



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Focused Crawling Ethnopharmacological References with Active and Reinforcement Learning

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΟΝΤΟΓΙΑΝΝΗ Γ. ΑΝΔΡΕΑ

Επιβλέποντες: Αλέξανδρος Ποταμιάνος, Αναπληρωτής Καθηγητής ΕΜΠ
Γεώργιος Γιαννακόπουλος, ΕΚΕΦΕ "Δημόκριτος"

Αθήνα, Μάρτιος 2021



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Focused Crawling Ethnopharmacological References with Active and Reinforcement Learning

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΟΝΤΟΓΙΑΝΝΗ Γ. ΑΝΔΡΕΑ

Επιβλέποντες: Αλέξανδρος Ποταμιάνος, Αναπληρωτής Καθηγητής ΕΜΠ
Γεώργιος Γιαννακόπουλος, ΕΚΕΦΕ “Δημόκριτος”

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Μαρτίου 2021.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Αλέξανδρος Ποταμιάνος
Αναπληρωτής Καθηγητής ΕΜΠ

.....
Ιωάννα Ρουσσάκη
Επίκουρος Καθηγήτρια ΕΜΠ

.....
Παναγιώτης Τσανακάς
Καθηγητής ΕΜΠ

Αθήνα, Μάρτιος 2021



Copyright © – All rights reserved. Με την επιφύλαξη παντός δικαιώματος.
Ανδρέας Κοντογιάννης, 2021.

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.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Ανδρέας Κοντογιάννης

9 Απριλίου 2020

Περίληψη

Η Εθνοφαρμακολογία είναι η επιστημονική μελέτη της χρήσης φυτικών φαρμάκων από εθνικές ομάδες. Ούσα κλάδος της παραδοσιακής ιατρικής, θεωρείται ευρέως ως μία εναλλακτική μέθοδος για συμπληρωματική θεραπεία ασθενειών ακόμα και στους γνωστούς ιατρικούς κλάδους του δυτικού κόσμου [1]. Ωστόσο, η αναζήτηση και η τεκμηρίωση - από τους ίδιους τους επιστήμονες - της τοπικής γνώσης για συγκεκριμένες ιδιότητες των φυτών καθίσταται αρκετά δύσκολη, δεδομένου του όγκου της πληροφορίας που μοιράζεται στην επιστημονική βιβλιογραφία.

Απαίτηση της επιστημονικής έρευνας είναι να μπορεί κάποιος να αναζητά αποδοτικά κείμενα σχετικά με το αντικείμενό του. Τέτοιου είδους προβλήματα μπορούν να αντιμετωπιστούν ως προβλήματα εστιασμένης αναζήτησης στο Διαδίκτυο. Για την υποστήριξη των ειδικών προτείνουμε την αξιοποίηση ευφυών συστημάτων εστιασμένης αναζήτησης, γνωστά ως *focused crawlers*. Ένα τέτοιο σύστημα λαμβάνει ως είσοδο μερικά αρχικά κείμενα/URLs ή/και ορισμένες λέξεις-κλειδιά, τα οποία είναι όλα σχετικά με ένα προκαθορισμένο θέμα αναζήτησης. Στόχος του *focused crawler* είναι να ανακαλύψει και να εξάγει όσο το δυνατόν περισσότερες σχετικές ιστοσελίδες με το εν λόγω θέμα.

Στην παρούσα διπλωματική εργασία, δημιουργούμε ευφυή *focused crawler* συστήματα, έτσι ώστε αυτά να αποτελέσουν υποστηρικτικά εργαλεία για την βιβλιογραφική εθνοφαρμακολογική αναζήτηση. Προτείνουμε το Πρότυπο Ερευνητή-Μαθητή στη Μηχανική Μάθηση, μέσα από το οποίο αναπτύσσουμε έναν *focused crawler* δύο σταδίων. Στο πρώτο στάδιο, συνιστούμε τη χρήση Ενεργού Μάθησης: το σύστημα εκπαιδεύεται, ώστε να αναγνωρίζει εάν ένα κείμενο είναι σχετικό με το θέμα αναζήτησης, με κατάλληλη ανατροφοδότηση (*feedback*) από τον ερευνητή. Στο δεύτερο στάδιο, προτείνουμε τη χρήση Ενισχυτικής Μάθησης, θεωρώντας τον *focused crawler* ως έναν ευφυή πράκτορα. Ο πράκτορας εκτιμά το πόσο ωφέλιμο είναι μακροχρόνια να επισκεφτεί καθένα από τα δυνατά URLs και επιλέγει τα πιο υποσχόμενα από αυτά.

Στο πλαίσιο της Ενισχυτικής Μάθησης, μοντελοποιούμε το περιβάλλον ως μία Μαρκοβιάνη Διαδικασία Λήψης Αποφάσεων, θεωρώντας κοινές αναπαραστάσεις ανάμεσα στις καταστάσεις και τις δράσεις του πράκτορα. Τα χαρακτηριστικά των αναπαραστάσεων αποτελούνται από τα διανύσματα λέξεων (*word embeddings*) των τίτλων των αναφορών, στατιστικά χαρακτηριστικά της ιστοσελίδας, λέξεις-κλειδιά ή/και εκτιμήσεις των προεκπαιδευμένων μοντέλων του πρώτου σταδίου. Επιπροσθέτως, θεωρούμε περιπτώσεις στις οποίες το μοντέλο Ενεργού Μάθησης, που εκπαιδεύτηκε στο πρώτο στάδιο, χρησιμοποιείται ως συνάρτηση επιβράδευσης.

Αξιολογούμε το σύστημά μας σε δύο διαφορετικά προβλήματα αναζήτησης: ένα γενικό, βασισμένο σε αρχικά κείμενα και ένα πιο ειδικό, βασισμένο σε αρχικά κείμενα και λέξεις-

κλειδιά. Συγκρίνουμε 6 διαφορετικά μοντέλα Ενεργού Μάθησης, όπως το MarginSVM και το DoubleLSTM, 3 διαφορετικές κοινές αναπαραστάσεις ανάμεσα στις καταστάσεις και τις δράσεις του πράκτορα (General/Keyword/Only NLP Representation) και 2 πράκτορες - τον Deep Q Network (DQN) και τον Double DQN (DDQN).

Ο focused crawler δύο σταδίων με χρήση των πρακτόρων DQN ή/και DDQN είναι πιο αποτελεσματικός σε σχέση με επίπεδα αναφοράς, όπως το τυχαίο πιλοτικό τρέξιμο και ένας άπληστος νιτερμινιστικός focused crawler που ορίζουμε. Τέλος, συγκρίνοντας τη μέθοδό μας στο ειδικό πρόβλημα, με την εκτιμώμενη ικανότητα του εμπειρογνώμονα για αναζήτηση των αντίστοιχων κειμένων σε πραγματικό χρόνο, ξεπερνάμε κατά 5,14 φορές την αποδοτικότητα και 3,31 φορές την αποτελεσματικότητα του ερευνητή.

Λέξεις Κλειδιά

Εστιασμένη Διαδικτυακή Ανίχνευση, Ενισχυτική Μάθηση, Ενεργός Μάθηση, Επιβλεπόμενη Μάθηση, Βαθιά Μάθηση, Deep Q-learning, Double Deep Q-learning, Αναδρομικά Νευρωνικά Δίκτυα, Πολυεπίπεδα Perceptron, Μηχανές Διανυσμάτων Υποστήριξης, Συμφωνία Επισημειωτών, Επιλογή Καλύτερου Μοντέλου, word2vec

Abstract

Ethnopharmacology is the scientific study of ethnic groups and their use of herbal medicines. It - being a particular field of traditional medicine - is now widely considered as a promising alternative medicine for complementary treatment of the well-known western world [1]. However, the search and documentation of indigenous knowledge on the use of specific plant properties by the experts themselves is a very challenging task, taking into account the volume of information shared through ethnopharmacological literature.

