



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΑ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

**Ταξινόμηση Κειμένων Ανοικτού Συνόλου με Χρήση της
Ομοιότητας Βάσει Κέντρων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Ηρώς Τριπιντή-Ιγνατίου

Επιβλέπων : Σταματάτος Ευστάθιος

Μέλη εξεταστικής επιτροπής: Καβαλιεράτου Εργίνα, Κωστούλας Θεόδωρος

Σάμος, Σεπτέμβριος 2023

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Ευχαριστώ την οικογένεια μου, τον αδερφό μου Γιάννη και τον Αντώνη που με βοήθησαν να φτάσω έως εδώ.

© 2023

της

Ηρώς Τριπιντή-Ιγνατίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή	13
1.1	Ταξινόμηση Κειμένων	13
1.2	Αντικείμενο διπλωματικής	13
1.3	Δομή της διπλωματικής	14
2	Θεωρητικό Υπόβαθρο	15
2.1	Γενικά	15
2.2	Εξόρυξη Πληροφορίας από Κείμενο	17
2.3	NLP & Named Entity Recognition	18
2.4	Διανυσματικές Αναπαραστάσεις και Στάθμιση χαρακτηριστικών	19
2.5	Ταξινόμηση με Κανόνες	20
2.5.1	Ταξινόμηση με Βάση τα Στατιστικά	21
2.5.2	Ταξινόμηση με Βάση τα Χαρακτηριστικά	23
2.6	Εκτίμηση Ανάκτησης Πληροφορίας	25
3	Text Classification Algorithms	27
3.1	Δέντρα Αποφάσεων	27
3.2	Rule-Based Classifiers	27
3.4	Neural Network Classifiers	31
3.5	Bayes Classifiers	32
4	Ανάλυση Αλγορίθμων Open-Set	33
4.1	Περιγραφή Αλγορίθμου cbsSVM	34
4.1.1	Open Space Risk	35
4.1.2	Διανυσματικές Αναπαραστάσεις	37
4.1.3	Σώμα Κειμένων Εκπαίδευσης	37
4.1.4	Μετατροπή απλού σε πιθανοτικό classifier	38
4.2	Λογική Αλγορίθμου	38
4.3	Εύρεση Κέντρων	41
4.4	Μετρικές Ομοιότητας	43
4.5	Ψευδοκώδικας	44
5	Περιβάλλον Πειραμάτων	46
5.1	Corpus	46
5.1.1	KI-04	48
5.1.2	Santini	49
5.2	Μετρικές	50
5.2.1	Μετρικές Αξιολόγησης Μοντέλου	50
5.2.2	Παραλλαγές Μετρικών για Αξιολόγηση Μοντέλων Ανοικτού Συνόλου	51
5.2.3	Openness-F1-Score Curve	53
5.2.4	Precision-Recall Curve	55
5.3	Υλοποίηση Πειραμάτων	55
5.3.1	Επαναλήψεις, Train-Test-Fold e.t.c	55

5.3.2 Openness-F1-Score Curve	56
5.3.3 Precision-Recall Curve	58
6 Συμπεράσματα	60
7 Βιβλιογραφία	61

Λίστα Σχημάτων

1. Σχήμα 3.1 : Αναπαράσταση δεδομένων από τον μονοδιάστατο στον δισδιάστατο χώρο.
(Manning, et al., 2008)
2. Σχήμα 3.2 : Ένα τυπικό σύστημα Νευρωνικού Δικτύου που χρησιμοποιείται για
Ταξινόμηση (Aggarwal, et al., 2012).
3. Σχήμα 4.1 : Αναπαράσταση διανύσματος κειμένου στο Document Space.
4. Σχήμα 4.2 : Αναπαράσταση διανύσματος κειμένου στο Center Based Similarity Space.
5. Σχήμα 5.1: Αποτελέσματα του *cbsSVM* Classifier για το KI-04 σε διαφορετικά επίπεδα
openness
6. Σχήμα 5.2 : Απόδοση του OCSVM σε διαφορετικά επίπεδα openness (Pritsos, et al., 2019)
7. Σχήμα 5.3 : Precision-Recall Curve του *cbsSVM* για το SANTINIS corpus με W1G, W3G,
C4G and POS3G features
8. Σχήμα 5.4: Precision-Recall Curves of OCSVM models on SANTINIS corpus using W1G,
W3G and C4G features (Pritsos, et al., 2019)

Λίστα Πινάκων

1. Πίνακας 5.1: Περιγραφή Corpora KI-04 και πλήθος σελίδων ανα κλάση (genre) (Pritsos, et al., 2019).
2. Πίνακας 5.2: Περιγραφή Corpora SANTINI και πλήθος σελίδων ανα κλάση (genre) (Pritsos, et al., 2019).

Ακρωνύμια

BoW	Bag of Words
CBS	Center-Based Similarity
cbsSVM	Center-Based Similarity based on SVM
CRFs	Conditional Random Fields
CV	Cross Validation
DCG	Discounted Cumulative Gain
DS	Document Space
HMMs	Hidden Markov Models
LDA	Latent Dirichlet Allocation
MEMMs	Maximum Entropy Markov Models
MAP	Mean Average Precision
NER	Named Entity Recognition
NLP	Natural Language Processing
NDCG	Normalized Discounted Cumulative Gain
OOV	Out Of Vocabulary
POS	Part Of Speech
PRC	Precision-Recall Curve

PLSI	Probabilistic Latent Semantic Indexing
Regex	Regular Expressions
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency

Περίληψη

Η ραγδαία αύξηση του όγκου των δεδομένων που ανεβαίνουν στο διαδίκτυο καθιστά πιο δύσκολη τη διαχείριση των πληροφοριών που φέρνει μαζί του, με νέες τεχνολογίες που εισάγονται στον τομέα των Big Data για τον σκοπό αυτό. Με την πλειονότητα των δεδομένων στο διαδίκτυο να έχουν τη μορφή μη δομημένου κειμένου, ένας νέος κλάδος έχει αναδυθεί από την εξόρυξη δεδομένων. Το Text Mining θεωρείται ως ένα πολύ σημαντικό εξειδικευμένο εργαλείο στον τρόπο ανάλυσης δεδομένων σε μορφή κειμένου. Τα δεδομένα σε κείμενο δεν είναι αυστηρά καθορισμένοι και εύκολα διαχειρίσιμοι πόροι, ενώ οι ασάφειες και η διαρκώς μεταβαλλόμενη φύση του περιεχομένου οδηγούν σε δυσκολία στην κατανόηση και την εξόρυξη δεδομένων. Σημαντικό στάδιο και θεμέλιος λίθος για στην εξόρυξη κειμένου αποτελεί η Ταξινόμηση Κειμένου/Text Classification. Μέσω της μηχανικής μάθησης, η Ταξινόμηση Κειμένων καλείται να δώσει λύση στο ζήτημα της διαχείρισης και κατηγοριοποίησης κειμένων σε ηλεκτρονική μορφή. Η ταξινόμηση κειμένων γίνεται συνήθως μέσω μιας ποικιλίας αλγορίθμων και διαδικασιών που αποτελούν διαφορετικά μοντέλα. Όλα αυτά τα μοντέλα απαιτούν μια διαδικασία «εκπαίδευσης» και «δοκιμής» όπου τους δίνεται ένα σύνολο κειμένων και πρέπει να τα κατηγοριοποιήσουν χωρίς λάθη. Σε αυτό το σημείο, γίνεται η σύγκριση απόδοσης των αλγορίθμων Ταξινόμησης Κειμένων, όπου γνωστά και καθιερωμένα μοντέλα Ταξινόμησης Κειμένων δοκιμάζονται στο ίδιο σύνολο κειμένων και αξιολογείται η απόδοσή τους. Σε αυτή την εργασία γίνεται υλοποίηση του αλγορίθμου (Fei, et al., 2016) και στην συνέχεια γίνονται πειράματα κάνοντας χρήση των datasets που αναφέρονται στο (Pritsos. et al., 2019) με σκοπό να συγκριθούν τα αποτελέσματα.

Λέξεις Κλειδιά: *Αλγόριθμοι Ταξινόμησης Κειμένου, Εξόρυξη Πληροφορίας, Μηχανική Μάθηση*

Abstract

The rapid increase in the amount of data uploaded to the internet makes it more difficult to manage the information it brings with it, with new technologies being introduced in the field of Big Data for this purpose. With the majority of data on the internet being in the form of unstructured text, a new industry has emerged from data mining. Text Mining is considered as a very important specialized tool on the way to analyze data in text format. Data in text are not strictly defined and easily manageable resources, while ambiguities and the ever-changing nature of content lead to difficulty in understanding and mining data. Text Classification is an important stage and cornerstone for text mining. Through machine learning, text classification is called upon to provide a solution to the issue of managing and categorizing texts in electronic form. Text Classification is usually done through a variety of algorithms and processes that constitute different models. All these models require a "training" and a "testing" process where they are given a set of texts and they have to categorize them without errors. At this point, the performance comparison of Text Classification algorithms is made, where known and established Text Classification models are tested on the same set of texts and their performance is evaluated. In this paper, we develop the algorithm of (Fei, et al., 2016), then we present the experimental results carried out using the datasets mentioned in (Pritsos. et al., 2019) in order to compare the results.

Keywords: *Text Classification Algorithms, Information Mining, Machine Learning*

1 ***Εισαγωγή***

1.1 Ταξινόμηση Κειμένων

Η παρούσα διπλωματική εργασία, εξετάζει αρχικά τα θεμελιώδη χαρακτηριστικά των αλγορίθμων ταξινόμησης κειμένων καθώς αυτοί αποτελούν βασικό εργαλείο στην εξαγωγή πληροφορίας και την εξόρυξη γνώσης. Μέσω των αλγορίθμων ταξινόμησης κειμένων καθίσταται εφικτή η αυτόματη κατηγοριοποίηση των ηλεκτρονικών κειμένων γεγονός που οδηγεί στην καλύτερη οργάνωση της αδόμητης πληροφορίας την οποία περιέχουν. Για την ορθή ταξινόμηση κειμένων υπάρχουν διάφορα μοντέλα αλγορίθμων τα οποία έχουν πλεονεκτήματα αλλά εμφανίζουν και περιορισμούς και προκλήσεις στο τρόπο με τον οποίο διαχειρίζονται τα δεδομένα τα οποία καθορίζουν τις κατηγορίες-κλάσεις βάσει των οποίων γίνεται τελικώς και η κατηγοριοποίηση τους. Η πληθώρα διαφορετικών προσεγγίσεων στο ζήτημα αυτό στοχεύει στην αντιμετώπιση των διαφορετικών προβλημάτων που δυσκολεύουν την ορθή κατηγοριοποίηση των κειμένων. Για αυτό το λόγο απαιτείται οι αλγόριθμοι ταξινόμησης να παραμένουν ικανοποιητικοί στο κομμάτι της απόδοσής τους μέσα από συγκεκριμένες μετρικές.

1.2 Αντικείμενο διπλωματικής

Σε αυτή τη εργασία, ο στόχος είναι η υλοποίηση και η εκτίμηση της αποδοτικότητας του αλγορίθμου των Geli Fei και Bing Liu (Fei, et al., 2016) και στην συνέχεια να συγκριθεί με

αντίστοιχους αλγορίθμους Text-Classification. Αρχικά, πρέπει να αναπτυχθεί σωστά ο αλγόριθμος που περιγράφεται στο (Fei, et al., 2016) σε γλώσσα Python μέσω του λογισμικού Visual Studio Code - Jupiter και του Colab. Έπειτα, να γίνει εκπαίδευση (training) με ένα σύνολο κειμένων και στην συνέχεια δοκιμή (testing). Κατόπιν να εκτιμηθεί ποιος είναι ο πιο αποδοτικός τρόπος για να γίνει το καταλληλότερο ποσοστό μεταξύ training και testing. Ενώ θα πρέπει να γίνεται και μια εκτενής αναφορά στο θεωρητικό υπόβαθρο των βασικών εννοιών του Text Mining & Text Classification.

1.3 Δομή της διπλωματικής

Στο πρώτο μέρος καλύπτονται οι βασικές έννοιες και το απαραίτητο θεωρητικό υπόβαθρο που χρειάζεται ένας αναγνώστης ώστε να μπορέσει να κατανοήσει ένα σημαντικό τομέα της Εξόρυξης Πληροφορίας, τους Αλγορίθμους Ταξινόμησης Κειμένων. Μετά από τις θεμελιώδεις αρχές της Εξόρυξης Κειμένων παρατίθενται βασικά ζητήματα για την Εξόρυξη Πληροφορίας μέσα από κείμενα και ύστερα γίνεται μια εισαγωγή στο κομμάτι των Ταξινομητών Κειμένων, τα χαρακτηριστικά τους, τις γενικές αρχές και τα προβλήματα που παρουσιάζει ο καθένας από αυτούς. Έπειτα εξηγείται εκτενώς ο Αλγόριθμος των Geli Fei και Bing Liu (Fei, et al., 2016), ενώ στη συνέχεια γίνεται η κατασκευή και η περαιτέρω μελέτη των αποτελεσμάτων διαφόρων δοκιμών. Τέλος, γίνεται μια εκτίμηση απόδοσης του Αλγορίθμου και παρατίθενται τα προβλήματα, τα συμπεράσματα και μελλοντικές κινήσεις πάνω στο πεδίο για δημιουργία αποδοτικότερων αλγορίθμων και διαδικασιών.

2

Θεωρητικό Υπόβαθρο

2.1 Γενικά

Τα ψηφιακά δεδομένα που προκύπτουν από τα κείμενα σε ηλεκτρονική μορφή, εντός ή εκτός διαδικτύου, προσφέρουν ένα ενδιαφέρον πεδίο για τη μηχανική μάθηση λόγω της εγγενούς τους πολυπλοκότητας και πληροφοριακού πλούτου. Καθώς εξερευνούμε τον κόσμο της ανάλυσης κειμένου, δύο κύρια παραδείγματα μάθησης ξεχωρίζουν: τα εποπτευόμενα και τα μη εποπτευόμενα μοντέλα. Ενώ τα εποπτευόμενα μοντέλα, όπως τα SVMs και τα Νευρωνικά Δίκτυα, βασίζονται σε κατηγοριοποιημένα δεδομένα για να κάνουν προβλέψεις, τα μη εποπτευόμενα μοντέλα, όπως το LDA, επιδιώκουν να ανακαλύψουν κρυμμένες κλάσεις και δομές χωρίς ρητή καθοδήγηση (LeCun, et al., 2015).

Σε σύγκριση με την παραδοσιακή βάση δεδομένων, η οποία ασχολείται κυρίως με δομημένα δεδομένα (data), τα δεδομένα κειμένου είναι υψηλής διάστασης, πολύπλοκα και απαιτούν εξειδικευμένες τεχνικές προεπεξεργασίας και αναπαράστασης. Η ιδιαίτερη φύση των δεδομένων κειμένου απαιτεί βαθύτερη κατανόηση των πολυπλοκοτήτων τους και των εργαλείων που έχουν σχεδιαστεί ειδικά γι' αυτά.

