Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης

Υβριδική Λεξιλογική Προσέγγιση της Μεθόδου Γράφων Λέξεων

Διπλωματικη Εργασια

ανθουσα καρκογλου

Επιβλέπουσα: Βαρβαρίγου Θ.
Καθηγητρια Ε.Μ.Π.

Αθήνα, Οκτώβρης 2017

Υβριδική Λεξιλογική Προσέγγιση της Μεθόδου Γράφων Λέξεων

Διπλωματικη Εργασια

ΑΝΘΟΥΣΑ ΚΑΡΚΟΓΛΟΥ

Επιβλέπουσα: Βαρβαρίγου Θ.
Καθηγητρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24η Οκτωβρίου 2017.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Βαρβαρίγου Θ.	Βαρβαρίγος Ε.	Ασκούνης Δ.
Καθηγητρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβρης 2017

(Υπογραφή)

.....

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ανάλυση Συναισθήματος συνιστά ο γνωστικός τομέας εξαγωγής και αναγνώρισης συναισθηματικής φόρτισης και άποψης σε κείμενο φυσικής γλώσσας. Η ανάπτυξη της επιστήμης της Ανάλυσης Συναισθήματος συμπίπτει με την εκρηκτική ανάπτυξη του Διαδικτύου και, επομένως, με την εμφάνιση της ανάγκης οργάνωσης και νοηματοδότησης όλης της πληροφορίας που προσπελάζεται από το Διαδίκτυο. Η Ανάλυση Συναισθήματος απαντάται σε πολλούς τομείς όπου βρίσκει πληθώρα εφαρμογών, όπως στον τομέα της Επιχειρηματικής Ευφυΐας και της Πολιτικής. Σε τεχνικά πλαίσια, η υλοποίοηση της εν λόγω επιστήμης μπορεί να πραγματοποιηθεί σε επίπεδο Βαθιάς Μάθησης (Deep Learning Algorithms), Λεξιλογικής Προσέγγισης (Lexicon Based Approach), είτε σε Υβριδικό επίπεδο που συνδυάζει τις δύο προηγούμενες προσεγγίσεις.

Στόχος της Διπλωματικής Εργασίας είναι η βελτιστοποίηση της Μεθόδου των Γράφων Λέξεων, η οποία συνιστά υλοποίηση βάσει Αλγορίθμου Βαθιάς Μάθησης. Σαφέστερα, επιχειρείται η δημιουργία μίας νέας Υβριδικής Μεθόδου, η οποία αξιοποιεί επιπρόσθετα τη συναισθηματική φόρτιση μεμονωμένων λέξεων σε συνδυασμό με την συντακτική θέση τους στο υπο μελέτη κείμενο.

Λέξεις Κλειδιά

Ανάλυση Συναισθήματος, Τεχνητή Νοημοσύνη, Αλγόριθμος Βαθιάς Μάθησης, Υβριδική Προσέγγιση, Λεξικό Συναισθήματος, Γράφοι Λέξεων, Λεξιλογική Προσέγγιση, Συντακτική Ανάλυση, Sentiment Analysis, Hybrid Method, SentiWordNet, Word Graphs, Deep Learning, Lexicon Based Approach

Abstract

Sentiment Analysis is the scientific sector of computational treatment of sentiment, opinion and subjectivity in text. The appearance and the explosive development of opinion-rich resources on the Internet have contributed to the development of Sentiment Analysis and its implementation in many areas, such as Business Intelligence and Politics. In technical terms, the implementation of Sentiment Analysis can be achieved with the assistance of Deep Learning Algorithms, Lexicon Based Approaches or Hybrid Methods that represent the combination of the first two methods.

This diploma thesis is concerned with the optimization of the method of Word Graphs. The actualization of the latter is based on the extension of the already existing Deep Learning Algorithm utilizing text features that stem from the Lexicon Based Approach of the subject. That is the sentiment charge of individual words, as well as, their position in the text and their syntactic relations and dependencies regarding the rest of the words that consist the text under examination.

Keywords

Deep Learning Algorithm, Sentiment Analysis, SentiWordNet, Word Graphs, Hybrid Method, Neural Networks, Lexicon Approach, Syntactic Analysis

Ευχαριστίες

Θα ήθελα να ευχαριστήσω, κατ΄ αρχάς, την καθηγήτρια Θ. Βαρβαρίγου για την ευκαρία ενασχόλησης και διερεύνησης ενός ιδιαίτερα ενδιαφέροντος θέματος Επιστημονικής Μελέτης που μου παρείχε.

Ακόμη, οφείλω να ευχαριστήσω τον διδάκτορα Βρεττό Μουλό για την άψογη συνεργασία μας, την υπομονή που επέδειξε και την καθοδήγησή του καθ΄όλη τη διάρκεια εκπόνησης της παρούσας Διπλωματικής Εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους και τους αγαπημένους μου ανθρώπους που ήταν δίπλα μου στην προσπάθεια αυτή. Το πλέον σημαίνον ευχαριστώ, ωστόσο, οφείλω στους γονείς μου και την αδερφή μου για την αγάπη και την στήριξή τους στα μπλε χρόνια του Πολυτεχνείου.

Αθήνα, Οκτώβρης 2017

Ανθούσα Καρκόγλου
Διπλωματούχος Ηλεκτρολόγος
Μηχανικός και Μηχανικός Η/Υ

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Ανάλυση Συναισθήματος συνιστά ο επιστημονικός τομέας εξόρυξης και ανάλυσης απόψεων, συναισθημάτων, εκτιμήσεων, αλλά και προδιαθέσεων της κοινής γνώμης όσον αφορά κάποια οντότητα, αξιοποιώντας υπολογιστικά εργαλεία.

1.1 Ανάλυση Συναισθήματος

Η Ανάλυση Συναισθήματος (Sentiment Analysis) αποτελεί μία προσέγγιση ανάλυσης και επεξεργασίας μη δομημένης πληροφορίας η οποία δεν μπορεί να επεξεργαστεί εύκολα από μία υπολογιστική μηχανή, αλλά προορίζεται για ανθρώπινη επικοινωνία. Η ανάπτυξη της επιστήμης της Ανάλυσης Συναισθήματος συμπίπτει με την εκρηκτική ανάπτυξη του Διαδικτύου (World Wide Web) και των εργαλείων που το ίδιο προσφέρει, ούτως ώστε να παρέχεται η δυνατότητα σε όλους να εκφέρουν δημόσια απόψεις μέσα από fora, ιστολόγια (blogs), μέσα κοινωνικής δικτύωσης (social networks and media), αλλά και πλατφόρμες δημοσιοποίησης περιεχομένου [6]. Η πρωτόγνωρη, σήμερα, καταγραφή ενός τεράστιου όγκου υποκειμενικών και συναισθηματικά φορτισμένων δεδομένων, καθώς και η σημαίνουσα θέση του ανθρώπινου συναισθήματος στην λήψη αποφάσεων, έχουν συντελέσει στην άνθιση της επιστήμης της Ανάλυσης Συναισθήματος.

Πιο συγκεκριμένα, η ανάπτυξη του γνωστικού τομέα της Ανάλυσης Συναισθήματος συνδέεται αλληλένδετα με την ραγδαία ανάπτυξη των Μέσων Κοινωνικής Δικτύωσης (Social Media) και την επιστημονική έρευνα πάνω σε αυτά. Για το λόγο αυτό, η εν λόγω επιστήμη έχει σημαίνουσα επιρροή όχι μόνο πάνω στον γνωστικό τομέα της Επεξεργασίας Φυσικής Γλώσσας (NLP, Natural Language Processing), αλλά, ακόμη, στις διοικητικές και τις πολιτικές επιστήμες, τα οικονομικά, αλλά και τις κοινωνικές επιστήμες, αφού όλες οι παραπάνω εξαρτώνται από την ανθρώπινη άποψη και δράση.

1.1.1 Εφαρμογές

Η ανάγκη διύλισης και ταξινόμησης της συναισθηματικής πολικότητας της κοινής γνώμης απαντάται, όπως αναφέρθηκε παραπάνω, σε πολλούς τομείς [7].

1. Επιχειρηματική Ευφυΐα (Business Intelligence)
   Ιδιαίτερο ρόλο κατέχει η επιστήμη της Συναισθηματικής Ανάλυσης στον επιχειρηματικό χώρο. Καταρχάς, η Συναισθηματική Ανάλυση παρέχει τα εργαλεία καταγραφής της φωνής των καταναλωτών και ανάλυσης της φήμης μίας εταιρείας (Brand Reputation). Ακόμη, η Συναισθηματικη Ανάλυση μπορεί να αξιοποιηθεί στη διαδικτυακή διαφήμιση, είτε εστιάζοντας στην συμπεριφορική διαφήμιση, δηλαδή τη διαφήμιση που βασίζεται στην πλοήγηση του χρήστη στο Διαδίκτυο, είτε στην ανταγωνιστική διαφήμιση βάσει εκφρασμένης δυσαρέσκειας του καταναλωτή.

2. Πολιτική
   Η Συναισθηματική Ανάλυση στην Πολιτική μπορεί να συντελέσει στην πρόβλεψη ενός εκλογικού αποτελέσματος, αλλά και την αποσαφήνιση θέσεων και δημοφιλίας πολιτικών προσώπων. Ακόμη, δίνεται η δυνατότητα παλμογράφησης της δημόσιας γνώμης σχετικά με τρέχοντες πολιτικές και προτεινόμενα νομοσχέδια.

3. Δημόσιες Δράσεις
   Σε παγκόσμιο επίπεδο, η Συναισθηματική Ανάλυση προσφέρει τη δυνατότητα παρακολούθησης ραγδαία εξελισσόμενων γεγονότων, όπως φυσικών καταστροφών και κοινωνικοπολιτικών δρώμενων.Ένας αναπτυσσόμενος τομέας εφαρμογής της Ανάλυσης Συναισθήματος είναι τα Ευφυή Συστήματα Μεταφοράς (Intelligent Transportation Systems,ITSs), τα οποία αποτελούν εφαρμογές που στοχεύουν στην παροχή πρωτοποριακών υπηρεσιών για την βελτιστοποίηση χρήσης των Μέσων Μεταφοράς και των Οδικών Δικτύων.

4. Χρηματοοικονομική Επιστήμη
   Στον τομέα της Χρηματοοικονομικής Επιστήμης η Ανάλυση Συναισθήματος προσφέρει τη δυνατότητα αναγνώρισης της διάθεσης της αγοράς ως προς μία εταιρεία και τη μετοχή της.

1.1.2 Μέθοδοι Προσέγγισης

Η υπάρχουσα πρόοδος σχετικά με την Ανάλυση Συναισθήματος μπορεί να ταξινομηθεί βάσει διαφορετικών σημείων αναφοράς, όπως αναφορικά της τεχνολογίας που χρησιμοποιείται, του επίπεδο λεπτομέρειας της ανάλυσης, της θεματολογία του κειμένου κ.ο.κ.

Σε τεχνικό επίπεδο, είναι σκόπιμο να αναφερθούν η μέθοδος του machine learning, οι lexicon-based και hybrid μέθοδοι προσέγγισης.

Machine Learning

Η Machine Learning μέθοδος αξιοποιεί τεχνικές ταξινόμησης προκειμένου να αποφανθεί για το εξεταζόμενο κείμενο. Κάνει χρήση δύο συνόλων κειμένων, ένα σύνολο κειμένων εκπαίδευσης (training set) και ένα σύνολο κειμένων για έλεγχο(test set). Το σύνολο κειμένων

εκπαίδευσης χρησιμοποιείται για τον εντοπισμό των ετερόκλητων χαρακτηριστικών των κειμένων, ενώ το σύνολο ελέγχου για την αξιολόγηση της απόδοσης του ταξινομητή (classifier).

Η παρούσα προσέγγιση περιλαμβάνει τεχνικές εξόρυξης συναισθήματος, όπως τα Μπεϋζιανά Δίκτυα (Bayesian Networks), την Απλή Μπεϋζιανή Ταξινόμηση (Naive Bayes Classification), την Μέγιστοποίηση Εντροπίας (Maximum Entropy), τα Νευρωνικά Δίκτυα (Neural Networks). Οι παραπάνω τεχνικές στηρίζονται κυρίως στα εξής χαρακτηριστικά κειμένου:

1. Παρουσία και Συχνότητα όρων ή συστάδων όρων, που αναπαρίστονται από unigrams και n-grams αντίστοιχα.

2. Νοηματική πληροφορία βάσει της αναγνώρισης μερών του λόγου μέσα στο κείμενο.

3. Άρνηση, η οποία είναι δυνατό να αντιστρέψει την συναισθηματική πολικότητα φράσεων και λέξεων.

Το σημαντικότερο πλεονέκτημα της εν λόγω μεθόδου είναι η δυνατότητα δημιουργίας και εκπαίδευσης αλγορίθμων με εξειδίκευση σε συγκεκριμένο τύπο κειμένων και τρόπων έκφρασης, εστιάζοντας περισσότερο στοχευμένα στον σκοπό της ανάλυσης. Εντούτοις, δεν υπάρχει η δυνατότητα ανάλυσης νέων δεδομένων, καθώς είναι απαραίτητη η ύπαρξη κατάλληλων συνόλων δεδομένων εκπαίδευσης, κάτι το οποίο μπορεί να αποβεί δαπανηρό ή ακόμα και μη επιτεύξιμο. Ακόμα και στην περίπτωση, ωστόσο, που υπάρχει ένα τέτοιο σύνολο δεδομένων, η δυνατότητα προσαρμογής ενός εκπαιδευμένου αλγορίθμου σε διαφορετικές θεματικές κατηγορίες και δομές κειμένων είναι αρκετά περιορισμένη.

Lexicon-Based

Η Lexicon-Based Προσέγγιση χρησιμοποιεί Λεξικά Συναισθήματος τα οποία περιέχουν λέξεις που εκφέρουν άποψη, προκειμένου να ψηλαφίσει την συναισθηματική πολικότητα ενός κειμένου.Προκειμένου να κατασκευαστεί ένα τέτοιο λεξικό, υπάρχουν τρεις διαφορετικοί τρόποι. Πέρα από την χειροκίνητη κατασκευή του, είναι δυνατό να κατασκευαστεί με λέξεις που εκφέρουν άποψη βάσει ενός συνόλου κειμένων με γνωστή συναισθηματική φόρτιση. Ακόμη, είναι δυνατό, μετά την συγκέντρωση ενός μικρού αριθμού συναισθηματικά φορτισμένων λέξεων, η αναζήτηση των συνωνύμων και αντώνυμων τους στο WordNet λεξικό.

Το πλεονέκτημα των Lexicon-Based μεθόδων είναι ότι τα Λεξικά συναισθηματικά φορτισμένων λέξεων καλύπτουν ένα ευρύ φάσμα θεματολογίας. Ωστόσο, η πεπερασμένη έκταση των Λεξικών αυτών συχνά δημιουργεί προβλήματα στην αναγνώριση συναισθήματος σε δυναμικά περιβάλλοντα, όπως είναι τα Μέσα Κοινωνικής Δικτύωσης(Social Media), στα οποία χρησιμοποιείται αργκό και νεολογισμοί. Επιπρόσθετα, τα Λεξικά Συναισθήματος(Sentiment Lexicons) αναθέτουν, συνήθως, σε καθεμία από τις λέξεις σε ένα κείμενο μία σταθερά συναισθηματικής φόρτισης αθροίζοντας σε έναν αριθμό ο οποίος δεν είναι αντιπροσωπευτικός του τρόπου με τον οποίο αυτές έχουν χρησιμοποιηθεί.

Hybrid

Η Hybrid Προσέγγιση αποτελεί τον συνδυασμό των δύο παραπάνω, και έχει την δυνατότητα βελτίωσης της απόδοσης και της ακρίβειας της συναισθηματικής ταξινόμησης κειμένων. Για το λόγο αυτό, τα σημαντικότερα πλεονεκτήματα της εν λόγω προσέγγισης είναι ο συγκερασμός των άλλων δύο μεθόδων με στόχο τον εντοπισμό και τη μέτρηση του συναισθήματος σε ένα αφαιρετικό επίπεδο, αλλά και τη μείωση της ευαισθησίας σε θεματικές αλλαγές. Από την άλλη πλευρά, το μειονέκτημα της είναι ο μεγάλος βαθμός επιρροής του αποτελέσματος από τον θόρυβο που μπορεί να υπάρχει μέσα σε ένα κείμενο.

Μία εναλλακτική ταξινόμηση στηρίζεται στη δομή του εξεταζόμενου κειμένου. Πιο συγκεκριμένα, μπορούν να διακριθούν τα εξής επίπεδα ανάλυσης:

- Επίπεδο Εγγράφου (Document Level): Στο παρόν επίπεδο ανάλυσης στόχος είναι η σκιαγράφηση του συναισθηματικού προσανατολισμού του εξεταζόμενου κειμένου σε ολόκληρη την έκτασή του. Σε επίπεδο εγγράφου είναι δυνατόν να αναγνωριστεί η άποψη ή η προδιάθεση του συγγραφέα απέναντι στο προϊόν ή την υπηρεσία στα οποία αναφέρεται το κείμενο.

- Επίπεδο πρότασης (Sentence Level): Εδώ, η Ανάλυση Συναισθήματος πραγματοποιείται μεμονωμένα για καθεμία από τις προτάσεις του εξεταζόμενου κειμένου.

- Επίπεδο χαρακτηριστικού (Entity/Aspect Level): Στο επίπεδο αυτό, η Συναισθηματική Ανάλυση στοχεύει να εντοπίσει την συναισθηματική φόρτιση που φέρει ο χρήστης για καθένα από τα διαφορετικά χαρακτηριστικά μίας οντότητας. Πιο συγκεκριμένα, δεν δίνεται έμφαση στα λεκτικά κατασκευάσματα, αλλά στην άποψη αυτή καθεαυτή. Βασίζεται στην ιδέα ότι μία άποψη συνίσταται στο συναίσθημα (θετικό ή αρνητικό) και στην οντότητα την οποία στοχεύει.

1.2 Αντικείμενο της Διπλωματικής

Η παρούσα Διπλωματική Εργασία επιχειρεί την βελτιστοποίηση ενός αλγορίθμου βαθιάς μάθησης (Deep Learning Algorithm), ο οποίος εφαρμόζεται για την Συνασθηματική Ανάλυση κριτικών ταινιών της βάσης IMDb. Ο αρχικός αλγόριθμος αναγνωρίζει μία κριτική ως ένα έγγραφο (Document Level Analysis) μίας παραγράφου. Ως είσοδο, λαμβάνει τα αποτελέσματα μίας μεθόδου εξαγωγής χαρακτηριστικών κειμένων με σκοπό την κατηγοριοποίηση της κριτικής συναισθηματικά [10].

Επιχειρείται η προσέγγιση του παραπάνω αλγορίθμου βαθιάς μάθησης υβριδικά. Διατηρώντας την στατιστική μορφή της μεθόδου εξαγωγής χαρακτηριστικών των κριτικών, πραγματοποιείται απόπειρα προσθήκης επιπρόσθετων χαρακτηριστικών εισόδου του αλγορίθμου βάσει της Lexicon-Based Μεθόδου. Σαφέστερα, επιχειρείται η αξιοποίηση της πληροφορίας που έγκειται στην Συντακτική Δομή των κειμένων, όπως επίσης η συναισθηματική φόρτιση των λέξεων που επιλέγονται σε καίριες συντακτικές θέσεις.

Ως συμβολή της παρούσας Διπλωματικής Εργασίας στοχεύεται η ακριβέστερη αξιολόγηση και συναισθηματική ταξινόμηση του κάθε κειμένου, καθώς και η μείωση του θορύβου της πληροφορίας που παρέχεται ως είσοδος στο εν λόγω νευρωνικό δίκτυο.

1.3 Δομή της Διπλωματικής

- Κεφάλαιο 2
  Στο Κεφάλαιο 2 της παρούσας Διπλωματικής παρουσιάζεται το Θεωρητικό Υπόβαθρο της Ανάλυσης Συναισθήματος. Σαφέστερα, παρουσιάζεται η βασική διάρθρωση ενός συστήματος Ανάλυσης Συναισθήματος και, στη συνέχεια, παρατίθενται και αναλύονται μέθοδοι Εξαγωγής Χαρακτηριστικών Κειμένων και Μέθοδοι Ταξινόμησης Κειμένων, οι οποίες συνιστούν συνιστώσες ενός τέτοιου συστήματος.

- Κεφάλαιο 3
  Στο κεφάλαιο 3 προσεγγίζονται με μεγαλύτερη λεπτομέρεια οι επιμέρους μέθοδοι που έχουν επιλεχθεί για τη σύσταση του μοντέλου Ανάλυσης Συναισθήματος που εξετάζεται στην παρούσα Εργασία. Παρουσιάζονται, δηλαδή, εκτενέστερα οι επιλεγμένες μέθοδοι Εξαγωγής Χαρακτηριστικών και Ταξινόμησης Κειμένων.

- Κεφάλαιο 4
  Στο Κεφάλαιο 4 παρατίθεται συνοπτικά ο υπάρχων κώδικας πάνω στον οποίο στηρίζεται η νέα υβριδική μέθοδος που παρουσιάζεται εδώ, όπως και οι τροποποιήσεις που πραγματοποιήθηκαν σε αυτόν ούτως ώστε η ίδια να επιτευχθεί.

- Κεφάλαιο 5
  Στο Κεφάλαιο 5 αναλύονται τα αποτελέσματα προσομοιώσεων της νέας υβριδικής μεθόδου που πραγματοποιήθηκαν στα πλαίσια της παρούσας Εργασίας. Αναφέρονται οι πολλαπλές παραμετροποιήσεις του νέου μοντέλου, ούτως ώστε να επιτευχθεί η μέγιστη ακρίβεια με το μικρότερο υπολογιστικό κόστος, ενώ, ταυτόχρονα, παρατίθεται η σύγκριση της νέας μεθόδου με δύο υπάρχουσες γνωστές και δημοφιλείς μεθόδους εξαγωγής χαρακτηριστικών κειμένου.

- Κεφάλαιο 6
  Στο Κεφάλαιο 6 της Διπλωματικής Εργασίας παρουσιάζονται τα συμπεράσματα που προκύπτουν από την εκτέλεση των προσομοιώσεων, οι οποίες παρουσιάζονται στο Κεφάλαιο 5. Εδώ αποφαίνεται η συμβολή ή όχι της νέας υβριδικής μεθόδου, ενώ προτείνονται ταυτόχρονα μελλοντικές επεκτάσεις της εν λόγω προσέγγισης.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Η Ανάλυση Συναισθήματος, είναι σύνηθες, να κατηγοριοποιείται στην βιβλιογραφία ως ένα δυϊκό πρόβλημα ταξινόμησης. Στην ουσία, αντιμετωπίζεται ως ένα πρόβλημα μοναδικής ταξινόμησης αφού κάθε έγγραφο εντάσσεται σε μοναδική συναισθηματική πολικότητα. Συνεπώς, η ουσία της Ανάλυσης Συναισθήματος έγκειται στην απόδοση θετικής ή αρνητικής πολικότητας σε ένα κείμενο [4]. Εγγύτερα στην πραγματικότητα, αλλά λιγότερο συχνή στον γνωστικό τομέα της Συναισθηματικής Ανάλυσης, είναι η τριχοτόμηση των πιθανών συναισθηματικών φορτίσεων ενός κειμένου σε θετική, αρνητική και ουδέτερη φόρτιση. Στην παρούσα προσέγγιση, η συναισθηματική αναγνώριση θεωρείται δυϊκό πρόβλημα.

2.1 Ανάλυση Συναισθήματος

Η περιοχή έρευνας της Ανάλυσης Συναισθήματος (Sentiment Analysis), η οποία συχνά αναφερέται και με τον όρο Opinion Mining, καλύπτει την υπολογιστική αντιμετώπιση των συναισθημάτων που εκφράζονται σε ένα κείμενο. Η Ανάλυση Συναισθήματος διαθέτει τέσσερις συνιστώσες, την Αναγνώριση Απόψεων, την Εξαγωγή Χαρακτηριστικών, την Ταξινόμηση Συναισθήματος και την Απεικόνιση και Σύνοψη των Αποτελεσμάτων.

Η Αναγνώριση Απόψεων στοχεύει στο να αποφανθεί εαν η πρόταση που βρίσκεται υπό ανάλυση περιέχει ή όχι έκφραση μίας άποψης. Για το λόγο αυτό, είναι απαραίτητο να διακριθούν οι απόψεις από τις απλές δηλώσεις. Μόνο οι προτάσεις που περιέχουν απόψεις συλλέγονται για περαιτέρω επεξεργασία. Όπως μπορεί να παρατηρηθεί, απόψεις και συναισθήματα αναφέρονται πάντα σε μία οντότητα (ένα αντικείμενο, μία υπηρεσία, ένα πρόσωπο) είτε ρητά είτε όχι. Η Εξαγωγή Χαρακτηριστικών στοχεύει στην αναγνώριση των οντοτήτων στις οποίες αναφέρεται το κείμενο. Η Ταξινόμηση στοχεύει στον καθορισμό της πολικότητας των απόψεων όσον αφορά τα χαρακτηριστικά που σχολιάζονται. Τέλος, παρουσιάζεται η Σύνοψη της Ανάλυσης, ούτως ώστε να υποστηριχθεί η αντίστοιχη διαδικασία λήψης αποφάσεων. Συνήθως, εδώ, παρουσιάζεται το πλήθος των θετικών και των αρνητικών εκτιμήσεων του κάθε χαρακτηριστικού. Κάποια συστήματα παρουσιάζουν επίσης γραφήματα προκειμένου να διευκρινιστεί η περιληπτική αυτή απεικόνιση. Τα παραπάνω στάδια, είναι σημαντικό να σημειωθεί ότι, δεν είναι απαραίτητο να εκτελεσθούν σειριακά. Στην πράξη, κάποιες προσεγγίσεις υλοποιούν περισσότερα του ενός

στάδια ταυτόχρονα. Παραδείγματος χάριν, είναι δυνατό να διακριθεί ένα κείμενο ως θετικά ή αρνητικά υποκειμενικό ή ως αντικειμενικό που δεν εκφέρει άποψη, εκτελώντας ταυτόχρονα τα στάδια της Αναγνώρισης και της Ταξινόμησης. Ωστόσο, η πλειονότητα των εργασιών στην Ανάλυση Συναισθήματος συνήθως εστιάζει μόνο σε ένα ή δύο από τα παραπάνω στάδια [19].

Στην παρούσα Διπλωματική Εργασία κρίνεται σημαντικό να παρουσιαστούν εκτενέστερα τα στάδια της Μεθόδου Εξαγωγής Χαρακτηριστικών και της Ταξινόμησης Συναισθήματος.

