



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Συναισθηματική Ανάλυση Tweets με  
Χρήση Μηχανικής Μάθησης και  
Υλοποίηση Διαδικτυακής Εφαρμογής

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Γεώργιου Νικολάου Στάθη

Επιβλέπων: Ανδρέας Σταφυλοπάτης  
Καθηγητής ΕΜΠ

Αθήνα, Ιούνιος 2019





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

# Συναισθηματική Ανάλυση Tweets με Χρήση Μηχανικής Μάθησης και Υλοποίηση Διαδικτυακής Εφαρμογής

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Γεώργιου Νικολάου Στάθη

Επιβλέπων: Ανδρέας Σταφυλοπάτης  
Καθηγητής ΕΜΠ

Εγκρίθηκε από την κάτωθι τριμελή επιτροπή την 4<sup>η</sup> Ιουνίου 2019.

---

Ανδρέας Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

---

Γεώργιος Στάμου  
Αναπληρωτής Καθηγητής Ε.Μ.Π.

---

Κωνσταντίνα Νικήτα  
Καθηγήτρια Ε.Μ.Π.

---

**Γεώργιος Νικολάου Στάθης**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

Copyright © Γεώργιος Στάθης, 2019

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

*Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.*

*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

Στην σύγχρονη κοινωνία βιώνουμε μία ραγδαία ανάπτυξη των μέσων πληροφόρισης, η οποία επισφραγίστηκε από το Διαδίκτυο και πιο συγκεκριμένα τα Μέσα Κοινωνικής Διεύθυνσης με κύριο εκπρόσωπό τους, το **Twitter**. Η παρούσα Διπλωματική πραγματεύεται την ανάπτυξη Μοντέλων και Αρχιτεκτονικών **Βαθειάς** και **Μηχανικής Μάθησης** που στοχεύουν στην Συναισθηματική Ανάλυση σε σχόλια του Twitter, καθώς και στην ανάπτυξη μιας **Διαδικτυακής Εφαρμογής** η οποία πραγματοποιεί Ανάλυση Συναισθήματος σε τέτοιου είδους σχόλια, σε πραγματικό χρόνο. Η ανάλυση επικεντρώνεται κυρίως στα 4 βασικά Συναισθήματα: Θυμός, Φόβος, Χαρά, Λύπη.

Αρχικά παρουσιάζεται το **Task: Affect in Tweets** που ασχοληθήκαμε, το οποίο αποτελεί μέρος των Διαγωνισμών που παρουσίασε το Συνέδριο Semeval (Semantic Evaluation) το 2018, ενώ στην συνέχεια πραγματοποιούμε μία έρευνα επάνω στην δουλειά που έχει προηγηθεί, τόσο σε θέματα που αφορούν τα ίδια τα νευρωνικά, αλλά και σε παρεμφερείς διαδικτυακές εφαρμογές. Ακολουθεί μία ενδελεχής ανάλυση του θεωρητικού και τεχνικού υπόβαθρου στα οποία στηριχθήκαμε για την εκπόνηση αυτής της διπλωματικής.

Στη συνέχεια παρουσιάζουμε τα στάδια των Αρχιτεκτονικών που υλοποιήσαμε και τους τρόπους με τους οποίους αυτές σχεδιάστηκαν, κάνοντας και μία αναλυτική αναφορά στους **Γενετικούς Αλγορίθμους** που χρησιμοποιήσαμε για τον Συνδιασμό των Μεθόδων μας (ensemble) και για Μετατροπή Προβλέψεων (mapping). Έπειτα, συνεχίζουμε με την αναλυτική περιγραφή των δεδομένων, από τον τρόπο συλλογής τους μέχρι και την προεπεξεργασία τους. Συγχρόνως, παρουσιάζονται τα αποτελέσματα των Μοντέλων που υλοποιήσαμε, αλλά και η σύγκρισή τους με τις επιδόσεις των συμμετεχόντων στο διαγωνισμό, πετυχαίνοντας μάλιστα, ειδικά στις μεθόδους που αφορούν παλινδρόμηση, αρκετά ανταγωνιστικές προβλέψεις. Επίσης στο Task Κατηγοριοποίησης Συναισθημάτων έχουμε την καλύτερη επίδοση.

Ακολουθεί η αναλυτική περιγραφή της Διαδικτυακής Εφαρμογής και πως αυτή υλοποιήθηκε τόσο από μεριάς frontend όσο και backend, σε συνδιασμό με όλα τα στιγμιότυπα λειτουργίας της εφαρμογής, από όλες τις σελίδες που την απαρτίζουν.

**Λέξεις-κλειδιά:** Συναισθηματική ανάλυση, επεξεργασία φυσικής γλώσσας, βαθειά μάθηση, μηχανική μάθηση, γενετικοί αλγόριθμοι, διαδικτυακή εφαρμογή



# Abstract

In modern society we experience a rapid development of information, which was sealed by the Internet and more specifically the social media, whose main representative is Twitter. The present thesis is about designing **Deep and Machine Learning** Models and Architectures, aimed at Sentiment Analysis in Twitter comments, as well as developing a **Web Application** that performs real time Emotion Analysis for such comments. The analysis mainly focuses on the 4 basic emotions: Anger, Fear, Joy, Sadness.

Initially we present the **Task: Affect in Tweets** we dealt with, which is part of the competitions presented by the Conference of Semeval (Semantic Evaluation) in 2018, and then we carry out a research into the work that has preceded both in issues related to the neural networks themselves, but also in similar web applications. What follows is a thorough analysis of the theoretical and technical background in which we relied for the elaboration of this thesis.

We then display the stages of the Architectures we designed and the ways in which they were implemented, making a detailed reference to **Genetic Algorithms** we used for Ensembling methods and Mapping Predictions. We then continue with the detailed description of data, starting from the way of collection up to their pre-processing. Furthermore, we list the results of the models we implemented, compared with the ones of the task's contestants, achieving in fact, especially in the methods of regression, quite competitive results. We also had the best results in Emotion Classification task.

We close with the detailed description of our Web Application and how it was implemented frontend and backend-wise, presenting all screenshots of the application, from every page included in it.

**Keywords:** Sentiment analysis, natural language processing, deep learning, machine learning, genetic algorithms, web application





# Ευχαριστίες

Η εκπόνηση της Διπλωματικής μου εργασίας σηματοδοτεί το τέλος των προπτυχιακών σπουδών μου στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Πραγματοποιήθηκε στα πλαίσια του Εργαστηρίου Ευφών και Υπολογιστικών Συστημάτων του τομέα Τεχνολογίας Πληροφορικής και Υπολογιστών, με επιβλέποντα καθηγητή τον κ. Ανδρέα Σταφυλοπάτη, τον οποίο και θα ήθελα να ευχαριστήσω πρωτίστως για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο επιστημονικό πεδίο, αλλά και για τα άτομα με τα οποία ο ίδιος με έφερε σε επικοινωνία και υπήρξαν αρωγοί στην προσπάθειά μου.

Ιδιαίτερως θα ήθελα να ευχαριστήσω τον κύριο Δρ. Γεώργιο Σιάλα ο οποίος στάθηκε δίπλα μου από την αρχή, δίνοντας μου τις κατάλληλες συμβουλές και κατευθύνσεις, προκειμένου να έχουμε το επιθυμητό αποτέλεσμα, με την βοήθεια του οποίου η συγκεκριμένη διπλωματική έγινε πραγματικότητα.

Θα ήθελα επίσης να ευχαριστήσω τον καθηγητή Γεώργιο Στάμου και την καθηγήτρια Κωνσταντίνα Νικήτα που συμπλήρωσαν την τριμελή επιτροπή.

Θα αποτελούσε παράλειψη να μην ευχαριστήσω όλους τους φίλους μου που με την στήριξή τους με βοηθούν όλα αυτά τα χρόνια να πετύχω τους στόχους μου και ιδιαίτερα την Ειρήνη, η οποία τον τελευταίο χρόνο με ενθάρρυνε καθημερινά και έμπρακτα.

Τέλος, ευχαριστώ βαθύτατα τους γονείς μου Νίκο και Παναγιώτα καθώς και τις αδερφές μου Γεωργία και Αγγελική για την αγάπη, την υπομονή και την στήριξη που μου έχουν προσφέρει όλα αυτά τα χρόνια.



Στους γονείς μου, Νίκο και Παναγιώτα  
Στις αδελφές μου, Γεωργία και Αγγελική

# Περιεχόμενα

Περίληψη	5
Abstract	7
Ευχαριστίες	9
<b>1 Εισαγωγή</b>	<b>19</b>
1.1 Κίνητρο	19
1.2 Μέσο Κοινωνικής Δικτύωσης Twitter	20
1.3 Ανάλυση Διαγωνισμού	21
1.4 Σκοπός Εργασίας	22
1.5 Διάρθρωση Διπλωματικής	23
<b>2 Συγγενείς εργασίες</b>	<b>25</b>
2.1 Πρώτες Έρευνες	25
2.2 Ένταση συναισθήματος	26
2.3 Συναισθηματική Ανάλυση στο Twitter	27
2.4 Διαδικτυακές Εφαρμογές ανάλυσης συναισθήματος	28
<b>3 Θεωρητικό Υπόβαθρο</b>	<b>29</b>
3.1 Μηχανική Μάθηση	29
3.1.1 Επιβλεπόμενη Μηχανική Μάθηση	29
3.1.2 Μη Επιβλεπόμενη Μηχανική Μάθηση	30
3.2 Αλγόριθμοι Μηχανικής Μάθησης	31
3.2.1 Μηχανές Διανυσμάτων Υποστήριξης - SVM	31
3.2.2 Δέντρα Απόφασεων - Decision Trees	32
3.2.2.1 Τυχαία Δάση - Random Forest	34
3.2.2.2 Bagging - Bootstrap Aggregating	35
3.3 Νευρωνικά Δίκτυα	35
3.3.1 Τεχνητά Νευρωνικά Δίκτυα	35
3.3.2 Στάδια Εκπαίδευσης	36
3.3.2.1 Συνάρτηση Ενεργοποίησης - Activation Function	37
3.3.2.2 Συνάρτηση Κόστους - Cost Function	40

3.3.2.3	Αλγόριθμος Βελτιστοποίησης - Optimization Algorithm	40
3.3.2.4	Μετρικές Αξιολόγησης - Evaluation Functions	42
3.3.3	Επαναλαμβανόμενα Νευρωνικά Δίκτυα - Recurrent Neural Networks	43
3.3.3.1	Μακρά Βραχυπρόθεσμη Μνήμη (LSTM)	44
3.3.3.2	Μηχανισμός Προσοχής (Attention Mechanism)	45
3.4	Επεξεργασία Φυσικής Γλώσσας	46
3.5	Τεχνικές Μετατροπής Κειμένου	48
3.5.1	Bag of Words	48
3.5.2	Μοντέλα n-γραμμάτων:	49
3.5.3	Λεξικά	49
3.5.4	Διανύσματα Λέξεων (Word Embeddings)	50
3.5.4.1	Word2Vec	50
3.5.4.2	Global Vectors (GloVe)	52
3.6	Γενετικοί Αλγόριθμοι	53
<b>4</b>	<b>Τεχνικό Υπόβαθρο</b>	<b>55</b>
4.1	Τεχνολογίες Νευρωνικών Δικτύων	55
4.1.1	Keras	55
4.1.2	Scikit-learn	56
4.2	Τεχνολογίες Διαδικτυακής Εφαρμογής	56
4.2.1	REST API εφαρμογές	56
4.2.2	TwitterAPI	57
4.2.3	Flask - Backend Framework	57
4.2.4	React - Frontend Framework	58
4.2.5	MongoDB - Βάση Δεδομένων	58
<b>5</b>	<b>Αρχιτεκτονική</b>	<b>59</b>
5.1	Υλοποιημένοι Αλγόριθμοι Μηχανικής Μάθησης	59
5.2	Αμφίδρομο LSTM	60
5.3	Αμφίδρομο LSTM με Μηχανισμό Προσοχής	61
5.4	Γενετικοί Αλγόριθμοι	62
5.4.1	Συνδιασμός Μοντέλων	62
5.4.2	Μετατροπή Ταξινομητών	63
5.5	Συμπληρωματικές Σημειώσεις	63
<b>6</b>	<b>Πειραματική Διαδικασία και Αποτελέσματα</b>	<b>65</b>
6.1	Δεδομένα	66
6.1.1	Πηγές δεδομένων	66
6.1.2	Διαδικασία Συλλογής	67
6.1.3	Τοποθέτηση Ετικετών	68
6.2	Προεπεξεργασία Δεδομένων	69

6.3	Δημιουργία Διανυσμάτων . . . . .	70
6.3.1	Διανύσματα Λέξεων . . . . .	71
6.3.2	Deerjmoji . . . . .	71
6.3.3	Λεξικά . . . . .	71
6.3.4	Εξαγωγή Χαρακτηριστικών - Feature Extraction . . . . .	73
6.4	Αξιολόγηση Αλγορίθμων . . . . .	73
6.4.1	Task #1: Emotion Intensity - Regression (EI-reg) . . . . .	73
6.4.2	Task #2: Emotion Intensity - Ordinal Classification (EI-oc) . . . . .	75
6.4.3	Task #5: Emotion - Classification (E-c) . . . . .	76
<b>7</b>	<b>Διαδικτυακή Εφαρμογή</b>	<b>77</b>
7.1	Αρχική Σελίδα . . . . .	77
7.2	Καρτέλα Tweets . . . . .	79
7.3	Καρτέλα κορυφαίων Tweets . . . . .	80
7.4	Καρτέλα Γραφημάτων Παλινδρόμησης . . . . .	81
7.5	Καρτέλα Γραφημάτων Ταξινόμησης . . . . .	81
7.6	Καρτέλα Γραφημάτων Θεματικής Ταξινόμησης . . . . .	83
7.7	Συμπληρωματικές Πληροφορίες . . . . .	83
<b>8</b>	<b>Συμπεράσματα</b>	<b>85</b>
8.1	Συμπληρωματικές Σημειώσεις . . . . .	86
	<b>Βιβλιογραφία</b>	<b>89</b>

# Κατάλογος Σχημάτων

1.1	Τροχός Συναισθημάτων - Plutchik's Wheel of Emotions . . . . .	21
3.1	Εφικτός / Μη Εφικτός Διαχωρισμός Δεδομένων . . . . .	31
3.2	Γραμμικός Διαχωρισμός Δεδομένων (πηγή: Wikipedia) . . . . .	31
3.3	Δέντρο Απόφασεων Διεξαγωγής Αγώνα (πηγή: <a href="#">geeksforgeeks</a> ) . . . . .	33
3.4	Τυχαία Δάση . . . . .	34
3.5	Perceptron . . . . .	35
3.6	Multilayer Perceptron (MLP) . . . . .	36
3.7	Συγμοειδής Συνάρτηση Ενεργοποίησης . . . . .	37
3.8	Υπερβολική Εφαπτομένη Συνάρτηση Ενεργοποίησης . . . . .	38
3.9	Συναρτήσεις ενεργοποίησης ReLU και Leaky ReLU . . . . .	39
3.10	Επαναλαμβανόμενο Νευρωνικό Δίκτυο (RNN) . . . . .	43
3.11	Επαναλαμβανόμενο Νευρωνικό Δίκτυο Διπλής Κατεύθυνσης (BRNN) . . . . .	44
3.12	Μονάδα Μακράς Βραχυπρόθεσμης Μνήμης (LSTM) . . . . .	44
3.13	Απλό RNN και RNN με Attention Mechanism . . . . .	46
3.14	Αρχιτεκτονικές CBoW και CSG της μεθόδου word2vec . . . . .	51
3.15	Γραφική Αναπαράσταση συγγενικών λέξεων μοντέλου GloVe . . . . .	52
3.16	Διασταύρωση Δεδομένων Γενετικών Αλγορίθμων . . . . .	54
3.17	Διάγραμμα Ροής Γενετικού Αλγορίθμου . . . . .	54
5.1	Υλοποιημένοι Αλγόριθμοι Μηχανικής Μάθησης . . . . .	60
5.2	Αρχιτεκτονική Αμφίδρομου LSTM . . . . .	61
5.3	Αρχιτεκτονική Αμφίδρομου LSTM με Μηχανισμό Προσοχής . . . . .	61
6.1	Στάδια Υλοποίησης . . . . .	65
7.1	Κύρια Σελίδα Διαδικτυακής Εφαρμογής . . . . .	77
7.2	Φιλτραρισμένα μη-αγγλικά trends . . . . .	78
7.3	Μέθοδοι Εισαγωγής Όρου (Free Text, Top Trends, Previous Searches) . . . . .	78
7.4	Καρτέλα Παρουσίασης Tweets . . . . .	79
7.5	Καρτέλα Παρουσίασης κορυφαίων Tweets . . . . .	80
7.6	Καρτέλα Παρουσίασης Γραφημάτων Παλινδρόμησης . . . . .	81
7.7	Καρτέλα Παρουσίασης Γραφημάτων Ταξινόμησης . . . . .	82
7.8	Καρτέλα Παρουσίασης Γραφημάτων Συναισθημάτων . . . . .	83





# Κατάλογος Πινάκων

6.1	Ετικέτες για αγγλικά tweets . . . . .	66
6.2	Συνολικά νούμερα για τα Αγγλικά tweets του Διαγωνισμού . . . . .	68
6.3	Ποσοστό tweets που επιλέχθηκαν ως αντιπροσωπευτικά δεδομένου συναισθήματος . . . . .	68
6.4	Αποτελέσματα Machine Learning Algorithms, Task #1: EI-reg . . . . .	74
6.5	Αποτελέσματα Bi-LSTM, Task #1: EI-reg . . . . .	74
6.6	Αποτελέσματα Bi-LSTM + Attention, Task #1: EI-reg . . . . .	74
6.7	Αποτελέσματα διαγωνιζόμενων στο Task #1: EI-reg . . . . .	75
6.8	Αποτελέσματα και Σύγκριση Προβλέψεων, Task #2: EI-oc . . . . .	75
6.9	Αποτελέσματα Machine Learning Algorithms, Task #5: E-c . . . . .	76
6.10	Αποτελέσματα και Σύγκριση Προβλέψεων, Task #5: E-c . . . . .	76



# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Κίνητρο

Η σύγχρονη εποχή χαρακτηρίζεται από πολλούς ως η Εποχή της Πληροφορίας. Η άποψη αυτή ενισχύεται όλο και περισσότερο με την ανάπτυξη του διαδικτύου και ιδιαιτέρως των Μέσων Κοινωνικής Δικτύωσης (Social Media). Στα διάφορα αυτά μέσα αποθηκεύεται και δημοσιεύεται καθημερινά ένας τεράστιος όγκος δεδομένων από απλούς χρήστες, επιχειρήσεις, πολιτικούς, κυβερνήσεις ακόμα και τρομοκράτες. Η δύναμη που έχουν καθώς και ο τρόπος με τον οποίο μπορούν να επηρεάσουν την κοινή γνώμη φαίνεται και από παραδείγματα απαγόρευσής τους όπως συμβαίνει μονίμως στην Κίνα, στην Αίγυπτο το 2011, στην Τουρκία το 2014, κοκ. Όλος αυτός ο όγκος δεδομένων οδήγησε στην ανάγκη αλλά και στην επιθυμία εξόρυξης πληροφορίας μέσα από τα Μέσα Κοινωνικής Δικτύωσης. Η Τεχνητή Νοημοσύνη και ιδιαιτέρως ο κλάδος της Μηχανικής και Βαθειάς Μάθησης συνέβαλαν τα μέγιστα στην επίτευξη αυτού του εγχειρήματος.

Η Συναισθηματική Ανάλυση (Sentiment Analysis) είναι τομέας που ασχολείται με την αναγνώριση της Συναισθηματικής Πόλωσης ενός χωρίου, είτε γραπτού είτε προφορικού. Η πολυπλοκότητα της γλώσσας και η ποικιλομορφία της (σύμφωνα με το Ethnologue <sup>1</sup> περισσότερες από 7,000 γλώσσες μιλούνται παγκοσμίως), κάνουν αυτό το πρόβλημα ιδιαιτέρως περίπλοκο. Έτσι καλούμαστε να αναπτύξουμε σύνθετες τεχνικές μοντελοποίησης του κειμένου προκειμένου να μπορούμε να αναγνωρίσουμε το συναίσθημα των λέξεων ως οντότητες αλλά και την αλληλεπίδραση μεταξύ τους.

Η αυτοματοποίηση αυτής της διαδικασίας μπορεί να βοηθήσει επιχειρήσεις και άτομα να λάβουν αποφάσεις προς δικό τους όφελος αλλά και να αποκτήσουν στατιστικά δεδομένα. Συχνά τέτοιες μέθοδοι χρησιμοποιούνται για προβλέψεις μετοχών χρηματιστηρίου, πρόθεσης ψήφου σε προεκλογική περίοδο, ακόμα και σε κυβερνητικές αποφάσεις. Παρατηρούμε, συνεπώς, πως το Συναίσθημα ως πηγή πληροφορίας μέσα από τον κόσμο των Μέσων Κοινωνικής Δικτύωσης μπορεί να μας επηρεάσει τόσο έμμεσα όσο και άμεσα.

---

<sup>1</sup><https://www.ethnologue.com/>

## 1.2 Μέσο Κοινωνικής Δικτύωσης Twitter

Αν και το Facebook αποτελεί το Μέσο με τους περισσότερους Χρήστες μέχρι και σήμερα, χρησιμοποιείται κυρίως για λόγους επικοινωνίας και διαφήμισης. Αντιθέτως το twitter αποτελεί το μεγαλύτερο κοινωνικό δίκτυο όσον αφορά τον τομέα της Ενημέρωσης και είναι στην 2η θέση παγκοσμίως σε θέματα επισκεψιμότητας (Μάιος 2019, σύγκριση μεταξύ κοινωνικών δικτύων)<sup>2</sup>. Δημιουργήθηκε τον Ιούλιο του 2016 από τον Jack Dorsey και την ομάδα του.

Οι δημοσιεύσεις στον συγκεκριμένο ιστιότοπο αποκαλούνται tweets. Αυτό που κάνει το Twitter ιδιαίτερος δημοφιλής είναι το επιτρεπτό άνω όριο χαρακτήρων που έχει για κάθε tweet. Η πλατφόρμα έγινε γνωστή με την χρήση των 140 χαρακτήρων, όμως το Νοέμβριο του 2017 το Twitter διπλασίασε αυτό το όριο, παρά ταύτα, όπως ανακοινώθηκε, το μέσο μήκος παρέμεινε σταθερό. Έτσι, σε ένα τόσο μικρό μήνυμα μπορούμε να βρούμε αρκετά συμπικνωμένη πληροφορία και σε συνδιασμό με τα emojis και emoticons, έχουμε στα χέρια μας ένα χρυσορυχείο tweets προς συναισθηματική ανάλυση. Αυτή ακριβώς η αμεσότητα είναι που διαφημίζει το Twitter, χρησιμοποιώντας ως slogan την φράση What's happening? (Τι συμβαίνει;), ενώ η διαδικασία επανακοινοποίησης (retweet) είναι επίσης ιδιαίτερος απλοϊκή. Στην προσπάθειά μας αυτή, έχουμε ως αρωγό το api που προσφέρει το Twitter για άντληση tweets, το TwitterAPI, τις δυνατότητες του οποίου αναλύουμε στη συνέχεια.

Το Twitter χρησιμοποιεί την έννοια της Τάσης (trend). Μία Τάση μπορεί να είναι είτε ένας αυτοτελής όρος (π.χ. usa), είτε μία ομάδα όρων (π.χ. hello world) είτε, με τον πιο γνωστό τρόπο, ένα hashtag (π.χ. #olympic\_games). Τα hashtags αποτελούνται από έναν μόνο όρο που ξεκινάει με τον χαρακτήρα # και χρησιμοποιούνται για την ομαδοποίηση δεδομένων, ενώ συνηθίζεται για τον νοητό διαχωρισμό λέξεων να χρησιμοποιείται ο χαρακτήρας underscore. Με τον χαρακτήρα @ ακολουθούμενο από το όνομα ενός λογαριασμού μπορούμε να αναφερθούμε δυναμικά στον λογαριασμό αυτό, ενώ υπάρχει η δυνατότητα like και σχολιασμού (comment) ενός tweet.

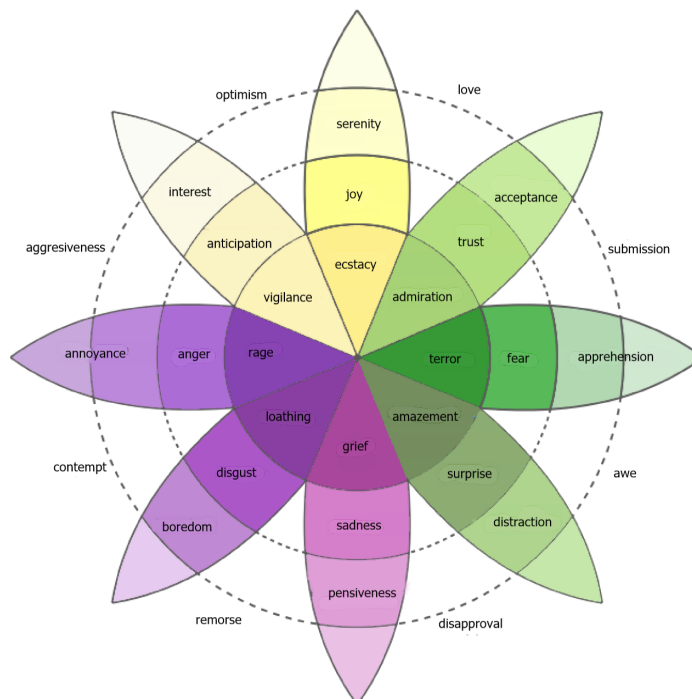
Πέρα από όλα τα πλεονεκτήματα που προσφέρει το κοινωνικό δίκτυο Twitter και τα οποία αναλύσαμε, διαθέτει επίσης και ένα τεράστιο πλήθος σκουπιδιών (garbage tweets). Η ανάλυση αν ένα tweet είναι άξιο εμπιστοσύνης αποτελεί επίσης τομέα που απασχολεί την επιστημονική κοινότητα της Τεχνητής Νοημοσύνης, όμως μια τέτοια διαδικασία δεν γίνεται στην παρούσα διπλωματική. Η ανάλυση γίνεται σε ό,τι πληροφορία μας δίνει το Twitter χωρίς να γίνεται έλεγχος αν αποτελεί garbage ή όχι. Επίσης, λόγω της δημοφιλίας του περιέχει και πολλές δημοσιεύσεις που είναι κατ' ουσία διαφημίσεις.

---

<sup>2</sup><https://www.alexa.com/topsites>

### 1.3 Ανάλυση Διαγωνισμού

Κάθε χρόνο πραγματοποίησης του, το Παγκόσμιο Συνέδιο Σημασιολογικής Αξιολόγησης (International Workshop on Semantic Evaluation) δημοσιεύει και μία σειρά από Διαγωνισμούς (tasks) που αφορούν ανάλυση κειμένου, στους οποίους μπορεί να συμμετάσχει ο καθένας. Έτσι, το 2018 <sup>3</sup> αποφασίσαμε να ασχοληθούμε με το πρώτο από αυτά τα tasks <sup>4</sup>, το οποίο ονομάζεται Affect in Tweets [1] (Επίδραση σε Tweets), χωρίς να συμμετάσχουμε επίσημα σε αυτόν, μιας και είχε ολοκληρωθεί. Ο διαγωνισμός αποτελούνταν από 5 διαφορετικά sub-tasks τα οποία αφορούν κυρίως την πρόβλεψη έντασης στα 4 βασικά συναισθήματα, όπως έχουν οριστεί από διάφορους ψυχολόγους (Ekman, 1992 [2], Plutchik, 1982 [3], Parrot, 2001[4], Frijda, 1988 [5]), Θυμό (Anger), Φόβο (Fear), Χαρά (Joy), Λύπη (Sadness) . Στο Σχήμα 1.1 βλέπουμε τον Τροχό των Συναισθημάτων του Plutchik [3]. Ο διαγωνισμός διέθετε ήδη δεδομένα με ετικέτες, οι οποίες είχαν τοποθετηθεί χειροκίνητα, κάνοντας το συγκεκριμένο dataset ιδιαίτερα ακριβές. Πέρα από την περίπτωση της Αγγλικής Γλώσσας με την οποία ασχοληθήκαμε, ο διαγωνισμός παρείχε dataset και για την Ισπανική και την Αραβική Γλώσσα. Στη συνέχεια αναλύουμε τα 3 από τα 5 subtasks στα οποία παρήξαμε προβλέψεις. Τα υπόλοιπα 2 με τα οποία δεν ασχοληθήκαμε, αφορούσαν Σθένος (Valance) Συναισθήματος, κλάδος ο οποίος θεωρήσαμε πως έχει ήδη αναλυθεί αρκετά.



Σχήμα 1.1: Τροχός Συναισθημάτων - Plutchik's Wheel of Emotions

<sup>3</sup><http://alt.qcri.org/semeval2018/>

<sup>4</sup><https://competitions.codalab.org/competitions/17751>

- **Task 1 - Emotion Intensity Regression (EI-reg):** Δεδομένου ενός tweet και ενός συναισθήματος  $E$  υπολογίστε την ένταση του  $E$  η οποία αντιπροσωπεύει καλύτερα την ψυχική κατάσταση του χρήστη, στο εύρος 0 (καθόλου ένταση  $E$ ) ως 1 (πλήρης ένταση  $E$ )
- **Task 2 - Emotion Intensity Ordinal Classification (EI-oc):** Δεδομένου ενός tweet και ενός συναισθήματος  $E$  ταξινομήστε την ένταση του  $E$  σε μία από τις 4 κλάσεις έντασης (καθόλου, λίγο, αρκετά, πολύ) η οποία αντιπροσωπεύει καλύτερα την ψυχική κατάσταση του χρήστη.
- **Task 5 - Emotion Classification (E-c):** Δεδομένου ενός tweet αναγνωρίστε την ύπαρξη ή μη των 11 κάτωθι συναισθημάτων ή την απώλεια όλων, η οποία αντιπροσωπεύει καλύτερα την ψυχική κατάσταση του χρήστη:
  1. Anger, (Θυμός)
  2. Anticipation (Ανυπομονησία)
  3. Disgust (Απέχθεια)
  4. Fear (Φόβος)
  5. Joy (Χαρά)
  6. Love (Αγάπη)
  7. Optimism (Αισιοδοξία)
  8. Pessimism (Απαισιοδοξία)
  9. Sadness (Λύπη)
  10. Surprise (Έκπληξη)
  11. Trust (Εμπιστοσύνη)

## 1.4 Σκοπός Εργασίας

. Συχνά δημιουργούμε **Νευρωνικά Δίκτυα** με αρκετά μεγάλη ακρίβεια στο Task με το οποίο ασχολούμαστε, όμως δεν τα χρησιμοποιούμε στην πράξη. Για τον λόγο αυτό αποφασίσαμε να δημιουργήσουμε μία **Διαδικτυακή Εφαρμογή** η οποία θα χρησιμοποιεί στην πράξη το Νευρωνικό μας δίκτυο και θα οπτικοποιεί τα αποτελέσματα. Επίσης, προκειμένου να αποφύγουμε την πεπατημένη αποφασίσαμε να χρησιμοποιήσουμε **Γενετικούς Αλγόριθμους** για τον συνδιασμό των διαφόρων προβλέψεων που επιτύχαμε, αλλά και για να δούμε πως μία απλή σχετική μέθοδος μπορεί να χρησιμοποιηθεί ως μεθοδολογία Ensemble. Τα τρία αυτά στοιχεία δηλαδή τα Νευρωνικά Δίκτυα, η Διαδικτυακή Εφαρμογή και οι Γενετικοί Αλγόριθμοι αποτελούν την βάση αυτής της Διπλωματικής.

## 1.5 Διάρθρωση Διπλωματικής

Στο Κεφάλαιο 2 παραθέτουμε τις προηγούμενες εργασίες που έχουν γίνει πάνω στην Συναισθηματική Ανάλυση και πιο συγκεκριμένα επικεντρωνόμαστε στην περίπτωση των tweets. Στο Κεφάλαιο 3 παραθέτουμε αναλυτικά όλο το θεωρητικό υπόβαθρο που χρησιμοποιήθηκε για την εκπόνηση αυτής της διπλωματικής, συνεχίζοντας στο Κεφάλαιο 4 όπου αναλύουμε όλες τις τεχνολογίες που χρησιμοποιήσαμε στα Νευρωνικά Δίκτυα αλλά και στην Διαδικτυακή Εφαρμογή. Στο Κεφάλαιο 5 δίνουμε μία Σχηματική αλλά και Γραπτή περιγραφή των αρχιτεκτονικών που χρησιμοποιήθηκαν, ενώ αντίστοιχα στο Κεφάλαιο 6 αναλύουμε ενδελεχώς τις μεθόδους που χρησιμοποιήσαμε καθώς και τα αποτελέσματα τα οποία αυτές κατάφεραν να επιτύχουν. Στη συνέχεια, έχουμε το Κεφάλαιο 7 όπου γίνεται η παρουσίαση όλης της Διαδικτυακής Εφαρμογής, ο τρόπος υλοποίησής της αλλά και όλες οι εικόνες που οπτικοποιούν τον τρόπο λειτουργία της. Κλείνουμε με το Κεφάλαιο 8 και τα Συμπεράσματα.