Πέρα από τις βασικές προκλήσεις, η ανάλυση κειμένου προσφέρει μια πληθώρα ευκαιριών. Οι πληροφορίες που περιέχονται στα κείμενα είναι συχνά πλούσιες και σημαντικές, καθιστώντας την εξαγωγή γνώσης από αυτά κρίσιμη για πολλές εφαρμογές, από τη δημόσια έκφραση μέχρι την αναγνώριση οντοτήτων. Ωστόσο, η σωστή επεξεργασία του κειμένου απαιτεί μια σειρά από εξειδικευμένες τεχνικές, όπως η λημματοποίηση, η αφαίρεση ακαθόριστων όρων και η

αναγνώριση σύνθετων σχέσεων μεταξύ λέξεων. Επιπλέον, με την ταχεία ανάπτυξη των τεχνολογιών τεχνητής νοημοσύνης, οι μεθοδολογίες που χρησιμοποιούνται για την επεξεργασία και την ανάλυση κειμένου συνεχώς εξελίσσονται, επεκτείνοντας τις δυνατότητες και τα όρια του τι μπορεί να επιτευχθεί σε αυτό το πεδίο.

Ξεκινώντας από τους αλγόριθμους Εξόρυξης Κειμένου, πρέπει να αναφερθούμε στα προβλήματα που προκύπτουν κατά την διαδικασία της Εξόρυξης Κειμένου. Το βασικότερο ζήτημα εδώ είναι η Εξόρυξη Πληροφορίας από το ίδιο το κείμενο. Διάφορες διαδικασίες με διαφορετικούς κανόνες και δευτερεύουσες διαδικασίες υπάρχουν ούτως ώστε να γίνεται σωστά η Εξόρυξη Πληροφοριών. Μερικές από αυτές είναι η Αναγνώριση Οντοτήτων με Ονομασία/Named Entity Recognition, η Καθιέρωση Κανόνων/Rule-based Approach, η Εξαγωγή Σχέσεων/Relation Extraction καθώς και η Μη-Εποπτευόμενη Εξαγωγή Πληροφορίας (Mintz, et al., 2009).

Επιπλέον σημαντικό σημείο για την Κατηγοριοποίηση Κειμένων είναι η θεματική μοντελοποίηση η οποία αποτελεί διαδικασία ανακάλυψης κρυφών "θεμάτων" ή κατηγοριών μέσα σε μια συλλογή κειμένων. Δεν πρόκειται απλά για την αναζήτηση συχνών λέξεων ή φράσεων, αλλά για την κατανόηση των υποκείμενων θεματικών δομών που χαρακτηρίζουν τη συλλογή. Στην ουσία, η θεματική μοντελοποίηση αναζητά τα κυρίαρχα θέματα που καλύπτονται σε αυτό το σύνολο δεδομένων και πόσο συχνά εμφανίζεται το καθένα. Η θεματική μοντελοποίηση μπορεί να είναι ιδιαίτερα χρήσιμη σε μεγάλες συλλογές κειμένων, όπως βιβλιοθήκες, αρχεία ειδήσεων ή ακόμη και στα κοινωνικά μέσα ενημέρωσης, όπου η χειροκίνητη κατηγοριοποίηση μπορεί να είναι αδύνατη. Εργαλεία όπως το PLSI (Probabilistic Latent Semantic Indexing) και το LDA (Latent Dirichlet Allocation) (Deerweste, et al., 1990) χρησιμοποιούνται για την ανακάλυψη αυτών των κρυφών θεματικών δομών, βοηθώντας τους ερευνητές να κατανοήσουν καλύτερα τα δεδομένα

τους, να οργανώσουν το περιεχόμενο και να παράγουν σχετικές πληροφορίες με βάση το περιεχόμενο.

Σημαντική ενότητα για την Εξόρυξη Πληροφορίας και κεντρικό σημείο έρευνας της παρούσας εργασίας αποτελούν οι Αλγόριθμοι Ταξινόμησης Κειμένου/Text Classification.

2.2 Εξόρυξη Πληροφορίας από Κείμενο

Η εξαγωγή πληροφορίας από κείμενο αναφέρεται στη διαδικασία ανίχνευσης και εξαγωγής συγκεκριμένων πληροφοριών ή γνωστικών οντοτήτων από ανόργανο κείμενο. Στόχος είναι η αυτόματη μετατροπή αυτού του αδόμητου κειμένου σε μια πιο δομημένη μορφή, όπως μια βάση δεδομένων ή ένα σετ δομημένων εγγράφων. Στο πλαίσιο της εξαγωγής πληροφορίας, μπορούν να αναγνωριστούν και να εξαχθούν οντότητες (όπως ονόματα, οργανισμοί, τοποθεσίες), σχέσεις μεταξύ οντοτήτων και συγκεκριμένα γεγονότα ή γεγονότα που περιγράφονται στο κείμενο.

Οι τεχνικές εξαγωγής πληροφορίας είναι ιδιαίτερα χρήσιμες σε πληθώρα εφαρμογών, όπως η παρακολούθηση ειδήσεων, η έρευνα αγοράς και η ανάλυση δεδομένων κοινωνικών μέσων ενημέρωσης. Επιπλέον, με την εξαγωγή συγκεκριμένων κομματιών πληροφορίας, τα κείμενα γίνονται πιο προσβάσιμα και αναζητήσιμα. Εντούτοις, παρά την αυτοματοποίηση, η εξαγωγή πληροφορίας από κείμενο αντιμετωπίζει πολλές προκλήσεις, όπως η αναγνώριση αμφίσημων λέξεων, η προσαρμογή σε διάφορες γλώσσες ή διαλέκτους, και η διαχείριση συνεχώς εξελισσόμενου περιεχομένου. Ως εκ τούτου, η διαρκής βελτίωση των αλγορίθμων και των μεθόδων είναι απαραίτητη για την αυξανόμενη ακρίβεια και αποδοτικότητα της διαδικασίας.

2.3 NLP & Named Entity Recognition

Η Επεξεργασία Φυσικής Γλώσσας (NLP) είναι ένας υποτομέας της τεχνητής νοημοσύνης (AI) και της γλωσσολογίας που εστιάζει στην αλληλεπίδραση μεταξύ των υπολογιστών και της ανθρώπινης γλώσσας (Ratinov, et al., 2009). Ο πρωταρχικός του στόχος είναι να επιτρέψει στους υπολογιστές να κατανοήσουν, να ερμηνεύσουν και να δημιουργήσουν την ανθρώπινη γλώσσα με τρόπο ουσιαστικό και χρήσιμο. Το NLP περιλαμβάνει ένα ευρύ φάσμα εργασιών, όπως η μετάφραση γλώσσας, ανάπτυξη chatbot, σύνοψη κειμένου και άλλα. Ένα θεμελιώδες στοιχείο του NLP είναι η Αναγνώριση Ονομαζόμενης Οντότητας/Named Entity Recognition (NER), η οποία διαδραματίζει κρίσιμο ρόλο στην εξαγωγή δομημένων πληροφοριών από μη δομημένα δεδομένα κειμένου. Το NER είναι ένα βασικό εργαλείο για πολλές εφαρμογές NLP, όπως η εξαγωγή πληροφορίας, η αναζήτηση πληροφοριών και η αναλυτική εξόρυξη γνώμης.

Συγκεκριμένα το NER βοηθά στον εντοπισμό και την κατηγοριοποίηση οντοτήτων για τις μηχανές αναζήτησης για την ανάκτηση πιο συναφών αποτελεσμάτων. Είναι επίσης πολύτιμο σε συστήματα απάντησης ερωτήσεων, όπου μπορεί να βοηθήσει στον εντοπισμό των απαντήσεων μέσα σε έγγραφα κειμένου.

Δίνοντας έναν πρώτο ορισμό, ένα σύστημα NER θεωρείται η αναγνώριση οντοτήτων, εξαγωγή οντοτήτων ή τμηματοποίηση οντοτήτων και αποτελεί μια δευτερεύουσα εργασία του NLP που περιλαμβάνει τον εντοπισμό και την ταξινόμηση ονομασμένων οντοτήτων σε κείμενο. Οι επώνυμες οντότητες είναι λέξεις ή φράσεις που αντιπροσωπεύουν συγκεκριμένες οντότητες, όπως ονόματα ατόμων, οργανισμών, τοποθεσιών, ημερομηνίες, χρηματικές αξίες και άλλα (Jurafsky, et al., 2008).

Για την καλύτερη κατανόηση της λειτουργίας ενός NER παρατίθεται το παρακάτω παράδειγμα.

Σε ένα κείμενο που λέει "Ο Δημήτρης είναι έμπορος από την Αθήνα", το "Δημήτρης" θα αναγνωριστεί ως όνομα προσώπου και η "Αθήνα" ως τοποθεσία.

Τα βασικά προβλήματα που αντιμετωπίζει τα NER είναι η αναγνώριση οντοτήτων σε ένα συγκεκριμένο πλαίσιο, ο χειρισμός διαφορετικών ονομάτων και η ταξινόμηση οντοτήτων σε προκαθορισμένες κατηγορίες (π.χ. άτομο, οργανισμός, τοποθεσία). Για αυτό το λόγο τα NER αξιολογούνται για να μπορεί να γίνει κατανοητό το ποσοστό των επιδόσεων τους εξαιτίας αυτών των ανακρίβειών και σφαλμάτων.

Τα συστήματα NER αξιολογούνται με βάση μετρήσεις όπως η ακρίβεια δηλαδή ο λόγος των σωστά αναγνωρισμένων οντοτήτων προς όλες οι αναγνωρισμένες οντότητες, η ανάκληση δηλαδή ο λόγος των σωστά αναγνωρισμένων οντοτήτων προς όλες οι πραγματικές οντότητες (περισσότερα στο Κεφάλαιο 2.6), και το f1-score (ένας αρμονικός μέσος όρος ακρίβειας και ανάκλησης).

Η αναγνώριση ονοματισμένων οντοτήτων (Named Entity Recognition, NER) είναι μια διαδικασία στην επεξεργασία φυσικής γλώσσας (NLP) που στοχεύει στην εντοπισμό και ταξινόμηση ονοματισμένων οντοτήτων σε προκαθορισμένες κατηγορίες, όπως ονόματα προσώπων, οργανισμών, τοποθεσιών, χρονικές εκφράσεις, ποσά και άλλα (Nadeau, et al., 2007).

2.4 Διανυσματικές Αναπαραστάσεις και Στάθμιση χαρακτηριστικών

Για την μετατροπή ενός κειμένου σε διάνυσμα θα πρέπει να αποδοθεί και το αντίστοιχο βάρος στον κάθε όρο (token) (Manning, et al., 2008). Η συχνότερη μέθοδος για στάθμιση χαρακτηριστικών είναι το term frequency (tf). Χρησιμοποιώντας το tf μπορεί να αποδοθεί το κατάλληλο βάρος στον κάθε όρο μέσα σε ένα κείμενο και ανάμεσα από μια συλλογή κειμένων. Το term frequency (tf) εκφράζει πόσες φορές εμφανίζεται ένας όρος στο κείμενο, Σχέση (2.1). Ένας

όρος που εμφανίζεται πολλές φορές στο κείμενο θα έχει και μεγάλο tf , άρα θα θεωρείται σημαντικός για την σημασία του κειμένου.

Χρησιμοποιώντας το term frequency (tf) όλοι οι όροι θεωρούνται εξίσου σημαντικοί όσον αφορά την συνάφεια τους με το περιεχόμενο του κειμένου. Το term frequency inverse document frequency ($tf-idf$) καταφέρνει και προσπερνά αυτό το πρόβλημα αποδίδοντας την βαρύτητα του όρου μέσα από μια συλλογή κειμένων, Σχέση (2.3). Για να υπολογιστεί το ($tf-idf$), πρώτα βρίσκουμε το inverse document frequency (idf), Σχέση (2.2). Παίρνουμε τον λογάριθμο του συνολικού πλήθους των κειμένων προς το πλήθος των κειμένων που εμφανίζεται ο συγκεκριμένος όρος. Στην συνέχεια συνδυάζουμε τα (tf) και (idf) παίρνοντας το εσωτερικό γινόμενο τους για έναν όρο και έχουμε το ($tf-idf$) του όρου, Σχέση (2.3). Ένας όρος που εμφανίζεται μερικές φορές σε ένα κείμενο ή ανάμεσα σε πολλά κείμενα θα έχει μικρότερο $tf-idf$ σε σχέση με έναν όρο που εμφανίζεται πολλές φορές σε μερικά από τα κείμενα και θα έχει το μικρότερο $tf-idf$ αν εμφανίζεται σε όλα τα κείμενα.

$$tf(t) = \frac{\text{πλήθος φορές που εμφανίζεται ο όρος } t \text{ στο κείμενο}}{\text{πλήθος όλων των όρων στο κείμενο}} \quad (2.1)$$

$$idf(t) = \log\left(\frac{\text{συνολικό πλήθος κειμένων}}{\text{πλήθος κειμένων στα οποία εμφανίζεται ο όρος } t}\right) \quad (2.2)$$

$$tf_idf(t) = tf(t) \times idf(t) \quad (2.3)$$

2.5 Ταξινόμηση με Κανόνες

Η προσέγγιση με βάση τους κανόνες αναφέρεται σε μια μεθοδολογία όπου οι αποφάσεις λαμβάνονται με βάση ένα σύνολο προκαθορισμένων κανόνων ή ευριστικών μεθόδων. Συνήθως

αυτοί οι κανόνες έχουν διαμορφωθεί χειροκίνητα από ειδικούς σε έναν συγκεκριμένο τομέα και έχουν προγραμματιστεί ρητά σε ένα σύστημα.

Μάλιστα στην επεξεργασία ενός κειμένου, το πρώτο βήμα είναι η διαδικασία του tokenization, δηλαδή η μετατροπή του κειμένου σε μια σειρά από διακριτικά/tokens. Αυτά τα tokens αποτελούν τη βασική μονάδα με την οποία εργαζόμαστε όταν δημιουργούμε κανόνες.

Στο πλαίσιο της Αναγνώρισης Ονοματισμένων Οντοτήτων (NER) ή άλλων εργασιών επεξεργασίας κειμένου, μια προσέγγιση με βάση τους κανόνες περιλαμβάνει πρωτίστως τα πρότυπα συντακτικής ανάλυσης, τις λίστες λεξικού, τους κανόνες αναφοράς καθώς και τις Κανονικές Εκφράσεις (Regular Expressions). Τα πρώτα, χρησιμοποιούνται για τον εντοπισμό συγκεκριμένων δομών στο κείμενο. Ενώ οι λίστες λεξικού βρίσκουν χρήση στον εντοπισμό συγκεκριμένων λέξεων ή φράσεων που ενδέχεται να αναφέρονται σε οντότητες. Τα Regular Expressions χρησιμοποιούνται για τον εντοπισμό μοτίβων στο κείμενο, όπως τηλεφωνικοί αριθμοί ή διευθύνσεις email. Τέλος, οι κανόνες αναφοράς λαμβάνουν υπόψη το περιβάλλον μιας λέξης ή φράσης και έπειτα γίνεται λήψη της απόφασης για το αν αναφέρεται σε μια γνωστή οντότητα ή όχι.

Παρόλο που η προσέγγιση με βάση τους κανόνες είναι πολύ διαισθητική και μπορεί να προσφέρει υψηλή ακρίβεια σε συγκεκριμένα σύνολα δεδομένων, απαιτεί πολύ χρόνο για την ανάπτυξη των κανόνων και μπορεί να μην αρκεί σε νέα ή διαφορετικά δεδομένα.