2.2 Μέθοδοι Εξαγωγής Χαρακτηριστικών (Feature Extraction Methods)

Θεωρώντας, όπως αναφέρθηκε και παραπάνω, την Ανάλυση Συναισθήματος ως ένα πρόβλημα Συναισθηματικής Ταξινόμησης, το πρώτο βήμα στην αντιμετώπισή του αποτελεί η εξαγωγή και η διαλογή χαρακτηριστικών κειμένου τα οποία θα δοθούν σαν είσοδο στο επόμενο στάδιο [15]. Κάποια από αυτά παρατίθενται παρακάτω:

- Παρουσία και Συχνότητα Όρων
  Στα εν λόγω χαρακτηριστικά συμπεριλαμβάνονται αυτόνομες λέξεις ή n-gram γραφήματα λέξεων, καθώς και η συχνότητα εμφάνισής τους. Στις λέξεις αποδίδεται δυϊκό βάρος (μηδέν στην περίπτωση που εμφανίζεται η λέξη ή ένα διαφορετικά) ή βάρος σχετικής συχνότητας εμφάνισης προκειμένου να προσδιοριστεί η σχετική σημασία των χαρακτηριστικών.

- Μέρη του Λόγου (Part Of Speech, POS)
  Συγκεκριμένα Μέρη του Λόγου θεωρείται ότι εμπεριέχουν συναίσθημα. Η εστίαση εντοπίζεται κυρίως στα επίθετα, καθώς συνιστούν σημαίνοντες δείκτες άποψης.

- Λέξεις και Εκφράσεις που εκφέρουν άποψη (Opinion words and phrases)
  Τέτοιες λέξεις χρησιμοποιούνται συχνά για να εκφράσουν απόψεις συμπεριλαμβανομένων των διπόλων θετικό/αρνητικό ή αρέσκεια/δυσαρέσκεια. Από την άλλη πλευρά, κάποιες φράσεις εκφράζουν απόψεις χωρίς να χρησιμοποιούν λέξεις που εκφέρουν άποψη.

- Αρνήσεις
  Η παρουσία αρνητικών λέξεων μπορεί να αντιστρέψει το πρόσημο μίας άποψης, όπως λόγου χάρη η φράση ΄όχι καλός΄ ισοδυναμεί με ΄κακός΄.

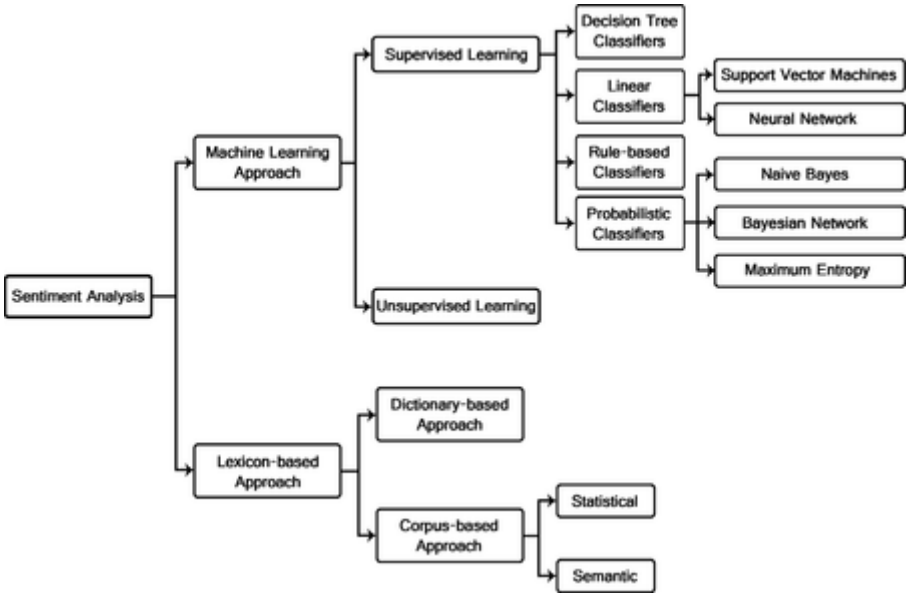
Οι Μέθοδοι Εξαγωγής Χαρακτηριστικών μπορούν να διακριθούν σε Lexicon-Based Μεθόδους, οι οποίες χρειάζονται ανθρώπινη παρέμβαση, και σε Στατιστικές Μεθόδους που είναι αυτοματοποιημένες και χρησιμοποιούνται ευρύτερα. Οι Lexicon-Based Μέθοδοι συνήθως στηρίζονται σε ένα σύνολο λέξεων που λειτουργεί ως Φύτρο (seed). Στη συνέχεια, το εν λόγω σύνολο επεκτείνεται μέσω συνωνύμων είτε μέσω Διαδικτυακών πόρων. Αντίθετα, οι Στατιστικές Μέθοδοι είναι πλήρως αυτοματοποιημένες.

Οι Μέθοδοι Εξαγωγής Χαρακτηριστικών αντιμετωπίζουν τα έγγραφα είτε ως ένα σύνολο λέξεων (Bag Of Words, BOWs), είτε ως μία αλληλουχία λέξεων. Η Μέθοδος Bag Of

Words χρησιμοποιείται συχνότερα κυρίως λόγω της απλότητας της γιια το μετέπειτα στάδιο της Ταξινόμησης. Εν γένει, το πιο συχνό βήμα Εξαγωγής Χαρακτηριστικών είναι η αφαίρεση των λέξεων-παύσης (stop-words) και η ανεύρεση των λέξεων-ριζών (stemming). Οι τρεις από τις ευρύτερα χρησιμοποιούμενες Στατιστικές Μεθόδους Εξαγωγής Χαρακτηριστικών είναι τα μέτρα Point-wise Mutual Information (PMI) και χ^2, καθώς και η Λανθάνουσα Σημασιολογική Δεικτοδότηση (Latent Semantic Indexing, LSI), ενώ δεν θεωρείται σκόπιμο να παρουσιαστούν εκτενώς στην παρούσα εργασία.

2.3 Μέθοδοι Ταξινόμησης (Classification Methods)

Η Ταξινόμηση Κειμένων μπορεί να προσεγγιστεί με πολλαπλούς τρόπους. Ο εκάστοτε τρόπος θα ενταχθεί, κατάρχάς, είτε στον τομέα της Μηχανικής Μάθησης είτε στον τομέα της Λεξιλογικής Προσέγγισης. Και στις δύο περιπτώσεις έγκειται περαιτέρω ταξινόμηση σε υποτομείς [15], όπως φαίνεται και στο Σχήμα 2.1.



Σχήμα 2.1: Μέθοδοι Ταξινόμησης

2.3.1 Machine Learning (Μηχανική Μάθηση)

Η Μηχανική Μάθηση στην Ανάλυση Συναισθήματος στηρίζεται στους Αλγορίθμους Βαθιάς Μάθησης (Deep Learning Algorithms) ώστε να προσεγγίσει το ζήτημα ως ένα τυπικό πρόβλημα κατηγοριοποίησης κειμένου, αξιοποιώντας συντακτικά ή/και λεξικολογικά χαρακτηριστικά. Σε ένα τέτοιο πρόβλημα, καθένα από τα αρχεία εκπαίδευσης αντιστοιχεί σε μία κατηγορία συναισθήματος. Το μοντέλο ταξινόμησης του αλγορίθμου σχετίζεται άμεσα με τα χαρακτηριστικά που δίνονται στον ίδιο ως είσοδο και τα οποία προκύπτουν από το στάδιο της Εξαγωγής Χαρακτηριστικών, ούτως ώστε, δοθέντος ενός κειμένου, το μοντέλο να βρίσκεται σε θέση να προβλέψει την κλάση στην οποία αυτό ανήκει. Η ταξινόμηση ενός κειμένου μπορεί

να διακριθεί σε απόλυτη (Hard Classification), όπου μοναδική κλάση αντιστοιχίζεται σε ένα κείμενο εισόδου, και σε πιθανοτική (Soft Classification), όπου σε κείμενο εισόδου αποδίδονται τιμές πιθανότητας κλάσεων στις μπορεί να ανήκει.

Supervised Learning (Επιβλεπόμενη Μάθηση)

Επιβλεπόμενη μηχανική μάθηση ονομάζεται η διαδικασία συμπερασμού μίας συνάρτησης από επιβλεπόμενα δεδομένα εκπαίδευσης. Τα δεδομένα εκπαίδευσης αποτελούνται από ένα σύνολο στιγμιοτύπων εκπαίδευσης. Κάθε στιγμιότυπο συνίσταται σε ζεύγη ενός δεδομένου εισόδου (συνήθως ένα διάνυσμα γνωρισμάτων) και την επιθυμητή τιμή εξόδου της συνάρτησης, η οποία ονομάζεται σήμα επίβλεψης (Supervisory Signal). Ένας αλγόριθμος επιβλεπόμενης μάθησης αναλύει τα δεδομένα εκπαίδευσης και συμπεραίνει μία συνάρτηση η οποία είτε ονομάζεται Ταξινομητής (Classifier), εαν το πεδίο τιμών της είναι διακριτό, είτε Συνάρτηση Παλινδρόμησης (Regression Function), εαν το πεδίο τιμών της είναι συνεχές. Στην συναισθηματική ανάλυση το πεδίο τιμών της ζητούμενης συνάρτησης είναι διακριτό και, επομένως, θεωρείται σκόπιμο να εστιαστεί η προσοχή της παρούσας εργασίας σε αυτού του είδους τις συναρτήσεις ταξινόμησης. Η συνάρτηση που συμπεραίνεται από τον αλγόριθμο πρέπει να προβλέπει το σωστό αποτέλεσμα για οποιοδήποτε έγκυρο αντικείμενο εισόδου. Για το λόγο αυτό, η εν λόγω συνάρτηση στοχεύει στην γενίκευση από τα δεδομένα εκπαίδευσης σε άγνωστες περιπτώσεις εισόδου βάσει μίας λογικής συρροής.

Οι μέθοδοι επιβλεπόμενης μάθησης εξαρτώνται από την ύπαρξη και τη μορφή των ταξινομημένων δεδομένων εκπαίδευσης του αλγορίθμου. Στη βιβλιογραφία υπάρχουν ποικίλοι αλγόριθμοι ταξινόμησης επιβλεπόμενης μάθησης. Στη συνέχεια παρατίθονται οι πιο δημοφιλείς [15].

- Πιθανοτικοί Ταξινομητές (Probabilistic classifiers)
  Οι Πιθανοτικοί Ταξινομητές αξιοποιούν ανάμεικτα μοντέλα προς ταξινόμηση. Ένα ανάμεικτο μοντέλο υποθέτει ότι κάθε κλάση είναι συστατικό στοιχείο του, και ότι κάθε τέτοιο συστατικό στοιχείο είναι ένα μοντέλο-γεννήτρια της πιθανότητας να προσπελαστεί στοιχείο εισόδου που ανήκει στην αντίστοιχη κατηγορία ταξινόμησης. Αυτού του είδους οι ταξινομητές ονομάζονται και Ταξινομητές-Γεννήτριες (Generative Classifiers). Οι τρεις διασημότεροι πιθανοτικοί ταξινομητές είναι οι ακόλουθοι:

  - Απλός Ταξινομητής Bayes (Naive Bayes Classifier)
    Ο εν λόγω Ταξινομητής αποτελεί τον απλούστερο ταξινομητή και απαντάται πιο συχνά από τους υπόλοιπους . Το μοντέλο ταξινόμησης Naive Bayes υπολογίζει την a posteriori πιθανότητα μίας κλάσης, λαμβάνοντας υπόψιν την κατανομή των λέξεων σε ένα έγγραφο. Για το λόγο αυτό, αξιοποιείται ο τύπος πιθανότητας του Bayes, ώστε να προβλεφθεί η πιθανότητα ενός δεδομένου να ανήκει σε κάποια κλάση.

$$P(label|features) = \frac{P(label) * P(features|label)}{P(features)}$$

,όπου $P(label)$ είναι η a priori πιθανότητα μίας κλάσης με διακριτικό label, $P(feautures|label)$ η a priori πιθανότητα ενός δοσμένου συνόλου χαρακτηριστικών (features) να εντάσσονται στην κλάση με διακριτό label και $P(feautures)$ η a priori πιθανότητα εμφάνισης ενός συνόλου χαρακτηριστικών. Δεδομένης της απλοϊκής (Naive) υπόθεσης ότι τα χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους, η παραπάνω εξίσωση θα μπορούσε να γραφεί ως εξής:

$$P(label|features) = \frac{P(label) * P(f1|label) * ... * P(f1|label)}{P(features)}$$

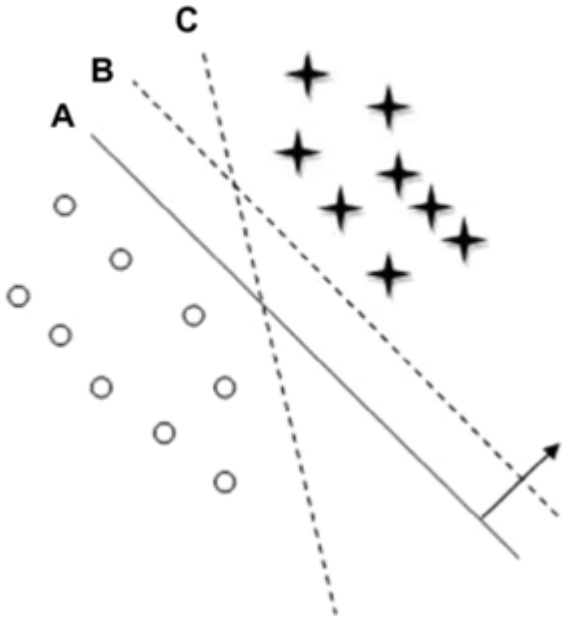
- Δίκτυο Bayes (Bayesian Network, BN)
  Η βασική υπόθεση του Ταξινομητή του Απλού Ταξινομητή Bayes είναι η ανεξαρτησία των χαρακτηριστικών (features). Στον αντίποδα αυτής της υπόθεσης βρίσκεται η υπόθεση ότι όλα τα χαρακτηριστικά εξαρτώνται μεταξύ τους. Η τελευταία οδηγεί στην δημιουργία ενός Δικτύου Bayes το οποίο αποτελεί ένα κατευθυνόμενο ακυκλικό γράφο του οποίου οι κόμβοι αναπαριστούν τυχαίες μεταβλητές, ενώ οι ακμές αναπαριστούν υποθετικές εξαρτήσεις. Ένα τέτοιο δίκτυο θεωρείται ως ένα πλήρες μοντέλο για τις τυχαίες μεταβλητές και τις εξαρτήσεις τους. Επομένως, για το εν λόγω μοντέλο υπολογίζεται η κοινή κατανομή (Joint Probability Distribution) όλων των τυχαίων μεταβλητών. Στον τομέα του Text Mining η υπολογιστική πολυπλοκότητα του Δικτύου Bayes είναι ακριβή και, για το λόγο αυτό, δεν χρησιμοποιείται συχνά.
- Ταξινομητής Μέγιστης Εντροπίας (Maximum Entropy Classifier)
  Ο Ταξινομητής Μέγιστης Εντροπίας μετατρέπει τα σύνολα χαρακτηριστικών αναγνωρισμένης κλάσης σε διανύσματα χρησιμοποιώντας κωδικοποίηση. Το κωδικοποιημένο διάνυσμα αξοποιείται για τον υπολογισμό των βαρών κάθε χαρακτηριστικού. Τα βάρη αυτά συνυπολογίζονται στη συνέχεια ούτως ώστε να προβλεφθεί η κατάλληλη κλάση στην οποία είναι πιθανότερο να ανήκει ένα σύνολο χαρακτηριστικών άγνωστης κλάσης. Ο Ταξινομητής παραμετροποιείται βάσει ενός συνόλου βαρών, το οποίο απεικονίζει τον συνδυασμό των κοινών χαρακτηριστικών που προκύπτει από την κωδικοποίηση ενός συνόλου χαρακτηριστικών εκπαίδευσης. Σαφέστερα, η κωδικοποιήση αντιστοιχεί κάθε ζεύγος συνόλου χαρακτηριστικών και κλάσης με το διακριτικό label σε ένα διάνυσμα. Στη συνέχεια, η πιθανότητα κάθε κλάσης με το διακριτικό label υπολογίζεται βάσει της ακόλουθης συνάρτησης:

$$P(fs|label) = \frac{dotprod(weights, encode(fs, label))}{sum(dotprod(weights, encode(fs, label))forlinlabels)}$$

- Γραμμικοί Ταξινομητές (Linear Classifiers)
  Έστω διάνυσμα $\overline{\mathbf{X}} = \{x_1....x_n\}$, το οποίο αναπαριστά την κανονικοποιημένη συχνότητα όρων σε ένα έγγραφο, $\overline{\mathbf{A}} = \{a_1....a_n\}$, το οποίο αποτελεί ένα διάνυσμα των γραμμικών συντελεστών με διάσταση ίση με αυτή του χώρου των χαρακτηριστικών και ένα μονόμετρο b. Το αποτέλεσμα του Γραμμικού Ταξινομητή ορίζεται ως $\overline{\mathbf{X}} * \overline{\mathbf{A}} + b$. Μεταξύ των

πολλών ειδών Γραμμικών Ταξινομητών, πλέον σημαίνοντα είναι τα Διανύσματα Υποστήριξης Μηχανής (Support Vector Machines (SVM)), τα οποία συνιστούν μία μορφή ταξινομητών που επιχειρούν να καθορίσουν επιτυχώς γραμμικούς διαχωριστές ανάμεσα στις διαφορετικές κλάσεις. Οι πλέον γνωστοί Γραμμικοί Ταξινομητές παρατίθενται παρακάτω:

– Διάνυσμα Υποστήριξης Μηχανής Support Vector Machines Classifiers (SVM)
Η βασική αρχή των Διανυσμάτων Υποστήριξης Μηχανής είναι ο προσδιορισμός των γραμμικών διαχωριστών στον χώρο αναζήτησης, οι οποίοι τον διαχωρίζουν βέλτιστα σε διαφορετικές κλάσης ταξινόμησης. Στο σχήμα 2.2 υπάρχουν δύο κλάσεις x, ο και τρία υπερεπίπεδα, A, B και C. Το υπερεπίπεδο A επιτυγχάνει τον βέλτιστο διαμερισμό του χώρου σε κλάσεις, καθώς η ευκλείδια απόσταση από οποιοδήποτε σημείο είναι η μέγιστη και, επομένως, ο διαχωρισμός αυτός αναπαριστά το μέγιστο περιθώριο διαχωρισμού.



Σχήμα 2.2: Διανύσματα Υποστήριξης Μηχανής (SVM) σε πρόβλημα ταξινόμησης

Τα Διανύσματα Υποστήριξης Μηχανής ταιριάζουν ιδιαίτερα στην επεξεργασία Δεδομένων Κειμένου, και αυτό λόγω της μικρής πυκνότητας περιεχομένου του οποίου τα χαρακτηριστικά τείνουν να συσχετίζονται μεταξύ τους σε γραμμικά διαχωρίσιμες κατηγορίες. Ο Ταξινομητής αυτός δύναται να κατασκευάσει μέσα στον αρχικό χώρο αναζήτησης μία μη-γραμμική επιφάνεια απόφασης αντιστοιχίζοντας τα στιγμιότυπα δεδομένων σε έναν ενδότερα παράγωγο χώρο όπου οι κλάσεις μπορούν να διαχωριστούν γραμμικά με ένα υπερεπίπεδο.

– Νευρωνικά Δίκτυα (Neural Network, NN)
Ένα Νευρωνικό Δίκτυο συνίσταται σε πολλαπλούς νευρώνες, όπου νευρώνας αποκαλείται η βασική δομική του μονάδα. Η είσοδος ενός νευρώνα σημειώνεται ως $\overline{\mathbf{X_i}}$,

το οποίο διάνυσμα αναπαριστά τις συχνότητες εμφάνισης λέξεων στο i-οστό έγγραφο. Ορίζεται, επίσης, ένα σύνολο βαρών A που σχετίζεται με κάθε νευρώνα και χρησιμοποιείται προς υπολογισμό της συνάρτησης των δεδομένων εισόδου του f. Η γραμμική συνάρτηση του Νευρωνικού Δικτύου είναι η ακόλουθη: $\mathbf{p_i} = \mathbf{A} * \overline{\mathbf{X_i}}$. Σε ένα δυϊκό πρόβλημα ταξινόμησης, υποτίθεται ότι το διακριτικό κάθε κλάσης σημειώνεται ως $\mathbf{y_i}$ και ότι το πρόσημο της συνάρτησης πρόβλεψης $\mathbf{p_i}$ αποφέρει το διακρικό της κάθε κλάσης.

Τα Νευρωνικά Δίκτυα πολλαπλών επιπέδων χρησιμοποιούνται σε περιπτώσεις μη-γραμμικών διαχωριστικών. Τα εν λόγω πολλαπλά επίπεδα χρησιμοποιούνται προς την παραγωγή πολλαπλών γραμμικών διαχωριστικών, τα οποία χρησιμοποιούνται για την προσέγγιση περιοχών έγκλειστων σε μία συγκεκριμένη κλάση. Η έξοδος των νευρώνων στα αρχικά επίπεδα συνιστά την είσοδο των νευρώνων στα μετέπειτα επίπεδα. Τα Νευρωνικά Δίκτυα αξιοποιούνται ιδιαίτερα στην Ανάλυση Κειμένων.

- Δέντρα Αποφάσεων (Decision tree classifiers)
  Τα Δέντρα Αποφάσεων παρέχουν μία ιεραρχική αποδόμηση του χώρου δεδομένων εκπαίδευσης στον οποίο μία συνθήκη χρησιμοποιοείται για τον διαμερισμό των δεδομένων. Η συνθήκη ή το κατηγόρημα που διαμερίζει τον χώρο είναι ουσιαστικά η απεικόνιση της ύπαρξης ή της απουσίας μίας ή περισσότερων λέξεων. Η διαμέριση του χώρου δεδομένων πραγματοποιείται αναδρομικά έως ότου τα φύλλα του δέντρου περιέχουν έναν καθορισμένα ελάχιστο αριθμό καταχωρήσεων προς ταξινόμηση.

  Υπάρχουν άλλα είδη κατηγορημάτων τα οποία στηρίζονται στην ομοιότητα των εγγράφων ώστε να συσχετίσουν σύνολα όρων τα οποία μπορούν να αξιοποιηθούν στην περαιτέρω διαμέριση των εγγράφων. Τα διαφορετικά είδη διαμερισμού συνιστούν Μοναδικού Χαρακτηριστικού Διαμερίσεις (Single Attribute split), οι οποίες βασίζονται στην ύπαρξη ή την απουσία συγκεκριμένων λέξεων ή φράσεων σε έναν κόμβο του δέντρου έτσι ώστε να πραγματοποιηθεί ο διαμερισμός. Η Διαμέριση Πολλαπλών Χαρακτηριστικών βάσει Ομοιότητας (Similarity-based multi-attribute split) αξιοποιεί έγγραφα ή συστάδες λέξεων που εμφανίζονται συχνά. Η Διαμέριση Πολλαπλών Χαρακτηριστικών βάσει Διακρίνουσας (Discriminat-based multi-attribute split) βασίζεται στις διακρίνουσες, όπως στην Διακρίνουσα Fisher για να πραγματοποίηση του εν λόγω διαμερισμού.

- Ταξινομητές βάσει Κανόνων (Rule-Based classifiers)
  Όσον αφορά τους Ταξινομητές αυτού του είδους, ο χώρος των δεδομένων μοντελοποιείται σύμφωνα με ένα σύνολο κανόνων. Το αριστερό κομμάτι του κανόνα εκφράζει μία συνθήκη πάνω στο σύνολο χαρακτηριστικών εισόδου σε συζευκτική κανονική μορφή, ενώ το δεξί του μέρος συνιστά το διακριτικό της κλάσης, label. Οι συνθήκες αναφέρονται στην παρουσία όρων. Η απουσία όρων χρησιμοποιείται σπανιότερα, αφού δεν περιέχει αρκετή πληροφορία σε δεδομένα μικρής πυκνότητας.

  Υπάρχει μία πληθώρα κριτηρίων παραγωγής κανόνων, τα οποία κατασκευάζονται κατα τη διάρκεια της φάσης εκπαίδευσης του αλγορίθμου. Τα πλέον διαδεδομένα κριτήρια είναι

ο απόλυτος αριθμός στιγμιοτύπων ενός συνόλου δεδομένων εκπαίδευσης που σχετίζονται με τον κανόνα, καθώς και η δεσμευμένη πιθανότητα ότι το δεξί μέρος του κανόνα επαληθεύεται εφόσον επαληθεύεται το αριστερό του μέρος.

Unsupervised Learning (Μη Επιβλεπόμενη Μάθηση)

Τον κύριο σκοπό της ταξινόμησης κειμένου αποτελεί η ταξινόμηση των εγγράφων σε έναν συγκεκριμένο αριθμό προκαθορισμένων κατηγοριών. Προκειμένου να επιτευχθεί αυτό και, όπως αναλύθηκε και παραπάνω, ογκώδη σύνολα κειμένων εκπαίδευσης αναγνωρισμένης πολικότητας αξιοποιούνται στην Επιβλεπόμενη Μηχανική Μάθηση. Στην ταξινόμηση κειμένων είναι συχνά δύσκολο να συλλεχθούν ήδη ταξινομημένα κείμενα, αλλά περισσότερο προσιτή η συλλογή μη ταξινομημένων εγγράφων. Υπάρχουν προσεγγίσεις της Μη Επιβλεπόμενης Μάθησης που στηρίζονται στον Συναισθηματικό Προσανατολισμό (Semantic Orientation) χρησιμοποιώντας το μέτρο Point-wise Mutual Information (PMI) ή λεξικούς συσχετισμούς βάσει του εν λόγω δείκτη, σημασιολογικούς χώρους και ομοιότητες κατανομών, ούτως ώστε να μετρηθεί η ομοιότητα ανάμεσα σε λέξεις και πρότυπα πολικότητας.

2.3.2 Lexicon-Based Approach (Λεξικολογική Προσέγγιση)

Οι λέξεις που εκφέρουν άποψη χρησιμοποιούνται σε πολλά ζητήματα αναγνώρισης συναισθηματικής φόρτισης. Οι θετικά φορτισμένες λέξεις άποψης αξιοποιούνται στην έκφραση επιθυμητών καταστάσεων, ενώ οι αρνητικά φορτισμένες λέξεις άποψης στην έκφραση λιγότερο επιθυμητών καταστάσεων. Απαντώνται, ακόμη, φράσεις που εκφέρουν απόψεις, αλλά και ιδιώματα που, συγκεντρωτικά, συγκροτούν Λεξικά Απόψεων (Opinion Lexicons). Προκειμένου να συνταχθεί μία λίστα τέτοιων λέξεων υπάρχουν τρεις κύριες προσεγγίσεις. Η Χειροκίνητη Προσέγγιση είναι δαπανηρή σε χρόνο και δεν υλοποιείται ανεξάρτητα. Είναι σύνηθες να συνδυάζεται με μία από τις παρακάτω αυτοματοποιημένες προσεγγίσεις, και αξιοποιείται στον τελικό έλεγχο προκειμένου να αποφευχθούν λάθη λόγω της αυτοματοποίησης της διαδικασίας. Οι εν λόγω αυτοματοιημένες διαδικασίες παρατίθονται στη συνέχεια.