Scientific research requires anyone to be able to efficiently search for relevant documents related to their subjects. These kinds of challenges can be faced as Internet focused search problems. To support experts, we propose the use of intelligent focused search systems, known as focused crawlers. Typically, such a system receives a few initial seed documents/URLs and optionally some keywords as input, all of which are relevant to a predefined search topic. The goal of a focused crawler is to discover and output as many relevant webpages as possible.

In the present thesis, we develop intelligent focused crawler systems, so that they become supportive tools for the ethnopharmacological research. We propose a two-stage Machine Learning focused crawler that follows a Researcher-Apprentice paradigm. In the first stage, we recommend the use of Active Learning (AL); the system is trained to identify the relevant documents by receiving feedback from the researcher, when that is needed. In the second stage, we propose the use of Reinforcement Learning (RL), regarding the focused crawler as an intelligent agent. The agent estimates how profitable would be to follow the available URLs, in the long term, and selects the most promising ones.

In the RL framework, we model the focused crawler environment as a Markov Decision Process (MDP), considering shared representations between the states and the actions of the agent. The representation features consist of the publication title word embeddings, statistical features extracted from the link structure, keywords and/or relevance predictions of the pretrained models from the first stage. Additionally, we consider cases where the AL model, trained in the first stage, is used as the reward function.

We evaluate two different search problems; one general, based on initial seed documents and one more specific, based on initial seed documents along with keywords. We compare 6 different AL models, such as the MarginSVM and the DoubleLSTM, 3 different state-action shared representations (General/Keyword/Only NLP Representation) and 2 RL agents; the Deep Q-Network (DQN) and the Double DQN (DDQN).

The two-stage focused crawler with the use of DQN, as well as DDQN, agent is more effective than baseline methods, such as random crawling and a greedy deterministic focused crawler we defined. Finally, comparing our method on the more specific setting

to an estimated real-time researcher performance, we outperform 5.14 times the efficiency and 3.31 times the effectiveness of the expert.

Keywords

Focused Crawling, Reinforcement Learning, Active Learning, Supervised Learning, Deep Learning, Deep Q-learning, Double Deep Q-learning, Recurrent Neural Networks, Multi-layer Perceptron, Support Vector Machines, Inter-annotator agreement, Model Selection, word2vec

στην οικογένειά μου

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω πολύ τον κ.Αλέξανδρο Ποταμιάνο, που μου έδωσε τη δυνατότητα για εκπόνηση αυτής της διπλωματικής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα και μέντορά μου κ.Γεώργιο Γιαννακόπουλο, για την εξαιρετική συνεργασία που αναπτύξαμε, παρά τις δυσκολίες που έχει προκαλέσει η κατάσταση με τον COVID-19. Ακόμα, θα ήθελα να ευχαριστήσω θερμά τον Ε.Αζιώτη και τον Ι.Σαμανίδη για τη πολύτιμη βοήθειά τους καθόλη τη διάρκεια της διπλωματικής μου.

Ευχαριστώ πολύ την κα.Ιωάννα Ρουσσάκη και τον κ.Παναγιώτη Τσανάκα, για την συμμετοχή τους στην απαιτούμενη τριμελή επιτροπή της εργασίας.

Ακόμα, ευχαριστώ πολύ τον Δ.Κακλή, που ήταν συνεπιβλέπωντας της διπλωματικής μου, και τον καλό μου φίλο, και συμφοιτητή, Βασίλη Π. για τις ωραίες μας συζητήσεις και την θετική επιρροή που μου έχει ασκήσει τα τελευταία χρόνια των σπουδών μου.

Ευχαριστώ πολύ την οικογένεια μου, Γεώργιο, Νικολέττα και Ταξιάρχη για την αγαπή, τη στήριξη και τις θυσίες που έχουν κάνει όλα αυτά τα χρόνια για μένα. Θα ήθελα να ευχαριστήσω τη Γεωργία, που είναι δίπλα μου σε κάθε μου βήμα. Τέλος, ευχαριστώ όλους τους φίλους και πρώην συμφοιτητές μου - Τάκη Π., Νίκο Σ., Βαγγέλη Χ., Μάριο Γ., Κώστα Π., Δημήτρη Κ., Νίκη Γ., Στέλιο Κ. και Γιάννη Σ. - για τα πολύ ωραία 5 χρόνια που περάσαμε μαζί...

Αθήνα, Μάρτιος 2021

Ανδρέας Κουτογιάννης

Περιεχόμενα

Περίληψη	1
Abstract	3
Ευχαριστίες	13
1 Εισαγωγή	29
1.1 Κίνητρα Μελέτης	30
1.2 Αντικείμενο της διπλωματικής	31
1.3 Συνεισφορές	31
1.4 Οργάνωση του τόμου	33
2 Βασικές Έννοιες	35
2.1 Εισαγωγή στη Μηχανική Μάθηση	35
2.1.1 Επιβλεπόμενη Μάθηση	36
2.1.2 Μη Επιβλεπόμενη Μάθηση	37
2.1.3 Ενισχυτική Μάθηση	37
2.2 Μοντέλα Επιβλεπόμενης Μάθησης	38
2.2.1 Μηχανές Διανυσμάτων Υποστήριξης	38
2.2.2 Αλγόριθμος Perceptron	42
2.2.3 Τεχνητά Νευρωνικά Δίκτυα	43
2.2.3.1 Πολυεπίπεδο Perceptron	44
2.2.3.2 Συναρτήσεις Ενεργοποίησης	45
2.2.3.3 Αναδρομικά Νευρωνικά Δίκτυα	47
2.2.3.4 Υπερπαράμετροι Μάθησης	50
2.2.3.5 Το Πρόβλημα της Υπερεκπαίδευσης	50
2.3 Ενεργός Μάθηση	51
2.3.1 Διαφορές Με Ημιεπιβλεπόμενη Μάθηση	51
2.3.2 Κατηγορίες Ενεργού Μάθησης	52
2.3.3 Στρατηγικές Ενεργού Μάθησης	53
2.3.3.1 Στρατηγικές Βασισμένες στην Ετερογένεια	53
2.3.3.2 Στρατηγικές Βασισμένες στην Επίδοση	54
2.3.3.3 Στρατηγικές Βασισμένες στην Αντιπροσωπευτικότητα	54
2.4 Ενισχυτική Μάθηση	54
2.4.1 Επιβράβευση και Ανταμοιβή	55
2.4.2 Μαρκοβιανή Διαδικασία Λήψης Αποφάσεων (MDP)	56