2.5.1 Ταξινόμηση με Βάση τα Στατιστικά

Η στατιστική μάθηση αφορά τη χρήση στατιστικών μεθόδων για την αναγνώριση προτύπων σε δεδομένα. Στον τομέα της επεξεργασίας φυσικής γλώσσας και της αναγνώρισης πληροφοριών,

υπάρχουν διάφορες στατιστικές μέθοδοι που χρησιμοποιούνται Κρυφά Μοντέλα Μάρκοβ (Hidden Markov Models - HMMs), τα Μοντέλα Μάρκοβ Μεγίστης Εντροπίας (Maximum Entropy Markov Models - MEMMs) και τέλος τα Πεδία Τυχαίων Συνθηκών (Conditional Random Fields - CRFs). Κάθε μία από αυτές τις μεθόδους έχει τα δικά της πλεονεκτήματα και μειονεκτήματα ανάλογα με την εφαρμογή και τη φύση των δεδομένων. Σε γενικές γραμμές, όλες στοχεύουν στην ανακάλυψη στατιστικών συσχετίσεων μεταξύ των δεδομένων για την πρόβλεψη και την ανάλυση φυσικής γλώσσας (Lafferty, et al., 2001).

HMMs

Τα Κρυφά Μοντέλα Μάρκοβ (HMMs) είναι στοχαστικά μοντέλα που χρησιμοποιούνται για την πρόβλεψη ακολουθιών. Κάθε κατάσταση σε ένα HMM αντιπροσωπεύει ένα διακριτό συμβάν, και οι μεταβάσεις μεταξύ των καταστάσεων περιγράφονται με πιθανότητες. Με εφαρμογές στη διαδικασία των Parts of Speech.

MEMMs

Τα Μοντέλα Μάρκοβ Μεγίστης Εντροπίας (MEMMs) είναι μια επέκταση των HMMs όπου κάθε πιθανότητα μετάβασης συνδέεται με ένα σύνολο χαρακτηριστικών, επιτρέποντας το μοντέλο να προβλέπει βάσει μεγαλύτερου πλήθους πληροφοριών. Χρησιμοποιούνται στην αναγνώριση ήδη γνωστών και κατηγοριοποιημένων οντοτήτων, στο Part Of Speech (POS) tagging και στη διαδικασία του chunking.

CRFs:

Σε αντίθεση με τα HMMs και τα MEMMs που προβλέπουν για κάθε σημείο ανεξάρτητα, τα Πεδία Τυχαίων Συνθηκών (CRFs) λαμβάνουν υπόψη την πληροφορία από ολόκληρη την ακολουθία. Αυτό επιτρέπει στα CRFs να λειτουργούν καλύτερα σε προβλήματα που η ακολουθία

των λέξεων είναι κρίσιμη. Όπως και τα MEMMs κάνουν αναγνώριση ονοματισμένων οντοτήτων και το POS tagging.

2.5.2 Ταξινόμηση με Βάση τα Χαρακτηριστικά

Η ταξινόμηση με βάση τα χαρακτηριστικά (feature-based classification) αναφέρεται στη διαδικασία κατηγοριοποίησης αντικειμένων ή οντοτήτων με βάση τα χαρακτηριστικά (ή τα χαρακτηριστικά γνωρίσματα) που παρουσιάζουν. Στο πλαίσιο της επεξεργασίας φυσικής γλώσσας και της αναλυτικής κειμένου, αυτά τα χαρακτηριστικά μπορεί να είναι λέξεις, φράσεις, γραμματικές σχέσεις, συναισθηματικοί τόνοι και άλλα γλωσσικά ή στατιστικά στοιχεία.

Στην ταξινόμηση με βάση τα χαρακτηριστικά, το πρώτο βήμα είναι η εξαγωγή και η αναπαράσταση αυτών των χαρακτηριστικών από τα δεδομένα. Στη συνέχεια, χρησιμοποιείται μια τεχνική ταξινόμησης (όπως η μηχανική μάθηση) για να "μάθει" από τα χαρακτηριστικά αυτά και να προβλέψει την κατηγορία του κάθε δείγματος.

Ας εξετάσουμε τους διάφορους τύπους χαρακτηριστικών που χρησιμοποιούνται συνήθως στην ταξινόμηση κειμένου και στην αναγνώριση ονοματισμένων οντοτήτων:

Entity Features

Περιγράφουν απευθείας την οντότητα που εξετάζεται. Για παράδειγμα, αν η οντότητα είναι μια λέξη ή φράση σε ένα κείμενο, τα χαρακτηριστικά της οντότητας μπορεί να περιλαμβάνουν τη μορφή της λέξης: π.χ., "Νέα Υόρκη", το μήκος της λέξης, τον τρόπο κεφαλαίωσης δηλαδή αν η λέξη έχει κεφαλαίο το πρώτο γράμμα, είναι όλα κεφαλαία ή πεζά. Επίσης τα χαρακτηριστικά περιλαμβάνουν και τα πιθανά προθέματα και επιθέματα της λέξης καθώς και αν η λέξη περιέχει ψηφία, ειδικούς χαρακτήρες κτλ.

Lexical Contextual Features

Καταγράφουν το τοπικό πλαίσιο της οντότητας, συχνά περιλαμβάνοντας τις λέξεις ή τα σύμβολα που βρίσκονται αμέσως γύρω της.

1. Η προηγούμενη ή η επόμενη λέξη σε σχέση με την οντότητα.
2. N-grams: ακολουθίες λέξεων ή συμβόλων γύρω από την οντότητα.
3. Μέρος του λόγου των γειτονικών λέξεων.

Syntactic Contextual Features

Καταγράφουν τις γραμματικές σχέσεις και δομές γύρω από την οντότητα.

1. Σχέσεις δέντρου εξαρτήσεων: Σχέσεις μεταξύ της οντότητας και άλλων λέξεων βάσει ενός δέντρου ανάλυσης.
2. Μέρος του λόγου της οντότητας: Εάν η λέξη είναι ουσιαστικό, ρήμα, επίθετο κτλ.
3. Πληροφορίες σχετικά με φράσεις: Αν η οντότητα είναι μέρος μιας φράσης ουσιαστικού, ρήματος κτλ.

Background Knowledge

Αφορά εξωτερική γνώση, η οποία μπορεί να μην είναι παρούσα στο κείμενο, αλλά μπορεί να βοηθήσει στην ταξινόμηση.

1. Οντολογίες ή ταξινομίες που κατηγοριοποιούν την οντότητα.
2. Γράφοι γνώσης ή βάσεις δεδομένων όπως το DBpedia, το YAGO κτλ.
3. Σημασιολογικοί ρόλοι της οντότητας σε σχέση με ένα ρήμα ή δράση.
4. Ενσωματώσεις από προεκπαιδευμένα μοντέλα όπως το Word2Vec, BERT κ.λπ., τα οποία αποτυπώνουν τη σημασιολογική σημασία από μεγάλα σώματα κειμένου.

Η ακριβής επιλογή και συνδυασμός αυτών των χαρακτηριστικών εξαρτάται από το συγκεκριμένο έργο και τη φύση των δεδομένων. Για την αναγνώριση ονοματισμένων οντοτήτων, για παράδειγμα, τα χαρακτηριστικά της οντότητας και τα λεξικά χαρακτηριστικά πλαισίου είναι εξαιρετικά σημαντικά, καθώς παρέχουν άμεσες πληροφορίες σχετικά με την οντότητα και το άμεσο πλαίσιο της. Τα συντακτικά χαρακτηριστικά μπορούν να βοηθήσουν στην κατανόηση του γραμματικού ρόλου της οντότητας, ενώ η γνώση πλαισίου μπορεί να παρέχει περισσότερο βάθος και πλούτο, ιδιαίτερα όταν το πλαίσιο στα δεδομένα είναι περιορισμένο.

2.6 Εκτίμηση Ανάκτησης Πληροφορίας

1. Ακρίβεια (Precision) & Ανάκληση (Recall):

Η ακρίβεια και η ανάκληση είναι δύο βασικές μετρικές που αξιολογούν την αποτελεσματικότητα ενός συστήματος ανάκτησης πληροφοριών.

Η ακρίβεια αντιπροσωπεύει το ποσοστό των σχετικών αποτελεσμάτων στα συνολικά επιστρεφόμενα αποτελέσματα, ενώ η ανάκληση αντιπροσωπεύει το ποσοστό των σχετικών αποτελεσμάτων που επιστράφηκαν αναφορικά με όλα τα πραγματικά σχετικά αποτελέσματα (Baeza-Yates, et al., 1999).

$$\text{Ακρίβεια} = \frac{\text{Σχετικά Αποτελέσματα}}{\text{Επιστρεφόμενα Αποτελέσματα}} \quad (2.4)$$

$$\text{Ανάκληση} = \frac{\text{Σχετικά Αποτελέσματα}}{\text{Πραγματικά Σχετικά Αποτελέσματα}} \quad (2.5)$$

2. Mean Average Precision (MAP):

Η MAP είναι μια μετρική που λαμβάνει υπόψη τόσο την ακρίβεια όσο και την ανάκληση στα καταταγμένα αποτελέσματα. Αυτή η μετρική υπολογίζει τον μέσο όρο της ακρίβειας σε κάθε δυνατό σημείο ανάκλησης, παρέχοντας ένα μέτρο της συνολικής απόδοσης του συστήματος ανάκτησης πληροφοριών (Manning, et al., 2008), όπως φαίνεται παρακάτω :

$$MAP = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \left(\frac{1}{m_q} \sum_{k=1}^{m_q} Precision \right) \quad (2.6)$$

3. Normalized Discounted Cumulative Gain (NDCG):

Η NDCG είναι μια μετρική που χρησιμοποιείται για να αξιολογήσει την ποιότητα των καταταγμένων λιστών αποτελεσμάτων λαμβάνοντας υπόψη την πληρότητα και τη σχετικότητα.

$$NDCGk = \frac{DCGk}{IDCGk} \quad (2.7)$$

Όπου στην παραπάνω σχέση το DCG είναι το Discounted Cumulative Gain και το IDCG είναι το ιδανικό DCG.

3

Text Classification Algorithms

3.1 Δέντρα Αποφάσεων

Τα δέντρα αποφάσεων είναι μια δημοφιλής μέθοδος για την κατηγοριοποίηση δεδομένων. Η βασική ιδέα είναι η αναδρομική διαίρεση του χώρου των δεδομένων με βάση τα χαρακτηριστικά τους (Quinlan, et al., 1993). Τα Δέντρα Αποφάσεων είναι ένας τύπος αλγορίθμου επιβλεπόμενης μάθησης που χρησιμοποιείται κυρίως για προβλήματα ταξινόμησης. Λειτουργούν διαχωρίζοντας το σύνολο δεδομένων σε υποσύνολα βάσει διαφορετικών συνθηκών, καταλήγοντας τελικά σε μια απόφαση.

Η διαδικασία λήψης αποφάσεων ξεκινά από τη ρίζα (πάνω) και φτάνει ως τα φύλλα (κάτω), ακολουθώντας τις συνθήκες που ορίζονται από τους εσωτερικούς κόμβους. Η κύρια πρόκληση στη δημιουργία δέντρων αποφάσεων είναι να προσδιορίσουμε σε ποια χαρακτηριστικά θα γίνει η διαίρεση και πότε θα σταματήσει η διαίρεση.

Αλγόριθμοι όπως οι ID3, C4.5, και CART χρησιμοποιούνται συνήθως για τη δημιουργία δέντρων αποφάσεων (Quinlan, 1986).

3.2 Rule-Based Classifiers

Οι ταξινομητές βασισμένοι σε κανόνες λαμβάνουν αποφάσεις βάσει ενός σετ προκαθορισμένων κανόνων. Αυτοί οι κανόνες προκύπτουν συχνά από τα ίδια τα δεδομένα χρησιμοποιώντας

μεθόδους που παράγουν ακριβείς και ερμηνευτικούς κανόνες. Οι κανόνες συχνά έχουν τη μορφή "if-then" (Cohen, 1995).

Μία κοινή μέθοδος για τη δημιουργία κανόνων είναι ο Αλγόριθμος Διαδοχικής Κάλυψης. Ο Αλγόριθμος Διαδοχικής Κάλυψης είναι μια κλασική μέθοδος που χρησιμοποιείται για τη δημιουργία κανόνων για rule-based classifiers. Λειτουργεί με έναν απλό αλλά αποτελεσματικό τρόπο, όπως περιγράφεται στην συνέχεια:

1. Αρχικά δημιουργείται ένα άδειο σετ κανόνων.
2. Ενώ υπάρχουν θετικά παραδείγματα:
 - a. Ο αλγόριθμος “μαθαίνει” έναν μεμονωμένο κανόνα από τα υπόλοιπα θετικά παραδείγματα.
 - b. Ο αλγόριθμος αφαιρεί από το σετ δεδομένων όλα τα θετικά παραδείγματα που καλύπτονται από τον νεοσύστατο κανόνα.
3. Τέλος ο αλγόριθμος επαναλαμβάνεται μέχρι να καλυφθούν όλα τα θετικά παραδείγματα ή μέχρι να εκπληρωθεί ένα προκαθορισμένο κριτήριο διακοπής.

Στόχος είναι να βρεθεί ο κανόνας που ταξινομεί σωστά όσα περισσότερα θετικά παραδείγματα είναι δυνατόν (μεγιστοποιώντας την κάλυψη) ελαχιστοποιώντας ταυτόχρονα την ταξινόμηση αρνητικών παραδειγμάτων. Αρκετοί αλγόριθμοι, όπως το CN2, χρησιμοποιούν παραλλαγές αυτής της μεθόδου (Clark, et al., 1989).

3.3 SVM Classifiers

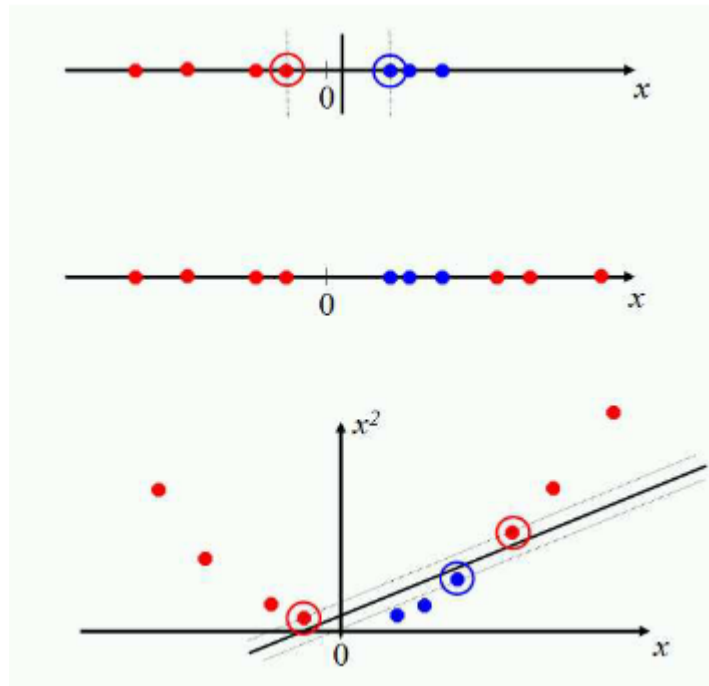
Οι αλγόριθμοι Support Vector Machine Classifiers βοηθούν στην ταξινόμηση κειμένου και αποτελούν ισχυρό εργαλείο για την ανάλυση και επεξεργασία γλωσσικών δεδομένων.