Dictionary-Based Approach

Την βασική στρατηγική της Προσέγγισης βάσει Λεξικού αποτελεί η σύνταξη ενός μικρού συνόλου λέξεων που εκφέρουν άποψη με γνωστή συναισθηματική πολικότητα χειροκίνητα. Στη συνέχεια, το σύνολο αυτό επεκτείνεται με γνωστά Συναισθηματικά Λεξικά, όπως το WordNet αναζητώντας τα συνώνυμα και τα αντώνυμα των λέξεων που το συγκροτούν. Οι νέες λέξεις που βρίσκονται με τον τρόπο αυτό προστίθονται στην αρχική λίστα και η διαδικασία επαναλαμβάνεται. Η επανάληψη της προσθήκης νέων λέξεων στο σύνολο σταματά όταν δεν βρίσκονται άλλες λέξεις να προστεθούν σε αυτό. Αφού ολοκληρωθεί η διαδικασία, συχνά ακολουθεί επιθεώρηση από τον ανθρώπινο παράγοντα, ούτως ώστε να αφαιρεθούν ή να διορθωθούν τυχόν λάθη. Το βασικό μειονέκτημα εδώ είναι η αδυναμία να βρεθούν λέξεις άποψης που να εντάσσονται σε έναν συγκεκριμένο εννοιολογικό ή/και θεματικό τομέα.

Corpus-Based Approach

Η Προσέγγιση αυτή συνιστά αρωγό στην ανεύρεση λέξεων βάσει εννοιολογικού περιεχομένου και συμφραζομένων. Οι μέθοδοι της στηρίζονται σε Συντακτικά Μοτίβα ή Μοτίβα που προκύπτουν από τη Λίστα-Φύτρο των λέξεων άποψης προκειμένου να βρεθούν και άλλες τέτοιες λέξεις σε ένα μεγάλο σώμα κειμένων. Η Corpus Based Προσέγγιση από μόνη της δεν θεωρείται τόσο αποτελεσματική όσο η Dictionary-Based Approach, αφού είναι δύσκολο να συγκροτηθεί τόσο μεγάλο σώμα κειμένων, ούτως ώστε να καλύψει όλες τις λέξεις του Λεξιλογίου μίας γλώσσας. Ωστόσο, το βασικό πλεονέκτημα αυτής της Μεθόδου Προσέγγισης είναι ότι συνδράμει στην ανεύρεση λέξεων που σχετίζονται με μία συγκεκριμένη μεθοδολογία, χτίζοντας το σώμα κειμένων που εκπαιδεύουν τον αλγόριθμο ανάλογα. Η Corpus Based Approach υλοποιείται βάσει της Στατιστικής ή της Σημασιολογικής Προσέγγισης, οι οποίες παρουσιάζονται παρακάτω:

- Στατιστική Προσέγγιση
  Η ανεύρεση συνύπαρξης μοτίβων και λέξεων άποψης που μπορούν να λειτουργήσουν ως φύτρο μπορεί να πραγματοποιηθεί με τη βοήθεια στατιστικών τεχνικών. Πιο συγκεκριμένα, το τελευταίο μπορεί να πραγματοποιηθεί αντλώντας a posteriori πολικότητες αξιοποιώντας την συνύπαρξη επιθέτων σε ένα σώμα κειμένων. Είναι δυνατό να χρησιμοποιηθεί ολόκληρο το σώμα των κατεταγμένων εγγράφων που βρίσκονται στο Διαδίκτυο για την κατασκευή ενός λεξικού. Ο τρόπος αυτός ξεπερνά το πρόβλημα της μη διαθεσιμότητας κάποιων λέξεων σε περίπτωση που το σώμα των κειμένων που χρησιμοποιήθηκε δεν είναι αρκετά μεγάλο.

  Η πολικότητα μίας λέξης μπορεί να αναγνωριστεί μελετώντας την συχνότητα εμφάνισης της ίδιας μέσα σε ένα εκτενές σώμα κειμένων που περιλαβάνει σχολιασμό. Εαν μία λέξη εμφανίζεται περισσότερο συχνά σε θετικά φορτισμένα κείμενα, τότε η πολικότητα της είναι μάλλον θετική. Εαν, ωστόσο, η λέξη εμφανίζεται συχνότερα σε αρνητικά κατεταγμένα κείμενα, τότε η πολικότητά της είναι αρνητική. Εάν η συχνότητα εμφάνισής της είναι ίδια στα κείμενα έτερης πολικότητας, η λέξη κατατάσσεται ως ουδέτερης πολικότητας.

  Οι παρεμφερείς λέξεις άποψης συνυπάρχουν συχνά σε ένα σώμα κειμένων. Η διαπίστωση αυτή συνιστά τη βάση όλων των σύγχρονων μεθόδων. Για το λόγο αυτό, στην περίπτωση που δύο λέξεις εμφανίζονται συχνά μαζί στα πλαίσια ενός κοινού θέματος, αναμένεται να έχουν και την ίδια συναισθηματική πολικότητα. Επομένως, η πολικότητα μίας άγνωστης λέξης μπορεί να καθοριστεί υπολογίζοντας τη σχετική συχνότητα της συνύπαρξης με μία άλλη λέξη με τη χρήση του δείκτη PMI (Pointwise mutual information).

  Οι στατιστικές μέθοδοι χρησιμοποιούνται σε ποικίλες εφαρμογές Συναισθηματικής Ανάλυσης. Μία από αυτές είναι ο εντοπισμός χειραγώγησης κριτικών μέσω της πραγματοποίησης στατιστικών δοκιμών τυχαιότητας που ονομάζονται Runs Test. Η Λανθάνουσα Σημασιολογική Ανάλυση (Latent Semantic Analysis (LSA)) συνιστά μία προσέγγιση που χρησιμοποιείται προς ανάλυση των σχέσεων ανάμεσα σε ένα σύνολο εγγράφων και

των όρων που εμπεριέχονται σε αυτά προκειμένου να παραχθεί ένα σύνολο μοτίβων και νοηματικών συσχετισμών μεταξύ τους. Ακόμη, ο Συναισθηματικός Προσανατολισμός μίας λέξης αποτελεί μία στατιστική προσέγγιση που χρησιμοποιείταται παράλληλα με το PMI, ενώ ο Εννοιολογικός Χώρος αποτελεί τον χώρο στον οποίο οι λέξεις αναπαρίστονται με σημεία και η θέση κάθε σημείου ως προς τον κάθε άξονα προσδιορίζει τη σημασία της λέξης. Μία εφαρμογή του Εννοιολογικού Χώρου ονομάζεται Hyperspace Analogue to Language (HAL).

- Σημασιολογική Προσέγγιση
  Η Σημασιολογική Προσέγγιση αποδίδει Συναισθηματικές Τιμές κατευθείαν και στηρίζεται σε ετερόκλητες αρχές υπολογισμού ομοιότητας ανάμεσα σε λέξεις. Ένα παράδειγμα σημασιολογικής προσέγγισης είναι το λεξικό Wordnet. Το WordNet παρέχει διαφορετικού τύπου σημασιολογικές σχέσεις ανάμεσα σε λέξεις οι οποίες χρησιμοποιούνται προς τον υπολογισμό συναισθηματικών πολικότητων. Το WordNet μπορεί να χρησιμοποιηθεί επίσης προς απόκτηση μίας λίστας λέξεων συναισθήματος μέσω της επαναλαμβανόμενης επέκτασης του αρχικού συνόλου με συνώνυμα και αντώνυμα και, στη συνέχεια, για τον καθορισμό της συναισθηματικής πολικότητας μίας άγνωστης λέξης από την αντίστοιχη μέτρηση των θετικών και αρνητικών συνωνύμων της.

  Η Σημασιολογική Προσέγγιση χρησιμοποιείται σε πολλές εφαρμογές προκειμένου να κατασκευαστεί ένα μοντέλο λεξικού για την περιγραφή των ρημάτων, ουσιαστικών και επιθέτων που μπορούν να χρησιμοποιηθούν στην Ανάλυση Συναισθήματος. Οι Σημασιολογικές Μέθοδοι μπορούν να συνδυαστούν με τις Στατιστικές Μεθόδους βελτιστοποιώντας την Προσέγγιση της Ανάλυσης Συναισθήματος βάσει αυτών των μεθόδων.

Κεφάλαιο 3

Προσέγγιση του Προβλήματος

Η παρούσα Διπλωματική Εργασία εστιάζει σε μία συγκεκριμένη προσέγγιση και υλοποίηση της Ανάλυσης Συναισθήματος. Στο κεφάλαιο αυτό παρουσιάζονται τρεις μέθοδοι εξαγωγής χαρακτηριστικών, οι οποίες στα επόμενα κεφάλαια θα συγκριθούν ως προς την απόδοσή τους όσον αφορά κριτικές ταινιών, οι οποίες αποτελούν και το επιλεγμένο σώμα κειμένων προς ανάλυση. Τα χαρακτηριστικά κειμένου που εξάγονται βάσει των μεθόδων αυτών αποτελούν την είσοδο ενός Μοντέλου Επιβλεπόμενης Μάθησης που επιστρέφει το συναίσθημα του κειμένου. Οι υπό μελέτη μέθοδοι εξαγωγής χαρακτηριστικών, όπως και οι μέθοδοι ταξινόμησης κειμένου παρουσιάζονται εκτενώς στη συνέχεια.

3.1 Βήματα Επιβλεπόμενης Μάθησης

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο η αλληλουχία δράσεων σε μία προσέγγιση Επιβλεπόμενης Μάθησης έχει ως εξής [18]:

1. Εκπαίδευση Μοντέλου
   Στην εκπαίδευση του Μοντέλου Επιβλεπόμενης Μάθησης, παρέχεται ως είσοδος στον αλγόριθμο ένα διάνυσμα χαρακτηριστικών (feature vector) το οποίο συνίσταται συνήθως σε μία πλειάδα αριθμών. Τα εν λόγω χαρακτηριστικά εισόδου είναι δυνατό να λαμβάνουν διακριτές είτε συνεχείς τιμές. Στο διάνυσμα χαρακτηριστικών περιλαμβάνεται, στο στάδιο αυτό, η επιθυμητή και αναμενόμενη έξοδος του Μοντέλου. Έτσι, τα χαρακτηριστικά που συνιστούν το διάνυσμα, όπως επίσης και το χαρακτηριστικό κλάσης (class attribute) ονομάζονται Στιγμιότυπο Εκπαίδευσης (training instance).

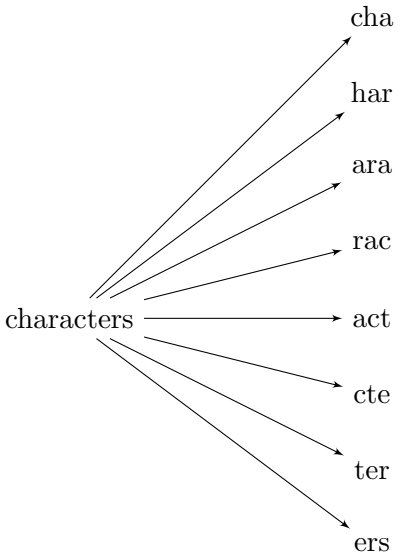
2. Αξιολόγηση Μοντέλου
   Αφού ολοκληρωθεί η εκπαίδευση του μοντέλου, ακολουθεί η αξιολόγησή του κατα την οποία παρέχονται στο μοντέλο νέα δεδομένα τα οποία πρέπει να ταξινομήσει. Όπως και στην Εκπαίδευση, έτσι και εδώ, στον Αλγόριθμο δίνεται ως είσοδος ένα διάνυσμα χαρακτηριστικών στο οποίο περιέχεται και το χαρακτηριστικό κλάσης. Ωστόσο, στην προκειμένη περίπτωση, το χαρακτηριστικό κλάσης δεν αξιοποιείται από το Αλγόριθμο, παρά η ταξινόμηση του κειμένου πραγματοποιείται σύμφωνα με τα χαρακτηριστικά ει-

σόδου. Αφού εξαχθεί το αποτέλεσμα από τον Ταξινομητή (Classifier), αυτό συγκρίνεται με το χαρακτηριστικό κλάσης, ούτως ώστε να αξιολογηθεί η απόδοση του εκπαιδευμένου πλέον Αλγορίθμου.

3.2 Επιλεγμένες Μέθοδοι Εξαγωγής Χαρακτηριστικών

3.2.1 N-grams

Στους γνωστικούς τομείς της Επεξεργασίας Φυσικής Γλώσσας και των Πιθανοτήτων, ένα n-gram συνίσταται σε μία ακολουθία n το πλήθος οντοτήτων που προκύπτει από μία ακολουθία κειμένου ή ομιλίας. Οι εν λόγω οντότητες μπορεί να είναι φωνήματα, συλλαβές, γράμματα ή λέξεις ανάλογα με την εφαρμογή. Τα n-grams τυπικά συλλέγονται από ένα σώμα κειμένων ή καταγραφών ομιλίας. Ένα n-gram μεγέθους ένα ονομάζεται unigram, αντίστοιχα bigram και trigram για την περίπτωση μεγέθους δύο και τρία. Με τον ίδιο τρόπο πραγματοποιείται γενίκευση για n-grams μεγαλύτερου μεγέθους [5].

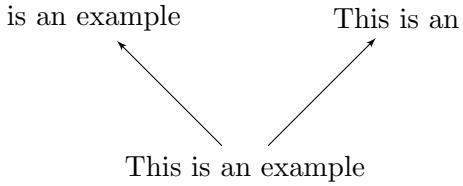


Σχήμα 3.1: Trigrams Χαρακτήρων Μεγέθους Τρία

Τα μοντέλα N-gram χρησιμοποιούνται ευρέως στην στατιστική Επεξεργασία Φυσικής Γλώσσας (Statistical Natural Language Processing). Στην Αναγνώριση Ομιλίας, τα φωνήματα και οι ακολουθίες φωνημάτων μοντελοποιούνται με τη χρήση μίας n-gram κατανομής. Στα πλαίσια της αναγνώρισης γλώσσας, αλληλουχίες χαρακτήρων και λέξεων μοντελοποιούνται με εφαρμογή σε διαφορετικές γλώσσες[1]. Σαφέστερα, η Προσέγγιση που στηρίζεται στα N-grams κατακερματίζει σε ισομερή κομμάτια χαρακτήρων μήκους n. Θεωρείται ότι κάθε γλώσσα χρησιμοποιεί συγκεκριμένα N-Grams συχνότερα από τις υπόλοιπες, παρέχοντας τοιουτοτρόπως έρεισμα για την αναγνώρισή της[2]. Τα n-grams, πέρα από την αλληλουχία λέξεων, είναι δυνατό να χρησιμοποιηθούν προς επεξεργασία όλων σχεδόν των τύπων δεδομένων. Παραδείγματος χάριν, έχουν χρησιμοποιηθεί για την εξαγωγή χαρακτηριστικών και τη

δημιουργία μεγάλων συστάδων φωτογραφιών της γης από δορυφόρο και, με τον τρόπο αυτό, επιτεύχθηκε η αναγνώριση του σημείου του πλανήτη που απεικονιζόταν σε αυτές. Επιπρόσθετα, έχουν χρησιμοποιηθεί επιτυχώς για μία αρχική προσπέλαση και αναζήτηση γενετικών αλληλουχιών, όπως επίσης για την αναγνώριση των ειδών από τα οποία προέρχονται σύντομες ακολουθίες DNA [20].

Εδώ, παρατίθονται δύο παραδείγματα N-grams μεγέθους τρία, όπου το πρώτο αντιστοιχίζεται σε εξαγωγή χαρακτήρων (Σχήμα 3.1), ενώ το δεύτερο σε εξαγωγή λέξεων (Σχήμα 3.2).



Σχήμα 3.2: Trigrams Λέξεων Μεγέθους Τρία

3.2.2 N-grams Graphs

Το κυριότερο μειονέκτημα του προηγούμενου μοντέλου αποτελεί η εξαγωγή των N-grams αγνοώντας την σημαντική πληροφορία που περιέχεται στην σχετική τους θέση στο αρχικό κείμενο. Για παράδειγμα, οι λέξεις ΄kiwi΄ και ΄wiki΄ αναπαρίστανται από το ίδιο δίγραμμα (bigram) μολονότι το νόημά τους είναι εντελώς διαφορετικό.

Προκειμένου να ξεπεραστεί αυτό το πρόβλημα, εισάγεται η μέθοδος των n-gram γράφων, η οποία συσχετίζει όλα τα ζεύγη των n-grams με ακμές που αναπαριστούν τη συχνότητα που βρίσκονται κοντά μεταξύ τους στο δοσμένο σωμα κειμένων. Με άλλα λόγια, η παρούσα μέθοδος εξαγωγής χαρακτηριστικών κατασκευάζει γράφους, των οποίων οι κόμβοι αποτελούν ανεξάρτητα n-grams, ενώ στις ακμές τους ανατίθεται βάρος ανάλογο της μέσης απόστασης μεταξύ τους στα πλαίσια αυτών. Ένας n-gram γράφος χαρακτηρίζεται από τρεις παραμέτρους:

1. Τον ελάχιστο n-gram βαθμό L_{min}
2. Το μέγιστο n-gram βαθμό L_{max}, όπως και
3. Το μήκος παραθύρου D_{win}

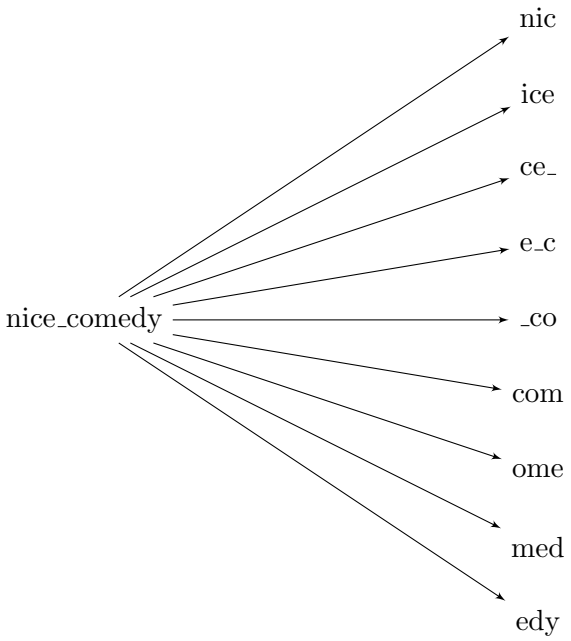
Στη παρούσα εργασία θεωρείται ότι $L_{min} = L_{max} = D_{win} = n$, η οποία σχέση ωστόσο έχει αποδειχθεί πειραματικά να αντιπροσωπεύει απόδοση κοντά στην μέγιστη [4].

Το μοντέλο των N-gram γράφων μπορεί να χρησιμοποιηθεί για την ενιαία αναπαράσταση ενός σώματος κειμένων με έναν γράφο. Σαφέστερα, μέσω της δυναμικής δημιουργίας των γράφων για καθεμία από τις πολικότητες συναισθήματος, είναι δυνατό να ανγωριστούν μοτίβα κοινά σε θεματολογία ή συναισθηματική φόρτιση, όπως επαναλαμβανόμενες ακολουθίες χαρακτήρων. Στην περίπτωση των κριτικών ταινιών που μελετάται εδώ, το εν λόγω μοντέλο αξιοποιείται για την αναπαράσταση ενός συνόλου κειμένων κριτικής ταινιών που εκφέρουν κοινό συναίσθημα. Το μοντέλο των n-gram γράφων έχει χρησιμοποιηθεί επιτυχώς για την Ανάλυση Συναισθήματος μικροϊστολογίων (tweets) [4].

Εφαρμογή αναπαράστασης N-gram γράφων

Θεωρώντας ένα κείμενο κριτικής ταινίας $'Exciting\ movie'$ και επιλέγοντας μέγεθος n ίσο με τρία, εξάγονται όλα τα επικαλυπτόμενα trigrams χαρακτήρων.

Καθεμία από τις τριπλέτες χαρακτήρων αναπαρίσταται ως ένας κόμβος στον γράφο. Η επιπρόσθετη πληροφορία που παρέχει ο γράφος αποτελεί η γειτνίαση των n-grams, η οποία κατ' επέκταση ισοδυναμεί με τη σχετική θέση τους στο αρχικό κείμενο ανάλυσης. Ανάλογα με το εύρος της πληροφορίας που επιθυμείται να περιέχεται στον γράφο ορίζεται η παράμετρος του μήκους παραθύρου (D_{win}). Η εν λόγω παράμετρος καθορίζει την τοποθέτηση των ακμών ανάμεσα στα γειτονικά n-grams που βρίσκονται μέσα στα όρια του παραθύρου. Ως βάρη ακμών ορίζονται τα πλήθη εμφανίσεων των επιμέρους ζευγών του υπο μελέτη n-gram με τα γειτονικά του μέσα στα πλαίσια ξανά του παραθύρου.

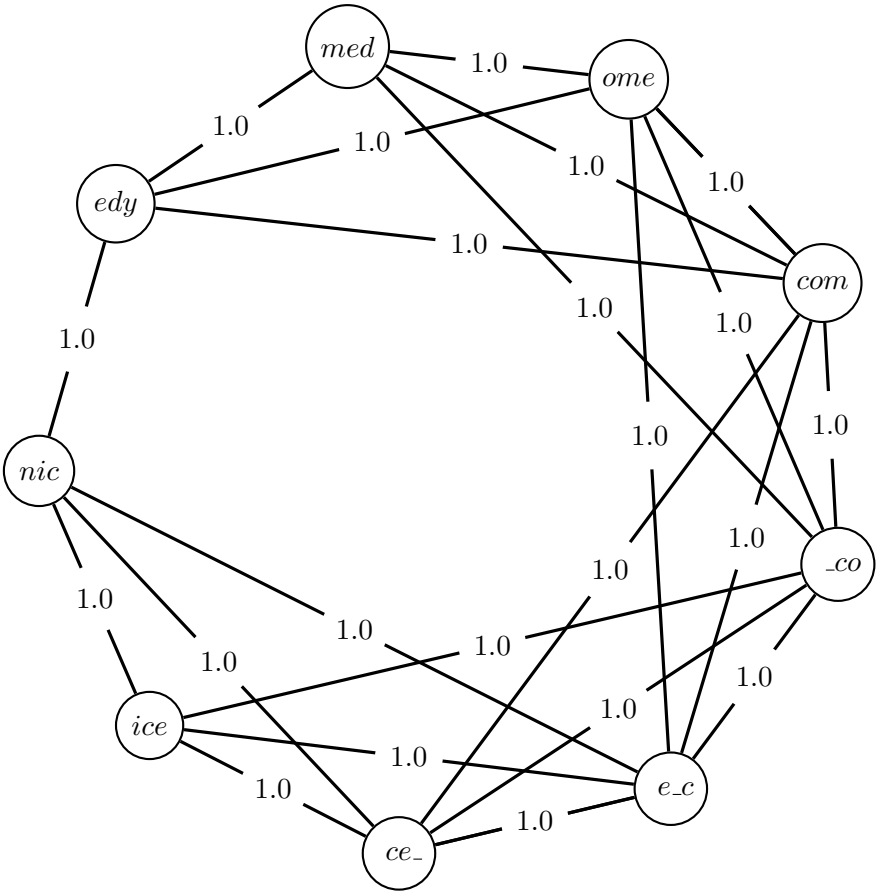


Σχήμα 3.3: Εξαγωγή N-grams Χαρακτήρων Μεγέθους Τρία

Επομένως, επιλέγοντας παράθυρο γειτνίασης ίσο με $n = 3$ προκύπτει το ακόλουθο γράφημα του κειμένου κριτικής ταινίας (Σχήμα 3.4).

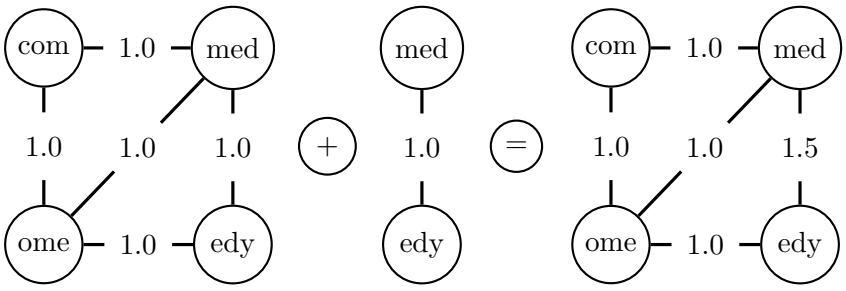
Εφαρμογή συγχώνευσης N-gram Γράφων

Στο στάδιο της εξαγωγής χαρακτηριστικών κατά την εκπαίδευση του αλγορίθμου βαθιάς μάθησης, δημιουργείται ένας γράφος μοντέλο για κάθε συναισθηματική φόρτιση κειμένου (positive, negative review), ο οποίος αξιοποιείται στο επόμενο στάδιο. Ο γράφος του μοντέλου αρχικοποιείται με το πρώτο κείμενο εκπαίδευσης. Για καθένα από τα επόμενα κείμενα δημιουργείται ένας γράφος που συγχωνεύεται με τον γράφο μοντέλου. Θεωρώντας αρχικοποιημένο γράφο μοντέλο με το κείμενο $'comedy'$ και παραμέτρους $n = 3$ και $D_{win} = 2$, καθώς και γράφο κειμένου με περιεχόμενο την ακολουθία $'medy'$, η επιθυμητή συγχώνευσή τους



Σχήμα 3.4: Γράφος N-grams Χαρακτήρων Μεγέθους Τρία και Παραθύρου ίδιου μεγέθους

προκύπτει ως εξής.



Σχήμα 3.5: Συγχώνευση N-gram Γράφων

Κατα τη συγχώνευση, προστίθονται όλοι οι διαφορετικοί κόμβοι στο μοντέλο γράφο, ενώ ταυτόχρονα τροποποιούνται τα βάρη των ακμών, ούτως ώστε σε περίπτωση που προστίθεται στο μοντέλο γράφο ακμή που υπάρχει ήδη, να ανανεώνεται το βάρος της ακμής αυτής και να ισούται με το μέσο όρο των δύο βαρών. Γενικεύοντας, εαν μία ακμή εμφανίζεται στον γράφο μοντέλου και σε $n-1$ γράφους κειμένων, υπολογίζεται ο μέσος όρων των n βαρών. Κατα τη $n-$οστή συγχώνευση, ο γράφος έχει ήδη συχωνεύσει $n-2$ φορές την ακμή με αποτέλεσμα αυτή να έχει βάος που αντιστοιχεί σε μέσο όρο $n-1$ προηγούμενων αριθμών.