2.4.2.1	Στρατηγικές και Συναρτήσεις Αξίας	57
2.4.2.2	Κατηγοριοποίηση Στρατηγικών	58
2.4.3	Εξερεύνηση vs Εκμετάλλευση	58
2.4.4	Μέθοδοι Monte Carlo	58
2.4.5	Μέθοδοι Temporal Difference	59
2.4.5.1	Q-learning	59
2.4.5.2	Double Q-learning	60
2.4.5.3	Deep Q-learning	61
2.4.5.4	Double Deep Q-learning	62
2.5	Διαχείριση Κειμένων και Επεξεργασία Φυσικής Γλώσσας	62
2.5.1	Προεπεξεργασία Κειμένου	63
2.5.2	Ανάκτηση Πληροφορίας με το Μοντέλο Λογικής Μπουλ	63
2.5.3	Διανυσματικές Αναπαραστάσεις Λέξεων και Κειμένων	64
2.5.3.1	Δείκτες του λεξικού	64
2.5.3.2	One-Hot Vectors	65
2.5.3.3	Διάνυσμα κειμένου με χρήση tfidf	65
2.5.3.4	Το word2vec μοντέλο	66
2.5.3.5	Το Glove μοντέλο	67
2.6	Συμφωνία Επισημειωτών	67
2.6.1	Raw Agreement	68
2.6.2	Cohen's kappa	69
2.6.3	Krippendorff's alpha	69
2.7	Στατιστικός Έλεγχος	70
2.7.1	paired-t-test	70
2.7.2	Shapiro-Wilk test	71
2.7.3	Wilcoxon signed-rank test	72
3	Focused Crawler - Εστιασμένη Διαδικτυακή Ανίχνευση	73
3.1	Εισαγωγή	73
3.2	Σχετική Βιβλιογραφία και Προηγούμενες Μελέτες	75
3.3	Καθορισμός του προβλήματος	77
4	Προτεινόμενες μέθοδοι	81
4.1	Focused Crawler Δύο Σταδίων	81
4.1.1	Πρώτο Στάδιο Focused Crawler: Ενεργός Μάθηση	82
4.1.1.1	Random Sampling	84
4.1.1.2	Uncertainty Sampling με Margin Sampling	85
4.1.1.3	Uncertainty Sampling με Softmax	86
4.1.2	Δεύτερο Στάδιο Focused Crawler: Ενισχυτική Μάθηση	88
4.1.2.1	Ο Focused Crawler ως Markov Decision Process (MDP)	90
4.1.2.2	Επιβράβευση - The Reward function	93
4.1.2.3	Κοινές Αναπαραστάσεις state και action	93
4.1.2.4	Focused Crawler Πράκτορες: DQN και DDQN	96

4.2	Ντετερμινιστικός Focused Crawler: Greedy Επιλογή	100
5	Πειραματική Ανάλυση και Αποτελέσματα	103
5.1	Γενικό Πρόβλημα Αναζήτησης	103
5.1.1	Προετοιμασία Πειράματος	103
5.1.2	Ενεργός Μάθηση	105
5.1.3	Ενισχυτική Μάθηση	108
5.1.3.1	Σύγκριση μεθοδολογιών	108
5.1.3.2	Συμπεράσματα	109
5.2	Ειδικό Πρόβλημα Αναζήτησης	111
5.2.1	Προετοιμασία Πειράματος	112
5.2.2	Ενεργός Μάθηση	113
5.2.2.1	Ταξινόμηση με 3 κλάσεις	114
5.2.2.2	Ταξινόμηση με 2 κλάσεις	115
5.2.3	Καθορισμός του Relevance Calculator	117
5.2.4	Ενισχυτική Μάθηση	118
5.2.4.1	Focused Crawler με Only NLP Representation - Baseline	118
5.2.4.2	Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο	120
5.2.4.3	Ντετερμινιστικός Greedy Focused Crawler: DetFC	120
5.2.4.4	Focused Crawler με Keyword Representation	121
5.2.4.5	Συμπεράσματα	125
5.2.5	Σύγκριση με ανθρώπινη προσωμείωση	127
6	Επίλογος	131
6.1	Συμπεράσματα και Συνεισφορές	131
6.2	Μελλοντικές Επεκτάσεις	132
	Βιβλιογραφία	140
	Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια	141
	Απόδοση ξενόγλωσσων όρων	143

Κατάλογος Σχημάτων

2.1	Διαδικασία Επιβλεπόμενης Μάθησης	36
2.2	Διαδικασία Ενισχυτικής Μάθησης. Πηγή [2]	38
2.3	Πολυεπίπεδο Perceptron (MLP) με ένα κρυφό επίπεδο	45
2.4	Αναπαράσταση sigmoid με tanh σε κοινό διάγραμμα. Με μπλε αναπαρίσταται η sigmoid, ενώ με πορτοκαλί η tanh.	46
2.5	Ξετύλιγμα RNN	48
2.6	Long short-term memory	49
2.7	Active Learning. Πηγή [3]	52
2.8	Pool-Based Active Learning. Πηγή [3]	53
2.9	Ενισχυτική Μάθηση	55
2.10	Continuous Bag-of-words. Πηγή [4]	66
2.11	skip-gram. Πηγή [4]	67
2.12	Επαναληπτική διαδικασία Inter-annotator agreement. Πηγή [5]	68
2.13	Κλίμακες αξιοπιστίας των τιμών του Cohen's kappa	69
2.14	Κλίμακες αξιοπιστίας των τιμών του Krippendorff's alpha	70
3.1	Αρχιτεκτονική γενικού μοντέλου focused crawler	74
3.2	Πρότυπο Ερευνητή-Μαθητή	79
4.1	Focused Crawler Δύο Σταδίων	82
4.2	Εναλλακτικός Focused Crawler Δύο Σταδίων	83
4.3	Random/Margin SVM	86
4.4	Μοντέλα Ενεργού Μάθησης με χρήση Uncertainty Sampling με Softmax. Πάνω αριστερά: LSTM, Πάνω δεξιά: AverageLSTM, Κάτω αριστερά: DoubleLSTM, Κάτω δεξιά: MeanMaxMLP	88
5.1	Διαγράμματα Ενεργού Μάθησης πάνω στο Μέσο Accuracy με Τυπική Απόκλιση συν τω χρόνω.	106
5.2	Διαγράμματα των συσσωρευτικών Harvest Rates του Γενικού Προβλήματος. Πάνω Αριστερά: Focused Crawler με General Representation και 1 seed Πάνω Δεξιά: Focused Crawler με General Representation και 10 seeds Κάτω Αριστερά: Focused Crawler με General Representation και 25 seeds Κάτω Δεξιά: Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο και 25 seeds	109

5.3	Διαγράμματα του μέσου reward με κυλιόμενο παράθυρο ίσο με 40 χρονικά βήματα συναρτήσεως των χρονικών βημάτων. Αριστερά: Focused Crawler με General Representation και 25 seeds Μέση: Focused Crawler με General Representation και 10 seeds Δεξιά: Focused Crawler με General Representation και 1 seed Οι κόκκινες βούλες αναπαριστούν τα χρονικά βήματα στα οποία ενημερώνεται το Target Q-Network	110
5.4	Διαγράμματα AverageLSTM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση για την σχετική κλάση, συν τω χρόνω.	115
5.5	Διαγράμματα DoubleLSTM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση για την σχετική κλάση, συν τω χρόνω.	115
5.6	Διαγράμματα Margin SVM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση για την σχετική κλάση	116
5.7	Διαγράμματα AverageLSTM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση, συν τω χρόνω, για την σχετική κλάση.	116
5.8	Διαγράμματα DoubleLSTM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση, συν τω χρόνω, για την σχετική κλάση.	116
5.9	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Only NLP Representation	119
5.10	Το διάγραμμα του Focused Crawler με Only NLP Representation με κυλιόμενο παράθυρο ίσο με 40 χρονικά βήματα. Οι κόκκινες βούλες συμβολίζουν τα χρονικά βήματα στα οποία έγινε ενημέρωση των βαρών του νευρωνικού στόχου (Target Q-Network).	119
5.11	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο	120
5.12	Το διάγραμμα του συσσωρευτικού Harvest Rate του DetFC	121
5.13	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DQN και 25 seeds	122
5.14	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DQN και 10 seeds	122
5.15	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DQN και 1 seed	122
5.16	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DDQN και 25 seeds	123
5.17	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DDQN και 10 seeds	123
5.18	Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DDQN και 1 seed	123