## Κεφάλαιο 2

# Συγγενείς εργασίες

Στο συγκεκριμένο κεφάλαιο θα αναπτύξουμε την προϋπάρχουσα έρευνα και εργασία που έχει γίνει στην Συναισθηματική Ανάλυση σε σύντομο κείμενο και πιο συγκεκριμένα σε Tweets. Η ραγδαία ανάπτυξη των Μέσων Κοινωνική Δικτύωσης έχει οδηγήσει στην ανάπτυξη μεθόδων ανάλυσης συναισθημάτων σε κείμενο που στοχεύει στην εξαγωγή συμπερασμάτων.

### 2.1 Πρώτες Έρευνες

Η έννοια της Συναισθηματικής Ανάλυσης (Sentiment Analysis) συναντάτε συχνά στην ξένη βιβλιογραφία και ως Εξόρυξη Άποψης (Opinion Mining). Ο όρος συναισθηματική ανάλυση σύμφωνα με τον Bing Liu 2015 [6] συναντάτε πρώτη φορά κατά πάσα πιθανότητα στο paper των Nasukawa and Yi (2003) [7]. Παρ' όλα αυτά, η έρευνα στο Συναίσθημα και την Άποψη είχε ξεκινήσει νωρίτερα (Wiebe, 2000 [8], Das and Chen 2001 [9], Morinaga et al., 2002 [10], Pang et al., 2002 [11], Turney, 2002 [12]). Ακόμα πιο πριν, μάλιστα, είχαμε σχετική έρευνα που αφορούσε:

- εξαγωγή υπομνημάτων: Hearst, 1992 [13]
- εύρεση άποψης σε αφηγήματα: Wiebe, 1994 [14]
- πρόβλεψη σημασιολογικού προσανατολισμού επιθέτων: Hatzivassiloglou and McKeown, 1997 [15]

Το πιο παρεμφερές paper με την εργασίας μας από όσα αναφέρθηκαν φαίνεται να είναι των Dad and Chen, 2001 [9] με τίτλο *Yahoo! For Amazon: Sentiment Parsing from Small Talk on the Web* όπου παρουσιάζεται μία μεθοδολογία για real time εξαγωγή συναισθήματος στον τομέα των Οικονομικών, από μηνύματα που δημοσιεύτηκαν στο διαδίκτυο.

## 2.2 Ένταση συναισθήματος

Η πλειοψηφία της μέχρι στιγμής έρευνας έχει επικεντρωθεί στην πρόβλεψη δυαδικών ετικετών σε δεδομένα κειμένου (πχ περιέχει χαρά - δεν περιέχει χαρά, περιέχει φόβο - δεν περιέχει φόβο, κτλ). Το 2005 οι Cecilia Ovesdotter Alm, Dan Roth, Richard Sproat [16] χρησιμοποιώντας Επιβλεπόμενη Μάθηση και ακολουθώντας την αρχιτεκτονική εκμάθησης SNoW <sup>1</sup>, μη έχοντας αρκετά δεδομένα προφορικού λόγου ώστε να προβλέψουν συναίσθημα στην ομιλία, όπως αρχικά σκόπευαν, ξεκίνησαν με δεδομένα γραπτού λόγου και προσπάθησαν να τα ομαδοποιήσουν με βάση τον βαθμό έντασης βασικών συναισθημάτων, χρησιμοποιώντας Bag of Words (BoW) καθώς και γλωσσικά features. Το 2013 μία ομάδα ερευνητών από το University of Washington, Seattle [17] έχοντας συλλέξει επί 4 χρόνια μηνύματα από chat που έχουν ανταλλαχθεί σε επιστημονικές συνεργασίες, στόχευουν στην αυτόματη εξαγωγή μίας ετικέτας που να προβλέπει κάθε φορά το είδος των συναισθημάτων που περιγράφουν καλύτερα το εκάστοτε μήνυμα. Η διαδικασία δημιουργίας των πραγματικών ετικετών για τα δεδομένα έγινε χειροκίνητα από την ίδια την ομάδα, καταλήγοντας σε 12 συναισθήματα μιας και για τα υπόλοιπα τα δεδομένα δεν ήταν αρκετά. Χρησιμοποιώντας Cross Validation για λόγους αποτελεσματικότητας αλλά και up-sampling και down-sampling, το οποίο τελικά και επιλέχθηκε, για να ισορροπήσουν τα δεδομένα τους, υλοποίησαν κώδικα σε Java <sup>2</sup> με κύρια βιβλιοθήκη τους για αλγορίθμους Classification την Weka <sup>3</sup>. Παρατήρησαν πως, λόγω του μικρού κατά κύριο λόγο μήκους των μηνυμάτων στο chat, τεχνικές όπως η Bag of Words οδηγούσαν σε overfitting, αλλά λόγω της συνεχούς ροής μηνυμάτων σε ένα chat αντιλήφθηκαν πως η πρόβλεψη ενός μηνύματος σχετίζεται άμεσα με μηνύματα τα οποία προηγήθηκαν ή ακολούθησαν, θέτοντας ως όριο σύνδεσης 2 μηνυμάτων τα 40 δευτερόλεπτα. Χρησιμοποίησαν το φίλτρο StringToVector <sup>4</sup> της Weka και για τις μεθόδους Classification πειραματίστηκαν σχεδόν με όλες τις επιλογές που δίνει η Weka αλλά καλύτερα αποτελέσματα έφεραν οι linear-kernel SVM και logistic regression πετυχαίνοντας ακρίβεια Precision της τάξης του 75%. Επιπλέον, η επιστημονική κοινότητα έχει δείξει αρκετό ενδιαφέρον στην σύνεση της εξαγωγής συναίσθηματος από τα Μέσα Κοινωνικής Δικτύωσης για λόγους κέρδους όπως για παράδειγμα έχει παρουσιαστεί από τους Lemmon and Portniaguina (2006)[18] υπάρχει σχέση ανάμεσα στο συναίσθημα και την αυτοπεποίθηση που έχουν οι επενδυτές, και στην αγορά μετοχών, ενώ οι Gilbert and Karahalios (2010)[19] υλοποιώντας ένα μοντέλο με πάνω από 85% ακρίβεια δείχνουν πως η εξαγωγή συναισθημάτων από το διαδίκτυο μπορεί να οδηγήσει σε αρκετά ικανοποιητικά αποτελέσματα πρόβλεψης στις μελλοντικές τιμές των μετοχών. Το μόνο dataset μέχρι πρότινος που περιελάμβανε ποσοστό συναίσθηματος (Regression) και όχι ομαδοποίηση (Classification) ήταν η εργασία των Strapparava και Mihalcea (2007)

<sup>1</sup>[https://cogcomp.org/page/software\\_view/SNoW](https://cogcomp.org/page/software_view/SNoW)

<sup>2</sup><https://www.java.com/en/>

<sup>3</sup><https://www.cs.waikato.ac.nz/ml/weka/>

<sup>4</sup><http://weka.sourceforge.net/doc.dev/weka/filters/unsupervised/attribute/StringToWordVector.html>

[20] στα πλαίσια του διαγωνισμού Semeval 2007 <sup>5</sup> όπου οι δημιουργοί του dataset έπρεπε να βαθμολογήσουν στο εύρος τιμών 0-100 διάφορα συναισθήματα για τίτλους εφημερίδων.

### 2.3 Συναισθηματική Ανάλυση στο Twitter

Από τις πρώτες έρευνες πρόβλεψης συναισθήματος για δεδομένα Twitter δημοσιεύτηκε στο paper του Go (2009) [21] όπου χρησιμοποιήθηκε Distant Learning ενώ συλλεχθήκαν ως θετικά tweets όσα τελειώνουν με θετικά emoticons όπως :) :-) και αντίστοιχα ως αρνητικά tweets όσα τελειώνουν σε αρνητικά emoticons όπως :( :-(. Για την πρόβλεψη χρησιμοποιήθηκαν μοντέλα όπως Naive Bayes, MaxEnt and Support Vector Machines (SVM) με τα καλύτερα αποτελέσματα για άλλη μια φορά να έρχονται, όπως είδαμε και σε προηγούμενες εργασίες που περιγράψαμε, από το SVM μοντέλο. Επίσης χρησιμοποιήθηκαν ως πρόσθετα χαρακτηριστικά μέθοδοι όπως Unigram, Bigram και part-of-speech (POS) χαρακτηριστικά με τα δύο τελευταία να μην φαίνεται πως βελτιώνουν τα αποτελέσματα. Οι Pak and Paroubek (2010) [22] χρησιμοποίησαν ίδια τεχνική για την άντληση tweets με τον τρόπο που αναφέρθηκε στο paper του Go (2009)[21] ακολουθώντας όμως μια διαφορετική προσέγγιση, αυτή του διαχωρισμού υποκειμενικότητας-αντικειμενικότητας, θεωρώντας ως υποκειμενικά tweets αυτά που συλλέχθηκαν με τον emoji τρόπο και ως αντικειμενικά, tweets που συλλέχθηκαν από λογαριασμούς όπως New York Times, Washington Posts κτλ. Διαφορετικός τρόπος στην άντληση δεδομένων από το Twitter εφαρμόζεται από τους Mohammad και Kiritchenko (2014) [23] όπου χρησιμοποιούνται τα hashtags συναισθημάτων (πχ #happy) με τη χρήση του Twitter Search API <sup>6</sup> για την αντιστοίχιση ετικετών στα tweets. Όπως και οι προηγούμενοι τους έτσι και αυτοί επικεντρώνονται στον SVM Classifier χρησιμοποιώντας επίσης βοηθητικές μεθόδους όπως Unigrams και Συναισθηματικά Λεξικά (Sentiment Lexicons) πετυχαίνοντας αρκετά καλά αποτελέσματα όσον αφορά τα 6 βασικά συναισθήματα του Ekman, (1992)[2] (θυμός, απέχθεια, φόβος, χαρά, θλίψη και έκπληξη). Οι Mohammad και Bravo-Marquez (2017)[24] δημιουργούν το πρώτο dataset αγγλικών tweets που περιέχει ετικέτες βαθμολογίας για κάθε tweet στα 4 βασικά συναισθήματα (θυμός, φόβος, χαρά, θλίψη). Το dataset δημιουργείται χειροκίνητα χρησιμοποιώντας την τεχνική Best Worst Scaling (BWS) η οποία σύμφωνα με τους Kiritchenko και Mohammad [25] φαίνεται να επιφέρει πιο ακριβή αποτελέσματα στην βαθμολογία συναισθημάτων μιας και ανάλογα τον άνθρωπο και την εποχή οι βαθμολογίες μπορεί να διαφέρουν. Αυτό το dataset χρησιμοποιήθηκε στην εργασία μας.

<sup>5</sup><http://web.eecs.umich.edu/~mihalcea/affectivetext/>

<sup>6</sup><https://developer.twitter.com/en/docs/tweets/search/overview>

## 2.4 Διαδικτυακές Εφαρμογές ανάλυσης συναισθήματος

Το **Tweet Visualizer** <sup>7</sup> είναι μία δωρεάν διαδικτυακή εφαρμογή που αναπτύχθηκε από τον Dr.Christopher Healey, Goodnight Distinguished Professor στο Institute of Advanced Analytics <sup>8</sup>. Είναι ίσως η πιο γνωστή διαδικτυακή εφαρμογή αυτόματης ανάλυσης συναισθημάτων για twitter. Ο τρόπος με τον οποίο δουλεύει μοιάζει αρκετά με την δική μας εφαρμογή μιας και με βάση έναν όρο αναζήτησης, αντλεί από το twitter tweets που περιέχουν αυτό τον όρο και τα αναλύει. Η συγκεκριμένη εφαρμογή έχει πολύ καλή οπτικοποίηση των δεδομένων όμως σε αντίθεση με την δική μας εφαρμογή, η ανάλυση των tweets γίνεται βάσει μόνο Συναισθηματικών Λεξικών και όχι Μηχανικής Μάθησης.

Το **Enguity** <sup>9</sup> είναι μία επί πληρωμή διαδικτυακή εφαρμογή που διαθέτει, όμως, μια βασική δωρεάν έκδοση. Η διαφορά της σε σχέση με τις παρεμφερείς της εφαρμογές είναι ότι σου επιτρέπει να αναζητήσεις πρόσφατες ειδησιογραφικές ιστορίες σε σχέση με έναν όρο αναζήτησης και στη συνέχεια αναζητώντας στο Facebook <sup>10</sup> και το Twitter <sup>11</sup> βρίσκει πόσες φορές έχει κοινοποιηθεί η συγκεκριμένη είδηση και ταυτόχρονα κάνει ανάλυση πάνω σε αυτήν. Η βασική διαφορά με την εφαρμογή μας είναι ότι η ανάλυση γίνεται μόνο με βάση το δίπολο θετική - αρνητική είδηση.

Το **MeaningCloud** <sup>12</sup> είναι ένα API και Excel Add-in που μέσω του Global sentiment, ενός feature που διαθέτει επιτρέπει την ομαδοποίηση tweets καθώς και την εξαγωγή θεμάτων για tweets. Επιπλέον, επιτρέπει στον εκάστοτε χρήστη να ανεβάσει αυτός δικά τους λεξικά που αφορούν την Εξαγωγή Θέματος (Topic Extraction) και την Ομαδοποίηση βάσει Συναισθήματος (Sentiment Classification).

Το **Socialmention** <sup>13</sup> είναι μία διαδικτυακή εφαρμογή που με βάση έναν όρο ψάχνει σε διάφορα κοινωνικά δίκτυα και microblogs, ανάμεσά τους και το Twitter. Η ανάλυση που κάνει αφορά περισσότερο στην εξαγωγή των πιο σημαντικών ειδικών λέξεων (keywords) και hashtags για τα tweets που έχει αντλήσει. Η συναισθηματική ανάλυση στην οποία επικεντρώνεται είναι η βαθμολογία στο εύρος 0-100 για της κατηγορίες θετικό, αρνητικό, ουδέτερο.

Αναφέρουμε επίσης και την **streamcrab** <sup>14</sup> όπου είναι μια διαδικτυακή εφαρμογή η οποία δεν βρίσκεται πλέον σε λειτουργία, αλλά η λογική που ακολουθούσε ήταν και εδώ αν ένα tweet έχει θετικό ή αρνητικό περιεχόμενο.

Παρατηρούμε συνεπώς πως καμία μέχρι στιγμής εφαρμογή δεν κάνει συναισθηματική ανάλυση σε Αγγλικά tweets όπως η εφαρμογή που υλοποιήσαμε.

<sup>7</sup>[https://www.csc2.ncsu.edu/faculty/healey/tweet\\_viz/tweet\\_app/](https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/)

<sup>8</sup><https://analytics.ncsu.edu/>

<sup>9</sup><https://www.theengenuity.com/>

<sup>10</sup><https://www.facebook.com/>

<sup>11</sup><https://twitter.com/>

<sup>12</sup><https://www.meaningcloud.com/products/sentiment-analysis>

<sup>13</sup><http://www.socialmention.com/>

<sup>14</sup><http://www.streamcrab.com/>

## Κεφάλαιο 3

# Θεωρητικό Υπόβαθρο

### 3.1 Μηχανική Μάθηση

Με τον όρο **Μηχανική Μάθηση** (Machine Learning) αναφερόμαστε στον υποκλάδο της Τεχνητής Νοημοσύνης (Artificial Intelligence) που στοχεύει στην εκμάθηση υπολογιστικών μηχανών ώστε αυτές να παίρνουν αυτόματα αποφάσεις. Μία τέτοια υπολογιστική μηχανή πρέπει να έχει την ικανότητα να βελτιώνει την απόδοσή της όταν λαμβάνει νέα ερεθίσματα, χωρίς να χρειάζεται να προγραμματιστεί από την αρχή. Το 1997 ο Mitchell [26] όρισε την Μηχανική Μάθηση ως εξής:

*“ Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία  $E$  ως προς κάποια κλάση εργασιών  $T$  και μέτρο απόδοσης  $P$ , αν η απόδοσή του σε εργασίες από το  $T$ , όπως μετρείται από το  $P$ , βελτιώνεται μέσω της εμπειρίας  $E$  ”*

Η διαδικασία η οποία συνήθως ακολουθείται είναι η τροφοδότηση του προγράμματος μας με δεδομένα, συνήθως αριθμητικά. Εάν αυτά δεν είναι αριθμητικά (όπως στην περίπτωση μας όπου έχουμε κείμενο) τότε το πρόγραμμα τα επεξεργάζεται με κατάλληλες μεθόδους όπως θα δούμε στη συνέχεια. Έπειτα, δημιουργούνται διανύσματα από αυτά τα δεδομένα, τα λεγόμενα Διανύσματα Χαρακτηριστικών (Feature Vectors) με τα οποία τροφοδοτούνται κατάλληλοι αλγόριθμοι. Για τον υπολογισμό της αξιοπιστίας κάθε προγράμματος υπάρχουν μετρικές οι οποίες επίσης θα αναλυθούν στην συνέχεια. Οι Αλγόριθμοι Μηχανικής Μάθησης χωρίζονται σε 3 μεγάλες κατηγορίες την Επιβλεπόμενη Μάθηση (Supervised Learning), την Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) και την Ενισχυμένη Μάθηση (Reinforcement Learning) που αποτελεί τον συνδιασμό των δύο προηγούμενων και δεν θα αναλυθεί στις επόμενες παραγράφους.

#### 3.1.1 Επιβλεπόμενη Μηχανική Μάθηση

Στην **Επιβλεπόμενη Μηχανική Μάθηση**, στόχος είναι η πρόβλεψη ζευγαριών εισόδου - εξόδου βάσει δεδομένων ζευγαριών εισόδου - εξόδου. Στην ουσία δηλαδή υπάρχει μία ετικέτα που χαρακτηρίζει κάθε δεδομένο εισόδου. Τα δεδομένα με τα οποία καλείται ο

αλγόριθμος να εκπαιδευτεί αποκαλούνται Δεδομένα Εκπαίδευσης (Training Data) και τα δεδομένα όπου αλγόριθμος καλείται να προβλέψει ονομάζονται Δεδομένα Ελέγχου (Test Data). Τα είδη προβλημάτων στα οποία χωρίζεται αυτή η κατηγορία μάθησης είναι τα ακόλουθα:

- **Ταξινόμησης (Classification):** Σε αυτήν την κατηγορία η ετικέτα (label) εξόδου αποτελείται από ένα σύνολο πεπερασμένων διακριτών επιλογών. Στόχος σε αυτή την περίπτωση είναι βάσει της εισόδου ο αλγόριθμος να αποφασίσει σε ποια κατηγορία εξόδου ανήκει. Σύνηθες τέτοιο πρόβλημα είναι η δυαδική κατηγοριοποίηση (ΝΑΙ, ΟΧΙ) πχ “Είναι η δίνια τραπεζική συναλλαγή ύποπτη;”, “Είναι αυτό το σχόλιο ρατσιστικό;”. Συνεπώς η έξοδος είναι της μορφής:

$$C = \{c_1, c_2, c_3, \dots, c_{|c|}\} \quad (3.1)$$

- **Παλινδρόμησης (Regression):** Σε αντίθεση με την προηγούμενη κατηγορία εδώ η έξοδος δεν είναι διακριτή αλλά συνεχής. Έτσι ο αλγόριθμος αυτή τη φορά έχει την δυνατότητα να προβλέψει μεταξύ ενός εύρους τιμών. Σύνηθες τέτοιο πρόβλημα είναι η πρόβλεψη της επόμενης τιμής μιας χρονοσειράς πχ τιμή μετοχής. Συνεπώς, η έξοδος είναι της μορφής:

$$C = [a, b] \text{ where } a, b \in \mathbb{R} \quad (3.2)$$

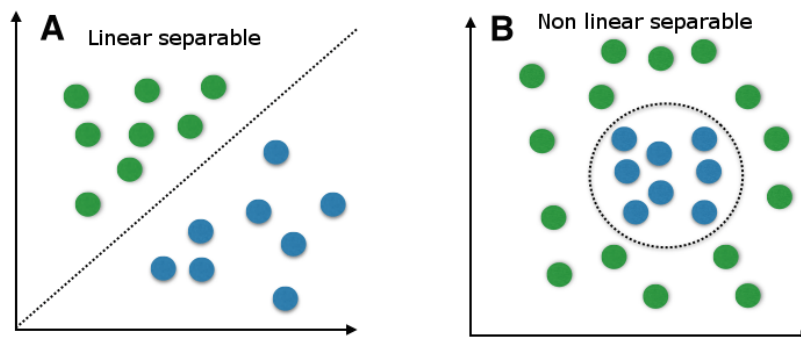
### 3.1.2 Μη Επιβλεπόμενη Μηχανική Μάθηση

Στην **Μη Επιβλεπόμενη Μηχανική Μάθηση**, γνωστή και ως Αυτο-οργάνωση, δεν υπάρχει εξωτερικός εκπαιδευτής ή κριτής που να επιβλέπει την διαδικασία μάθησης, όπως πχ είχαμε πριν με τα δεδομένα εξόδου. Αντ’ αυτού, υπάρχει ένα ανεξάρτητο από την εργασία μέτρο της ποιότητας της αναπαράστασης που καλείται να μάθει το δίκτυο και οι ελεύθερες παράμετροι του δικτύου βελτιστοποιούνται σε σχέση με αυτό το μέτρο (Haykin, 1998 [27]). Τέτοια προβλήματα αφορούν συνήθως Ομαδοποίηση (Clustering), όπου τα δεδομένα εισόδου ομαδοποιούνται σε συστάδες όπως πχ είναι η ομαδοποίηση ταινιών βάσει της περιγραφής τους ή η Αναγνώριση Ανωμαλιών (Anomaly detection), όπου διακρίνονται αποκλίνουσες συμπεριφορές σε σχέση με την πλειοψηφία των δεδομένων. Η Μη Επιβλεπόμενη Μηχανική Μάθηση είναι ιδιαίτερος σημαντικός κλάδος της Μηχανικής Μάθησης μιας και μπορούμε να διακρίνουμε πρότυπα και μοτίβα που δεν έχουν παρατηρηθεί από τον Άνθρωπο.

## 3.2 Αλγόριθμοι Μηχανικής Μάθησης

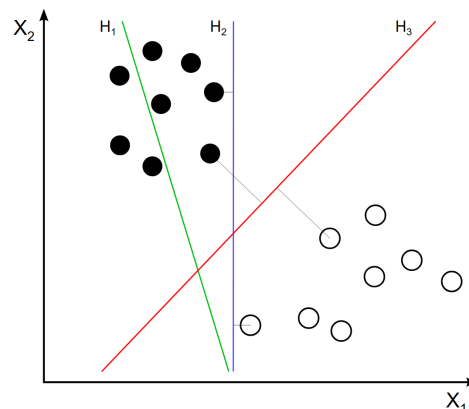
### 3.2.1 Μηχανές Διανυσμάτων Υποστήριξης - SVM

Οι Μηχανές Διανυσμάτων Υποστήριξης αποτελούν ένα από τα σημαντικότερα μοντέλα ταξινόμητων στο χώρο της Μηχανικής Μάθησης. Στηρίζονται στη γραφική αναπαράσταση των διάφορων στοιχείων και τον διαχωρισμό αυτών ανάλογα την κλάση τους. Στόχος τους είναι η εύρεση ενός υπερεπίπεδου το οποίο διαχωρίζει με τον καλύτερο δυνατό τρόπο τις διάφορες κλάσεις. Αυτό σημαίνει ότι αφενός πρέπει να διαχωρίζει πλήρως ή σχεδόν πλήρως τα στοιχεία των κλάσεων, αφετέρου να επιλεγεί εκείνο το υπερεπίπεδο το οποίο μεγιστοποιεί την απόσταση όλων των σημείων του προβλήματος από αυτό το δεδομένο υπερεπίπεδο.



Σχήμα 3.1: Εφικτός / Μη Εφικτός Διαχωρισμός Δεδομένων

Όπως παρατηρούμε στο Σχήμα 3.1 στο Διάγραμμα A τα δεδομένα μπορούν να διαχωριστούν γραμμικά συνεπώς αποκαλούνται Γραμμικά Διαχωρήσιμα, ενώ στο Διάγραμμα B τα δεδομένα δεν μπορούν να διαχωριστούν γραμμικά. Οι Μηχανές Διανυσμάτων Υποστήριξης, για να μπορέσουν να διαχωρίσουν αυτά τα δεδομένα πρέπει αρχικά να μην είναι γραμμικές, και συγκαταλέγονται στην κατηγορία των Μη Γραμμικών SVMs, οι οποίες χρησιμοποιούν τεχνικές μετασχηματισμού των συντεταγμένων σε άλλες διαστάσεις όπου η εύρεση υπερεπίπεδων είναι εφικτή.



Σχήμα 3.2: Γραμμικός Διαχωρισμός Δεδομένων (πηγή: Wikipedia)



Στο Σχήμα 3.2 παρατηρούμε πως η ευθεία  $H1$  αποτυγχάνει να διαχωρίσει τα διαφορετικά δεδομένα ενώ οι ευθείες  $H2$  και  $H3$  τα καταφέρνουν, με την τρίτη να αποτελεί την καλύτερη δυνατή επιλογή μιας και μεγιστοποιεί την απόσταση και από τα δύο δεδομένα. Στην ουσία ψάχνουμε να βρούμε το υπερεπίπεδο που ικανοποιεί τις συνθήκες 3.3 με το  $\vec{w}$  να αποτελεί το Κανονικό Διάνυσμα (Normal Vector) του υπερεπιπέδου.

$$\vec{w} \cdot \vec{x} - b = 0 \text{ and } \vec{w} \cdot \vec{x} - b = \begin{cases} 1, & \text{if } x \in \text{Class1} \\ -1, & \text{if } x \in \text{Class2} \end{cases} \quad (3.3)$$

Όπως αναφέραμε και πριν, εάν η εύρεση αυτού του υπερεπιπέδου δεν είναι εφικτή, τα δεδομένα υφίστανται ορισμένους μετασχηματισμούς οι οποίοι αποκαλούνται πυρήνες, με κάποιους από τους πιο γνωστούς να είναι:

- Ο πολυωνυμικός ομοιογενής πυρήνας:  $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)^d$
- Ο πολυωνυμικός ανομοιογενής πυρήνας:  $k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$
- Ο πυρήνας Gauss ακτινικής βάσης:  $k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2)$ , ( $\gamma > 0$ )
- Ο πυρήνας υπερβολικής εφαπτομένης:  $k(\vec{x}_i, \vec{x}_j) = \tanh(k\vec{x}_i \cdot \vec{x}_j + c)$ , ( $k > 0, c < 0$ )

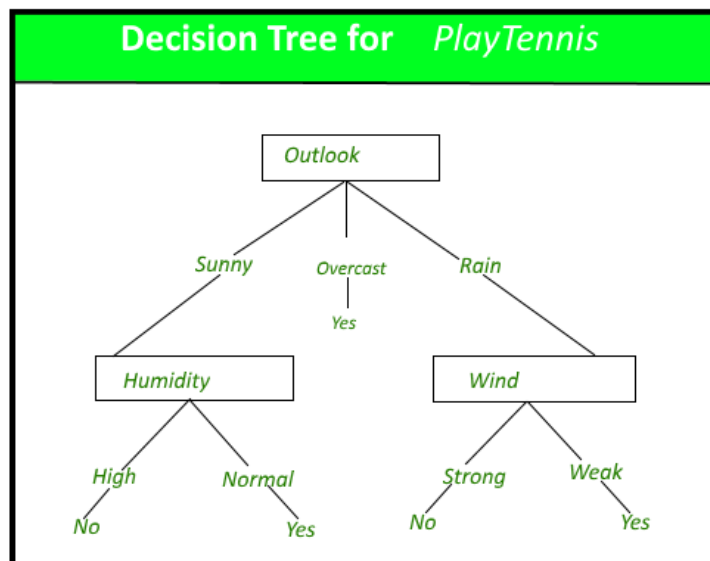
Το βασικό μειονέκτημα των SVMs είναι ότι εάν ο διαχωρισμός που θέλουμε να επιτύχουμε περιέχει  $n$  κλάσεις δεδομένων, τότε πρέπει να υπολογιστούν  $n - 1$  υπερεπίπεδα. Όπως καταλαβαίνουμε αυτό αυξάνει ιδιαίτερος τους υπολογιστικούς πόρους που πρέπει να καταναλωθούν. Παρ' όλα αυτά οι ΜΔΥ αποτελούν στην επεξεργασία φυσικής γλώσσας μία από τις πρώτες αλγοριθμικές προτιμήσεις των ερευνητών.

### 3.2.2 Δέντρα Απόφασεων - Decision Trees

Τα **Δέντρα Απόφασεων** (Decision Trees) αποτελούν μία από τις πιο γνωστές ομάδες αλγορίθμων Επιβλεπόμενης Μηχανικής Μάθησης και έχουν εφαρμοστεί σε διάφορους τομείς που απαιτούν ταξινόμηση όπως πχ η ιατρική διάγνωση περιστατικών και η αναγνώριση προτύπων σε εικόνες.

Ο αλγόριθμος ΔΑ οδηγεί στη δημιουργία μιας δενδροειδούς μορφής που τα φύλλα της αποτελούν κατηγορίες ταξινόμησης (classes), είναι δηλαδή η έξοδος του αλγορίθμου. Η δενδροειδής αυτή μορφή μπορεί να αναγνωστεί και ως ένα σύνολο κανόνων που καλούνται κανόνες ταξινόμησης (classification rules). Κάθε κόμβος του δένδρου αποτελεί μία συνθήκη ταξινόμησης. Αφού έχουμε σχηματίσει το δέντρο και θέλουμε να προβλέψουμε την έξοδο ενός δείγματος, κάνουμε μία απλή διάσχιση στο δέντρο όπου η τιμή του φύλλου στο οποίο θα καταλήξουμε αποτελεί και την πρόβλεψή μας. Για να γίνουν πιο κατανοητά τα παραπάνω ακολουθεί ένα απλό παράδειγμα καθημερινότητας Δέντρου Απόφασεων που δείχνει αν είναι εφικτή η διεξαγωγή ενός αγώνα αντισφαίρισης (tennis).





Σχήμα 3.3: Δέντρο Απόφασεων Διεξαγωγής Αγώνα (πηγή: [geeksforgeeks](#))

Έστω ότι η είσοδος προς εξέταση είναι η ακόλουθη:

*(Outlook = Sunny, Humidity = High, Wind = Strong )*

Τότε όπως βλέπουμε θα ακολουθεί η πιο αριστερή κάθετη διαδρομή οπότε η είσοδος θα καταχωρηθεί στην αρνητική Κλάση. Όπως είναι φυσικό σε ένα σε ένα υπολογιστικό ΔΑ δεν θα έχουμε λογικές ερωτήσεις αλλά συγκρίσεις μεταξύ διανυσμάτων. Το παράδειγμα δίνεται για λόγους κατανόησης.

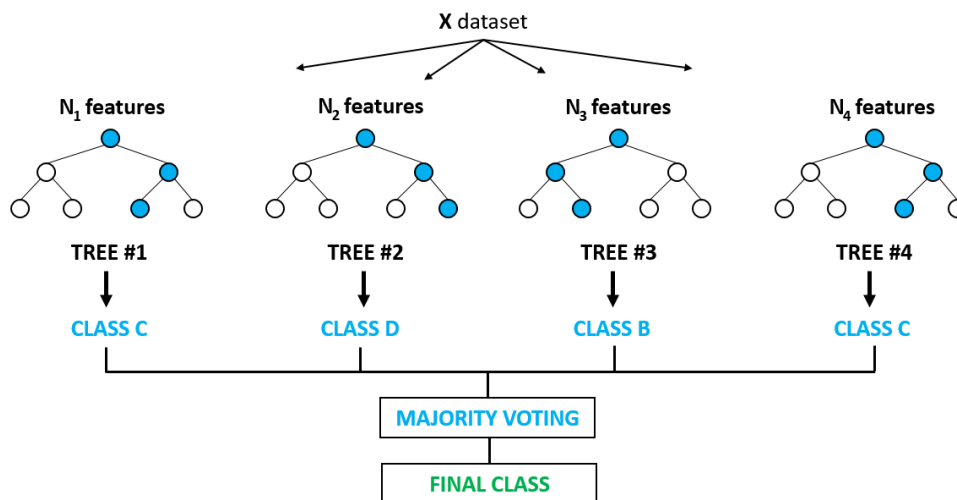
Τα ΔΑ μπορούν εύκολα να παράξουν κατανοητούς κανόνες καθώς και να υποδείξουν αυτές τις μεταβλητές της εισόδου οι οποίες έχουν το μεγαλύτερο βάρος στην τελική πρόβλεψη, ενώ σε αντίθεση με τις Μηχανές Διανυσμάτων Υποστήριξης που είδαμε πριν, δεν επηρεάζονται αρνητικά από προβλήματα πολλών Κλάσεων. Στον αντίποδα και στα αρνητικά του αλγορίθμου, τα ΔΑ βρίσκουν δυσκολίες όταν το πρόβλημα αφορά συνεχείς τιμές και όχι ταξινόμηση ενώ ακόμα δεν προσφέρουν ικανοποιητικά αποτελέσματα όταν τα Δεδομένα Εκπαίδευσης δεν είναι αρκετά στο πλήθος. Έτσι, καταλαβαίνουμε πως οι αλγόριθμοι Δέντρων Απόφασεων χρειάζονται ένα αρκετά μεγάλο όγκο δεδομένων για την εκπαίδευσή τους με αποτέλεσμα να καταναλώνουν αρκετούς υπολογιστικούς πόρους, έχοντας μεγάλο κόστος που αφορά χώρο και χρόνο. Παρά τα αρνητικά τους όμως, τα ΔΑ αποτελούν μία από τις πιο διαδεδομένες μεθόδους στην Μηχανική Μάθηση.