το βάρος της ακμής πρέπει να ανανεωθεί κατάλληλα ώστε να αποτελεί το μέσο όρο όλων των n αριθμών. Επειδή ο αλγόριθμος πουυ χρησιμοποιείται δεν αποθηκεύει τα προηγούμενα βάρη για να υπολογίζει εκ νέου τον μέσο όρο, αλλά λαμβάνει τα βάρη στις υπο συγχώνευση ακμές γνωρίζοντας το πλήθος τους(n), μπορούμε να τον υπολογίσουμε με την παρακάτω παράσταση [10].

$$new\_average = old\_average + \frac{1}{n} * (new\_weight - old\_average)$$

Η παράσταση αποδεικνύεται εύκολα ότι δίνει το μέσο όρο n στοιχείων αν θέσουμε

$$old\_average = \frac{w_1 + w_2 + ... + w_{n-1}}{n-1}$$

Συνεπώς, κατά την εκπαίδευση του αλγορίθμου μηχανικής μάθησης, κατασκευάζονται δύο μοντέλα κειμένων εκπαίδευσης, ένα για καθεμία από τις συναισθηματικές πολικότητες. Πιο συγκεκριμένα, και υπενθυμίζοντας ότι στην παρούσα εργασία η Ανάλυση Συναισθήματος προσεγγίζεται δυϊκά, κατά την εκπαίδευση του αλγορίθμου δημιουργείται ένας γράφος θετικής πολικότητας (positive graph) και ένας γράφος αρνητικής πολικότητας (negative graph). Οι γράφοι αυτοί αξιοποιούνται στα επόμενα στάδια της διαδικασίας μάθησης.

Δείκτες Σύγκρισης

Όπως αναφέρθηκε παραπάνω, για καθένα από τα κείμενα εκπαίδευσης του αλγορίθμου δημιουργείται ένας γράφος που συγχωνεύτεαι με ένα από τα δύο μοντέλα γράφων, το θετικό γράφο μοντέλο και τον αρνητικό. Προκειμένου να αποφανθεί σε ποιον από τους δύο γράφους μοντέλα θα συγχωνευτεί ο n-gram γράφος του κειμένου παρουσιάζονται παρακάτω ορισμένοι δείκτες ομοιότητας, οι οποίοι αξιοποιούνται όχι μόνο για την αναγωγή του n-gram γράφου του κειμένου προς ταξινόμηση, αλλά αποτελούν ταυτόχρονοα ένα σύνολο χαρακτηριστικών (features), δηλαδή ένα instance εισόδου του αλγορίθμου μηχανικής μάθησης.

Οι δείκτες σύγκρισης που χρησιμοποιήθηκαν κατά το στάδιο της εξαγωγής χαρακτηριστικών κειμένων είναι οι ακόλουθοι [9]:

1. Concurrence ή Containment Similarity (CS)
   Αναπαριστά το πλήθος των κοινών ακμών που βρίσκονται και στους δύο γράφους χωρίς να λαμβάνεται υπόψη το βάρος των ακμών. Σαφέστερα, εκφράζει το ποσοστό των ακμών του γράφου με το μικρότερο πλήθος ακμών που εμφανίζεται στον μεγαλύτερο γράφο. Για τον υπολογισμό της τιμής του χαρκατηριστικού αυτού πραγματοποιείται έλεγχος ύπαρξης καθεμίας από τις ακμές του μικρότερου γράφου στον μεγαλύτερο.

   $$CS = \frac{no\ of\ edges\ of\ the\ smallest\ graph\ coocccuring\ in\ the\ biggest\ graph}{number\ of\ edges\ of\ the\ smallest\ graph}$$

2. Size Similarity (SS)
   Αποτελεί ένα μέγεθος που εκφράζει την αναλογία μεγεθών των δύο γράφων και λαμβάνει τιμές στο διάστημα $[0, 1]$. Η παράσταση του:

   $$SS = \frac{no\ of\ edges\ of\ the\ smallest\ graph}{number\ of\ edges\ of\ the\ biggest\ graph}$$

3. Value Similarity (VS)

   Χρησιμοποιείται για την αναπαράσταση του πλήθος των κοινών ακμών που βρίσκονται και στους δύο γράφους, λαμβάνοντας ωστόσο υπόψη τα βάρη των ακμών. Στην περίπτωση που μία ακμή βρίσκεται και στους δύο γράφους με διαφορετικό βάρος, στον υπολογιστμό του εν λόγω δείκτη λαμβάνεται η ακμή με το μικρότερο βάρος, καθώς αντιστοιχεί σε πλήθος εμφανίσεων του ζεύγους των n-grams που αποθηκεύτηκε και στους δύο γράφους. Ορίζοντας, λοιπόν, w_c^i το βάρος της κοινής ακμής στον ένα γράφο και w_c^j το βάρος της στον άλλο γράφο, και θεωρώντας ότι οι τιμές του δείκτη κυμαίνονται ανάμεσα στο 0 και το 1:

$$VS = \frac{\sum \frac{\min(w_c^i, w_c^j)}{\max(w_c^i, w_c^j)}}{number\ of\ edges\ of\ the\ biggest\ graph}$$

4. Normalized Value Similarity (NVS)

   Ο σκοπός του δείκτη αυτού έγκειται στην αποσύνδεση του δείκτη VS από το μέγεθος των γράφων. Για το σκοπό αυτό ο δείκτης VS κανονικοποιείται με την διαίρεση του με τον δείκτη SS. Το σύνολο τιμών του δείκτη κυμαίνεται ανάμεσα στο 0 και το 1.

$$NVS = \frac{VS}{SS}$$

   ή

$$NVS = \frac{\sum \frac{\min(w_c^i, w_c^j)}{\max(w_c^i, w_c^j)}}{number\ of\ edges\ of\ the\ smallest\ graph}$$

3.2.3 Bag Of Words

Το μοντέλο Bag Of Words συνιστά μία απλοποιημένη αναπαράσταση ενός κειμένου. Σαφέστερα, στο μοντέλο ένα κείμενο (όπως μία πρόταση ή ένα έγγραφο) αναπαρίσταται ως ένας σάκος (multiset) από τις λέξεις που το απαρτίζουν αγνοώντας γραμματικές σχέσεις και εξαρτήσεις, όπως και σειρά των λέξεων, αλλά διατηρώντας την πληροφορία της πολλαπλότητάς τους. Το εν λόγω μοντέλο χρησιμοποιείται επίσης και στον τομέα της Όρασης των Υπολογιστών. Στον γνωστικό τομέα της Ανάλυσης Συναισθήματος, ή της κατάταξης κειμένων εν γένει, η μέθοδος Bag Of Words είναι κοινότοπο να αξιοποιείται στο στάδιο εκπαίδευσης ενός ταξινομητή, όπου η συχνότητα εμφάνισης ή/και η παρουσία κάθε λέξης, χρησιμοποιείται ως χαρακτηριστικό (feature) εισόδου του αλγορίθμου. Τετριμμένα, κάθε αναπαράσταση της μεθόδου μπορεί να θεωρηθεί ως ένα n-gram με παράθυρο ίσο με 1, όπου δεν συγκρατείται καμία πληροφορία συσχετισμού ανάμεσα στις λέξεις του κειμένου παρά μόνο η πληροφορία της παρουσίας τους.

Παράδειγμα Εφαρμογής

Πρακτικά, το μοντέλο Bag Of Words χρησιμοποιείται κυρίως ως ένα εργαλείο εξαγωγής χαρακτηριστικών. Αφού μετατραπεί ένα κείμενο σε ένα ασκό με λέξεις (BOW), είναι δυνατό

Term	**Frequency**
This	1
is	1
truly	1
a	1
great	1
movie	2
I	1
loved	1
not	1
only	1
the	2
but	1
cast	1
too	1

Πίνακας 3.1: Πίνακας συχνότητας εμφάνισης όρων βάσει της BOW Μεθόδου

να υπολογιστούν πολλαπλά μέτρα για να χαρακτηρίσουν το υπο μελέτη κείμενο. Ο πιο κοινός τύπος χαρακτηριστικών που εξάγονται βάσει αυτής της μεθόδου είναι η συχνότητα όρων (term frequency), δηλαδή το πλήθος εμφάνισης ενός όρου στο κείμενο.

Ως παράδειγμα θέτουμε την εξής πρόταση:
'This is truly a great movie. I loved not only the movie but the cast too!'

Το διάνυσμα $[1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1]$ που εξάγεται αναπαριστά το κείμενο χωρίς να διατηρεί τη σειρά των λέξεων στις αρχικές προτάσεις, το οποίο είναι και το κύριο γνώρισμα της μεθόδου. Αυτού του είδους η αναπαράσταση βρίσκει ευρεία εφαρμογή σε ποικίλους τομείς, όπως, για παράδειγμα, τη διαλογή ηλεκτρονικού ταχυδρομείου (spam detection and e-mail filtering) [16]. Παρόλα αυτά, η συχνότητα όρων δεν συνιστά απαραίτητα την βέλτιστη αναπαράσταση ενός κειμένου. Συχνά εμφανιζόμενες λέξεις, όπως τα άρθρα, αποτελούν τις περισσότερες φορές τους όρους με τη μεγαλύτερη συχνότητα. Συνεπώς, μία υψηλή απόλυτη τιμή δεν σημαίνει απαραίτητα ότι η αντίστοιχη λέξη είναι σημαντική. Η πλέον δημοφιλής πρακτική, ούτως ώστε να αντιμετωπιστεί το πρόβλημα είναι η κανονικοποίηση των συχνοτήτων εφάνισης

όρων. Επιπρόσθετα, για το σκοπό της ταξινόμησης έχουν αναπτυχθεί εναλλακτικές μέθοδοι προκειμένου να λαμβάνεται υπόψη το διακριτικό κλάσης (class label) του κάθε εγγράφου [12], Τέλος, δυαδική ανάθεση βάρους (παρουσία/απουσία όρου ισοδύναμη με 1/0) χρησιμοποιείται αντί της συχνότητας σε διάφορα προβλήματα, όπως επίσης εφαρμόζεται και από το σύστημα λογισμικού βαθιάς μάθησης WEKA που αφορά την παρούσα εργασία.

3.2.4 Word Graphs

Βασιζόμενη στη μέθοδο των N-gram Γράφων, η παρούσα μέθοδος συνδυάζει τις μεθόδους Bag Of Words και N-Gram Γράφων. Σαφέστερα στην μέθοδο των Word Graphs εξάγονται από το κείμενο και αναπαρίστανται ως κόμβοι λέξεις και όχι αλληλουχίες χαρακτήρων σταθερού μήκους, όπως συμβαίνει με τους N-gram γράφους. Η αναπαράσταση του μοντέλου-κειμένου πλέον γίνεται με έναν γράφο λέξεων, ο οποίος δημιουργείται και πάλι σταδιακά με τη συγχώνευση του γράφου λέξεων κάθε κειμένου κριτικής ταινιών [10]. Το εν λόγω μοντέλο συνεχίζει να διατηρεί τις υπόλοιπες ιδιότητες της μεθόδου και γνωρισμάτων των N-Gram Γράφων, ενώ αντιμετωπίζει ταυτόχρονα ελλείματά τους. Πιο συγκεκριμένα, είναι πιθανό δύο λέξεις διαφορετικής πολικότητας να έχουν τα ίδια n-grams και η διαίρεσή τους σε αυτά να χάνει τη δυνατότητα ένδειξης της πολικότητας. Για παράδειγμα, οι λέξεις happy και unhappy έχουν κοινά 3-grams, αλλά και 4-grams:

1. hap, app, ppy
2. unh, nha, hap, app, ppy

1. happ, appy
2. unha, nhap, happ, appy

Επομένως, παρατηρείται η εμφάνιση της λέξης happy και στους δύο γράφους πολικότητας, γεγονός που ισοδυναμεί με ίδιο δείκτη ομοιότητας Containment Similarity και με τους δύο γράφους, με αποτέλεσμα τη δημιουργία θορύβου. Σύμφωνα με τη μέθοδο των Word Graphs, ωστόσο, οι λέξεις happy και unhappy τοποθετούνται στους αντίστοιχους γράφους πολικότητας, αποτελώντας η καθεμία ένα αυτοτελή κόμβο που σηματοδοτεί την παρουσία της λέξης, όπως και στη μέθοδο Bag Of Words [10].

Παράδειγμα Εφαρμογής Μεθόδου

Εδώ, σε κάθε βήμα εξάγουμε από το κείμενο μία λέξη αντί για n συνεχόμενους χαρακτήρες. Συνεπώς, θέτοντας την παράμετρο των N-gram γράφων ίση με ένα, λαμβάνουμε uni-grams λέξεων ή word uni-grams. Θεωρώντας την ακόλουθη πρόταση ως παράδειγμα "I thought this was a quite good movie", οι εξαγόμενες λέξεις είναι οι εξής:

1. I
2. thought

3. this

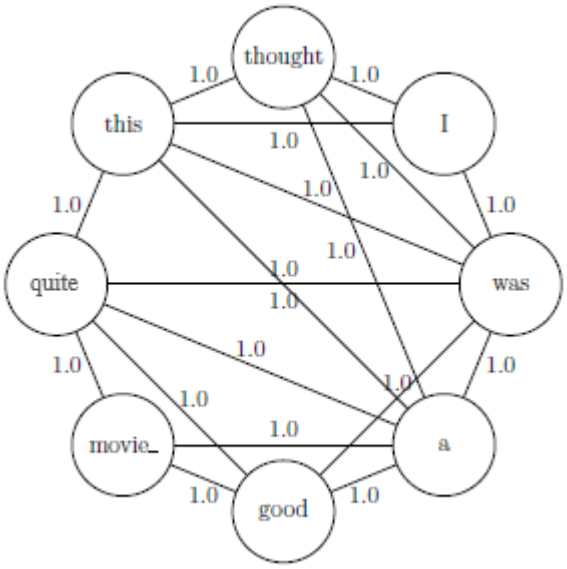
4. was

5. a

6. quite

7. good

8. movie



Σχήμα 3.6: Αναπράσταση πρότασης σε (Word Graph)

Καθεμία από τις παραπάνω λέξεις θα αποτελέσει έναν κόμβο του παραγόμενου γράφου. Η διαδικασία τοποθέτησης των ακμών προϋποθέτει την ύπαρξη ενός παραθύρου, το οποίο πλέον δηλώνει το πλήθος των γειτονικών λέξεων για κάθε λέξη, με τις οποίες αυτή θα γειτνιάζει. Τα βάρη των ακμών αντιπροσωπεύουν και πάλι το πλήθος εμφανίσεων του ζεύγους των λέξεων μέσα στο παράθυρο αυτό. Ο γράφος της πρότασης ακολουθεί στο Σχήμα 3.1 [10].

Εφαρμογή συγχώνευσης Word Γράφων

Οι γράφοι των κειμένων συγχωνεύονται ανάλογα με τον τρόπο συγχώνευσης των N-gram γράφων. Λαμβάνοντας υπόψη ότι η συγχώνευση πραγματοποιείται βάσει τροποποίησης των βαρών των ακμών των γράφων, το γεγονός ότι οι κόμβοι περιλαμβάνουν μία ολόκληρη λέξη αντί μίας αλληλουχίας χαρακτήρων συγκεκριμένου μήκους δεν επηρεάζει τη διαδικασία συγχώνευσης.

S. no	First Word	Second Word
1	JJ	NN/NNS
2	RB/RBR/RBS	JJ
3	JJ	JJ
4	NN/NNS	JJ
5	RB/RBR/ RBS	VB/VBD/VBG
6	VB/VBG/VBD	NN/NNS
7	VB/VBG/VBD	JJ/JJR/JJS
8	JJ	VB/VBD/VBG
9	RB/RBR/RBS	RB/RBR/RBS

Πίνακας 3.2: Πίνακας POS μοτίβων

3.2.5 Sentic Patterns

Η σημασιολογική εξαγωγή χαρακτηριστικών κειμένων βασίζεται σε μέρη του λόγου που φέρουν συναισθηματική φόρτιση, όπως επίθετα και επιρρήματα [21]. Αυτά τα μέρη του λόγου χρησιμοποιούνται συνήθως για την έκφραση συναισθημάτων στο γραπτό λόγο [11]. Λόγου χάρη, στην πρόταση "$this\_DT\ was\_VB\ a\_DT\ great\_JJ\ movie\_NN$" η λέξη "$great$" είναι επίθετο και φανερώνει θετικό συναίσθημα. Αντίθετα, οι λέξεις "$this$", "was", "a" και "$movie$" δεν φανερώνουν συναισθηματική φόρτιση ή πολικότητα. Συνεπώς, εδώ λέξεις θεωρούνται χαρακτηριστικά εαν εμπίπτουν σε συγκεκριμένες συντακτικές σημάνσεις (POS tags), όποως επίθετο, επίρρημα, ουσιαστικό και ρήμα.

Χαρακτηριστικά βάσει POS (Part Of Speech) μοτίβων

Οι φράσεις σε ένα κείμενο είναι ιδιαίτερα χρήσιμες για την εξαγωγή πληροφοριών συντακτικής δομής και συμφραζομένων, οι οποίες είναι ιδιαίτερα σημαντικές για την Ανάλυση Συναισθήματος. Στο πίνακα που ακολουθεί παρατίθονται τα σημανιτκότερα POS μοτίβα (POS Patterns). Για παράδειγμα, προσαρτώντας το επίρρημα "$very$" στο ήδη φορτισμένο θετικά επίθετο "$good$" αυξάνεται η ένταση και η πολικότητα του συναισθήματος που εκφράζεται. Αυτή η πληροφορία μπορεί να εμφανιστεί χρήσιμη για την αναγνώριση και κατάταξη συναισθήματος. Επιπρόσθετα, οι φράσεις σε ένα κείμενο μπορούν να αιχμαλωτίσουν το νόημα των συμφραζομένων, όπως "$unpredictable\ story$". Επομένως, στην εξαγωγή χαρακτηριστικών βάσει POS μοτίβων εξάγονται φράσεις δύο λέξεων που συνάδουν με τα μοτίβα του Πίνακα 3.2 [3].

S. no	Relation	Meaning	Example
1	Acomp	adjectival complement	(look,good)
2	Advmod	adverbial complement	(cool,pretty)
3	Amod	adjectival modifier	(performance, poor)
4	Dobj	direct object	(appreciated,actor)
5	Neg	negation modifier	(happy,not)
6	Nsubj	nominal subject	(good, actors)
7	Rcmod	relative clause modifiers	(film, exhilarate)
8	Xcomp	open clause complement	(bored,watching)
9	Cop	Copula	(beautiful, is)
10	Ccomp	clausal complement	(happens,bored)

Πίνακας 3.3: Πίνακας συντακτικών εξαρτήσεων

Σχέσεις Συντακτικής Εξάρτησης

Μία βαθύτερη γλωσσική ανάλυση των συντακτικών σχέσεων αποβαίνει σημαντική για την Ανάλυση Συναισθήματος, και χρησιμοποιείται ευρέως. Οι δενδρικές δομές αναπαράστασης εξαρτήσεων σε μία πρόταση παράγουν πληροφορία συντακτικής εξάρτησης από το κείμενο. Οι *Wiebe et al.* [22] διερεύνησαν την σημασία και την αποδοτικότητα των συντακτικών μοτίβων στον επιτυχή εντοπισμό υποκειμενικότητας, ένα βήμα πριν την αναγνώριση συναισθήματος. Τα μοτίβα εξαρτήσεων που μπορούν να αξιοποιηθούν προς άντληση πλούσιων συναισθηματικά χαρακτηριστικών από ένα κείμενο, παρουσιάζονται στον ακόλουθο Πίνακα 3.3 [3].

Σύνθετα Χαρακτηριστικά

Συνδυάζοντας τα POS μοτίβα και τα χαρακτηριστικά βάσει συντακιτκών εξαρτήσεων προκύπτουν νέα Σύνθετα Γνωρίσματα. Φράσεις που εξάγονται βάσει POS μοτίβων δεν συμπεριλαμβάνουν όλες τις φράσεις που είναι συναισθηματικά φορτισμένες. Ακόμη, η αναγνώριση μερών του λόγου (POS) δεν είναι αρκετή για τον προσδιορισμό των συμφραζομένων και των εξαρτήσεων ανάμεσα σε λέξεις που δεν βρίσκονται σε διαδοχή, κάτι το οποίο πετυχαίνει η αναγνώριση συντακτικών εξαρτήσεων. Για παράδειγμα, η πρόταση "*This movie is very impressive and effective.*" ΄μαρκάρεται΄ στις επιμέρους λέξεις που την απαρτίζουν ως εξής "*This_DT Movie_NN is_VBZ very_RB impressive_JJ and_CC effective_JJ*". Σύμφωνα, λοιπόν, με τα μοτίβα των Μερών του Λόγου, ως συναισθηματικά φορτισμένη φράση θα εξαγόταν η "very impressive". Ωστόσο, υπάρχουν περισσότερες συναισθηματικά φορτισμένες φράσεις που μπορούν να αξιοποιηθούν στα πλαίσια της Ανάλυσης Συναισθήματος και που

μπορούν να εξαχθούν βάσει των μοτίβων Συντακτικών Εξαρτήσεων. Παραδείγματος χάριν, nsubj(impressive, movie), nsubj(effective, movie), cop(impressive, is), advmod(impressive, very), advmod(effective, very) είναι μερικές από αυτές. Συνδυάζοντας τις δύο παραπάνω μεθόδους εξαγωγής χαρακτηριστικών επιτυγχάνεται και ο συνδυασμός του είδους της πληροφορίας που αντλείται από κάθε κείμενο. Συνεπώς, τα Σύνθετα Χαρακτηριστικά αξιοποιούν και τα δύο είδη χαρακτηριστικών που αναφέρθηκαν παραπάνω.

3.2.6 Λεξικό SentiWordNet

Το SentiWordNet Λεξικό αποτελεί μία επέκταση του WordNet Λεξικού. Εκκινώντας, λοιπόν, από το WordNet, το ίδιο αποτελεί ένα 'λεξικό με νόημα' με την έννοια ότι επιχειρεί την οργάνωση λεξιλογικής πληροφορίας όσον αφορά το νόημα των λέξεων, η οποία είναι περισσότερο ευθυγραμμισμένη με την ανθρώπινη αναπαράσταση των λέξεων και των νοημάτων, καθώς και τον τρόπο επεξεργασίας τους από τον ανθρώπινο εγκέφαλο.

WordNet

Το WordNet περιέχει αγγλικά ουσιαστικά, ρήματα, επίθετα και επιρρήματα. Αυτά σχηματίζουν synsets, σύνολα δηλαδή συνωνύμων στα οποία προσαρτώνται περιγραφές όπως και παραδείγματα. Ακόμη, στο λεξικό περιλαμβάνονται σύνδεσμοι ανάμεσα στα synsets που εμφανίζουν μία συγκεκριμένη λεξιλογική ή νοηματική σύνδεση. Παραδείγματος χάριν, τα ουσιαστικά είναι δυνατό να συνδέονται μέσω σχέσεων υπερωνυμίας/υπονυμίας και μερονυμίας/ολονυμίας οι οποίες μπορούν να κληροδοτηθούν. Με τον τρόπο αυτό σχηματίζεται μία ιεραρχία. Υπάρχει ακόμα μία διαφοροποίηση ανάμεσα σε τύπους (κοινά ουσιαστικά) και στιγμιότυπα (άτομα, οντότητες). Τα ρήματα οργανώνονται μέσω σχέσεων τροπωνυμίας, υπερνυμίας και λογικής συνέπειας. Τα επίθετα συνδέονται με τα αντώνυμά τους και τα σχεσιακά επίθετα αναφέρονται στα αντίστοιχα ουσιαστικά. Τα επιρρήματα συντάσσουν το μικρότερο σύνολο από synsets. Προέρχονται κυρίως από επίθετα και συνδέονται με αυτά. Επιπρόσθετα, περιλαμβάνονται ορισμένες σχέσεις ανάμεσα σε διαφορετικά Μέρη του Λόγου. Οι μορφοσημαντικές συνδέσεις (Morphosemantic Links) συνδέουν λέξεις που διαθέτουν την ίδια ρίζα, όπως συμβαίνει με πολλά επιρρήματα και επίθετα. Ακόμη, κάποια ζεύγη ουσιαστικό-ρήμα περιλαμβάνονται ως ρόλοι δράσης. Στην σύγχρονη έκδοση περιλαμβάνονται 82115 διακριτά ουσιαστικά synsets, 13767 synsets ρημάτων, 18156 επιθέτων και 3621 επιρρημάτων, γεγονός που ισοδυναμεί με 117659 synsets στο σύνολο, τα οποία αποτελούνται από 155287 μοναδικές λέξεις. Λαμβάνοντας υπόψη ότι το Oxford Αγγλικό Λεξικό περιλαμβάνει 171476 λέξεις σε χρήση, και Θεωρώντας το σύνολο αυτό ως μία εκτίμηση του μεγέθους της Αγγλικής γλώσσας, το λεξικό WordNet καλύπτει ένα μεγάλο κομμάτι της [13].

Πέρα από το αγγλικό WordNet υπάρχει ένας μεάλος αριθμός εργασιών που στοχεύουν να πραγματοποιήσουν αντίστοιχα συστήματα για παραπάνω από 45 διαφορετικές γλώσσες, όπως και πολυγλωσσικά συστήματα.Σε εφαρμογές Αναγνώρισης και Επεξεργασίας Φυσικής Γλώσσας (NLP) το εν λόγω λεξικό συνιστά μία δημοφιλή πηγή γνώσης για αναγνώριση αμφισημίας λέξεων και υπολογισμού ομοιοτήτων ανάμεσα σε λέξεις. Το πλέον ελκυστικό

POS	ID	PosScore	NegScore	SynsetTerms	Gloss
a	00005599	0.5	0.5	unquestioning#2 implicit#2	being without doubt or reserve; "implicit trust"

Πίνακας 3.4: Παράδειγμα Καταχώρησης στο SentiWordNet

χαρακτηριστικό του είναι το γεγονός ότι προσφέρει διαφοροποιημένες ιεραρχίες νοηματικά οργανωμένων συνόλων λέξεων χωρίς να χωλαίνει στην ευρεία κάλυψη λεξιλογίου και την αξιοπιστία του.

Παρόλα αυτά, μία ολοκληρωμένη ανάλυση μίας φράσης, ενός κειμένου ή συνόλου κειμένων μπορεί να χρειάζεται τον εντοπισμό και την επεξεργασία συναισθημάτων που σχηματίζουν τα συναισθηματικά και πολωμένα προς μία συναισθηματική κατεύθυνση κομμάτια του νοήματός του. Είναι δυνατή η αμφισημία των λέξεων, η μέτρηση του συσχετισμού τους με άλλες, ο προσδιορισμός και η περιγραφή του νοήματος τους με τη χρήση του WordNet. Ωστόσο, ούτως ώστε να αντιμετωπισθούν προβλήματα αναγνώρισης συναισθηματικής πολικότητας το λεξικό χρειάζεται να επεκταθεί με επιπρόσθετη πληροφορία. Εδώ έρχεται το λεξικό SentiWordNet, το οποίο στοχεύει στην επέκταση της εφαρμογής του WordNet στον NLP τομέα σε διαφορετική διάσταση.