Σε αντίθεση με άλλους αλγορίθμους, οι Μηχανές Υποστήριξης επιδιώκουν την εύρεση ενός hyperplane/υπερεπιπέδου που να διαχωρίζει αποτελεσματικά τις διάφορες κατηγορίες κειμένου/κλάσεις μεταξύ τους και να κατατάσει αντίστοιχα τα αποτελέσματα σε αυτές. Μάλιστα τα SVM χρησιμοποιούν την ιδέα του hyperplane για να διαχωρίσουν τα δεδομένα σε δύο ή περισσότερες κατηγορίες.

Η βάση αυτού του μοντέλου χωρίζεται σε 2 σημαντικά στάδια. Το πρώτο στάδιο είναι το στάδιο της “εκπαίδευσης” του αλγορίθμου, στο οποίο στη παρούσα εργασία θα αποκαλούμε και training και το οποίο αφορά πρακτικά τη διαδικασία κατά την οποία γίνεται ο καθορισμός των κλάσεων του περιβάλλοντος. Συνοπτικά στο στάδιο της εκπαίδευσης, ο αλγόριθμος “εκπαιδεύεται ” έτσι ώστε το κάθε κείμενο να ανήκει σε μία από τις προκαθορισμένες κατηγορίες. Κατά τη διάρκεια της φάσης του testing, αξιολογούμε την απόδοση του αλγορίθμου ως προς το πόσο αποτελεσματικά δημιουργεί το hyperplane, ώστε τα κείμενα του testing να αποδοθούν στην κατηγορία που ανήκουν. Τα SVM επιτρέπουν την αποτελεσματική αντιμετώπιση προβλημάτων πολυκατηγορικής ταξινόμησης κειμένου και αποτελούν ιδανική επιλογή για εφαρμογές όπως η αυτόματη κατηγοριοποίηση κειμένων, όπως θα δούμε και στο πλαίσιο της παρούσας εργασίας.

Τα SVM επιλύονται τυπικά χρησιμοποιώντας μεθόδους βελτιστοποίησης όπως οι αλγόριθμοι τετραγωνικού προγραμματισμού.

Για λόγους κατανόησης του αναγνώστη σχετικά με τη δημιουργία του hyperplane, παρατίθενται παρακάτω τα σχεδιαγράμματα, Σχήμα (3.1), από το βιβλίο (Manning, et al., 2008).



Σχήμα 3.1 : Αναπαράσταση δεδομένων από τον μονοδιάστατο στον δισδιάστατο χώρο. (Manning, et al., 2008)

Στην πρώτη εικόνα του Σχήμα (3.1), βλέπουμε σημεία δεδομένων (σε μπλε και κόκκινο) κατανομημένα κατά μήκος μιας μονοδιάστατης κλίμακας (άξονας x). Τα σημεία διαχωρίζονται γραμμικά από μια κάθετη μαύρη γραμμή. Στο σχήμα ακριβώς μετά, όμως, βλέπουμε μια διαφορετική κατανομή των σημείων στην οποία είναι προφανές ότι αυτά τα σημεία δεν είναι γραμμικά διαχωρίσιμα σε αυτόν τον χώρο.

Αυτό μπορεί να αντιμετωπιστεί όπως βλέπουμε στην τελευταία εικόνα του Σχήμα (3.1), εισάγεται δηλαδή ένας μετασχηματισμός, προβάλλοντας τα δεδομένα σε έναν χώρο μεγαλύτερης διάστασης (ειδικά, δισδιάστατο εδώ, με άξονες x και x^2). Σε αυτόν τον μετασχηματισμένο χώρο, τα προηγούμενα μη γραμμικά διαχωρίσιμα σημεία δεδομένων γίνονται γραμμικά διαχωρίσιμα, όπως φαίνεται από την ευθεία γραμμή (το hyperplane) που διαχωρίζει τα μπλε και κόκκινα σημεία.

Ουσιαστικά στο παραπάνω μοντέλο, γίνεται η χρήση μετασχηματισμών οι οποίοι μπορούν να βρουν τους ιδανικούς πολυδιάστατους χώρους ώστε τα “σημεία” να μπορούν να διαχωριστούν γραμμικά.

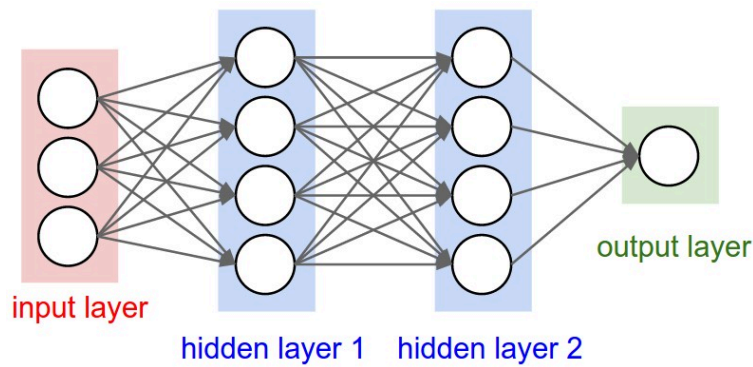
3.4 Neural Network Classifiers

Τα νευρωνικά δίκτυα είναι συστήματα εμπνευσμένα από τη δομή του ανθρώπινου εγκεφάλου και χρησιμοποιούνται για διάφορες εφαρμογές, όπως η κατηγοριοποίηση και η παλινδρόμηση. (Rumelhart, et al., 1986).

Οι ταξινομητές νευρωνικών δικτύων είναι ένα υποσύνολο των μοντέλων μηχανικής μάθησης που εμπνέονται από τον ανθρώπινο εγκέφαλο. Αποτελούνται από διασυνδεδεμένους κόμβους (ή “νευρώνες”) που επεξεργάζονται πληροφορίες σε σειριακά επίπεδα.

Οι Neural Network Classifiers/Ταξινομητές Νευρωνικών Δικτύων, έχουν συγκεκριμένη δομή, μια συνάρτηση ενεργοποίησης και την διαδικασία εκπαίδευσής τους .

Η δομή τους αποτελείται από 3 επίπεδα. Το Input Layer, τα Hidden Layers και το Output Layer. Το Input Layer, πρακτικά λαμβάνει τα αρχικά δεδομένα εισόδου. Κάθε νευρώνας αντιστοιχεί σε ένα χαρακτηριστικό των δεδομένων. Τα Hidden Layers επεξεργάζονται τα δεδομένα από το επίπεδο εισόδου χρησιμοποιώντας βάρη και συναρτήσεις ενεργοποίησης. Τέλος το Output Layer, δίνει ως έξοδο την τελική κλάση που ανήκουν τα δεδομένα.



Σχήμα 3.2 : Ένα τυπικό σύστημα Νευρωνικού Δικτύου που χρησιμοποιείται για Ταξινόμηση (Aggarwal, et al., 2012).

Για λόγους κατανόησης του αναγνώστη, παρατίθεται το παραπάνω σχεδιάγραμμα, Σχήμα (3.2), από το βιβλίο: (Aggarwal, et al., 2012).

3.5 Bayes Classifiers

Αυτοί οι κατηγοριοποιητές χρησιμοποιούν τις αρχές της στατιστικής Bayes για να υπολογίσουν την πιθανότητα που έχει ένα δείγμα να ανήκει σε μια συγκεκριμένη κατηγορία βάση των χαρακτηριστικών του. Ο πιο γνωστός εκπρόσωπος αυτής της κατηγορίας είναι ο Naive Bayes Classifier (Rish, 2001).

4

Ανάλυση Αλγορίθμων Open-Set

Σε αυτό το κεφάλαιο, γίνεται αρχικά ανάλυση εις βάθος των τεχνικών που χρησιμοποιούνται για την Ταξινόμηση Κειμένου. Επιπλέον, γίνεται αναφορά σε θεωρητικά στοιχεία και όροι που χρησιμοποιούνται για την μελέτη του Αλγορίθμου και για την αναπτυξη του αλγορίθμου των Geli Fei και Bing Liu (Fei,et al., 2016). Τέλος, γίνεται μια παρουσίαση και ανάλυση του αλγορίθμου των Fei G. και Lui B., όπως τον αναφέρουν στα (Fei,et al., 2015) και (Fei,et al., 2016).

Όπως έγινε αναφορά και στο Κεφάλαιο 2 στον τομέα της μηχανικής μάθησης, τα Support Vector Machine (SVM) χρησιμοποιούνται συνήθως για εργασίες ταξινόμησης. Οι ταξινομητές SVM λειτουργούν βρίσκοντας το βέλτιστο hyperplane που διαχωρίζει τα σημεία δεδομένων σε διαφορετικές κλάσεις. Είναι ιδιαίτερα αποτελεσματικά σε προβλήματα δυαδικής ταξινόμησης, όπου ο στόχος είναι να γίνει διάκριση μεταξύ δύο κλάσεων/κατηγοριών. Οι ταξινομητές SVM προσφέρουν υψηλές ταχύτητες ακρίβειας και ανάκλησης όταν έχουν ρυθμιστεί σωστά, καθιστώντας τους ένα πολύτιμο εργαλείο σε πολλές εφαρμογές.

Στο πλαίσιο της μηχανικής μάθησης, η μετάβαση από την binary (δυαδική) ταξινόμηση στην multiclass (πολλές κλάσεις) ταξινόμηση περιλαμβάνει την επέκταση της ικανότητας του ταξινομητή να αναγνωρίζει περισσότερες από δύο κατηγορίες. Αυτό τυπικά απαιτεί την τροποποίηση του αλγορίθμου για τον χειρισμό πολλαπλών “ετικετών” κλάσεων και μπορεί να περιλαμβάνει τεχνικές όπως η ταξινόμηση one-vs-all (ή αλλιώς one-vs-rest) ή one-vs-one.

Τα δομημένα σύνολα δεδομένων είναι οργανωμένα δεδομένα που ακολουθούν μια σαφή μορφή, καθιστώντας ευκολότερη την εργασία με τους αλγόριθμους μηχανικής εκμάθησης. Το άνοιγμα σε μη δομημένο θόρυβο αναφέρεται στην ευαισθησία του συνόλου δεδομένων σε θόρυβο ή άσχετες πληροφορίες επηρεάζουν αρνητικά την απόδοση του classifier.

4.1 Περιγραφή Αλγορίθμου *cbsSVM*

Ο αλγόριθμος των Geli Fei και Bing Liu που πραγματεύεται η παρούσα εργασία είχε αρχικά δημοσιευθεί ως λύση για το covariate shift στην δυαδική ταξινόμηση κειμένου (Fei, et al., 2015). Ένα πρόβλημα που εμφανίζεται στην περίπτωση όπου θεωρούμε ότι τα δεδομένα του testing έχουν ίδια διασπορά με τα δεδομένα του training. Η υπόθεση πως τα δεδομένα του training και του testing θα έχουν την ίδια κατανομή, δεν εμπίπτει σε πολλές περιπτώσεις της ταξινόμησης κειμένων ανοικτού συνόλου, όπως στην ταξινόμηση δεδομένων από τα Social Media. Ο αλγόριθμος που προτείνουν για εποπτευόμενη μάθηση λέγεται **CBS-L** (Center-Based Similarity Learning). Η βασική διαφορά του είναι ότι η εκπαίδευση δεν γίνεται με τις αναπαραστάσεις των κειμένων στο document space (DS) με n-gram χαρακτηριστικά αλλά με τις αναπαραστάσεις τους σε έναν άλλο χώρο, τον center-based similarity space (CBS). Σε μια συνηθισμένη εκπαίδευση ενός SVM στο document space, η θετική κλάση ορίζεται με ένα hyperplane, το οποίο χωρίζει τον χώρο των θετικών παραδειγμάτων να είναι ένα άπειρο ημισφαίριο. Αυτό σημαίνει ότι το open space είναι πολύ μεγάλο και κατά συνέπεια θα είναι πολύ μεγάλο το open space risk. Σε αντίθεση, ο αλγόριθμος CBS-L, περιορίζει την θετική κλάση σε έναν πεπερασμένο χώρο στο similarity space, που είναι μια σφαίρα στο document space. Κατά αυτόν τον τρόπο το open space risk μειώνεται σημαντικά.

Σε συνέχεια οι ίδιοι ανέφεραν ότι ο συγκεκριμένος αλγόριθμος μπορεί να εφαρμοστεί σε ταξινομήσεις ανοικτού συνόλου δίνοντας πολύ καλά αποτελέσματα (Fei, et al., 2016). Κατά την εφαρμογή του σε αυτή την περίπτωση ο αλγόριθμος ονομάζεται cbsSVM (center-based similarity based on SVM).

Η βασική ιδέα πίσω από τον cbsSVM είναι αντί να γίνεται ο διαχωρισμός των κειμένων στο document space (DS) να γίνεται σε έναν similarity space. Πρέπει αρχικά να μεταφερθούν τα κείμενα από το document space σε έναν center-based similarity space (CBS). Σε αυτόν τον νέο χώρο το covariate shift είναι σημαντικά μικρότερο, το οποίο προσφέρει καλύτερα αποτελέσματα ακρίβειας από τους ταξινομητές.

Στην συνέχεια παραθέτουμε και εξηγούμε έννοιες που είναι χρήσιμες για την κατανόηση του αλγορίθμου.