### 3.2.2.1 Τυχαία Δάση - Random Forest

Μία από τις πιο συνηθισμένες τεχνικές στην Επιβλεπόμενη Μάθηση είναι η εκπαίδευση πολλαπλών ταξινομητών ανεξάρτητα μεταξύ τους και ο συνδιασμός τους στο τέλος για την πρόβλεψη της τελικής τιμής (Ensemble). Η συγκεκριμένη τεχνική φαίνεται να επιφέρει σαφώς καλύτερα αποτελέσματα από την απλή επικέντρωση σε έναν ταξινομητή, μιας και κάθε ταξινομητής φαίνεται να επικεντρώνεται σε διαφορετικές παραμέτρους αδιαφορώντας για άλλες που αντίστοιχοι ταξινομητές θα αξιοποιούσαν.

Έτσι, κάθε ταξινομητής εκπαιδεύεται ξεχωριστά και στο τέλος συνδιάζοντας όλες τις τιμές και χρησιμοποιώντας μία συνάρτηση συσχέτισης η οποία θα πάρει όλες τις τιμές επιστρέφει την τελική πρόβλεψη. Τέτοιες συναρτήσεις είναι συνήθως η Πλειοψηφία, ο Μέσος Όρος, το Μέτρο Διαφοράς, η Τυπική Απόκλιση, κτλ, οι οποίες επιλέγονται ανάλογα με την φύση του προβλήματος.

Τα **Τυχαία Δάση (Random Forest)** αποτελούν μία τέτοια κατηγορία Συνδιαστικής Ταξινόμησης όπου οι Ταξινομητές Βάσης (Base Classifiers), οι παράλληλοι και ανεξάρτητοι δηλαδή ταξινομητές, είναι δέντρα απόφασεων. Στο Σχήμα 3.4 βλέπουμε ένα παράδειγμα ταξινομητή Τυχαίων Δασών όπου η συνάρτηση συσχέτισης είναι η Πλειοψηφία.



Σχήμα 3.4: Τυχαία Δάση

Ο πρώτος αλγόριθμος Τυχαίων Δασών δημιουργήθηκε το 1995 από τον Tin Kam Ho [28] χρησιμοποιώντας την τεχνική Random Subspace, η οποία στοχεύει στην μείωση της συσχέτισης ανάμεσα στους ταξινομητές. Στην συνέχεια, δύο χρόνια αργότερα, είχαμε την εξέλιξη αυτού του αλγορίθμου από τον Breiman [29] ο οποίος εισήγαγε την τεχνική bagging που ο ίδιος είχε αναπτύξει το 1994, την οποία αναλύουμε στη συνέχεια. Ο συνδιασμός των δύο τεχνικών του Ho και Breiman οδηγεί στην δημιουργία Δέντρων Απόφασεων με ελεγχόμενη διακύμανση.

### 3.2.2.2 Bagging - Bootstrap Aggregating

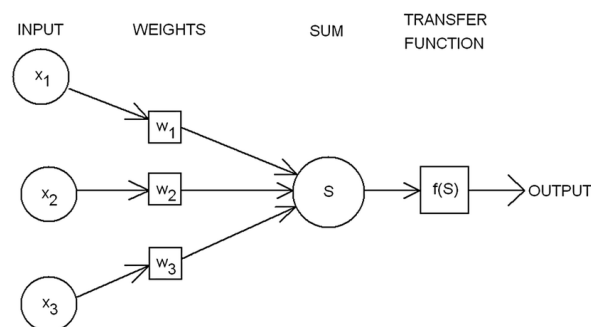
Όπως αναφέραμε και προηγουμένως, η τεχνική **Bagging (Bootstrap Aggregating)** εισήχθη το 1994 από τον Breiman [30]. Η τεχνική αυτή βασίζεται στην ιδέα ότι τα δείγματα του συνόλου εκπαίδευσης μπορούν να επαναχρησιμοποιηθούν για περαιτέρω εκπαίδευση. Έτσι, έχοντας επιλέξει για ένα υπομοντέλο ότι από τα  $n$  δεδομένα εκπαίδευσης θα χρησιμοποιήσουμε τα  $k$ , με  $n < k$ , τότε έχουμε ένα πλήθος  $n - k$  δεδομένων που μπορούμε να τα χρησιμοποιήσουμε για την έλεγχο αξιοπιστίας του υπομοντέλου μας. Ο αλγόριθμος ξεκινάει έχοντας αρχικά ένα σύνολο Δεδομένων Εκπαίδευσης μεγέθους  $n$  από τα οποία παράγει με χρήση ομοιόμορφης δειγματοληψίας με αντικατάσταση,  $m$  υποσύνολα εκπαίδευσης. Αυτό το δείγμα καλείται bootstrap δείγμα. Η τεχνική αυτή φαίνεται να έχει ιδιαίτερα καλά αποτελέσματα όταν τα υπομοντέλα που χρησιμοποιούνται είναι Δέντρα Απόφασεων, ενώ έχει αποδειχθεί πως βοηθούν αρκετά στην μείωση του overfitting και της διακύμανσης, που στην περίπτωση μας εκφράζει το πόσο διαφέρουν μεταξύ τους οι προβλέψεις του κάθε ταξινομητή.

## 3.3 Νευρωνικά Δίκτυα

### 3.3.1 Τεχνητά Νευρωνικά Δίκτυα

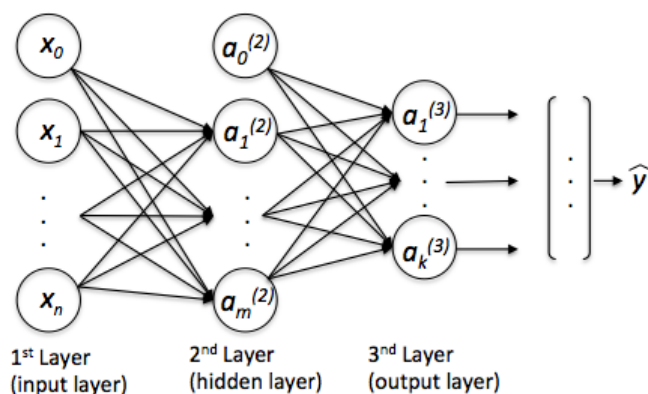
Τα **Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)** αποτελούν την προσπάθεια του ανθρώπου να προσομοιώσει τις εγκεφαλικές λειτουργίες σε θέματα που αφορούν επεξεργασία και μετάδοση πληροφορίας. Ένα Νευρωνικό Δίκτυο δεν αποτελεί έναν αλγόριθμο, αλλά στην ουσία ένα εργαλείο - τεχνική - framework πάνω στο οποίο τρέχουν διάφοροι αλγόριθμοι Μηχανικής Μάθησης.

Το πρώτο βήμα στον κλάδο των Νευρωνικών Δικτύων έγινε από τους McCulloch και Pitts το 1943 [31] οι οποίοι προσπάθησαν να φτιάξουν έναν τεχνητό νευρώνα. Ακολούθησε το 1949 ο Hebb [32] που εισήγαγε μία Μαθησιακή Υπόθεση βασισμένη στην πλαστικότητα των Νευρώνων η οποία έμεινε γνωστή ως Hebbian-ή Μάθηση, φτάνοντας στο 1958 και την δημιουργία του Νευρώνα **Perceptron** από τον Rosenblatt [33], έναν νευρώνα που αναγνωρίζει πρότυπα, ιδιαίτερα διαδεδομένο στην επιστήμη του Machine Learning (Σχ. 3.5).



Σχήμα 3.5: Perceptron

Ο νευρώνας του Rosenblatt από μόνος του αποτελεί έναν γραμμικό ταξινομητή. Η πλειοψηφία όμως των προβλημάτων δεν μπορεί να επιλυθεί με τόσο απλές μεθόδους. Για την αντιμετώπισή τους πρέπει να συνδιαστούν κατάλληλα πολλαπλοί νευρώνες, δημιουργώντας ένα Πολυεπίπεδο Δίκτυο Perceptron, το λεγόμενο **MultiLayer Perceptron (MLP)**. Το MLP αποτελείται από 3 στρώματα, το Στρώμα Εισόδου (Input Layer), το Κρυφό Στρώμα (Hidden Layer) και το Στρώμα Εξόδου (Output Layer), όπως παρατηρούμε στο Σχ. 3.6.



Σχήμα 3.6: Multilayer Perceptron (MLP)

### 3.3.2 Στάδια Εκπαίδευσης

Ένα νευρωνικό δίκτυο για να είναι αποτελεσματικό οφείλει να μπορεί να εκπαιδευτεί μέσα σε λογικά πλαίσια χρόνου βάσει των υπολογιστικών πόρων που διαθέτουμε, αλλά ταυτόχρονα να είναι και αξιόπιστο. Για να γίνει αυτό πρέπει να επιλεγθούν οι κατάλληλες παράμετροι και μέθοδοι που σε κάποιες περιπτώσεις μπορεί εκ των προτέρων να γνωρίζουμε ποιες είναι αυτές και απλώς να χρειάζεται να υλοποιήσουμε την αρχιτεκτονική του νευρωνικού μας, ενώ σε άλλες να πρέπει να εργαστούμε με εξαντλητικές μεθόδους ή τεχνικές δοκιμής - σφάλματος (trial and error), για να αποφασίσουμε ποιες υπερπαράμετροι είναι πιο αποτελεσματικές. Ταυτόχρονα το νευρωνικό μας πρέπει να μπορεί να γενικεύσει, δηλαδή να μην κάνει overfit πάνω στα δεδομένα εκπαίδευσης. Στην ουσία το νευρωνικό μας είναι μία επαναληπτική διαδικασία, όπου σε κάθε επανάληψη προσπαθεί να βελτιώσει την απόδοσή του. Κάθε τέτοια επανάληψη την αποκαλούμε εποχή (epoch) και το πλήθος της εναπόκειται στην κρίση του προγραμματιστή. Η διαδικασία με την οποία ανανεώνονται τα βάρη του κάθε στρώματος χωρίζεται σε δύο κατηγορίες, οι οποίες διαχωρίζονται με βάση τον τρόπο εισαγωγής των δεδομένων:

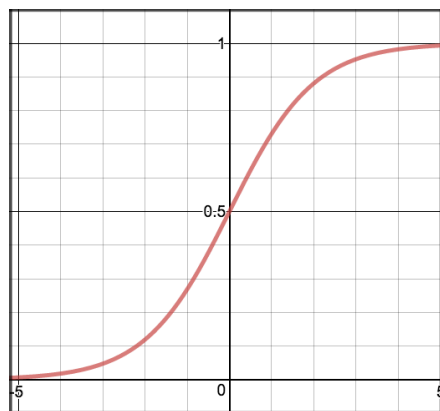
- **Στοχαστική Εκμάθηση (Stochastic Learning)**, όπου τα δεδομένα εισόδου εισέρχονται το ένα μετά το άλλο και το καθένα δημιουργεί μία νέα τροποποίηση στα βάρη.
- **Εκμάθηση Τεμαχίων (Batch Learning)**, όπου τα βάρη τροποποιούνται βάσει ενός πλήθους (batch) πολλαπλών δεδομένων.

Όλη η διαδικασία εκμάθησης εξαρτάται από τον υπολογισμό διαφόρων παραγώγων και συναρτήσεων όπως η Συνάρτηση Ενεργοποίησης, η Συνάρτηση Κόστους και η Συνάρτηση Βελτιστοποίησης, τις οποίες αναλύουμε παρακάτω.

### 3.3.2.1 Συνάρτηση Ενεργοποίησης - Activation Function

Η **Συνάρτηση Ενεργοποίησης** είναι στην ουσία η συνάρτηση που παίρνει την τιμή του προηγούμενου νευρώνα ως είσοδο και επιστρέφει την έξοδο που με την σειρά της θα είναι είσοδος στον επόμενο νευρώνα. Αποτελεί δηλαδή ένα φίλτρο ή καλύτερα έναν ενδιάμεσο κόμβο απόφασης. Το νευρωνικό μπορεί να λειτουργήσει και χωρίς καμία συνάρτηση ενεργοποίησης όπου η έξοδος δίνεται κατευθείαν ως είσοδος, είναι σαν να έχουμε ως Συνάρτηση Ενεργοποίησης δηλαδή, την γραμμική συνάρτηση ταυτότητας  $f(x) = x$ . Στη συνέχεια παρουσιάζουμε μερικές από τις πιο γνωστές συναρτήσεις ενεργοποίησης.

- **Σιγμοειδής (Sigmoid)**: Μια από τις πιο γνωστές και ευρέως χρησιμοποιούμενες συναρτήσεις. Λειτουργεί με την αντιστοίχιση όλων των τιμών στο διάστημα  $(0, 1)$ . Όπως βλέπουμε στο σύνολο είναι ανοικτό συνεπώς οι τιμές 0 και 1 δεν μπορούν στην ουσία να υπάρξουν. Στα θετικά της συνάρτησης αυτής συγκαταλέγεται η κανονικοποίηση που υφιστάνται τα δεδομένα στο διάστημα  $(0, 1)$ . Στο αντίποδα όμως αυτή η κανονικοποίηση, για τιμές εισόδου αρκετά μεγάλες ή αρκετά μικρές δεν έχουμε κάποια αισθητή διαφοροποίηση στην έξοδο. Αυτό το φαινόμενο είναι γνωστό ως Εξαφανιζόμενη Κλίση (Vanishing gradient). Η γραφική παράσταση της συνάρτησης δίνεται στο Σχήμα που ακολουθεί (Σχ. 3.7)

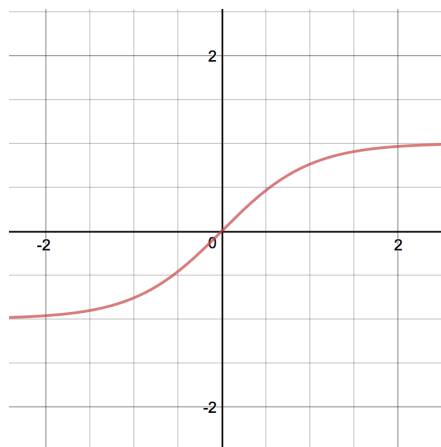


Σχήμα 3.7: Σιγμοειδής Συνάρτηση Ενεργοποίησης

Η Σιγμοειδής Συνάρτηση δίνεται από τον τύπο:

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.4)$$

- **Συνάρτηση Υπερβολικής Εφαπτομένης (tanh / Hyperbolic Tangent):**  
Η Υπερβολική Εφαπτομένη μοιάζει αρκετά με την Σιγμοειδή σε θέματα κατανομής των τιμών, με την διαφορά ότι τώρα γίνονται στο διάστημα  $(-1, 1)$ . Είναι μηδενοκεντρική που της δίνει πλεονέκτημα σε δεδομένα που εστιάζουν στο 3-πολο (αρνητικό, ουδέτερο, θετικό). Τα υπόλοιπα πλεονεκτήματά και μειονεκτήματά της είναι όμοια με την Σιγμοειδή. Ακολουθεί το γράφημα της στο επόμενο Σχήμα (Σχ. 3.8).



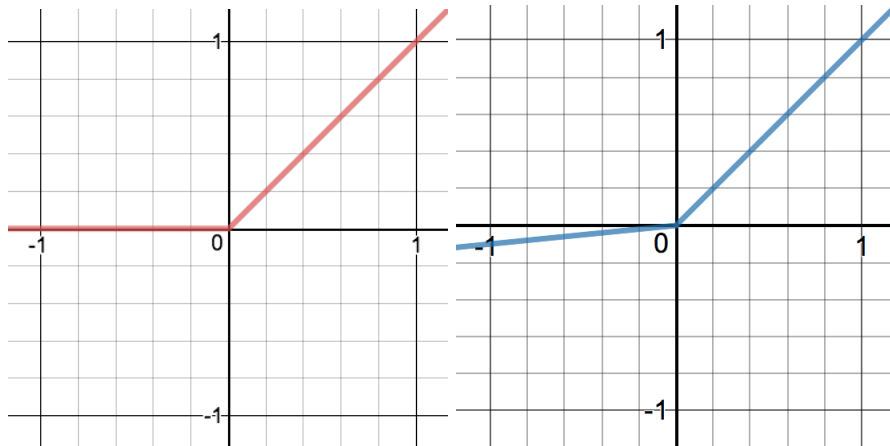
Σχήμα 3.8: Υπερβολική Εφαπτομένη Συνάρτηση Ενεργοποίησης

Η Υπερβολική Εφαπτομένη δίνεται από τον τύπο:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.5)$$

- **ReLU (Rectified Linear Unit):** Αποτελεί την πιο διαδεδομένη πλέον Συνάρτηση Ενεργοποίησης. Στην ουσία είναι η γραμμική συνάρτηση για τις θετικές τιμές σε συνδιασμό με τον μηδενισμό όλων των αρνητικών τιμών. Η συνάρτηση αυτή οδηγεί σε γρήγορη σύγκλιση του δικτύου κάτι που την κάνει υπολογιστικά αποδοτική, ενώ αν και φαίνεται γραμμική, έχει μη σταθερή παράγωγο, γεγονός που κάνει επιτρεπτεί την τεχνική του Back Propagation που θα αναλύσουμε στη συνέχεια. Ταυτόχρονα, δεν εμφανίζονται φαινόμενα Εξαφανιζόμενη Κλίσης. Στα αρνητικά, η εκμηδένιση όλων των αρνητικών τιμών οδηγεί στην *θανάτωση* όλων των νευρώνων που παίρνουν οποιαδήποτε στιγμή αρνητική τιμή, μιας και αυτή δεν μπορεί να ξανααλλάξει, έτσι αυτοι οι νευρώνες δεν μπορούν να επηρεάσουν στην συνέχεια με οποιοδήποτε τρόπο το νευρωνικό μας δίκτυο. Το φαινόμενο αυτό είναι γνωστό ως *The Dying ReLU*

*problem*. Για την αντιμετώπιση του, έχει δημιουργηθεί μία παραλλαγή της ReLU, όπου οι αρνητικές τιμές πολλαπλασιάζονται με μία μικρή σταθερά  $c$ , στην περίπτωση μας με  $c = 0.01$ , αντί να μηδενίζονται, η οποία ονομάζεται **Leaky ReLU**. Τα γραφήματα και των δύο παρουσιάζονται παρακάτω στις δυο γραφικές παραστάσεις που ακολουθούν (Σχ. 3.9).



Σχήμα 3.9: Συναρτήσεις ενεργοποίησης ReLU και Leaky ReLU

Η ReLU και Leaky ReLU δίνονται από τους τύπους:

$$f(x) = x^+ = \max(0, x) \text{ and } f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01x, & \text{otherwise} \end{cases} \quad (3.6)$$

Στις παραπάνω συναρτήσεις θα μπορούσαμε να προσθέσουμε και την συνάρτηση **Softmax** η οποία δεν είναι Συνάρτηση Ενεργοποίησης αλλά συνήθως εφαρμόζεται στο στάδιο εξόδου κάθε στρώματος συχνά σε συνδιασμό με μία κανονική Συνάρτηση Ενεργοποίησης. Η χρησιμότητα της Softmax είναι η κανονικοποίηση των δεδομένων στο εύρος  $[0, 1]$  παίρνοντας ως είσοδο δεδομένα μεγέθους  $K$  και δημιουργώντας μία πιθανοτική κατανομή η οποία αποτελείται από  $K$  πιθανότητες οι οποίες αθροίζονται στο 1. Η συνάρτησή της δίνεται από τον ακόλουθο τύπο:

$$f_i(x) = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}} \text{ for } i=1,2,\dots,J \quad (3.7)$$

### 3.3.2.2 Συνάρτηση Κόστους - Cost Function

Όπως έχουμε αναφέρει σκοπός ενός νευρωνικού δικτύου είναι συνεχώς να βελτιώνει την απόδοσή του. Για να γίνει αυτό πρέπει να υπάρχει ένα μέτρο που θα ενημερώνει το νευρωνικό για το αν βελτιώνεται ή όχι. Η **Συνάρτηση Κόστους (Cost Function or Loss Function)** αποτελεί αυτό ακριβώς το μέτρο που ενημερώνει και βοηθά το νευρωνικό να διορθώσει την συμπεριφορά του ώστε να ελαχιστοποιήσει το σφάλμα. Στην ουσία είναι μία μετρική που μας ενημερώνει για το πόσο κακά τα πηγαίνει ένα μοντέλο σε θέματα σύγκρισης τιμών Πρόβλεψης και Πραγματικών τιμών. Συμβολίζεται συνήθως με  $J(\theta)$ .

- **Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error):** όπου υπολογίζει τον μέσο όρο των τετραγώνων των σφαλμάτων. Η ύψωση στο τετράγωνο γίνεται προφανώς ώστε οι θετικές και αρνητικές αποκλίσεις να μην αλληλοαναιρούνται. Χρησιμοποιείται κυρίως για προβλήματα παλινδρόμησης (Σχέση 3.8) :

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3.8)$$

- **SVM Loss (Hinge loss):** Χρησιμοποιείται για προβλήματα Κατηγοριοποίησης. Με απλά λόγια το άθροισμα των σωστών προβλέψεων θέλουμε να είναι μεγαλύτερο από το άθροισμα των λανθασμένων. Είναι πιο γνωστή για την χρήση της στις Μηχανές Διανυσμάτων Υποστήριξης και αν και δεν είναι διαφορίσιμη είναι αρκετά εύκολη στην χρήση (Σχέση 3.9):

$$J(\theta) = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \quad (3.9)$$

- **Crossentropy Loss:** Αποτελεί την πιο συχνή επιλογή για προβλήματα Κατηγοριοποίησης. Στην ουσία αυτό που κάνει είναι να συγκρίνει δύο πιθανοτικές κατανομές αυτή της Πρόβλεψης με την αντίστοιχη της Πραγματικής τιμής. Σημαντικό στοιχείο της συνάρτησης αυτής είναι ότι είναι ιδιαίτερος αυστηρή, δηλαδή δίνει μεγάλη ποινή (penalty) ακόμα και για καλές αλλά όχι ακριβείς προβλέψεις (Σχέση 3.10):

$$J(\theta) = H(p, q) = - \sum_{x \in X} p(x) \log(q(x)) \quad (3.10)$$

### 3.3.2.3 Αλγόριθμος Βελτιστοποίησης - Optimization Algorithm

Οι **Αλγόριθμοι Βελτιστοποίησης** στην γενική τους χρήση, βοηθούν στο να μεγιστοποιήσουμε ή να ελαχιστοποιήσουμε μία συνάρτηση, στην περίπτωσή μας την Συνάρτηση Κόστους. Οι υπερπαράμετροι στο Κρυφό Στρώμα του νευρωνικού μας δικτύου παίζουν σημαντικό ρόλο στην τελική του απόδοση, συνεπώς χρειάζεται να βρούμε έναν αποτελεσματικό τρόπο για την επιτυχή ανανέωσή τους και για αυτό χρησιμοποιούμε τους Αλγορίθμους Βελτιστοποίησης. Χωρίζονται σε 2 μεγάλες κατηγορίες, τους αλγορίθμους



πρώτης τάξης οι οποίοι χρησιμοποιούν την πρώτη παράγωγο της Συνάρτησης κόστους και αντίστοιχα δεύτερης τάξης οι οποίοι χρησιμοποιούν την δεύτερη παράγωγο. Η δεύτερη κατηγορία δεν χρησιμοποιείται τόσο όσο η πρώτη καθώς αρκετά συχνά, ο υπολογισμός της τιμής της δεύτερης παραγώγου δεν συμφέρει υπολογιστικά, συνεπώς θα εστιάσουμε στην πρώτη κατηγορία όπου θα δούμε αλγορίθμους που στηρίζονται στην τεχνική της **Κατάβασης Κλίσης (Gradient Descent)**.

Ο πρώτος και ίσως ο πιο γνωστός Αλγόριθμος Βελτιστοποίησης είναι αυτός της **Οπίσθιας Διάδοσης Σφάλματος (Backpropagation)**. Προτάθηκε ως τεχνική από τον Kelley το 1960 [34], ενώ η ιδέα για την εφαρμογή αυτού του αλγορίθμου στα νευρωνικά δίκτυα πρωτοεμφανίζεται στην διπλωματική εργασία του Werbos το 1974 [35], παίρνοντας ιδιαίτερη δημοσιότητα τελικά από την εργασία των Rumelhart, Hinton, Williams το 1986 [36]. Η ιδέα του αλγορίθμου βασίζεται στην κατάστρωση ενός οργανωμένου πλάνου μικρής τροποποίησης κάθε βάρους του δικτύου, λαμβάνοντας υπόψιν το σφάλμα που έχουμε για μία συγκεκριμένη είσοδο, την αντίστοιχη επιθυμητή έξοδο και την ανάκληση του δικτύου. Εφαρμόζουμε το πλάνο σε εποχές, όπως και στο απλό perceptron διατρέχοντας με οργανωμένο τρόπο όλα τα δεδομένα και προσαρμόζουμε τα βάρη ανάλογα με το πόσο συνεισέφεραν στο συνολικό σφάλμα του δικτύου (credit assignment). Για τον καθορισμό της συνεισφοράς αυτής χρησιμοποιούμε την παράγωγο του μέσου τετραγωνικού σφάλματος ως προς το συγκεκριμένο βάρος (gradient descent).

Πλέον έχουν αναπτυχθεί διάφοροι αλγόριθμοι οι οποίοι στηρίζονται στην τεχνική του Gradient Descent και οι περισσότεροι από αυτούς έχουν βασιστεί στον αλγόριθμο **Στοχαστικής Κατάβασης Κλίσης (Stochastic Gradient Descent, SGD)**, Robbins 1951 [37]. Σε αντίθεση με την απλή Gradient Descent τεχνική όπου τα δείγματα επιλέγονται σε ομάδες, επιλέγει τυχαία εκείνα που θα επεξεργαστεί για αυτό και αποκαλείται Στοχαστικός. Υπολογίζεται από τον τύπο:

$$\theta' = \theta - \lambda \nabla J(\theta) \quad (3.11)$$

όπου  $\theta$  είναι οι προς υπολόγηση παράμετροι του νευρωνικού,  $\lambda$  το βήμα μάθησης και  $J(\theta)$  η Συνάρτηση Κόστους. Το πρόβλημα με τον SGD είναι πως το βήμα μάθησης  $\lambda$  όπως βλέπουμε είναι σταθερό και εάν το θέσουμε σε μεγάλη τιμή κατά την διάρκεια εύρεσης του τοπικού ελαχίστου θα έχουμε πολλές μεταπηδήσεις εκατέρωθέν του, εάν πάλι το μειώσουμε, η ταχύτητα σύγκλισης μειώνεται δραστικά. Για το λόγο αυτό έχουν δημιουργηθεί διάφορες παραλλαγές του όπως Mini Batch Gradient Descent, Momentum [38], Adagrad [39], AdaDelta [40]. Εκείνος όμως ο αλγόριθμος ο οποίος φαίνεται να λειτουργεί καλύτερα από τους υπόλοιπους και είναι ίσως αυτή τη στιγμή ο πιο ευρέως χρησιμοποιούμενος είναι ο **Adam (Adaptive Moment Estimation)** [41] ο οποίος χρησιμοποιεί μεταβλητό βήμα μάθησης  $\lambda$  ενώ αξιοποιούνται μέσοι όροι τόσο της πρώτης όσο και της δεύτερης βαθμίδας της κλίσης των διαφορικών.

### 3.3.2.4 Μετρικές Αξιολόγησης - Evaluation Functions

Οι **Μετρικές Αξιολόγησης** αν και δεν αποτελούν συστατικό του Δικτύου, είναι αρκετά σημαντικές για την εξαγωγή συμπερασμάτων, προκειμένου να μπορούμε να συγκρίνουμε διαφορετικά μοντέλα μεταξύ τους, ούτως ώστε να μπορούμε να κρίνουμε πιο αποδίδει αποτελεσματικότερα. Σημαντική επίσης είναι και η διαφοροποίηση ανάμεσα στις μετρικές. Για παράδειγμα αν ένα μοντέλο πετυχαίνει την καλύτερη επίδοση σε ένα task για την μετρική Macro-averaged Recall δεν σημαίνει απαραίτητα ότι για την μετρική F1 Score θα έχει επίσης την καλύτερη απόδοση.

- **Accuracy:** Ο τίτλος αποτελεί και την εξήγηση αυτής της μετρικής. Είναι στην πραγματικότητα το πλήθος των σωστών προβλέψεων προς το πλήθος των συνολικών προβλέψεων. Χρησιμοποιείται για προβλήματα Ταξινόμησης και υπολογίζεται από την Σχέση 3.12.

$$Accuracy = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}} \quad (3.12)$$

- **F1-score:** Για να αναλύσουμε την δεδομένη μετρική πρέπει πρώτα να ορίσουμε κάποιες ενδιάμεσες. Θεωρώντας ότι ένα πρόβλημα ταξινόμησης παίρνει μόνο δυαδικές τιμές Θετικό - Αρνητικό (Positive - Negative) ορίζουμε τις παρακάτω έννοιες:
  - True Positive-TP: Δεδομένα που ταξινομούνται σωστά στην κατηγορία Positive.
  - True Negative-TN: Δεδομένα που ταξινομούνται σωστά στην κατηγορία Negative.
  - False Positive-FP: Δεδομένα που ταξινομούνται λάθος στην κατηγορία Positive.
  - False Negative-FN: Δεδομένα που ταξινομούνται λάθος στην κατηγορία Negative.
 Έτσι ορίζουμε τις ενδιάμεσες μετρικές Ανάκλησης (Recall) και Ακρίβειας (Precision):

$$Recall = \frac{TP}{TP + FN} \quad \text{and} \quad Precision = \frac{TP}{TP + FP} \quad (3.13)$$

Συνεπώς έχουμε την τελική τιμή της F1-score:

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \quad (3.14)$$

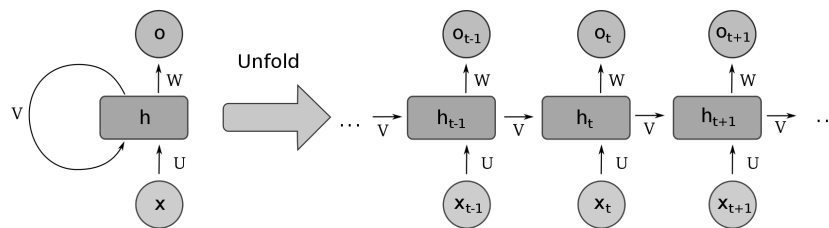
- **Pearson correlation coefficient** [42]: Δεν αποτελεί μία από τις πιο διαδεδομένες μετρικές (πχ δεν υπάρχει στο keras, αλλά υπάρχει στο tensorflow), αλλά την αναφέρουμε μιας και αποτελεί την μετρική που χρησιμοποιήσαμε για την τελική αξιολόγηση των μοντέλων μας. Υπολογίζει την γραμμική συσχέτιση δύο διανυσμάτων (Σχέση 3.15):

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (3.15)$$

Επίσης, συχνά χρησιμοποιούνται οι **Mean Absolute Error (MAE)** και **Mean Squared Error (MSE)** που ακολουθούν την λογική των Αντίστοιχων Συναρτήσεων Κόστους που αναλύσαμε πριν.

### 3.3.3 Επαναλαμβανόμενα Νευρωνικά Δίκτυα - Recurrent Neural Networks

Τα δίκτυα που είδαμε μέχρι τώρα, δεν έχουν συνδέσεις επανατροφοδοσίας (feedback), δηλαδή συνδέσεις μέσω βαρών που ξεκινούν από την έξοδο ενός στρώματος και καταλήγουν στην είσοδο του ίδιου ή ενός άλλου προηγούμενου στρώματος. Αυτή είναι ειδική τάξη δικτύων που ονομάζονται non recurrent ή feedforward δίκτυα. Γενικά, δίκτυα που περιλαμβάνουν feedback συνδέσεις λέγεται ότι επαναλαμβάνονται. Τα nonrecurrent δίκτυα δεν έχουν μνήμη έτσι η έξοδος τους καθορίζεται πάντα από την παρούσα είσοδο και από τις τιμές των βαρών. Σε μερικές διατάξεις δικτύων τα recurrent δίκτυα επανατροφοδοτούν με προηγούμενες εξόδους την είσοδο, έτσι η έξοδος τους καθορίζεται και από τις αντίστοιχες εισόδους και από τις προηγούμενες εξόδους. Για αυτό τον λόγο τα recurrent δίκτυα μπορούν να παρουσιάσουν ιδιότητες παρόμοιες με αυτές της μικρόχρονης ανθρώπινης μνήμης, έτσι η κατάσταση των εξόδων του δικτύου εξαρτάται κατά ένα μέρος από τις προηγούμενες τους εισόδους.