SentiWordNet

Ο σκοπός του SentiWordNet είναι η παροχή μίας επέκτασης για το WordNet, τέτοιας ώστε όλα τα synsets να σχετίζονται με ένα αρνητικό, θετικό ή ουδέτερο χαρακτηρισμό. Αυτή η επέκταση χαρακτηρίζει κάθε synset με μία τιμή για καθεμία κατηγορία ανάμεσα στο 0.0 και το 1.0. Το σύνολο των τριών τιμών είναι πάντα 1.0, ούτως ώστε κάθε synset να μπορεί να διαθέτει μη μηδενική τιμή για κάθε συναίσθημα, αφού κάποια synsets μπορεί να είναι θετικά, αρνητικά ή ουδέτερα ανάλογα τα συμφραζόμενα που τα περιβάλλουν. Το πλεονέκτημα χρήσης synsets και όχι όρων είναι ότι προσφέρουν διαφορετική συναισθηματική αξία για καθεμία από τις έννοιες της λέξης, και για το λόγο αυτό οι συναισθηματικές αξίες μία λέξης διαφέρουν ανάλογα με την έννοια της [13]. Στον Πίνακα 3.4 παρατίθεται παράδειγμα καταχώρησης λέξης στο SentiWordNet.

3.3 Επιλεγμένες Μέθοδοι Ταξινόμησης

3.3.1 Naive Bayes

Εν γένει οι Ταξινομητές Bayes αναθέτουν την περισσότερο πιθανή κλάση σε ένα δοθέν παράδειγμα, το οποίο περιγράφεται από ένα διάνυσμα χαρακτηριστικών. Ο Naive Bayes Ταξινομητής, συγκεκριμένα. πραγματοποιεί την άφελή΄ και απλή υπόθεση ότι τα χαρακτηριστικά εισόδου του ταξινομητή είναι ανεξάρτητα μεταξύ τους, δηλαδή ότι ισχύει:

$$\mathbf{P(X|C)} = \prod_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{P(X_i|C)}$$

, όπου $\mathbf{X} = (\mathbf{X_1}, .., \mathbf{X_n})$ είναι ένα διάνυσμα χαρακτηριστικών εισόδου και C μία κλάση.

Παρά το γεγονός ότι η παραπάνω υπόθεση χαρακτηρίζεται ως μη ρεαλιστική, ο ταξινομητής Naive Bayes απαντά ιδιαίτερη επιτυχία στην πράξη και συγκρινόμενος με περισσότερο πολύπλοκες τεχνικές [17]. Στην ουσία, ο ταξινομητής εκπαιδεύεται από τα δεδομένα εκπαίδευσης την υπο συνθήκη πιθανότητα χαρακτηριστικού X_i, δοσμένης της κλάσης με διακριτικό C. Η ταξινόμηση σε μετέπειτα στάδιο πραγματοποιείται εφαρμόζοντας τον κανόνα πιθανοτήτων του Bayes που παρατίθεται παραπάνω, προκειμένου να υπολογιστεί η πιθανότητα της κλάσης C για το συγκεκριμένο στιγμιότυπο χαρακτηριστικών εισόδου και, στη συνέχεια, να πραγματοποιηθεί πρόβλεψη της κλάσης με τη μεγαλύτερη a-posteriori πιθανότητα [8].

Η επιτυχία του naive Bayes όσον αφορά την ύπαρξη εξαρτήσεων ανάμεσα στα χαρακτηριστικά (features) έγκειται στο γεγονός ότι η βελτιστοποίηση όσον αφορά το λάθος ταξινόμησης δεν σχετίζεται απαραίτητα με την ποιότητα της εφαρμογής σε μία πιθανοτική κατανομή. Αντίθετα, ένας βέλτιστος ταξινομητής επιτυχγάνεται αρκεί η εκτιμώμενη και η πραγματική κατονομή να συμπίπτουν στην πλέον πιθανή κλάση [17].

Αφαιρετικά, ο naive Bayes είναι ένα μοντέλο πιθανοτήτων υπο συνθήκη. Δοθέντος ενός διανύσματος χαρακτηριστικών εισόδου $\mathbf{X} = (\mathbf{X_1}, .., \mathbf{X_n})$, όπου X_i για i=1,2,...n ανεξάρτητες μεταβλητές, ο ταξινομητής αναθέτει στο στιγμιότυπο πιθανότητες

$p(C_k|x_1, ..., xn)$ για καθένα από τα k στιγμιότυπα ή για καθεμία από τις κλάσεις C_k.

Το πρόβλημα με την παραπάνω παράσταση είναι ότι εαν ο αριθμός των χαρακτηριστικών n είναι μεγάλος ή εαν ένα χαρακτηριστικό λαμβάνει μεγάλο αριθμό διαφορετικών τιμών, τότε το μοντέλο δεν μπορεί να στηριχτεί στις πιθανότητες. Για το λόγο αυτό, το μοντέλο μπορεί να προσαρμοστεί με τη χρήση του θεωρήματος του Bayes, αποσυντίθοντας την υπο συνθήκη πιθανότητα ως εξής

$$P(C_k|x) = \frac{P(C_k) * P(x|C_k)}{P(x)}$$

,όπου στην ουσία

$$posterior = \frac{prior\ x\ likelihood}{evidence}$$

Πρακτικά, το ενδιαφέρον έγκειται στον ονομαστή του παραπάνω κλάσματος, καθώς ο παρονομαστής δεν εξαρτάται από την κλάση C και οι τιμές των χαρακτηριστικών F_i είναι δοσμένες. Συνεπώς ο παρονομαστής είναι σταθερός. Ο ονομαστής είναι ισοδύναμος με το μοντέλο πιθανοτήτων

$$p(C_k, x_1, ..., xn)$$

το οποίο μπορεί να ξαναγραφεί ως εξής βάσει του κανόνα της αλυσίδας:

$$\begin{aligned} p(C_k|x_1, ..., xn) &= p(x_1|..., xn, C_k) \\ &= p(x_1|..., xn, C_k)p(x_2, ..., x_n, C_k) \\ &= p(x_1|..., xn, C_k)p(x_2|..., x_n, C_k)p(x_3, ..., x_n, C_k) \\ &... \\ &= p(x_1|..., xn, C_k)p(x_2|..., x_n, C_k)...p(x_{n-1}|..., x_n, C_k) \end{aligned}$$

Σε αυτό το σημείο γίνεται φανερή και η άφελής΄ (naive) υπόθεση της ανεξαρτησίας υπο συνθήκη. Υποθέτοντας ότι κάθε χαρακτηριστικό F_i είναι υπο συνθήκη εξαρτώμενη από κάθε

άλλο χαρακτηριστικό F_j για $j \neq i$, δεδομένης της κατηγορίας C, προκύπτει ότι:

, το οποίο ισοδυναμεί:

$$p(C_k \mid x_1, \ldots, x_n) \propto p(C_k, x_1, \ldots, x_n) \propto p(C_k)\ p(x_1 \mid C_k)\ p(x_2 \mid C_k)\ p(x_3 \mid C_k) \cdots \propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,.$$

Επομένως, η υπο συνθήκη κατανομή για τη μεταβλητή κλάσης C είναι η ακόλουθη:

$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

,όπου $Z = p(x)$ ένας συντελεστής κλίμακας που εξαρτάται από τα χαρακτηριστικά. Συνεπώς, αν τα χαρακτηριστικά είναι γνωστά και η ποσότητα Z είναι σταθερή. Ο ταξινομητής naive Bayes συνδυάζει το παραπάνω μοντέλο με ένα κανόνα απόφασης. Έναν κοινό κανόνα αποτελεί ο maximum a posteriori (MAP). Ο αντίστοιχος Bayes ταξινομητής είναι η συνάρτηση που αναθέτει το διακριτικό $\hat{y} = C_k$ της κλάσης για κάποιο k:

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

3.3.2 Decision Tree

Το εν λόγω μοντέλο έχει στόχο την πρόβλεψη της τιμής μίας μεταβλητής-στόχου βασιζόμενο σε διάφορες μεταβλητές εισόδου και μπορεί να αναπαρασταθεί ως μία δενδρική δομή. Σε αυτή τη δομή κάθε εσωτερικός κόμβος αντιστοιχεί σε μία από τις μεταβλητές εισόδου, ενώ οι ακμές παιδιά για καθεμία από τις πιθανές τιμές της συγκεκριμένης μεταβλητής εισόδου. Κάθε κόμβος-φύλλο αναπαριστά τη τιμή της αντίστοιχης μεταβλητής εισόδου, δεδομένων των μεταβλητών εισόδου που αναπαρίστανται από το μονοπάτι από τη ρίζα του δέντρου ως αυτό τον κόμβο-φύλλο. Ένα δέντρο αποφάσεων (decision tree) συνιστά μία απλή αναπαράσταση παραδειγμάτων ταξινόμησης.

Ένα δέντρο αποφάσεων συνιστά ένα δέντρο στο οποίο κάθε εσωτερικός κόμβος, ο οποίος δεν αποτελεί κόμβο-φύλλο, χαρακτηρίζεται από ένα χαρακτηριστικό εισόδου. Οι ακμές που πηγάζουν από τον κόμβο λαμβάνουν τις τιμές των πιθανών τιμών του εν λόγω κόμβου ή του χαρακτηριστικού εξόδου, είτε οδηγούν σε έναν διαφορετικό κόμβο απόφασης για διαφορετικό χαρακτηριστικό εισόδου. Καθένα από τα φύλλα του δέντρου χαρακτηρίζεται με μία κλάση ή την κατανομή πιθανότητας καθεμίας από τις κλάσεις.

Η δημιουργία του δέντρου αποφάσεων μπορεί να πραγματοποιηθεί χωρίζοντας το σύνολο δεδομένων εισόδου σε δύο υποσύνολα βάσει ελέγχου τιμών. Η διαδικασία επαναλαμβάνεται για καθένα από τα υποσύνολα που προκύπτουν αναδρομικά και ονομάζεται αναδρομικός διαμερισμός (recursive partitioning). Η αναδρομή ολοκληρώνεται όταν ένα υποσύνολο σε κάποιον κόμβο έχει την ίδια τιμή με τη μεταβλητή στόχο, ή στην περίπτωση που η διαμέριση του συνόλου δεδομένων δεν προσθέτει καμία αξία στις προβλέψεις. Η διαδικασία αυτή ονομάζεται Top-Down Induction Of Decision Trees (TDIDT) και είναι ένα παράδειγμα ενός άπληστου αλγορίθμου.

Συγκεκριμένα στην εξόρυξη δεδομένων, τα δέντρα αποφάσεων μπορούν να περιγραφούν ως ο συνδυασμός ανάμεσα σε μαθηματικές και υπολογιστικές τεχνικές, ούτως ώστε να βοηθήσουν στην περιγραφή, κατηγοριοποίηση και γενίκευση ενός δοσμένου συνόλου δεδομένων. Τα

δεδομένα, όπως έχει αναφερθεί και παραπάνω, εμφανίζονται στην μορφή:

$$(x, Y) = (x_1, x_2, ..., x_k, Y)$$

Η εξαρτημένη μεταβλητή Y αποτελεί την μεταβλητή στόχο, και την οποία το εν λόγω μοντέλο στοχεύει να κατανοήσει, να ταξινομήσει ή να γενικεύσει. Το διάνυσμα x απαρτίζεται από τις μεταβλητές εισόδου $x_1, x_2, .., x_k$ που αξιοπούνται προς αυτό τον σκοπό.

3.3.3 Multinomial Naive Bayes

Στο μοντέλο Multinomial Bayes, τα διανύσματα χαρακτηριστικών αναπαριστούν τις συχνότητες με τις οποίες συγκεκριμένα γεγονότα έχουν παραχθεί από ένα πολυώνυμο $(p_1, ..., p_n)$ όπου p_i είναι η πιθανότητα του γεγονότος i να εμφανιστεί. Ένα διάνυσμα χαρακτηριστικών $X = (x_1, ...x_n)$ αποτελεί, επομένως, ένα ιστόγραμμα με το x_i να αποτελεί έναν μετρητή του αριθμού των φορών που το γεγονός i παρατηρείται σε ένα συγκεκριμένο στιγμιότυπο. Το μοντέλο αυτό χρησιμοποιείται κυρίως για ταξινόμηση κειμένων, με τα γεγονότα λέξεων να αναπαριστούν την εμφάνιση μίας λέξης σε ένα έγγραφο.

Στο μοντέλο Multinomial Bayes κάθε έγγραφο αποτελείται από μία ταξινομημένη ακολουθία γεγονότων λέξεων, τα οποία αντλούνται από το ίδιο λεξιλόγιο V. Η υπόθεση που πραγματοποιείται εδώ είναι ότι η πιθανότητα κάθε γεγονότος λέξης σε ένα έγγραφο είναι ανεξάρτητη των συμφραζόμενων που πλαισιώνουν την λέξη και της θέσης της στο έγγραφο. Επομένως, κάθε έγγραφο d_i αποτελείται από μία πολυωνυμική κατανομή λέξεων με τόσες ανεξάρτητες δοκιμές όσες και το μήκος του d_i. Έστω, N_{it} ο μετρητής του αριθμού των φορών που η λέξη w_t εμφανίζεται στο έγγραφο d_i. Η πιθανότητα, επομένως, του εγγράφου δεδομένης της κλάσης του προκύπτει από μία πολυωνυμική κατανομή:

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{\mathrm{P}(w_t|c_j;\theta)^{N_{it}}}{N_{it}!}.$$

Οι παράμετροι του συστατικού στοιχείου της εξίσωσης για καθεμία από τις κλάσεις είναι οι πιθανότητες για κάθε λέξη, και γράφονται $\theta_{w_t|c_j} = P(w_t|c_j;\theta)$, όπου $0 \leq \theta_{w_t|c_j} \leq 1$ και $\sum \theta_{w_t|c_j} = 1$. Είναι δυνατό να υπολογιστούν οι εν λόγω παράμετροι από ένα σύνολο ταξινομημένων δεδομένων εκπαίδευσης. Εδώ, η εκτίμηση της πιθανότητας της λέξης w_t στην κλάση c_j είναι

$$\hat{\theta}_{w_t|c_j} = \mathrm{P}(w_t|c_j;\hat{\theta}_j) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N_{it}\mathrm{P}(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N_{is}\mathrm{P}(c_j|d_i)}.$$

,όπου

$$\hat{\theta}_{c_j} = \mathrm{P}(c_j|\hat{\theta}) = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}.$$

η εξίσωση υπολογισμού μέγιστης πιθανοφάνειας[14].

Κεφάλαιο 4

Ανάλυση και Υλοποίηση

Στο παρόν κεφάλαιο παρουσιάζεται η συμβολή της παρούσας Διπλωματικής Εργασίας στο να συνδυάσει διαφορετικές προσεγγίσεις εξαγωγής χαρακτηριστικών κειμένου, και, κατ΄ επέκταση, η προσπάθεια βελτιστοποίησης του μοντέλου των Word Graphs. Εδώ, παρατίθεται η λογική ροή της τροποποίησης του ήδη υπάρχοντα κώδικα του μοντέλου αυτού, όπως και οι λεπτομέρειες υλοποίησής της.

4.1 Ανάλυση

Στη συνέχεια παρουσιάζεται η υλοποίηση της μεθόδου των Word Graphs, όπως σχεδιάστηκε στη Διπλωματική Εργασία της Π. Κιούρτη [10]. Το κομμάτι της μηχανικής μάθησης υλοποιήθηκε με τη βιβλιοθήκη Weka[1], η οποία ενσωματώνει υλοιποίησεις πολλών ταξινομητών καθώς και μεθόδους αξιολόγησής τους. Ακολουθούν οι υλοποιήσεις των τριών συγκρινόμενων μεθόδων εξαγωγής χαρακτηριστικών κειμένου. Οι Υλοποιήσεις των Μεθόδων Bag Of Words, των N-gram γράφων, όπως και της Σύγκρισης των μεθόδων διατηρείται στην παρούσα Εργασία, ενώ τροποποιείται η μέθοδος των Γράφων Λέξεων.

Πιο συγκεκριμένα, στην παρούσα Διπλωματική Εργασία επιχειρείται ο συνδυασμός των μεθόδων εξαγωγής χαρακτηριστικών κειμένου μέσω της σύγκρισης γράφων και της εξαγωγής συντακτικών εξαρτήσεων στο κείμενο. Για το λόγο αυτό αξιοποιήθηκαν οι δείκτες που αναφέρονται στο Κέφαλαιο 3, οι οποίοι δίνονται ήδη ως διάνυσμα εισόδου στη βιβλιοθήκη Weka. Στους δείκτες αυτούς προσαρτώνται νέα χαρακτηριστικά εισόδου που αντιπροσωπεύουν τις συντακτικές εξαρτήσεις που συνήθως εμπεριέχουν συναίσθημα, όπως δίνονται στον Πίνακα 3.3 του Κεφαλαίου 3. Προκειμένου να ποσοτικοποιηθούν οι παραπάνω εξαρτήσεις αξιοποιήθηκε το Λεξικό SentiWordNet. Επιλέχθηκε η μέθοδος των Word Graphs, καθώς η σύσταση γράφων από κόμβους λέξεων καθιστά δυνατή την συντατική ανάλυση του κειμένου βάσει λέξεων-κλειδιά που βρίσκονται σε αυτούς.

[1] http://www.cs.waikato.ac.nz/ml/weka/

4.1.1 Υπάρχουσες Υλοποιήσεις

- Υλοποίηση μεθόδου N-gram Γράφων
  Για την υλοποίηση της εν λόγω μεθόδου χρησιμοποιήθηκε η βιβλιοθήκη JInsect[2]. Στην πλειονότητα χρησιμοποιήθηκαν οι μέθοδοι της κλάσης DocumentNGramGraph, οι οποίες και τροποποιήθηκαν στη συνέχεια για την υλοποίηση της μεθόδου Γράφων Λέξεων.

- Υλοποίηση μεθόδου Γράφων Λέξεων
  Όπως αναφέρθηκε και παραπάνω, επεκτείνοντας τη κλάση DocumentNGramGraph (extend σε γλώσσα Java), η δημιουργία γράφων και η επεξεργασία τους τροποποιήθηκε ούτως ώστε να εξάγονται από το κείμενο λέξεις και όχι ακολουθίες χαρακτήρων σταθερού μήκους n.

- Υλοποίηση μεθόδου Bag Of Words
  Η παρούσα μέθοδος υλοποιήθηκε με τη βοήθεια της βιβλιοθήκης Weka. Σαφέστερα, χρησιμοποιήθηκε ένα φίλτρο που παρέχει η βιβλιοθήκη, το οποίο μετατρέπει ένα κείμενο σε μορφή String σε ένα διάνυσμα λέξεων (StringToWordVector filter). Το διάνυσμα αυτό χρησιμοποιείται μαζί με το χαρακτηριστικό κλάσης για την εκπαίδευση και τον έλεγχο του εκάστοτε ταξινομητή.

- Σύγκριση των Μεθόδων
  Η σύγκριση των μεθόδων πραγματοποιήθηκε σε τρία στάδια.

  Αρχικά , έλαβε χώρα η δημιουργία και η αποθήκευση των γράφων μοντέλων πολικότητας, οι οποίοι δημιουργούνται από τα ίδια κείμενα κριτκής ταινιών. Το πρώτο αυτό στάδιο θεωρήθηκε στάδιο προεπεξεργασίας, και δεν ανήκει στην διαδικασία μηχανικής μάθησης.

  Στη συνέχεια, πραγματοποιήθηκε η δημιουργία και η αποθήκευση αρχείων εκπαίδευσης και ελέγχου (Attribute Relation File Format, ARFF Files). Τα εν λόγω αρχεία συνιστούν στιγμιότυπα εκπαίδευσης αλλά και αξιολόγησης του μοντέλου. Το δεύτερο στάδιο είναι και το πρώτο στάδιο στο οποίο εμπλέκεται η μέθοδος Bag Of Words με τη δημιουργία των αντίστοιχων αρχείων εκπαίδευσης και αξιολόγησης της.

  Τέλος, πραγματοποιήθηκε η δημιουργία του ταξινομητή και η αξιολόγησή του.

4.1.2 Υλοποίηση Τροποποίησης της Μεθόδου των Word Graphs

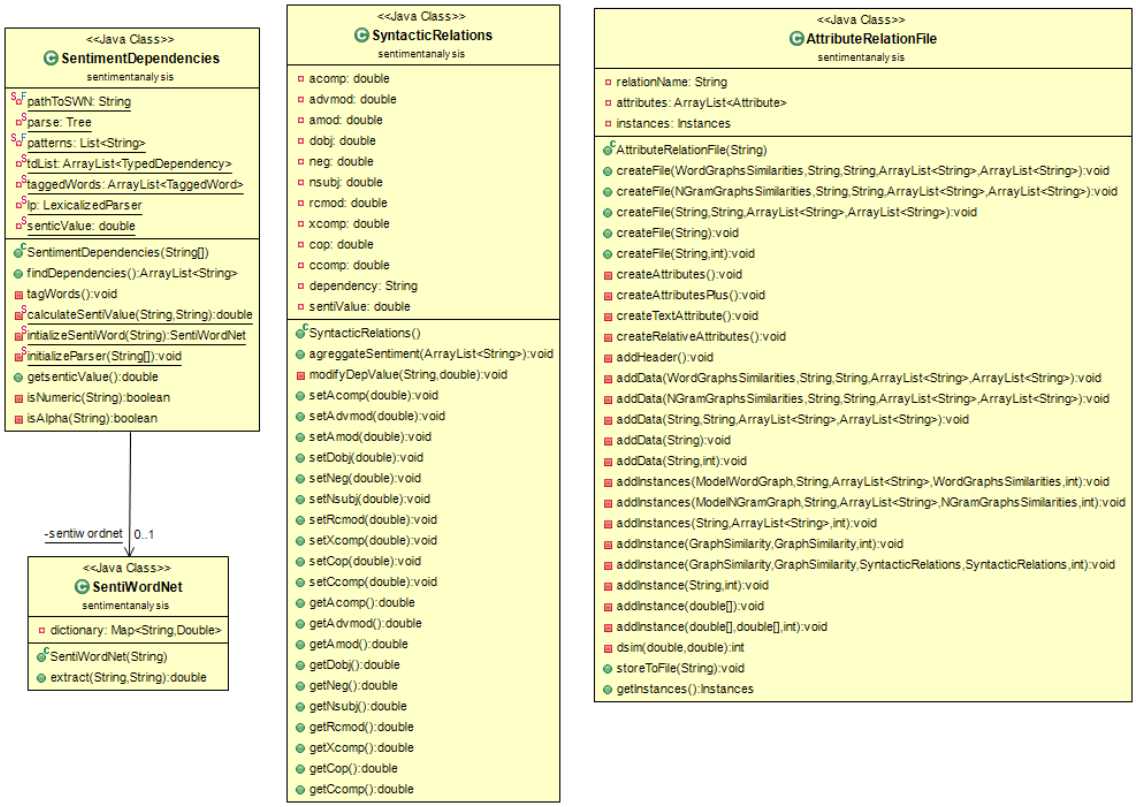
Στο διάνυσμα χαρακτηριστικών εισόδου διατηρούνται οι ήδη υπάρχοντες δείκτες CS, SS, VS και NVS. Οι εν λόγω τιμές προκύπτουν μετά από τη σύγκριση του γράφου που δημιουργείται από το κείμενο εισόδου με καθένα από τους γράφους πολικότητας που κατασκευάζονται στο πρώτο στάδιο σύγκρισης μεθόδων. Συνεπώς, κάθε δείκτης διαθέτει διπλή παρουσία στο διάνυσμα εισόδου, μία για τη θετική και μία για την αρνητική πολικότητα. Σε αυτά προστίθονται τιμές για καθεμία από τις συντακτικές εξαρτήσεις του Πίνακα 3.3. Στις συντακτικές εξαρτήσεις αντιστοιχίζεται μία τιμή που προκύπτει από την ανεύρεση της τιμής συναισθηματικής φόρτισης καθεμίας από τις δύο λέξεις που απαρτίζουν τις εξαρτήσεις σύμφωνα με το

[2]http://sourceforge.net/projects/jinsect/

SentiWordNet. Ακολουθούν οι κλάσεις που προστίθενται στην υπάρχουσα εργασία, καθώς και η κλάση που τροποποιήθηκε προς την ενσωμάτωση των τροποποιημένων κλάσεων στο πρόγραμμα.

Κλάσεις

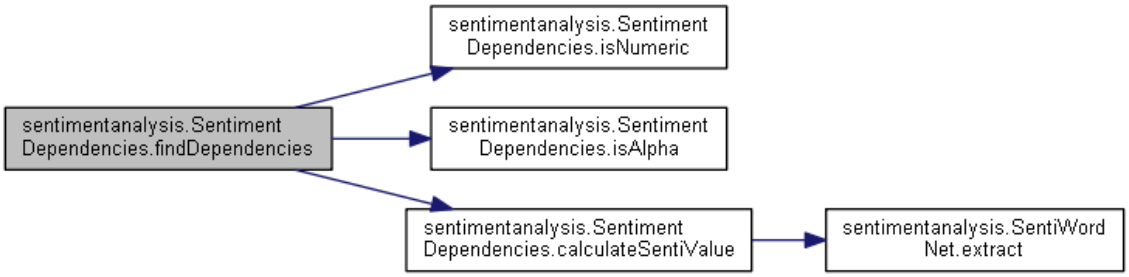
Ο κώδικας των παρακάτων κλάσεων παρατίθεται στο Παράρτημα Α΄.



Σχήμα 4.1: UML Διάγραμμα των κλάσεων που τροποποιήθηκαν

1. SentimentDependencies

   Η κλάση SentimentDependencies δημιουργήθηκε για την ανεύρεση συντακτικών εξαρτήσεων σε μία πρόταση. Η πρόταση προς ανάλυση δίνεται ως όρισμα στον κονστράκτορα της κλάσης (Constructor) και στη συνέχεια βάσει της μεθόδου findDependencies() οι λέξεις χαρακτηρίζονται βάσει των συντακτικών εξαρτήσεων στις οποίες συμμετέχουν. Η μέθοδος tagWords() αναθέτει σε καθεμία από τις λέξεις της πρότασης τον χαρακτηρισμό του Μέρους του Λόγου που η καθεμία αποτελεί (βλ. Παράρτημα Β΄), ενώ η calculateSentiValue() υπολογίζει την ΄τιμή Συναισθήματος΄ που χαρακτηρίζει την δυάδα λέξεων που συνιστά μία συντακτική εξάρτηση. Η εν λόγω τιμή υπολογίζεται βάσει της απλής συνισταμένης των τιμών των λέξεων που προκύπτουν από το SentiWordNet Λεξικό, το οποίο προσπελάζεται με τη κλάση SentiWordNet (Σχήμα 4.2).

