commonSubgraph.isEmpty() && (i < LIMIT));

 return noiseFreeGraphs;
}
```

Κώδικας 4.1: Υλοποίηση της Αφαίρεσης Θορύβου από τους Γράφους Κλάσεων

Στον Πίνακα παρατίθενται τα ποσοστά ακρίβειας των ταξινομητών για μέγεθος ν-γράμματος $n = 4$ όπως προέκυψαν μετά από την αφαίρεση του θορύβου από τους γράφους για αριθμό επαναλήψεων $i = 10, 30, 100, 200, 1000, 2000$.

Παρατηρούμε πως καθώς αυξάνεται ο αριθμός των επαναλήψεων και αφαιρούνται περισσότερες ακμές του κοινού υπογράφου, βελτιώνεται σταθερά η ακρίβεια των ταξινομητών Naive Bayes και Multinomial Naive Bayes ενώ για τους υπόλοιπους δεν μπορεί να εξαχθεί ασφαλές συμπέρασμα.

4.4.5.4 Διακριτοποίηση Δεικτών Ομοιότητας

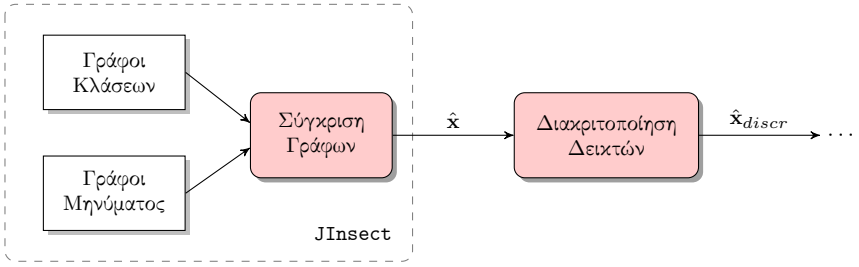
Η διαδικασία διακριτοποίησης προτείνεται από τους Aisopos et al. στο [2] με κύριο στόχο την αύξηση της αποδοτικότητας του συστήματος και αναπαρίσταται στο Σχήμα 4.10.

Στηρίζεται σε συγκρίσεις ανά ζεύγη μεταξύ των τιμών της ίδιας μετρικής (CS, VS, NVS) που αντιστοιχούν στις διάφορες κατηγορίες πολικότητας και υπολογίζει τη διακριτή τιμή τους μέσω της ακόλουθης συνάρτησης :

$$dsim(sim_{pol_1}, sim_{pol_2}) = \begin{cases} pol_1, & \text{αν} \quad sim_{pol_1} < sim_{pol_2} \\ equal, & \text{αν} \quad sim_{pol_1} = sim_{pol_2} \\ pol_2, & \text{αν} \quad sim_{pol_1} > sim_{pol_2} \end{cases}$$

όπου sim ο δείκτης ομοιότητας $sim \in \{CS, VS, NVS\}$ και pol_1, pol_2 οι συγκεκριμένες κατηγορίες του εκάστοτε δείκτη $pol_1, pol_2 \in \{neg, neu, pos\}$.

Να σημειωθεί ότι η αρχική προσέγγιση μετατρέπει τους δείκτες από αριθμητικές τιμές σε αλφαριθμητικά (nominals) που υποδηλώνουν την κατηγορία ή ισότητα. Λόγω της αδυναμίας ορισμένων ταξινομητών να διαχειριστούν μη αριθμητικές τιμές χαρακτηριστικών, χρησιμοποιήσαμε την εξής αντιστοίχηση : $neg \rightarrow 1$, $pos \rightarrow 2$, $neu \rightarrow 3$ και $equal \rightarrow 4$.



Σχήμα 4.10: Τροποποίηση Διακριτοποίησης Δεικτών Ομοιότητας

Κατά τη διεξαγωγή των πειραμάτων παρατηρήθηκε αισθητή μείωση του απαιτούμενου χρόνου εκπαίδευσης στους περισσότερους ταξινομητές, επιβεβαιώνοντας την αύξηση της αποδοτικότητας. Παράλληλα,όπως προκύπτει από τον Πίνακα 4.8, ενώ η συγκεκριμένη διαφοροποίηση του συστήματος δεν βοηθά τους ταξινομητές με την καλύτερη επίδοση (MLP, SVM και Logistic), συντελεί καθοριστικά στη βελτίωση της ακρίβειας των ταξινομητών που εμφανίζουν μέτρια και χαμηλά ποσοστά στην αρχική αρχιτεκτονική. Ιδιαίτερα θεαματική είναι η αύξηση της ακρίβειας των MNB και k-NN οι οποίοι επιτυγχάνουν 61.75% και 65.43% αντίστοιχα και είναι πλέον συγκρίσιμοι με τους υπόλοιπους. Το φαινόμενο αυτό αποδεικνύει από διαφορετική οπτική γωνία αυτή τη φορά ότι για συγκεκριμένους αλγόριθμους μάθησης δεν έχει σημασία τόσο η αριθμητική τιμή των δεικτών όσο η διάταξη και οι σχετικές διαφορές μεταξύ των δεικτών της ίδιας κατηγορίας οι οποίες τονίζονται μέσω της διακριτοποίησης και βελτιώνουν έτσι τη διακριτική ικανότητα των ταξινομητών.

4.4.5.5 Συνδυασμός Ταξινομητών

Ο συνδυασμός ταξινομητών είναι μία από πιο διαδεδομένες μεθόδους για την αναγνώριση προτύπων η οποία εφαρμόζεται ευρύτατα τα τελευταία χρόνια. Η τεχνική αυτή - γνωστή με τον όρο *ensemble* - οδηγεί συχνά σε μικρότερα ποσοστά σφάλματος κατηγοριοποίησης συγκριτικά με μεμονωμένους ταξινομητές. Όπως αναλύεται στο [25], αυτό οφείλεται στο γεγονός ότι ο συνδυασμός διαφορετικών τεχνικών με στόχο την λήψη μίας τελικής απόφασης για την κατηγοριοποίηση των δεδομένων έχει ως αποτέλεσμα το σύστημα να

NGram	MLP		SVM		Logistic		kNN		NaiveBayes		MNB		C4.5	
1	42.48	(-0.60)	41.04	(+1.34)	44.40	(-0.52)	40.63	(+0.38)	40.21	(+1.65)	36.97	(+0.00)	43.58	(+0.05)
2	57.07	(-0.71)	56.76	(+0.49)	58.71	(-0.48)	48.02	(-0.45)	48.34	(+3.59)	36.97	(+0.00)	53.24	(-1.93)
3	64.22	(-0.01)	64.45	(-0.02)	64.33	(-0.27)	54.18	(-1.00)	57.15	(+5.43)	37.86	(+0.88)	61.06	(-1.14)
4	65.73	(+0.25)	65.54	(-0.08)	65.63	(+0.02)	55.84	(-1.41)	60.27	(+4.37)	45.28	(+5.13)	63.57	(-0.31)
5	65.00	(-0.14)	65.05	(-0.03)	65.27	(-0.03)	55.35	(-1.50)	60.17	(+2.65)	53.75	(+10.94)	63.55	(-0.03)
6	63.85	(-0.14)	63.74	(-0.01)	64.32	(+0.05)	54.82	(-1.08)	57.24	(+1.88)	56.47	(+11.53)	62.62	(-0.20)
7	62.45	(-0.04)	62.30	(+0.15)	62.89	(-0.02)	52.68	(-1.95)	53.40	(+2.17)	56.29	(+9.97)	61.40	(-0.81)

Πίνακας 4.7: Ποσοστά Ακρίβειας μετά τη χρήση των Δεδομένων Κατασκευής στη διαδικασία της Εκπαίδευσης

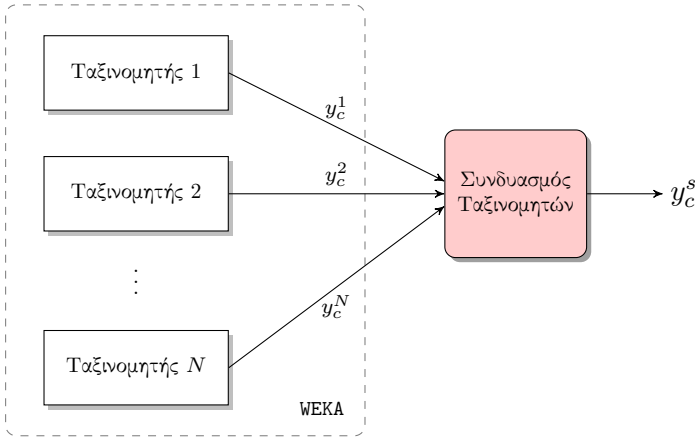
NGram	MLP		SVM		Logistic		kNN		NaiveBayes		MNB		C4.5	
1	41.35	(-1.73)	38.40	(-1.30)	40.54	(-4.38)	42.13	(+1.88)	40.23	(+1.67)	36.97	(+0.00)	42.26	(-1.26)
2	56.77	(-1.00)	55.12	(-1.16)	54.44	(-4.75)	57.81	(+9.33)	54.51	(+9.76)	50.50	(+13.52)	57.94	(+2.77)
3	63.95	(-0.28)	64.03	(-0.44)	63.90	(-0.71)	64.05	(+8.86)	56.46	(+4.75)	61.27	(+24.30)	64.16	(+1.96)
4	65.42	(-0.06)	65.44	(-0.18)	65.28	(-0.33)	65.43	(+8.17)	56.74	(+0.84)	61.75	(+21.59)	65.44	(+1.56)
5	64.84	(-0.31)	65.15	(+0.06)	64.87	(-0.44)	65.03	(+8.18)	56.65	(-0.87)	61.11	(+18.29)	65.15	(+1.57)
6	63.94	(-0.06)	64.35	(+0.60)	63.91	(-0.36)	64.27	(+8.37)	57.09	(+1.74)	59.63	(+14.69)	64.45	(+1.63)
7	63.03	(+0.53)	63.22	(+1.07)	62.39	(-0.52)	63.20	(+8.58)	56.84	(+5.61)	57.43	(+11.11)	63.28	(+1.08)

Πίνακας 4.8: Ποσοστά Ακρίβειας μετά τη Διακριτοποίηση Δεικτών Ομοιότητας

Επαναλήψεις	MLP		SVM		Logistic		kNN		NaiveBayes		MNB		C4.5	
50	65.77	(+0.30)	65.63	(+0.01)	65.63	(+0.02)	57.36	(+0.11)	56.23	(+0.33)	40.23	(+0.08)	64.04	(+0.16)
100	65.77	(+0.30)	65.59	(-0.03)	65.71	(+0.10)	57.38	(+0.13)	56.97	(+1.07)	40.44	(+0.29)	63.87	(-0.01)
200	65.66	(+0.18)	65.60	(-0.02)	65.69	(+0.08)	59.88	(+2.63)	57.95	(+2.05)	40.69	(+0.54)	63.92	(+0.03)
1000	65.20	(-0.28)	65.73	(+0.11)	65.74	(+0.13)	59.53	(+2.28)	63.50	(+7.60)	43.47	(+3.31)	64.55	(+0.66)
2000	64.73	(-0.75)	65.67	(+0.06)	65.77	(+0.16)	57.08	(-0.17)	64.64	(+8.74)	46.89	(+6.74)	64.77	(+0.89)

Πίνακας 4.9: Ποσοστά Ακρίβειας μετά τη Αφαίρεση Θορύβου από τους Γράφους Κλάσεων συναρτήσει του αριθμού των επαναλήψεων

εμφανίζει ευσταθή συμπεριφορά , αντιμετωπίζοντας δυσκολίες που ένας ταξινομητής - ατομικά - ενδεχομένως συναντά σε ένα συγκεκριμένο σύνολο δεδομένων. Έχουν προταθεί διάφορες μέθοδοι για το συνδυασμό των ταξινομητών εξετάζοντας διαφορετικούς τρόπους εκπαίδευσης, σχήματα συνδυασμού και αλγορίθμους κατηγοριοποίησης. Στη συνέχεια, περιγράφονται συνοπτικά και αξιολογούνται ως προς την επίδοση δύο διαφορετικά σχήματα κατηγοριοποίησης στα οποία συνδυάζονται οι βασικοί ταξινομητές που εξετάστηκαν στην Ενότητα 3.2, έχοντας εκπαιδευτεί με το ίδιο σύνολο χαρακτηριστικών από το ίδιο σύνολο εκπαίδευσης.