4.1.1 Open Space Risk

Μια σημαντική διάκριση στις εργασίες ταξινόμησης είναι μεταξύ προβλημάτων ανοικτού και κλειστού συνόλου (open-set vs closed-set). Σε ένα σενάριο κλειστού συνόλου, όλες οι κλάσεις είναι γνωστές και προκαθορισμένες, καθιστώντας ευκολότερο τον σχεδιασμό ενός classifier. Αντίθετα, τα ανοιχτά προβλήματα περιλαμβάνουν κλάσεις που μπορεί να μην είναι προκαθορισμένες, τα οποία αποτελούν ένα πιο ρεαλιστικό σενάριο επειδή σε πολλές περιπτώσεις δεν είναι γνωστές όλες οι πιθανές κλάσεις κατά την διάρκεια της εκπαίδευσης. Με τα προβλήματα ανοικτού συνόλου εισάγονται ποικίλες προκλήσεις, με μια από αυτές να είναι το Open Space Risk/ Ρίσκο Ανοικτού Συνόλου. Το Open Space Risk αναφέρεται στην πιθανή εσφαλμένη ταξινόμηση σημείων δεδομένων που δεν ανήκουν σε καμία γνωστή κλάση. Το Open Space Risk είναι αλληλένδετο με το open space, δηλαδή τον θετικό χώρο αρκετά μακριά από το κέντρο των θετικών παραδειγμάτων του training. Όσο μεγαλύτερο είναι το Open Space τόσο μεγαλύτερο

γίνεται και το Open Space Risk. Συγκεκριμένα, όταν το Open Space είναι μεγάλο, τότε ένα παράδειγμα που βρίσκεται σε μέσα σε αυτό θα χαρακτηριστεί ως θετικό ακόμα και αν βρίσκεται μακριά από τα θετικά παραδείγματα του training. Το πιθανοτικό Open Space Risk $R_o(f)$ μιας συνάρτησης f για μια κλάση y δίνεται από την Σχέση (4.1), (Scheirer, et al., 2013). Το S_o είναι μια μεγάλη σφαίρα με ακτίνα r_o που περιλαμβάνει τον θετικό ανοιχτό χώρο και τα θετικά παραδείγματα του training, και η f είναι μια μετρήσιμη συνάρτηση ταξινόμησης η οποία όταν αναγνωρίζει μια κλάση y επιστρέφει 1 ($f_y(x) = 1$) ή αν δεν αναγνωρίζει σε ποια κλάση ανήκει επιστρέφει 0 ($f_y(x) = 0$). Από την άλλη το O εκφράζει μόνο τον θετικό ανοιχτό χώρο.

$$R_o(f) = \frac{\int_{S_o} f_y(x) dx}{\int_{S_o} f_y(x) dx} \quad (4.1)$$

Από την παραπάνω σχέση, συμπεραίνουμε ότι είναι κρίσιμο να ορίζεται ο μικρότερος δυνατός χώρος για το Open Space ώστε να αποδίδονται μικρότερες τιμές στο Open Space Risk. Σκοπός, λοιπόν, είναι να μειωθεί το Open Space χωρίς να παραληφθούν τα θετικά παραδείγματα. Η αντιμετώπιση αυτού του ζητήματος συχνά περιλαμβάνει τεχνικές όπως η δημιουργία ενός hyperplane για τον καλύτερο καθορισμό των ορίων κλάσης ή με την εύρεση μιας βέλτιστης ακτίνας σε ένα συγκεκριμένο επίπεδο. Μια ακόμα δυσκολία έγκειται στο ότι δεν υπάρχει σαφής ορισμός για το Open Space. Τελικά, στο (Fei, et al., 2016), το Open Space O ορίζεται από την Σχέση (4.2) :

$$O = S_o - B_{r_y}(cen_y) \quad (4.2)$$

όπου το $B_{r_y}(cen_y)$ είναι μια κλειστή σφαίρα ακτίνας r_y που ιδανικά περιέχει όλα τα θετικά παραδείγματα της κλάσης y και S_o μια μεγαλύτερη σφαίρα $B_{r_o}(cen_o)$ ίδιου κέντρου και ακτίνας r_o που περιλαμβάνει όλες τις θετικές περιπτώσεις μιας θετικής κλάσης και τον θετικό ανοιχτό

χώρο. Η συνάρτηση ταξινόμησης επιστρέφει $f_y(x) = 1$ αν $x \in B_{r_o}(cen_y)$ ή διαφορετικά επιστρέφει $f_y(x) = 0$. Η ακτίνα r_o καθορίζεται από τα δεδομένα της εκπαίδευσης για κάθε γνωστή θετική κλάση. Από αυτόν τον τύπο προκύπτει ότι το Open Space είναι μια πεπερασμένη σφαιρική περιοχή η οποία μειώνει σημαντικά το Open Space Risk συγκριτικά με έναν παραδοσιακό SVM ή 1-vs-Set Machine (Scheirer, et al., 2013).

4.1.2 Διανυσματικές Αναπαραστάσεις

Είναι απαραίτητη η επεξεργασία ενός κειμένου με έναν τρόπο ώστε να είναι κατανοητό από τον υπολογιστή, επειδή ένα κείμενο σε plain text δεν είναι κάτι που μπορεί να επεξεργαστεί. Η επεξεργασία του κειμένου αφορά μόνο το περιεχόμενο του και όχι άλλες πληροφορίες όπως η δομή του ή το URL, αν και είναι εξίσου σημαντικά μπορεί να επηρεάζονται από τεχνολογικές αλλαγές κατά την πάροδο του χρόνου. Η διανυσματική αναπαράσταση του κειμένου μπορεί να γίνει με πολλούς διαφορετικούς τρόπους.

Στην παρούσα χρησιμοποιούμε τέσσερις τύπους token όπως στην διατριβή (Pritsos, et al., 2019) :

- Character 4-grams (C4G)
- Word unigrams (W1G)
- Word 3-grams (W3G)
- Part-of-speech 3-grams (POS3G)

4.1.3 Σώμα Κειμένων Εκπαίδευσης

Έστω D , το σύνολο των κειμένων d_i που θα χρησιμοποιήσουμε στην εκπαίδευση. Τα κείμενα έχουν την μορφή plain text.

Έστω Y , το σύνολο των κλάσεων y_j που έχουμε.

Έστω D_Y , ένα σύνολο που αντιστοιχεί κάθε κείμενο $d_i \in D$ στην κλάση $y_j \in Y$ στην οποία ανήκει. Αυτή η πληροφορία μας δίνεται έτοιμη απο το corpus.

4.1.4 Μετατροπή απλού σε πιθανοτικό classifier

Ένας απλός SVM classifier, επιστρέφει 1 αν η είσοδος ανήκει στην κλάση και 0 αν δεν ανήκει. Αυτό, όμως, είναι ένα πολύ απλοϊκό μέτρο και δεν έχει τα ιδανικά αποτελέσματα σε οποιαδήποτε εφαρμογή του. Σε μερικές περιπτώσεις χρειαζόμαστε περισσότερη πληροφορία από τον ταξινομητή από μια απλή έξοδο 0 ή 1. Υπάρχουν εργαλεία, που περιγράφονται στο (Chang, et al., 2011) που χρησιμοποιούν έναν απλό SVM για να επιστρέψουν την πιθανότητα να ανήκει το κείμενο σε αυτήν την κλάση. Πολύ αφαιρετικά, επιστρέφει την πιθανότητα με βάση την απόσταση της αναπαράστασης απο το decision hyperplane του classifier.

4.2 Λογική Αλγόριθμου

Εδώ περιγράφονται τα βήματα του αλγορίθμου *cbsSVM* και παρατίθενται τα αντίστοιχα αποσπάσματα από τον κώδικα.

1. Μεταφέρουμε τα κείμενα του D , που είναι σε plain text, στο document space. Γίνεται, δηλαδή, μετατροπή των κειμένων σε διανύσματα στον χώρο document space. Με τον συγκεκριμένο αλγόριθμο έχουμε ελευθερία επιλογής, καθώς έχει εφαρμογή σε διαφορετικές διανυσματικές αναπαραστάσεις. Επιπλέον, έχει την δυνατότητα να χρησιμοποιηθούν πολλές διαφορετικές αναπαραστάσεις ταυτόχρονα. Συγκεκριμένα, στην παρούσα, ως token παίρνουμε τις διανυσματικές αναπαραστάσεις με C4G, W1G, W3G και POS3G. Για κάθε τύπο token, κάνουμε τα επόμενα δύο βήματα.

```
class document:
    def __init__(self,directory,genre):
        self.genre=genre
        self.vectors=[]
        self.document=open(directory,'r',encoding='iso-8859-15').read()
        self.document_pos=["".join(x) for x in nltk.pos_tag(nltk.word_tokenize(self.document))]
        self.document_pos=" ".join(self.document_pos)
```

- Έχοντας δεδομένα τα κείμενα που ανήκουν και δεν ανήκουν στην κλάση y_j , είναι σημαντικό να συμπεριλάβουμε πληροφορία και από τα κείμενα που δεν ανήκουν στην κλάση, υπολογίζουμε το κέντρο της κλάσης (centroid) όπως αναφέρεται στο [4.3 Εύρεση Κέντρων](#). Δηλαδή παίρνουμε μια διανυσματική αναπαράσταση που έχει πληροφορία για το που βρίσκεται το κέντρο της κλάσης. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να έχουμε τα κέντρα όλων των κλάσεων. Όμως, όπως ήδη αναφέρθηκε, το κάθε κείμενο έχει τέσσερις διαφορετικές αναπαραστάσεις, οπότε η διαδικασία επαναλαμβάνεται για να βρεθούν τα κέντρα των κλάσεων και για τους υπόλοιπους τύπους token. Στο τέλος, λοιπόν, έχουμε τέσσερα κέντρα για την κάθε κλάση.

```
def calc_centroids(self,documents_len):
    self.centroids=[]
    alpha=16
    beta=4
    for i in range(genre.features):
        self.centroids.append([alpha*self.documents_sum[i]/len(self.documents)
                                -beta*self.others_sum[i]/(documents_len-len(self.documents))])
```

- Έχοντας τις διανυσματικές αναπαραστάσεις των κειμένων και τα κέντρα των κλάσεων, φτιάχνουμε ένα καινούργιο διάνυσμα για κάθε κείμενο που εκφράζει το πόσο μοιάζει με το κέντρο της αντίστοιχης κλάσης. Από εδώ έχουμε και το όνομα της μεθόδου, Center Based Similarity. Μπορούμε να υιοθετήσουμε πολλές μετρικές ομοιότητας για τα δύο διανύσματα. Στο (Cha, 2007) περιέχεται μια πληθώρα μετρικών, από τις οποίες μπορούμε να διαλέξουμε. Χρησιμοποιώντας πέντε μετρικές ομοιότητας (Fei, et al., 2015), όπως αναφέρονται αναλυτικά στο [4.4 Μετρικές Ομοιότητας](#), έχουμε μεταφέρει τα κείμενά μας στο similarity space. Επειδή έχουμε πολλούς τύπους token, συνδυάζουμε τα διαφορετικά

διανύσματα σε ένα και προκύπτει ένα 20-διάστατο διάνυσμα στο αντίστοιχο similarity space.

```
def calculate_similarity_features(self, centroids, document):
    #Ta document representations
    documents=[document.vectors[i] for i in range(genre.features)]
    #Esoterika ginomena centroid/document representation
    dot_products=[centroids[i].dot(documents[i].transpose()) for i in range(len(centroids))]
    #Metro centroids
    centroids_magn=[norm(centroids[i]) for i in range(len(centroids))]
    #Metro document representations
    document_magn=[norm(documents[i]) for i in range(len(centroids))]
    similarity_features=[]
    for i in range(len(centroids)):
        similarity_features.append(cbsSVC.sim_cos(dot_products[i],centroids_magn[i],document_magn[i]).toarray()[0][0])
        similarity_features.append(cbsSVC.sim_gow(centroids[i],documents[i],centroids_magn[i],document_magn[i]))
        similarity_features.append(cbsSVC.sim_lor(centroids[i],documents[i]))
        similarity_features.append(cbsSVC.sim_dice(dot_products[i],centroids_magn[i],document_magn[i]).toarray()[0][0])
        similarity_features.append(cbsSVC.sim_jac(dot_products[i],centroids_magn[i],document_magn[i]).toarray()[0][0])
    return similarity_features
```

4. Εκπαιδεύουμε έναν binary classifier να μάθει το hyperplane που χωρίζει τις αναπαραστάσεις των κειμένων που ανήκουν στην κλάση, από αυτές που δεν ανήκουν.

```
#Ekpaideysi aploy SVC me thetika auta poy anikoun kai arnitika ola ta ypoloipa
c=SVC(kernel="linear",C=100,probability=True,cache_size=8000,class_weight="balanced")
labels=["Positive" if l==g.genre else "Negative" for l in self.y_]
self.classifiers.append(c.fit(similarities,labels))
```

Στην συνέχεια, γίνεται η δοκιμή του μοντέλου δίνοντάς του καινούργια κείμενα ώστε να δούμε που τα κατατάσει. Αν ξέρουμε που ανήκουν, μπορούμε να μετρήσουμε την αποδοτικότητα του μοντέλου.

Η παραπάνω διαδικασία, περιγράφει την περίπτωση που έχουμε μία κλάση y_j . Για όλες τις κλάσεις στο Y , εκτελούμε τα παραπάνω βήματα, και συνδυάζουμε τους classifier, όπου ο καθένας θα είναι One-vs-Rest. Ένα κείμενο, ανήκει στην κλάση για την οποία επιστρέφει την μεγαλύτερη πιθανότητα.

5. Στην περίπτωση που θέλουμε να λάβουμε υπόψη το ενδεχόμενο ένα κείμενο να μην ανήκει σε καμία από τις γνωστές κλάσεις (open world), βάζουμε ένα κάτω όριο (threshold) στο πόσο χαμηλή μπορεί να είναι η πιο υψηλή πιθανότητα. Επομένως, αν για ένα κείμενο δεν υπάρχει καμία πιθανότητα που να ξεπερνάει το threshold που έχουμε ορίσει, τότε θα

αναγνωρίζει ότι το κείμενο δεν ανήκει σε καμία από τις γνωστές κλάσεις, επομένως θα το αναγνωρίζει ως θόρυβο (noise).

```
#Synarthsh eyresis pio pithanou genre gia keimeno
def predict(self,X): ...

#Epistrofh analytikon pithanotiton
def predict_proba(self,X): ...
```

4.3 Εύρεση Κέντρων

Όπως έχει ήδη αναφερθεί, για να μπορέσουμε να ορίσουμε το Open Space πρέπει αρχικά να υπολογιστεί η ακτίνα r_o για κάθε γνωστή θετική κλάση κάνοντας χρήση του CBS Learning.

Αρχικά, για τον υπολογισμό των κέντρων (centroid) των κλάσεων θεωρείται ένα σύνολο D από τα παραδείγματα των κειμένων της εκπαίδευσης, όπως φαίνεται παρακάτω (Fei, et al., 2016) :

$$D = \{(d_1, y_1), (d_2, y_2), \dots, (d_n, y_n)\} \quad (4.3)$$

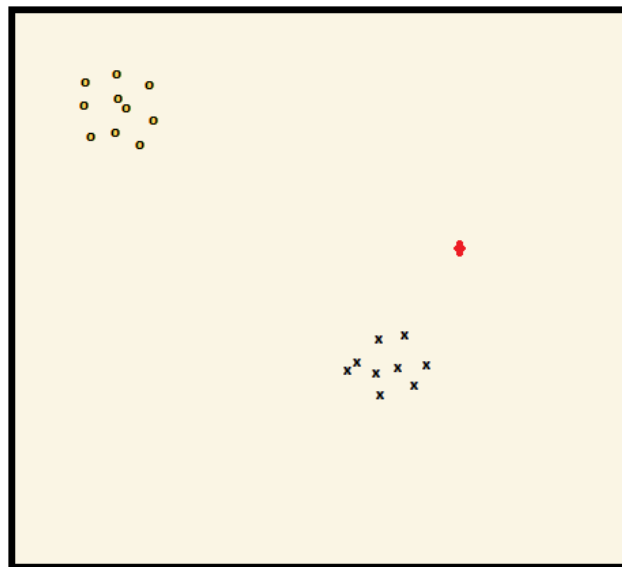
με d_i να είναι η διανυσματική αναπαράσταση, δημιουργημένη με το αντίστοιχο token, που αντιπροσωπεύει ένα κείμενο d_i και $y_i \in \{1, -1\}$ να είναι η πληροφορία για το αν ανήκει ή όχι σε μια κλάση. Το κάθε (d_i, y_i) αποτελεί το διάνυσμα του κειμένου i στο document space και ονομάζεται document space vector (εν συντομία ds-vector). Σε αντίθεση με παραδοσιακούς τρόπους ταξινόμησης στους οποίους δημιουργείται ο δυάδικος classifier χρησιμοποιώντας το σύνολο D , στο (Fei, et al., 2016) χρησιμοποιείται ο CBS Learning για να μεταφέρει το κάθε ds-vector d_i σε Center Based Similarity vectors (cbs- v_i). Κάθε στοιχείο του cbs- v_i προκύπτει από το συσχετισμό μεταξύ του κέντρου της κλάσης c_j και του d_i .