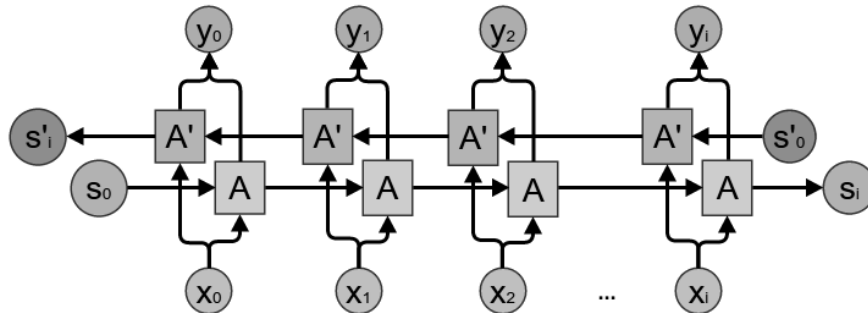
Σχήμα 3.10: Επαναλαμβανόμενο Νευρωνικό Δίκτυο (RNN)

Στο Σχήμα 3.10 βλέπουμε ένα Απλό Επαναλαμβανόμενο Δίκτυο, όπου κάθε νευρώνας πέρα από την είσοδο  $x_i$  που δέχεται, δέχεται και μία επιπλέον  $h_{i-1}$  από τον προηγούμενο νευρώνα. Συνεπώς, πέρα από τα βάρη  $W_{xi}, W_{yi}$  των υπερπαραμέτρων  $\theta$  που είχαμε να υπολογίσουμε στα απλά νευρωνικά δίκτυα, τώρα επιπλέον έχουμε και τα βάρη  $U_i$  που πρέπει να υπολογίσουμε, με  $\sigma_x, \sigma_y$  ως Συναρτήσεις Ενεργοποίησης (Εξισώσεις 3.16).

$$\begin{aligned} h_t &= \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \\ y_t &= \sigma_y(W_y h_t + b_y) \end{aligned} \quad (3.16)$$

Πέρα βέβαια από την ενημέρωση των Νευρώνων που προηγούνται στους επόμενους έχουμε και την περίπτωση όπου οι Νευρώνες που έπονται ενημερώνουν τους προηγούμενους τους. Τα Επαναλαμβανόμενα αυτά νευρωνικά ονομάζονται **Διπλής Κατεύθυνσης (Bidirectional)** [43]. Έχουν σχεδιαστεί ώστε να αυξάνουν την ποσότητα της εισαγόμενης πληροφορίας σε ένα δίκτυο. Στην ουσία είναι να έχουμε συνδέσει δύο Επα-

ναλαμβανόμενα Νευρωνικά Δίκτυα του προηγούμενου παραδείγματος ένα με κατεύθυνση ενημέρωσης από τα αριστερά προς τα δεξιά και ένα με κατεύθυνση ενημέρωσης αντρίστροφα της πρώτης. Στο Σχήμα 3.11 βλέπουμε μία τέτοια αρχιτεκτονική.

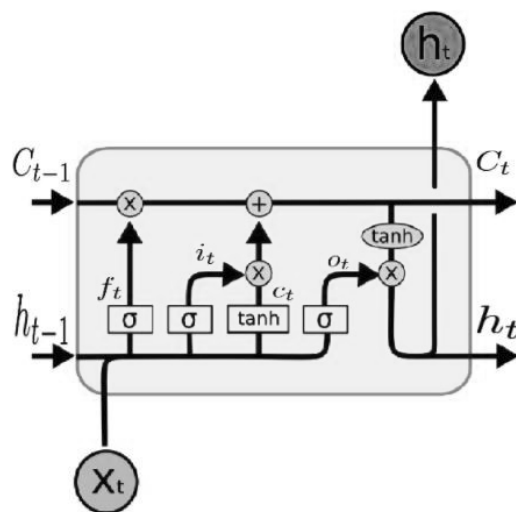


Σχήμα 3.11: Επαναλαμβανόμενο Νευρωνικό Δίκτυο Διπλής Κατεύθυνσης (BRNN)

### 3.3.3.1 Μακρά Βραχυπρόθεσμη Μνήμη (LSTM)

Όπως αναπτύχθηκε από τον Bengio το 1994 [44], τα RNNs αν και θεωρητικά μπορούν να διαχειριστούν μακροπρόθεσμες συσχετίσεις, στην πράξη έχουν παρατηρηθεί δυσκολίες. Αυτές οι δυσκολίες στηρίζονται στο φαινόμενο που αναφέραμε και πριν ως Εξαφανιζόμενη Κλίση (Vanishing gradient). Για τον λόγο αυτό δημιουργήθηκαν οι μονάδες **Μακράς Βραχυπρόθεσμης Μνήμης (LSTMs)**, οι οποίες προτάθηκαν από τους Hochreiter, Schmidhuber το 1997 [45], που δεν αντιμετωπίζουν αυτό το φαινόμενο.

Όπως αναφέραμε και πριν για τα BRNN έτσι και εδώ έχουμε αντίστοιχα μοντέλα Μακράς Βραχυπρόθεσμης Μνήμης Διπλής Κατεύθυνσης BiLSTM, τα οποία είναι ιδιαίτερος δημοφιλή στην Ανάλυση Κειμένου και χρησιμοποιήθηκαν στην Αρχιτεκτονική μας.



Σχήμα 3.12: Μονάδα Μακράς Βραχυπρόθεσμης Μνήμης (LSTM)

Είναι σχεδιασμένες για ακριβώς αυτόν τον σκοπό, δηλαδή να μπορούν αν διαχειριστούν

συσχετίσεις απομακρυσμένων δεδομένων με ιδιαίτερη ευκολία. Σε αντίθεση με τα απλά RNN δεν εφαρμόζει συνάρτηση ενεργοποίησης στις αναδρομικές συνδέσεις εξασφαλίζοντας ότι οι ενημερώσεις θα είναι γραμμικές και ότι τα σφάλματα δεν θα εξαφανίζονται από την επαναληπτική εφαρμογή των ενημερώσεων. Η αρχιτεκτονική της μονάδας αυτής φαίνεται στο Σχήμα 3.12. Οι τιμές των παραμέτρων της μονάδας του LSTM υπολογίζονται σύμφωνα με τους τύπους 3.17.

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 c_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
 h_t &= o_t * \tanh(C_t) \\
 C_t &= f_t * C_{t-1} + i_t * C_t
 \end{aligned} \tag{3.17}$$

Η αρχιτεκτονική που περιγράψαμε αποτελεί μία από τις πιο συνηθισμένες και αποτελείται από 3 εξομαλυντές που τους αποκαλούμε θύρες, την θύρα εισόδου (input gate), την θύρα εξόδου (output gate) και την θύρα λήθης (forget gate), όπου στο σχήμα μας συμβολίζονται με τα γράμματα  $i$ ,  $o$  και  $f$  αντίστοιχα. Κάποιες εναλλακτικές αρχιτεκτονικές δεν χρησιμοποιούν όλες τις πύλες όπως για παράδειγμα η Gated Recurrent Unit (GRU) [46] που δεν διαθέτει πύλη εξόδου.

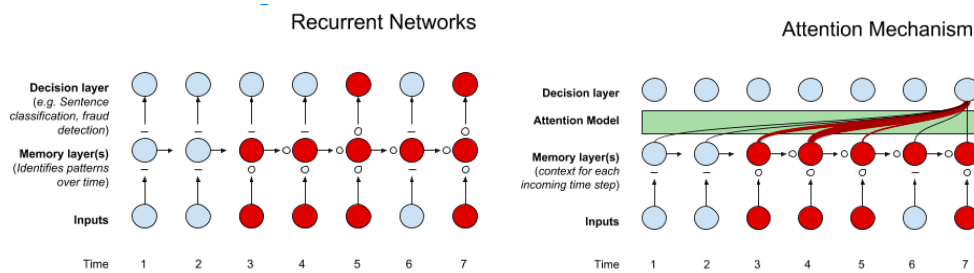
### 3.3.3.2 Μηχανισμός Προσοχής (Attention Mechanism)

Όπως είδαμε προηγουμένως, ένα RNN χρησιμοποιεί την τελευταία τιμή της εσωτερικής του κατάστασης ως την διανυσματική αναπαράσταση όλης της ακολουθίας. Η αναπαράσταση αυτή ενημερώνεται καθώς το δίκτυο επεξεργάζεται την ακολουθία και στο τέλος περιέχει μία σύνοψη/περίληψη (summary) της ακολουθίας. Όμως, ειδικά όταν η ακολουθία είναι αρκετά μεγάλη, υπάρχει η περίπτωση το δίκτυο να μην μπορεί να συγκρατήσει όλες τις σημαντικές πληροφορίες στην εσωτερική του κατάσταση. Για να αντιμετωπίσουμε αυτό το πρόβλημα, χρησιμοποιούμε μία τεχνική που ονομάζεται **Μηχανισμός Προσοχής (Attention Mechanism)**, η οποία επιχειρεί να ενισχύσει την την προσοχή που δίνουμε στις σημαντικές παραμέτρους του δικτύου.

Αυτό το πετυχαίνει αξιοποιώντας όλες τις ενδιάμεσες καταστάσεις του RNN. Έτσι κάθε χρονική στιγμή  $t$ , η εσωτερική κατάσταση του RNN  $y_i$ , χρησιμοποιείται ως την ερμηνεία της λέξης  $x_i$ . Για την παραγωγή της διανυσματικής αναπαράστασης ολόκληρου του κειμένου, χρησιμοποιούμε το άθροισμα των ερμηνειών των λέξεων, σταθμισμένο ως προς την σημαντικότητά τους.

Ο μηχανισμός προσοχής, αποτελεί ένα επίπεδο του δικτύου, το οποίο εκπαιδεύεται μαζί με τα υπόλοιπα και μαθαίνει να αποδίδει μεγαλύτερη βαρύτητα στις σημαντικές λέξεις, για το επίμαχο πρόβλημα. Έτσι, όταν η ακολουθία είναι ένα κείμενο, αυτό σημαίνει ότι

ο μηχανισμός δίνει μεγαλύτερη προσοχή στις πιο σημαντικές λέξεις του κειμένου. Στο πλαίσιο της Συναισθηματικής Ανάλυσης, μαθαίνει να αποδίδει μεγαλύτερα βάρη στις λέξεις οι οποίες είναι καθοριστικές για τον προσδιορισμό του συναισθήματος σε ένα κείμενο. Στο Σχήμα που ακολουθεί 3.13 παρατηρούμε πως λειτουργεί ο Μηχανισμός Προσοχής σε θέμα λογικής.



Σχήμα 3.13: Απλό RNN και RNN με Attention Mechanism

### 3.4 Επεξεργασία Φυσικής Γλώσσας

Η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP), αποτελεί υποκλάδο της Επιστήμης των Υπολογιστών και της Τεχνικής Νοημοσύνης, όντας ένας από τους μεγαλύτερους ερευνητικούς κλάδους σε τεχνικές Μηχανικής και Βαθείας Μάθησης. Αν και υπάρχει προγενέστερη δουλειά, η πρώτη αξιόπαινη προσπάθεια γίνεται από τον μεγάλο θεωρητικό της Επιστήμης των Υπολογιστών Alan Turing το 1950 [47], όπου προτείνει για πρώτη φορά, όπως αποκαλείται σήμερα, το Turing Test, ένα πείραμα όπου ένας Άνθρωπος αλληλεπιδρά με μία οντότητα και καλείται να αναγνωρίσει εάν πρόκειται για Άνθρωπο ή για Μηχανή. Στην συνέχεια δημιουργήθηκε το SHRDLU από τον Terry Winograd το 1968 [48], όπου ένας υπολογιστής μπορούσε να συζητήσει με έναν άνθρωπο όσον αφορά ένα πλαίσιο Σχημάτων και Συλλογών που είχε δημιουργηθεί σε θεωρητικό περιβάλλον και στην συνέχεια υλοποιήθηκε και σε εικονικό. Προηγήθηκε, βέβαια, το ELIZA το 1964 [49], ένα υπολογιστικό πρόγραμμα που προσωμείωνε έναν Ψυχολόγο Ατομοκεντρικής Ψυχοθεραπείας που αν και δεν είχε καμία πληροφορία ανθρώπινης σκέψης ή συναισθήματος κατάφερνε να πραγματοποιεί αληθοφανείς συζητήσεις.

Τα πεδία έρευνας που επικεντρώνεται το επιστημονικό ενδιαφέρον κατά κύριο λόγο χωρίζονται στις παρακάτω ομάδες:

- **Σύνταξη (Syntax):** Όπου αναφερόμαστε στην συντακτική ανάλυση κειμένου όπως η διάσπαση προτάσεων και λέξων και η εξαγωγή ορολογιών.
- **Σημασιολογία (Semantics):** Για την εξαγωγή πληροφορίας από το κείμενο, κλάδος ο οποίος περιέχει την μετάφραση κειμένου, συναισθηματική ανάλυση (όπου είναι και ο τομέας της εργασίας μας), αυτόματη εξαγωγή θεμάτων, κτλ

- **Λόγος (Discourse):** Ίσως ο ελληνικός τίτλος να μην αποδίδει πλήρως την έννοια, αναφερόμαστε όμως στην λογική επεξεργασία του κειμένου με τομείς όπως η εξαγωγή περιλήψης αλλά και η αναγνώριση ερωτήσεων, δηλώσεων, ισχυρισμών, κτλ
- **Ομιλία (Speech):** Που αφορά την ανάλυση της ομιλίας όπως η διάσπαση προφορικών προτάσεων και η μετατροπή προφορικού λόγου σε γραπτό.

Φυσικά υπάρχουν παραδείγματα που αφορούν τον κλάδο NLP όμως δεν συμπεριλαμβάνονται σε καμία από τις παραπάνω κατηγορίες όπως για παράδειγμα το “1 The Road”<sup>1</sup> που είναι ένα μυθιστόρημα 60 εκατομμυρίων λέξεων που έχει γραφτεί από ένα νευρωνικό δίκτυο.

Η Φυσική Γλώσσα είτε προφορική είτε γραπτή, περιέχει συνήθως πληροφορία που μπορεί να μην χρειαζόμαστε ή πληροφορία που δεν είναι ιδιαίτερος σημαντική και απλώς θα περιπλέξει τη διαδικασία. Στόχος είναι να απλοποιήσουμε το εκάστοτε κείμενο ούτως ώστε να ελαχιστοποιήσουμε οποιουδήποτε θορύβους αλλά κυρίως ούτως ώστε να έχουμε όσο το δυνατόν μεγαλύτερη συνέπεια και συνοχή ανάμεσα στα διαφορετικά δεδομένα. Για τον λόγο αυτό έχουν αναπτυχθεί διάφορες μέθοδοι προ-επεξεργασίας του κειμένου. Η διαδικασία αυτή αποκαλείται **Προεπεξεργασία Κειμένου (Text Pre-processing)** και στη συνέχεια αναλύουμε ορισμένες από τις πιο χαρακτηριστικές της μεθόδους:

- **Ομαδοποίηση Λέξεων (Tokenization):** Ένα κείμενο αποτελείται από μια ακολουθία συνεχόμενων χαρακτήρων, την οποία πρέπει να μετατρέψουμε σε ακολουθία λέξεων, σημείων στίξης, αριθμών, κτλ (tokens). Η διαδικασία αυτή μπορεί να φαντάζει εύκολη σε γλώσσες όπως τα Ελληνικά και τα Αγγλικά όπου οι λέξεις χωρίζονται με κενό, όμως υπάρχουν και πιο δύσκολες περιπτώσεις όπως για παράδειγμα τα Βιετναμέζικα όπου οι συλλαβές χωρίζονται με κενό. Επίσης πρέπει να συνυπολογιστούν περιπτώσεις όπως πχ η φράση “*Τρίτη-Τέταρτη*” που δεν έχει κενό αλλά είναι στην ουσία δύο λέξεις αλλά και αντρίστροφα η φράση “*Νέος Κόσμος*” που είναι δύο λέξεις αλλά πρέπει να μην διαχωριστεί. Για τον λόγο αυτό η συγκεκριμένη διαδικασία μπορεί να φαντάζει εύκολη αλλά περιέχει παγίδες και είναι απαραίτητη για την εκτέλεση των υπόλοιπων που ακολουθούν.
- **Αναγνώριση Μερών του Λόγου (Part-of-Speech Tagging):** Αφορά την αντιστοίχιση κάθε όρου με ένα μέρος του λόγου (πχ ρήμα, επίθετο, ουσιαστικό, κτλ). Οι κατηγορίες στις οποίες μπορεί να ενταχθεί ένας όρος σαφώς εξαρτώνται και από τις γειτονικές του λέξεις, ενώ ανάλογα με την αναλυτικότητα της μεθόδου που έχουμε επιλέξει, αυτές οι επιλογές μπορεί να είναι απλές όπως αυτές που προαναφέραμε ως παραδείγματα, αλλά και πιο σύνθετες πχ τοπική αντωνυμία, χρονικό επίρρημα, κτλ.
- **Αποκατάληξη (Stemming):** Αφορά την διαδικασία κατά την οποία αφαιρούμε τις καταλήξεις των λέξεων (stem). Ο όρος που μας απομένει δεν είναι απαραίτητα

<sup>1</sup><https://singularityhub.com/1-the-road>



υπάρχουσα λέξη, βοηθά όμως στο να έχουμε μια καλή αναπαράσταση (mapping) ίδιων στην ουσία λέξεων σε έναν κοινό όρο. Ένας από τους πιο γνωστούς αλγόριθμους Stemming στα αγγλικά είναι αυτός του Porter το 1980 [50] όπου για παράδειγμα οι λέξεις *rejection* και *rejected* αντιστοιχούνται στον όρο *reject*. Η Python διαθέτει μάλιστα βιβλιοθήκη για stemming στα Ελληνικά<sup>2</sup>, η οποία όμως δεν αποτελεί την μόνη επιλογή μιας και υπάρχουν πλήθος εναλλακτικών.

- **Λημματοποίηση (Lemmatization):** Σε αντίθεση με την αποκατάληξη στόχος αυτής της μεθόδου είναι η αντικατάσταση κάθε όρου με την ρίζα του. Σε αυτήν την περίπτωση κάθε αντικατάσταση οδηγεί σε λέξη που είναι υπαρκτή. Το πιο κλασικό παράδειγμα αυτής της μεθόδου είναι η μετατροπή του όρου “*am*” στον όρο “*be*”. Συχνά για να έχουμε καλύτερα αποτελέσματα αυτής της μεθόδου είναι αναγκαίο να προηγηθεί η μέθοδος Αναγνώρισης Μερών του Λόγου.
- **Εντοπισμός Τερματικών Όρων (StopWords):** Στην συγκεκριμένη μέθοδο αφαιρούμε εκείνες τις λέξεις οι οποίες δεν προσδίδουν κάποια επιπλέον πληροφορία στο κείμενο. Τέτοιες λέξεις είναι συνήθως οι “*the, as, it, etc*”. Η μέθοδος η οποία ακολουθείται είναι χρησιμοποιώντας ένα λεξικό που διαθέτει όλες αυτές τις λέξεις και αφαιρώντας τες, διατρέχοντας το κείμενο.

## 3.5 Τεχνικές Μετατροπής Κειμένου

Όπως αναφέραμε πριν, κατα κύριο λόγο η είσοδος στα νευρωνικά δίκτυα γίνεται με την μορφή διανυσμάτων (feature vectors). Το κείμενο όπως καθαυτό, ακόμα και μετά την προεπεξεργασία που έχει υποστεί, δεν είναι σε τέτοια μορφή. Για τον λόγο αυτό έχουν αναπτυχθεί τεχνικές οι οποίες μας βοηθούν να επεξεργαστούμε το κείμενο κατάλληλα ούτως ώστε να το φέρουμε σε μορφή αποδεκτή για το νευρωνικό μας. Κάθε προεπεξεργασία επιφέρει και διαφορετικά αποτελέσματα για αυτό πρέπει κάθε φορά να επιλέγουμε τον συνδυασμό εκείνων των τεχνικών που ταιριάζουν καλύτερα στην φύση του προβλήματος μας. Στην συνέχεια παρουσιάζουμε ορισμένες από τις πιο γνωστές μεθόδους μετατροπής κειμένου, δίνοντας περισσότερη έμφαση σε αυτές που χρησιμοποιήθηκαν από εμάς.

### 3.5.1 Bag of Words

Εμφάνιση του όρου **Bag of Words** έχουμε από το 1954 [51]. Αποτελεί στην ουσία την πιο απλή μορφή αναπαράστασης κειμένου. Κάθε κείμενο αναπαρίσταται ως το σύνολο των λέξεων που το αποτελούν. Αφού συγκεντρώσουμε όλες τις λέξεις που εμφανίζονται στο σύνολο των κειμένων, φτιάχνουμε ένα λεξικό που είναι ένας πίνακας αντιστοιχίας κατά βάση, αφού αντιστοιχίζουμε κάθε όρο σε ένα μοναδικό αναγνωριστικό (id). Για να εισαχθούν στον αλγόριθμο και να μπορούν να γίνουν συγκρίσεις πρέπει κάθε είσοδος να

<sup>2</sup><https://pypi.org/project/greek-stemmer/>



έχει τις ίδιες διαστάσεις διανυσμάτων με τις υπόλοιπες. Έτσι δημιουργούμε ένα διάνυσμα το οποίο έχει το μήκος του λεξικού που κατασκευάσαμε. Ορισμένοι τρόποι συμπλήρωσής του είναι οι παρακάτω:

- **Δυαδική Αναπαράσταση:** Σημειώνουμε με άσσο (1) εάν η λέξη υπάρχει στο εκάστοτε κείμενο ή με μηδεν (0) εάν όχι.
- **Πλήθος Εμφανίσεων:** Για κάθε λέξη σημειώνουμε το πλήθος εμφανίσεών της.
- **Συχνότητα Εμφανίσεων:** Για κάθε λέξη σημειώνουμε το ποσοστό εμφανίσεών της. Είναι στην ουσία η ίδια κατηγορία με την προηγούμενη, με κάθε όρο διαιρούμενο με το μήκος  $N$ .
- **TF-IDF:** Πρόκειται για το γινόμενο της Συχνότητας Εμφάνισης (Term Frequency) [52] ενός όρου σε ένα κείμενο με την Αντίστροφη Εμφάνιση Όρου (Inverse Document Frequency) [53] που είναι ο λογάριθμος του πλήθους των κειμένων διά το πλήθος των κειμένων στα οποία εμφανίζεται ο όρος. Στην ουσία πρόκειται για μία μετρική που υπολογίζει σημαντικότητα εμφάνισης.

### 3.5.2 Μοντέλα $n$ -γραμμάτων:

Αποτελεί μία γενίκευση του Bag of Words. Ανάλογα με την τιμή του  $n$  δημιουργούμε ομάδες γειτονικών λέξεων πλήθους  $n$ . Για  $n = 1$  έχουμε τα λεγόμενα unigrams, για  $n = 2$  τα bi-grams,  $n = 3$  τα tri-grams, κοκ. Όπως είναι φανερό, όσο περισσότερο αυξάνουμε το  $n$  τόσο αυξάνεται και το μέγεθος της εισόδου, αυξάνοντας την πολυπλοκότητα. Ακολουθεί παράδειγμα για  $n = 2$ :

Text: ['you', 'are', 'very', 'brave'],

bi-gram: [ ['you', 'are'], ['are', 'very'], ['very', 'brave'] ]

### 3.5.3 Λεξικά

Στην συγκεκριμένη μέθοδο χρησιμοποιούνται λεξικά τα οποία αποδίδουν μία τιμή σε κάθε όρο. Η τιμή αυτή μπορεί να αφορά αν ένας όρος είναι θετικός, ουδέτερος ή αρνητικός αλλά και το ποσοστό αυτών των 3 κλάσεων, ενώ πλέον υπάρχουν και λεξικά που έχουν πιο συγκεκριμένη πληροφορία που αφορά το Συναισθημα (Sentiment Lexicons). Όπως καταλαβαίνουμε, ιδιαίτερα στην περίπτωση της Συναισθηματικής Ανάλυσης αυτή η κατηγορία είναι πολύ χρήσιμη, όμως δεν είναι αρκετή αφού δεν μας δίνει την πλήρη πληροφορία ενός κειμένου. Αυτό σημαίνει πως πρέπει να συνδυαστεί και με άλλες μεθόδους προκειμένου να έχουμε ικανοποιητικά αποτελέσματα, ενώ συνήθως χρησιμοποιούμε περισσότερα από ένα διαφορετικά λεξικά. Τα λεξικά αυτά μπορούν να δημιουργηθούν χειροκίνητα δηλαδή η τιμές να υπολογίζονται από ανθρώπους, είτε αυτόματα χρησιμοποιώντας λέξεις που επεκτείνουν ήδη υπάρχον λεξικό. Η συγκεκριμένη μέθοδος σε αντίθεση με την Bag of Words για παράδειγμα, διαφέρει από γλώσσα σε γλώσσα, με την έννοια ότι χρειαζόμαστε διαφορετικό

λεξικό για κάθε διαφορετική γλώσσα. Τα λεξικά αυτά επικεντρώνονται συνήθως σε επίθετα, ρήματα και αντωνυμίες οι οποίες εξάγονται από το εκάστοτε προς ανάλυση κείμενο, επισημαίνονται με την αντίστοιχη τιμή και οι τιμές αυτές στο τέλος προστίθενται δίνοντάς μας το τελική τιμή για το συγκεκριμένο κείμενο. Η παραπάνω διαδικασία, βασίζεται στην ακόλουθη παραδοχή (Osgood, 1957) [54]:

“ Οι μεμονωμένες λέξεις έχουν αυτό που αναφέρεται ως «πρότερη πολικότητα» (*prior polarity*), δηλαδή συναισθηματική κλίση που είναι ανεξάρτητη του κειμένου. ”

Αυτή η συναισθηματική κλίση, μπορεί να εκφραστεί ως αριθμητική τιμή. Ένα από τα παλαιότερα και πιο γνωστά Λεξικά για τα Αγγλικά είναι το WordNet [55]. Στα Κεφάλαιο 5 παρατίθενται περισσότερα παραδείγματα λεξικών, τα οποία και χρησιμοποιήθηκαν.

### 3.5.4 Διανύσματα Λέξεων (Word Embeddings)

Τα **Διανύσματα Λέξεων** αφορούν αναπαραστάσεις λέξεων σε διανύσματα πραγματικών αριθμών και λίγων διαστάσεων συνήθως μερικών εκατοντάδων (100-500), ενώ αποτελούν μέθοδο η οποία λαμβάνει άνθιση κυρίως την τελευταία δεκαετία. Η λογική τους έγκειται στην προσπάθεια παρόμοιες σημασιολογικά λέξεις να χαρτογραφούνται κοντά στον ίδιο διανυσματικό χώρο. Έτσι μία λέξη αναπαρίσταται πλέον όχι μονοδιάστατα αλλά ως συσχέτιση με άλλες λέξεις. Το θετικό με την μέθοδο αυτή είναι ότι επειδή τα διανύσματα έχουν δημιουργηθεί από εκπαίδευση πολύ μεγάλου dataset, έχουμε συσχετίσεις δεδομένων που μπορεί στα Δεδομένα Εκπαίδευσής μας να μην υπάρχουν. Τα ΔΛ έρχονται σε δύο διαφορετικά κατηγορίες, μία με την οποία οι λέξεις εκφράζονται ως διανύσματα λέξεων που συν-απαντώνται (*co-occurring words*), και σε μια άλλη στην οποία οι λέξεις εκφράζονται ως διανύσματα γλωσσικών πλαισίων στα οποία εμφανίζονται οι λέξεις, τεχνοτροπίες που αναλύονται από τον Lavelli το 2004 [56]. Η πρώτη προσπάθεια δημιουργίας ΔΛ ήρθε από τον Bengio το 2003 [57] με την παραγωγή γλωσσικών μοντέλων, όπου λόγω απουσίας υπολογιστικών πόρων δεν είχε την δυνατότητα εκπαίδευσης με ένα αρκετά μεγάλο λεξικό. Στην συνέχεια αναλύουμε δύο από τις πιο γνωστές μεθόδους ΔΛ την Word2Vec και την GloveVectors.

#### 3.5.4.1 Word2Vec

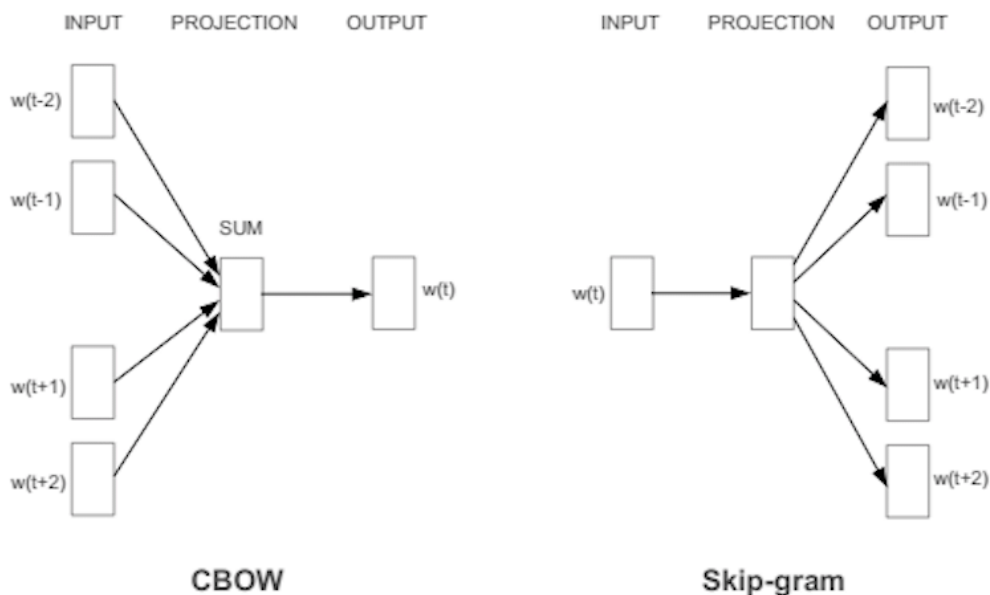
Μία ομάδα Ερευνητών της Google με υπεύθυνο τον T.Mikolon το 2013 παρουσιάζει την μέθοδο Word2Vec [58]. Χρησιμοποιεί ένα Ρηχό Νευρωνικό Δίκτυο (Shallow Neural Network), δηλαδή με μόνο ένα κρυφό επίπεδο (*hidden layer*). Με την τεχνική αυτή μπορούμε να βάλουμε το ΤΝΔ να εκπαιδευτεί για μία τυχαία εργασία και στο τέλος αντί να χρησιμοποιήσουμε το νευρωνικό για προβλέψεις, μπορούμε να πάρουμε τα βάρη κάθε λέξης του Κρυφού Στρώματος ως τα Διανύσματά μας. Σύμφωνα με τον Mikolon η μέθοδος Word2Vec φαίνεται να έχει πλεονέκτημα σε σχέση με προηγούμενες της, ιδιαίτερος σε

τομείς όπως η Λανθάνουσα Σημασιολογική Ανάλυση (Latent semantic analysis - LSA), μιας ειδικής τεχνοτροπίας του κλάδου NLP.

Η μέθοδος Word2vec μπορεί να χρησιμοποιήσει μία από τις ακόλουθες δύο αρχιτεκτονικές μοντέλων για να παράξει μια κατανομημένη αναπαράσταση των λέξεων:

- **continuous bag-of-words:** Στη συγκεκριμένη αρχιτεκτονική, το μοντέλο προβλέπει την τρέχουσα λέξη από ένα εύρος λέξεων που την περιβάλλουν. Η σειρά των λέξεων δεν επηρεάζει την πρόβλεψη (υπόθεση BoW).
- **continuous skip-gram:** Σε αυτήν την αρχιτεκτονική, το μοντέλο χρησιμοποιεί την τρέχουσα λέξη για να προβλέψει το γύρω εύρος των λέξεων περιβάλλοντος. Η αρχιτεκτονική αναθέτει μεγαλύτερα βάρη στις κοντινές λέξεις από τις πιο απόμακρες λέξεις του περιβάλλοντος.

Στο Σχήμα 3.14 που ακολουθεί βλέπουμε αυτές τις δύο Αρχιτεκτονικές.



Σχήμα 3.14: Αρχιτεκτονικές CBoW και CSG της μεθόδου word2vec

Με την CBoW αρχιτεκτονική προβλέπουμε την λέξη  $w(t)$  χρησιμοποιώντας ως γνώση τις προηγούμενες και επόμενες  $n$  λέξεις. Στο παράδειγμά μας στο Σχήμα 3.14 που προηγήθηκε, το  $n = 2$  ενώ συνηθίζεται να έχουμε  $n = 5$ . Αντίστροφα στην CSG Αρχιτεκτονική, έχοντας ως γνώση τη λέξη  $w(t)$  προβλέπουμε τις  $n$  προηγούμενες και επόμενες λέξεις. Ομοίως, στο παράδειγμά μας το  $n = 2$ , όμως συνηθίζεται να έχουμε  $n = 10$ .