Σχήμα 4.11: Τροποποίηση Συνδυασμού Ταξινομητών

4.4.5.5.1 Σχήμα Ψηφοφορίας Αποτελεί την πιο απλή μέθοδο συνδυασμού ταξινομητών και είναι από τις πρώτες που χρησιμοποιήθηκαν. Στηρίζεται στη διαδικασία της ψηφοφορίας κατά την οποία κάθε ταξινομητής αποφασίζει μεμονωμένα και "ψηφίζει" την κλάση που θεωρεί πιο πιθανή. Επομένως, η τελική κλάση που ανήκει το δεδομένο είναι αυτή η οποία συγκέντρωσε τις περισσότερες ψήφους.

Έστωσαν το πλήθος των προτύπων του συνόλου εκπαίδευσης $\mathcal{D} = \{\mathbf{x}\}_{i=1}^{N}$, το πλήθος των κλάσεων $\mathcal{C} = \{c_j\}_{j=1}^{K}$ και το πλήθος των ταξινομητών C_k που συμμετέχουν στη ψηφοφορία. Εξετάζουμε τα εξής σχήματα ψηφοφορίας [25] :

(i) Πλειοψηφία (Plurality Vote - PV) : το στιγμιότυπο εισόδου αντιστοιχίζεται στη κλάση c_j η οποία συγκέντρωσε τις περισσότερες ψήφους. Στη συγκεκριμένη προσέγγιση όλοι οι ταξινομητές θεωρούνται ισότιμοι ανεξάρτητα από την ακρίβεια κατηγοριοποίησης. Κάθε ταξινομητής έχει συντελεστή βαρύτητας :

$$w_m = \frac{1}{M} \quad \forall m \in [1, M]$$

(ii) Απλή Σταθμισμένη Ψήφος (Simple Weighted Vote - SWV) : η απόφαση κάθε ταξινομητή σταθμίζεται ανάλογα με την εκτιμώμενη ακρίβεια a_m στο σύνολο εκπαίδευσης. Ο συντελεστής βαρύτητας (βάρος) της ψήφου είναι :

$$w_m = \frac{a_m}{\sum_l a_l}$$

Σε όλα τα υπόλοιπα σχήματα ψηφοφορίας που παρουσιάζονται - εκτός του τελευταίου - τα βάρη w_m αναφέρονται στις κανονικοποιημένες τιμές των a_k όπως προκύπτουν από την παραπάνω εξίσωση.

(iii) Αναβαθμονομημένη Σταθμισμένη Ψήφος (Rescaled Weighted Vote - RWV) : ανατίθενται μηδενικά βάρη σε όσους ταξινομητές έχουν $^N/_K$ ή λιγότερες σωστές "προβλέψεις" στο σύνολο εκπαίδευσης και οι τιμές των βαρών αναπροσαρμόζονται αναλογικά. Συνεπώς, ταξινομητές με εκτιμώμενη ακρίβεια $a_m \leq {}^1/_K$ αποκλείονται από τη ψηφοφορία. Επομένως, αν e_m τα λάθη του ταξινομητή C_m τότε :

$$a_m = max\{0, 1 - \frac{K \cdot e_k}{N \cdot (K-1)}\}$$

(iv) Σταθμισμένη Ψήφος Καλύτερου-Χειρότερου (Best-Worst Weighted Vote - BWWMV) : σε αυτή τη προσέγγιση ο καλύτερος και χειρότερος ταξινομητής προσδιορίζονται βάσει της εκτιμώμενης ακρίβειας λαμβάνοντας $a_m = 1$ ο πρώτος και $a_m = 0$ ο δεύτερος (αποκλεισμός). Οι υπόλοιποι ταξινομητές αξιολογούνται γραμμικά οπότε:

$$a_m = 1 - \frac{e_m - e_B}{e_W - e_B}$$

όπου

$$e_B = \min_k\{e_m\} \quad \text{και} \quad e_W = \max_k\{e_m\}$$

Από τα αποτελέσματα του Πίνακα 4.10, διαπιστώνουμε ότι τα πιο απλά σχήματα ψηφοφορίας (Πλειοψηφία και Απλή Σταθμισμένη Ψήφος) επιτυγχάνουν καλύτερα ποσοστά ακρίβειας σε σχέση με τις πιο πολύπλοκα σχήματα (Αναβαθμονομημένη Σταθμισμένη και Σταθμισμένη Ψήφος Καλύτερου-Χειρότερου) για όλες τις τιμές ν-γράμματος που εξετάσαμε. Αυτό συμβαίνει πιθανότατα εξαιτίας του τρόπου με τον οποίο ανατίθενται τα βάρη ψηφοφορίας. Ταξινομητές με χαμηλό ποσοστό σφάλματος στα δεδομένα μάθησης αποκτούν ιδιαίτερη βαρύτητα στην ψηφοφορία - χωρίς ωστόσο να εμφανίζουν την ίδια συμπεριφορά και στα δεδομένα εξέτασης με αποτέλεσμα να επηρεάζονται τα τελικά ποσοστά ακρίβειας του συνδυασμού.

NGram	**PV**		**SWV**		**RWV**		**BWWV**	
1	43.89	(-1.03)	43.94	(-0.99)	40.25	(-4.68)	42.93	(-1.99)
2	57.80	(-1.38)	58.29	(-0.89)	57.44	(-1.75)	56.56	(-2.63)
3	64.59	(-0.02)	**64.67**	(0.06)	64.50	(-0.11)	64.17	(-0.43)
4	**65.83**	(0.21)	**65.84**	(0.22)	**65.75**	(0.13)	65.40	(-0.22)
5	65.23	(-0.08)	65.26	(-0.05)	65.16	(-0.14)	65.02	(-0.29)
6	64.13	(-0.14)	64.02	(-0.25)	64.04	(-0.24)	63.84	(-0.44)
7	62.80	(-0.11)	62.77	(-0.14)	62.69	(-0.22)	62.20	(-0.71)

Πίνακας 4.10: Ποσοστά Ακρίβειας με τη συμμετοχή όλων των ταξινομητών στο σχήμα Ψηφοφορίας

Από την παραπάνω ανάλυση εξάγεται το συμπέρασμα ότι η συμπεριφορά του σχήματος ψηφοφορίας εξαρτάται από την κατάλληλη επιλογή των ταξινομητών που συμμετέχουν σε αυτό. Εξετάζουμε, λοιπόν, όλους τους πιθανούς συνδυασμούς ταξινομητών σε σχήματα από 2 έως και 6 ψηφοφόρους. Βέλτιστη επιλογή αποτελεί η χρήση 3 ταξινομητών σε

όλες τις περιπτώσεις ν-γράμματος. Στον Πινακα 4.11 παρουσιάζονται οι συνδυασμοί με τρεις ταξινομητές οι οποίοι σημείωσαν την υψηλότερη επίδοση για μέγεθος ν-γράμματος $n = 4$.

			PV	**SWV**	**RWV**	**BWWV**
MLP	SVM	Logistic	65.88	65.88	65.88	65.87
MLP	SVM	kNN	65.88	65.76	57.25	65.91
MLP	Logistic	kNN	65.93	65.77	57.25	65.97
MLP	Logistic	C4.5	65.86	65.83	65.83	65.87

Πίνακας 4.11: Συνδυασμοί Ταξινομητών με βέλτιστη επίδοση στο σχήμα Ψηφοφορίας

Παρατηρούμε ότι οι ταξινομητές MLP, SVM και Logistic οι οποίοι εμφανίζουν υψηλά ποσοστά ακρίβειας, δρουν αποτελεσματικά και σε συνδυασμό σε όλα τα σχήματα ψηφοφορίας. Αυτό επιβεβαιώνει τα κοινά χαρακτηριστικά που διαθέτουν και υποδηλώνει ότι στο συγκεκριμένο σύνολο που εξετάζεται, υπάρχουν δεδομένα τα οποία κανείς τους δεν μπορεί να αναγνωρίσει σωστά. Επιπλέον, τη δράση των παραπάνω ενισχύουν και ταξινομήτες με μέτρια επίδοση όπως οι k-NN και C4.5 αποδεικνύοντας ότι στις τεχνικές ensemble εκτός από την ακρίβεια, καθοριστικής σημασίας είναι και η συμπληρωματικότητα των ταξινομητών.

Τέλος, τα απλά σχήματα ψηφοφορίας εμφανίζουν ευσταθή συμπεριφορά ανεξάρτητα από την επιλογή των ταξινομητών σε αντίθεση με τα πιο πολύπλοκα τα οποία αποδίδουν μόνο σε συγκεκριμένους συνδυασμούς. Χαρακτηριστικό παράδειγμα είναι η Αναβαθμονομημένη Σταθμισμένη Ψήφος και ο ταξινομητής k-NN ο οποίος λόγω της οκνηρής μάθησης εμφανίζει μηδενικό σφάλμα εκπαίδευσης ενώ έχει μέτρια επίδοση στα άγνωστα δεδομένα με συνέπεια να διαμορφώνει το τελικό αποτέλεσμα και να ευθύνεται για τα χαμηλά ποσοστά του σχήματος ψηφοφορίας.

4.4.5.5.2 Σχήμα Συνένωσης με Τριγωνικές Νόρμες Μία άλλη μέθοδος συνδυασμού είναι η συνένωση ταξινομητών (*fusion*). Οι περισσότερες τεχνικές συνένωσης προϋποθέτουν την ανεξαρτησία μεταξύ των ταξινομητών, απαίτηση που δεν πληρούται τις περισσότερες φορές στην πραγματικότητα καθώς οι ταξινομήτες έχουν την τάση να εμφανίζουν τα ίδια λάθη στις πιο δύσκολες περιπτώσεις κατηγοριοποίησης προτύπων. Στα πλαίσια της Μηχανικής Μάθησης, δύο ταξινομήτες είναι *θετικά συσχετισμένοι* όταν κατηγοριοποιούν λανθασμένα στην ίδια κατηγορία ενώ *αρνητικά συσχετισμένοι* όταν κατηγοριοποιούν λανθασμένα σε διαφορετικές κατηγορίες.

Ακολουθούμε την προσέγγιση που περιγράφεται στο [8] και η οποία ενδείκνυται για περιπτώσεις όπου οι ταξινομητές μπορεί να εμφανίζουν οποιαδήποτε μορφής συσχέτιση (θετική ή αρνητική) μεταξύ τους. Στηρίζεται στις Τριγωνικές Νόρμες - μία γενίκευση του λογικού τελεστή της τομής στο χώρο των πολλών μεταβλητών - και προσδιορίζει την τελική κατηγορία μέσω της τομής των επιμέρους αποφάσεων. Η πιθανή συσχέτιση μπορεί να αναπαρασταθεί μέσω κατάλληλης Τριγωνικής Νόρμας - T-$norm$ έτσι ώστε να αποφευχθούν περιπτώσεις υπο ή υπερ-εκτιμήσεων.