- 5.19 Διαγράμματα του μέσου reward για το DQN με κυλιόμενο παράθυρο ίσο με 100 χρονικά βήματα συναρτήσεως των χρονικών βημάτων. **Αριστερά:** Focused Crawler με Keyword Representation με DQN και **25 seeds** **Μέση:** Focused Crawler με Keyword Representation με DQN και **10 seeds** **Δεξιά:** Focused Crawler με Keyword Representation με DQN και **1 seed** Οι κόκκινες βούλες αναπαριστούν τα χρονικά βήματα στα οποία ενημερώνεται το Target Q-Network **124**
- 5.20 Διαγράμματα του μέσου reward για το DDQN με κυλιόμενο παράθυρο ίσο με 100 χρονικά βήματα συναρτήσεως των χρονικών βημάτων. **Αριστερά:** Focused Crawler με Keyword Representation με DDQN και **25 seeds** **Μέση:** Focused Crawler με Keyword Representation με DDQN και **10 seeds** **Δεξιά:** Focused Crawler με Keyword Representation με DDQN και **1 seed** Οι κόκκινες βούλες αναπαριστούν τα χρονικά βήματα στα οποία ενημερώνεται το Target Q-Network **124**
- 5.21 Συγκεντρωτικά τα διαγράμματα των συσσωρευτικών Harvest Rates με Keyword Representation (DQN-DDQN). Αριστερά απεικονίζουμε για το DQN και δεξιά για το DDQN. Τα δύο πρώτα διαγράμματα (της πρώτης γραμμής) αφορούν για 25 seeds, τα επόμενα δύο για 10 seeds και τα τελευταία δύο για 1 seed **128**
- 5.22 Συγκεντρωτικά τα διαγράμματα του μέσου reward με κυλιόμενο παραθύρο ίσο με 100 χρονικά βήματα, με χρήση Keyword Representation (DQN-DDQN). Αριστερά απεικονίζουμε για το DQN και δεξιά για το DDQN. Τα δύο πρώτα διαγράμματα (της πρώτης γραμμής) αφορούν για 25 seeds, τα επόμενα δύο για 10 seeds και τα τελευταία δύο για 1 seed **129**

Κατάλογος Εικόνων

2.1	Αναπαράσταση των slack variables. Πηγή [6]	41
2.2	Διαφορές ανθρώπινου και τεχνητού νευρώνα	44
2.3	Ο Αλγόριθμος Q-learning. Πηγή [7]	60
2.4	Ο Αλγόριθμος Double Q-learning. Πηγή [7]	60
2.5	Ο Αλγόριθμος Deep Q-learning. Πηγή [8]	61
2.6	Το σχήμα του Inter-annotator agreement. Πηγή [5]	68

Κατάλογος Πινάκων

5.1	Μετρικές Interannotator Agreement Γενικού Προβλήματος	104
5.2	Confusion Matrix Επισημειώσεων Γενικού Προβλήματος	104
5.3	Μέσες τιμές Accuracy, Τυπικές Αποκλείσεις για τον Γύρο 13 με ακρίβεια τριών μηδενικών και ομαδοποίηση βάσει στατιστικής σημαντικότητας. Η ομάδα A περιέχει τα καλύτερα μοντέλα, ενώ η ομάδα D τα χειρότερα.	107
5.4	Αποτελέσματα των focused crawlers για τον Γενικό Πρόβλημα αναζήτησης . .	108
5.5	Υπερπαράμετροι Γενικού Προβλήματος Δεύτερου Σταδίου	110
5.6	Μετρικές Interannotator Agreement Ειδικού Προβλήματος	112
5.7	Confusion Matrix Επισημειώσεων Ειδικού Προβλήματος	112
5.8	Confusion Matrix πάνω στο pool για 3 κλάσεις	115
5.9	Αποτελέσματα Ενεργού Μάθησης στο Ειδικό Πρόβλημα με 2 κλάσεις (ακρίβεια τριών μηδενικών)	116
5.10	Σύγκριση Καλύτερου Μοντέλου Ενεργού Μάθησης με τον ERC πάνω σε όλο το σύνολο δεδομένων	118
5.11	Υπερπαράμετροι της Keyword Representation για το Δεύτερο Στάδιο του Ειδικού Προβλήματος	125
5.12	Αποτελέσματα των focused crawlers για το Ειδικό Πρόβλημα αναζήτησης Σημειώνουμε πως το harvest rate (75.68%) με Keyword Representation για τα 400 χρονικά βήματα αποτελεί την μέση τιμή των τιμών που προέκυψαν για 25, 10 και 1 seed(s).	126

Κατάλογος Αλγορίθμων

3.1 Γενικός Αλγόριθμος Focused Crawler	75
4.1 Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο	85
4.2 Δεύτερο Στάδιο Focused Crawler: Ενισχυτική Μάθηση	99
4.3 Άπληστος Ντετερμινιστικός Αλγόριθμος Focused Crawler: DetFC	101

Κεφάλαιο **1**

Εισαγωγή

Η Εθνοφαρμακολογία αποτελεί τη μελέτη που αποσκοπεί στην παρατήρηση και τον προσδιορισμό των φαρμακευτικών ιδιοτήτων, κυρίως των φυτών, μιας περιοχής. Στόχο της αποτελεί η καταγραφή και η σύγκριση της χρήσης των θεραπευτικών φυτών ως μία εναλλακτική, συνήθως, μέθοδο αντιμετώπισης συγκεκριμένων ασθενειών. Η χρήση τέτοιων φυτών ταυτίζεται συχνά με την παράδοση του κάθε τόπου και αποτελεί σημαντικό στοιχείο της κουλτούρας του, αφού με ιδιαίτερα αξιοσημειώτο τρόπο κάθε περιοχή συχνά εμφανίζει πρωτότυπες συμπεριφορές ως προς τη χρήση συγκεκριμένων φυτών [9].

Λόγω της σημαντικότητας των ιδιοτήτων ορισμένων φυτών και της σχέσης τους με την τοπική παράδοση, συχνά επιστήμονες εθνοφαρμακολόγοι επισκέπτονται πληθυσμούς, με σκοπό να καταγράψουν την εθνοφαρμακολογική και πολιτιστική τους σημασία. Τα βασικότερα βήματα που περιλαμβάνει μια κλασική μέθοδος εθνοφαρμακολογικής έρευνας [10] είναι τα ακόλουθα. Αρχικά, απαιτείται ο προσδιορισμός του τόπου μελέτης και των κοινωνικών ομάδων (πχ. ντόπιοι, Μικρασιάτες, Αρβανίτες κá) που θα συμμετάσχουν στη μελέτη. Στη συνέχεια, ακολουθεί η πειραματική προσέγγιση της μελέτης. Οι επιστήμονες συγκεκριμένων άτομα και των δύο φύλων, από όλες τις κοινωνικές ομάδες, που θα συμμετέχουν σε συνεντεύξεις, προκειμένου να δώσουν απαντήσεις σε ειδικά ερωτηματολόγια. Συνήθως, συλλέγεται μικρός αριθμός συμμετέχοντων, πχ 100 άτομα. Έπειτα, ακολουθεί εξέταση και διερεύνηση των φαρμακευτικών ιδιοτήτων των φυτών, μέσα από τις μαρτυρίες των συμμετέχοντων και από διάφορες λογοτεχνικές ή βιβλιογραφικές αναφορές. Όταν έχει ολοκληρωθεί το πειραματικό στάδιο και υπάρχει σημαντική συλλογή πληροφοριών, ξεκινά η ανάλυση της έρευνας μέσω της χρήσης του Informant Consensus Factor (FIC) [11] και του υπολογισμού του Use Value (UV) [12] για εξαγωγή αντίστοιχων συμπερασμάτων.