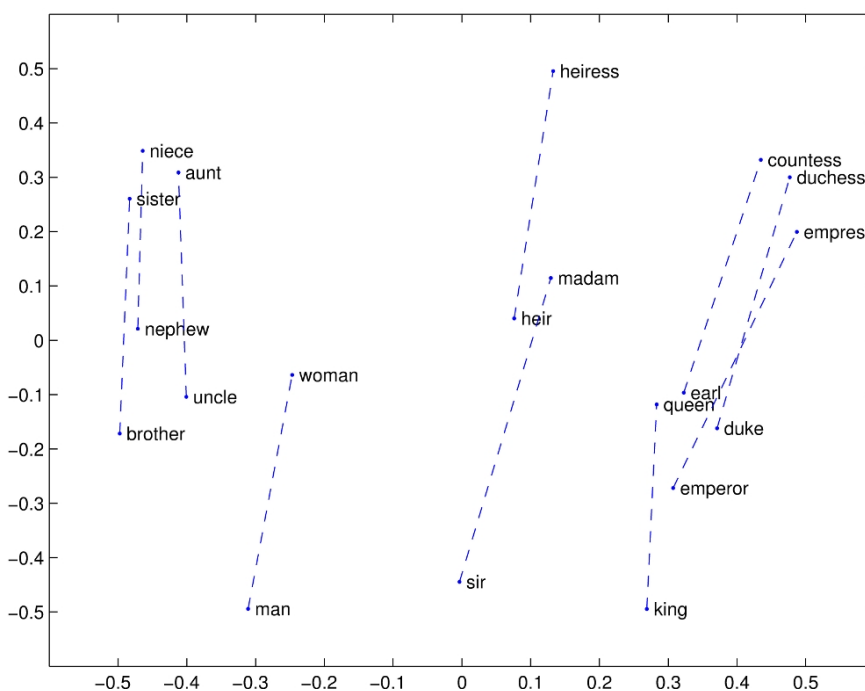
- **CBOW** = context  $\rightarrow$  Neural Network  $\rightarrow$  word
- **CSG** = word  $\rightarrow$  Neural Network  $\rightarrow$  context

### 3.5.4.2 Global Vectors (GloVe)

Το **GloVe**<sup>3</sup> παρουσιάστηκε από τον Pennington το 2014 [59], μια συλλογή από προ-εκπαιδευμένες αναπαραστάσεις λέξεων η οποία εκπαιδεύτηκε από ένα υπερμεγέθες corpus λέξεων. Αξίζει να σημειωθεί ότι οι αναπαραστάσεις λέξεων είναι από τις λίγες επιτυχημένες εφαρμογές της μη-εποπτευόμενης μάθησης. Το κύριο πλεονέκτημα τους είναι ότι δεν χρειάζονται τον ανθρώπινο παράγοντα όπως οι γράφοι αλλά μπορούν να εξαχθούν μέσα από μεγάλες συλλογές κειμένων όπως είναι το Wikipedia. Ο αλγόριθμος κωδικοποιεί τις λέξεις χρησιμοποιώντας διανυσματικές διαφορές ενώ όπως αναφέρει ο Pennington στην παρουσίαση του μοντέλου GloVe:

*“ Τα στατιστικά στοιχεία εμφάνισης λέξεων σε ένα κείμενο είναι η κύρια πηγή πληροφορίας διαθέσιμη για όλες τις Μη Εποπτευόμενες μεθόδους εκμάθησης Διανυσμάτων Λέξεων ”*

Λόγω των παραπάνω τα διανύσματα λέξεων παρουσίαζαν ενδιαφέρουσες γραμμικές υποδομές / συσχετίσεις (Linear Substructures). Όπως βλέπουμε και στο Σχήμα 3.15 τα γραμμές που ενώνουν τις λέξεις *man - woman* και *sir - madam* είναι σχεδόν παράλληλες αφού έχουν υπολογιστεί με βάση λεξιλογικές διαφορές.



Σχήμα 3.15: Γραφική Αναπαράσταση συγγενικών λέξεων μοντέλου GloVe

<sup>3</sup><https://nlp.stanford.edu/projects/glove/>

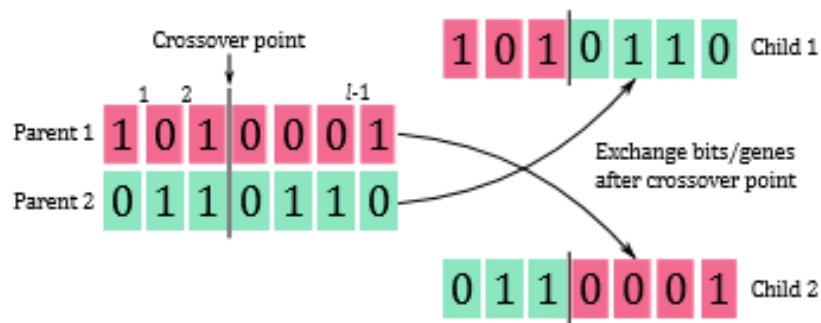
### 3.6 Γενετικοί Αλγόριθμοι

Οι **Γενετικοί Αλγόριθμοι (Genetic Algorithms - GA)** ανήκουν στο κλάδο της Επιστήμης Υπολογιστών και αποτελούν μια μέθοδο αναζήτησης βέλτιστων λύσεων σε συστήματα που μπορούν να περιγραφούν ως μαθηματικό πρόβλημα. Έγιναν γνωστοί από τον Holland [60] το 1975, ο οποίος βασίστηκε στην Δαρβινική Θεωρία της Εξέλιξης. Είναι χρήσιμοι σε προβλήματα που περιέχουν πολλές παραμέτρους/διαστάσεις και δεν υπάρχει αναλυτική μέθοδος που να μπορεί να βρει το βέλτιστο συνδυασμό τιμών για τις μεταβλητές ώστε το υπό εξέταση σύστημα να αντιδρά με όσο το δυνατόν πιο επιθυμητό τρόπο. Υπερτερεί παρόμοιων μεθόδων εύρεσης βέλτιστης λύσης όπως η Simplex η οποία εμπεριέχει την έννοια του περιορισμού (constraint) αλλά δεν μπορεί να βελτιστοποιήσει μη γραμμικές εξισώσεις.

Ο τρόπος λειτουργίας των Γενετικών Αλγορίθμων είναι εμπνευσμένος από τη βιολογία. Χρησιμοποιεί την ιδέα της εξέλιξης μέσω **γενετικής μετάλλαξης, φυσικής επιλογής και διασταύρωσης**. Οι Γενετικοί Αλγόριθμοι είναι αρκετά απλοί στην υλοποίησή τους. Οι τιμές για τις παραμέτρους του συστήματος πρέπει να κωδικοποιούνται με τρόπο ώστε να αναπαρασταθούν από μια μεταβλητή που περιέχει σειρά χαρακτήρων ή δυαδικών ψηφίων (0/1). Αυτή η μεταβλητή μιμείται το γενετικό κώδικα που υπάρχει στους ζωντανούς οργανισμούς. Ακολουθούν τα στάδια εκτέλεσης του αλγορίθμου:

- **Αρχικοποίηση Πληθυσμού (Initial Population):** Αρχικά ο Αλγόριθμος παράγει πολλαπλά αντίγραφα της μεταβλητής/γεννητικού κώδικα, συνήθως με τυχαίες τιμές, δημιουργώντας ένα πληθυσμό λύσεων. Κάθε πιθανή λύση ονομάζεται **Χρωμόσωμα (Chromosome)** και τα επιμέρους στοιχεία που την αποτελούν (συνήθως δυαδικές τιμές) ονομάζονται **Γονίδια (Genes)**. Ο πληθυσμός κάθε επανάληψης αποκαλείται **Γενιά (Generation)**.
- **Αξιολόγηση (Evaluation):** Στη συνέχεια κάθε λύση (τιμές για τις παραμέτρους του συστήματος) δοκιμάζεται για το πόσο κοντά φέρνει την αντίδραση του συστήματος στην επιθυμητή, μέσω μιας συνάρτησης που δίνει το μέτρο ικανότητας της λύσης και η οποία ονομάζεται **Συνάρτηση Ικανότητας (Fitness Function)**. Αν το κριτήριο που έχουμε θέσει έχει καλυφθεί ή φτάσαμε το όριο Γενεών ο Αλγόριθμος επιστρέφει την καλύτερη λύση.
- **Επιλογή (Selection):** Η λογική αυτού του βήματος είναι να επιλεγθούν εκείνα τα Χρωμοσώματα που έχουν καλύτερες επιδόσεις ώστε τα Γονίδιά τους να περάσουν στις επόμενες Γενιές. Χρωμοσώματα με υψηλό σκορ στην Συνάρτηση Ικανότητας έχουν μεγαλύτερη πιθανότητα να επιλεγθούν.
- **Διασταύρωση (Crossover):** Η Διασταύρωση αποτελεί το πιο σημαντικό βήμα του Αλγορίθμου. Τα Χρωμοσώματα χωρίζονται σε δυάδες που αποκαλούνται Γονείς και κρατώντας ένα πλήθος γονιδίων σταθερό, ανταλλάσσουν τα υπόλοιπα μεταξύ τους,

δημιουργώντας έτσι μία νέα Γενιά. Στο Σχήμα 7.1 βλέπουμε πιο σχηματικά αυτή τη διαδικασία.



Σχήμα 3.16: Διασταύρωση Δεδομένων Γενετικών Αλγορίθμων

- **Μετάλλαξη (Mutation):** Προκειμένου να έχουμε ανανέωση των Δεδομένων του Αλγορίθμου, πέρα από την Διασταύρωση, επιλέγουμε τυχαία με μικρή πιθανότητα ορισμένα από τα Χρωμοσώματα και τους αναστρέφουμε την τιμή ορισμένων γονιδίων. Έστω το χρωμόσωμα  $A_0 = 1|1|0|0|0$ . Τότε μετά την μετάλλαξη στα σημειωμένα γονίδια θα έχουμε το  $A'_0 = 1|1|0|1|0$

Ακολουθεί το Διάγραμμα Ροής το Αλγορίθμου:

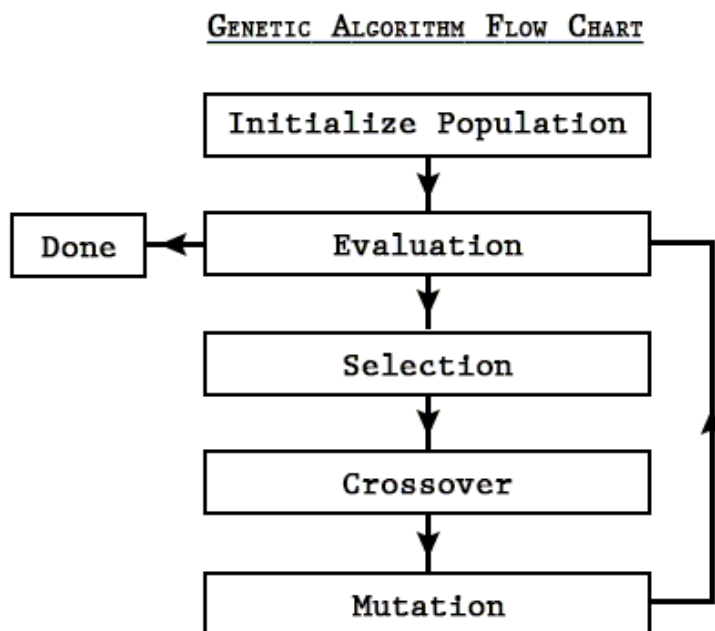


FIGURE 2

Σχήμα 3.17: Διάγραμμα Ροής Γενετικού Αλγορίθμου

## Κεφάλαιο 4

# Τεχνικό Υπόβαθρο

Για την υλοποίηση των νευρωνικών και της διαδικτυακής εφαρμογής χρησιμοποιήθηκαν οι προγραμματιστικές γλώσσες **Python**, **JavaScript**, **HTML** και **CSS**. Θεωρώντας ότι οι παραπάνω γλώσσες είναι γνωστές, παρακάτω αναλύουμε μόνο τα εργαλεία και Frameworks που χρησιμοποιήθηκαν πάνω σε αυτές καθώς και κάποιες έξτρα τεχνολογίες. Επίσης για την υλοποίηση των γενετικών αλγορίθμων χρησιμοποιήσαμε την βιβλιοθήκη **DEAP**<sup>1</sup> της Python

### 4.1 Τεχνολογίες Νευρωνικών Δικτύων

#### 4.1.1 Keras

Το **Keras**<sup>2</sup> είναι ένα API - βιβλιοθήκη υψηλού επιπέδου που ξεκίνησε να αναπτύσσεται το 2015, γραμμένο στην γλώσσα προγραμματισμού Python, το οποίο μπορεί να τρέξει επάνω στις βιβλιοθήκες Βαθιάς Μάθησης TensorFlow<sup>3</sup>, Microsoft Cognitive Toolkit<sup>4</sup> και Theano<sup>5</sup>. Το όνομά του προέρχεται από την Ελληνική Λέξη Κέρας = Κέρατο, ενώ κύριος δημιουργός αυτού του API είναι ο François Chollet [61], μηχανικός της Google. Είναι έτσι παραμετροποιημένο ούτως ώστε να μπορεί να τρέξει και σε CPU αλλά και σε GPU για πιο γρήγορα αποτελέσματα. Η σχεδίαση του έχει γίνει με σκοπό να είναι φιλικό προς το χρήστη στους πειραματισμούς του με Βαθιά Νευρωνικά Δίκτυα, να είναι εύκολα παραμετροποιήσιμο καθώς και να μπορεί εύκολα να διευρυνθεί για άλλους σκοπούς. Το Keras αποτελεί την βιβλιοθήκη που χρησιμοποιήσαμε για την υλοποίηση των νευρωνικών μοντέλων 3 και 4 του επόμενου κεφαλαίου.

<sup>1</sup><https://github.com/deap/deap>

<sup>2</sup><https://keras.io/>

<sup>3</sup><https://www.tensorflow.org/>

<sup>4</sup><https://docs.microsoft.com/en-us/cognitive-toolkit/>

<sup>5</sup><http://www.deeplearning.net/software/theano/>

### 4.1.2 Scikit-learn

Το **Scikit-learn** προτάθηκε το 2012 [62] ως μία δωρεάν βιβλιοθήκη της γλώσσας προγραμματισμού Python η οποία περιέχει υλοποιημένους πληθώρα αλγορίθμων Μηχανικής Μάθησης όπως για παράδειγμα οι Support Vector Machine, Random Forest, Gradient Boosting και k-means. Η συγκεκριμένη βιβλιοθήκη έχει αποτελέσει το εναρκτήριο λάκτισμα για το ευρύ κοινό στην ενασχόληση του με την Μηχανική Μάθηση. Είναι σχεδιασμένη ώστε να μπορεί εύκολα να αλληλεπιδρά με τις αριθμητικές και επιστημονικές βιβλιοθήκες NumPy <sup>6</sup> και SciPy <sup>7</sup>. Το θετικό με την συγκεκριμένη βιβλιοθήκη είναι πως υπάρχει συνοχή από αλγόριθμο σε αλγόριθμο σε θέμα σύνταξης κώδικα με αποτέλεσμα ο χρήστης να μπορεί εύκολα αφού χρησιμοποιήσει έναν αλγόριθμο να μεταπηδήσει σε έναν άλλον.

## 4.2 Τεχνολογίες Διαδικτυακής Εφαρμογής

### 4.2.1 REST API εφαρμογές

Το REpresentational State Transfer (REST) προτάθηκε το 2000 από τον Roy Thomas Fielding [63]. Είναι ένα στυλ αρχιτεκτονικής λογισμικού που αποτελείται από κατευθυντήριες γραμμές και πρακτικές για τη δημιουργία επεκτάσιμων υπηρεσιών διαδικτύου. Το REST είναι ένα σύνολο περιορισμών που εφαρμόζεται στη σχεδίαση των μερών σε ένα διαμοιρασμένο σύστημα υπερμέσων (υπερμέσα: graphics, audio, video, plain text και hyperlinks) που μπορεί να οδηγήσει σε μια πιο αποδοτική και εύκολα συντηρήσιμη αρχιτεκτονική. Το REST είναι πια ευρέως αποδεκτό σαν μια απλούστερη εναλλακτική αντί του SOAP [64]. Τα συστήματα RESTful επικοινωνούν συνήθως μέσω του πρωτοκόλλου HTTP με τα ίδια HTTP verbs (GET, POST, PUT, DELETE...) που χρησιμοποιούν και οι web browsers για να λάβουν σελίδες του διαδικτύου και να στείλουν δεδομένα σε απομακρυσμένους servers. Οι απαντήσεις του server σε κάθε call συνηθίζονται να στέλνονται σε μορφή HTML <sup>8</sup>, XML <sup>9</sup> και κυρίως σε μορφή JSON <sup>10</sup>. Ο παγκόσμιος ιστός αναπαριστά τη μεγαλύτερη υλοποίηση ενός συστήματος που συμμορφώνεται με το REST. Πλέον η παράλληλη ανάπτυξη της ίδια εφαρμογής σε διαφορετικά μέσα (πχ Android App, IOS App, Web App, κτλ) έχει οδηγήσει στην άνθιση του συγκεκριμένου πρωτοκόλλου μιάς και με μία διεπαφή backend μπορούμε να επικοινωνήσουμε με πολλαπλές διεπαφές frontend, δυνατότητα που δεν υποστήριζαν οι εφαρμογές υλοποιημένες με full-stack τεχνικές. Η διαδικτυακή εφαρμογή μας χρησιμοποιεί το παραπάνω πρωτόκολλο.

---

<sup>6</sup><https://www.numpy.org/>

<sup>7</sup><https://www.scipy.org/>

<sup>8</sup><https://www.w3.org/html/>

<sup>9</sup><https://www.w3.org/XML/>

<sup>10</sup><https://www.json.org/>



### 4.2.2 TwitterAPI

Για την live άντληση tweets από το Twitter χρησιμοποιήθηκε το **twitter-API** <sup>11</sup>. Πρόκειται για το API που έχει αναπτυχθεί από το ίδιο το Twitter ξεκινώντας από τα μέσα του 2016 δίνοντας την δυνατότητα στο ευρύ κοινό να έχει πρόσβαση σε μία μεγάλη γκάμα από δυνατότητες. Το API επιστρέφει τα δεδομένα του σε μορφή JSON και η δομή τους είναι προκαθορισμένη από το documentation του twitter. Στην περίπτωση μας ασχοληθήκαμε μόνο με την μέθοδο search που παρέχει το API, η οποία πέρα από την δημοσίευση καθαυτή επιστρέφει και μέτα-δεδομένα που αφορούν την τοποθεσία, το πλήθος αναδημοσιεύσεων, στοιχεία του χρήστη δημοσίευσης και πολλά άλλα, ενώ ακόμα δέχεται παραμέτρους περιορισμού τοποθεσίας, χρόνου, γλώσσας, πλήθους, κτλ. Το API επίσης διαθέτει μεθόδους που επιτρέπουν την άντληση των κορυφαίων τάσεων (trends) ανά περιοχή, το οποίο επίσης χρησιμοποιήσαμε, προκειμένου να συλλέξουμε τις κορυφαίες τάσεις.

### 4.2.3 Flask - Backend Framework

Το **Flask** <sup>12</sup> αποτελεί ένα backend micro-framework γραμμένο σε Python. Ξεκίνησε το 2011 από τον Armin Ronacher [65] ως ένα πρωταπριλιάτικο αστείο, υποσχόμενο ένα framework που δεν θα χρειάζεται επιπλέον εργαλεία και βιβλιοθήκες. Η ιδέα αυτή όμως προσέελκυσε ενδιαφέρον με αποτέλεσμα το αστείο να γίνει πραγματικότητα, δημιουργώντας έτσι ένα framework πολύ απλό στην χρήση, σε αντίθεση με άλλα αντίστοιχα. Το 2016 αποτελούσε το πιο δημοφιλές framework της Python στο github <sup>13</sup>, όντας πλέον συμβατό με το Google App Engine <sup>14</sup>, ενώ διαθέτει ενσωματωμένο development server και debugger. Πλέον χρησιμοποιείται και από κολοσσούς του διαδικτύου όπως το Pinterest <sup>15</sup> και το LinkedIn <sup>16</sup>. Ακολουθεί το 'Hello World!' πρόγραμμα για το Flask που αποδεικνύει την απλότητα του σε θέματα κώδικα.

---

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

---

<sup>11</sup><https://developer.twitter.com/en/docs.html>

<sup>12</sup><http://flask.pocoo.org/>

<sup>13</sup><https://github.com/>

<sup>14</sup><https://cloud.google.com/appengine/>

<sup>15</sup><https://www.pinterest.com/>

<sup>16</sup><https://www.linkedin.com/>

#### 4.2.4 React - Frontend Framework

Το **React** <sup>17</sup> (γνωστό και ως **React.js** ή **ReactJS**) είναι μία βιβλιοθήκη - framework της JavaScript. Δημιουργήθηκε το 2011 από τον Jordan Walke για σκοπούς του Facebook και έγινε δημόσιας διανομής (open-source) το 2013, αφού χρησιμοποιήθηκε με επιτυχία στο news feed του Facebook και στην ιστοσελίδα του Instagram <sup>18</sup>. Χρησιμοποιεί την τεχνική του Single Page Application <sup>19</sup>, η οποία αποτελεί την πιο αποδοτική τεχνική για γρήγορη ανανέωση δεδομένων σε μία ιστοσελίδα, ανανεώνοντας κάθε φορά μόνο τον component που απαιτείται, χωρίς να χρειάζεται η ανανέωση ολόκληρης της σελίδας, ενώ ταυτόχρονα έχει ως σημαντικό προνόμιο την χρήση του JSX (JavaScript XML) που είναι στην ουσία μία επέκταση της σύνταξης της JavaScript παρόμοια σε μορφή με αυτή της HTML. Χρησιμοποιείται κατά κύριο λόγο ως ο χριστής του "V" (View) στρώματος στο μοντέλο MVC (Model View Controller) [66]. Παράλληλα αναπτύχθηκε και το React Native όπου αποτελεί την πρόταση της React για υλοποίηση εφαρμογών Android και IOS. Για έρευνα του State Of JS <sup>20</sup> το React αποτελεί το πιο δημοφιλές frontend framework για JavaScript από το 2018 μέχρι και σήμερα.

#### 4.2.5 MongoDB - Βάση Δεδομένων

Η **MongoDB** <sup>21</sup> αποτελεί μία δωρεάν NoSQL, μη-σχεσιακή βάση δεδομένων, απαλλαγμένη από πίνακες και σχέσεις μεταξύ αυτών. Επιπλέον, πρόκειται για μία document-based βάση δεδομένων που προσφέρει τη δυνατότητα αποθήκευσης αρχείων εντός της. Τα αρχεία αυτά έχουν μορφή BSON, δηλαδή JSON σε δυαδική αναπαράσταση. Η απώλεια πινάκων ισοσκελίζεται με την ύπαρξη συλλογών (collections) από αρχεία, τα οποία όμως δεν χρειάζεται να είναι του ίδιου τύπου, όπως στις SQL βάσεις, αρκεί να είναι σε μορφή JSON. Διαθέτει μία μεγάλη γκάμα queries από απλή εύρεση αρχείου βάσει κλειδιού, μέχρι και αναζήτηση με regular expressions. Επίσης επιτρέπει την ευρετηρίαση (indexing) στα πεδία του, γεγονός που μπορεί να κάνει αναζητήσεις σαφώς πιο γρήγορες. Στα αρνητικά της βάσης συγκαταλέγονται ο χώρος που μπορεί να χρειάζονται τα λεξικά ευρετηρίασης για αποθήκευση και η δυσκολία δημιουργίας σχέσεων από συλλογή σε συλλογή μιας και το μόνο αντίστοιχο που διαθέτει η Mongo σε Ξένο Κλειδί (Foreign Key) είναι η αναφορά (Reference) σε αρχείο άλλης συλλογής. Η Mongo αποτελεί την πιο δημοφιλή NoSQL βάση μέχρι σήμερα.

---

<sup>17</sup><https://reactjs.org/>

<sup>18</sup><https://www.instagram.com/>

<sup>19</sup>[https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application)

<sup>20</sup><https://stateofjs.com/>

<sup>21</sup><https://www.mongodb.com/>

## Κεφάλαιο 5

# Αρχιτεκτονική

Στο συγκεκριμένο κεφάλαιο θα κάνουμε μία ανάλυση των αρχιτεκτονικών που χρησιμοποιήθηκαν για την πρόβλεψη Συναισθήματος σε αγγλικά tweets. Τα μοντέλα που ακολουθούν αποτελούν τόσο νευρωνικά δίκτυα όσο και υλοποιημένου αλγόριθμους Μηχανικής Μάθησης που χρησιμοποιήσαμε.

### 5.1 Υλοποιημένοι Αλγόριθμοι Μηχανικής Μάθησης

Αν και πρόκειται για ήδη υλοποιημένους αλγόριθμους θεωρούμε πως είναι σημαντικό να αναλύσουμε τις υπερπαραμέτρους που χρησιμοποιήσαμε στους αλγόριθμους αυτούς, αλλά και τη μέθοδο με την οποία αυτές βρέθηκαν. Στο Σχήμα 5.1 βλέπουμε τα στάδια εισόδου της πληροφορίας στον αλγόριθμο, τα οποία θα αναλυθούν ενδελεχώς στο επόμενο κεφάλαιο.

Το dataset που είχαμε στη διάθεσή μας για να εκπαιδύσουμε το δίκτυό μας δεν ήταν ιδιαίτερα μεγάλο. Συνεπώς, ήταν σχετικά ελάχιστα χρονοβόρο για εμάς να ψάξουμε με εξαντλητικό τρόπο ώστε να βρούμε τις υπερπαραμέτρους, την οποία μέθοδο και ακολουθήσαμε αντί για παράδειγμα να επιλέξουμε μία μέθοδο όπως η GridSearchV.

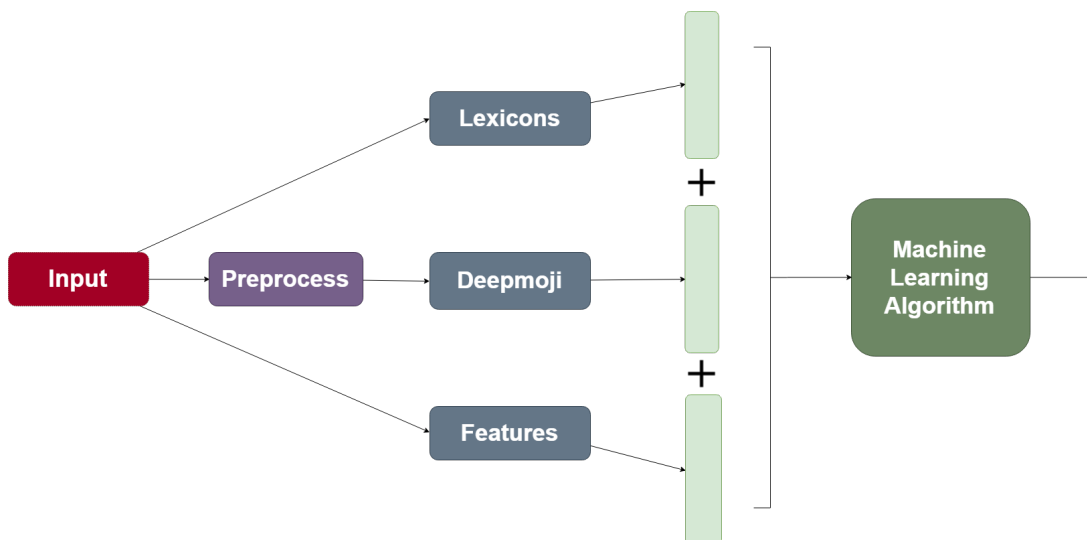
Έτσι για τους δύο αλγόριθμους που χρησιμοποιήσαμε, οι οποίοι έχουν υλοποιηθεί μέσω της βιβλιοθήκη ScikitLearn που αναλύσαμε στο Κεφάλαιο 4, έχουμε:

#### 1. Τυχαία Δάση - κλάση RandomForestRegressor:

- `n_estimators = 600`, πλήθος δέντρων.
- `bootstrap = true`, δηλαδή αν θα χρησιμοποιούνται bootstrap δείγματα κατά την δημιουργία των δέντρων
- `max_depth = 10`: Μέγιστο βάθος Δέντρων.
- `min_samples_leaf = 4`, ο ελάχιστος αριθμός δειγμάτων που πρέπει να υπάρχουν σε κάθε φύλο.
- `min_samples_split = 2`, ο ελάχιστος αριθμών δειγμάτων ώστε να διασπαστεί ένας εσωτερικός κόμβος

## 2. Μηχανές Διανυσμάτων Υποστήριξης - κλάση svm.SVR:

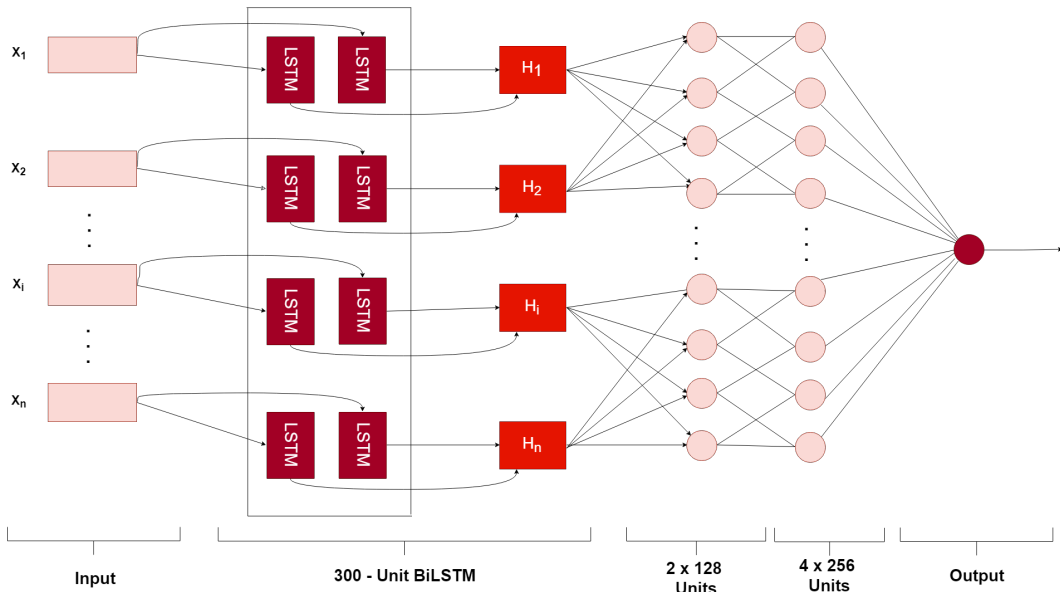
- kernel = 'rbf', είδος πυρήνα (Radial Basis Function).
- C = 10, παράμετρος σφάλματος.
- gamma = 0.0001, παράμετρος πυρήνα  $\gamma$ .
- epsilon = 0.05, όριο όπου το σφάλμα δεν συνδέεται με την συνάρτηση κόστους.



Σχήμα 5.1: Υλοποιημένοι Αλγόριθμοι Μηχανικής Μάθησης

## 5.2 Αμφίδρομο LSTM

Στα προηγούμενα κεφάλαια και πιο συγκεκριμένα στο Κεφάλαιο 3, είδαμε πως οι Μονάδες Μακράς Βραχυπρόθεσμης Μνήμης χρησιμοποιούνται ιδιαιτέρως συχνά σε θέματα που αφορούν Επεξεργασία Φυσικής Γλώσσας μιας και μπορούν να εκφράσουν εύκολα συσχετίσεις μεταξύ των διαφόρων εννοιών και λέξεων. Στο Σχήμα 5.2 βλέπουμε τα στάδια του συγκεκριμένου μοντέλου, ξεκινώντας με την είσοδο η οποία αποτελείται από προεκπαιδευμένα Διανύσματα Λέξεων (Word Embeddings) και τροφοδοτείται σε ένα Αμφίδρομο LSTM 300 μονάδων το οποίο συνδιάζεται και με ένα στρώμα Dropout της τάξης του 50%. Εσωτερικά του BiLSTM επιλέχθηκαν οι τιμές παραμέτρων (return\_sequences=True, dropout=0.25, recurrent\_dropout=0.25). Στην συνέχεια έχουμε δύο μονάδες 128 νευρώνων και 4 μονάδες 256 νευρώνων απλής διάταξης Dense, όλες από τις οποίες χρησιμοποιούν ως αρχικοποίηση των πυρήνων του την συνάρτηση gloriot\_normal [67] (γνωστή και ως Xavier), χωρίς χρήση seed. Ως συνάρτηση Βελτιστοποίησης χρησιμοποιήθηκε ο Adam Optimizer χωρίς έξτρα παραμετροποιήσεις, ενώ ως Συνάρτηση Κόστους χρησιμοποιήθηκε η Mean Squared Error (mse).