Ορισμός 4.4.1. Η Τριγωνική Νόρμα - T-$norm$ είναι μία συνάρτηση $T : [0,1] \times [0,1] \to [0,1]$ με τις εξής ιδιότητες :

- Αντιμεταθετικότητα : $T(a,b) = T(b,a)$
- Μονοτονία : $T(a,b) \leq T(c,d)$ αν $a \leq c$ και $b \leq d$
- Προσεταιριστικότητα : $T(a,T(b,c)) = T(T(a,b),c)$
- Ουδέτερο Στοιχείο : $T(a,1) = a$

T-$norm$	Τύπος Συσχέτισης
$T_1(x,y) = max\{0, x+y-1\}$	Ισχυρά Αρνητική
$T_2(x,y) = [max\{0, \sqrt{x}+\sqrt{y}\}]^2$	Μερικώς Αρνητική
$T_3(x,y) = x \cdot y$	Ουδέτερη
$T_4(x,y) = [\frac{1}{\sqrt{x}} + \frac{1}{\sqrt{y}} - 1]^{-1/2}$	Ελαφρώς Θετική
$T_5(x,y) = [\frac{1}{x} + \frac{1}{y} - 1]^{-1}$	Μερικώς Αρνητική
$T_6(x,y) = min\{x,y\}$	Ισχυρά Θετική

Πίνακας 4.12: Οι Τριγωνικές Νόρμες και ο αντίστοιχος Τύπος Συσχέτισης

Έστωσαν m ταξινομητές $S_1, S_2, \cdots, S_m$. Η έξοδος κάθε ταξινομητή S_j περιγράφεται μέσω του διανύσματος $\mathbf{I}^j \in \mathbb{R}^{N+1}$ αναπαριστώντας την κανονικοποιημένη απόφασή για κάθε μία από τις N κλάσεις. Το τελευταίο στοιχείο $I^j(N+1)$ αναπαριστά το σύνολο των κλάσεων U και αναφέρεται στην περίπτωση όπου ο ταξινομητής δεν καταλήγει σε απόφαση (δε παρατηρείται στο πρόβλημα που εξετάζουμε) :

$$\mathbf{I}^j = [I^j(1), I^j(2), \cdots, I^j(N+1)] \quad \text{όπου} \quad I^j(i) \in [0,1] \quad \text{και} \quad \sum_{i=1}^{N+1} I^j(i) = 1$$

Η μη-κανονικοποιημένη συνένωση των εξόδων δύο ταξινομητών S_1 και S_2 ορίζεται ως :

$$\mathcal{F}(\mathbf{I^1}, \mathbf{I^2}) = Extraction[Outerproduct(\mathbf{I^1}, \mathbf{I^2}, T)] = Extraction[A]$$

όπου το εξωτερικό γινόμενο $Outerproduct$ είναι μία καλώς ορισμένη μαθηματική πράξη η οποία δέχεται ως ορίσματα δύο διανύσματα διαστάσεων N και παράγει ως έξοδο ένα πίνακα A διαστάσεων $N \times N$. Κάθε στοιχείο του πίνακα $A(i,j)$ προκύπτει από την εφαρμογή του τελεστή T στα αντίστοιχα στοιχεία των δύο διανυσμάτων , $\mathbf{I^1}(i), \mathbf{I^2}(j)$ δηλαδή :

$$A(i,j) = [\mathbf{I^1}(i), \mathbf{I^2}(j)]$$

Ο τελεστής εξαγωγής $Extraction$ αξιοποιεί το γεγονός ότι οι κλάσεις είναι ξένες μεταξύ τους και ανακτά τη μη - κανονικοποιημένη έξοδο σε μορφή διανύσματος :

$$Extraction[A] = [I'(1), I'(2), \cdots, I'(N+1)]$$

όπου :

$$\begin{aligned} I'(i) &= A(i,i) + A(i,N+1) + A(N+1,i) \quad ,i = 1,2,\cdots,N \\ I'(N+1) &= A(N+1,N+1) \end{aligned}$$

Από την αντιμεταθετικότητα των $T\text{-}norms$ προκύπτει και η αντιμεταθετικότητα της συνένωσης:

$$\mathcal{F}(\mathbf{I^1}, \mathcal{F}(\mathbf{I^2}, \mathbf{I^3})) = \mathcal{F}(\mathcal{F}(\mathbf{I^1}, \mathbf{I^2}), \mathbf{I^3}))$$

με την προϋπόθεση ότι $A(i, N+1) = A(N+1, i) = 0 \quad ,i = 1,2,\cdots,N$

Θεωρώντας ότι οι κλάσεις είναι διατεταγμένες ($C_{neg} < C_{neu} < C_{pos}$) μπορούμε να υπολογίσουμε το βαθμό εμπιστοσύνης σε μία συνένωση ορίζοντας ένα μέτρο διασποράς γύρω από την κύρια διαγώνιο του πίνακα σφάλματος-σύγχυσης (confusion matrix). Όσο περισσότερα βάρη ανατίθενται σε στοιχεία εκτός της κυρίας διαγωνίου, τόσο λιγότερος είναι ο βαθμός ομοφωνίας μεταξύ των ταξινομητών. Αυτή η έννοια μπορεί να μοντελοποιηθεί μέσω ενός πίνακα ποινής της μορφής :

$$\mathcal{P}(i,j) = \begin{cases} max\{0, (1 - W \cdot |i-j|)^d\}, & \text{αν} \quad 1 \leq i \geq N \quad \text{και} \quad 1 \leq j \geq N \\ 1, & \text{αν} \quad i = N+1 \quad \text{ή} \quad j = N+1 \end{cases}$$

Η παρουσία στοιχείων στην κύρια διαγώνιο υποδηλώνει συμφωνία μεταξύ των ταξινομητών ενώ η παρουσία στοιχείων εκτός της κυρίας διαγωνίου υποδηλώνει διαφωνία. Το μέτρο της διαφωνίας αυξάνεται αναλογικά με την απόσταση από τη κύρια διαγώνιο. Με τον τρόπο αυτό αποτυπώνεται το γεγονός ότι η απόκλιση ανάμεσα στις κλάσεις C_{neg} και C_{neu} έχει μικρότερη ποινή σε σχέση με την απόκλιση ανάμεσα στις κλάσεις C_{neg} και C_{pos}.

Το μέτρο του κανονικοποιημένου βαθμού εμπιστοσύνης $\widehat{\mathcal{C}}$ υπολογίζεται από τη σχέση :

$$\widehat{\mathcal{C}} = NormalizedConfidence(\widehat{A}, \mathcal{P}) = \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} \widehat{A}(i,j) \cdot \mathcal{P}(i,j)$$

όπου $\widehat{A}$ ο κανονικοποιημένος πίνακας συνένωσης.

Από τα αποτελέσματα του Πίνακα 4.13 προκύπτει ότι οι Τριγωνικές Νόρμες που υποθέτουν καμία ή ελαφρώς θετική συσχέτιση μεταξύ των ταξινομητών εμφανίζουν αισθητά καλύτερα ποσοστά ακρίβειας. Το γεγονός αυτό υποδηλώνει - όπως στην περίπτωση του σχήματος ψηφοφορίας αλλά από διαφορετική οπτική γωνία αυτή τη φορά - ότι υπάρχουν tweets στα οποία όλοι οι ταξινομητές αποφασίζουν λανθασμένα την κατηγορία πολικότητας η οποία όπως αποδεικνύεται είναι συχνά η ίδια (θετική συσχέτιση). Εντούτοις, η καλύτερη επίδοση του σχήματος συνένωσης (T_4 64.26%) δεν ξεπερνά αλλά υπολείπεται της αντίστοιχης βέλτιστης στην αρχική αρχιτεκτονική. Συνεπώς, και σε αυτή τη μέθοδο συνδυασμού η χρήση όλων των ταξινομητών δεν αποφέρει αποτέλεσμα.

Σε επόμενο βήμα, εκμεταλλευόμενοι την αντιμεταθετική ιδιότητα της Τριγωνικής Νόρμας, εξετάσαμε τη συνένωση των ταξινομητών σε συνδυασμούς από 2 έως 6. Διαπιστώσαμε ότι χρησιμοποιώντας μικρό πλήθος ταξινομητών (2 ή 3) και μέγεθος ν-γράμματος

NGram	T_1	T_2	T_3	T_4	T_5	T_6
1	32.78	32.83	42.19	43.42	42.18	41.22
2	32.78	32.83	52.25	55.70	51.87	50.80
3	32.78	40.84	59.20	62.65	58.78	58.01
4	32.78	44.20	61.29	64.26	60.70	59.81
5	33.20	45.25	60.72	63.84	60.29	59.60
6	33.54	45.16	60.20	63.10	59.68	58.77
7	35.25	44.64	59.10	62.50	58.65	57.67

Πίνακας 4.13: Ποσοστά Ακρίβειας κατά τη συνένωση όλων των ταξινομητών

			T_1	T_2	T_3	T_4	T_5	T_6
MLP	Logistic		63.22	**65.98**	**65.98**	**65.99**	**65.98**	**65.98**
MLP	MNB		56.51	**65.96**	**65.93**	**65.90**	**65.92**	65.22
MLP	Logistic	SVM	54.38	**65.65**	**65.63**	**65.82**	**65.64**	**65.68**
MLP	Logistic	MNB	46.39	62.87	**65.98**	64.12	**65.99**	65.26

Πίνακας 4.14: Συνδυασμοί Ταξινομητών με βέλτιστη ακρίβεια στην περίπτωση ν-γράμματος $n = 4$

$n = 4$ το σχήμα συνένωσης εμφανίζει σταθερά ικανοποιητική συμπεριφορά - ανεξάρτητα από τη νόρμα που εφαρμόζεται. Στον Πίνακα 4.14 παρατίθενται οι συνδυασμοί με τα καλύτερα ποσοστά ακρίβειας :

Τα βελτιωμένα ποσοστά ακρίβειας κατά τη συνένωση των MLP, SVM και Logistic με όλους τους πιθανούς συνδυασμούς είναι ως ένα βαθμό αναμενόμενα καθώς η μεταξύ τους συμβατότητα είχε διαφανεί και στο σχήμα ψηφοφορίας. Αντίθετα, ιδιαίτερη εντύπωση προκαλεί η συμμετοχή του αισθητά χειρότερου MNB σε σχήμα συνένωσης με αρκετά καλά αποτελέσματα. Κρίνεται σκόπιμο η συγκεκριμένη συνένωση να επαναληφθεί με διαφορετικά δεδομένα για να διαπιστωθεί αν η θετική συμβολή του MNB είναι συστηματική.

Σε σχέση με άλλες μεθόδους συνδυασμού, το σχήμα συνένωσης μάς παρέχει μία επιπλέον πληροφορία : το βαθμό ομοφωνίας μεταξύ των συμμετεχόντων ταξινομητών. Στο σημείο αυτό πρέπει να διευκρινισθεί ότι οι παράμετροι βάρους W και απόστασης d του πίνακα ποινής $\mathcal{P}$ δεν επηρεάζουν τη διαδικασία κατηγοριοποίησης και την απόκριση του συστήματος αλλά διαμορφώνουν τα επίπεδα στα οποία κυμαίνεται οι δείκτης εμπιστοσύνης C για κάθε tweet. Δηλαδή, για μία δεδομένη νόρμα το σύστημα εμφανίζει τα ίδια ποσοστά ακρίβειας, ταξινομώντας κάθε tweet στην ίδια κατηγορία ανεξάρτητα από τις παραμέτρους W και d αλλά ανάλογα με τις τιμές τους θα είναι περισσότερο ή λιγότερο επιφυλακτικό σε κάθε απόφασή του.

Επομένως, μέσω του δείκτη εμπιστοσύνης μπορούμε να αναγνωρίσουμε και να απομονώσουμε tweets στα οποία δεν παρατηρείται ικανοποιητικός βαθμός ομοφωνίας και να αποφανθούμε για όσα επικρατούν ευνοϊκές συνθήκες. Το σύστημα αποκρίνεται σε ένα μέρος του συνόλου των δεδομένων αλλά με αρκετά υψηλότερα ποσοστά επιτυχίας.

Στο πλαίσιο αυτό, η αποτελεσματικότητα της συνένωσης έγκειται στον προσδιορισμό των παραμέτρων W και d για τους οποίους προκύπτει αντιπροσωπευτική κατανομή του

Κατώφλι $\widehat{\mathcal{C}}$	T_1		T_2		T_3		T_4		T_5		T_6	
≥ 0.55	71.88	(76.66)	66.46	(97.55)	67.40	(92.18)	68.29	(78.84)	67.32	(92.70)	66.86	(95.69)
≥ 0.60	72.00	(76.16)	68.22	(87.53)	71.83	(71.42)	87.92	(7.68)	73.48	(64.93)	71.13	(76.05)
≥ 0.65	71.13	(76.05)	69.56	(79.72)	76.79	(51.63)	96.99	(1.09)	79.46	(43.80)	76.60	(53.33)
≥ 0.70	72.30	(74.76)	72.20	(67.20)	83.13	(34.85)	100.0	(0.01)	85.61	(28.87)	83.44	(34.24)
≥ 0.75	72.49	(73.66)	79.48	(44.44)	87.55	(24.38)	-	-	89.42	(20.21)	88.31	(22.76)
≥ 0.80	72.71	(72.13)	84.84	(30.08)	90.32	(17.22)	-	-	91.43	(14.40)	91.05	(15.51)
≥ 0.85	72.89	(70.08)	89.55	(18.87)	92.14	(11.96)	-	-	92.45	(10.08)	92.39	(10.44)
≥ 0.90	73.00	(67.18)	92.60	(10.61)	93.63	(7.13)	-	-	93.67	(5.99)	93.45	(5.87)
≥ 0.95	72.29	(60.06)	93.90	(2.69)	94.95	(2.27)	-	-	96.64	(1.81)	96.80	(1.74)

Πίνακας 4.15: Διαστρωμάτωση Δείκτη Εμπιστοσύνης $\widehat{\mathcal{C}}$ ως προς την Τριγωνική Νόρμα Συνένωσης

δείκτη εμπιστοσύνης και οδηγούμαστε σε βέλτιστη αντιστάθμιση ανάμεσα στην ακρίβεια κατηγοριοποίησης και το ποσοστό του πληθυσμού στο οποίο επικεντρωνόμαστε. Ύστερα από ενδελεχή αναζήτηση προέκυψε ότι η εν λόγω ισορροπία επιτυγχάνεται για $W = 0.25$ και $d = 3$. Στον Πίνακα 4.15 παρουσιάζονται τα αποτελέσματα κατά τη συνένωση των MLP και Logistic στην περίπτωση ν-γράμματος $n = 4$ ως προς το επίπεδο του δείκτη εμπιστοσύνης. Για κάθε νόρμα αναγράφεται το ποσοστό ακρίβειας και σε παρένθεση το αντίστοιχο ποσοστό επί του συνόλου των δεδομένων στο οποίο απαντάται βαθμός εμπιστοσύνης μεγαλύτερος $\widehat{\mathcal{C}}$ ή ίσος του καθορισμένου κατωφλίου.