Ωστόσο, η μελέτη της εθνοφαρμακολογικής γνώσης, πέρα από την εξέταση της παραδοσιακής ιστορίας ενός τόπου, μπορεί να οδηγήσει στην εμφάνιση νέων φυσικών φαρμακευτικών γνώσεων και προϊόντων, που συχνά προτιμώνται σε σχέση με τα χημικά για την καταπολέμηση συγκεκριμένων ασθενειών ή παθημάτων [9]. Επιπλέον, υπάρχουν ασθένειες που καθιστούν επιτακτική την ανάγκη για χρήση φυσικών φαρμάκων για την αντιμετώπισή τους, δεδομένου πως τα υπάρχοντα χημικά φάρμακα είτε παράγουν επικίνδυνες παρενέργειες για την υγεία είτε δεν είναι αρκετά αποτελεσματικά [9]. Ακόμα, ιδιαίτερα σημαντική καθίσταται η ορθή χρήση της Εθνοφαρμακολογίας έναντι των χημικών φαρμάκων για ασθένειες ή παθήσεις που δεν απειλούν τη ζωή, πχ. δερματικά νοσήματα, συνταγογράφηση φαρμάκων για εγκύους κá. Έτσι, είναι δεδομένο πως περαιτέρω έρευνα πάνω στην εθνοφαρμακολογία

μπορεί να οδηγήσει σε νέες καινοτόμες θεραπείες που χρησιμοποιούν συγκεκριμένα φυτά για την αντιμετώπιση ασθενειών, τις οποίες σήμερα δεν γνωρίζουμε καν ότι μπορεί να θεραπεύσουν.

Παράλληλα, είναι γεγονός πως η καταγραφή της γνώσης γίνεται συχνά στο Διαδίκτυο. Έτσι, μεγάλο κομμάτι της έκφρασης απόψεων σχετικές με τα φυτά βρίσκεται διασκορπισμένη στο Διαδίκτυο και τα μέσα κοινωνικής δικτύωσης.

1.1 Κίνητρα Μελέτης

Για όλους τους παραπάνω λόγους καθίσταται αναγκαία, από τους επιστήμονες εθνοφαρμακολογίας, η συλλογή βιβλιογραφικών αναφορών από το Διαδίκτυο σχετικές με φυτά που έχουν παρατηρηθεί για την ευεργετική και ιατρική τους δράση. Ακόμη, ιδιαίτερο ενδιαφέρον έχει και η αναζήτηση για φυτά που μέχρι στιγμής δεν συνδέονται εμφανώς με καμία εθνοφαρμακολογική παράδοση, προκειμένου να διαπιστωθεί περαιτέρω θεραπευτική δράση για την καταπολέμηση ασθενειών και παθήσεων.

Πέραν τούτου, συζητώντας με ειδικούς εθνοφαρμακολογίας, βεβαιωθήκαμε πως αυτοί συχνά θα ήθελαν να περιορίσουν την αναζήτηση τέτοιων εθνοφαρμακολογικών αναφορών σε συγκεκριμένα γεωγραφικά και πολιτισμικά πλαίσια, ώστε να έχουν μια πλήρη εικόνα για την εθνοφαρμακολογική παράδοση των περιοχών αυτών. Επιπλέον, ιδιαίτερα σημαντικό για την αναζήτηση τους αποτελεί η εύρεση βιβλιογραφικών αναφορών που σχετίζονται με συγκεκριμένα φυτά, ή/και οικογένειες φυτών, που θεωρούν σημαντικά για την μελέτη τους. Ακόμα, ο ερευνητής, που θέλει να διεξάγει την έρευνα, μέχρι σήμερα αναζητά δημοσιεύσεις σε βιβλιογραφικές μηχανές αναζήτησης, όπως το PubMed [13], το Google Scholar [14] κ.ά., δίνοντας απλά λέξεις-κλειδιά (keywords). Αυτό έχει ως αποτέλεσμα με αυτήν την αναζήτηση να του εμφανίζονται μόνο δημοσιεύσεις που περιέχουν αυτές τις λέξεις-κλειδιά ή συνδυασμό τους, χωρίς ενδεχομένως να λαμβάνονται υπόψιν επιμέρους προτεραιότητες αυτών των λέξεων κλειδιών. Τέλος, υπάρχουν πολλές περιπτώσεις στις οποίες οι λέξεις-κλειδιά είτε δεν είναι διαθέσιμες στις δημοσιεύσεις είτε δεν υπάρχουν στο αντίστοιχο διαθέσιμο κείμενο αυτών (όπως σε τίτλους), με αποτέλεσμα να μην επιστρέφονται αρκετές αναφορές που θα ενδιέφεραν τον χρήστη.

Από την άλλη πλευρά, η τεχνολογική άνθηση της Τεχνητής Νοημοσύνης, και συγκεκριμένα της Μηχανικής Μάθησης (Machine Learning) και της Βαθιάς Μάθησης (Deep Learning), έδωσε τη δυνατότητα για ανάπτυξη των αλληλεπιδράσεων μεταξύ των υπολογιστών και της ανθρώπινης γλώσσας, με άμεση συνέπεια την πρόοδο στο πεδίο της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing). Ένας σκοπός της Επεξεργασίας Φυσικής Γλώσσας είναι η ανάδειξη ειδικών αλγορίθμων για σημασιολογική αναπαράσταση της ανθρώπινης γλώσσας και η εφαρμογή της σε συστήματα εξόρυξης γνώσης και ανάκτησης πληροφορίας.

Παράλληλα, η πρόοδος της Βαθιάς Μάθησης έχει στρέψει τα τελευταία χρόνια το ερευνητικό ενδιαφέρον πολλών επιστημόνων για ανάπτυξη ευφυών συστημάτων που καταφέρνουν και επιδεικνύουν πολύ υποσχόμενες επιδόσεις σε προβλήματα αποφάσεων, εξερεύνησης λύσεων και αλληλεπιδράσεων με τον άνθρωπο μέσω Ενισχυτικής Μάθησης. Με αυτόν τον τρόπο, είμαστε σε θέση να δημιουργούμε ευφυή συστήματα που είναι αρκετά ικανά να μα-

θαίνουν να αναζητούν χρήσιμη σημασιολογική πληροφορία στο Διαδίκτυο. Ταυτόχρονα τα συστήματα αυτά μπορούν να αντιλαμβάνονται σε βάθος τις αλληλεπιδράσεις με τον χρήστη και να εξάγουν επιθυμητά αποτελέσματα, χωρίς να έχουν προγραμματιστεί ρητά εξαρτήσεις για αυτόν τον σκοπό.

1.2 Αντικείμενο της διπλωματικής

Είναι σαφές ότι μια τέτοια απόπειρα από τους επιστήμονες για τη συλλογή μεγάλου αριθμού βιβλιογραφικών αναφορών καθίσταται ιδιαίτερα δύσκολη και χρονοβόρα, δεδομένου και του χρόνου που απαιτείται για την ανάγνωση και κατανόηση των αντίστοιχων κειμένων. Για αυτό το λόγο, είναι αναγκαίο να επιδιώξουμε άλλες πρακτικές, μέσω της δημιουργίας κατάλληλου συστήματος, προκειμένου να υποστηριχθεί με αυτοματισμό η διαδικασία αναζήτησης των βιβλιογραφικών αναφορών. Κύριος στόχος μιας τέτοιας υλοποίησης είναι να ελαχιστοποιηθεί ο συνολικός της χρόνος αναζήτησης αλλά και να εξάγονται άρτια αποτελέσματα μέσα στο χρόνο αυτό. Με άλλα λόγια, καλούμαστε το σύστημα μας να εκτελεί μια διαδικτυακή αναζήτηση αναφορών που στηρίζεται στις απαιτήσεις της εκάστοτε εθνοφαρμακολογικής έρευνας.