Χρησιμοποιώντας τον τύπο (4.4), υπολογίζουμε το κέντρο της κλάσης (Rocchio, 1971), (Manning, et al., 2008) :

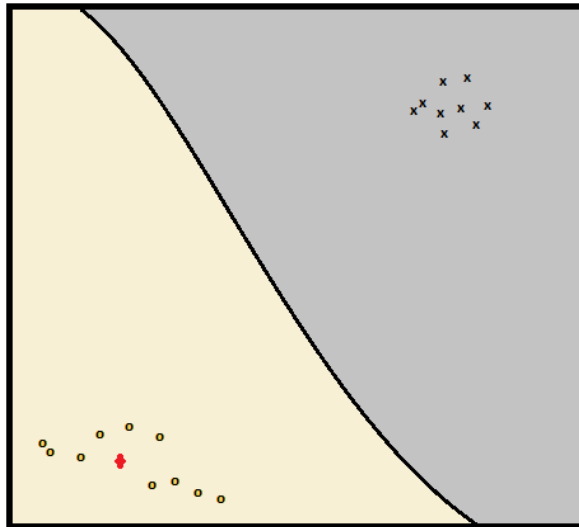
$$c_j = \frac{a}{|D_+|} \sum_{d_i \in D_+} \frac{x_j^i}{\|x_j^i\|} - \frac{\beta}{|D-D_+|} \sum_{d_i \in D-D_+} \frac{x_j^i}{\|x_j^i\|} \quad (4.4)$$

Παρατηρούμε ότι παίρνουμε υπόψη τόσο τα κείμενα που ανήκουν στην θετική κλάση ($d_i \in D_+$), όσο και τα αρνητικά κείμενα ($d_i \in D-D_+$). Οι σταθερές α και β εξομαλύνουν ή εντείνουν το βάρος των διανυσμάτων στον υπολογισμό και ορίζονται εμπειρικά. Θέτοντας $\alpha=16$ και $\beta=4$ παίρνουμε περισσότερη πληροφορία από τα θετικά κείμενα σε tf-idf αναπαραστάσεις (Buckley, et al., 1994).

Στο παράδειγμα που ακολουθεί, γίνεται μια προσπάθεια αναπαράστασης της παραπάνω θεωρίας, Σχήμα (4.1). Έχουμε τα διανύσματα διαφορετικών κειμένων που φαίνονται να σχηματίζουν κλάσεις σε έναν Document Space, με το κόκκινο σημάδι να αποτελεί ένα διάνυσμα ds ενός κειμένου του testing. Μετά από το μετασχηματισμό και τη μεταφορά των κειμένων από το Document Space στο Center Based Similarity Space έχουμε καλύτερη αναπαράσταση και κατανομή των κειμένων με ξεκάθαρη ταξινόμηση μεταξύ των κλάσεων, Σχήμα (4.2).



Σχήμα 4.1 : Αναπαράσταση διανύσματος κειμένου στο Document Space.



Σχήμα 4.2 : Αναπαράσταση διανύσματος κειμένου στο Center Based Similarity Space.

4.4 Μετρικές Ομοιότητας

Σημαντικό κομμάτι είναι η πληροφορία της ομοιότητας ανάμεσα στο κέντρο της κλάσης και στην διανυσματική αναπαράσταση του κειμένου. Στην παρούσα εργασία υιοθετήθηκαν τα ίδια μέτρα που χρησιμοποίησαν οι Geli Fei και Bing Liu στην εργασία τους για την μεταφορά των διανυσμάτων από το Document Space στο Similarity Space, Σχέσεις (4.5), (Fei, et al., 2015). Οι μετρικές εφαρμόζονται σε κάθε διανυσματική αναπαράσταση κειμένου στο document space και το αντίστοιχο κέντρο της κλάσης ώστε να μπορεί να εξαχθεί ένα συμπέρασμα για την ομοιότητά τους. Στην δική μας περίπτωση παράγουμε 20 CBS χαρακτηριστικά χρησιμοποιώντας C4G, W1G, W3G και POS3G για το κάθε κείμενο. Επομένως, έχουμε για κάθε κείμενο ένα 20-διάστατο διάνυσμα στο Similarity Space για κάθε κλάση του training. Στις Σχέσεις (4.5) τα P και Q είναι τα διανύσματα του κέντρου της κλάσης και του κειμένου στο document space και d είναι η διάσταση των P και Q.

$$\begin{aligned}
 S_{Cos} &= \frac{\sum_{i=1}^d P_i Q_i}{\sqrt{\sum_{i=1}^d P_i^2} \sqrt{\sum_{i=1}^d Q_i^2}} \\
 S_{gow} &= 1 - \frac{1}{d} \sum_{i=1}^d \left| \frac{P_i}{\sqrt{\sum_{i=1}^d P_i^2}} - \frac{Q_i}{\sqrt{\sum_{i=1}^d Q_i^2}} \right| \\
 S_{Lor} &= 1 - \sum_{i=1}^d \ln(1 + |P_i - Q_i|) \\
 S_{Dice} &= \frac{2 \sum_{i=1}^d P_i Q_i}{\sum_{i=1}^d P_i^2 + \sum_{i=1}^d Q_i^2} \\
 S_{jac} &= \frac{\sum_{i=1}^d P_i Q_i}{\sum_{i=1}^d P_i^2 + \sum_{i=1}^d Q_i^2 - \sum_{i=1}^d P_i Q_i} \quad (4.5)
 \end{aligned}$$

Μετά την δημιουργία των διανυσμάτων στο Similarity Space και σε συνδυασμό με τα κέντρα των κλάσεων εξάγεται το hyperplane για την κάθε κλάση, όπως φαίνεται και στο σχήμα της προηγούμενης ενότητας, Σχήμα (4.2).

4.5 Ψευδοκώδικας

Ακολουθεί η περιγραφή του αλγορίθμου *cbsSVM Classifier* με την μορφή ψευδοκώδικα.

Αλγόριθμος: cbsSVM

Δεδομένα: G ένα σύνολο genre και w ένα μέλος του Wg που είναι ένα σύνολο από κείμενα ιστοσελίδων για κάθε genre $g \in G$, d το κείμενο μιας τυχαίας ιστοσελίδας, σ το όριο πιθανότητας, Tf ένα σύνολο από

διαμορφώσεις token

Αποτέλεσμα: $r \in \{G, \emptyset\}$

for each $f \in Tf$ **do:**

Μετατρέπουμε τα κείμενα στις διανυσματικές αναπαραστάσεις τους

end

for each $g \in G$ **do:**

for each $f \in Tf$ **do:**

Με βάση το αν ανήκει ή όχι το κάθε κείμενο στο genre, υπολογίζουμε το centroid του genre

end

end

for each $g \in G$ **do:**

for each $f \in Tf$ **do:**

for each $w \in Wg$ **do:**

Χρησιμοποιούμε τους 5 τύπους ομοιότητας για να φτιάξουμε το διάνυσμα ομοιότητας ανάμεσα στο w και το centroid του g με f διαμόρφωση

end

end

Ενώνουμε τα παραγμένα διανύσματα σε ένα μεγαλύτερο. Αποτελεί την διανυσματική αναπαράσταση του κειμένου στο cbs similarity space του g

end

for each $g \in G$ **do:**

Εκπαιδεύουμε ένα SVM classifier που χωρίζει τα κείμενα που ανήκουν στο g από αυτά που δεν ανήκουν.

end

for each $g \in G$ **do:**

Βρίσκουμε την πιθανότητα το d να ανήκει στο g , $\text{score}[g]$

end

$\text{maxw} = \max(\text{score})$ Βρίσκουμε την μέγιστη πιθανότητα για το κείμενο d

if $\text{maxw} > \sigma$ **then**

$r = g$, το w ανήκει στο αντίστοιχο genre

else

$r = \emptyset$, το w δεν ανήκει ανήκει σε κανένα genre

end

5

Περιβάλλον Πειραμάτων

Σε αυτό το κεφάλαιο γίνεται μια περιγραφή των πειραμάτων που υλοποιήθηκαν χρησιμοποιώντας τον cbsSVM Classifier, συγκεκριμένα περιγράφονται τα datasets που χρησιμοποιήθηκαν, στην συνέχεια περιγράφονται τα πειράματα που υλοποιούνται σε αυτή την εργασία και τέλος γίνεται μια σύγκριση αυτών με άλλους αλγόριθμους.

5.1 Corpus

Το corpus είναι μια δομημένη συλλογή κειμένου μεγάλου μεγέθους, προφορικής γλώσσας ή άλλων γλωσσικών δεδομένων που χρησιμοποιείται ως πηγή για γλωσσική ανάλυση και έρευνα. Τα σώματα (corpus) χρησιμοποιούνται σε διάφορους τομείς της γλωσσολογίας, της επεξεργασίας φυσικής γλώσσας (NLP) και της υπολογιστικής γλωσσολογίας για τη μελέτη γλωσσικών προτύπων. Είναι απαραίτητα εργαλεία για την κατανόηση και τη μοντελοποίηση της γλώσσας και χρησιμοποιούνται για ένα ευρύ φάσμα γλωσσικών μελετών και εφαρμογών.

Τα Corpora (πληθυντικός Corpus) μπορούν να λάβουν διάφορες μορφές, όπως το Text Corpora. Αυτά περιέχουν γραπτό κείμενο, όπως βιβλία, άρθρα, ιστότοπους και άλλο γραπτό υλικό. Τα σώματα κειμένων χρησιμοποιούνται συνήθως για εργασίες όπως μοντελοποίηση γλώσσας, ανάλυση κειμένου και ανάκτηση πληροφοριών.

Τα Corpora εξυπηρετούν διάφορους σκοπούς στη γλωσσική έρευνα και σε εφαρμογές που σχετίζονται με τη γλώσσα, όπως η Γλωσσική Ανάλυση, η Μοντελοποίηση Γλώσσας, η Μηχανική μάθηση και τα NLP. Μέσω των Corpora, γίνονται έρευνες για την ανάλυση γλωσσικών μοτίβων, συχνότητες λέξεων, γραμματικές δομές και συντακτικά φαινόμενα, τα οποία μπορούν να οδηγήσουν σε γνώσεις σχετικά με τη γλωσσική παραλλαγή και χρήση.

Επίσης, τα Corpora είναι ζωτικής σημασίας για τη δημιουργία στατιστικών μοντέλων γλώσσας, τα οποία είναι απαραίτητα για διάφορες εργασίες επεξεργασίας φυσικής γλώσσας NLP, συμπεριλαμβανομένης της αυτόματης μετάφρασης, της αναγνώρισης ομιλίας και της δημιουργίας κειμένου.

Ο βασικός, όμως, σκοπός που ερευνά η παρούσα εργασία για τα Corpora είναι αυτός της Μηχανικής μάθησης και των NLP. Τα Corpora χρησιμοποιούνται για την εκπαίδευση και την αξιολόγηση μοντέλων μηχανικής μάθησης και αλγορίθμων επεξεργασίας φυσικής γλώσσας, διευκολύνοντας εργασίες όπως η αναγνώριση ονομαστικών οντοτήτων/κλάσεων και η ταξινόμηση κειμένου.

Στη συνέχεια αυτού του κεφαλαίου γίνεται αναφορά στα δυο Corpora που χρησιμοποιήθηκαν για την υλοποίηση των πειραμάτων, το KI-04 (Eissen, et al., 2004) και το SANTINI (Mehler, et al., 2010), τα οποία είναι τα ίδια που χρησιμοποιήθηκαν στο (Pritsos, et al., 2019) αλλά και σε προηγούμενες εργασίες ταξινόμησης σε Web Genre Identification (Eissen, et al., 2004) (Santini, 2007) (Kanaris, et al., 2009). Και τα δύο σώματα, αποτελούνται από κείμενα που προέρχονται από σελίδες ιστότοπων, από τους οποίους έχει αφαιρεθεί εντελώς όλη πληροφορία που δεν αποτελεί κείμενο (π.χ μορφολογική πληροφορία σε html, εικόνες, σύνδεσμοι κ.α).

Η επιλογή των συγκεκριμένων Corpora γίνεται λόγω της διαφορετικότητά τους στην δομή. Στο KI-04 τα δεδομένα περιέχουν δομημένο θόρυβο (structured noise).

Ενώ το SANTINIS περιέχει αδόμητο θόρυβο (unstructured noise) αφού δεν υπάρχει διαφοροποίηση μεταξύ των άγνωστων κλάσεων, δηλαδή όλες οι άγνωστες κλάσεις περιλαμβάνονται σε μια μοναδική υπερκλάση. Αυτή η δομή των δεδομένων είναι πιο ρεαλιστική αφού σε πραγματικά σενάρια ταξινόμησης κειμένων από το διαδίκτυο, δεν είναι δυνατό να προκαθοριστεί το πλήθος των κλάσεων.

5.1.1 KI-04

Αυτό το corpus αποτελείται από 8 κλάσεις κειμένων (Article, Discussion, Download, Help, Link Collection, Portrayal-Non Private, Portrayal-Private και Shop) (Pritsos, et al., 2019). Οι κλάσεις (genre) δεν περιέχουν το ίδιο πλήθος σελίδων αλλά είναι αρκετά κοντά, με ελάχιστες σελίδες τις 127 και μέγιστες τις 208. Η δομή του σώματος αυτού το καθιστά χρήσιμο για χρήση σαν δομημένος θόρυβος, καθώς κάθε κείμενο ανήκει σε μία κλάση. Στον Πίνακα (5.1) φαίνεται ένας λεπτομερής κατάλογος με την κατανομή των κειμένων ανά κλάση (genre).

KI-04	
Genre	Pages
Article	127
Discussion	127
Download	152
Help	140
Link Collection	208
Portrayal-Non Private	179
Portrayal- Private	131
Shop	175

Πίνακας 5.1: Περιγραφή Corpora KI-04 και πλήθος σελίδων ανα κλάση (genre) (Pritsos, et al., 2019).

5.1.2 Santini

Το Santini corpus, αποτελείται από 11 κλάσεις κειμένων (Pritsos, et al., 2019). Σε αντίθεση με το KI-04, έχουμε την παρουσία της κλάσης Noise, που είναι κείμενα που έχουν παρθεί τυχαία από ένα άλλο σώμα κειμένων, το Spirit. Στην πραγματικότητα, τα κείμενα του Spirit-1000 δεν είναι ταξινομημένα και έχουν παρθεί τυχαία. Αυτό το καθιστά καλό υποψήφιο για πειράματα με unstructured noise, καθώς η κατανομή των κειμένων είναι τυχαία. Κατά την διάρκεια του testing, βέβαια, μπορεί τελικά να βρίσκαμε ότι κάποια από τα κείμενα άνηκαν σε μία από τις υπόλοιπες 10 κλάσεις. Αυτό όμως δεν θα μας απασχολήσει και ακόμα και αν θεωρηθεί από το μοντέλο ότι ανήκει σε κάποια από τις κλάσεις, εμείς πρέπει να το μετρήσουμε σαν false positive. Αναλυτικότερα, οι επτά κλάσεις (Blog, Eshop, FAQ, Frontpage, Listing, Personal Home Page και Search Page) έχουν το ίδιο πλήθος σελίδων με την κάθε μια να αποτελείται από 200 σελίδες, ενώ οι επόμενες τέσσερις κλάσεις (DIY Mini Guide (BBC), Editorial(BBC), Features (BBC) και Short Bio (BBC)) αποτελούνται από 20 σελίδες αντιστοίχως. Τέλος, η κλάση Noise (Spirit1000) έχει το μεγαλύτερο πλήθος σελίδων, με αυτές να είναι 1000. Υπάρχει αισθητή διαφορά στο πλήθος των σελίδων, ειδικά στην κλάση Noise.