Παρατηρούμε ότι κάθε Τριγωνική νόρμα οδηγεί σε διαφορετική διαστρωμάτωση του βαθμού ομοφωνίας με αποτέλεσμα για ένα δεδομένο δείκτη εμπιστοσύνης να εντοπίζονται διαφορές στο πλήθος των εξεταζόμενων tweets και στην ακρίβεια κατηγοριοποίησης. Αυτό υποδηλώνει ότι η επιφυλακτικότητα του εκάστοτε σχήματος συνένωσης εξαρτάται μεν από τις παραμέτρους του πίνακα $\mathcal{P}$ αλλά και τη νόρμα που εφαρμόζεται για την εξαγωγή του τελικού αποτελέσματος. Επιπλέον, το γεγονός ότι δύο από τους ταξινομητές με την καλύτερη επίδοση συνδυαζόμενοι με οποιαδήποτε νόρμα επιτυγχάνουν ακρίβεια της τάξης του 90% μόνο στο 15% ή και μικρότερο ποσοστό του πληθυσμού (δηλ. 9500 tweets) αποτελεί σοβαρή ένδειξη ότι μόνο ένα μικρό πλήθος tweets εμφανίζει ξεκάθαρη πολικότητα η οποία αποτυπώνεται στους δείκτες ομοιότητας.

Συνοψίζοντας, το σχήμα συνένωσης είναι μία αρκετά πιο ευέλικτη και σταθερή μέθοδος συνδυασμού συγκριτικά με το σχήμα ψηφοφορίας. Δεν κατατάσσει τους ταξινομητές ως προς κάποιο χαρακτηριστικό τους συνολικά π.χ. σφάλμα εκπαίδευσης, δεν αναθέτει a priori βάρη στις αποφάσεις τους αλλά εξετάζει το πόσο συμφωνούν ή διαφωνούν σε επίπεδο tweet. Σε όλα αυτά υποθέτει μόνο τη συσχέτιση μεταξύ των ταξινομητών, στοιχείο το οποίο εξαρτάται περισσότερο από το μοντέλο γράφων ν-γραμμάτων και λιγότερο από τα δεδομένα στα οποία εφαρμόζεται.

4.5 Σύνοψη Συμπερασμάτων Αξιολόγησης

Ανακεφαλαιώνοντας, τα βασικά συμπεράσματα που συνάγονται από την ανάλυση των πειραματικών αποτελεσμάτων είναι :

- Το μοντέλο γράφων ν-γραμμάτων πέτυχε μέγιστη ακρίβεια 65.62% στο εξεταζόμενο πολυγλωσσικό και πολυθεματικό σύνολο δεδομένων που προέκυψε από συνένωση 11 χειροκίνητα ταξινομημένων (manually annotated) υποσυνόλων που έχουν μελετηθεί και σε άλλες σχετικές εργασίες. Το αποτέλεσμα κρίνεται ικανοποιητικό δεδομένης της ανομοιογένειας του συνόλου και αποδεικνύει ότι η μέθοδος μπορεί να εφαρμοσθεί με επιτυχία σε δεδομένα κοινωνικών δικτύων.

- Η εφαρμογή της μεθόδου στα επιμέρους υποσύνολα δεδομένων εμφάνισε ποσοστά ακρίβειας από 50% έως 75%, αποτελέσματα που διαπιστώθηκε ότι δεν εξαρτώνται από το πλήθος των δεδομένων αλλά ενδεχομένως οφείλονται στα εγγενή χαρακτηριστικά του κάθε υποσυνόλου.

- Το ποσοστό ακρίβειας 65.62% σημειώθηκε σε αναλογία συνόλου εκπαίδευσης (δεδομένα κατασκευής και μάθησης) προς το σύνολο εξέτασης 50:50 - ιδιαίτερα χαμηλή σε σχέση με εκείνες των περισσότερων μεθόδων επιβλεπόμενης μάθησης. Ακόμη

και με αναλογία 10:90 η συγκεκριμένη προσέγγιση εμφάνισε ακρίβεια 55% ενώ με αντίστροφη αναλογία 90:10 είχε μικρή αύξηση και έφτασε το 68%.

- Σύμφωνα με τους δείκτες Precision και Recall, το μοντέλο γράφων ν-γραμμάτων εμφάνισε παρόμοια συμπεριφορά ως προς όλες τις κλάσεις πολικότητας στο σύνολο των δεδομένων.

- Η βέλτιστη επίδοση εμφανίζεται για μέγεθος ν-γράμματος $n = 4$, επιβεβαιώνοντας τις προηγούμενες μελέτες.

- Εκτός από τον ταξινομητή Support Vector Machine (SVM) - ο οποίος έχει ήδη μελετηθεί στο πλαίσιο της ανάλυσης συναισθήματος με γράφους ν-γραμμάτων - διαπιστώθηκε ότι εξίσου κατάλληλοι είναι και οι Multilayer Perceptron (MLP) και Λογιστική Παλινδρόμηση (Logistic). Αν και καθένας τους ανήκει σε διαφορετική κατηγορία αλγορίθμων Μηχανικής Μάθησης, και οι τρεις ταξινομητές εμφανίζουν ικανοποιητικά ποσοστά ακρίβειας σε όλες τις συνθήκες υπό τις οποίες εξετάστηκαν. Άλλες κατηγορίες ταξινομητών όπως οι k-NN, Naive Bayes και ο Πολυωνυμικός Naive Bayes (MNB) επιτυγχάνουν μέτρια ποσοστά ακρίβειας τις περισσότερες φορές και μόνο με κατάλληλες τροποποιήσεις εμφάνισαν συγκρίσιμα αποτέλεσμα με την $1^{η}$ ομάδα.

Κατά την τροποποίηση της αρχικής αρχιτεκτονικής του συστήματος ανάλυσης συναισθήματος παρατηρήθηκαν τα εξής :

- Σε αντίθεση με άλλες προσεγγίσεις, στο μοντέλο γράφων ν-γραμμάτων η προεπεξεργασία των δεδομένων (αφαίρεση ή αντικατάσταση Twitter tokens, αφαίρεση μόνο των υπερσυνδέσμων) δεν απέφερε αποτέλεσμα.

- Η χρήση των δεδομένων κατασκευής μαζί με τα δεδομένα μάθησης στη διαδικασία της εκπαίδευσης οδηγεί σε οριακή αύξηση κατά 0.25% της βέλτιστης επίδοσης ενώ ενισχύει σημαντικά τους NaiveBayes (+4.37%) και Multinomial NaiveBayes (MNB) (+11.53%).

- Η διακριτοποίηση των δεικτών ομοιότητας αυξάνει την αποδοτικότητα του συστήματος σε όλες τις περιπτώσεις και την αποτελεσματικότητα συγκεκριμένων μόνο ταξινομητών, ιδιαίτερα του Multinomial NaiveBayes (MNB) με ποσοστό ακρίβειας 61.75%, αύξηση 21.59%.

- Ο συνδυασμός όλων των ταξινομητών στο σχήμα Ψηφοφορίας οδηγεί σε μικρή βελτίωση της ακρίβειας. Με κατάλληλη επιλογή μέρους των συνδυαζόμενων ταξινομητών, τα ποσοστά ακρίβειας διαμορφώνονται σε ακόμη υψηλότερα επίπεδα.

- Ο συνδυασμός όλων των ταξινομητών στο σχήμα Συνένωσης δεν επιτυγχάνει καλύτερη επίδοση σε όλες τις Τριγωνικές Νόρμες που μελετήθηκαν. Ωστόσο, όπως παρατηρείται και στο σχήμα ψηφορίας, ο κατάλληλος συνδυασμός ταξινομητών διαμορφώνει τα ποσοστά κατηγοριοποίησης συστηματικά σε βελτιωμένα επίπεδα ανεξάρτητα από τη νόρμα που εφαρμόζεται. Επιπρόσθετα, μέσω του δείκτη εμπιστοσύνης, μπορεί να προσδιορισθεί ο βαθμός ομοφωνίας μεταξύ των ταξινομητών σε επίπεδο tweet και να εκτιμηθεί out-of-sample η δυσκολία κατηγοριοποίησης των εξεταζόμενων δεδομένων.

5

Επίλογος

5.1 Σύνοψη & Συμπεράσματα

Στο πλαίσιο αυτής της εργασίας, μελετήσαμε το πρόβλημα της Ανάλυσης Συναισθήματος σε δεδομένα από το κοινωνικό δίκτυο Twitter με το μοντέλο γράφων ν-γραμμάτων. Όπως προέκυψε από τη βιβλιογραφική ανασκόπηση, η συγκεκριμένη προσέγγιση είναι κατάλληλη για εφαρμογή στο πολυγλωσσικό και πολυθεματικό περιβάλλον που εξετάζουμε χάρη στην ανεξαρτησία της από τη γλώσσα (language-independent) και την υψηλή αντοχή στο θόρυβο που εμφανίζεται στα δεδομένα μικρο-ιστολογίων. Ύστερα από μία σύντομη θεωρητική θεμελίωση της μεθόδου και των υπάρχοντων αλγορίθμων Μηχανικής Μάθησης, διερευνήσαμε τον τρόπο με τον οποίο το μοντέλο γράφων ν-γραμμάτων ανταποκρίνεται στη μεταβολή των βασικών του παραμέτρων : μέγεθος συνόλου εκπαίδευσης και μέγεθος ν-γράμματος καθώς και τη συμβατότητα του με διαφορετικούς αλγορίθμους μάθησης. Έπειτα, τροποποιώντας την αρχική αρχιτεκτονική της μεθόδου, αυξήσαμε σημαντικά την ακρίβεια ταξινομητών με μέτρια επίδοση. Τέλος, προσδιορίσαμε τους συνδυασμούς ταξινομητών σε σχήματα ensemble οι οποίοι εμφανίζουν βελτιωμένα ποσοστά ακρίβειας σε συστηματική βάση.

Η συνεισφορά μας εντοπίζεται στα εξής σημεία :

- Συνθέσαμε ένα αρκετά μεγάλου μεγέθους σύνολο δεδομένων από επιμέρους σύνολα δεδομένων διαφορετικής γλώσσας και θεματολογίας ευρέως διαθέσιμα στην ακαδημαϊκή κοινότητα.

- Προτείναμε μία μέθοδο διαχωρισμού των επιμέρους υποσυνόλων ποσοστιαία ανά πολικότητα σε δεδομένα κατασκευής, μάθησης και εξέτασης η οποία εξασφαλίζει αντιπροσωπευτική εκπαίδευση και έχει ως αποτέλεσμα το σύστημα ανάλυσης συναισθήματος να εμφανίζει ικανοποιητικά ποσοστά ακρίβειας με ομοιόμορφη συμπεριφορά ως προς τις τρεις κλάσεις πολικότητας, ήδη από αρκετά μικρό ποσοστό εκπαίδευσης.

- Εξετάσαμε αλγορίθμους από τις κατηγορίες Μηχανικής Μάθησης : δένδρα αποφάσεων, παρεμβολής, στατιστικής μάθησης,νευρωνικών δικτύων, μηχανών διανυσμάτων υποστήριξης (SVMs) και μάθησης κατά περίπτωση και διαπιστώσαμε ότι

- εκτός από τα SVMs τα οποία έχουν προταθεί σε προηγούμενες μελέτες- εξίσου αποτελεσματικοί στη συγκεκριμένη μέθοδο είναι το Πολυεπίπεδο Perceptron (MLP) και η Λογιστική Παλινδρόμηση (Logistic).

- Επιβεβαιώσαμε πειραματικά ότι η χρήση γράφων ν-γραμμάτων μεγέθους $n = 4$ αποτελεί τη βέλτιστη επιλογή για την αναπαράσταση δεδομένων από μικρο-ιστολόγια.