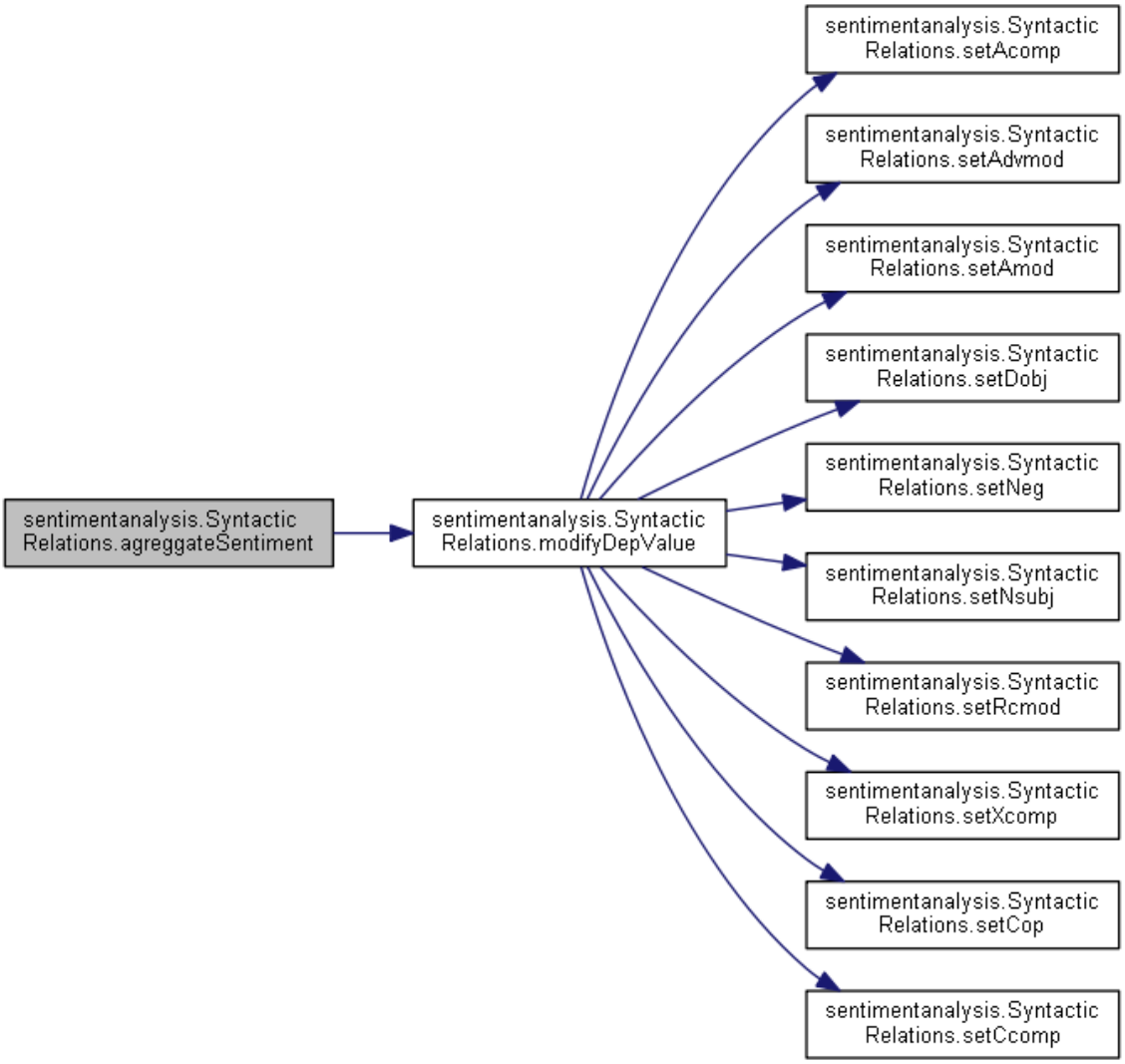


Σχήμα 4.2: Γράφος κλήσεων της κλάσης SentimentDependencies

2. SyntacticRelations
   Η παρούσα κλάση αρχικοποιεί και ανανεώνει τις τιμές καθενός από τα χαρακτηριστικά εισόδου που αντιστοιχούν σε συντακτικές εξαρτήσεις (Σχήμα 4.3). Δημιουργείται ένα στιγμιότυπο της κλάσης για τις αρνητικές και ένα για τις θετικές συντακτικές εξαρτήσεις κάθε κειμένου.

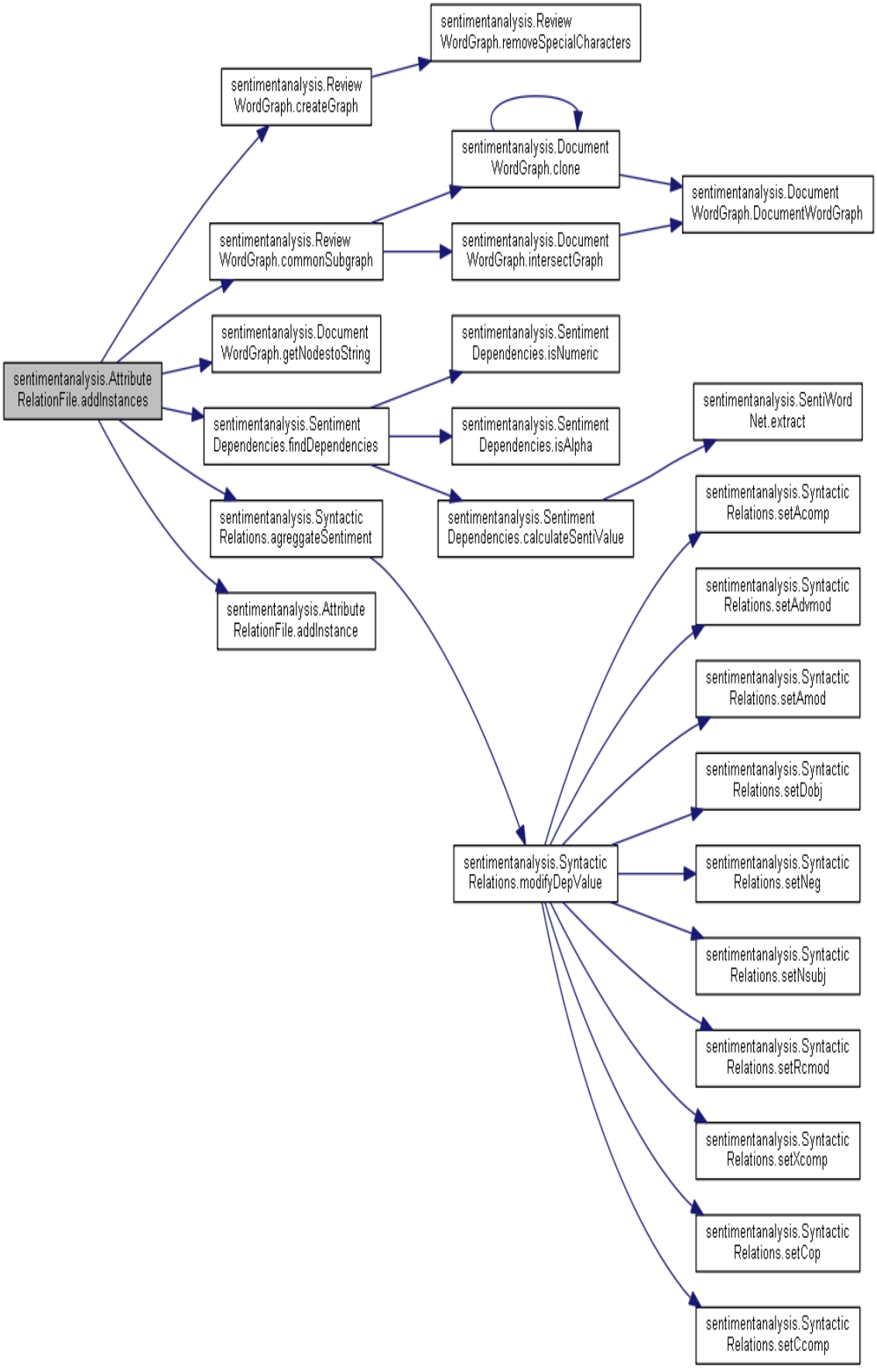
3. SentiWordNet
   Η κλάση SentiWordNet πραγματοποιεί την μετάβαση της αναπαράστασης του SentiWordNet από ένα txt αρχείο σε μία δομή δεδομένων προσπελάσιμη από κώδικα Java, υπολογίζοντας ταυτόχρονα μία τιμή Συναισθήματος για καθεμία από τις λέξεις που περιέχονται σε αυτό σύμφωνα με τις υπάρχουσες τιμές στο Λεξικό (Σχήμα 4.4).

4. AttributeRelationFile
   Η κλάση AttributeRelationFile προϋπήρχε. Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία των ARFF αρχείων εκπαίδευσης και αξιολόγησης του ταξινομητή και του μοντέλου εν γένει. Στον κώδικά της προστέθηκαν τα επιπλέον χαρακτηριστικά εισόδου. Ακόμη, προστίθεται η εξής λογική. Κατά τον υπολογισμό των χαρακτηριστικών εισόδου δημιουργούνται οι κοινοί υπογράφοι του κειμένου κριτικής που εξετάζεται με καθένα από τους Γράφους Μοντέλα Πολικότητας (Model Graphs). Μετέπειτα, το κείμενο διατρέχεται προκειμένου να ανευρεθούν οι λέξεις που συντάσσουν είτε τον θετικό είτε τον αρνητικό κοινό υπογράφο. Στην περίπτωση που βρεθεί μία τέτοια λέξη, η πρόταση που την περιέχει αναλύεται συντακτικά. Εδώ, λοιπόν, γίνεται η υπόθεση ότι η πρόταση που περιέχει λέξη που ανήκει σε Γράφο Πολικότητας θα περιέχει πιθανώς λέξεις και συντακτικές εξαρτήσεις που φανερώνουν συναισθηματική φόρτιση ή δήλωση (Σχήμα 4.5). Με το τρόπο αυτό υπολογίζονται 20 νέες τιμές οι οποίες αντιστοιχούν στις 10 συντακτικές εξαρτήσεις για καθεμία από τις πολικότητες συναισθήματος. Στον αλγόριθμο της βιβλιοθήκης Weka, εν τέλει, επιλέγεται να δοθεί η απλή συνισταμένη των δύο τιμών της εκάστοτε εξάρτησης για τη θετική και την αρνητική πολικότητα. Επιλέγεται, λοιπόν, να δοθούν ως επιπλέον χαρακτηριστικά εισόδου 10 νέες τιμές.



Σχήμα 4.3: Γράφος κλήσεων της κλάσης SyntacticRelations

Σχήμα 4.4: Γράφος κλήσεων της κλάσης SentiWordNet



Σχήμα 4.5: Γράφος κλήσεων της τροποποιημένης μεθόδου addInstances()

Κεφάλαιο 5

Αξιολόγηση

Στο παρόν κεφάλαιο διερευνάται η απόδοση της νέας μεθόδου συγκριτικά με τις μεθόδους των Bag Of Words και N-gram Γράφων. Τα δεδομένα εκπαίδευσης και ελέγχου αφορούν κριτικές ταινιών. Οι μέθοδοι συγκρίνονται στη βάση της ακρίβειας ταξινόμησης, ενώ, παράλληλα, παρατίθενται δείκτες μέτρησης της απόδοσης της κάθε μεθόδου συνολικά.

5.1 Αξιολόγηση Νέας Υβριδικής Μεθόδου

5.1.1 Δεδομένα

Για την εκπαίδευση και αξιολόγηση του συστήματος επιλέχθηκαν τα ίδια δεδομένα με τα δεδομένα που χρησιμοποιήθηκαν για την αξιολόγηση του μοντέλου των Γράφων Λέξεων. Σαφέστερα, χρησιμοποιήθηκαν τα δεδομένα της βάσης IMDB[1]. Τα εν λόγω δεδομένα περιλαμβάνουν 12500 θετικές και 12500 αρνητικές κριτικές. Στο σύνολο, δηλαδή, διατίθονται 50000 κριτικές ταινιών προκειμένου να συνταχθούν σύνολα εκπαίδευσης και ελέγχου των μοντέλων. Τα κείμενα κριτικών οργανώνονται σε θετικές και αρνητικές κριτικές ταινιών. Το κριτήριο διαχωρισμού αποτελεί η βαθμολογία (rating) του χρήστη που συντάσσει το εκάστοτε κείμενο κριτικής. Για την ακρίβεια, τα αρχεία που συνοδεύονται από βαθμολογία που κυμαίνεται ανάμεσα στο 7 και το 10 κατατάσσονται στα θετικά κείμενα κριτικής. Αντίστοιχα, για βαθμολογία από 0 εως 4 το κείμενο κατατάσσεται στα αρνητικά κείμενα κριτικής. Η πληροφορία αυτή αναγράφεται στο τίτλο του κάθε αρχείου κειμένου συνοδευόμενη από ένα μοναδικό αναγνωριστικό αυτού (id).

5.1.2 Παράμετροι προσδιορισμού Μοντέλων

Καθένα από τα μοντέλα που εξατάζονται μπορεί να παραμετροποιηθεί ανάλογα τις ανάγκες των διαδικασιών ελέγχου, αλλά και τη μορφή που εμφανίζουν τα κείμενα που εξετάζονται. Στην υλοποίηση των μεθόδων που εξετάζεται εδώ, οι μεταβλητές του συστήματος προς παραμετροποίηση είναι οι εξής:

[1] http://ai.stanford.edu/amaas/data/sentiment/

- training reviews: το πλήθος των κειμένων κριτικής που απαρτίζουν το σύνολο εκπαίδευσης του μοντέλου.
- test reviews: το πλήθος των κειμένων κριτικής που απαρτίζουν το σύνολο ελέγχου του μοντέλου.
- positive rate: το ποσοστό των κειμένων κριτικής που επιλέγονται από το σύνολο των θετικών κειμένων κριτικής.
- classifier: ο ταξινομητής της βιβλιοθήκης Weka που επιλέγεται
- minPosRating: το κάτω όριο βαθμολογίας που διαθέτουν τα κείμενα κριτικής που επιλέγονται από το σύνολο θετικών κριτικών ταινιών.
- maxPosRating: το άνω όριο βαθμολογίας που διαθέτουν τα κείμενα κριτικής που επιλέγονται από το σύνολο θετικών κριτικών ταινιών.
- minNegRating: το κάτω όριο βαθμολογίας που διαθέτουν τα κείμενα κριτικής που επιλέγονται από το σύνολο αρνητικών κριτικών ταινιών
- maxNegRating :το άνω όριο βαθμολογίας που διαθέτουν τα κείμενα κριτικής που επιλέγονται από το σύνολο αρνητικών κριτικών ταινιών.
- remove: λογική παράμετρος που καθορίζει την αφαίρεση ή όχι του κοινού υπογράφου των δύο γράφων πολικότητας.
- preprocess: λογική παράμετρος που καθορίζει την αφαίρεση ειδικών χαρακτήρων του κειμένου προτού ενσωματωθεί σε γράφο πολικότητας.

Όσον αφορά τα επιμέρους μοντέλα:

N-gram Graphs

- graph reviews: το πλήθος των κριτικών ταινιών που απαρτίζουν καθένα από τους γράφους πολικότητας.
- n: το σταθερό μήκος ακολουθίας χαρακτήρων που εξάγονται για καθένα από τους κόμβους των N-gram γράφων.
- D_{win}: το μήκος παραθύρου.

Word Graphs

- graph reviews: το πλήθος των κριτικών ταινιών που απαρτίζουν καθένα από τους γράφους πολικότητας.
- D_{win}: το μήκος παραθύρου.

5.1.3 Καθορισμός τιμών παραμέτρων

Στην προϋπάρχουσα εργασία της Π. Κιούρτη [10] πραγματοποιείται ανάλυση ευαισθησίας όσον αφορά τις παραμέτρους που καθορίζουν την υλοποίηση καθενός από τα εξεταζόμενα μοντέλα. Όσον αφορά την πραγματοποίηση όλων των προσομοιώσεων οι βέλτιστες τιμές των αντίστοιχων παραμέτρων παρατίθενται στη συνέχεια:

- minPosRating = 7
- maxPosRating = 10
- minNegRating = 0
- maxNegRating = 4
- positive rate = 50
- remove = true
- shuffle = true
- preprocess = false
- classifier = weka.classifiers.bayes.NaiveBayes

Όσον αφορά τα εκάστοτε μοντέλα Ανάλυσης Συναισθήματος συμπεράθηκαν τα εξής:

Bag Of Words

- training reviews = 2000
- test reviews = 1000
- shuffle = true
- preprocess = true
- classifier = weka.classifiers.bayes.NaiveBayesMultinomial

N-gram Graphs

- $D_{win} = n = 4$
- graph reviews = 800
- training reviews = 2000
- test reviews = 1000
- shuffle = true
- remove = true

- preprocess = true
- classifier = weka.classifiers.bayes.NaiveBayes

Word Graphs

- $D_{win} = 6$
- graph reviews = 4000
- training reviews = 2000
- test reviews = 1000
- shuffle = true
- remove = true
- preprocess = true
- classifier = weka.classifiers.bayes.NaiveBayes

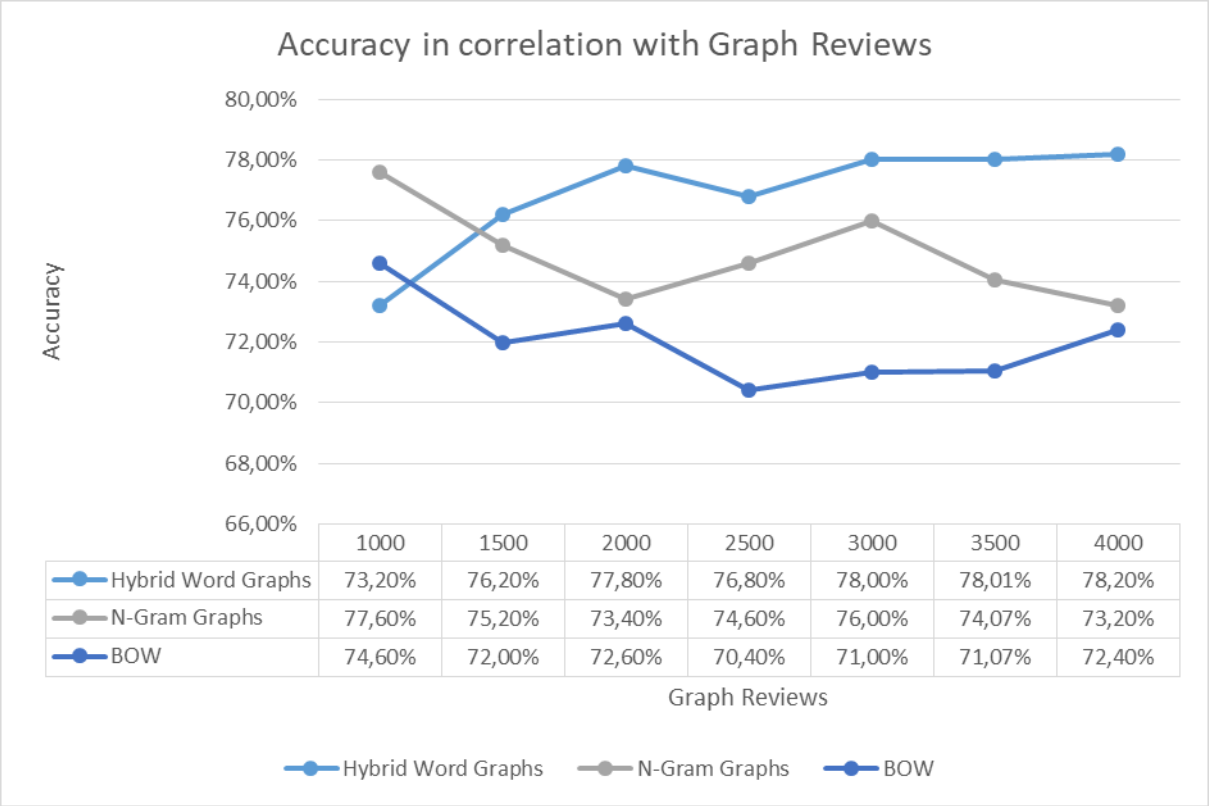
Λαμβάνοντας υπόψιν τις τροποποιήσεις που πραγματοποιήθηκαν στο μοντέλο των Γράφων Λέξεων, θα προαγματοποιηθεί ανάλυση ευαισθησίας των παρακάτω παραμέτρων του συστήματος στα πλαίσια της μεθόδου των Γράφων Λέξεων. Οι παράμετροι που επιλέχθηκαν είναι οι graph reviews, training reviews και test reviews. Η παράμετρος D_{win} θεωρείται ότι δεν επηρεάζει την σημασία των νέων χαρακτηριστικών εισόδου αφού η συντακτική ανάλυση των κειμένων κριτικής πραγματοποιείται στο αρχικό κείμενο βάσει παρουσίας λέξεων στον κοινό υπογράφο, χωρίς επομένως να έχει σημασία η σχετική θέση της κάθε λέξης ως προς τις άλλες στον γράφο. Ακόμα, οι παράμετροι shuffle, remove, preprocess δεν επηρεάζουν τα νέα χαρακτηριστικά εισόδου, οπότε διατηρούνται αφού τα ποϋπάρχοντα χαρακτηριστικά εισόδου διατηρούνται επίσης.

5.2 Αποτελέσματα Προσομοιώσεων

Εδώ, πραγματοποιείται σύγκριση ανάμεσα στην προτεινόμενη Μέθοδο και τις baseline Μεθόδους.

5.2.1 Μεταβολή πλήθους Graph Reviews

Μεταβάλλοντας το πλήθος των Κριτικών Ταινιών που απαρτίζουν τους γράφους πολικότητας, πραγματοποιείται παρατήρηση της μεταβολής της ακρίβειας ταξινόμησης των κειμένων σε καθεμία από τις δύο καθορισμένες πολικότητες συναισθήματος. Αναλυτικότερα, στο παρακάτω γράφημα παρατηρείται η εν λόγω μεταβολή επιλέγοντας πληθικότητες των Graph Reviews στο διάστημα [1000, 4000] με βήμα 500 (Σχήμα 5.1). Το πλήθος των Train Reviews και των Test



Σχήμα 5.1: Μεταβολή Ακρίβειας Ταξινόμησης συναρτήσει του πλήθους των Graph Reviews

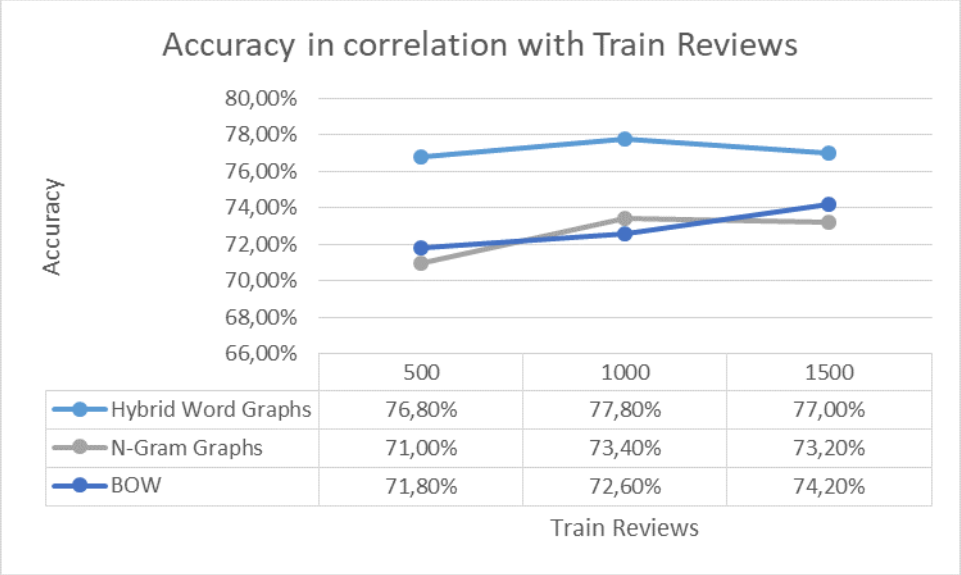
Reviews διατηρήθηκε σταθερό στις τιμές 1000 και 500 αντίστοιχα, ενώ ως ταξινομητής επιλέχθηκε ο Naive Bayes.

Παρατηρείται ότι για μικρό πλήθος κειμένων κριτικής ταινιών που απαρτίζουν τον καθένα από τους γράφους πολικότητας η ακρίβεια ταξινόμησης της προτεινόμενης μεθόδου βρίσκεται χαμηλότερα από την ακρίβεια ταξινόμησης των μεθόδων των N-gram Γράφων και Bag Of Words. Αυξάνοντας, ωστόσο, το πλήθος του μεγέθους των γράφων πολικότητας η ακρίβεια της υβριδικής μεθόδου των Γράφων Λέξεων αυξάνεται σημαντικά και ξεπερνάει την ακρίβεια των άλλων δύο μεθόδων. Τοπικά μέγιστα και ελάχιστα παρουσιάζονται, κυρίως, λόγω του θορύβου που εμπεριέχουν τα εξεταζόμενα δεδομένα, παρά την εξομάλυνση της ακρίβειας ταξινόμησης μέσα από την εκτέλεση κάθε πειράματος πολλαπλές φορές και την εύρεση του μέσου όρου των τιμών.

Η συμπεριφορά που σημειώνεται εδώ μπορεί να δικαιολογηθεί από το γεγονός ότι η συντακτική ανάλυση που έχει ενσωματωθεί στην επεξεργασία κειμένων και την εξόρυξη των χαρακτηριστικών τους βασίζεται στον κοινό υπογράφο του κάθε γράφου πολικότητας και του γράφου του εξεταζόμενου κειμένου. Όσο μικρότερος είναι, λοιπόν, ο κοινός υπογράφος τόσο μεγαλύτερος είναι ο θόρυβος που εισάγεται στα χαρακτηριστικά εισόδου του αλγορίθμου βαθιάς μάθησης. Η μέθοδος των τροποποιημένων γράφων λέξεων κατέχει υπεροχή απόδοσης ως προς τις άλλες δύο μεθόδους.