SANTINIS	
Genre	Pages
Blog	200
Eshop	200
FAQ	200
Frontpage	200
Listing	200
Personal Home Page	200
Search Page	200
DIY Mini Guide (BBC)	20
Editorial (BBC)	20
Features (BBC)	20
Short Bio (BBC)	20
Noise (Spirit1000)	1000

Πίνακας 5.2: Περιγραφή Corpora SANTINI και πλήθος σελίδων ανα κλάση (genre) (Pritsos, et al., 2019).

5.2 Μετρικές

5.2.1 Μετρικές Αξιολόγησης Μοντέλου

Η ακρίβεια/precision, η ανάκληση/recall και το F-score είναι κρίσιμες μετρήσεις απόδοσης για την αξιολόγηση των ταξινομητών. Το Precision υπολογίζει την αναλογία των αληθινών θετικών προβλέψεων προς τις συνολικές θετικές προβλέψεις, υπογραμμίζοντας την ικανότητα του ταξινομητή να αποφεύγει τα ψευδώς θετικά (false positive). Το Recall, από την άλλη πλευρά, αξιολογεί την ικανότητα του ταξινομητή να προσδιορίζει όλες τις πραγματικές θετικές περιπτώσεις. Το F-score, γνωστό και ως εντολή F1, είναι ένας συνδυασμός ακρίβειας και ανάκλησης, παρέχοντας μια ισορροπημένη αξιολόγηση της απόδοσης ενός ταξινομητή.

5.2.1.1 Precision

Το precision εκφράζει την αποτελεσματικότητα του μοντέλου σχετικά με τα αποτελέσματα που γυρνάει ως θετικά. Υπολογίζεται με βάσει τα True Positive και False Positive των αποτελεσμάτων που γυρνάει ως θετικά.

$$P = \frac{TP}{TP+FP} \quad (5.1)$$

5.2.1.2 Recall

Αντίστοιχα, το Recall δείχνει πόσο καλά αναγνωρίζει το μοντέλο τα δείγματα που είναι όντως θετικά. Αρα, χρησιμοποιεί τα True Positive και False Negative στον υπολογισμό:

$$R = \frac{TP}{TP+FN} \quad (5.2)$$

5.2.1.3 *F1-Score*

Το F_{β} -Score (ή αλλιώς F -Measure) είναι ο σταθμισμένος μέσος όρος ανάμεσα στην ακρίβεια (precision) και στην ανάκληση (recall). Υπολογίζεται με την ακόλουθη σχέση (Manning, et al., 2008) :

$$F_{\beta} = (1 + \beta^2) \frac{P \cdot R}{\beta^2 P + R} \quad (5.3)$$

Ο όρος β αλλάζει ανάλογα με το πιο μέτρο μας απασχολεί περισσότερο, το precision ή το recall. Όταν $\beta > 1$ τότε δίνεται μεγαλύτερη έμφαση στο precision, ενώ όταν $0 < \beta < 1$ η βαρύτητα δίνεται στο recall. Όταν είναι $\beta=1$ τότε δίνουμε ίση αξία και στα δύο και η Σχέση (5.3) γίνεται :

$$F_1 = \frac{2PR}{P+R} \quad (5.4)$$

Το F_1 -Score είναι ο ισορροπημένος μέσος όρος του precision και του recall. Όπως το precision και το recall, έτσι και το F_{β} -Score παίρνει τιμές από 0 έως 1.

5.2.1.4 *Macro Παραλλαγές*

Οι παραπάνω μετρικές, γίνονται για κάθε κλάση ξεχωριστά. Τι γίνεται όμως, όταν θέλουμε να εκφράσουμε την απόδοση ενός αλγορίθμου για όλες τις κλάσεις; Εδώ, χρησιμοποιούμε την macro παραλλαγή των μετρικών.

Η macro παραλλαγή μιας μετρικής, υπολογίζεται παίρνοντας την μέση τιμή των μετρικών για όλες τις κλάσεις.

5.2.2 *Παραλλαγές Μετρικών για Αξιολόγηση Μοντέλων Ανοικτού Συνόλου*

Κατα την αξιολόγηση μοντέλων ταξινόμησης ανοικτού συνόλου, γίνεται απαραίτητο να ορίσουμε καινούργιες μετρικές, καθώς οι παραδοσιακές δεν αρκούν.

Για να μετατρέψουμε τις παραπάνω μετρικές στις αντίστοιχες ανοικτού συνόλου, παίρνουμε τις μετρικές για τις κλάσεις που χρησιμοποιήσαμε στην εκπαίδευση και βρίσκουμε τον μέσο όρο.

5.2.2.1 Open Set F1 Macro Score

Στο πλαίσιο της μηχανικής μάθησης και της ταξινόμησης κειμένων, το F1 Macro Score συνδυάζει έννοιες από την αναγνώριση ανοικτού συνόλου και το F1 Score, μια κοινή μέτρηση που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου ταξινόμησης.

Όπως έχει γίνει αναφορά και σε προηγούμενο κεφάλαιο σε αντίθεση με την αναγνώριση κλειστών συνόλων, όπου όλες οι κατηγορίες είναι γνωστές κατά τη διάρκεια της εκπαίδευσης, η αναγνώριση ανοικτών συνόλων αναγνωρίζει ότι ο πραγματικός κόσμος περιέχει κλάσεις που δεν φαίνονται κατά τη διάρκεια της εκπαίδευσης. Επομένως, ένας ταξινομητής ανοικτού συνόλου δεν πρέπει μόνο να ταξινομεί σωστά τις προβλεπόμενες κλάσεις αλλά και να απορρίπτει σωστά τις λανθασμένες.

Το F1 Score, είναι ένα μέτρο της ακρίβειας μιας δοκιμής και λαμβάνει υπόψη τόσο την ακρίβεια όσο και την ανάκληση της δοκιμής. Η ακρίβεια είναι ο αριθμός των σωστών θετικών αποτελεσμάτων διαιρεμένος με τον αριθμό όλων των θετικών αποτελεσμάτων, συμπεριλαμβανομένων εκείνων που δεν προσδιορίστηκαν σωστά, και η ανάκληση είναι ο αριθμός των σωστών θετικών αποτελεσμάτων διαιρεμένος με τον αριθμό των θετικών αποτελεσμάτων που θα έπρεπε να είχαν εντοπιστεί. Το F1 Score είναι το αρμονικό μέσο ακρίβειας και ανάκλησης.

Κατά την αξιολόγηση προβλημάτων ταξινόμησης πολλαπλών κλάσεων, ο μέσος όρος μακροεντολών θα υπολογίσει τη μέτρηση ανεξάρτητα για κάθε τάξη και στη συνέχεια θα λάβει τον μέσο όρο (επομένως αντιμετωπίζοντας όλες τις κατηγορίες εξίσου). Αυτό είναι ιδιαίτερα

σημαντικό σε σύνολα δεδομένων όπου οι ανισορροπίες κλάσεων θα μπορούσαν διαφορετικά να παραμορφώσουν τη συνολική μέτρηση.

Έτσι, το F1 Macro Score ανοικτού συνόλου είναι μια μέτρηση που αξιολογεί τον αρμονικό μέσο όρο ακρίβειας και ανάκλησης για ένα πρόβλημα ταξινόμησης ανοικτών συνόλων, πράττοντάς το με τρόπο που αντιμετωπίζει όλες τις κλάσεις εξίσου. Είναι ένας μακρο-μέσος όρος βαθμολογίας F1 που λαμβάνει υπόψη την πρόσθετη πρόκληση της σωστής απόρριψης δειγμάτων από τάξεις που δεν εμφανίστηκαν κατά τη διάρκεια της προπόνησης.

Για να υπολογιστεί, το F1 Macro Score πρέπει πρώτα να υπολογιστεί το F1 Score για κάθε τάξη ανεξάρτητα, εκτός της κλάσης «απόρριψης» για όλες τις άγνωστες κατηγορίες. Στη συνέχεια, θα πρέπει να υπολογιστεί ο μέσος όρος αυτών των βαθμολογιών.

Αυτή η μέτρηση δεν είναι τυπική όπως η κανονική F1 Score, επομένως η χρήση και ο ορισμός της μπορεί να είναι συγκεκριμένα για ένα συγκεκριμένο έγγραφο ή ερευνητικό πλαίσιο. Εάν προέρχεται από μια συγκεκριμένη μελέτη ή τομέα, θα πρέπει να παρέχουν ακριβή ορισμό και μεθοδολογία για τη μέτρηση αξιολόγησής τους.

5.2.3 Openness-F1-Score Curve

Καθώς είδαμε, ο αλγόριθμος μπορεί να χρησιμοποιηθεί για την ανάπτυξη μοντέλων ταξινόμησης ανοικτού συνόλου. Για να μετρήσουμε την απόδοση του μοντέλου, πρέπει να δούμε πως αντιδρά σε διαφορετικές συνθήκες θορύβου. Εδώ, βρίσκουμε την απόδοση όταν ο θόρυβος είναι δομημένος. Δηλαδή, όταν όλα τα κείμενα θορύβου ανήκουν σε κλάση. Μας ενδιαφέρει η απόδοση, σε σχέση με τα κείμενα εκπαίδευσης και δοκιμής. Για αυτήν την εργασία, χρησιμοποιούμε το KI-04 corpus.

5.2.3.1 Openness

Η δυσκολία των open-set ταξινομήσεων υποδηλώνεται από την μετρική του openness λαμβάνοντας υπόψη το πλήθος των κλάσεων της εκπαίδευσης και το πλήθος των κλάσεων του testing (Mendes, 2016). Το πόσες κλάσεις χρησιμοποιήθηκαν στην εκπαίδευση σε σχέση με την δοκιμή εκφράζεται με το openness. Υπολογίζεται από την ακόλουθη σχέση (Scheirer, 2013):

$$openness = 1 - \sqrt{\frac{|TrainingClasses|}{|TestingClasses|}} \quad (5.5)$$

Όταν το openness είναι 0 πρόκειται για ταξινόμηση κλειστού συνόλου αφού οι γνωστές κλάσεις είναι όσες και οι κλάσεις του testing και κατά επέκταση σε αυτή την περίπτωση θεωρείται ότι ο θόρυβος είναι μηδενικός. Από την άλλη, στην περίπτωση όπου το openness είναι κοντά στο 1 έχουμε ότι το πλήθος των κλάσεων του testing είναι πολύ μεγαλύτερο από το πλήθος των κλάσεων της εκπαίδευσης. Σε αυτή την περίπτωση, ο θόρυβος είναι πολύ μεγάλος και παρουσιάζει ετερογένεια.

Σε αντίθεση με άλλες μετρικές όπως το precision, recall και f1-score τα οποία μπορούν να εφαρμοστούν σε structured και unstructured δεδομένα, το openness έχει εφαρμογή μόνο σε structured δεδομένα αφού είναι απαραίτητο για τον υπολογισμό του το πλήθος των κλάσεων στο testing.

Μας ενδιαφέρει να παρατηρήσουμε την απόδοση του αλγορίθμου μεταβάλλοντας την τιμή του openness. Στην διατριβή (Pritsos, et al., 2019), τα δεδομένα με θ γνωστές κλάσεις χωρίζονται σε δύο μέρη, το πρώτο $K \subset \theta$ για τις γνωστές κλάσεις και το δεύτερο $S \subset \varphi$ ($K \cup U = \theta$) για τον θόρυβο. Έτσι, οι κλάσεις της εκπαίδευσης παίρνουν τιμές από 1 έως $|\theta| - 1$, ενώ οι κλάσεις του testing είναι πάντα $|\theta|$. Για τον αλγόριθμο της συγκεκριμένης εργασίας όμως, είναι απαραίτητο οι κλάσεις της εκπαίδευσης να ξεκινάνε από 2, επειδή χρειάζονται 2 κλάσεις τουλάχιστον για την εκπαίδευση και δημιουργία του hyperplane.

5.2.4 Precision-Recall Curve

Τα precision και recall είναι οι πιο βασικές μετρικές για την απόδοση ενός αλγορίθμου. Όμως, όπως είδαμε και παραπάνω, είναι ανταγωνιστικές, καθώς όσο αυξάνουμε το δείγμα που εξετάζουμε, έχουμε λιγότερα False Negative, άρα αυξάνεται το recall, αλλά αυξάνονται τα False Positive, και άρα πέφτει το precision. Ο τρόπος με τον οποίο γίνεται αυτή η ανταλλαγή τιμών, μας δείχνει την αποτελεσματικότητα του μοντέλου.

Για να κάνουμε την μέτρηση, παίρνουμε τα αποτελέσματα των δοκιμών, τα χωρίζουμε ανα κλάση, και τα κατατάσσουμε με βάση την βεβαιότητα (δηλαδή την πιθανότητα). Ξεκινώντας από τα αποτελέσματα με μεγάλη σιγουριά, υπολογίζουμε το precision, το οποίο περιμένουμε να είναι υψηλό στην αρχή. Αντίθετα, περιμένουμε το recall να είναι χαμηλό, καθώς έχουμε μετρήσει μόνο ένα δείγμα, και τα υπόλοιπα false negative είναι πιο μετά. Εξετάζουμε το επόμενο αποτέλεσμα, και επανυπολογίζουμε τα precision και recall. Καθώς προχωράμε, περιμένουμε να αυξάνεται το recall, καθώς τα false negative θεωρούνται true positive, ενώ αντίθετα, το precision να πέφτει όσο αυξάνονται τα false positive.

Καθώς αυτή η διαδικασία μπορεί να είναι επίπονη, πολύ συχνά δεν μετράμε δείγμα-δείγμα, αλλά τμήμα-τμήμα, και βγάζουμε τόσα αποτελέσματα όσα τμήματα.

Στην δική μας περίπτωση, θέλουμε να υπολογίζουμε το open set macro PRC. Για να το κάνουμε αυτό, υπολογίζουμε το Precision-Recall Curve (PRC) για κάθε κλάση που χρησιμοποιήθηκε στην εκπαίδευση και παίρνουμε τον μέσο όρο.