- Διαπιστώσαμε ότι η προεπεξεργασία των δεδομένων και η χρήση των δεδομένων μάθησης στη διαδικασία της εκπαίδευσης βελτιώνουν σε μικρό βαθμό την ακρίβεια των k-NN και Naive Bayes - Multinomial Naive Bayes αντίστοιχα.

- Βελτιώσαμε σημαντικά τις επιδόσεις των Naive Bayes και Multinomial Naive Bayes με την αφαίρεση του θορύβου από τους γράφους κλάσεων.

- Αναδείξαμε ότι η διακριτοποίηση εκτός από την αύξησης της αποδοτικότητας σε όλες τις περιπτώσεις, συμβάλλει και στην αύξηση της αποτελεσματικότητας των k-NN , Naive Bayes και Multinomial Naive Bayes.

- Υλοποιήσαμε δύο διαφορετικές εκδοχές της τεχνικής ensemble το σχήμα Ψηφοφορίας και το σχήμα Συνένωσης και προσδιορίσαμε τους κατάλληλους συνδυασμούς ταξινομητών και εσωτερικών παραμέτρων για τους οποίους εμφανίζονται συστηματικά ευσταθής συμπεριφορά σε αυξημένα ποσοστά ακρίβειας.

Συνοψίζοντας, ο προσδιορισμός της πολικότητας του μηνύματος είναι ιδιαίτερα δύσκολο πρόβλημα ειδικά στις συνθήκες πολυγλωσίας και ποικίλης θεματολογίας των κοινωνικών δικτύων. Το ποσοστό ακρίβειας 65.6% και η ευστάθεια των αποτελεσμάτων που πετύχαμε με τη συγκεκριμένη μέθοδο στο σύνολο δεδομένων που δημιουργήσαμε έτσι ώστε να προσομοιάζει τη φύση και το βαθμό δυσκολίας κατηγοριοποίησης των μηνυμάτων του Twitter θεωρούνται ικανοποιητικά. Καταδείξαμε, λοιπόν, ότι με κατάλληλους συνδυασμούς παραμέτρων το μοντέλο γράφων ν-γραμμάτων είναι κατάλληλο για την Ανάλυση Συναισθήματος σε δεδομένα κοινωνικών δικτύων.

5.2 Μελλοντικές Προεκτάσεις

Πραγματοποιήσαμε μία διεξοδική έρευνα γύρω από τη συνολική λειτουργικότητα του μοντέλου γράφων ν-γραμμάτων σε πολυγλωσσικά δεδομένα. Με βάση τα πειραματικά δεδομένα, κρίνεται σκόπιμη η περαιτέρω διερεύνηση όσο αφορά την αφαίρεση θορύβου από τους γράφους κλάσεων για να διαπιστωθεί αν οδηγεί σε συστηματική βελτίωση της ακρίβειας κατηγοριοποίησης.

Επιπρόσθετα, μπορεί να εξετασθεί αν ο εμπλουτισμός του διανύσματος χαρακτηριστικών με περισσότερα γνωρίσματα σχετικά με το περιεχόμενο του μηνύματος και το ευρύτερο περιβάλλον στο οποίο εντάσσεται συμβάλλει στην αύξηση της αποτελεσματικότητας της μεθόδου.

Τέλος, η εμβέλεια της μελέτης που πραγματοποιήθηκε σε αυτή την εργασία περιορίστηκε στην ανάλυση συναισθήματος σε δεδομένα από το κοινωνικό δίκτυο Twitter. Έχει ιδιαίτερο ενδιαφέρον η εξέταση της συμπεριφορά του μοντέλου γράφων ν-γραμμάτων σε δεδομένα από άλλες πηγές όπως οι αναρτήσεις στο Facebook, τα σχόλια στο YouTube και οι κριτικές προϊόντων στο Amazon.

Βιβλιογραφία

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38. Association for Computational Linguistics, 2011.

[2] Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora Varvarigou. Content vs. context for sentiment analysis: a comparative analysis over microblogs. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 187–196. ACM, 2012.

[3] Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora A. Varvarigou. Textual and contextual patterns for sentiment analysis over microblogs. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 453–454, 2012.

[4] Nathan Aston, Timothy Munson, Jacob Liddle, Garrett Hartshaw, Dane Livingston, and Wei Hu. Sentiment analysis on the social networks using stream algorithms. *Journal of Data Analysis and Information Processing*, 2(02):60, 2014.

[5] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[6] Adam Bermingham and Alan F Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836. ACM, 2010.

[7] Albert Bifet and Eibe Frank. Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, pages 1–15. Springer, 2010.

[8] Piero Bonissone, Kai Goebel, and Weizhong Yan. Classifier fusion using triangular norms. In *Multiple Classifier Systems*, pages 154–163. Springer, 2004.

[9] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 241–249. Association for Computational Linguistics, 2010.

[10] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[11] Daniel Gayo-Avello, Panagiotis Takis Metaxas, and Eni Mustafaraj. Limits of electoral predictions using twitter. In *ICWSM*, 2011.

[12] George Giannakopoulos, Vangelis Karkaletsis, George A. Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *TSLP*, 5(3), 2008.

[13] George Giannakopoulos and Themis Palpanas. Content and type as orthogonal modeling features: a study on user interest awareness in entity subscription services. *International Journal On Advances in Networks and Services*, 3(1 and 2):296–309, 2010.

[14] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[15] Tobias Günther and Lenz Furrer. Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 328–332, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[16] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[17] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web*, pages 607–618. International World Wide Web Conferences Steering Committee, 2013.

[18] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.

[19] Soo-Min Kim and Eduard Hovy. Identifying and analyzing judgment opinions. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 200–207, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[20] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541, 2011.

[21] Akshi Kumar and Teeja Mary Sebastian. Sentiment analysis on twitter. *IJCSI International Journal of Computer Science Issues*, 9(3):372–378, 2012.

[22] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[23] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, 2012.

[24] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR*, abs/1308.6242, 2013.

[25] Francisco Moreno-Seco, José M Inesta, Pedro J Ponce De León, and Luisa Micó. Comparison of classifier fusion methods for classification in pattern recognition tasks. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 705–713. Springer, 2006.

[26] Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *Machine Learning: ECML 2006*, pages 318–329. Springer, 2006.

[27] Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. 2013.

[28] Sascha Narr, Michael Hülfenhaus, and Sahin Albayrak. Language-independent twitter sentiment analysis. In *KDML workshop on knowledge discovery, data mining and machine learning*, 2012.

[29] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11:122–129, 2010.

[30] Reynier Ortega, Adrian Fonseca, and Andrés Montoyo. Ssa-uo: unsupervised twitter sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 501–507, 2013.

[31] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.

[32] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[33] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[34] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[35] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics, 2010.

[36] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, 2005.

[37] Eshrag Refaee and Verena Rieser. Subjectivity and sentiment analysis of arabic twitter feeds with limited resources. In *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme*, page 16, 2014.

[38] Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. *Proc. SemEval*, 2014.

[39] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *11th Extended Semantic Web Conference ESWC2014*, 2014.

[40] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *The Semantic Web: Trends and Challenges*, pages 83–98. Springer, 2014.

[41] Hassan Saif, Yulan He, and Harith Alani. Semantic smoothing for twitter sentiment analysis. In *Proceedings of the 10th International Semantic Web Conference (ISWC) (2011.*

[42] Hassan Saif, Yulan He, and Harith Alani. Alleviating data sparsity for twitter sentiment analysis. *Making Sense of Microposts (# MSM2012)*, 2012.

[43] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. *The Semantic Web–ISWC 2012*, pages 508–524, 2012.

[44] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[45] David A Shamma, Lyndon Kennedy, and Elizabeth F Churchill. Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of the first SIGMM workshop on Social media*, pages 3–10. ACM, 2009.

[46] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics, 2011.

[47] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[48] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[49] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[50] Erik Tromp. Multilingual sentiment analysis on social media. Master's thesis, Eindhoven University of Technology, Eindhoven, 7 2011.

[51] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[52] Julio Villena Román, Sara Lana Serrano, Eugenio Martínez Cámara, and José Carlos González Cristóbal. Tass-workshop on sentiment analysis at sepln. 2013.

[53] Cynthia Whissell, Michael Fournier, René Pelland, Deborah Weir, and Katherine Makarec. A dictionary of affect in language: Iv. reliability, validity, and applications. *Perceptual and Motor Skills*, 62(3):875–888, 1986.

[54] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.

[55] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[56] Ley Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89, 2011.

[57] Ιωάννης Βλαχάβας, Πέτρος Κεφαλάς, Νικόλαος Βασιλειάδης, Φώτης Κόκκορας, and Ηλίας Σακελλαρίου. *Τεχνητή Νοημοσύνη*. Εκδόσεις Πανεπιστημίου Μακεδονίας, Θεσσαλονίκη, 3rd edition, 2006.

Παράρτημα

Α΄ Πειραματικά Αποτελέσματα

Παρατίθενται τα πειραματικά αποτελέσματα που προέκυψαν από την εφαρμογή της μεθόδου γράφων ν-γραμμάτων σε κάθε ένα από τα επιμέρους σύνολα δεδομένων για το συνδυασμό Buildset = 20 Trainset = 30 και Testset = 50.