5.2.2 Μεταβολή πλήθους Train Reviews

Πραγματοποιώντας προσομοιώσεις με διαφορετικό πλήθος αρχείων εκπαίδευσης, επιχειρείται να αξιολογηθεί η επίδραση του πλήθους των κειμένων εκπαίδευσης στην ακρίβεια ταξινόμησης του αλγορίθμου. Εδώ, επιλέχθηκαν πληθικότητες στο διάστημα [500, 1500] με βήμα 500, ενώ το πλήθος των Graph Reviews και των Test Reviews διατηρήθηκε σταθερό στις 2000 και 500 κριτικές ταινιών αντίστοιχα. Όπως είναι έκδηλο από το Σχήμα 5.2, επιβεβαιώνεται η

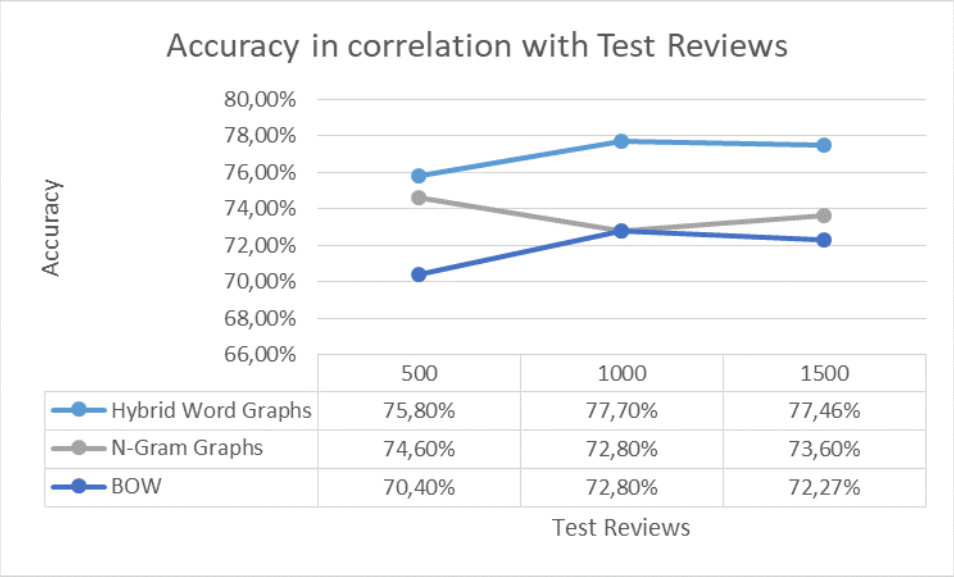


Σχήμα 5.2: Μεταβολή Ακρίβειας Ταξινόμησης συναρτήσει του πλήθους των Train Reviews

υπεροχή σε ακρίβεια της νέας μεθόδου από τις baseline Μεθόδους. Ωστόσο, και αναφορικά ως προς τη Υβριδική νέα Μέθοδο γράφων, παρατηρείται ένα τοπικό μέγιστο για πλήθος Train Reviews ίσο με 1000. Η μικρή απόκλιση από την ακρίβεια της Μεθόδου για πλήθος Train Reviews ίσο με 1500, και λαμβάνοντας υπόψιν το θόρυβο που εμπεριέχεται στα εξεταζόμενα δεδομένα οδηγεί στο συμπέρασμα ότι από μία τιμή-κατώφλι του πλήθους Train Reviews και πάνω επιτυγχάνεται ικανοποιητική από άποψη βελτιστοποίησης τιμή ακρίβειας. Το τελευταίο στηρίζεται και στη λογική επαγωγή ότι όσα περισσότερα κείμενα εκπαίδευσης αναλυθούν και αξιοποιηθούν, τόσο καλύτερο επίπεδο εκπαίδευσης θα επιτευχθεί για τον αλγόριθμο, και, επομένως, μεγαλύτερη ακρίβεια ταξινόμησης συναισθήματος.

5.2.3 Μεταβολή πλήθους Test Reviews

Αντίστοιχα, πραγματοποιήθηκε Ανάλυση Ευαισθησίας για το πλήθος των Test Reviews. Το πλήθος των κριτικών κυμάνθηκε στο διάστημα [500, 1500] με βήμα 500. Όπως και παραπάνω, οι τιμές των Graph και Train Reviews διατηρήθηκαν σταθερές και ίσες με 2000 και 1000 αντίστοιχα. Παρατηρείται από το διάγραμμα του Σχήματος 5.3 ότι όσον αφορά την ακρίβεια ταξινόμησης, αποκλίσεις στο σύνολο των Test Reviews δεν προκαλεί ιδιαίτερα μεγάλες αποκλίσεις στην ακρίβεια του αλγορίθμου. Εδώ, πραγματοποιείται η υπόθεση ότι οι εν λόγω αποκλίσεις οφείλονται κατα κύριο λόγο στον θόρυβο που εμπεριέχουν τα εξεταζόμενα



Σχήμα 5.3: Μεταβολή Ακρίβειας Ταξινόμησης συναρτήσει του πλήθους των Test Reviews

δεδομένα, καθώς και τη διακύμανση του ποσοστού των εξεταζόμενων χαρακτηριστικών κειμένου που εξετάζονται στο προπαρασκευαστικό στάδιο της κάθε Μεθόδου. Και αυτό επειδή το στάδιο της εκπαίδευσης του αλγορίθμου έχει ολοκληρωθεί, και, επομένως, η διαδικασία ταξινόησης του κάθε κειμένου είναι η ίδια.

Συνεπώς, θεωρείται δόκιμο η επιλογή ενός ικανοποιητικά μεγάλου συνόλου κριτικών ταινών προς αξιολόγηση του αλγορίθμου, ούτως ώστε να εξαλειφθεί όσο το δυνατόν ο θόρυβος που εμπεριέχουν τα δεδομένα.

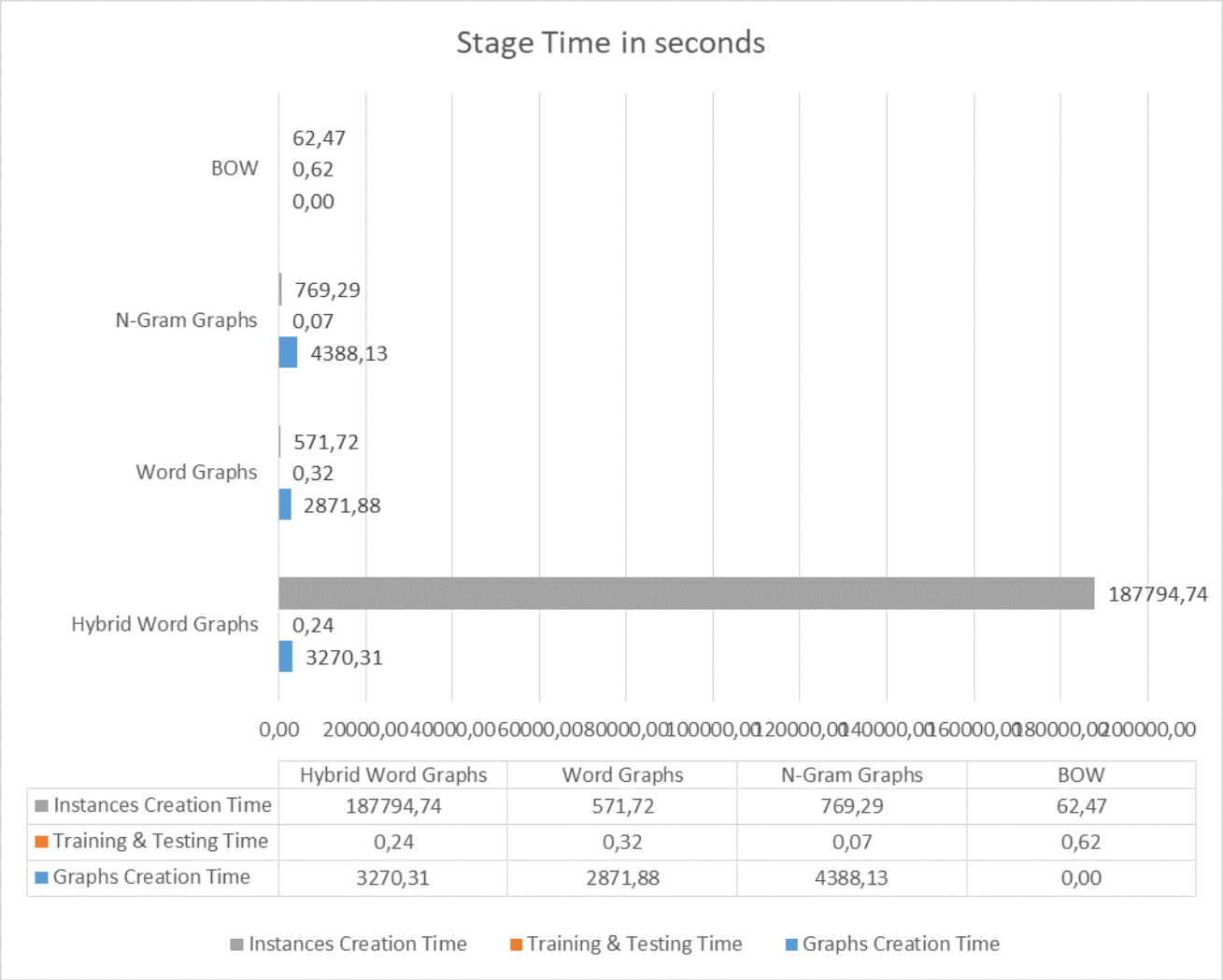
5.2.4 Σύγκριση Χρόνων Εκτέλεσης

Σημαίνον κομμάτι αξιολόγησης της προτεινόμενης Μεθόδου συνιστά ο Χρόνος Εκτέλεσης της. Η νέα Υβριδική Μέθοδος συγκρίνεται στη βάση του χρόνου με την προϋπάρχουσα Μέθοδο, όπως και με τις baseline Μεθόδους. Οι παράμετροι προσομοιώσης όσον αφορά το πλήθος κριτικών ταινιών ορίζονται ως εξής:

- $GraphReviews = 2000$
- $TrainReviews = 1000$
- $TestReviews = 800$

Οι υπόλοιπες παράμετροι διατηρούνται σταθερές όπως ορίστηκαν παραπάνω.

Όσον αφορά το Σχήμα 5.4 τα στάδια που απεικονίζονται αντιστοιχούν στα εξής στάδια της διαδικασίας Βαθιάς Μάθησης του Αλγορίθμου:



	Hybrid Word Graphs	Word Graphs	N-Gram Graphs	BOW
Instances Creation Time	187794,74	571,72	769,29	62,47
Training & Testing Time	0,24	0,32	0,07	0,62
Graphs Creation Time	3270,31	2871,88	4388,13	0,00

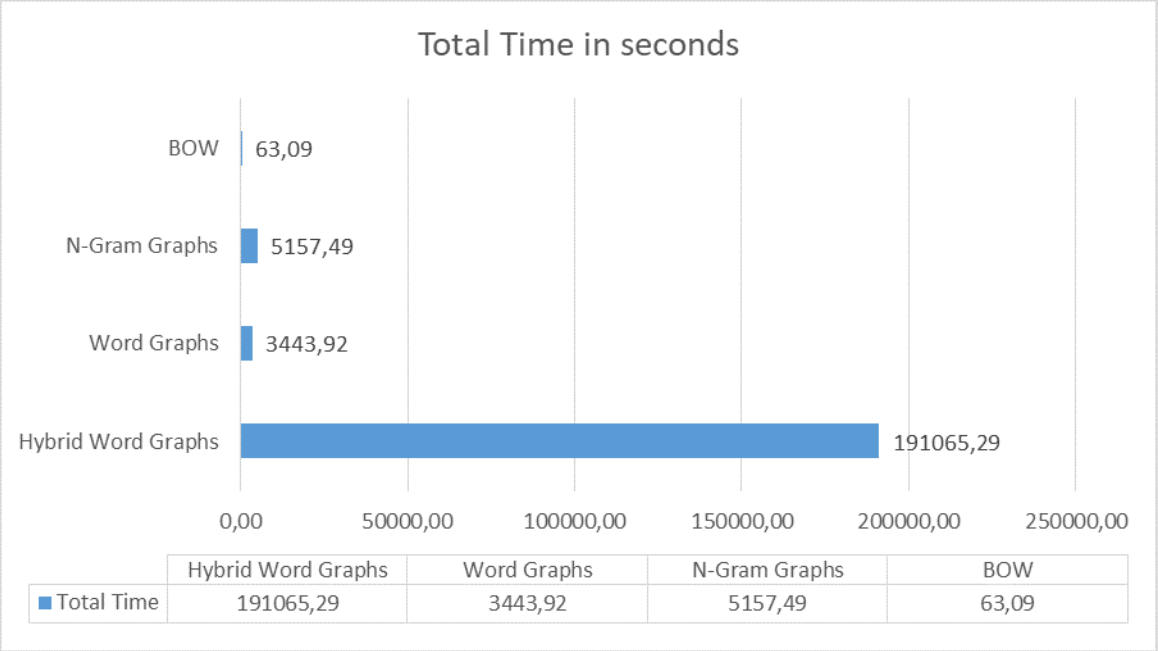
Σχήμα 5.4: Χρόνος κάθε σταδίου για καθεμία από τις Μεθόδους

1. Graph Creation Time: Συνιστά το στάδιο δημιουργίας των γράφων Μοντέλων Πολικότητας.

2. Instances Creation TimeΑποτελεί το στάδιο δημιουργίας των ARFF Αρχείων εκπαίδευσης και αξιολόγησης του Αλγορίθμου

3. Training & Testing Time: Συνιστά το στάδιο δημιουργίας και αξιολόγησης του ταξινομητή του Αλγορίθμου

Όπως είναι φανερό ο χρόνος επεξεργασίας των κειμένων για τη νέα Μέθοδο είναι περίπου εξαπλάσιος του αντίστοιχου χρόνου της βασικής Μεθόδου των Γράφων Λέξεων που τροποποιήθηκε. Το γεγονός αυτό ήταν αναμενόμενο, καθώς η εξαγωγή της συντακτικής πληροφορίας και της φόρτισής της υλοποιήθηκαν με την προσπέλαση κάθε εξεταζόμενου κειμένου κριτικής ταινίας και την συντακτική ανάλυση καθεμίας πρότασης που περιείχε λέξη που βρισκόταν ταυτόχρονα και στον κοινό υπογράφο του γράφου του κειμένου με τον γράφο Μοντέλο Πολι-



Σχήμα 5.5: Συνολικός Χρόνος για καθεμία από τις Μεθόδους

κότητας. Επομένως, είναι λογικό να απαιτείται αρκετά περισσότερος χρόνος για την δημιουργία των ARFF Αρχείων. Η λογική πίσω από αυτή την επιλογή είναι το εγχείρημα επίτευξης μεγαλύτερης ακρίβειας ταξινόμησης θυσιάζοντας χρόνο δημιουργίας αρχείων εκπαίδευσης και αξιολόγησης του αλγορίθμου. Η επιλογή αυτή, ωστόσο, επηρεάζει και τον συνολικό χρόνο της προτεινόμενης Μεθόδου.

5.2.5 Μεταβολή Ταξινομητή

Επικεντρώνοντας την προσοχή στην επιλογή του κατάλληλου ταξινομητή, εξετάζονται οι τρεις πιθανές επιλογές:

- Naive Bayes
  Έχει αποδειχθεί ότι συνιστά τον βέλτιστο ταξινομητή για το ήδη υπάρχον μοντέλο. Μετά την πραγματοποίηση μερικών δοκιμαστικών προσομοιώσεων η υπόθεση αυτή επιβεβαίωνεται όσον αφορά την νέα προτεινόμενη Μέθοδο.

- Multinomial Naive Bayes
  Όπως έχει αποδειχθεί[10], τα χαρακτηριστικά εισόδου της ήδη υπάρχουσας μεθόδου δεν ακολουθούν πολυωνυμική κατανομή. Ωστόσο, το ίδιο ισχύει και για τα χαρακτηριστικά εισόδου που προστέθηκαν στην παρούσα Διπλωματική Εργασία. Συνεπώς, συμπεραίνεται ότι ο ταξινομητής Multinomial Naive Bayes δεν συνιστά τον βέλτιστο για την νέα μέθοδο.

- Decision Tree
  Αντίστοιχα, παρατηρώντας τα χαρακτηριστικά εισόδου της νέας Μεθόδου ως σύνο-

λο, αυτά εμφανίζουν τη διαχωρισιμότητα που απαιτεί ο ταξινομητής αυτός. Ωστόσο, διεξάγοντας δοκιμαστικές προσομοιώσεις η ακρίβεια ταξινόμησης που είναι δυνατό να επιτευχθεί βάσει αυτού του ταξινομητή δεν προσεγγίζει την αντίστοιχη του ταξινομητή Naive Bayes.

Πρέπει εδώ να σημειωθεί ότι οι αλλαγές στον υπάρχοντα κώδικα όσον αφορά την διεξαγωγή προσομοιώσεων αξιοποιώντας άλλους ταξινομητές δεν διατηρήθηκαν, καθώς δεν θεωρήθηκαν αξιοποιήσιμες.

Κεφάλαιο 6

Συμπεράσματα

Στο παρόν κεφάλαιο παρουσιάζεται σχολιασμός των αποτελεσμάτων των προσομοιώσεων του Κεφαλαίου 5, όπως επίσης τα συμπεράσματα που εμπίπτουν από αυτές. Αποφαίνεται, δηλαδή, κατα πόσο η νέα υβριδική προσέγγιση της μεθόδου των γράφων λέξεων υπερέχει των baseline μεθόδων. Τέλος, παρουσιάζονται μελλοντικές επεκτάσεις του θέματος με το οποίο ασχολήθηκε η Διπλωματική Εργασία.

6.1 Συμπεράσματα Προσομοιώσεων

Τα αναλυτικά αποτελέσματα των προσομοιώσεων που πραγματοποιήθηκαν όσον αφορά την υβριδική νέα μέθοδο Ανάλυσης Συναισθήματος Κειμένων παρουσιάστηκαν εκτενώς στο Κεφάλαιο 5. Σε ένα εποπτικό επίπεδο, η νέα Μέθοδος σε σύγκριση με τις baseline Μεθόδους κατέχει σημαντικό προβάδισμα. Σαφέστερα, είναι προφανές ότι για μεγάλο πλήθος κειμένων κριτικής που απαρτίζουν τον γράφο πολικότητας, η νέα Υβριδική Μέθοδος κατέχει μεγάλο προβάδισμα ακρίβειας ταξινόμησης έναντι των baseline Μεθόδων, δηλαδή των N-gram Γράφων και του BOW.

Γίνεται, επομένως, αντιληπτό ότι είναι δυνατό να αξιοποιηθούν συνδυαστικά η φιλοσοφία της ύπαρξης ή της απουσίας οντοτήτων από το κείμενο (είτε συντακτικών εξαρτήσεων στην προτεινόμενη Μέθοδο είτε λέξεων στην μέθοδο Bag Of Words) και της στατιστικής σύγκρισης της μεθόδου των N-gram και Word Γράφων, διατηρώντας υψηλή την απόδοση ταξινόμησης συναισθήματος. Ο συνδυασμός αυτός μπορεί να αξιοποιηθεί σε κείμενα ποικίλης μορφολογίας και περιεχομένου κατα μήκος ενός ευρέως φάσματος.

Όσον αφορά το επιλεγμένο Dataset κειμένων κριτικής, διατυπώνεται η παρατήρηση ότι εισάγεται επιπρόσθετος θόρυβος στα χαρακτηριστικά εισόδου που πλαισιώνει την ουσιαστική πληροφορία για την συναισθηματική φόρτιση των κειμένων. Παρατηρώντας τα ARFF Αρχεία Εκπαίδευσης και Αξιολόγησης της νέας Υβριδικής Μεθόδου παρατηρείται ότι τα περισσότερα χαρακτηριστικά κάθε διανύσματος εισόδου είναι μηδενικά. Συμπεραίνεται, λοιπόν, ότι στην πλειονότητα των κειμένων δεν απαντώνται όλες οι εξεταζόμενες συντακτικές εξαρτήσεις, γεγονός αναμενόμενο λόγω του μεγέθους και της μορφής των κειμένων του Dataset που χρησιμοποιήθηκε. Εισάγεται, επομένως, θόρυβος λόγω των μορφολογικών χαρακτηριστικών και

του μεγέθους των κειμένων κριτικής ταινιών..

Συνεπώς, πραγματοποιείται η υπόθεση ότι η προτεινόμενη Μέθοδος είναι δυνατό να πετύχει μεγαλύτερη τιμή ακρίβειας στα πλαίσια Ανάλυσης Συναισθήματος μεγαλύτερων κειμένων, τα οποία έχουν μεγαλύτερη πιθανότητα να περιέχουν όλες τις καίριες συντακτικές εξαρτήσεις οι οποίες, ταυτόχρονα, να δομούνται από συναισθηματικά φορτισμένες λέξεις. Παραδείγματα τέτοιων κειμένων αποτελούν τα κείμενα προσωπικής τοποθέτησης όπως τα προσωπικά ιστολόγια (blogs), αλλά και τα άρθρα και τα δοκίμια πολιτικού σχολιασμού που αναρτώνται στο Διαδίκτυο.

Τέλος, είναι σημαντικό να σημειωθεί ότι η παρούσα Διπλωματική πραγματοποίησε για πρώτη φορά μία απόπειρα συγκερασμού δύο ανεξάρτητων φιλοσοφιών προσέγγισης της Ανάλυσης Συναισθήματος. Λαμβάνοντας ως κοινό σημείο εκκίνησης την κατασκευή γράφων για την αναπαράσταση της καθεμίας πολικότητας αλλά και του κάθε υπο εξέταση κειμένου, επιχειρήθηκε ο συνδυασμός ετερόκλητων χαρακτηριστικών κειμένων, ούτως ώστε να δομηθεί ένας αλγόριθμος που να δύναται να εξετάσει κείμενα διαφορετικής μορφής, ύφους και περιεχομένου. Δημιουργήθηκε, συνεπώς, ένας αλγόριθμος που αξιοποιεί την συναισθηματική πληροφορία που εμπεριέχεται σε κάθε κείμενο, είτε η ίδια βρίσκεται σε στατιστική είτε σε λεξιλογική ή/και συντακτική έκφανση.

6.2 Μελλοντικές Επεκτάσεις

Μελλοντικά, η προτεινόμενη Μέθοδος μπορεί να αξιοποιηθεί για την Ανάλυση διαφορετικού συνόλου κειμένων και, κατέπέκταση, να μετρηθεί η απόδοση και η ακρίβεια ταξινόμησής της σε διαφορετικά σύνολα εκπαίδευσης και ελέγχου. Ακόμη, είναι δυνατό να μελετηθεί η αξιοποίηση αποκλειστικά λεξιλογικής πληροφορίας ως προς τα χαρακτηριστικά εισόδου του αλγόριθμου ταξινόμησης, η οποία μπορεί να συνοδεύεται και από τις συντακτικές πληροφορίες του εξεταζόμενου κειμένου. Τέλος, μπορεί να επιχειρηθεί η υβριδοποίηση των Baseline Μεθόδων (Bag Of Words, N-gram Γράφων κ.ο.κ.) βάσει της Λεξιλογικής Προσέγγισης και η διερεύνηση της μεταβολής στην ακρίβεια ταξινόμησης.

Παράρτημα Α′

Κώδικας

Α′.1 SentimentDependencies.java

```
package sentimentanalysis;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;

import edu.stanford.nlp.ling.CoreLabel;
import edu.stanford.nlp.ling.SentenceUtils;
import edu.stanford.nlp.ling.TaggedWord;
import edu.stanford.nlp.parser.lexparser.LexicalizedParser;
import edu.stanford.nlp.trees.GrammaticalStructure;
import edu.stanford.nlp.trees.GrammaticalStructureFactory;
import edu.stanford.nlp.trees.PennTreebankLanguagePack;
import edu.stanford.nlp.trees.Tree;
import edu.stanford.nlp.trees.TreebankLanguagePack;
import edu.stanford.nlp.trees.TypedDependency;

public class SentimentDependencies {

 private static SentiWordNet sentiwordnet;
 private static final String pathToSWN="SentiWordNet.txt" ;
 private static Tree parse;
 private static final List<String> patterns = new ArrayList<>
```

```
    (Arrays.asList("acomp", "advmod", "amod","dobj", "neg", "
       nsubj","rcmod", "xcomp", "cop","ccomp"));

private static ArrayList<TypedDependency> tdList = new
    ArrayList<TypedDependency>();
private static ArrayList<TaggedWord> taggedWords;
private static LexicalizedParser lp;

private static double senticValue;

public SentimentDependencies(String[] sent) throws
    IOException{
        SentimentDependencies.sentiwordnet =
           intializeSentiWord(pathToSWN);
        initializeParser(sent);
        tagWords();
        }

public ArrayList<String> findDependencies(){

 TreebankLanguagePack tlp = new PennTreebankLanguagePack();
 GrammaticalStructureFactory gsf = tlp.
    grammaticalStructureFactory();
 GrammaticalStructure gs = gsf.newGrammaticalStructure(parse
    );
 tdList = (ArrayList<TypedDependency>) gs.
    typedDependenciesCCprocessed();

 Iterator<TypedDependency> dependency = tdList.iterator();
 ArrayList<String> syntaxRelations = new ArrayList<String>()
    ;

 senticValue = 0.0;

 while(dependency.hasNext()){
        String[] dep = dependency.next().toString().split("
           \\(");
        String[] words = dep[1].split(",");
        CharSequence dash = "-";

        if(patterns.contains(dep[0]) && (words.length == 2)){
```

```
        String first = words[0];
        String second = words[1].replaceAll("\\)","").trim()
           ;

        if(first.isEmpty() || second.isEmpty() || !first.
           contains(dash) || !second.contains(dash))
         continue;

        String firstWord = first.substring(0, first.
           lastIndexOf("-")).replaceAll("[^a-zA-Z]", "");
    String firstPlace = first.substring(first.lastIndexOf("-
       ") + 1).replaceAll("[^0-9]", "");

        String secondWord = second.substring(0, second.
           lastIndexOf("-")).replaceAll("[^a-zA-Z]", "");
        String secondPlace = second.substring(second.
           lastIndexOf("-") + 1).replaceAll("[^0-9]", "");

        if(!isNumeric(firstPlace) || !isAlpha(firstWord) ||
           !isNumeric(secondPlace) || !isAlpha(secondWord))
         continue;

        TaggedWord firstPosTag = taggedWords.get(Integer.
           valueOf(firstPlace) - 1);
        TaggedWord secondPosTag = taggedWords.get(Integer.
           valueOf(secondPlace) - 1);

        String firstTag = firstPosTag.tag().substring(0, 1).
           toLowerCase();
        String secondTag = secondPosTag.tag().substring(0,
           1).toLowerCase();

        if(firstTag.equals("j"))
               firstTag = "a";

        if(secondTag.equals("j"))
               secondTag = "a";

         double firstValue = calculateSentiValue(firstWord,
            firstTag);
```

```
            double secondValue = calculateSentiValue(secondWord
               , secondTag);
            senticValue = firstValue + secondValue;

            syntaxRelations.add(dep[0]);
            syntaxRelations.add(Double.toString(senticValue));
     }
    }
    return syntaxRelations;
  }

 private void tagWords() {
  taggedWords = new ArrayList<TaggedWord>();
  taggedWords = parse.taggedYield();
 }

 private static double calculateSentiValue(String word,
    String tag){
  return sentiwordnet.extract(word, tag);
 }

 private static SentiWordNet intializeSentiWord(String
    pathToSWN) throws IOException{
 SentiWordNet sentiwordnet = new SentiWordNet(pathToSWN);
 return sentiwordnet;
 }

 private static void initializeParser(String[] sent) {
  lp = LexicalizedParser
  loadModel("edu/stanford/nlp/models/lexparser/englishPCFG.
     ser.gz");
  lp.setOptionFlags(new String[] { "-outputFormat", "
     wordsAndTags", "-maxLength", "80","-
     retainTmpSubcategories" });
  List<CoreLabel> rawWords = SentenceUtils.toCoreLabelList(
     sent);
  parse = lp.apply(rawWords);
 }

 public double getsenticValue(){
  return senticValue;
```

```
 }

 private boolean isNumeric(String s) {
  return s.matches("[-+]?\\d*\\.?\\d+");
 }

 private boolean isAlpha(String s) {
  return s.matches("[a-zA-Z]+");
 }
}
```