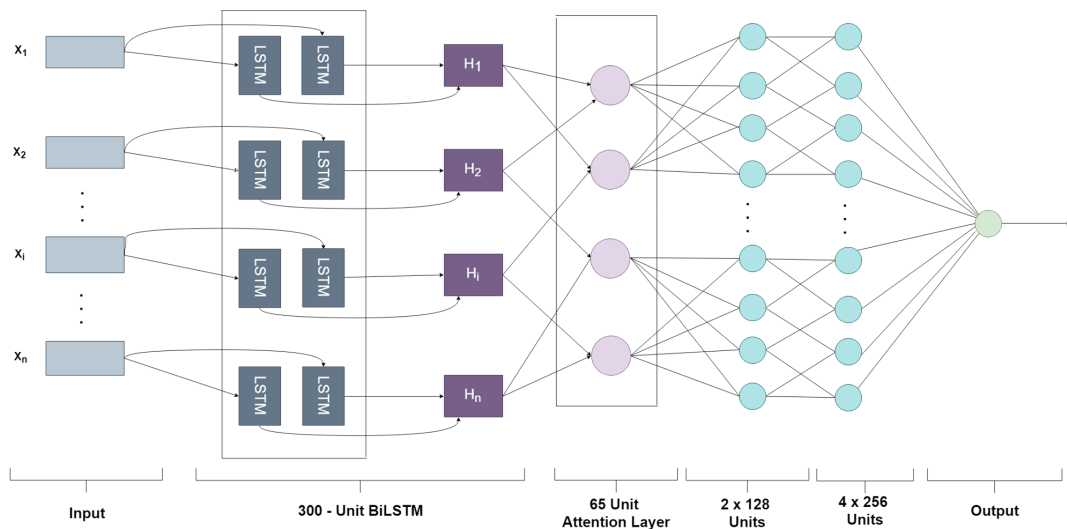


Σχήμα 5.2: Αρχιτεκτονική Αμφίδρομου LSTM

### 5.3 Αμφίδρομο LSTM με Μηχανισμό Προσοχής

Πέρα από τις Μονάδες Μακράς Βραχυπρόθεσμης Μνήμης στο Κεφάλαιο 3 αναφέραμε και την μέθοδο Μηχανισμού Προσοχής (Attention Mechanism) η οποία δίνει μεγαλύτερη προσοχή στους σημαντικούς όρους κάθε κειμένου. Βασιζόμενοι σε αυτή τη λογική υλοποιήσαμε μία αρχιτεκτονική ανάλογη του Σχήματος 5.2 η οποία παρατίθεται στο Σχήμα 5.3. Συγκεκριμένα, έχουμε τις ίδιες ακριβώς μονάδες με το προηγούμενο σχήμα με την διαφορά πως ανάμεσα στην Μονάδα BiLSTM και τους νευρώνες Dense προσθέσαμε ένα στρώμα Μηχανισμού Προσοχής 65 μονάδων. Η Συναρτήσεις Ενεργοποίησης και Βελτιστοποίησης είναι οι ίδιες όπως και πριν.

#### 5.3



Σχήμα 5.3: Αρχιτεκτονική Αμφίδρομου LSTM με Μηχανισμό Προσοχής

## 5.4 Γενετικοί Αλγόριθμοι

### 5.4.1 Συνδιασμός Μοντέλων

Σημαντικό βήμα στην Μηχανική Μάθηση είναι όπως αναφέραμε ο συνδιασμός μεθόδων. Σε τέτοιες περιπτώσεις συχνά χρησιμοποιούνται έξτρα Νευρωνικά όπου έχουν ως είσοδο όλες τις εξόδους των Νευρωνικών που έχουμε δημιουργήσει και ως τιμή πρόβλεψης την τιμή που θέλουμε να πετύχουμε. Στόχος αυτού του νευρωνικού είναι να βρει τον καλύτερο συνδιασμό μεταξύ αυτών των προβλέψεων. Μια άλλη μέθοδος είναι βρίσκοντας δυναμικά τα βάρη με τα οποία πρέπει να πολλαπλασιαστεί κάθε διαφορετικό διάνυσμα πρόβλεψης ώστε να έχουμε το καλύτερο δυνατό αποτέλεσμα. Αυτήν την λογική ακολουθούμε και εμείς, χρησιμοποιώντας όμως Γενετικούς Αλγορίθμους.

Δημιουργήσαμε, έτσι, μία συνάρτηση όπου μπορούμε να της περάσουμε όσες προβλέψεις θέλουμε πάνω σε ένα Task και τις πραγματικές τιμές αυτού του Task, επιστρέφοντας τα βάρη για κάθε διάνυσμα προβλέψεων.

Οι γενετικοί αλγόριθμοι δέχονται περιορισμό στις τιμές που μπορούν να πάρουν τα βάρη και για αυτό χρησιμοποιήσαμε τιμές στο εύρος  $[0,1]$ . Δεν δέχονται όμως περιορισμούς που αφορούν τις σχέσεις μεταξύ των βαρών. Εμείς στην περίπτωσή μας θα θέλαμε τα βάρη να αθροίζονται στο 1. Για να το καταφέρουμε αυτό, μιας και οι Γενετικοί Αλγόριθμοι επιτρέπουν την χρήση οποιασδήποτε Fitness Function εμείς επιθυμούμε, κάναμε μία κανονικοποίηση επάνω στα βάρη. Έτσι έχουμε για  $k$  διαφορετικές προβλέψεις:

$$final\_pred[X] = \sum_{i=1}^k pred_i(X) * weights(i) / sum(weights) \quad (5.1)$$

Στην συνέχεια υπολογίζαμε την τιμή Pearson Correlation για το διάνυσμα  $final\_pred$ .

Όσον αφορά τις εσωτερικές παραμέτρους του Γενετικού Αλγορίθμου επιλέξαμε έναν Πληθυσμό 200 χρωμοσωμάτων που αρχικοποιήθηκαν τυχαία. Ως Συνάρτηση Διασταύρωσης χρησιμοποιήσαμε την κλασική συνάρτηση two-point crossover της DEAP, που ονομάζεται `cxTwoPoint`, όπου βρίσκει 2 τυχαία γονίδια μεταξύ δυο χρωμοσωμάτων και τα ανταλλάσσει. Η πιθανότητα να επιλεγεί ένα χρωμόσωμα για Διασταύρωση τέθηκε στο 0.5. Ως Συνάρτηση Μετάλλαξη επιλέξαμε την `mutFlipBit` που αντιστρέφει την τιμή ενός γονιδίου κάθε χρωμοσώματος με παραμετροποιημένη πιθανότητα `mutFlipBit` την οποία θέσαμε στο 0.1. Η πιθανότητα να επιλεγεί ένα χρωμόσωμα για Μετάλλαξη τέθηκε στο 0.1 Τέλος ως Συνάρτηση Επιλογής διαλέξαμε την `selTournament` η οποία επιλέγει το καλύτερο από τυχαία δείγματα μεταβλητού πλήθους που θέσαμε στον αριθμό 3, επαναλαμβάνοντας την διαδικασία  $k$  φορές, όπου  $k$  το μέγεθος του Πληθυσμού.

Τέλος, θέσαμε το πλήθος των γεννεών σε 40 και μετά το πέρας της διαδικασίας τυπώνουμε τα 10 καλύτερα βάρη με την διαδικασία όμως της κανονικοποίησης που αναφέραμε πριν.

### 5.4.2 Μετατροπή Ταξινομητών

Οι Γενετικοί Αλγόριθμοι, πέρα από την συμβολή τους στον Συνδιασμό Μοντέλων, χρησιμοποιήθηκαν και ως μετατροπείς των προβλέψεων Παλινδρόμησης (regression) σε προβλέψεις Βαθμωτής Ταξινόμησης (ordinal classification). Συγκεκριμένα χρησιμοποιήσαμε τις προβλέψεις από το Task 1: EI-reg που αποτελούν τιμές στο διάστημα  $[0, 1]$  για να υπολογίσουμε τις τιμές κλάσεων του Task 2: EI-oc. Ψάχνουμε έτσι να βρούμε εκείνα τα κατώφλια (thresholds) για κάθε συναίσθημα, τα οποία θα καθορίζουν την κλάση πρόβλεψης.

Έτσι χρησιμοποιήσαμε ακριβώς τον ίδιο Γενετικό Αλγόριθμο με αυτόν που περιγράψαμε στο προηγούμενο υποκεφάλαιο, αλλάζοντας όμως την Fitness Function. Σχεδιάσαμε μία συνάρτηση η οποία ταξινομούσε αρχικά τα βάρη και στη συνέχεια υπολόγιζε το μεγαλύτερο από αυτά, διαιρώντας κάθε στοιχείο με αυτό το βάρος, ώστε το μέγιστο κατώφλι να είναι ο αριθμός 1, όπως και επιθυμούμε. Ακολουθεί η Fitness Function ώστε να γίνει πιο κατανοητή:

---

```
def make_oc_predictions(weights, predictions):
    max_weight = max(weights)
    weights = sorted(weights)
    for weight in weights:
        weight = weight / max_weight
    ordinal_c = []
    for pr in predictions:
        if pr < weights[0]:
            ordinal_c.append(0)
        elif pr < weights[1]:
            ordinal_c.append(1)
        elif pr < weights[2]:
            ordinal_c.append(2)
        else:
            ordinal_c.append(3)
    return ordinal_c
```

---

Στην συνέχεια υπολογίζαμε την τιμή Pearson Correlation για το διάνυσμα *ordinal\_c*.

## 5.5 Συμπληρωματικές Σημειώσεις

Στις περιπτώσεις των Νευρωνικών BiLSTM και BiLSTM με Attention Mechanism χρησιμοποιήσαμε σε όλα τα στρώματα την Συνάρτηση Ενεργοποίησης relu. Εξαιρέση αποτελεί το τελευταίο στρώμα, δηλαδή το στρώμα εξόδου όπου χρησιμοποιήσαμε την συνάρτηση softmax. Και οι δύο συναρτήσεις χρησιμοποιήθηκαν ως έχουν χωρίς περαιτέρω παραμετροποιήσεις. Οι εποχές εκπαίδευσης ορίστηκαν στις 15 μιας και παρατηρήσαμε πως περισσότερα στάδια εκπαίδευσης δεν οδηγούσαν σε βελτίωση των προβλέψεων. Το εσω-

τερικό διάνυσμα για επικύρωση ορίστηκε στο 10% του συνολικού (`validation_split = 0.1`), ενώ το πλήθος των δειγμάτων σε κάθε βαθμωτή ανανέωση τέθηκε στον αριθμό 32 (`batch_size = 32`).

Για την δημιουργία των Σχημάτων αυτού του Κεφαλαίου καθώς και του Σχήματος 6.1 χρησιμοποιήθηκε η πλατφόρμα `draw.io`<sup>1</sup>.

---

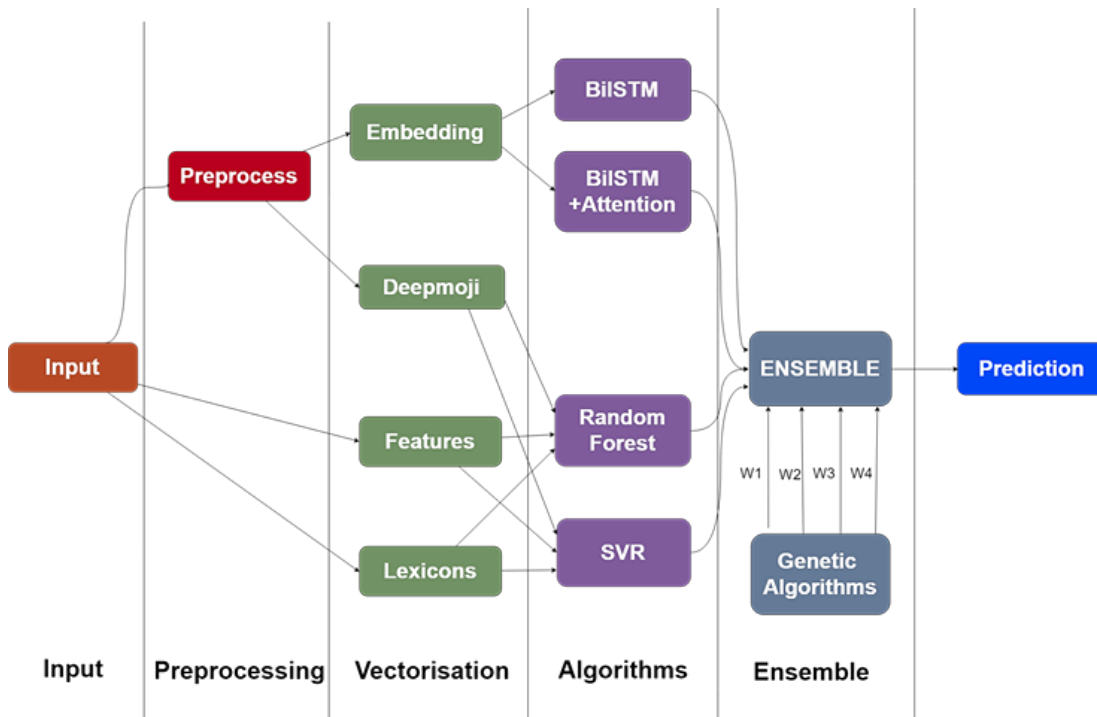
<sup>1</sup><https://www.draw.io/>



## Κεφάλαιο 6

# Πειραματική Διαδικασία και Αποτελέσματα

Στο συγκεκριμένο κεφάλαιο θα παρουσιάσουμε αναλυτικά τον διαγωνισμό με τον οποίο ασχοληθήκαμε αν και δεν συμμετείχαμε κατά την διάρκειά του, αλλά και τα δεδομένα που χρησιμοποιήσαμε. Θα δούμε τις διαφορετικές αποδόσεις των αλγορίθμων σε κάθε διαφορετικό task. Στο Σχήμα 6.1 βλέπουμε ότι ακολουθούνται ορισμένα στάδια από το διάβασμα των δεδομένων μέχρι και την πρόβλεψη, τον τρόπο με τον οποίο υλοποιήθηκαν, αναλύουμε στη συνέχεια. Θεωρήσαμε πως εφόσον τόσο τα λεξικά όσο και η εξαγωγή χαρακτηριστικών (Feature Extraction) δημιουργούν κάποιου είδους κωδικοποίηση κειμένου → διάνυσμα, θα πρέπει να τοποθετηθούν στην ομάδα διανυσματοποίησης (Vectorization)



Σχήμα 6.1: Στάδια Υλοποίησης

## 6.1 Δεδομένα

### 6.1.1 Πηγές δεδομένων

Στο συγκεκριμένο υποκεφάλαιο αναλύουμε τα δεδομένα που μας παρείχε ο διαγωνισμός καθώς και κάποια χαρακτηριστικά πάνω σε αυτά. Τα δεδομένα συλλέχθηκαν για σκοπούς του Διαγωνισμού Semeval-2018 και πιο συγκεκριμένα για το Task 1: Affect in Tweets. [1]. Αναλυτικά ο τρόπος με τον οποίο συλλέχθηκαν τα αγγλικά tweets στο paper ανάλυσης των δεδομένων που έχει συσταθεί από τους διοργανωτές [68]. Ο διαγωνισμός διέθετε επίσης Ισπανικά και Αραβικά δεδομένα με τα οποία όμως δεν δουλέψαμε και δεν θα προβούμε σε λεπτομέρειες. Θα αναφερόμαστε στα τρία tasks που αναλύσαμε στο Κεφάλαιο 1 ως EI-reg, EI-oc, E-c.

Οι Mohammad and Bravo-Marquez (2017) [24] σύλλεξαν και τοποθέτησαν ετικέτες συναισθήματος σε 7,100 αγγλικά tweets που δημοσιεύτηκαν το 2016 χρησιμοποιώντας την τεχνική Best Worst Scaling [69]. Θα αναφερόμαστε σε αυτό το dataset ως *Tweets-2016*. Το dataset αυτό χρησιμοποιήθηκε για τον Διαγωνισμό 2017 WASSA Shared Task on Emotion Intensity <sup>1</sup>.

Στη συνέχεια στηριζόμενοι στην προηγούμενη δουλειά των Mohammad, Bravo-Marquez συλλέχθηκαν καινούργια αγγλικά tweets από το 2017 και τοποθετήθηκαν ετικέτες με την ίδια τεχνική. Θα αναφερόμαστε σε αυτό το dataset ως *Tweets-2017*. Σε αντίθεση με την δουλειά των Mohammad, Bravo-Marquez τοποθετούνται ετικέτες σε όλα τα αρνητικά συναισθήματα (φόβος, λύπη, θυμός) ξεχωριστά. Αυτό επιτρέπει να μελετηθούν οι διάφορες σχέσεις ανάμεσα σε αυτά τα τρία αρνητικά συναισθήματα. Ο Πίνακας 6.1 δείχνει τα δύο στάδια που έγιναν οι τοποθετήσεις ετικετών στα αγγλικά tweets για τα τρία tasks με τα οποία ασχοληθήκαμε.

Dataset	Source of Tweets	Annotated in	
		2016	2017
E-c	Tweets-2016	-	✓
	Tweets-2017	-	✓
EI-reg, EI-oc	Tweets-2016	✓	-
	Tweets-2017	-	✓

Table 6.1: Ετικέτες για αγγλικά tweets

Για να δημιουργηθεί ένα dataset πλούσιο σε ένα συναίσθημα ακολουθήθηκε η παρακάτω διαδικασία.

Για κάθε συναίσθημα X, επιλέχθηκαν από 50 ως 100 όροι που συνδέονται με αυτό το συναίσθημα σε διαφορετικές επίπεδα έντασης. Για παράδειγμα, για το συναίσθημα θυμού, χρησιμοποιήθηκαν οι όροι (στα αγγλικά) : angry, mad, frustrated, annoyed, peeved, irritated, miffed, fury, antagonism, etc. Θα αναφερόμαστε σε αυτούς τους όρους ως *όρους ερωτήσεως (query terms)*. Τα query terms που έχουν επιλέχθει περιείχαν συναισθηματικές

<sup>1</sup><http://saifmohammad.com/WebPages/EmoInt2017.html>

λέξεις που αντλήθηκαν από τα κάτωθι:

- Roget's Thesaurus [70]
- Τις πιο κοντινές λέξεις του συναισθήματος στο διανυσματικό χώρο λέξεων
- Συχνώς χρησιμοποιούμενα σχετικά με το συναίσθημα emojis και emoticons

Όλα τα query terms είναι διαθέσιμα στην ιστοσελίδα του διαγωνισμού <sup>2</sup> και διαθέσιμα προς κατέβασμα (download).

### 6.1.2 Διαδικασία Συλλογής

Χρησιμοποιήθηκε το Twitter-API για δύο μήνες (Ιούνιος και Ιούλιος 2017) ώστε να συλλεχθούν tweets που περιείχαν τα query terms. Αρχικά επιλέχθηκαν 1,400 tweets από την κατηγορία του συναισθήματος χαράς προκειμένου να τους τοποθετηθούν ετικέτες έντασης. Για τα τρία αρνητικά συναισθήματα επιλέχθηκαν 200 tweets τυχαία για καθεμία από τις αντίστοιχες συλλογές. Αυτά τα 600 tweets χρησιμοποιήθηκαν για τοποθέτηση ετικετών έντασης και για τα τρία συναισθήματα προκειμένου να μπορούν να εξαχθούν συμπεράσματα ανάμεσα σε φόβο-θυμό, θυμό-λύπη και λύπη-φόβο. Επίσης για κάθε ένα από τα τρία αρνητικά συναισθήματα επιλέχθηκαν 800 tweets για καθεμία από τις αντίστοιχες συλλογές τους όπου τοποθετήθηκαν ετικέτες έντασης μόνο για το αντίστοιχο συναίσθημα. Έτσι τα συνολικά tweets για κάθε αρνητικό συναίσθημα είναι επίσης  $1400 = 600 + 800$ . Τέλος, για κάθε συναίσθημα επιλέχθηκαν 100 τυχαία tweets από τα 1400 που περιείχαν ένα hashtag συναισθήματος ή ένα emoticon ή ένα emoji στο τέλος τους (trailing query term). Το query term αυτό αφαιρέθηκε από τα tweets, ούτως ώστε το dataset να περιέχει και μερικά tweets χωρίς ξεκάθαρο συναισθηματικό προσανατολισμό.

Έτσι το dataset Tweets-2017 διαθέτει πλέον 1,400 tweets για κάθε συναίσθημα στα οποία τοποθετήθηκαν ετικέτες έντασης χειροκίνητα. Το dataset Tweets-2016 διέθετε ήδη από 1500 ως 2300 tweets για κάθε συναίσθημα, στα οποία επίσης τοποθετήθηκαν ετικέτες χειροκίνητα. Όλα αυτά τα tweets χρησιμοποιήθηκαν τόσο για το EI-reg Task όσο και για το EI-oc. Τα tweets του dataset Tweets-2017 αποτέλεσαν τα δεδομένα αξιολόγησης (test) και ελέγχου (development) ενώ τα δεδομένα του dataset Tweets-2016 χρησιμοποιήθηκαν ως δεδομένα εκπαίδευσης (train).

Σημείωση: Με τον χειροκίνητο έλεγχο των δεδομένων για το EI-reg Task παρατηρήθηκε ότι υπήρχαν μερικά ζευγάρια σχεδόν πανομοιότυπων tweets τα διπλότυπα των οποίων αφαιρέθηκαν, για αυτό και όπως βλέπουμε στον Πίνακα 6.2 το άθροισμα των dev και test δεδομένων ανθροίζει λίγο κάτω από 1,400.

Για το E-c Task επιλέχθηκαν όλα τα tweets από τα τέσσερα συναισθήματα του EI-reg dataset (αφού αφαιρέθηκαν οι επαναλήψεις) στα οποία τοποθετήθηκαν ετικέτες ύπαρξης / απώλειας 11 συναισθημάτων. Ακολουθεί ο πίνακας με τα ακριβή νούμερα για κάθε task, συναίσθημα και σετ.

<sup>2</sup><http://saifmohammad.com/WebDocs/AIT-2018/SemEval2018-Task1-QueryTerms.zip>

Dataset	Train	Dev	Test	Total
English				
E-c	6,838	886	3,259	10,983
EI-reg, EI-oc				
anger	1,701	388	1,002	3,091
fear	2,252	389	986	3,627
joy	1,616	290	1,105	3,011
sadness	1,533	397	975	2,905

Table 6.2: Συνολικά νούμερα για τα Αγγλικά tweets του Διαγωνισμού

### 6.1.3 Τοποθέτηση Ετικετών

Για την τοποθέτηση ετικετών ακολουθήθηκε η παρακάτω διαδικασία. Χρησιμοποιήθηκε η τεχνική crowdsourcing, όπου μέσω της πλατφόρμας Figure Eight <sup>3</sup> (παλαιότερα γνωστή ως CrowdFlower) κοινοποιήθηκαν τα tweets και οι ερωτήσεις σε άτομα που θα αναλάμβαναν την διαδικασία ετικετοποίησης. Από εδώ και στο εξής θα αποκαλούμε αυτά τα άτομα ως *συμμετέχοντες*.

Ορισμένα από τα tweets (περίπου 5%) ετικετοποιήθηκαν από τους διοργανωτές του διαγωνισμού, τα οποία ονομάζουμε gold tweets. Αυτά τα tweets συνδιάστηκαν μαζί με τα υπόλοιπα. Εάν ένας συμμετέχοντας είχε σημαντική απόκλιση στην πρόβλεψη ενός gold tweet ενημερωνόταν αυτομάτως. Στην περίπτωση που το συνολικό ποσοστό ευστοχίας ενός συμμετέχοντα στα gold tweets ήταν κάτω από 70% όλες οι προβλέψεις του δεν προσμετρούνταν. Αυτό ακολουθήθηκε προκειμένου να αποφευχθούν κακοήθεις προβλέψεις.

Όσον αφορά το task E-c δώθηκαν στους συμμετέχοντες tweets και τους ζητήθηκε να αποφασίσουν για το ποιά από τα 11 συναισθήματα του task εμφανίζονται σε αυτό. Ως οδηγία είχαν δοθεί δεδομένα με κατάλληλες απαντήσεις. Στις ρυθμίσεις του Figure-Eight καθορίστηκε ότι χρειάζονται προβλέψεις από επτά άτομα για κάθε tweet. Ωστόσο, λόγω του τρόπου με τον οποίο έχουν συσταθεί τα gold tweets, υπήρχαν περισσότερες από 7 προβλέψεις ανά tweet. Ο διάμεσος αριθμός (median) ετικετοποιήσεων ήταν όμως ακόμα εφτά. Συνολικά, 303 άτομα τοποθέτησαν ετικέτες σε εύρος μεταξύ 10 και 4,670 tweets ο καθένας. Σύνολο 174.356 αποκρίσεις.

anger	antic.	disg.	fear	joy	love	optim.	pessi.	sadn.	surp.	trust	neutral
36,1	13,9	36,6	16,8	39,3	12,3	31,3	11,6	29,4	5,2	5,0	2,7

Table 6.3: Ποσοστό tweets που επιλέχθηκαν ως αντιπροσωπευτικά δεδομένου συναισθήματος

Τα κριτήρια που ακολουθήθηκαν για την διαδικασία συνδιασμού των δεδομένων των συμμετεχόντων στο task E-c προκειμένου να έχουμε συνέπεια στο dataset που θα δημιουργηθεί είναι τα ακόλουθα:

<sup>3</sup><https://www.figure-eight.com/>

- Αν το ποσοστό που συμφωνεί ότι ένα συναίσθημα E συναντάται στο tweet είναι μεγαλύτερο του 25% (δηλαδή περισσότεροι από 2 / 7) τότε το συναίσθημα επιλεγόταν ως υπάρχον. Αν κανένα συναίσθημα δεν πήρε τουλάχιστον 25%, τότε το συναίσθημα δεν υπάρχει.
- Αν κανένα συναίσθημα δεν έλαβε τουλάχιστον το 40% των ψήφων και το ποσοστό των ψήφων που δόθηκε στην κλάση ουδέτερος (neutral) είναι μεγαλύτερο από το 50%, το tweet αυτό χαρακτηρίστηκε ως ουδέτερο.
- Σε περίπτωση που δεν ίσχυε κανένα από τα παραπάνω τα tweets αποσυρόταν. Μόλις 107 αγγλικά tweets αποσύρθηκαν.

Στον Πίνακα 6.3 εμφανίζεται το ποσοστό των tweets που είχαν επισημανθεί με ένα δεδομένο συναίσθημα χρησιμοποιώντας την μέθοδο που αναλύσαμε. Παρατηρούμε ότι η χαρά, ο θυμός, η απέχθεια, η θλίψη και η αισιοδοξία έχουν πάρει μεγάλο αριθμό ψήφων. Εμπιστοσύνη και έκπληξη είναι δύο από τα χαμηλότερα σε ψήφους συναισθήματα ενώ τα tweets που δεν κατάφεραν να μαζέψουν αρκετές ψήφους για ένα συναίσθημα τουλάχιστον, είναι ελάχιστα.

## 6.2 Προεπεξεργασία Δεδομένων

Όπως αναφέραμε και στα προηγούμενα κεφάλαια τα δεδομένα πριν κωδικοποιηθούν σε διανύσματα υφίστανται μία προεπεξεργασία. Προκειμένου όλη αυτή η διαδικασία να μην γίνει χειροκίνητα και μιας και τα tweets αποτελούν μία ειδική κατηγορία κειμένων έχοντας αρκετές ιδιομορφίες, αποφασίσαμε να χρησιμοποιήσουμε μία βοηθητική βιβλιοθήκη. Ανάμεσα σε διάφορες η βιβλιοθήκη `tweetokenize`<sup>4</sup> θεωρήθηκε η πιο κατάλληλη.

Οι μετατροπές που κάνει η συγκεκριμένη βιβλιοθήκη είναι οι κάτωθι:

- Μετατροπή όλων των κεφαλαίων γραμμάτων σε μικρά. Εξαιρούνται οι λέξεις οι οποίες έχουν όλα τα γράμματά τους σε κεφαλαία οι οποίες και διατηρήθηκαν ως είχαν χωρίς πεζοποίηση.
- Μετατροπή όλων των usernames στον όρο 'USERNAME'  
Μετατροπή όλων των urls στον όρο 'URL'  
Μετατροπή όλων των τηλεφώνων στον όρο 'PHONENUMBER'  
Μετατροπή όλων των ωρών στον όρο 'TIME'
- Κανονικοποίηση επαναλαμβανόμενων γραμμάτων. Αν το πλήθος των επαναλαμβανόμενων γραμμάτων είναι  $\leq 2$  τότε δεν μετατρέπονται (πχ η λέξη `illness` δεν πρέπει να μετατραπεί). Σε αντίθετη περίπτωση έχουμε σύμπτυξη των συνεχόμενων ίδιων γραμμάτων σε ένα γράμμα.
- Διαχωρισμός των συνεχόμενων emojis σε ξεχωριστά.

<sup>4</sup><https://github.com/jaredks/tweetokenize>

Ακολουθεί ένα παράδειγμα της βιβλιοθήκης αυτής σε χρήση:

```
text: heyyy playa!):3.....@SHAQ can you still dunk? #old 🏀 😂 LOL)
```

```
tokenization: ['hey', 'playa', '!', ':)', ':3', '...', 'USERNAME', 'can', 'you', 'still', 'dunk',
'?', '#old', '🏀', '😂', 'LOL']
```

Η συγκεκριμένη όμως βιβλιοθήκη είχε κάποιες ελλείψεις τις οποίες όμως εμείς θέλαμε να συμπεριλάβουμε στην προεπεξεργασία μας. Για τον λόγο αυτό μετά την επεξεργασία που περιγράψαμε, συμπληρώσαμε με τις παρακάτω τεχνικές τις οποίες υλοποιήσαμε χειροκίνητα:

- Αφαίρεση χαρακτήρων αλλαγών γραμμής.
- Διάσπαση των hashtags με βάση τον χαρακτήρα underscore (Για παράδειγμα το hashtag #hello\_world μετατράπηκε σε ['hello', 'world'])
- Τα emojis αντικαταστάθηκαν με την περιγραφή τους. Για να γίνει αυτό χρησιμοποιήθηκε η βιβλιοθήκη emoji <sup>5</sup> της Python. Παρατηρήθηκε σημαντική βελτίωση με την χρήση αυτού του βήματος.

Αποφασίστηκε να μην χρησιμοποιηθεί ούτε lemming ούτε stemming μιας και θεωρήσαμε ότι τα ιδιαίτερα χαρακτηριστικά του λόγου συνδέονται άμεσα με το συναίσθημα. Επίσης δεν έγινε ορθογραφική διόρθωση μιας και τα πολλά ορθογραφικά μπορεί να σημαίνουν έντονη συναισθηματική κατάσταση. Για τον λόγο αυτό συμπεριελάβαμε το πλήθος των ορθογραφικών λαθών ενός tweet στην μέθοδο Εξαγωγής Χαρακτηριστικών που θα αναλύσουμε στη συνέχεια.

### 6.3 Δημιουργία Διανυσμάτων

Μετά την προεπεξεργασία κειμένου έχουμε πλέον κάθε tweet σε μία λίστα από tokens. Αυτή ακριβώς τη λίστα πρέπει να την μετατρέψουμε σε διανύσματα πραγματικών αριθμών. Στη συνέχεια αναλύουμε τις μεθόδους που χρησιμοποιήσαμε. Η πρώτη μέθοδος, δηλαδή τα Διανύσματα Λέξεων, χρησιμοποιήθηκε στους Αλγορίθμους με τις μονάδες Μακράς Βραχυπρόθεσμης Μνήμης (LSTM), ενώ οι υπόλοιπες τρεις (Διανύσματα Deepmoji, Εξαγωγή χαρακτηριστικών καθώς και Λεξιικά) χρησιμοποιήθηκαν στους Αλγορίθμους Τυχαίων Δέντρων (Random Trees) και Μηχανών Διανυσμάτων Υποστήριξης τύπου Παλινδρόμησης (SVR).

<sup>5</sup><https://pypi.org/project/emoji/>

### 6.3.1 Διανύσματα Λέξεων

Αρχικά εκπαιδεύσαμε τα δικά μας Διανύσματα Λέξεων με τη χρήση του αλγορίθμου Word2Vec της βιβλιοθήκης gensim [71] που αναλύσαμε στο Κεφάλαιο 3, μόνο και μόνο ως σύγκριση για τους υπόλοιπους αλγορίθμους. Τα αποτελέσματα όπως περιμέναμε δεν ήταν ικανοποιητικά για αυτό και προβήκαμε σε χρήση προ-εκπαιδευμένων διανυσμάτων. Έτσι χρησιμοποιήσαμε τα Glove Vectors (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, 300d vectors) τα οποία έχουν εκπαιδευτεί από δεδομένα που έχουν συλλεχθεί από την Wikipedia το 2014. Επίσης χρησιμοποιήθηκαν τα Google News vectors<sup>6</sup>. Περιλαμβάνουν διανύσματα για ένα λεξιλόγιο τριών εκατομμυρίων λέξεων και φράσεων που εκπαιδεύτηκαν σε περίπου εκατό δισεκατομμύρια λέξεις από ένα σύνολο δεδομένων από ειδήσεις στην Google. Το μήκος των διανυσμάτων είναι 300.