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	**58.22**	58.15	57.36	43.50	48.66	58.15	55.74
2	**58.18**	58.15	57.57	45.98	53.51	58.15	57.43
3	57.57	58.15	**58.18**	47.25	54.09	58.15	56.16
4	58.01	**58.15**	**58.15**	47.66	53.89	**58.15**	56.74
5	58.08	**58.15**	57.98	46.49	54.16	**58.15**	**58.15**
6	57.87	**58.15**	57.94	47.25	55.98	**58.15**	58.12
7	58.12	**58.15**	58.01	49.11	55.95	**58.15**	57.81

RRarabic

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	**57.88**	**57.88**	**57.88**	51.47	54.51	**57.88**	49.96
2	59.48	57.88	**60.32**	52.91	53.07	57.88	57.03
3	**61.33**	57.88	60.91	55.43	56.70	57.88	59.22
4	**61.08**	58.55	60.15	55.52	57.37	57.88	58.64
5	**60.74**	58.64	60.07	56.36	52.99	57.88	55.86
6	**60.99**	58.64	60.07	55.27	50.55	57.88	57.12
7	**60.83**	58.64	59.98	53.66	48.19	58.05	50.88

HCR

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	68.59	68.59	69.35	64.49	51.33	68.59	**69.43**
2	71.86	72.02	**72.70**	70.95	67.30	68.59	70.72
3	**73.38**	71.86	73.08	70.95	69.66	68.59	70.80
4	**73.38**	71.79	72.02	71.18	69.66	69.58	70.57
5	**72.55**	71.79	72.47	70.65	69.96	71.79	67.68
6	**72.70**	71.71	72.09	70.57	70.27	71.79	69.66
7	**72.09**	71.71	71.86	71.41	70.34	71.79	71.41

MDL-DE

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	64.15	63.96	64.19	62.11	50.38	63.96	**64.36**
2	69.10	67.81	**69.81**	69.55	63.58	63.96	68.79
3	69.32	68.83	**70.74**	69.74	69.10	63.96	69.49
4	69.70	68.74	**70.85**	69.08	69.36	63.96	68.66
5	68.72	68.74	**70.06**	68.36	69.29	68.21	68.87
6	69.46	68.66	**69.72**	69.42	68.96	68.64	67.36
7	**69.32**	68.64	69.25	67.96	68.57	68.68	68.57

MDL-EN

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	59.53	54.51	58.70	55.44	57.02	54.51	**60.19**
2	62.51	60.84	**64.84**	61.95	60.09	54.51	61.40
3	63.63	60.74	**64.00**	63.07	61.30	54.51	61.02
4	**63.63**	60.74	62.88	**63.63**	60.09	60.37	63.07
5	61.02	60.84	62.42	61.95	58.70	60.84	**63.07**
6	62.60	60.84	**63.16**	62.42	56.37	60.84	61.40
7	60.84	60.84	62.05	61.58	55.53	60.93	**62.33**

MDL-FR

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	51.80	50.63	**53.64**	53.31	44.27	38.58	50.54
2	60.25	60.75	**63.35**	61.42	55.15	42.01	58.91
3	62.01	59.00	61.09	**62.93**	54.23	53.81	57.24
4	56.82	56.15	59.58	**61.34**	51.63	55.82	55.82
5	53.14	53.05	58.16	**62.76**	49.29	56.07	56.65
6	50.13	50.79	57.66	**59.50**	48.37	55.31	55.23
7	44.85	50.54	56.40	**57.49**	47.53	55.65	55.82

MDL-PT

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	43.95	45.56	47.58	47.58	41.13	44.35	**48.39**
2	44.35	48.79	**51.61**	48.39	45.56	44.35	44.76
3	44.76	45.16	**50.00**	40.73	43.15	44.35	46.77
4	46.77	49.60	**50.00**	38.31	36.29	44.35	42.74
5	45.16	46.37	**48.79**	39.52	37.90	44.35	44.35
6	**44.35**	43.55	**44.35**	42.74	38.31	**44.35**	**44.35**
7	44.35	43.95	**47.18**	43.55	37.50	44.35	45.97

NTUA

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	69.72	65.19	**71.08**	64.25	66.88	62.78	70.24
2	68.03	68.03	**73.50**	62.57	67.72	62.78	68.35
3	72.13	**73.08**	**73.08**	61.51	66.67	62.78	65.62
4	72.87	72.98	**73.19**	63.41	68.03	62.78	70.14
5	**73.19**	71.92	72.66	62.15	68.03	62.78	70.87
6	70.56	69.51	**71.08**	61.41	66.56	62.78	70.45
7	**68.66**	68.56	**68.66**	59.52	65.93	62.78	67.51

OMD

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	68.68	68.74	**68.81**	55.59	53.49	68.74	67.03
2	69.06	68.74	**71.35**	58.70	58.58	68.74	68.74
3	70.46	69.12	**70.84**	59.78	63.41	68.74	68.87
4	**70.46**	69.19	70.27	59.09	54.00	68.74	65.06
5	**70.08**	69.19	69.76	59.40	52.41	68.74	69.31
6	**70.33**	69.19	69.82	60.36	52.54	68.74	66.26
7	**70.08**	69.25	69.50	60.80	50.95	69.12	67.03

Sanders

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	46.67	46.73	**49.27**	38.95	46.53	46.73	46.75
2	48.03	53.33	**53.86**	41.98	42.40	46.73	47.51
3	53.34	54.73	**55.25**	44.22	48.59	46.73	50.24
4	53.17	54.26	**54.48**	44.72	50.52	46.73	50.51
5	52.08	52.95	**54.14**	44.36	46.98	46.75	51.30
6	51.55	50.91	**53.59**	43.16	44.11	46.97	51.43
7	50.82	48.95	**52.71**	42.98	43.79	47.12	50.48

SemEval2014

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	58.00	56.08	**58.64**	48.01	55.02	53.73	56.71
2	60.92	60.03	**61.76**	50.12	53.67	53.73	57.68
3	**63.46**	61.97	62.85	51.67	60.54	53.73	60.56
4	**63.82**	61.42	62.87	53.14	61.99	53.73	61.00
5	**63.80**	61.28	62.41	52.62	62.99	53.73	61.68
6	**63.60**	60.96	62.09	53.24	63.40	55.18	60.64
7	62.95	60.48	61.70	52.94	**63.56**	57.94	60.88

SentiTuites-PT

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	45.33	46.08	**46.79**	38.67	41.60	46.08	45.23
2	48.25	46.18	**51.04**	40.56	45.33	46.08	47.07
3	49.10	46.98	**50.00**	40.18	44.52	46.08	45.00
4	48.35	46.08	**48.63**	38.10	45.14	46.08	45.42
5	47.73	46.08	**48.25**	40.56	37.06	46.08	44.85
6	46.93	46.08	**47.64**	40.08	35.17	46.08	45.37
7	**47.88**	46.08	46.84	41.55	36.17	46.08	46.84

SSTweet

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	54.07	54.07	**56.26**	51.96	52.27	54.07	55.93
2	65.61	65.64	**67.74**	59.25	56.38	54.07	63.68
3	72.79	72.77	**73.08**	66.76	59.83	54.07	70.87
4	74.96	**75.15**	**75.15**	69.71	63.28	54.07	73.30
5	74.78	**75.04**	74.96	70.24	67.09	54.07	73.92
6	74.08	74.23	**74.25**	69.80	67.83	54.22	73.07
7	72.97	72.91	**73.11**	68.38	66.89	54.40	72.22

TASS

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	**62.84**	59.66	62.42	56.69	56.54	40.29	61.52
2	72.79	71.85	**74.33**	67.23	64.99	40.29	70.50
3	74.97	75.53	**75.88**	69.96	70.14	49.58	71.65
4	75.02	75.40	**75.70**	69.71	70.78	65.20	72.03
5	72.48	73.88	**74.06**	67.89	69.15	67.22	71.23
6	71.52	72.23	**72.45**	67.02	66.79	65.80	69.72
7	**70.93**	69.81	70.53	66.80	64.91	61.71	69.34

Tromp-EN

NGram	MLP	SVM	Logistic	kNN	NB	MNB	C4.5
1	**62.15**	54.12	59.20	54.31	48.84	43.44	56.75
2	**71.17**	69.32	71.05	65.46	62.70	43.44	68.22
3	72.82	**73.26**	72.86	68.26	66.40	48.40	69.83
4	71.76	**71.88**	71.33	66.99	66.84	60.10	69.24
5	70.15	**70.22**	70.07	64.99	65.73	61.44	65.62
6	**68.37**	65.89	67.51	62.74	63.73	59.00	65.85
7	**66.36**	60.97	65.89	60.85	62.03	54.75	64.40

Tromp-NL

Β΄ Πηγαίος Κώδικας

Παρατίθεται ο πηγαίος κώδικας των βασικών Κλάσεων του Συστήματος Ανάλυσης Συναισθήματος. Η υλοποίηση της πλήρους λειτουργικότητάς του βρίσκεται στο αποθετήριο: https://github.com/dtz/NTUA-THESIS-Twitter-Sentiment-Analysis.

Η κατασκευή των γράφων κλάσεων και η λειτουργικότητα της αφαίρεσης θορύβου υλοποιούνται μέσω της κλάσης `BuildGraphs` χρησιμοποιώντας τη βιβλιοθήκη `JInsect`.

```
1  import Models.TokenNGramGraphs;
2  import gr.demokritos.iit.jinsect.documentModel.representations.DocumentNGramGraph;
3  import gr.demokritos.iit.jinsect.documentModel.representations.DocumentWordGraph;
4  import java.io.File;
5  import java.util.Random;
6
7  public class BuildGraphs {
8
9   private static int nSize;
10
11  public static void build(String [] buildset,String [] graphs,int ngramSize){
12
13   nSize = ngramSize;
14   for (int k = 0 ; k < buildset.length;k++) {
15    String build = buildset[k];
16    String graphOutput = graphs[k];
17    File file = new File(build);
18    // load buildset serialized object
19    String[] tweetsArray = (String[])
       SerializationUtilities.loadSerializedObject(file.getAbsolutePath());
20    int noOfTweets = tweetsArray.length;
21    int noOfDocuments = 0;
22    final DocumentNGramGraph graphModel = new DocumentNGramGraph(nSize, nSize, nSize);
23    final DocumentNGramGraph tempGraph = new DocumentNGramGraph(nSize, nSize, nSize);
24    for (int i = 0; i < noOfTweets; i++) {
25     noOfDocuments++;
26     // create temporary tweet graph
27     tempGraph.setDataString(tweetsArray[i]);
28     // merge using update functionality
29     graphModel.merge(tempGraph, 1.0 - (noOfDocuments-1.0)/noOfDocuments);
30    }
31    // store graphs to memory
32    SerializationUtilities.storeSerializedObject(graphModel,graphOutput);
33   }
34  }
35
36  public DocumentNGramGraph [] removeNoiseFromGraphs(DocumentNGramGraph [] graphs){
37
38   DocumentNGramGraph [] noiseFreeGraphs = new DocumentNGramGraph[3];
39   DocumentNGramGraph tempGraph ;
40   DocumentNGramGraph commonSubgraph;
41
42   noiseFreeGraphs[0] = graphs[0].clone(); // negative class
43   noiseFreeGraphs[1] = graphs[1].clone(); // neutral  class
44   noiseFreeGraphs[2] = graphs[2].clone(); // positive class
45
46   int i = 0;
```

```
47   do{
48    // intersection of negative and neutral class graphs
49    tempGraph = noiseFreeGraphs[0].intersectGraph(noiseFreeGraphs[1]);
50    // intersection of tempGraph and positive class graphs
51    commonSubgraph = noiseFreeGraphs[2].intersectGraph(tempGraph);
52    //remove calculated commonSubgraph from class graphs
53    noiseFreeGraphs[0] = noiseFreeGraphs[0].allNotIn(commonSubgraph);
54    noiseFreeGraphs[1] = noiseFreeGraphs[1].allNotIn(commonSubgraph);
55    noiseFreeGraphs[2] = noiseFreeGraphs[2].allNotIn(commonSubgraph);
56    i++;
57    // iterate until convergance condition is satisfied or limit has been reached
58   }while (!commonSubgraph.isEmpty() && (i < LIMIT));
59
60   return noiseFreeGraphs;
61  }
62 }
```

Κώδικας Β΄.1: Λειτουργικότητες Κατασκευής Γράφων Κλάσεων και Αφαίρεσης Θορύβου

Η αναπαράσταση των δεικτών ομοιότητας στο διάνυσμα χαρακτηριστικών των μηνυμάτων (tweets) για την Εκπαίδευση και Εξέταση των ταξινομητών γίνεται μέσω αρχείου της μορφής **ARFF** (Attribute-Relation File Format) το οποίο κατασκευάζεται μέσω της κλάσης `ARFFConstructor`.