Ένα σύστημα που δέχεται ως είσοδο ένα επιλεγμένο θέμα μελέτης από το χρήστη και προσπαθεί να βρει όσο το δυνατόν πιο σχετικές, με αυτό το θέμα, ιστοσελίδες στο Διαδίκτυο, συχνά φέρει στη βιβλιογραφία την ονομασία του προγράμματος εστιασμένης διαδικτυακής ανίχνευσης (*focused crawler*)[15], όπως θα δούμε και στο Κεφάλαιο 3.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία ενός συστήματος *focused crawler* με Μηχανική Μάθηση, που θα αναζητά βιβλιογραφικές αναφορές στο Διαδίκτυο, σχετικές με την εθνοφαρμακολογία. Λόγω των αναγκών για ιδιαίτερα στοχευμένη αναζήτηση αναφορών που σχετίζονται με συγκεκριμένα γεωγραφικά πλαίσια, αλλά και οικογένειες φυτών θα σχεδιάσουμε ευφυή ***focused crawler*** συστήματα που θα αντιμετωπίζουν γενικά, καθώς και πιο ειδικά προβλήματα αναζήτησης πάνω στον ερευνητικό τομέα της ***εθνοφαρμακολογίας***. Θα αναπτύξουμε το κάθε *focused crawler* σύστημα κάτω από την επίβλεψη ειδικών εθνοφαρμακολογίας, που θα λειτουργήσουν ως εμπειρογνώμονες πεδίου (*domain experts*). Αυτοί θα παρέχουν κατά την εκπαίδευση άμεση ανάδραση (*feedback*), όταν το σύστημα το θεωρεί αναγκαίο. Έτσι, αυτό θα είναι σε θέση να προσαρμόζεται στις προτιμήσεις τους και να κατανοεί το πραγματικό επίπεδο συνάφειας (*relevance*) μίας υποψήφιας πηγής, σχετικά με το επιθυμητό θέμα αναζήτησης. Επιπλέον, καλούμαστε το σύστημα σε κάθε πρόβλημα αναζήτησης, πέραν από το να μας επιστρέφει όσο το δυνατόν πιο σχετικές ιστοσελίδες με το θέμα μας, να χρησιμοποιεί το ανθρώπινο *feedback* όσο το δυνατόν λιγότερο αλλά παράλληλα με βέλτιστο τρόπο ώστε να εκπαιδεύεται αποτελεσματικά, όπως θα δούμε και στη συνέχεια.

1.3 Συνεισφορές

Σε αυτή την εργασία, μοντελοποιούμε έναν καινοτόμο *focused crawler* δύο σταδίων μάθησης που στοχεύει στην υποστήριξη της ερευνητικής αναζήτησης για εθνοφαρμακολογικές αναφορές στο Διαδίκτυο. Ο *focused crawler* μας χρησιμοποιεί Ενεργό Μάθηση (Active

Learning) στο πρώτο στάδιο και Βαθιά Ενισχυτική Μάθηση (Deep Reinforcement Learning) στο δεύτερο. Σε αντίθεση με προηγούμενες μελέτες της βιβλιογραφίας, εμείς προτείνουμε το συνδυασμό της Ενεργού με Ενισχυτική Μάθηση, ώστε το σύστημα να κατανοήσει αρχικά το θέμα αναζήτησης, με τη βοήθεια της ανάδρασης, και σε δεύτερο χρόνο να ψάξει με αποδοτικό τρόπο για σχετικά κείμενα. Κατά αυτόν τον τρόπο, παραθέτουμε τις εξής συνεισφορές μας:

- Προτείνουμε έναν αποτελεσματικό και αποδοτικό *focused crawler* που αναζητά έναν πολύ μεγάλο αριθμό βιβλιογραφικών αναφορών σχετικών με την Εθνοφαρμακολογία. Αναφέρουμε πως ένα τέτοιο εργαλείο δεν υπάρχει μέχρι σήμερα, με αποτέλεσμα η εστιασμένη έρευνα στον τομέα της Εθνοφαρμακολογίας να είναι ιδιαίτερα δύσκολη.
- Εισάγουμε το Πρότυπο Ερευνητή-Μαθητή στο *focused crawling*, και γενικότερα στην Μηχανική Μάθηση, με το οποίο μοντελοποιούμε την αρχιτεκτονική του ευφύους μας συστήματος.
- Χρησιμοποιούμε έναν καινοτόμο συνδυασμό Ενεργού και Ενισχυτικής Μάθησης πάνω σε *focused crawler* που διαθέτει τα αντίστοιχα δύο στάδια μάθησης.
- Δημιουργούμε ένα σύνολο δεδομένων μεγέθους 1012 URLs με αντίστοιχες επισημειώσεις, οι οποίες σχετίζονται με ένα ειδικό θέμα αναζήτησης πάνω στην Εθνοφαρμακολογία. Το σύνολο δεδομένων αυτών περιέχει 3 διαφορετικές κλάσεις, που αναφέρονται στη συνάφεια των δειγμάτων σε σχέση με επιλεγμένα θέματα αναζήτησης που ορίζουμε στη συνέχεια.
- Δημιουργούμε ένα δεύτερο σύνολο δεδομένων μεγέθους 813 URLs με δύο κλάσεις. Η μία κλάση αφορά URLs αναφορών σχετικών με την Εθνοφαρμακολογία και η δεύτερη κλάση αφορά αναφορές οι οποίες δεν σχετίζονται καθόλου με την Εθνοφαρμακολογία.
- Δείχνουμε άλλη μία περίπτωση στην οποία η Βαθιά Μάθηση δίνει αρκετά ενθαρρυντικά αποτελέσματα σε σχέση με μοντέλα κλασικής Μηχανικής Μάθησης, που λάβαμε ως βασικής γραμμής (*baseline*), κατά τη διαδικασία της Ενεργού Μάθησης.
- Μοντελοποιούμε το περιβάλλον της Ενισχυτικής Μάθησης ως ένα μεγάλο Μάρκον Decision Process (*large MDP*). Σε αυτό, καταφέρνουμε να μοντελοποιήσουμε το *state* σαν έναν υπογράφο του Παγκοσμίου Ιστού θεωρώντας ως κόμβους του τις ιστοσελίδες και ως ακμές του τα εκάστοτε URLs που εξάγονται από αυτές. Με αυτόν τον τρόπο, το *state* περιέχει όλη τη διαδρομή που έχει διαγράψει ο *focused crawler* από το ξεκίνημά του. Έτσι, με ένα θεμελειώδη τρόπο ενθαρρύνουμε το *focused crawling* με χρήση Βαθιάς Ενισχυτικής Μάθησης.
- Χρησιμοποιούμε την κοινή αναπαράσταση (*state, action*) που έχει ξαναχρησιμοποιηθεί σε περιπτώσεις *focused crawling* με Ενισχυτική Μάθηση. Από τη μεριά μας, προτείνουμε αναπαραστάσεις που συνδυάζουν *word embeddings*, στατιστικά χαρακτηριστικά, λέξεις-κλειδιά ή/και συνεχή χαρακτηριστικά που εξάγονται από νευρωνικά δίκτυα.