### 6.3.2 Deepmoji

Το DeepMoji (Felbo et al., 2017) [72] αποτελεί ένα προεκπαιδευμένο μοντέλο το οποίο στοχεύει στην πρόβλεψη emojis σε κείμενο. Συγκεκριμένα έχει ως πιθανή έξοδο 64 από τα πιο συχνώς χρησιμοποιούμενα emojis. Εκπαιδεύτηκε από ένα corpus 1.2 δισεκατομμυρίων tweets που περιέχουν emojis και αρχιτεκτονικά όπως και εμείς χρησιμοποιεί ένα BiLSTM και ένα Attention Layer. Εμείς από αυτό το μοντέλο λαμβάνουμε την έξοδο ακριβώς πριν το softmax layer, η οποία έχει μέγεθος 2304 και δεν χρησιμοποιούμε καθόλου τις προβλέψεις του softmax layer. Το μοντέλο αυτό χρησιμοποιήθηκε γιατί θεωρούμε πως υπάρχει άμεση σχέση ανάμεσα στα emojis και το συναίσθημα και λόγω του πολύ μεγάλου όγκου δεδομένων με το οποίο εκπαιδεύτηκε, το έκανε κατάλληλο για την περίπτωσή μας. Επίσης το συγκεκριμένο μοντέλο χρησιμοποιήθηκε από δύο εκ των τριών πρώτων σε αξιολόγηση συμμετεχόντων στο διαγωνισμό.

Το DeepMoji επίσης διαθέτει μία ιστοσελίδα<sup>7</sup> παρουσίασης του μοντέλου τους όπως και εμείς, όπου ο χρήστης μπορεί να πληκτρολογήσει μία φράση και το μοντέλο να προβλέψει τα emojis που ταιριάζουν σε αυτήν την φράση.

### 6.3.3 Λεξικά

Για τον διαγωνισμό στον οποίο παράξαμε προβλέψεις, οι διοργανωτές μέσω του πακέτου AffectiveTweets<sup>8</sup> των Mohammad, Brav Marquez [24], έκαναν διαθέσιμα τα tweets σε μορφή arff<sup>9</sup> ούτως ώστε να είναι αποδεκτά από το πακέτο. Το συγκεκριμένο πακέτο επιτρέπει την εξαγωγή χαρακτηριστικών από διάφορα λεξικά καθώς και επιτρέπει στον χρήστη την προεπεξεργασία του κειμένου πριν την εξαγωγή των χαρακτηριστικών από τα λεξικά. Εμείς χρησιμοποιήσαμε τα default. Από τα λεξικά επιλέξαμε τα κάτωθι:

<sup>6</sup><https://github.com/mmihaltz/word2vec-GoogleNews-vectors>

<sup>7</sup><https://deepmoji.mit.edu/>

<sup>8</sup><https://affectivetweets.cms.waikato.ac.nz/>

<sup>9</sup><https://www.cs.waikato.ac.nz/ml/weka/arff.html>



- **MPQA**: Μετρά το πλήθος θετικών και αρνητικών λέξεων βάσει της MPQA υποκειμενικότητας [73]
- **Bing Liu**: Μετρά το πλήθος θετικών και αρνητικών λέξεων βάσει του λεξικού Bing Liu. [74]
- **AFINN**: Υπολογίζει τις θετικές και αρνητικές μεταβλητές συγκεντρώνοντας τις θετικές και αρνητικές βαθμολογίες λέξεων που παρέχονται από αυτό το λεξικό. [75]
- **Sentiment140**: Υπολογίζει τις θετικές και αρνητικές μεταβλητές συγκεντρώνοντας τις θετικές και αρνητικές βαθμολογίες λέξεων που παρέχονται από αυτό το λεξικό που δημιουργήθηκε με tweets ετικετοποιημένα από emoticons. [76]
- **NRC Hashtag Sentiment lexicon**: Υπολογίζει τις θετικές και αρνητικές μεταβλητές συγκεντρώνοντας τις θετικές και αρνητικές βαθμολογίες λέξεων που παρέχονται από αυτό το λεξικό που δημιουργήθηκε με tweets ετικετοποιημένα από συναισθηματικά hashtags. [76]
- **NRC Word-Emotion Association Lexicon**: Μετρά το πλήθος των λέξεων που αντιστοιχεί στο κάθε συναίσθημα αυτού του λεξικού. [77]
- **NRC-10 Expanded**: Προσθέτει τις συναισθηματικές συσχετίσεις των λέξεων που υπάρχουν στο Twitter Specific Expansion του NRC Word-Emotion Association Lexicon. [78]
- **NRC Hashtag Emotion Association Lexicon**: Προσθέτει τις συναισθηματικές σχέσεις των λέξεων που υπάρχουν στο δεδομένο λεξικό. [79]
- **SentiWordNet**: Υπολογίζει τα θετικά και αρνητικά σκορ χρησιμοποιώντας το SentiWordnet, μέσω ενός σταθμισμένου μέσου των συναισθηματικών κατανομών των σχέσεων του συνόλου των λέξεων που εμφανίζονται σε πολλαπλά υποσύνολα. Τα βάρη αντιστοιχούν στις αμοιβαίες τάξεις των συναισθημάτων για να δώσουν υψηλότερα βάρη στις πιο δημοφιλείς συναισθήματα. [80]
- **Emoticons**: Υπολογίζει τα θετικά και αρνητικά σκορ συγκεντρώνοντας τις συσχετίσεις λέξεων οι οποίες εξάγονται από μία λίστα από emoticons. Αυτή η λίστα προέρχεται από το AFINN.
- **SentiStrength**: Υπολογίζει τις θετικές και αρνητικές συναισθηματικές εντάσεις για ένα tweet χρησιμοποιώντας το SentiStrength. [81]

Το δημιουργηθέν διάγραμμα έχει μήκος 50. Το αρνητικό με την συγκεκριμένη βιβλιοθήκη είναι πως επειδή είναι γραμμένη σε Java για να την τρέξεις μία φορά είναι εύκολο αφού τρέχουμε απλά μία javac εντολή. Στην περίπτωση μας, όμως, που θέλουμε να κάνουμε ζωντανή πρόβλεψη tweet για λόγους της διαδικτυακής εφαρμογής, έπρεπε μέσω της python να τρέξουμε τις javac εντολές στο terminal και να συλλέξουμε τα αποτελέσματα.



### 6.3.4 Εξαγωγή Χαρακτηριστικών - Feature Extraction

. Υπάρχουν κάποια χαρακτηριστικά τα οποία δεν μπορούν να εξαχθούν από καμία από τις μεθόδους τις οποίες αναφέραμε παραπάνω. Πρόκειται για τα ιδιαίτερα χαρακτηριστικά της φύσης κάθε tweet. Αυτά τα χαρακτηριστικά στην Επεξεργασία Φυσικής Γλώσσας που αφορά tweets ονομάζονται Tweet Specific Features. Συγκεκριμένα χρησιμοποιούμε τα χαρακτηριστικά που παρουσιάζονται στη συνέχεια:

1. Πλήθος χαρακτήρων ενός tweet
2. Πλήθος λέξεων ενός tweet
3. Πλήθος emojis ενός tweet
4. Πλήθος συνεχόμενων γραμμάτων ενός tweet (με ελάχιστο τα 3 συνεχόμενα γράμματα)
5. Πλήθος σημείων στίξης στο tweet
6. Πλήθος λέξεων που είναι γραμμένες με κεφαλαία γράμματα στο tweet
7. Πλήθος συντακτικών λαθών. Η διαδικασία αυτή έγινε μέσω της βιβλιοθήκης Hunspell <sup>10</sup>. Ελέγχουμε αν μία λέξη έχει μόνο χαρακτήρες της αλφαβήτου και αν ναι βρήκαμε την πιο κοντινή της λέξη μέσω της βιβλιοθήκης. Αν η κοντινή λέξη και η κανονική λέξη δεν είναι ίδιες, θεωρούσαμε πως πρόκειται για συντακτικό λάθος.

Όπως φαίνεται και από την λίστα που προηγήθηκε τα χαρακτηριστικά αυτά είναι 7 στο σύνολο. Συνεπώς, πλέον έχουμε δημιουργήσει για τους Αλγορίθμους Τυχαίων Δέντρων (Random Trees) και Μηχανών Διανυσμάτων Υποστήριξης τύπου Παλινδρόμησης (SVR) ένα διάνυσμα μήκους  $2304 + 50 + 7 = 2361$  (DeepMoji + Λεξικά + Χαρακτηριστικά).

## 6.4 Αξιολόγηση Αλγορίθμων

### 6.4.1 Task #1: Emotion Intensity - Regression (EI-reg)

Η διαδικασία που ακολουθήσαμε έχει ως εξής: αφού χρησιμοποιήσαμε το train dataset για να βρούμε τις υπερπαραμέτρους που φέρνουν τα καλύτερα αποτελέσματα, συγχωνεύσαμε τα datasets train και dev για να εκπαιδεύσουμε το τελικό μας Νευρωνικό Δίκτυο. Επίσης υπολογίσαμε τα βάρη που πρέπει να πολλαπλασιαστούν με κάθε λίστα από τις προβλέψεις τα οποία χρησιμοποιήσαμε στο τέλος για τις τελικές μας προβλέψεις. Στους πίνακες 6.4, 6.5, 6.6 βλέπουμε τα αποτελέσματα για όλα τα νευρωνικά μας, ενώ στον πίνακα 6.7 βλέπουμε και την σύγκρισή τους μαζί τους άλλους διαγωνιζόμενους.

Όπως παρατηρούμε στον πίνακα 6.4 τα tweet specific features βελτιώνουν σε κάθε περίπτωση τις προβλέψεις, στον αλγόριθμο Μηχανών Διανυσμάτων Υποστήριξης (SVR).

<sup>10</sup><http://hunspell.github.io/>

Model	Anger	Fear	Joy	Sadness
SVR	0.800	0.703	0.718	0.754
SVR + Lexicons	0.789	0.739	0.726	0.742
SVR + Features	<b>0.802</b>	0.704	0.721	<b>0.755</b>
SVR + Lexicons + Features	0.791	<b>0.740</b>	<b>0.727</b>	0.743
Random Forest	0.802	0.705	0.734	0.765
Random Forest + Lexicons	<b>0.811</b>	0.726	<b>0.741</b>	0.774
Random Forest + Features	0.805	0.707	0.734	0.765
Random Forest + Lexicons + Features	0.809	<b>0.728</b>	0.740	<b>0.775</b>

Table 6.4: Αποτελέσματα Machine Learning Algorithms, Task #1: EI-reg

Η χρήστη λεξικών όμως βελτιώνει τις προβλέψεις μας μόνο στα Συναισθήματα του Φόβου (κατά πολύ) και της Χαράς. Εικάζουμε πως στα Συναισθήματα του Θυμού και της Λύπης, τα λεξικά υπερτονίζουν τις λέξεις με συναισθηματικό φόρτο, οδηγώντας σε υπερεκτίμηση της πρόβλεψης.

Αντιθέτως στον Ταξινομητή Τυχαίων Δασών (Random Forests) βλέπουμε πως πάντα βελτιώνεται. Η μείωση βέβαια που έχουμε στον SVR είναι της τάξης του 1%, συνεπώς πιθανώς να είναι θέμα τυχαιότητας. Ούτως η άλλως τα DeepMojji vectors διαθέτουν ήδη συναισθηματική γνώση που πιθανώς τα λεξικά δεν μπορούν να ενισχύσουν περαιτέρω.

Model	Anger	Fear	Joy	Sadness
Trained Embeddings	0.602	0.573	0.631	0.577
Google Embeddings	<b>0.694</b>	<b>0.679</b>	<b>0.673</b>	<b>0.667</b>
Glove Emb. - 50	0.608	0.653	0.622	0.626
Glove Emb. - 100	0.654	0.651	0.642	0.632
Glove Emb. - 200	0.660	0.670	0.645	0.633
Glove Emb. - 300	0.674	0.665	0.646	0.631

Table 6.5: Αποτελέσματα Bi-LSTM, Task #1: EI-reg

Στην αρχιτεκτονική του απλού BiLSTM, παρατηρούμε πως τα καλύτερα αποτελέσματα φέρουν τα google pretrained vectors σε όλα τα Συναισθήματα, ενώ τα δικά μας trained vectors, όπως περιμέναμε, έχουν την χαμηλότερη επίδοση, η οποία όμως και πάλι είναι καλύτερη του baseline model (SVM - Unigrams) του πίνακα 6.7

Algorithm	Anger	Fear	Joy	Sadness
Trained Embeddings	0.610	0.611	0.613	0.590
Google Embeddings	<b>0.704</b>	0.688	<b>0.675</b>	<b>0.675</b>
Glove Emb. - 50	0.617	0.685	0.606	0.664
Glove Emb. - 100	0.667	0.689	0.645	0.662
Glove Emb. - 200	0.672	<b>0.695</b>	0.661	0.667
Glove Emb. - 300	0.680	0.677	0.660	0.660

Table 6.6: Αποτελέσματα Bi-LSTM + Attention, Task #1: EI-reg

Όπως και στην Αρχιτεκτονική του απλού BiLSTM έτσι και εδώ, στην Αρχιτεκτονική που συνδυάζει BiLSTM με Μηχανισμό Προσοχής (Attention Mechanism), τα καλύτερα αποτελέσματα προβλέψεων επιτυγχάνου τα Διανύσματα της Google (εκτός του fear).

Pearson r (all instances)							
Test Set	Rank	Team Name	avg.	anger	fear	joy	sadness
English							
	(2)	Tweet-ai	78.5	82.1	<b>78.3</b>	76.1	77.5
	1	SeerNet	<b>79.9</b>	<b>82.7</b>	77.9	<b>79.2</b>	<b>79.8</b>
	2	NTUA-SLP	77.6	78.2	75.8	77.1	79.2
	3	PlusEmo2Vec	76.6	81.1	72.8	77.3	75.3
	23	Median Team	65.3	65.4	67.2	64.8	63.5
	37	SVM-Unigrams	52.0	52.6	52.5	57.5	45.3
	46	Random Baseline	-0.8	-1.8	2.4	-5.8	2.0

Table 6.7: Αποτελέσματα διαγωνιζόμενων στο Task #1: EI-reg

#### 6.4.2 Task #2: Emotion Intensity - Ordinal Classification (EI-oc)

Τα datasets Για το Task #2 είναι τα ίδια με το Task #1. Συνεπώς αποφασίσαμε να μην εκπαιδεύσουμε εκ νέου καινούργια νευρωνικά δίκτυα αλλά να φτιάξουμε μία αντιστοίχιση από το προηγούμενο Task, σε αυτό. Έτσι ψάχνουμε να βρούμε τα κατώφλια εκείνα που θα μας δώσουν το υψηλότερο δυνατό σκορ επιτυχίας. Υπενθυμίζουμε πως στο συγκεκριμένο Task έχουμε να προβλέψουμε ανάμεσα σε 4 διαφορετικές κλάσεις (καθόλου Συναίσθημα E, ελάχιστο Συναίσθημα E, αρκετό Συναίσθημα E, πολύ Συναίσθημα E).

- **Θυμός (Anger:** [0.460, 0.497, 0.529, 1.0]
- **Φόβος (Fear:**[0.534, 0.615, 0.693, 1.0]
- **Χαρά (Joy:** [0.464, 0.604, 0.680, 1.0]
- **Λύπη (Sadness:** [0.456, 0.526, 0.577, 1.0])

Pearson r (all instances)							
Test Set	Rank	Team Name	avg.	anger	fear	joy	sadness
English							
	(2)	Tweet-ai	67.2	70.1	61.0	66.1	<b>71.8</b>
	1	SeerNet	<b>69.5</b>	<b>70.6</b>	<b>63.7</b>	<b>72.0</b>	71.7
	2	PlusEmo2Vec	65.9	70.4	52.8	72.0	68.3
	3	psyML	65.3	67.0	58.8	68.6	66.7
	17	Median Team	53.0	53.0	47.0	55.2	56.7
	26	SVM-Unigrams	39.4	38.2	35.5	46.9	37.0
	37	Random Baseline	-1.6	-6.2	4.7	1.4	-6.1

Table 6.8: Αποτελέσματα και Σύγκριση Προβλέψεων, Task #2: EI-oc

### 6.4.3 Task #5: Emotion - Classification (E-c)

Η διαδικασία που ακολουθήσαμε όσον αφορά τα datasets είναι αντίστοιχη του Task 1. Αφού δηλαδή βρήκαμε τις κατάλληλες υπερπαραμέτρους των μοντέλων μας, συνενώσαμε τα δύο datasets train και dev για να εκπαιδεύσουμε τα τελικά μας νευρωνικά. Στο συγκεκριμένο Task αποφασίσαμε να ασχοληθούμε μόνο με Αλγορίθμους Μηχανικής Μάθησης λόγω του ότι τα tweets που είχαμε προς εκπαίδευση ήταν πάρα πολλά (>7,000) και η Βαθιά Μάθηση θα ήταν ιδιαίτερος χρονοβόρα. Έτσι χρησιμοποιήσαμε τους Αλγορίθμους Μηχανικής Μάθησης LinearSVC και RandomForestClassifier σε συνδιασμό με την μέθοδο MultiOutputClassifier της βιβλιοθήκης scikit-learn, μιας και είχαμε πολλαπλές εξόδους. Στον πίνακα 6.9 παραθέτουμε τα αποτελέσματα από αυτούς τους αλγορίθμους σε συνδιασμό με τις τεχνικές που αναφέραμε (DeepMoji, λεξικά, χαρακτηριστικά).

#	Model	micro macro		
		acc.	F1	F1
1.	LinearSVC	52.9	65.5	48.1
2.	LinearSVC + Lexicons	56.9	69.4	56.0
3.	LinearSVC + Features	53.5	65.9	48.2
4.	LinearSVC + Lexicons + Features	<b>57.3</b>	<b>69.8</b>	<b>56.4</b>
5.	Random Forest	48.7	61.4	38.4
6.	Random Forest + Lexicons	52.8	65.9	47.4
7.	Random Forest + Features	48.8	61.5	38.5
8.	Random Forest + Lexicons + Features	<b>53.2</b>	<b>66.0</b>	<b>47.5</b>

Table 6.9: Αποτελέσματα Machine Learning Algorithms, Task #5: E-c

Για τον συνδιασμό των ταξινομητών επιλέξαμε 5 από τους ταξινομητές του Πίνακα 6.9 και πιο συγκεκριμένα τους [2, 3, 4, 6, 8]. Ο τρόπος με τον οποίο κάναμε τον συνδιασμό (ensemble) έχει ως εξής: Εάν 2 ή περισσότεροι ταξινομητές θεωρούσαν ότι το συναίσθημα E υπάρχει για ένα δεδομένο tweet, τότε το σημειώναμε ως υπάρχον. Αλλιώς το θεωρούσαμε ως μη υπάρχον. Στον Πίνακα 6.10 βλέπουμε την σύγκριση με τους άλλους διαγωνιζόμενους, όπου όπως παρατηρούμε έχουμε τις καλύτερες επιδόσεις, σε όλες τις μετρικές.

Rank	Team Name	acc.	micro macro	
			F1	F1
English				
(1)	Tweet-ai	<b>59.9</b>	<b>71.2</b>	<b>56.8</b>
1	NTUA-SLP	58.8	70.1	52.8
2	TCS Research	58.2	69.3	53.0
3	PlusEmo2Vec	57.6	69.2	49.7
17	Median Team	47.1	59.9	46.4
21	SVM-Unigrams	44.2	57.0	44.3
28	Random Baseline	18.5	30.7	28.5

Table 6.10: Αποτελέσματα και Σύγκριση Προβλέψεων, Task #5: E-c

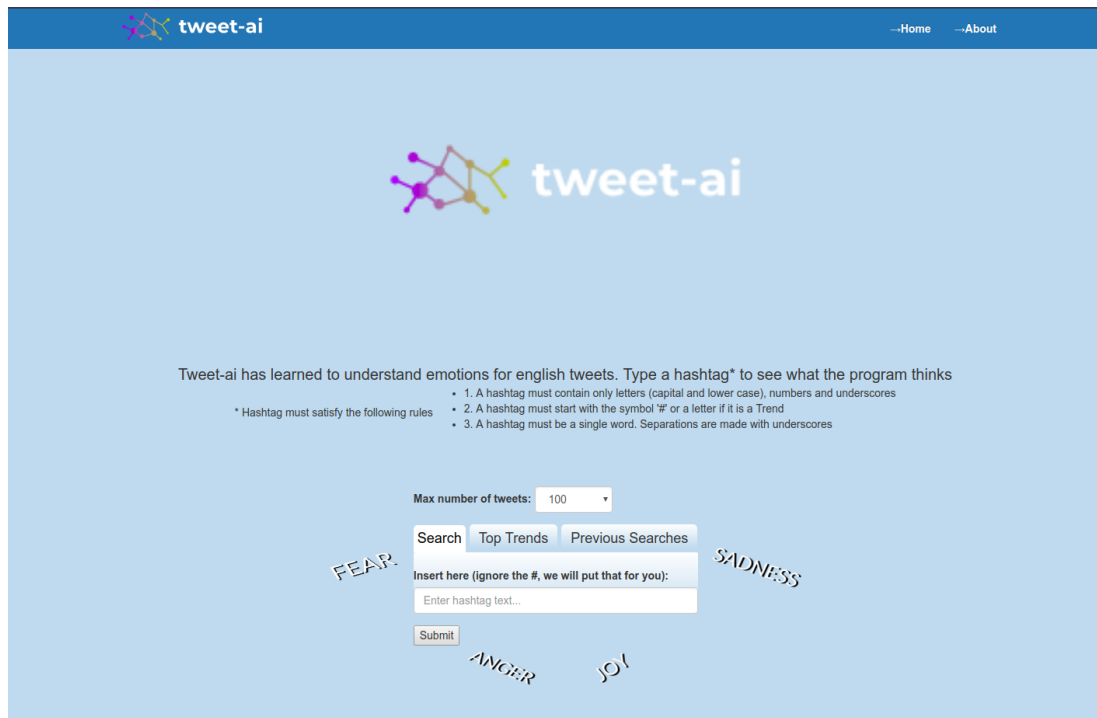
## Κεφάλαιο 7

# Διαδικτυακή Εφαρμογή

### 7.1 Αρχική Σελίδα

Μετά την υλοποίηση των νευρωνικών μας αποφασίσαμε να δημιουργήσουμε μία Διαδικτυακή Εφαρμογή η οποία θα πραγματοποιεί Συναισθηματική Ανάλυση σε αγγλικά tweets σε πραγματικό χρόνο. Η εφαρμογή ονομάστηκε **tweet-ai** που είναι η σύνδεση των λέξεων tweet και ai (Artificial Intelligence) ενώ ως λέξη αποτελεί τον Χρήστη του twitter που έγραψε ένα tweet στο παρελθόν:

**tweetai**: first-person singular past historic of tweeter, (πηγή Wiktionary <sup>1</sup>)

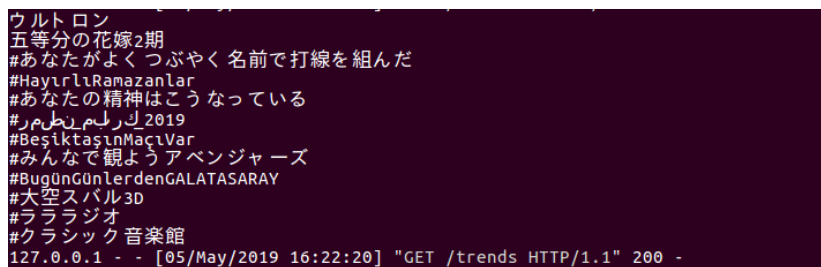


Σχήμα 7.1: Κύρια Σελίδα Διαδικτυακής Εφαρμογής

<sup>1</sup>[https://en.wiktionary.org/wiki/Wiktionary:Main\\_Page](https://en.wiktionary.org/wiki/Wiktionary:Main_Page)

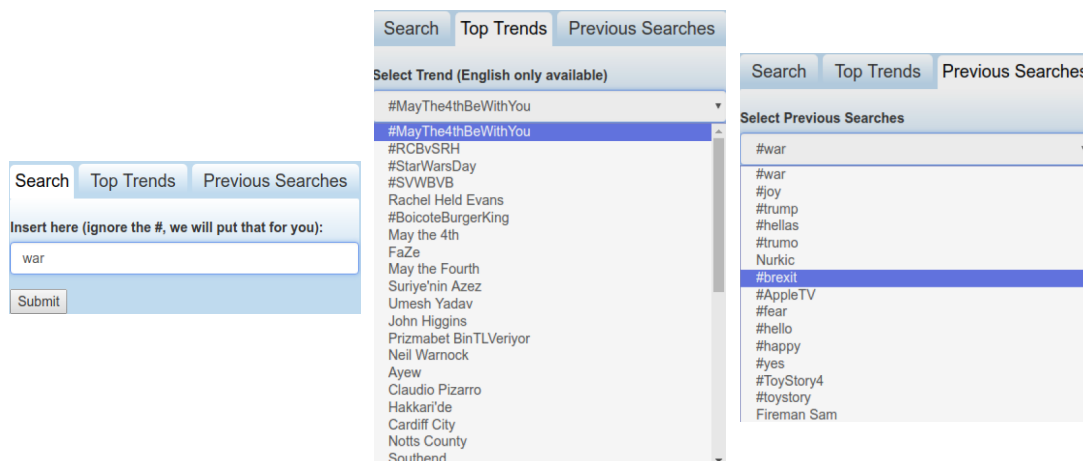
Υπάρχουν τρεις τρόποι με τους οποίους μπορεί ο χρήστης να επιλέξει τον όρο αναζήτησης όπου τα tweets που θα επιστραφούν θα εμπεριέχουν:

- **Ελεύθερη Επιλογή:** Όπου έχει την δυνατότητα να πληκτρολογήσει επακριβώς το hashtag ή το trend που αυτός επιθυμεί, αρκεί να τηρεί του κανόνες εισαγωγής που αναφέρονται στην κεντρική σελίδα, εάν πρόκειται να πληκτρολογήσει hashtag.
- **Επιλογή από trends:** Ο χρήστης έχει στην διάθεσή του μία λίστα με τα κορυφαία trends αγγλικών χαρακτήρων εκείνη την στιγμή στον κόσμο, τα οποία εξάγονται από το twitter με την βοήθεια του api του. Στο Σχήμα 7.2 βλέπουμε ορισμένα trends που φιλτράραμε μιας και δεν έχουν εξ ολοκλήρου αγγλικούς χαρακτήρες.



Σχήμα 7.2: Φιλτραρισμένα μη-αγγλικά trends

- **Προηγούμενες Αναζητήσεις:** Όπου μπορεί να δει μία λίστα με τις τελευταίες 15 αναζητήσεις που έχουν κάνει προηγούμενοι χρήστες στην εφαρμογή μας. Η λίστα αυτή ανανεώνεται με κάθε νέα αναζήτηση ενώ τα δεδομένα της αποθηκεύονται στη βάση δεδομένων μας (MongoDB)



Σχήμα 7.3: Μέθοδοι Εισαγωγής Όρου (Free Text, Top Trends, Previous Searches)

Στα Σχήματα 7.3 φαίνονται και οι τρεις τρόποι εισαγωγής. Μάλιστα οι φωτογραφίες τραβήχτηκαν στις 4 Μαΐου που είναι η Παγκόσμια Ημέρα StarWars οπότε βλέπουμε πως αρκετά από τα πρώτα trends αφορούν αυτή την ταινία, αποδεικνύοντας ότι το twitter επιστρέφει όντως σωστά και real time trends.



### 7.3 Καρτέλα κορυφαίων Tweets

Κατα την πρόβλεψη των συναισθημάτων θυμού, φόβου, χαράς και λύπης των αγγλικών tweets συγκρατούσαμε ταυτόχρονα και εκείνα τα οποία έχουν το υψηλότερο σκορ στο εκάστοτε συναίσθημα. Για να είναι φανερό σε ποιο συναίσθημα αναφερόμαστε χρησιμοποιήσαμε τα αντίστοιχα emoji για καλύτερη οπτικοποίηση. Επίσης στο κάτω μέρος κάθε κύκλου παρουσίασης του tweet με το μεγαλύτερο σκορ για το δεδομένο συναίσθημα προσθέσαμε και σκορ το οποίο διαθέτει στην συγκεκριμένη κατηγορία. Όπως και πριν με χρήση του κέρσορα (κλικ) μεταφερόμαστε στην σελίδα του twitter στην συγκεκριμένη δημοσίευση. Στο Σχήμα 7.5 που ακολουθεί βλέπουμε την συγκεκριμένη καρτέλα τα αποτελέσματα που προέκυψαν με όρο αναζήτησης το hashtag: *#what*.



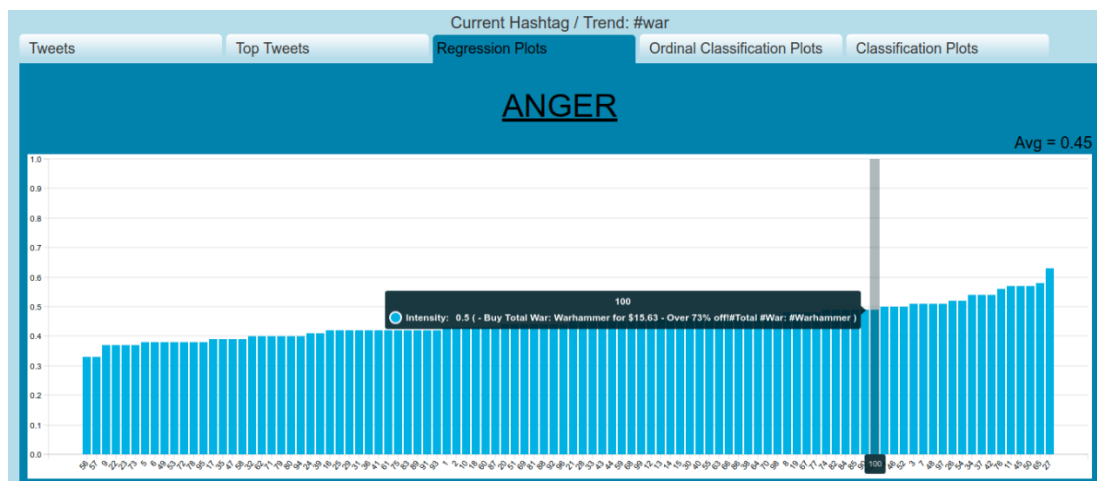
Σχήμα 7.5: Καρτέλα Παρουσίασης κορυφαίων Tweets

Είναι συχνό φαινόμενο σε συναισθήματα τα οποία δεν μοιάζουν κατ' ουσία αλλά συνήθως συναντώνται μαζί όπως είναι ο θυμός και η λύπη να έχουμε κοινά αποτελέσματα. Έτσι και στο παράδειγμα του σχήματος 7.5 βλέπουμε πως το κορυφαίο tweet τόσο για τον θυμό όσο και για την λύπη είναι το ίδιο. Θα μπορούσαμε να ελέγχουμε εάν έχουμε τέτοιου είδους σύγκρουση και να παίρνουμε για παράδειγμα το δεύτερο κορυφαίο tweet λύπης για να παρουσιάσουμε, όμως αυτό θα αποτελούσε αλλοίωση δεδομένων.



## 7.4 Καρτέλα Γραφημάτων Παλινδρόμησης

Έπειτα οδηγούμαστε στην συνολική παρουσίαση των δεδομένων. Πρόκειται για τις προβλέψεις των tweets που αφορούν το Task 1 της εργασίας μας, το επονομαζόμενο Emotion-Intensity Regression (EI-reg) στο οποίο εμβαθύνουμε περισσότερο από όλα. Εδώ παρουσιάζουμε για κάθε συναίσθημα την βαθμολογία για όλα τα tweets που συγκεντρώσαμε. Μάλιστα ταξινομούμε κάθε φορά τα tweets βάσει του αντίστοιχου συναισθήματος προκειμένου να έχουμε μία καλύτερη οπτικοποίηση των δεδομένων. Επιπλέον έχουμε υπολογίσει και τον μέσο όρο για κάθε συναίσθημα ώστε να έχουμε μία μέση ιδέα για την άποψη του κοινού πάνω στον δεδομένο όρο αναζήτησης. Τέλος με πέρασμα του κέρσορα (hover) πάνω από κάθε στήλη που αντιπροσωπεύει το σκορ κάθε tweet μπορούμε να δούμε το κείμενο του συγκεκριμένου tweet. Στο σχήμα 7.6 που ακολουθεί βλέπουμε τον γράφημα του συναισθήματος φόβου με όρο αναζήτησης το hashtag: #war.



Σχήμα 7.6: Καρτέλα Παρουσίασης Γραφημάτων Παλινδρόμησης

Για την παρουσίαση των γραφημάτων χρησιμοποιήθηκε η βιβλιοθήκη για ReactJS, React-Charts <sup>2</sup>.

Σημείωση: Κάτω από το γράφημα του συναισθήματος φόβου βρίσκονται με την σειρά και τα υπόλοιπα γραφήματα τα οποία όμως δεν δείχνουμε σε κάποια εικόνα για λόγους αποφυγής επανάληψης.