5.3 Υλοποίηση Πειραμάτων

5.3.1 Επαναλήψεις, Train-Test-Fold e.t.c

Σκοπός είναι για τα πειράματα με αδόμητο θόρυβο, κάνοντας χρήση του Santinis Corpus, να γίνουν με 10 folds Cross Validation (CV) παίρνοντας όλο τον θόρυβο, δηλαδή το genre Noise, σε κάθε fold. Το genre Noise είναι σημαντικά μεγαλύτερο από τα υπόλοιπα genres αλλά αυτό το

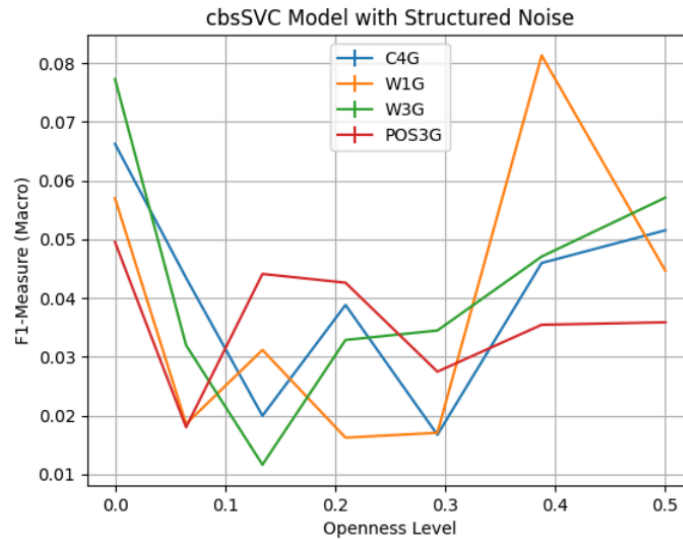
καθιστά ένα πιο ρεαλιστικό σενάριο της πραγματικότητας. Στην παρούσα, όμως, λόγω περιορισμών στο υλικό και στον χρόνο εκτέλεσης, τα πειράματα εκτελέστηκαν μόνο για ένα fold και τα αποτελέσματα που παρουσιάζονται στην συνέχεια είναι χωρίς τις εκτελέσεις των υπόλοιπων fold.

Ενώ για τα πειράματα με δομημένο θόρυβο, χρησιμοποιώντας το KI-04 corpus, η εκπαίδευση ξεκινάει με openness από 2 και φτάνει έως 8 κλάσεις. Οι δοκιμές γίνονται πάντα με τις υπόλοιπες κλάσεις κάνοντας 8 φορές (με τυχαίες κλάσεις) 10 fold Cross Validation. Όπως και για τα πειράματα με unstructured noise, έτσι και εδώ λόγω περιορισμών στο υλικό και στον χρόνο, τα πειράματα εκτελέστηκαν μόνο μια φορά για ένα fold. Τα αποτελέσματα αναφέρονται στην συνέχεια, όπως και ακολουθεί μια σύγκριση με τα αποτελέσματα του (Pritsos, et al., 2019).

5.3.2 Openness-F1-Score Curve

Σε αυτήν την υποενότητα παρουσιάζονται τα αποτελέσματα του F1-Measure του *cbsSVM* σε διαφορετικά επίπεδα του openness για το μοντέλο του cbsSVM Classifier με Structured Noise.

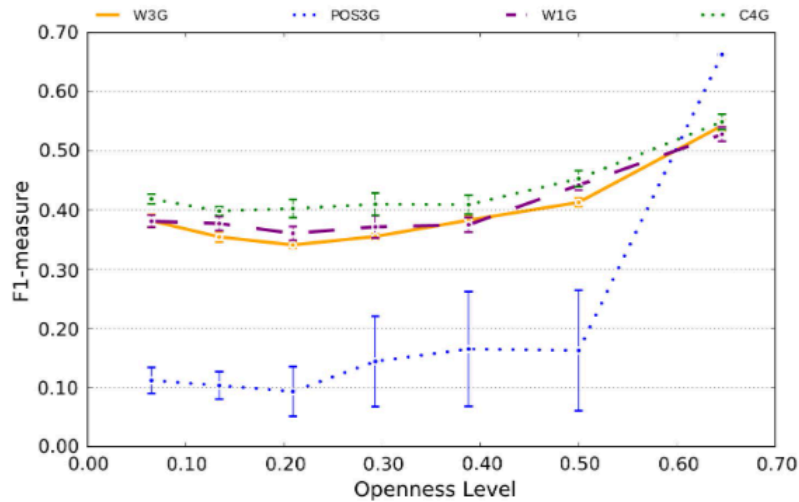
Στο Σχήμα (5.1), βλέπουμε την απόδοση του αλγορίθμου cbsSVM και τις τιμές που παίρνει το F1-Measure (Macro) καθώς μεταβάλλεται το openness από 0 έως 0.5, για κάθε τύπο token ξεχωριστά (C4G, W1G, W3G και POS3G). Το openness είναι 0 όταν στην εκπαίδευση γνωρίζουμε όλες τις κλάσεις που θα εμφανιστούν κατά την διάρκεια του testing, ενώ παίρνει την τιμή 0.5 όταν η εκπαίδευση γίνεται με μόνο 2 από τις 8 κλάσεις που υπάρχουν στις δοκιμές, όπως προκύπτει και από την Σχέση (5.5).



Σχήμα 5.1: Αποτελέσματα του *cbsSVM* Classifier για το KI-04 σε διαφορετικά επίπεδα openness

Γενικά, φαίνεται πως οι διανυσματικές αναπαραστάσεις αποδίδουν τιμές πολύ κοντά μεταξύ τους για κάθε τύπο token, με τα διανύσματα με W1G να διαφοροποιούνται δίνοντας υψηλότερη τιμή σε σχέση με τα υπόλοιπα όταν υπάρχει μεγάλο openness.

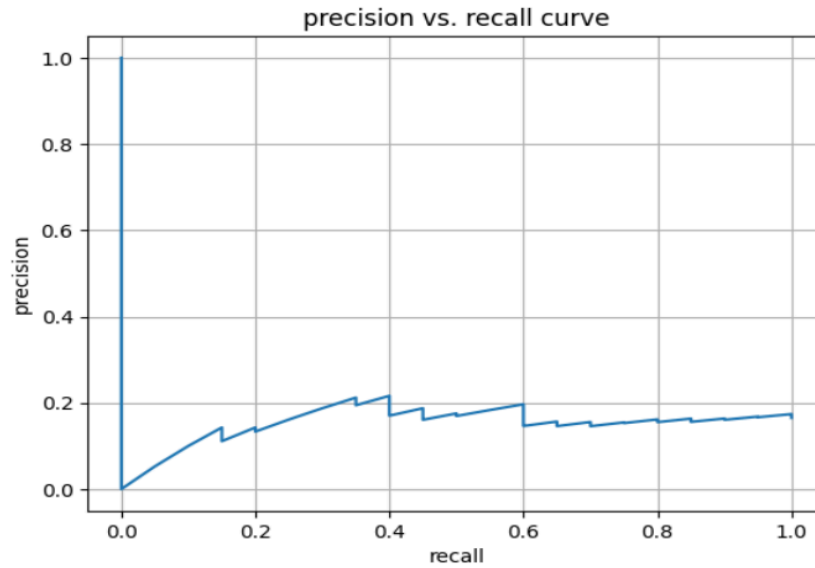
Συγκρίνοντας, όμως, αυτά τα αποτελέσματα με εκείνα του (Pritsos, et al., 2019) Σχήμα (5.2), στο οποίο τα πειράματα γίνονται με τον αλγόριθμο OCSVM και η απόδοσή του είναι αρκετά υψηλή και όταν το openness ξεπεράσει το 0.5 η απόδοσή του βελτιώνεται ακόμα περισσότερο, βλέπουμε ότι στην δική μας περίπτωση οι τιμές απόδοσης είναι σημαντικά υποπολλαπλάσιες και παρουσιάζονται πολύ ανεπαρκή αποτελέσματα.



Σχήμα 5.2 : Απόδοση του OCSVM σε διαφορετικά επίπεδα openness (Pritsos, et al., 2019)

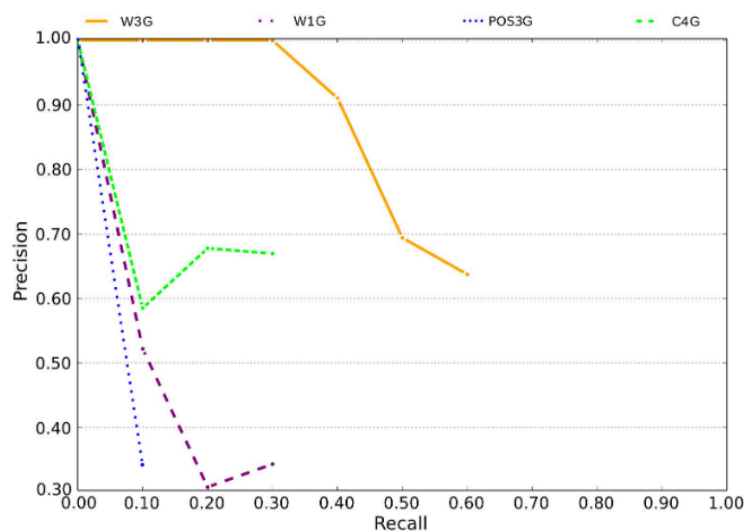
5.3.3 Precision-Recall Curve

Σε αυτήν την υποενότητα παρουσιάζονται τα αποτελέσματα του Precision ανάλογα το Recall για πειράματα με unstructured noise του cbsSVM Classifier, στο Corpus του SANTINIS. Τα αποτελέσματα δεν παρουσιάζονται ξεχωριστά για κάθε τύπο token όπως έγινε παραπάνω στα πειράματα με structured noise, αλλά παρουσιάζονται στο σύνολό τους. Δηλαδή, τα αποτελέσματα αφορούν στο σύνολο όλα τα features για τις περιπτώσεις W1G, W3G, C4G και των POS3G. Από το Σχήμα (5.3), συμπεραίνουμε πως όταν το recall έχει μηδενική τιμή τότε το precision παίρνει την μέγιστη τιμή, που είναι 1. Επίσης, μόλις το recall αποκτήσει κάποια τιμή παρατηρούμε μια απότομη πτώση στην τιμή του precision, που στην συνέχεια παραμένει σχετικά σταθερή στο 0.2 καθώς αυξάνεται το recall.



Σχήμα 5.3 : Precision-Recall Curve του *cbsSVM* για το SANTINIS corpus με W1G, W3G, C4G and POS3G features

Στην προσπάθεια να γίνει μια σύγκριση με τα αποτελέσματα στο (Pritsos, et al., 2019), όπου το precision παρουσιάζει μια πιο ομαλή πτώση στην τιμή του καθώς αυξάνεται το recall, στο Σχήμα (5.4) παρατηρείται μια πιο σταθερή ανταλλαγή ανάμεσα στις τιμές των precision και recall. Επιπλέον, φαίνεται πως στα πειράματα με τον OCSVM το recall δεν πλησιάζει την τιμή 1, ενώ ο *cbsSVM* καταφέρνει και την αγγίζει.



Σχήμα 5.4: Precision-Recall Curves of OCSVM models on SANTINIS corpus using W1G, W3G and C4G features (Pritsos, et al., 2019)

6 Συμπεράσματα

Αξιολογώντας τα αποτελέσματα των πειραμάτων που παρουσιάστηκαν στην προηγούμενη ενότητα καταλήγουμε πως ο αλγόριθμος δεν απέδωσε τα θεωρητικά αναμενόμενα αποτελέσματα. Η ανεπαρκής απόδοση των πειραμάτων έρχεται σε αντίθεση με την αρχική αναφορά και αποδόσεις του αλγορίθμου όπως περιγράφονται στο (Fei, et al., 2016). Αυτό το χαμηλό επίπεδο αποτελεσματικότητας πιθανώς οφείλεται στην ελλιπή υλοποίηση του αλγορίθμου λόγω υλικού και χρόνου εκτέλεσης, αλλά και στην δυσκολία εύρεσης των σωστών hyperparameters. Ως αποτέλεσμα αυτών, συνεπάγεται ότι υπάρχουν περιθώρια βελτίωσης του αλγορίθμου ώστε να αποδώσει καλύτερα αποτελέσματα στην ταξινόμηση των κειμένων στις σωστές κλάσεις.

7

Βιβλιογραφία

1. Mendes, P. R. J., Souza, R., Werneck, R., Stein, B., Pazinato, D., Almeida, W., Penatti, O., Torres, R., & Rocha, A. (2016). Nearest neighbors distance ratio open-set classifier. *Machine Learning*. 106. 1-28.
2. Aggarwal, C. C. , & Zhai, C. (2012). *Mining Text Data*.
3. Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology*, 60(3),538-556.
4. Weiss, S., Indurkha, N., Zhang, T., & Damerau, F. (2010). *Text mining: predictive methods for analyzing unstructured information*. Springer Science & Business Media.
5. Chang, C.-C. & Lin, C.-J. (2011). LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, Volume 2, Issue 3, Article No.: 27, pp 1-27
6. Cha, Sung-Hyuk. (2007). Comprehensive Survey on Distance/Similarity Measures Between Probability Density Functions. *Int. J. Math. Model. Meth. Appl. Sci.*. 1.
7. Sebastiani, F., (2001). *Machine Learning in Automated Text Categorization*. Final version published in *ACM Computing Surveys*, 34(1):1-47, 2002
8. Pritsos, D., & Stamatatos, E. (2019). Open-set Web Genre Identification.
9. Fei, G., & Liu, B. (2016). Linguistics Breaking the Closed World Assumption in Text Classification. In *Proceedings of the 2016 Conference of the North American Chapter of*

- the Association for Computational Linguistics: Human Language Technologies, 506–514, San Diego, California. Association for Computational Linguistics.
10. Scheirer, W., Rocha A., Sapkota A., & Boulton T. E. (2013). Towards open set recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp.1757 -1772.
 11. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*.
 12. Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*
 13. Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*.
 14. Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. ACM press
 15. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
 16. Quinlan, J. R., & Kaufmann, M. (1993). *C4.5: Programs for Machine Learning*. Inc., 1993. *Mach Learn* 16, 235–240 (1994). <https://doi.org/10.1007/BF00993309>.
 17. Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*
 18. Cohen, W.W. (1995). Fast Effective Rule Induction. *International Conference on Machine Learning*.
 19. Clark, P., & Niblett, T. (1989). The CN2 induction algorithm
 20. Cortes, C., & Vapnik, V. "Support-vector networks". *Machine learning*, 1995.
 21. Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing, Volume 1: Explorations in the Microstructure of Cognition: Foundations*.

22. Rish, I. (2001). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence, 3, page 41-46. IBM New York.
23. Fei, G. & Liu, B. (2015). Social Media Text Classification under Negative Covariate Shift. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2347-2356, Lisbon, Portugal. Association for Computational Linguistics.
24. Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (ed.), The Smart retrieval system - experiments in automatic document processing (pp. 313--323) . Englewood Cliffs, NJ: Prentice-Hall .
25. Mehler, A., & Sharoff, S., & Santini, M.,. (2010). Genres on the Web: Computational Models and Empirical Studies. 10.1007/978-90-481-9178-9.
26. Eissen, S., & Stein, B., (2004). Genre Classification of Web Pages... 256-269.
27. Jurafsky, D., & Martin, J. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.
28. Mintz, M., & Bills, S., & Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.
29. Ratinov, L., & Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pages 147–155, Boulder, Colorado. Association for Computational Linguistics.
30. LeCun, Y., & Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature. 521. 436-44. 10.1038/nature14539.