```
1  import java.io.File;
2  import java.io.FileNotFoundException;
3  import java.io.FileOutputStream;
4  import java.io.PrintWriter;
5  import java.util.ArrayList;
6
7  import weka.core.Attribute;
8  import weka.core.DenseInstance;
9  import weka.core.Instances;
10
11 public class ARFFConstructor {
12  private ArrayList<Attribute> atts;
13  private Instances    data;
14  private FileOutputStream   output ;
15  private PrintWriter    writer ;
16  private ArrayList<String>  attVals;
17  public ARFFConstructor(String file) {
18   super();
19   try {
20    output = new FileOutputStream(file, false);
21   } catch (FileNotFoundException e) {
22    e.printStackTrace();
23   }
24   writer =  new PrintWriter(output);
25   atts = new ArrayList<Attribute>();
26   // creating the appropriate header
27   atts.add(new Attribute(''NegCont''));
28   atts.add(new Attribute(''NegNormVal''));
29   atts.add(new Attribute(''NegVal''));
30   atts.add(new Attribute(''NeutCont''));
31   atts.add(new Attribute(''NeutNormVal''));
```

```
  atts.add(new Attribute(''NeutVal''));
  atts.add(new Attribute(''PosCont''));
  atts.add(new Attribute(''PosNormVal''));
  atts.add(new Attribute(''PosVal''));

  attVals = new ArrayList<String>();
  attVals.add(''1'');
  attVals.add(''2'');
  attVals.add(''3'');
  atts.add(new Attribute(''Polarity'',attVals));
  // initialising data object
  data = new Instances(relationLabel(file), atts, 0);
 }
 public void add(double [] values){
  int pol = (int) values[9];
  // save similarities and class index to data object
  values[9] = attVals.indexOf(String.valueOf(pol));
  data.add(new DenseInstance(1.0, values));
 }
 public void export(){
  // export to .arff file
  writer.print(data.toString());
  writer.close();
 }
 private String relationLabel(String filename){
  // create relation label from filename
  String separator = ''\\\\'';
  if (File.separator.equals(''/'')){
   separator = ''/'';
  }
  String [] tokens = filename.split(separator);
  String name = tokens[tokens.length-1];
  if (name.indexOf(''.arff'')!= -1){
   name = name.replace(''.arff'',"");
  }
  return name;
 }
}
```

Κώδικας Β΄.2: Δημιουργία αρχείου ARFF για την αναπαράσταση του διανύσματος χαρακτηριστικών

Οι λειτουργικότητες που αφορούν τους ταξινομητές υλοποιούνται μέσω της κλάσης `ClassifierFunctions` με τη βοήθεια της βιβλιοθήκης WEKA. Η κατασκευή των ταξινομητών επιτυγχάνεται χρησιμοποιώντας την τεχνική reflection προσδιορίζοντας τον εκάστοτε τύπου του ταξινομητή μέσω αρχείου. Η Εκπαίδευση και Εξέταση των ταξινομητών πραγματοποιούνται μέσω κατάλληλου αρχείου ARFF και των μέθοδων `buildClassifier` και `classifyInstance` αντίστοιχα. Υποστηρίζονται επίσης οι λειτουργίες κατασκευής αρχείων ARFF Εκπαίδευσης και Εξέτασης καθώς και της αντίστοιχης εκδοχής τους με διακριτοποιημένους δείκτες ομοιότητας.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
```

```
import java.io.IOException;
import Fusion.Fuser;

import gr.demokritos.iit.jinsect.documentModel.comparators.NGramCachedGraphComparator;
import gr.demokritos.iit.jinsect.documentModel.representations.DocumentNGramGraph;
import gr.demokritos.iit.jinsect.documentModel.representations.DocumentNGramHGraph;
import gr.demokritos.iit.jinsect.structs.GraphSimilarity;

import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.functions.MultilayerPerceptron;
import weka.classifiers.trees.J48;
import weka.classifiers.bayes.NaiveBayes;
import weka.classifiers.bayes.NaiveBayesMultinomial;
import weka.classifiers.functions.SMO;
import weka.classifiers.lazy.IBk;
import weka.core.DenseInstance;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;

public class ClassifierFunctions {

 private static int N_GRAM_SIZE;
 private final static int NEGATIVE =1;
 private final static int POSITIVE = 2;
 private final static int NEUTRAL = 3;
 private final static int EQUAL = 4;

 private final static NGramCachedGraphComparator comparator = new
    NGramCachedGraphComparator();

 private static Classifier classifier;
 private static Classifier [] classifierArray;
 private static DocumentNGramGraph negativeGraph;
 private static DocumentNGramGraph neutralGraph;
 private static DocumentNGramGraph positiveGraph;
 private static Instances trainInstances;
 private static Instances testInstances;
 private static BufferedReader reader = null;

 private static void loadGraphs(String[] graphPaths) throws Exception {
  Log.println("Loading negative class graph...");
  negativeGraph = (DocumentNGramGraph)
    SerializationUtilities.loadSerializedObject(graphPaths[0]);
  Log.println("Loading neutral class graph...");
  neutralGraph = (DocumentNGramGraph)
    SerializationUtilities.loadSerializedObject(graphPaths[1]);
  Log.println("Loading positive class graph...");
  positiveGraph = (DocumentNGramGraph)
    SerializationUtilities.loadSerializedObject(graphPaths[2]);
 }
 private static void loadTTInstances(String trainPath,String testPath){

  DataSource source,source2;
  try {
   Log.println("Loading train instances...");
   source = new DataSource(trainPath);
   trainInstances = source.getDataSet();
```

```
59    trainInstances.setClassIndex(trainInstances.numAttributes() - 1);
60    source2 = new DataSource(testPath);
61
62    Log.println("Loading test instances...");
63    testInstances = source2.getDataSet();
64    testInstances.setClassIndex(testInstances.numAttributes() - 1);
65   } catch (Exception e) {
66    e.printStackTrace();
67   }
68  }
69
70  private static Classifier buildTheClassifier(String trainARFF,String classifierName){
71   String classifierLabel = LabellingUtilities.classifierLabel(trainARFF,
       classifierName);
72   Classifier theClassifier = null;
73   Log.println("Building classifier "+classifierName+"...");
74   try {
75    // reflection technique
76    theClassifier = (Classifier) Class.forName(classifierName).newInstance();
77    theClassifier.buildClassifier(trainInstances);
78    Log.println("Storing classifier...");
79    SerializationUtilities.storeSerializedObject(theClassifier,classifierLabel);
80   } catch (Exception e) {
81    Log.println("Building classifier failed!");
82    e.printStackTrace();
83   }
84   return theClassifier;
85
86  }
87
88  public static void createARFFsOnly(String [] graphs,String trainset,String testset)
       throws Exception{
89   N_GRAM_SIZE = getNGramSize(graphs[0]);
90   Log.label("ClassifierFunctions");
91   Log.println("Starting...");
92   loadGraphs(graphs);
93   createTrainTestARFFs(trainset,testset);
94  }
95
96  private static void createTrainTestARFFs(String trainset, String testset) throws
       Exception {
97
98   String [] files = { trainset,testset};
99   Log.println("Creating Train & Test ARFF files...");
100  for (String filename : files){
101   String ARFFname = LabellingUtilities.ARFFLabel(filename);
102   ARFFConstructor arff = new ARFFConstructor(ARFFname);
103   SerialTweet [] tweetList = (SerialTweet [] )
       SerializationUtilities.loadSerializedObject(filename);
104
105   for (SerialTweet tweet : tweetList){
106    double [] values =  calculateSimilarities(tweet.getText());
107    values[9] = tweet.getPolarity();
108    arff.add(values);
109   }
110   arff.export();
111  }
112 }
```

```
113
114 public static void removeNoise(String [] graphs,String trainset,String testset)
      throws Exception{
115  N_GRAM_SIZE = getNGramSize(graphs[0]);
116  Log.label(''ClassifierFunctions'');
117  Log.println(''Starting...'');
118  loadGraphs(graphs);
119  createTrainTestARFFsNoNoise(trainset,testset);
120 }
121 private static void createTrainTestARFFsNoNoise(String trainset, String testset)
      throws Exception {
122
123  String [] files = { trainset,testset};
124  Log.println(''Creating Train & Test ARFF files...'');
125  for (String filename : files){
126   String ARFFname = LabellingUtilities.ARFFLabelNoNoise(filename);
127   ARFFConstructor arff = new ARFFConstructor(ARFFname);
128   SerialTweet [] tweetList = (SerialTweet [] )
       SerializationUtilities.loadSerializedObject(filename);
129
130   for (SerialTweet tweet : tweetList){
131    double [] values =  calculateSimilarities(tweet.getText());
132    values[9] = tweet.getPolarity();
133    arff.add(values);
134   }
135   arff.export();
136  }
137 }
138 public static void discretizeARFFsOnly(String trainset,String testset){
139  Log.label(''ClassifierFunctions'');
140  Log.println(''Starting Discretization...'');
141  String discr_trainset = discretizeSimilarities(trainset);
142  Log.println(''Discretization of Train ARFF completed...'');
143  String discr_testset = discretizeSimilarities(testset);
144  Log.println(''Discretization of Test ARFF completed...'');
145  Log.println(''Completed Discretization...'');
146 }
147
148 private static String discretizeSimilarities(String original){
149  String ARFFname = ''error'';
150  try {
151   reader = new BufferedReader(new FileReader(original));
152   ARFFname = LabellingUtilities.ARFFDiscretizedLabel(original);
153   ARFFConstructor arff = new ARFFConstructor(ARFFname);
154   String line ;
155   do {
156    line = reader.readLine();
157   }while (!line.equals(''@data''));
158   while ((line = reader.readLine())!= null){
159    String [] tokens = line.split('','');
160    double [] values = new double[tokens.length];
161    for (int i = 0 ; i < tokens.length;i++){
162     values[i] = Double.parseDouble(tokens[i]);
163    }
164    double [] discrete = new double [values.length];
165    for (int i = 0 ; i < 6 ;i++){
166     discrete[i] = dsim(values[i],i,values[i+3],i+3);
167    }
```

```
168     for (int i = 6 ; i < 9 ;i++){
169      discrete[i] = dsim(values[i-6],i-6,values[i],i);
170     }
171     discrete[9] = values[9];
172     arff.add(discrete);
173    }
174    arff.export();
175    reader.close();
176
177   } catch (IOException e) {
178    e.printStackTrace();
179   }
180   return ARFFname;
181  }
182  public static double [] fusionEnsemble(String trainset,String testset,int
       norm,double weight , double distance,String [] classifiers) throws Exception {
183
184   loadTTInstances(trainset,testset);
185   classifierArray = new Classifier[classifiers.length];
186
187   int [] accuracy = new int [classifiers.length+1];
188   int [] pred = new int [classifiers.length];
189   double polarity = 0;
190
191   for (int i = 0 ; i < classifiers.length ; i++){
192    buildClassifier(LabellingUtilities.ARFFLabel(trainset),classifiers[i]);
193    classifierArray[i]= classifier;
194   }
195
196   Fuser fuser = new Fuser(3,weight,distance);
197   fuser.setTNorm(norm);
198
199   for (int i = 0; i < testInstances.numInstances(); i++) {
200    polarity = testInstances.instance(i).classValue();
201    for (int j = 0 ; j < classifierArray.length;j++){
202     pred[j] = (int) classifierArray[j].classifyInstance(testInstances.instance(i));
203     if (pred[j] == polarity){
204      accuracy[j]= accuracy[j]+1;
205     }
206     double[] probabilities =
       classifierArray[j].distributionForInstance(testInstances.instance(i));
207     fuser.add(probabilities);
208    }
209    double prediction = fuser.combine();
210    if (prediction == polarity){
211     accuracy[classifiers.length]++;
212    }
213    double t = fuser.confidenceIndex();
214   }
215   double[] results = new double[accuracy.length];
216   for (int i = 0 ; i < results.length ; i++){
217    results[i] = 100*(double)accuracy[i]/testInstances.numInstances();
218   }
219   return results;
220  }
221
222  public static double [] calculateSimilarities(String document) throws Exception {
```

```
223   DocumentNGramGraph tempGraph = new DocumentNGramHGraph(N_GRAM_SIZE, N_GRAM_SIZE,
        N_GRAM_SIZE, 50);
224   tempGraph.setDataString(document); // create tweet graph and calculate similarities
225
226   GraphSimilarity negSimilarity = comparator.getSimilarityBetween(tempGraph,
        negativeGraph);
227   GraphSimilarity neuSimilarity = comparator.getSimilarityBetween(tempGraph,
        neutralGraph);
228   GraphSimilarity posSimilarity = comparator.getSimilarityBetween(tempGraph,
        positiveGraph);
229
230   double[] values = new double[3*3+1];
231
232   values[0] = negSimilarity.ContainmentSimilarity;
233   values[1] = negSimilarity.ValueSimilarity/negSimilarity.SizeSimilarity;
234   values[2] = negSimilarity.ValueSimilarity;
235
236   values[3] = neuSimilarity.ContainmentSimilarity;
237   values[4] = neuSimilarity.ValueSimilarity/neuSimilarity.SizeSimilarity;
238   values[5] = neuSimilarity.ValueSimilarity;
239
240   values[6] = posSimilarity.ContainmentSimilarity;
241   values[7] = posSimilarity.ValueSimilarity/posSimilarity.SizeSimilarity;
242   values[8] = posSimilarity.ValueSimilarity;
243
244   values[9] = -1;
245
246   for (int i = 0 ; i < 9 ; i++){
247   // weka precision overflow fix
248    if (values[i] < 1E-6){
249     values[i] = 0;
250    }
251   }
252   return values;
253  }
254
255  public static double [] calculateProbabilities(String trainset,String testset,String
       [] classifiers) throws Exception {
256   Log.label(''ClassifierFunctions'');
257   Log.println(''Loading Train & Test ARFFs...'');
258   loadTTInstances(trainset,testset);
259   classifierArray = new Classifier[classifiers.length];
260   int [] correct = new int [classifiers.length];
261
262   Log.println(''Building classifiers...'');
263   for (int i = 0 ; i < classifiers.length ; i++){
264    classifierArray[i]=buildTheClassifier(trainset,classifiers[i]);
265   }
266
267   for (int j = 0 ; j < classifierArray.length;j++){
268
269    String clfname = classifierName(classifiers[j]);
270    int ngram = getNGramSize(trainset);
271    Output output = new
        Output(clfname,ngram,trainInstances.numInstances(),testInstances.numInstances());
272    String label = LabellingUtilities.outputLabel(trainset, classifiers[j]);
273    Log.println(''Calculating Probabilities for ''+clfname);
274
```

```
275     for (int i = 0; i < trainInstances.numInstances(); i++) {
276      output.trainActualClass[i] = (int) trainInstances.instance(i).classValue();
277      output.trainProbs[i] =
         classifierArray[j].distributionForInstance(trainInstances.instance(i));
278     }
279     for (int i = 0; i < testInstances.numInstances(); i++) {
280      int pol = (int) testInstances.instance(i).classValue();
281      output.testActualClass[i] = pol;
282      output.testProbs[i]
         =classifierArray[j].distributionForInstance(testInstances.instance(i));
283      int prediction = (int)
         classifierArray[j].classifyInstance(testInstances.instance(i));
284      if (prediction == pol){
285       correct[j] += 1 ;
286      }
287     }
288
289     SerializationUtilities.storeSerializedObject(output,label);
290     Log.println("Stored output object...");
291    }
292
293    double[] results = new double[classifiers.length];
294    for (int i = 0 ; i < results.length ; i++){
295     results[i] = 100*(double)correct[i]/testInstances.numInstances();
296    }
297    return results;
298   }
299
300   private static int getNGramSize(String filepath) {
301    String [] filename = filepath.split(Configurator.separator);
302    String [] tokens = filename[filename.length-1].split("_");
303    String tmp = tokens[tokens.length-3];
304    tmp = tmp.replace("n","");
305    return Integer.parseInt(tmp);
306
307   }
308   private static String classifierName(String cname){
309    String [] wekapath = cname.split("\\.");
310    return wekapath[wekapath.length-1];
311   }
312
313   private static int dsim(double fvalue,int first,double svalue,int second){
314    int firstClass = polarityClass(first);
315    int secondClass = polarityClass(second);
316    double max = Math.max(fvalue,svalue);
317    if (fvalue == svalue)
318     return EQUAL;
319    else if (fvalue == max)
320     return firstClass;
321    return secondClass;
322   }
323
324   private static int polarityClass(int index){
325    int response = -1;
326    if ( index < 3)
327     response = NEGATIVE;
328    else if (index < 6)
329     response = NEUTRAL;
```

```
330   else
331    response = POSITIVE;
332   return response;
333  }
334 }
```

Κώδικας Β΄.3: Βασικές Λειτουργικότητες Κατασκευής, Εκπαίδευσης και Εξέτασης Ταξινομητών

Οι λειτουργικότητες του Σχήματος Ψηφοφορίας υλοποιούνται μέσω της κλάσης **EnsembleVote**.