1.4 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε έξι κεφάλαια: Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο των βασικών εννοιών που σχετίζονται με τη διπλωματική εργασία αυτή. Αρχικά, δίνουμε μια μικρή εισαγωγή στις βασικές έννοιες και κατηγορίες της Μηχανικής Μάθησης. Προχωράμε με μοντέλα Επιβλεπόμενης Μάθησης που θα συναντήσουμε στη μελέτη μας. Στη συνέχεια, αναφέρουμε ορισμένες θεωρητικές έννοιες της Ενεργού και της Ενισχυτικής Μάθησης. Τέλος, περιγράφουμε σημαντικές τεχνικές της Επεξεργασίας Φυσικής Γλώσσας για την διαχείριση κειμένων, μαζί με βασικές έννοιες περί συμφωνίας επισημειωτών και στατιστικού ελέγχου. Στο Κεφάλαιο 3 περιγράφουμε ένα γενικό μοντέλο Focused Crawler, μαζί με σχετικές βιβλιογραφικές μελέτες για το θέμα αυτό, και στη συνέχεια δίνεται ένας τυπικός ορισμός του προβλήματος της εργασίας αυτής. Στο Κεφάλαιο 4 παρουσιάζονται οι προτεινόμενες μας μέθοδοι για τη δημιουργία ενός Focused Crawler δύο σταδίων. Στο Κεφάλαιο 5 αναλύουμε με λεπτομέρεια την πειραματική διαδικασία της εργασίας και παρουσιάζουμε τα αντίστοιχα αποτελέσματα που εξάγαμε. Τέλος, στο Κεφάλαιο 6 δίνεται μια συγκεντρωτική παρουσίαση των γενικών συμπερασμάτων μας καθώς και μερικές σκέψεις μας για μελλοντικές επεκτάσεις της μελέτης μας.

Κεφάλαιο 2

Βασικές Έννοιες

Στο κεφάλαιο αυτό παρουσιάζουμε τις απαραίτητες βασικές γνώσεις και το θεωρητικό υπόβαθρο πάνω στο οποίο θα στηρίζουμε την μελέτη μας. Θα μιλήσουμε, αρχικά, στην Ενότητα 2.1 για βασικές έννοιες της Μηχανικής Μάθησης, όπως η Επιβλεπόμενη και η Μη Επιβλεπόμενη Μάθηση. Στην Ενότητα 2.2 θα αναφερθούμε σε συγκεκριμένα μοντέλα Επιβλεπόμενης Μάθησης που θα χρησιμοποιήσουμε στο πειραματικό κομμάτι της μελέτης μας. Στην Ενότητα 2.3, θα συζητήσουμε για την Ενεργό Μάθηση (Active Learning), ενώ στην Ενότητα 2.4 για την Ενισχυτική Μάθηση. Προχωρώντας, στην Ενότητα 2.5 θα περιγράψουμε θεωρητικές έννοιες της Διαχείρισης Κειμένων και Επεξεργασίας Φυσικής Γλώσσας. Τέλος, θα αναφερθούμε σε ορισμένες θεωρητικές έννοιες που αφορούν τη στατιστική στήριξη των αποτελεσμάτων. Έτσι, στις Ενότητες 2.6 και 2.7, θα μιλήσουμε για περιπτώσεις που χρειαζόμαστε τη Συμφωνία Επισημειωτών (Inter-Annotator Agreement) και για το Στατιστικό Ελέγχο των αποτελεσμάτων της Μηχανικής Μάθησης, αντίστοιχα.

Σημειώνουμε πως καθόλη την διάρκεια της εργασίας θα γράφουμε τα διανύσματα με έντονο μαύρο χρώμα (bold), ώστε να τα διαχωρίζουμε από τα βαθμωτά μεγέθη.

2.1 Εισαγωγή στη Μηχανική Μάθηση

Η Μηχανική Μάθηση (**Machine Learning**) είναι ένα υποπεδίο της Επιστήμης των Υπολογιστών (Computer Science) και θεωρείται υποσύνολο της Τεχνητής Νοημοσύνης (Artificial Intelligence). Ο πιο συνηθισμένος τρόπος να την εκφράσουμε είναι ο εξής: “Η ανάπτυξη αλγορίθμων ώστε η μύχανη να βελτιώνεται αυτόματα μέσα από εμπειρία, χωρίς ωστόσο αυτή να έχει προγραμματιστεί ρητά για αυτό.” Ωστόσο, ένας επίσημος ορισμός δόθηκε από τον T.Mitchell:

“Ένα πρόγραμμα υπολογιστή θεωρούμε ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P , αν η επίδοσή του σε εργασίες της κλάσης T , όπως αποτιμάται από το μέτρο P , βελτιώνεται με την εμπειρία E ” [16].

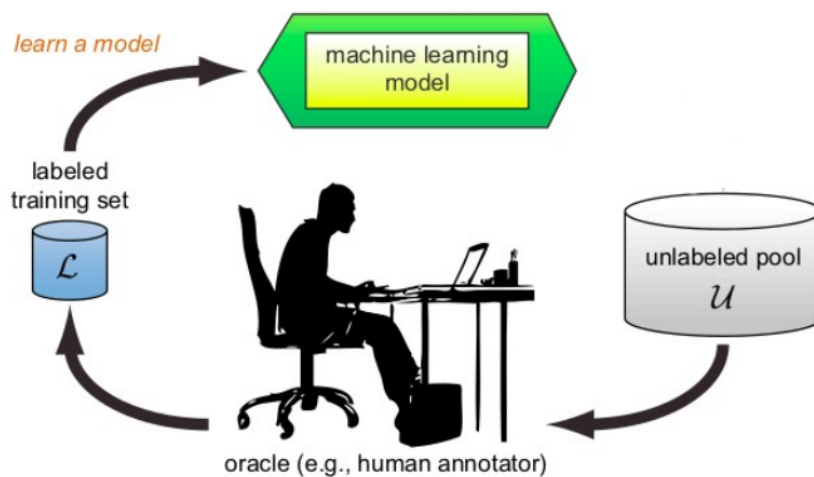
Οι αλγόριθμοι της Μηχανικής Μάθησης τις περισσότερες φορές απαιτούν την υπάρξη ενός συνόλου δεδομένων, το οποίο περιέχει παρατηρήσεις για ένα συγκεκριμένο πρόβλημα, και προσπαθούν να προβλέψουν ή να πάρουν αποφάσεις με αυτοματοποιημένο τρόπο. Ορισμένες κλασικές εφαρμογές της αφορούν την Όραση Υπολογιστών, την ταξινόμηση κειμένων και την Επεξεργασία της Ανθρώπινης Γλώσσας, στις οποίες είναι εξαιρετικά δύσκολο ή/και ακατόρθωτο να δημιουργηθούν συμβατικοί αλγόριθμοι καθολικής χρησιμότητας.

Οι κύριες κατηγορίες της Μηχανικής Μάθησης είναι η Επιβλεπόμενη, η Μη Επιβλεπόμενη Μάθηση και η Ενισχυτική Μάθηση. Για αυτές θα μιλήσουμε στις υποενότητες που ακολουθούν.

2.1.1 Επιβλεπόμενη Μάθηση

Η Επιβλεπόμενη Μάθηση (**Supervised Learning**) είναι η διαδικασία μάθησης μιας συνάρτησης μέσω της οποίας μπορεί να γίνει αντιστοίχιση μιας εισόδου σε μια καθορισμένη έξοδο δεδομένων δειγμάτων ζευγαριών της μορφής είσοδος-έξοδος. Η διαδικασία της Επιβλεπόμενης Μάθησης προϋποθέτει την ύπαρξη ενός συνόλου δεδομένων, το οποίο περιλαμβάνει επισημειωμένες παρατηρήσεις (labeled data). Το σύνολο δεδομένων αυτών αποτελεί το σύνολο εκπαίδευσης (training set) της μάθησης. Ένας αλγόριθμος Επιβλεπόμενης Μάθησης αναλύει τα δεδομένα του συνόλου εκπαίδευσης και συμπεραίνει μια συνάρτηση που μπορεί να χρησιμοποιηθεί για να αντιστοιχήσει νέες παρατηρήσεις στις σωστές τους επισημειώσεις. Βέλτιστα, ένας αλγόριθμος Επιβλεπόμενης Μάθησης θα θέλαμε να προβλέπει σωστά τις επισημειώσεις (labels) ακόμη και για παρατηρήσεις που δεν έχει δει προηγουμένως.