A′.2 SyntacticRelations.java

```
package sentimentanalysis;

import java.util.ArrayList;

public class SyntacticRelations {

        private double acomp;
        private double advmod;
        private double amod;
        private double dobj;
        private double neg;
        private double nsubj;
        private double rcmod;
        private double xcomp;
        private double cop;
        private double ccomp;

        private String dependency;
        private double sentiValue;

        public SyntacticRelations(){
                this.acomp = 0.0;
                this.advmod = 0.0;
                this.amod = 0.0;
                this.dobj = 0.0;
                this.neg = 0.0;
                this.nsubj = 0.0;
                this.rcmod = 0.0;
```

```
            this.xcomp = 0.0;
            this.cop = 0.0;
            this.ccomp = 0.0;

        }
        public void agreggateSentiment(ArrayList<String>
           syntaxRelations){
            for(int index =0; index < syntaxRelations.
               size(); index++){
                if(index % 2 == 0){
                        dependency = syntaxRelations.
                           get(index);
            }
                else {
                        sentiValue = Double.valueOf(
                           syntaxRelations.get(index)
                           );
                        modifyDepValue(dependency ,
                           sentiValue);
                }
            }
        };

        private void modifyDepValue(String dependency, double
            sentiValue){
            switch(dependency){
            case("acomp"):
                    setAcomp(sentiValue);
                    break;
            case("advmod"):
                    setAdvmod(sentiValue);
                    break;
            case("amod"):
                    setAmod(sentiValue);
                    break;
            case("dobj"):
                    setDobj(sentiValue);
                    break;
            case("neg"):
                    setNeg(sentiValue);
                    break;
```

```
                case("nsubj"):
                        setNsubj(sentiValue);
                        break;
                case("rcmod"):
                        setRcmod(sentiValue);
                        break;
                case("xcomp"):
                        setXcomp(sentiValue);
                        break;
                case("cop"):
                        setCop(sentiValue);
                        break;
                case("ccomp"):
                        setCcomp(sentiValue);
                        break;
                default:
                        break;
                }

        }

        public void setAcomp(double acomp){
                this.acomp += acomp;
        }

        public void setAdvmod(double advmod){
                this.advmod += advmod;
        }

        public void setAmod(double amod){
                this.amod += amod;
        }

        public void setDobj(double dobj){
                this.dobj += dobj;
        }

        public void setNeg(double neg){
                this.neg += neg;
        }
```

```
public void setNsubj(double nsubj){
        this.nsubj += nsubj;
}

public void setRcmod(double rcmod){
        this.rcmod += rcmod;
}

public void setXcomp(double xcomp){
        this.xcomp += xcomp;
}

public void setCop(double cop){
        this.cop += cop;
}

public void setCcomp(double ccomp){
        this.ccomp += ccomp;
}

public double getAcomp(){
        return acomp;
}

public double getAdvmod(){
        return advmod;
}

public double getAmod(){
        return amod;
}

public double getDobj(){
        return dobj;
}

public double getNeg(){
        return neg;
}

public double getNsubj(){
```

```
                return nsubj;
        }

        public double getRcmod(){
                return rcmod;
        }

        public double getXcomp(){
                return xcomp;
        }

        public double getCop(){
                return cop;
        }

        public double getCcomp(){
                return ccomp;
        }


}
```

A′.3 AttributeRelationFile.java

```
package sentimentanalysis;

import weka.core.Attribute;
import weka.core.DenseInstance;
import weka.core.Instances;
import gr.demokritos.iit.jinsect.structs.GraphSimilarity;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;


public class AttributeRelationFile {
```

```
private String relationName;
private ArrayList<Attribute> attributes;
private Instances instances;

public AttributeRelationFile(String relationName) {
 this.relationName = relationName;
}

public void createFile(WordGraphsSimilarities values,
String posFilepath,
String negFilepath, ArrayList<String> posReviewFilenames,
ArrayList<String> negReviewFilenames) throws  IOException,
   CloneNotSupportedException {
 createAttributesPlus();
 addHeader();
 addData(values, posFilepath, negFilepath,
    posReviewFilenames,negReviewFilenames);
}

public void createFile(NGramGraphsSimilarities values,
   String posFilepath,
String negFilepath, ArrayList<String> posReviewFilenames,
ArrayList<String> negReviewFilenames) throws IOException {
 createAttributes();
 addHeader();
 addData(values, posFilepath, negFilepath,
    posReviewFilenames,negReviewFilenames);
}

public void createFile(String posFilepath, String
   negFilepath,
 ArrayList<String> posReviewFilenames,
 ArrayList<String> negReviewFilenames) throws IOException {

 createTextAttribute();
 addHeader();
 addData(posFilepath, negFilepath, posReviewFilenames,
                                negReviewFilenames);
}
```

```
public void createFile(String file) throws IOException {
 createRelativeAttributes();
 addHeader();
 addData(file);
}

public void createFile(String file, int levels) throws
   IOException {
 createAttributes();
 addHeader();
 addData(file, levels);
}

private void createAttributes() {
 attributes = new ArrayList<Attribute>();
 attributes.add(new Attribute("PositiveContainmentSimilarity
    "));
 attributes.add(new Attribute("
    PositiveNormalizedValueSimilarity"));
 attributes.add(new Attribute("PositiveValueSimilarity"));

 attributes.add(new Attribute("NegativeContainmentSimilarity
    "));
 attributes.add(new Attribute("
    NegativeNormalizedValueSimilarity"));
 attributes.add(new Attribute("NegativeValueSimilarity"));

 ArrayList<String> sentimentValues = new ArrayList<String>()
    ;
 sentimentValues.add("0");
 sentimentValues.add("1");
 attributes.add(new Attribute("Sentiment", sentimentValues))
    ;
}

private void createAttributesPlus() {
 attributes = new ArrayList<Attribute>();

 attributes.add(new Attribute("PositiveContainmentSimilarity
    "));
```

```
 attributes.add(new Attribute("
    PositiveNormalizedValueSimilarity"));
 attributes.add(new Attribute("PositiveValueSimilarity"));

 attributes.add(new Attribute("NegativeContainmentSimilarity
    "));
 attributes.add(new Attribute("
    NegativeNormalizedValueSimilarity"));
 attributes.add(new Attribute("NegativeValueSimilarity"));

 attributes.add(new Attribute("AcompDependency"));
 attributes.add(new Attribute("AdvmodDependency"));
 attributes.add(new Attribute("AmodDependency"));
 attributes.add(new Attribute("DobjDependency"));
 attributes.add(new Attribute("NegDependency"));
 attributes.add(new Attribute("NsubjDependency"));
 attributes.add(new Attribute("RcmodDependency"));
 attributes.add(new Attribute("XcompDependency"));
 attributes.add(new Attribute("CopDependency"));
 attributes.add(new Attribute("CcompDependency"));

 ArrayList<String> sentimentValues = new ArrayList<String>()
    ;
 sentimentValues.add("0");
 sentimentValues.add("1");
 attributes.add(new Attribute("Sentiment", sentimentValues))
    ;
}

private void createTextAttribute() {
 attributes = new ArrayList<Attribute>();
 attributes.add(new Attribute("TextReview", (ArrayList<
    String>) null));
 ArrayList<String> sentimentValues = new ArrayList<String>()
    ;
 sentimentValues.add("0");
 sentimentValues.add("1");
 attributes.add(new Attribute("ReviewSentiment",
    sentimentValues));
}
```

```
private void createRelativeAttributes() {
 attributes = new ArrayList<Attribute>();
 attributes.add(new Attribute("RelativeContainmentSimilarity
    "));
 attributes.add(new Attribute("
    RelativeNormalizedValueSimilarity"));
 attributes.add(new Attribute("RelativeValueSimilarity"));
 ArrayList<String> sentimentValues = new ArrayList<String>()
    ;
 sentimentValues.add("0");
 sentimentValues.add("1");
 attributes.add(new Attribute("Sentiment", sentimentValues))
    ;
}

private void addHeader() {
 instances = new Instances(relationName, attributes, 0);
}

private void addData(WordGraphsSimilarities values, String
   posFilepath,
String negFilepath, ArrayList<String>  posReviewFilenames,
   ArrayList<String> negReviewFilenames) throws  IOException
   ,CloneNotSupportedException {

 addInstances(values.getPosModelGraph(), posFilepath,
    posReviewFilenames,   values, 1);
 addInstances(values.getNegModelGraph(), negFilepath,
    negReviewFilenames,   values, 0);
}

private void addData(NGramGraphsSimilarities values, String
   posFilepath,   String negFilepath, ArrayList<String>
   posReviewFilenames,ArrayList<String> negReviewFilenames)
   throws  IOException {

addInstances(values.getPosModelGraph(), posFilepath,
   posReviewFilenames,  values, 1);
addInstances(values.getNegModelGraph(), negFilepath,
   negReviewFilenames,  values, 0);
}
```

```
private void addData(String posFilepath, String negFilepath,
ArrayList<String> posReviewFilenames, ArrayList<String>
   negReviewFilenames) throws IOException {

 addInstances(posFilepath, posReviewFilenames, 1);
 addInstances(negFilepath, negReviewFilenames, 0);
}

private void addData(String file) throws IOException {

 BufferedReader reader = new BufferedReader(new FileReader(
    file));
 for (int line = 0; line < 11; line++)
  reader.readLine();

  String line = reader.readLine();
  while(line != null) {
   String[] values = line.split(",");
   double[] simValues = new double[values.length];
  for (int index = 0; index < values.length - 1; index++)
   simValues[index] = Double.parseDouble(values[index]);
   simValues[values.length - 1] = Integer.parseInt(values[
      values.length -    1]);

  addInstance(simValues);
  line = reader.readLine();
  }
reader.close();
}

private void addData(String file, int levels) throws
   IOException {
 BufferedReader reader = new BufferedReader(new FileReader(
    file));
 for (int line = 0; line < 11; line++)
  reader.readLine();

  double[] max = {0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

  String line = reader.readLine();
```

```
  while(line != null) {
   String[] values = line.split(",");
   double[] simValues = new double[6];
   for (int index = 0; index < values.length - 1; index++)
    simValues[index] = Double.parseDouble(values[index]);

   for (int i = 0; i < simValues.length; i++)
    if (simValues[i] > max[i])
     max[i] = simValues[i];
   line = reader.readLine();
   }

 reader.close();
 reader = new BufferedReader(new FileReader(file));
 for (int iline = 0; iline < 11; iline++)
  reader.readLine();

 line = reader.readLine();
 while(line != null) {
  String[] values = line.split(",");
  double[] simValues = new double[values.length];
  for (int index = 0; index < values.length - 1; index++)
   simValues[index] = Double.parseDouble(values[index]);
  simValues[values.length - 1] = Integer.parseInt(values[
     values.length -   1]);

 addInstance(simValues, max, levels);
 line = reader.readLine();
 }
reader.close();
}

private void addInstances(ModelWordGraph graph, String
   reviewsFilepath,
                      ArrayList<String> reviewFilenames,
                         WordGraphsSimilarities values,
                      int sentiment) throws IOException {

ReviewWordGraph reviewGraph = new ReviewWordGraph(graph.
   getWindow(),
```

```
                                graph.getReviewsGraph().
                                   isPreprocess());

for (String s: reviewFilenames) {
 reviewGraph.createGraph(reviewsFilepath.concat(s));

 DocumentWordGraph posSubgraph = reviewGraph.commonSubgraph(
    values.getPosModelGraph().getReviewsGraph().getGraph());
 DocumentWordGraph negSubgraph = reviewGraph.commonSubgraph(
    values.getNegModelGraph().getReviewsGraph().getGraph());

 if(posSubgraph.isEmpty() && negSubgraph.isEmpty())
                                continue;

 ArrayList<String> posNodes = new ArrayList<String>();
 ArrayList<String> negNodes = new ArrayList<String>();

 boolean posFlag =false;
 boolean negFlag = false;

 if(!posSubgraph.isEmpty()){
  posNodes = posSubgraph.getNodestoString();
  for(int index = 0; index < posNodes.size();index++){
                                    posNodes.set(index,
                                       posNodes.get(index
                                       ).replaceAll("[^a-
                                       zA-Z]", ""));
       }
}
else
 posFlag = true;

if(!negSubgraph.isEmpty()) {
 negNodes = negSubgraph.getNodestoString();
 for(int index = 0; index < negNodes.size();index++){
  negNodes.set(index, negNodes.get(index).replaceAll("[^a-zA
     -Z]", ""));
 }
}
else
 negFlag = true;
```

```
 SyntacticRelations posSyntaxValues = new SyntacticRelations
    ();
 SyntacticRelations negSyntaxValues = new SyntacticRelations
    ();
 Scanner reviewFile = new Scanner(new BufferedReader (new
    FileReader(reviewsFilepath.concat(s))));
 reviewFile.useDelimiter("[.:!?]");

 while(reviewFile.hasNext()){
  String[] sentence = reviewFile.next().split("␣");

   for(String chunk : sentence){
    String word= chunk.replaceAll("^\\p{Punct}|\\p{Punct}$
       |[,.;!?(){}\\[\\]%]|␣<br␣\\>", "");
    if(!word.isEmpty() && (!posFlag ||!negFlag)){
     if(posNodes.contains(word) && !negNodes.contains(word)
        && !posFlag){
          SentimentDependencies sentiValues = new
             SentimentDependencies(sentence);
          ArrayList<String> posSyntaxRelations = sentiValues.
             findDependencies();

                      posSyntaxValues.agreggateSentiment(
             posSyntaxRelations);
      posFlag = true;
    }
    else if(negNodes.contains(word) && !posNodes.contains(
       word) && !negFlag){
     SentimentDependencies sentiValues = new
        SentimentDependencies(sentence);
         ArrayList<String> negSyntaxRelations = sentiValues.
            findDependencies();
         negSyntaxValues.agreggateSentiment(
            negSyntaxRelations);
         negFlag = true;
   }
  }
 }
}
 reviewFile.close();
```

```
 values.graphsSimilaritiesWith(reviewGraph);
 addInstance(values.getPosGraphSimilarities(),values.
    getNegGraphSimilarities(), posSyntaxValues,
    negSyntaxValues, sentiment);
 }
}

 private void addInstances(ModelNGramGraph graph, String
    reviewsFilepath, ArrayList<String> reviewFilenames,
    NGramGraphsSimilarities values,int sentiment) throws
    IOException {

  ReviewNGramGraph reviewGraph = new ReviewNGramGraph(graph.
     getNSize(),
                                graph.getReviewsGraph().
                                   isPreprocess());
  for (String s: reviewFilenames) {
   reviewGraph.createGraph(reviewsFilepath.concat(s));
   values.graphsSimilaritiesWith(reviewGraph);
   addInstance(values.getPosGraphSimilarities(),
   values.getNegGraphSimilarities(), sentiment);
 }

 private void addInstances(String reviewsFilepath, ArrayList<
    String> reviewFilenames, int sentiment) throws
    IOException {

  String reviewFile = null;
  BufferedReader reader = null;

  for (String s: reviewFilenames) {
   reviewFile = reviewsFilepath.concat(s);
   reader = new BufferedReader(new FileReader(reviewFile));
   addInstance(reader.readLine(), sentiment);
  }
  reader.close();
 }

 private void addInstance(GraphSimilarity posGraphSim,
    GraphSimilarity negGraphSim, int sentiment) {
```

```
 double[] instance = new double[instances.numAttributes()];

 instance[0] = posGraphSim.ContainmentSimilarity;
 instance[1] = posGraphSim.ValueSimilarity/posGraphSim.
    SizeSimilarity;
 instance[2] = posGraphSim.ValueSimilarity;

 instance[3] = negGraphSim.ContainmentSimilarity;
 instance[4] = negGraphSim.ValueSimilarity/negGraphSim.
    SizeSimilarity;
 instance[5] = negGraphSim.ValueSimilarity;

 instance[6] = sentiment;

 instances.add(new DenseInstance(1.0, instance));
}

private void addInstance(GraphSimilarity posGraphSim,
    GraphSimilarity negGraphSim, SyntacticRelations
    posSentiValues,SyntacticRelations negSentiValues, int
    sentiment) {

 double[] instance = new double[instances.numAttributes()];

 instance[0] = posGraphSim.ContainmentSimilarity;
 instance[1] = posGraphSim.ValueSimilarity/posGraphSim.
    SizeSimilarity;
 instance[2] = posGraphSim.ValueSimilarity;

 instance[3] = negGraphSim.ContainmentSimilarity;
 instance[4] = negGraphSim.ValueSimilarity/negGraphSim.
    SizeSimilarity;
 instance[5] = negGraphSim.ValueSimilarity;

 instance[6] = posSentiValues.getAcomp()+ negSentiValues.
    getAcomp();
 instance[7] = posSentiValues.getAdvmod() + negSentiValues.
    getAdvmod();
 instance[8] = posSentiValues.getAmod() +negSentiValues.
    getAmod() ;
```

```
 instance[9] = posSentiValues.getCcomp() + negSentiValues.
    getCcomp();
 instance[10] = posSentiValues.getCop() + negSentiValues.
    getCop();
 instance[11] = posSentiValues.getDobj() + negSentiValues.
    getDobj();
 instance[12] = posSentiValues.getNeg() + negSentiValues.
    getNeg();
 instance[13] = posSentiValues.getNsubj() + negSentiValues.
    getNsubj();
 instance[14] = posSentiValues.getRcmod() + negSentiValues.
    getRcmod();
 instance[15] = posSentiValues.getXcomp() + negSentiValues.
    getXcomp();
 instance[16] = sentiment;

 instances.add(new DenseInstance(1.0, instance));
}

private void addInstance(String text, int sentiment) {

 double[] instance = new double[instances.numAttributes()];
 instance[0] = instances.attribute(0).addStringValue(text);
 instance[1] = sentiment;
 instances.add(new DenseInstance(1.0, instance));
}

private void addInstance(double[] simValues) {
 double[] instance = new double[instances.numAttributes()];

 instance[0] = dsim(simValues[0], simValues[3]);
 instance[1] = dsim(simValues[1], simValues[4]);
 instance[2] = dsim(simValues[2], simValues[5]);

 instance[3] = simValues[6];

 instances.add(new DenseInstance(1.0, instance));
}

private void addInstance(double[] simValues, double[] max,
   int levels) {
```

```
  double[] instance = new double[instances.numAttributes()];

  for (int i = 0; i < simValues.length - 1; i++)
   instance[i] = Math.ceil((simValues[i] / max[i]) * levels);

  instance[instances.numAttributes() - 1] = simValues[
     simValues.length - 1];

  instances.add(new DenseInstance(1.0, instance));
 }

 private int dsim(double posSim, double negSim) {
  int equal = 0;
  int positive = 1;
  int negative = 2;

  if (posSim < negSim) return negative;
  else if (posSim > negSim) return positive;
  else return equal;
 }

 public void storeToFile(String outputFile) throws
    IOException {
  BufferedWriter writer = new BufferedWriter(new FileWriter(
     outputFile));
  writer.write(instances.toString());
  writer.flush();
  writer.close();
 }

 public Instances getInstances() {
  return instances;
 }
}
```

A′.4 SentiWordNet.java

```
package sentimentanalysis;

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

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class SentiWordNet {

 private Map<String, Double> dictionary;

 public SentiWordNet(String pathToSWN){

 this.dictionary = new HashMap<String, Double>();

 HashMap<String, HashMap<Integer, Double>> tempDictionary =
    new HashMap<String, HashMap<Integer, Double>>();

 BufferedReader csv = null;
 try {
  csv = new BufferedReader(new FileReader(pathToSWN));
  int lineNumber = 0;

  String line;
  while ((line = csv.readLine()) != null) {
   lineNumber++;

  if (!line.trim().startsWith("#")) {
   String[] data = line.split("\t");
   String wordTypeMarker = data[0];

   if (data.length != 6) {
        throw new IllegalArgumentException("Incorrect␣
           tabulation␣format␣in␣file,␣line:␣"+ lineNumber);
   }

        Double synsetScore = Double.parseDouble(data[2])-
           Double.parseDouble(data[3]);

        String[] synTermsSplit = data[4].split("␣");

        for (String synTermSplit : synTermsSplit) {
         String[] synTermAndRank = synTermSplit.split("#");
```

```
        String synTerm = synTermAndRank[0] + "#"+
           wordTypeMarker;

        int synTermRank = Integer.parseInt(synTermAndRank
           [1]);
        if (!tempDictionary.containsKey(synTerm)) {
     tempDictionary.put(synTerm,new HashMap<Integer, Double
        >());
        }

        tempDictionary.get(synTerm).put(synTermRank,
           synsetScore);
   }
 }
}

for (Map.Entry<String, HashMap<Integer, Double>> entry :
   tempDictionary.entrySet()) {
       String word = entry.getKey();
       Map<Integer, Double> synSetScoreMap = entry.getValue
          ();

       double score = 0.0;
       double sum = 0.0;
       for (Map.Entry<Integer, Double> setScore :
          synSetScoreMap.entrySet())       {
          score += setScore.getValue() / (double) setScore.
             getKey();
          sum += 1.0 / (double) setScore.getKey();
         }
        score /= sum;

        this.dictionary.put(word, score);
 }
 } catch (Exception e) {
                       e.printStackTrace();}
   finally {
    if (csv != null) {
      try {
           csv.close();
      } catch (IOException e)
```

```
            e.printStackTrace();
        }
      }
   }
  }

  public double extract(String word, String pos) {

   if(this.dictionary.get(word + "#" + pos) != null) {
    return this.dictionary.get(word + "#" + pos);
   }
   else {
    return 0.0;
   }
  }
}
```

Παράρτημα Β΄

Part-of-speech tags

Στην επόμενη σελίδα ακολουθεί πίνακας ευρετήριο των POS (Part Of Speech) χαρακτηρισμών. Οι χαρακτηρισμοί Μερών του Λόγου ανατίθενται σε μία λέξη ανάλογα με τη θέση της και το ρόλο της σε μία πρόταση. Παραδοσιακά, η γραμματική ανάλυση αναθέτει σε καθεμία από τις λέξεις έναν από τους εξής οχτώ χαρακτηρισμούς:

1. Ρήμα (VB)
2. Ουσιαστικό (NN)
3. Επίθετο (JJ)
4. Αντωνυμία (PR+DT)
5. Επίρρημα (RB)
6. Πρόθεση (IN)
7. Σύνδεσμος (CC)
8. Επιφώνημα (UH)

Οι παρακάτω χαρακτηρισμοί αποτελούν το Treebank Tag-set του Πανεπιστημίου της Πενσυλβάνια.

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Πίνακας Β΄.1: POS Tags

Βιβλιογραφία

[1] *Statistical Identification of Language*. New Mexico State University, 1994.

[2] *Graph-Based N-gram Language Identification on Short Text*. Department of Computer Science, Eindhoven University of Technology, 2017.

[3] Basant Agarwal, Namita Mittal και Vijay Kumar. Feature extraction methods for semantic orientation based approaches to sentiment analysis. 2013.

[4] F. Aisopos, G. Papadakis και T. Varvarigou. Sentiment analysis of social media content using n-gram graphs. 2011.

[5] Andrei Z. Broder, Steven C Glassman και Geoffrey Manasse, Mark S.and Zweig. Syntactic clustering of the web. 1997.

[6] E. Cambria, B. Schuller, Y. Xia και C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2), 2013.

[7] A. D'Andrea, F. Ferri, P. Grifoni και T. Guzzo. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125, 2015.

[8] N. Friedman, D. Geiger και M. Goldszmidt. Bayesian network classifiers*. 1997.

[9] G. Giannakopoulos, V. Karkaletsis, G. Vouros και P. Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. 2008.

[10] Π. Κιούρτη. Ανάλυση Συναισθήματος με χρήση υβριδικών n-grams. NTUA, 2013.

[11] Vasileios Hatzivassiloglou και Kathleen R. McKeown. Predicting the semantic orientation of adjectives. Department of Computer Science, Columbia University, 1997.

[12] Youngjoong Ko. A study of term weighting schemes using class information for text classification. 2012.

[13] Julia Kreutzer και Neele Witte. Opinion mining using sentiwordnet. χ.χ.

[14] Andrew McCallum και Kamal Nigam. A comparison of event models for naive bayes text classification. 1998.

[15] W. Medhat, A. Hassan και H. Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5, 2014.

[16] T Finin T Oates A Joshi P Kolari, A Java. Detecting spam blogs: A machine learning approach. 2006.

[17] I. Rish. An empirical study of the naive bayes classifier. χ.χ.

[18] Stuart J. Russell και Peter Norvig. *Artificial Intelligence: A Modern Approach. Pearson Education.* 2η έκδοση, 2003.

[19] Henrique Siqueira και Flavia Barros. A feature extraction process for sentiment analysis of opinions on services. Centro de Informatica Universidade Federal de Pernambuco (UFPE), 2013.

[20] Andrija Tomović, Predrag Janičić και Vlado Kešelj. N-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer Methods and Programs in Biomedicine*, 2006.

[21] Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. 2002.

[22] Janyce Wiebe, Theresa Wilson και Claire Cardie. Annotating expressions of opinions and emotions in language. 2005.

Απόδοση ξενόγλωσσων όρων

Απόδοση	Ξενόγλωσσος όρος
χαρακτηριστικά	features
Επεξεργασία Φυσικής Γλώσσας	Natural Language Processing
Αναγνώριση Ομιλίας	Speech Recognition
ταξινομητής	classifier
Στιγμιότυπο Εκπαίδευσης	Training Instance
Διάνυσμα Χαρακτηριστικών	Feauture Vector
Συχνότητα Όρων	Term Frequency
Διακριτικό Κλάσσης	Class Label
Σάκος Λέξεων	Bag Of Words
Χαρακτηριστικά Μερών του Λόγου	Part Of Speech Tags
Μορφοσημαντικές Συνδέσεις	Morphosemantic Links
Ανάλυση Συναισθήματος	Sentiment Analysis
Μέσα Κοινωνικής Δικτύωσης	Social Media
Αναδρομικός Διαμερισμός	Recursive Partitioning
Νευρωνικά Δίκτυα	Neural Networks
Μπεϋζιανά Δίκτυα	Bayesian Networks
Λεξικά Συναισθήματος	Sentiment Lexicons
Μέγιστη Εντροπία	Maximum Entropy
σύνολο εκπαίδευσης	training set
σύνολο ελέγχου	testing set
Ευφυή Συστήματα Μεταφοράς	Intelligent Transportation Systems
Εταιρική Φήμη	Brand Reputation
Μέθοδοι Ταξινόμησης	Classification Methods
Αλγόριθμοι Βαθιάς Μάθησης	Deep Learning Algorithms
Συνάρτηση Παλινδρόμησης	Regression Function
Πιθανοτικοί Ταξινομητές	Probabilistic Classifiers
Γραμμικοί Ταξινομητές	Linear Classifiers
Διάνυσμα Υποστήριξης Μηχανής	Support Vector Machines
Δέντρο Αποφάσεων	Decision Tree
Συναισθηματικός Προσανατολισμός	Semantic Orientation
Λανθάνουσα Σημασιολογική Ανάλυση	Latent Semantic Analysis

Εντοπισμός κακόβουλων μηνυμάτων spam detection