```
1  public class EnsembleVote {
2   private double [] Wplurality;
3   private double [] Wsimple;
4   private double [] Wrescaled;
5   private double [] Wbestworst;
6   private int [] initial ;
7   private int N;
8
9   public EnsembleVote(int numInstances, int[] correct) {
10   Wplurality = new double[correct.length];
11   Wsimple = new double[correct.length];
12   Wrescaled = new double[correct.length];
13   Wbestworst = new double[correct.length];
14
15   initial = correct;
16   N = numInstances;
17   double sum = 0;
18
19   calculatePluralityWeights();
20   calculateSimpleWeigths();
21   calculateRescaledWeights();
22   calculateBestWorstWeights();
23  }
24  private void calculatePluralityWeights(){
25
26   double w = 1 / (double)initial.length;
27   for (int i = 0 ; i < Wplurality.length;i++){
28    Wplurality[i] = w;
29   }
30  }
31  private void calculateSimpleWeigths(){
32   int sum = 0;
33   for (int t : initial){
34    sum += t;
35   }
36   for (int i = 0 ; i < Wsimple.length ;i++){
37    double w = (double)initial[i]/(double)sum;
38    Wsimple[i] = w;
39   }
40  }
41  private void calculateRescaledWeights(){
42
43   double [] temp = new double[initial.length];
44   for (int i = 0 ; i < initial.length;i++){
45    int e = N - initial[i];
46    double nominator = 1.5 * e ;
```

```
47    int denominator = N ;
48    double frac = 1 - (nominator/denominator);
49    temp[i] = Math.max(0, frac);
50   }
51   double sum = 0;
52   for (double t : temp){
53    sum += t;
54   }
55   for (int i = 0 ; i < Wrescaled.length ;i++){
56    double w = temp[i]/sum;
57    Wrescaled[i] = w;
58   }
59  }
60  private void calculateBestWorstWeights(){
61   int minValue = N;
62   int maxValue = 0;
63   for (int t : initial){
64    int error = N - t ;
65    if (minValue < error){
66     minValue = error;
67    }
68    if (maxValue > error){
69     maxValue = error;
70    }
71   }
72   // calculated max and min values of errors ;
73   double [] temp = new double[initial.length];
74   double [] squared = new double[initial.length];
75   for (int i = 0 ; i < initial.length;i++){
76    int e = N - initial[i];
77    int nominator = e - minValue;
78    double denominator = (double)maxValue - minValue;
79    double frac = nominator/denominator;
80    temp[i] = 1 - frac;
81    nominator = maxValue - e;
82    frac = nominator/denominator;
83   }
84   double sum = 0;
85   for (double t : temp){
86    sum += t;
87   }
88   for (int i = 0 ; i < Wbestworst.length ;i++){
89    double w = temp[i]/sum;
90    Wbestworst[i] = w;
91   }
92  }
93  private void calculateMajorityWeights(){
94   for (int i = 0 ; i < initial.length;i++){
95    double nominator = initial[i] / (double) N;
96    double denominator =1 -nominator;
97    double frac = nominator/denominator;
98    Wmajority[i] = Math.log10(frac);
99   }
100 }
101 public int [] calculate(int [] votes){
102  int [] results = new int[4];
103
104  results[0] = core(votes,Wplurality);
```

```
  results[1] = core(votes,Wsimple);
  results[2] = core(votes,Wrescaled);
  results[3] = core(votes,Wbestworst);

  return results;
 }
 private int core (int [] votes,double [] matrix){
  double [] results = {0,0,0};
  for (int i = 0 ; i < votes.length ; i++){
   results[votes[i]] += matrix[i];
  }
  double max = results[0];
  int index = 0;
  int i = 0;
  for (double d :results){
   if (d > max ){
    max = d;
    index = i;
   }
   i++;
  }
  return index;
 }
}
```

Κώδικας Β'.4: Υλοποίηση του Σχήματος Ψηφοφορίας

Οι λειτουργικότητες του Σχήματος Συνένωσης υλοποιούνται μέσω της κλάσης **Fuser**.

```
package Fusion;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.ArrayList;

public class Fuser {

 private FusionTNorm norm;
 private FusionTNorm [] normlist;
 private double [][] penalty;
 private final int N ;
 private ArrayList<Double []> vectorlist = new ArrayList<Double []>();
 public double [][] array;
 public double [][] Anorm;
 private double W;
 private double d;
 private double [] distributionForClasses;
 private double sum = 0;

 public Fuser(int classNum,double weight,double distance) {
  super();
  // N : how many classes are defined (3 in our problem)
  N = classNum;
  array  = new double[N+1][N+1];
  penalty = new double [N+1][N+1];
  distributionForClasses = new double[classNum];
  W = weight;
  d = distance;
```

```
 init();
}
// Initialise arrays : array(A),penalty
// Populate Tnorms 1 to 6
private  void init(){
 // create penalty matrix P
 createPenaltyMatrix();
 // initialise array A
 resetArray(array);
 // insert the 6 Tnorms
 normlist = new FusionTNorm[6];
 for (int i = 0 ; i < normlist.length ; i ++){
  String className = ''Fusion.T''+(i+1);
  try {
   normlist[i] = (FusionTNorm) Class.forName(className).newInstance();
  } catch (InstantiationException | IllegalAccessException
    | ClassNotFoundException e) {
   e.printStackTrace();
  }
 }
}
// Array A operation
public void resetArray(double[][] array2){
 for (int i = 0 ; i < array2.length ; i++){
  for (int j = 0 ; j < array2.length; j++){
   array2[i][j] = 0 ;
  }
 }
}
//Vector I operations
public void add (double[] b){
 Double [] fprob = new Double [b.length+1];
 for (int i = 0 ; i < b.length;i++){
  fprob[i] = b[i];
 }
 fprob[fprob.length-1] = 0.0;
 vectorlist.add(fprob);
}
// Initialize Penalty Matrix P
private void createPenaltyMatrix(){
 for (int i = 0 ; i < penalty.length -1;i++){
  for (int j = 0 ; j < penalty.length -1; j++){
   double temp = Math.pow(1 - W * Math.abs(i-j),d);
   BigDecimal bd = new BigDecimal(temp).setScale(3, RoundingMode.HALF_EVEN);
   temp = bd.doubleValue();
   penalty[i][j] = Math.max(0, temp);
  }
 }
 for (int i = 0 ; i < penalty.length;i++){
  penalty[i][penalty.length-1] = 1 ;
 }
 for (int i = 0 ; i < penalty.length ; i++){
  penalty[penalty.length-1][i] = 1 ;
 }
}
private double [][] normalize(){
 double [][] A = new double [N+1][N+1];
 sum= 0;
```

```
 for (int i = 0 ; i < array.length;i++){
  for (int j = 0 ; j < array.length ;j++){
   sum += array[i][j];
  }
 }
 for (int i = 0 ; i < array.length;i++){
  for (int j = 0 ; j < array.length ;j++){
   double temp = 0;
   if (sum != 0){
    temp = array[i][j] / sum;
   }
   A[i][j] =  temp;
  }
 }
 return A;
}
public double confidenceIndex(){
 double index = 0 ;
 for (int i = 0 ; i < Anorm.length ; i++){
  for (int j = 0 ; j < Anorm.length ; j++){
   index += Anorm[i][j]*penalty[i][j];
  }
 }
 return index;
}
private void fuse(Double [] v1 , Double[] v2){
 // Tnorm must be defined
 for (int i = 0 ; i < v1.length ; i ++){
  for (int j = 0 ; j < v2.length ; j++){
   array[i][j] = outerproduct(v1[i],v2[j]);
  }
 }
}
private Double [] extract(){
 Double [] vector = new Double[N+1];
 for (int i = 0 ; i < array.length -1 ; i++){
  vector[i] = array[i][i] + array[i][N]+array[N][i];
 }
 vector[vector.length-1] = array[N][N];
 return vector;
}
private double outerproduct(Double a, Double b) {
 return this.norm.calculate(a, b);
}
public void setTNorm(int i) {
 this.norm = this.normlist[i-1];
}
private Double [] fifo(){
 Double [] temp = this.vectorlist.get(0);
 this.vectorlist.remove(0);
 return temp;
}

public double combine(){
 Double [] v1 = fifo();
 Double [] v2 = fifo();
 if (this.norm == null){
  setTNorm(1);
```

```
145   }
146   fuse(v1,v2);
147   Double [] v = extract() ;
148   while (!this.vectorlist.isEmpty()){
149    Double [] tmp = fifo();
150    fuse(tmp,v);
151    v = extract();
152   }
153   Anorm = normalize();
154   Double [] vector = new Double[v.length];
155   double max = -1.0;
156   // vector[v.length] is always 0.0
157   int index = v.length;
158   for (int i = 0 ; i < v.length  ; i++){
159    double temp = 0;
160    if (sum != 0){
161     temp = v[i] / sum;
162    }
163    vector[i] = temp;
164    if (temp > max){
165     max = temp;
166     index = i;
167    }
168   }
169   this.vectorlist.removeAll(vectorlist);
170   return index;
171  }
172  public double[] getDistributionForClasses() {
173   return distributionForClasses;
174  }
175 }
```

Κώδικας Β′.5: Υλοποίηση του Σχήματος Συνένωσης

Τέλος, χάρη στη διαπροσωπεία **FusionTnorm** και το Strategy Design Pattern που εφαρμόζει η κλάση Fuser, υπάρχει η δυνατότητα να επιλέγεται δυναμικά μία από τις 6 Τριγωνικές Νόρμες που εξετάζουμε.

```
1  public interface FusionTNorm{
2   public double calculate(double a,double b);
3  }
4
5  public class T1 implements FusionTNorm {
6   public double calculate(double a, double b) {
7    return Math.max(0, a+b-1);
8   }
9  }
10
11 public class T2 implements FusionTNorm {
12  public double calculate(double a, double b) {
13   double t = Math.sqrt(a) + Math.sqrt(b)-1 ;
14   double result = Math.max(0,t);
15   return result*result;
16  }
17 }
18
```

```
19 public class T3 implements FusionTNorm {
20  public double calculate(double a, double b) {
21   return a*b;
22  }
23 }
24
25 public class T4 implements FusionTNorm {
26  public double calculate(double a, double b) {
27   double temp = (1/Math.sqrt(a)) + (1/Math.sqrt(b)) -1;
28   return 1/Math.sqrt(temp);
29  }
30 }
31
32 public class T5 implements FusionTNorm {
33  public double calculate(double a, double b) {
34   double temp = 1/a +1/b -1;
35   return 1/temp;
36  }
37 }
38
39 public class T6 implements FusionTNorm {
40  public double calculate(double a, double b) {
41   return Math.min(a, b);
42  }
43 }
```

Κώδικας Β΄.6: Ορισμός Διαπροσωπίας Τριγωνικής Νόρμας και υλοποίηση 6 διαφορετικών τύπων