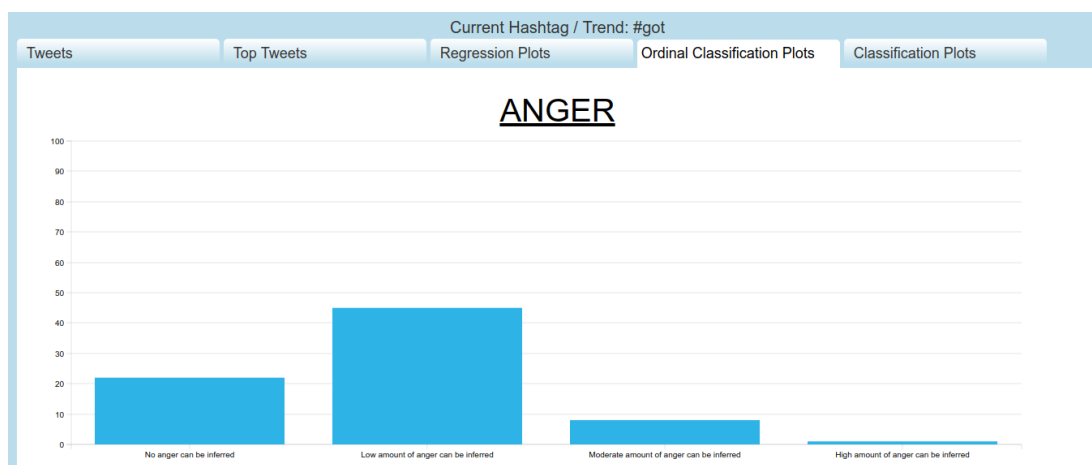
## 7.5 Καρτέλα Γραφημάτων Ταξινόμησης

Στην τέταρτη καρτέλα παρουσιάζονται τα γραφήματα που αφορούν το Task 2 του διαγωνισμού το οποίο όπως έχουμε αναφέρει ονομάζεται Emotion Intensity Ordinal Classification (EI-oc) και αφορά την κατηγοριοποίηση των tweets σε 4 βασικές κατηγορίες έντασης για κάθε συναίσθημα:

<sup>2</sup><https://react-charts.js.org/>

- Το συναίσθημα E δεν υπάρχει (No E can be inferred)
- Το συναίσθημα E υπάρχει σε μικρό βαθμό (Low amount of E can be inferred)
- Το συναίσθημα E υπάρχει σε μέτριο βαθμό (Moderate amount of E can be inferred)
- Το συναίσθημα E υπάρχει σε υψηλό βαθμό (High amount of E can be inferred)

Εν αντιθέση με την προηγούμενη καρτέλα, σε αυτήν δεν δείχνουμε για το κάθε tweet την τιμή που πήρε. Αυτό το κάνουμε γιατί κάθε tweet μπορεί να πάρει μόνο 4 διακριτές τιμές και μία οπτικοποίηση ανά tweet θα οδηγούσε σε μία μορφή που θα έμοιαζε με κανονικοποίηση των προηγούμενων γραφημάτων. Αντ' αυτού αθροίσαμε για κάθε κατηγορία το πλήθος των tweets που αντιστοιχίστηκαν σε αυτήν και πλέον κάθε στήλη είναι αυτό ακριβώς το άθροισμα. Με πέρασμα του κέρσορα (hover) πάνω από κάθε στήλη βλέπουμε και το ακριβές άθροισμα.

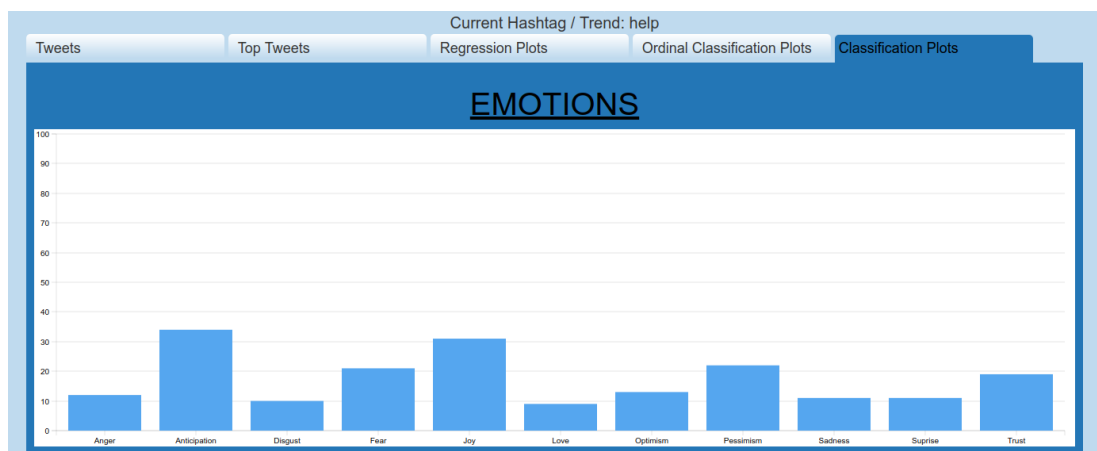


Σχήμα 7.7: Καρτέλα Παρουσίασης Γραφημάτων Ταξινόμησης

Τα δεδομένα που διαθέταμε για να εκπαιδεύσουμε το νευρωνικό μας καθώς και αυτά που χρησιμοποιήσαμε για αξιολόγηση περιείχαν αρκετά tweets από όλες τις κατηγορίες, για παράδειγμα είχαν ένα αρκετά ικανοποιητικό ποσοστό tweets τα οποία είχαν την ετικέτα *Το συναίσθημα E υπάρχει σε ψηλό βαθμό*. Είχαν δηλαδή καλή κατανομή. Αυτό όμως δεν είναι κάτι που συμβαίνει στην πραγματικότητα, δηλαδή τα περισσότερα tweets έχουν είτε λίγη ή και καθόλου ένταση από τα γνωστά συναισθήματα ως απί το πλήστον. Για τον λόγο αυτό χρησιμοποιήσαμε την πρόβλεψη παλινδρόμησης και ανάλογα με κάποια κατώφλια (thresholds) που είχαμε θέσει κρίναμε σε ποια κατηγορία ανήκει το κάθε tweet. Τα κατώφλια αυτά είναι [0.3, 0.4, 0.5, 1.0] για όλα τα συναισθήματα. Στο Σχήμα 7.7 που προηγήθηκε βλέπουμε την συγκεκριμένη καρτέλα για το συναίσθημα του θυμού και με όρο αναζήτησης το hashtag: #got (game of thrones).

## 7.6 Καρτέλα Γραφημάτων Θεματικής Ταξινόμησης

Στην τελευταία καρτέλα έχουμε το Task 3 που ονομάζεται Emotion Classification (E-c). Λόγω της φύσης του task, που αφορά το να προβλέπουμε ανάμεσα σε 11 συναισθηματικές κατηγορίες, σε ποιες ανήκει το κάθε tweet, ακολουθούμε και εδώ μία λογική οπτικοποίησης ανάλογη της προηγούμενης για να έχουμε πιο σαφή προς τον χρήστη δεδομένα. Όπως είναι λογικό μιας και κάθε tweet μπορεί να ανήκει σε όσες κατηγορίες θέλει, ακόμα και σε όλες, οι στήλες στην συγκεκριμένη περίπτωση δεν αθροίζονται στον αριθμό του πλήθους των tweets. Με πέρασμα του κέρσορα (hover) πάνω από κάθε στήλη βλέπουμε και το ακριβές άθροισμα. Στο Σχήμα 7.8 που ακολουθεί βλέπουμε μία τέτοια αναπαράσταση.



Σχήμα 7.8: Καρτέλα Παρουσίασης Γραφημάτων Συναισθημάτων

## 7.7 Συμπληρωματικές Πληροφορίες

Τα χρώματα τα οποία χρησιμοποιήθηκαν είναι οι μπλε και γαλάζιες αποχρώσεις που χρησιμοποιεί και το ίδιο το twitter. Η εφαρμογή είναι Single Page Application λόγω της React οπότε κάθε component που ανανεώνεται δεν οδηγεί σε ανανέωση όλης της σελίδας. Η εφαρμογή επίσης διαθέτει μία about σελίδα, όπως βλέπουμε στην αρχική σελίδα πάνω δεξιά, που περιγράφεται συνοπτικά η διαδικασία πρόβλεψης, καθώς και κάποιες πληροφορίες του δημιουργού.



## Κεφάλαιο 8

# Συμπεράσματα

Στην συγκεκριμένη Διπλωματική Εργασία ασχοληθήκαμε με την Συναισθηματική Ανάλυση σε Αγγλικά Tweets. Όπως αναφέρουμε στο Κεφάλαιο 2 η έρευνα που έχει γίνει στην Συναισθηματική Ανάλυση είναι αρκετά ενδεδεγμένη όμως η Συναισθηματική Ανάλυση που αφορά Affect, δηλαδή Επίδραση, που στην περίπτωσή μας είναι η ένταση Βασικών Συναισθημάτων, έχει αρχίσει να αποφέρει καρπούς μόλις την τελευταία δεκαετία. Στην περίπτωσή μας χρησιμοποιήσαμε τα δεδομένα που δόθηκαν από τον Διαγωνισμό Task 1: Affect in Tweets [1] του Συνεδρίου Semeval 2018. Ως πηγή άντλησης μεθόδων χρησιμοποιήθηκαν τα papers [82]–[85] άλλων συμμετεχόντων. Συγκεκριμένα ασχοληθήκαμε με την πρόβλεψη ποσοστού Συναισθήματος (Task 1, EI-reg), για τα Συναισθήματα Θυμός, Φόβος, Χαρά και Λύπη, με την ταξινόμηση σε 4 διακριτές κλάσεις (Task 2, EI-oc) των ίδιων Συναισθημάτων και με την πρόβλεψη ύπαρξης ή όχι 11 βασικών Συναισθημάτων (Task 5, E-c). Αυτά είναι τα 3 Tasks τα οποία αναφέρονται σε όλα τα προηγούμενα Κεφάλαια.

Για την υλοποίηση συνδιάστηκαν Μοντέλα και Αρχιτεκτονικές Μηχανικής αλλά και Βαθειάς Μάθησης που περιελάμβαναν χρήση Διανυσμάτων Λέξεων, Μηχανισμών Μακράς Βραχυπρόθεσμης Μνήμης διπλής Κατεύθυνσης, Μηχανισμούς Προσοχής, Συναισθηματικά Λεξιικά, Χαρακτηριστικά Κειμένων, Μηχανές Διανυσμάτων Υποσήριξης, Τυχαία Δάση, κτλ. Πριν προβούμε στην χρήση των συγκεκριμένων μεθόδων, το θεωρητικό υπόβαθρο των οποίων αναλύεται στο Κεφάλαιο 3 και 4, επεξεργαστήκαμε κατάλληλα το κείμενό μας ούτως ώστε να το φέρουμε στην επιθυμητή μορφή. Η διαδικασία προεπεξεργασίας αλλά και υλοποίησης αναπτύσσεται ενδεδεγμένα στα Κεφάλαια 6.

Στην συνέχεια χρησιμοποιήσαμε Γενετικούς Αλγορίθμους ώστε να συνδιάσουμε τις διαφορετικές προβλέψεις των ταξινομητών που είχαν τις καλύτερες επιδόσεις. Είδαμε ότι αυτή η μέθοδος είναι αρκετά πιο γρήγορη από χρήση νέου Νευρωνικού Δικτύου συνδιασμού μοντέλων όπως συνήθίζεται, μιας και μπορούμε κατευθείαν να βρούμε πολλαπλασιαστικά βάρη των προβλέψεων, πετυχαίνοντας αρκετά ικανοποιητικά αποτελέσματα. Επίσης χρησιμοποιήσαμε τους Γενετικούς Αλγορίθμους και πάλι για να αποφύγουμε την εκπαίδευση νέου νευρωνικού, μεταφέροντας τις προβλέψεις Παλινδρόμησης σε προβλέψεις Ταξινόμησης από το Task 1 στο Task 2.

Αν και δεν συμμετείχαμε στον Διαγωνισμό κατά την διάρκειά του έχουμε την ευκαιρία να συγκρίνουμε τα αποτελέσματά μας με τους υπόλοιπους διαγωνιζόμενους μιας και οι χρυσές ετικέτες (golden labels) είναι διαθέσιμες προς το κοινό. Στο Task 1 με το οποίο και ασχοληθήκαμε περισσότερο από όλα αναπτύσσοντας τους περισσότερους ταξινομητές πετύχαμε την 2η καλύτερη επίδοση ανάμεσα στους διαγωνιζόμενους πετυχαίνοντας μέσο όρο 78.5 και έχοντας μάλιστα στο Συναισθήμα του Φόβου την καλύτερη από αυτές (78.3). Η μετρική σύγκρισης είναι η Pearson Correlation. Στο Task 2 έχουμε την 2η καλύτερη επίδοση έχοντας μέσο όρο 67.2, ενώ είχαμε ιδιαίτερα καλά αποτελέσματα στο Task 3 αφού πετύχαμε τις καλύτερες επιδόσεις σε όλες τις μετρικές ( $accuracy = 59.9$ ,  $f1\_macro = 71.2$ ,  $f1\_micro = 56.8$ ).

Από τα παραπάνω καθίσταται σαφές πως πλέον διαθέτουμε την τεχνογνωσία και τις μεθόδους ούτως ώστε να μπορούμε να κάνουμε ικανοποιητικές προβλέψεις σε θέματα Συναισθηματικής Ανάλυσης και ιδιαίτερα στα 4 βασικά συναισθήματα. Παρόλα αυτά έχουμε ακόμα αρκετό δρόμο μπροστά μας στην πρόβλεψη πιο σύνθετων Συναισθημάτων όπως είναι φανερό και από τα αποτελέσματα στο Task 3, που ενώ έχουμε τα καλύτερα αποτελέσματα, ένα ποσοστό ακρίβειας (accuracy) της τάξης του 60% σίγουρα επιδέχεται βελτίωσης. Για να γίνει αυτό φυσικά θα πρέπει να δημιουργηθούν και τα κατάλληλα datasets. Αυτό είναι το πρώτο βήμα για να εξελίξουμε τις προβλέψεις μας.

Στη συνέχεια αποφασίσαμε να εξελίξουμε τις μεθόδους που αναπτύξαμε και να τις ενσωματώσουμε σε μία Διαδικτυακή Εφαρμογή που εμείς υλοποιήσαμε. Τα νευρωνικά δίκτυα και ο κλάδος της Μηχανικής και Βαθιάς Μάθησης έχουν ως στόχο να μας προσφέρουν ορισμένες δυνατότητες ανάλυσης δεδομένων που υπό άλλες συνθήκες δεν θα είχαμε. Το να διατηρούμε μόνο σε θεωρητικό επίπεδο τις Αρχιτεκτονικές και τα Μοντέλα που χρησιμοποιούμε, δεν οδηγεί σε εκπλήρωση αυτού του στόχου. Αυτός είναι και ο λόγος που σχεδιάσαμε και υλοποιήσαμε την Διαδικτυακή Εφαρμογή η οποία αναλύεται ενδελεχώς στο Κεφάλαιο 7 αλλά και επειδή παρατηρήσαμε πως μία τέτοια ανάλυση και οπτικοποίηση όπως η δική μας δεν υπάρχει αυτή την στιγμή στα πλαίσια του Διαδικτύου.

## 8.1 Συμπληρωματικές Σημειώσεις

Επόμενος στόχος μας αποτελεί η δημοσιοποίηση της Διαδικτυακής Πλατφόρμας στο διαδίκτυο κάτω από ένα public domain ούτως ώστε να μπορεί να είναι διαθέσιμη για χρήση από όποιον χρήστη επιθυμεί αλλά και για την βελτίωσή της μέσα από αλληλεπίδραση με άλλους χρήστες. Επίσης σημαντική είναι και η βελτίωση της ταχύτητας με την οποία λειτουργεί η εφαρμογή μιας και όπως έχουμε αναφέρει κάθε call χρειάζεται περίπου 20 δευτερόλεπτα, χρόνος που υπό φυσιολογικές συνθήκες ίσως είναι μεγάλος.

Επίσης όπως αναφέραμε και στο Κεφάλαιο 1 πολλά από τα tweets που λαμβάνουμε από το twitter μέσω του TwitterAPI δεν διαθέτουν χρήσιμη πληροφορία, είναι δηλαδή σκουπίδια (garbage), αλλοιώνοντας την πληροφορία που παρουσιάζεται στην εφαρμογή. Για τον λόγο αυτό θα ήταν ιδιαίτερα χρήσιμο να αναπτυχθεί ένα ενδιάμεσο νευρωνικό

δίκτυο (middleware) το οποίο θα κρίνει αν ένα tweet είναι άξιο πρόβλεψης και ανάλυσης ή όχι. Έτσι θα είχαμε σαφώς καλύτερα και πλουσιότερα σε πληροφορία αποτελέσματα.

Τέλος θα ήταν σημαντική και η περαιτέρω έρευνα πάνω σε άλλους τρόπους με τους οποίους θα μπορούσαν να φανούν χρήσιμοι οι Γενετικοί Αλγόριθμοι, χρησιμοποιώντας διαφορετικές Fitness Functions.





# Βιβλιογραφία

- [1] S. M. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, “Semeval-2018 Task 1: Affect in tweets”, in *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA, 2018.
- [2] P. Ekman, “An argument for basic emotions”, *Cognition Emotion*, vol. 6, pp. 169–200, May 1992. DOI: [10.1080/02699939208411068](https://doi.org/10.1080/02699939208411068).
- [3] R. Plutchik, “A psychoevolutionary theory of emotions”, *Social Science Information*, vol. 21, no. 4-5, pp. 529–553, 1982. DOI: [10.1177/053901882021004003](https://doi.org/10.1177/053901882021004003). [Online]. Available: <https://doi.org/10.1177/0539018820210040030>.
- [4] W. Parrot, *Emotions in Social Psychology : Essential Readings*. 2001.
- [5] N. H. Frijda, “The laws of emotion”, *American Psychologist*, vol. 43, p. 349, May 1988. DOI: [10.1037/0003-066X.43.5.349](https://doi.org/10.1037/0003-066X.43.5.349).
- [6] B. Liu, “Sentiment analysis: Mining sentiments, opinions, and emotions”, in. Nov. 25, 2016. [Online]. Available: <https://www.cs.uic.edu/~liub/FBS/sentiment-opinion-emotion-analysis.html>.
- [7] T. Nasukawa and J. Yi, “Sentiment analysis: Capturing favorability using natural language processing”, Jan. 2003, pp. 70–77. DOI: [10.1145/945645.945658](https://doi.org/10.1145/945645.945658).
- [8] J. M. Wiebe, *Learning subjective adjectives from corpora*. [Online]. Available: <https://people.cs.pitt.edu/~wiebe/pubs/papers/aaai2000.pdf>.
- [9] S. R. Das and M. Y. Chen, *Yahoo! For Amazon: Sentiment Parsing from Small Talk on the Web*. Aug. 2001. [Online]. Available: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=276189](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=276189).
- [10] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, “Mining product reputations on the web”, Jul. 2002. DOI: [10.1145/775047.775098](https://doi.org/10.1145/775047.775098).
- [11] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques”, *EMNLP*, vol. 10, Jun. 2002. DOI: [10.3115/1118693.1118704](https://doi.org/10.3115/1118693.1118704).

- [12] P. D. Turney, “Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews”, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, pp. 417–424. DOI: [10.3115/1073083.1073153](https://doi.org/10.3115/1073083.1073153). [Online]. Available: <https://doi.org/10.3115/1073083.1073153>.
- [13] M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora”, in, ch. 2. [Online]. Available: <https://www.aclweb.org/anthology/C92-2082>.
- [14] J. M. Wiebe, *Tracking point of view in narrative*. [Online]. Available: <http://people.cs.pitt.edu/~wiebe/pubs/papers/CL94.pdf>.
- [15] V. Hatzivassiloglou and K. R. Mckeown, “Predicting the semantic orientation of adjectives”, May 2002. DOI: [10.3115/979617.979640](https://doi.org/10.3115/979617.979640).
- [16] C. Alm, D. Roth, and R. Sproat, “Emotions from text: Machine learning for text-based emotion prediction”, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 347–354, Jan. 2005.
- [17] M. Brooks, K. Kuksenok, M. K. Torkildson, D. Perry, J. J. Robinson, T. J. Scott, O. Anicello, A. Zukowski, P. Harris, and C. R. Aragon, “Statistical affect detection in collaborative chat”, in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13, San Antonio, Texas, USA: ACM, 2013, pp. 317–328. DOI: [10.1145/2441776.2441813](https://doi.org/10.1145/2441776.2441813). [Online]. Available: <http://doi.acm.org/10.1145/2441776.2441813>.
- [18] M. Lemmon and E. V. Portniaguina, “Consumer confidence and asset prices: Some empirical evidence”, *Review of Financial Studies*, vol. 19, pp. 1499–1529, Feb. 2006. DOI: [10.2139/ssrn.335240](https://doi.org/10.2139/ssrn.335240).
- [19] E. Gilbert and K. Karahalios, “Predicting tie strength with social media”, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09, Boston, MA, USA: ACM, 2009, pp. 211–220. DOI: [10.1145/1518701.1518736](https://doi.org/10.1145/1518701.1518736). [Online]. Available: <http://doi.acm.org/10.1145/1518701.1518736>.
- [20] C. Strapparava and R. Mihalcea, “Learning to identify emotions in text”, in *Proceedings of the 2008 ACM Symposium on Applied Computing*, ser. SAC '08, Fortaleza, Ceara, Brazil: ACM, 2008, pp. 1556–1560. DOI: [10.1145/1363686.1364052](https://doi.org/10.1145/1363686.1364052). [Online]. Available: <http://doi.acm.org/10.1145/1363686.1364052>.
- [21] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision”, *Processing*, vol. 150, Jan. 2009.
- [22] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining”, vol. 10, Jan. 2010.

- [23] S. Mohammad and S. Kiritchenko, “Using hashtags to capture fine emotion categories from tweets”, *Computational Intelligence*, vol. 31, Jan. 2014. DOI: [10.1111/coin.12024](https://doi.org/10.1111/coin.12024).
- [24] S. M. Mohammad and F. Bravo-Marquez, “Emotion intensities in tweets”, *CoRR*, vol. abs/1708.03696, 2017. [Online]. Available: <http://arxiv.org/abs/1708.03696>.
- [25] S. Kiritchenko and S. Mohammad, “Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation”, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 465–470. DOI: [10.18653/v1/P17-2074](https://doi.org/10.18653/v1/P17-2074). [Online]. Available: <https://www.aclweb.org/anthology/P17-2074>.
- [26] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997, p. 2.
- [27] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998, p. 37.
- [28] T. K. Ho, “Random decision forests”, in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, Aug. 1995, 278–282 vol.1. DOI: [10.1109/ICDAR.1995.598994](https://doi.org/10.1109/ICDAR.1995.598994).
- [29] L. Breiman, “Random forests”, *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324). [Online]. Available: <https://doi.org/10.1023/A:1010933404324>.
- [30] L. Breiman, *Bagging predictors*, 1994.
- [31] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943. DOI: [10.1007/BF02478259](https://doi.org/10.1007/BF02478259). [Online]. Available: <https://doi.org/10.1007/BF02478259>.
- [32] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. New York: Wiley, Jun. 1949.
- [33] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain”, *Psychological Review*, pp. 65–386, 1958.
- [34] H. J. Kelley, “Gradient theory of optimal flight paths”, *ARS Journal*, vol. 30, no. 10, pp. 947–954, 1960. DOI: [10.2514/8.5282](https://doi.org/10.2514/8.5282). [Online]. Available: <https://doi.org/10.2514/8.5282>.
- [35] P. J. Werbos, “Beyond regression: New tools for prediction and analysis in the behavioral sciences”, PhD thesis, Harvard University, 1974.

- [36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors”, vol. 323, pp. 533–536, Oct. 1986. DOI: [10.1038/323533a0](https://doi.org/10.1038/323533a0).
- [37] H. Robbins and S. Monro, “A stochastic approximation method”, *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951. DOI: [10.1214/aoms/1177729586](https://doi.org/10.1214/aoms/1177729586). [Online]. Available: <https://doi.org/10.1214/aoms/1177729586>.
- [38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Neurocomputing: Foundations of research”, in J. A. Anderson and E. Rosenfeld, Eds., Cambridge, MA, USA: MIT Press, 1988, ch. Learning Representations by Back-propagating Errors, pp. 696–699. [Online]. Available: <http://dl.acm.org/citation.cfm?id=65669.104451>.
- [39] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization”, *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1953048.2021068>.
- [40] M. D. Zeiler, “ADADELTA: an adaptive learning rate method”, *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: <http://arxiv.org/abs/1212.5701>.
- [41] D. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *International Conference on Learning Representations*, Dec. 2014.
- [42] K. Pearson, “Note on regression and inheritance in the case of two parents”, *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895. [Online]. Available: <http://www.jstor.org/stable/115794>.
- [43] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks”, *IEEE Trans. Signal Processing*, vol. 45, pp. 2673–2681, 1997.
- [44] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult”, *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994. DOI: [10.1109/72.279181](https://doi.org/10.1109/72.279181).
- [45] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [46] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation”, *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>.
- [47] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE”, *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950. DOI: [10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433). [Online]. Available: <https://doi.org/10.1093/mind/LIX.236.433>.

- [48] T. Winograd, *Shrdlu*, 1968-1970. [Online]. Available: <http://hci.stanford.edu/winograd/shrdlu/>.
- [49] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine”, *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966. DOI: [10.1145/365153.365168](https://doi.org/10.1145/365153.365168). [Online]. Available: <http://doi.acm.org/10.1145/365153.365168>.
- [50] M. Porter, “An algorithm for suffix stripping”, *Program*, vol. 14, no. 3, pp. 130–137, 1980. DOI: [10.1108/eb046814](https://doi.org/10.1108/eb046814). [Online]. Available: <https://doi.org/10.1108/eb046814>.
- [51] Z. S. Harris, “Distributional structure”, *WORD*, vol. 10, no. 2-3, pp. 146–162, 1954. DOI: [10.1080/00437956.1954.11659520](https://doi.org/10.1080/00437956.1954.11659520). [Online]. Available: <https://doi.org/10.1080/00437956.1954.11659520>.
- [52] H. P. Luhn, “A statistical approach to mechanized encoding and searching of literary information”, *IBM Journal of Research and Development*, vol. 1, no. 4, pp. 309–317, Oct. 1957. DOI: [10.1147/rd.14.0309](https://doi.org/10.1147/rd.14.0309).
- [53] K. S. JONES, “A statistical interpretation of term specificity and its application in retrieval”, *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972. DOI: [10.1108/eb026526](https://doi.org/10.1108/eb026526). [Online]. Available: <https://doi.org/10.1108/eb026526>.
- [54] J. B. Carroll, *Language*, vol. 35, no. 1, pp. 58–77, 1959. [Online]. Available: <http://www.jstor.org/stable/411335>.
- [55] G. A. Miller, “Wordnet: A lexical database for english”, *COMMUNICATIONS OF THE ACM*, vol. 38, pp. 39–41, 1995.
- [56] A. Lavelli, F. Sebastiani, and R. Zanoli, “Distributional term representations: An experimental comparison”, in *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, ser. CIKM '04, Washington, D.C., USA: ACM, 2004, pp. 615–624. DOI: [10.1145/1031171.1031284](https://doi.org/10.1145/1031171.1031284). [Online]. Available: <http://doi.acm.org/10.1145/1031171.1031284>.
- [57] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model”, *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003. [Online]. Available: <http://dl.acm.org/citation.cfm?id=944919.944966>.
- [58] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient estimation of word representations in vector space”, Jan. 2013, pp. 1–12.
- [59] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation”, in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: <http://www.aclweb.org/anthology/D14-1162>.

- [60] J. Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, Mass: MIT Press, 1992.
- [61] F. Chollet, *Keras*, <https://github.com/keras-team/keras>, 2015.
- [62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine learning in python”, *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1953048.2078195>.
- [63] R. T. Fielding, “Architectural styles and the design of network-based software architectures”, AAI9980887, PhD thesis, 2000.
- [64] D. Winer, D. Box, B. Atkinson, and M. Al-Ghosein, “Soap: Simple object access protocol”, in. 1998. [Online]. Available: <https://tools.ietf.org/html/draft-box-http-soap-01>.
- [65] A. Ronacher, *Opening the flask*, 2011. [Online]. Available: <http://mitsuhiko.pocoo.org/flask-pycon-2011.pdf>.
- [66] G. E. Krasner and S. T. Pope, “A cookbook for using the model-view controller user interface paradigm in smalltalk-80”, *J. Object Oriented Program.*, vol. 1, no. 3, pp. 26–49, Aug. 1988. [Online]. Available: <http://dl.acm.org/citation.cfm?id=50757.50759>.
- [67] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterton, Eds., ser. Proceedings of Machine Learning Research, vol. 9, Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: <http://proceedings.mlr.press/v9/glorot10a.html>.
- [68] S. M. Mohammad and S. Kiritchenko, “Understanding emotions: A dataset of tweets to study interactions between affect categories”, in *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan, 2018.
- [69] J. J. Louviere and G. G. Woodworth, “Best-worst scaling: A model for the largest difference judgments”, *University of Alberta: Working Paper*, 1991.
- [70] P. M. Roget, in *Roget’s Thesaurus*, Apr. 1852.
- [71] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora”, English, in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, <http://is.muni.cz/publication/884893/en>, Valtetta, Malta: ELRA, May 2010, pp. 45–50.

- [72] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm”, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1615–1625. DOI: [10.18653/v1/D17-1169](https://doi.org/10.18653/v1/D17-1169). [Online]. Available: <https://www.aclweb.org/anthology/D17-1169>.
- [73] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis”, in *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada, 2005*, pp. 347–354. [Online]. Available: <http://aclweb.org/anthology/H/H05/H05-1044.pdf>.
- [74] M. Hu and B. Liu, “Mining and summarizing customer reviews”, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, 2004, pp. 168–177. DOI: [10.1145/1014052.1014073](https://doi.org/10.1145/1014052.1014073). [Online]. Available: <https://doi.org/10.1145/1014052.1014073>.
- [75] F. Å. Nielsen, “A new ANEW: evaluation of a word list for sentiment analysis in microblogs”, in *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Heraklion, Crete, Greece, May 30, 2011*, 2011, pp. 93–98. [Online]. Available: [http://ceur-ws.org/Vol-718/paper%5C\\_16.pdf](http://ceur-ws.org/Vol-718/paper%5C_16.pdf).
- [76] X. Z. Svetlana Kiritchenko and S. M. Mohammad, “Sentiment analysis of short informal texts”, vol. 50, pp. 723–762, 2014.
- [77] S. M. Mohammad and P. D. Turney, “Crowdsourcing a word-emotion association lexicon”, vol. 29, no. 3, pp. 436–465, 2013.
- [78] F. Bravo-Marquez, E. Frank, S. M. Mohammad, and B. Pfahringer, “Determining word-emotion associations from tweets by multi-label classification”, in *2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016*, 2016, pp. 536–539. DOI: [10.1109/WI.2016.0091](https://doi.org/10.1109/WI.2016.0091). [Online]. Available: <https://doi.org/10.1109/WI.2016.0091>.
- [79] S. M. Mohammad and S. Kiritchenko, “Using hashtags to capture fine emotion categories from tweets”, *Computational Intelligence*, vol. 31, no. 2, pp. 301–326, 2015. DOI: [10.1111/coin.12024](https://doi.org/10.1111/coin.12024). [Online]. Available: <http://dx.doi.org/10.1111/coin.12024>.



- [80] S. Baccianella, A. Esuli, and F. Sebastiani, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining”, in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*, 2010. [Online]. Available: <http://www.lrec-conf.org/proceedings/lrec2010/summaries/769.html>.
- [81] M. Thelwall, K. Buckley, and G. Paltoglou, “Sentiment strength detection for the social web”, *JASIST*, vol. 63, no. 1, pp. 163–173, 2012. DOI: [10.1002/asi.21662](https://doi.org/10.1002/asi.21662). [Online]. Available: <https://doi.org/10.1002/asi.21662>.
- [82] V. Duppada, R. Jain, and S. Hiray, “Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets”, *CoRR*, vol. abs/1804.06137, 2018. [Online]. Available: <http://arxiv.org/abs/1804.06137>.
- [83] J. H. Park, P. Xu, and P. Fung, “Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and #hashtags”, *CoRR*, vol. abs/1804.08280, 2018. [Online]. Available: <http://arxiv.org/abs/1804.08280>.
- [84] G. Gee and E. Wang, “PsyML at SemEval-2018 task 1: Transfer learning for sentiment and emotion analysis”, in *Proceedings of The 12th International Workshop on Semantic Evaluation*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 369–376. DOI: [10.18653/v1/S18-1056](https://doi.org/10.18653/v1/S18-1056). [Online]. Available: <https://www.aclweb.org/anthology/S18-1056>.
- [85] C. Baziotis, N. Athanasiou, A. Chronopoulou, A. Kolovou, G. Paraskevopoulos, N. Ellinas, S. Narayanan, and A. Potamianos, “NTUA-SLP at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning”, *CoRR*, vol. abs/1804.06658, 2018. [Online]. Available: <http://arxiv.org/abs/1804.06658>.