Στο σχήμα 2.1 αναπαριστούμε μια τυπική διαδικασία Επιβλεπόμενης Μάθησης. Θεωρούμε ότι ένας Oracle, πχ ένας άνθρωπος, επισημειώνει ένα σύνολο παρατηρήσεων U , ώστε να προκύψει το επισημειωμένο σύνολο δεδομένων L . Στην συνέχεια, πάνω στο L εκπαιδεύεται κάποιο μοντέλο y με Επιβλεπόμενη Μάθηση. Τυπικά, μία είσοδος X , θα έχει ως έξοδο του μοντέλου το $Y = y(X)$.



Σχήμα 2.1: Διαδικασία Επιβλεπόμενης Μάθησης

Οι δύο υποκατηγορίες της Επιβλεπόμενης Μάθησης είναι η ταξινόμηση (**classification**) και η παλινδρόμηση (**regression**). Στην ταξινόμηση, το πρόβλημα περιέχει ως επισημειώσεις *κλάσεις*, δηλαδή κατηγορίες της ταξινόμησης. Έτσι, το μοντέλο εκπαιδεύεται στο να προβλέπει τη σωστή κλάση της εισόδου. Στην παλινδρόμηση, το πρόβλημα περιέχει ως επισημειώσεις *συνεχείς τιμές* και το μοντέλο εκπαιδεύεται ώστε να προβλέπει την τιμή που αντιστοιχεί στην είσοδο.

2.1.2 Μη Επιβλεπόμενη Μάθηση

Η Μη Επιβλεπόμενη Μάθηση (**Unsupervised Learning**) είναι η μάθηση στην οποία τα δεδομένα δεν είναι επισημειωμένα (unlabeled data). Ειδικότερα, προσπαθεί να εντοπίσει μοτίβα στο σύνολο των παρατηρήσεων που δεν έχουν εντοπιστεί. Σε αντίθεση με την Επιβλεπόμενη Μάθηση που χρησιμοποιεί κατά πολύ τον ανθρώπινο παράγοντα για τις καταγραφές των επισημειώσεων, η Μη Επιβλεπόμενη συνήθως τον χρειάζεται πιο έμμεσα: για την εύρεση του συνόλου παρατηρήσεων και για την αξιολόγηση της χρησιμότητας των παραγόμενων αποτελεσμάτων.

Δεδομένου πως δεν υπάρχουν επισημειώσεις, η Μη Επιβλεπόμενη Μάθηση σκοπεύει να μάθει την εσωτερική δόμη ή την οργάνωση των παρατηρήσεων στον χώρο των χαρακτηριστικών τους (feature space). Η συνηθέστερη υποκατηγορία της αφορά την συσταδοποίηση (**clustering**) των παρατηρήσεων: αυτές ομαδοποιούνται με βάση τα χαρακτηριστικά τους. Στην συσταδοποίηση, ένα μέτρο ομοιότητας εξετάζεται ώστε να ομαδοποιηθούν οι παρατηρήσεις εισόδου σε συστάδες (clusters). Ωστόσο, πολλές φορές είναι δύσκολος ο προσδιορισμός του ακριβέστερου αριθμού συστάδων που χρειάζεται.

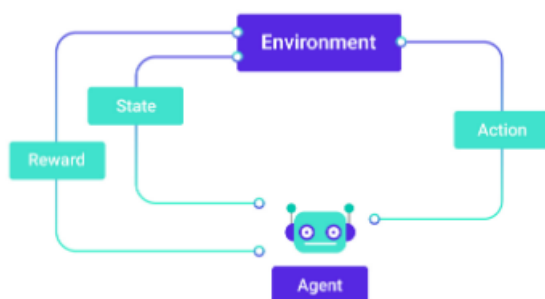
Μια άλλη υποκατηγορία της Μη Επιβλεπόμενης Μάθησης είναι τα Παραγωγικά Μοντέλα (**Generative Models**): λαμβάνουν τις μη επισημειωμένες παρατηρήσεις σαν είσοδο και επιστρέφουν νέες παραγόμενες προερχόμενες από την ίδια κατανομή. Με αυτόν τον τρόπο, τέτοια μοντέλα μαθαίνουν την οργάνωση και την δομή των παρατηρήσεων, με σκοπό να παράξουν νέες πανομοιότυπες.

2.1.3 Ενισχυτική Μάθηση

Στην Ενισχυτική Μάθηση (**Reinforcement Learning** ή **RL**) υπάρχει ένας ευφυής πράκτορας (agent), ο οποίος "ζει" σε ένα περιβάλλον (environment). Το περιβάλλον παρέχει καταστάσεις (states). Ο πράκτορας μπορεί κάθε στιγμή να βρίσκεται σε μία κατάσταση. Όταν ο πράκτορας βρίσκεται σε μία κατάσταση, το περιβάλλον του παρέχει δράσεις (actions), τις οποίες μπορεί να επιλέξει. Όταν ο πράκτορας επιλέγει κάποια δράση, τότε μετακινείται σε κάποια άλλη κατάσταση. Η μετακίνησή του αυτή ανταμοίβεται από το περιβάλλον με μία επιβράβευση (reward). Σκοπός του πράκτορα είναι να μεγιστοποιήσει το μελλοντική συσσωρευτική επιβράβευση. Μια τέτοια συμπεριφορά του περιβάλλοντος συνήθως εκφράζεται τυπικά με το **Markov Decision Process (MDP)**, όπως θα συζητήσουμε στην συνέχεια. Η διαδικασία που περιγράψαμε παραπάνω απεικονίζεται στο Σχήμα 2.2.

Μπορούμε να φανταστούμε ένα πρόβλημα της Ενισχυτικής Μάθησης σαν ένα παιχνίδι, όπου ο πράκτορας είναι ο παίκτης που προσπαθεί να παίξει βέλτιστα το παιχνίδι. Ο πράκτορας - παίκτης - βρίσκεται σε μία κατάσταση, πχ μία πίστα ή ένα στιγμιότυπο του παιχνιδιού, και μπορεί να επιλέξει κάποιες δράσεις, πχ. κινήσεις. Σκοπός του είναι να ανακαλύψει μία βέλτιστη στρατηγική (policy) ώστε με σωστή αλληλουχία κινήσεων να κερδίσει με τον πιο κερδοφόρο τρόπο το παιχνίδι.

Η Ενισχυτική Μάθηση διαφέρει από την Επιβλεπόμενη Μάθηση στο ότι δεν χρειάζονται επισημειωμένα ζευγάρια είσοδος-έξοδος. Επίσης, δεν απαιτείται οι υποβέλτιστες δράσεις να είναι ρητά σωστές. Αντίθετα, ο πράκτορας καλείται να βρει μια ισορροπία μεταξύ της εξερεύνησης και της εκμετάλλευσης της τρέχουσας γνώσης για το πρόβλημα.



Σχήμα 2.2: Διαδικασία Ενισχυτικής Μάθησης. Πηγή [2]

2.2 Μοντέλα Επιβλεπόμενης Μάθησης

Στην ενότητα αυτή, θα συζητήσουμε για τα μοντέλα Επιβλεπόμενης Μάθησης που θα συναντήσουμε στην παρούσα διπλωματική εργασία.

Όπως συζητήσαμε και προηγουμένως, στην Επιβλεπόμενη Μάθηση τυπικά έχουμε να αντιμετωπίσουμε το εξής πρόβλημα: Δεδομένων των παρατηρήσεων \mathbf{x}_i , πλήθους N , με $i \in [1, N]$, του συνόλου εκπαίδευσης και των επισημειώσεών τους t_i , θέλουμε να βρούμε μία συνάρτηση y ώστε ιδανικά για κάθε παρατήρηση \mathbf{x} με πραγματική επισημείωση t , να ικανοποιείται $y(\mathbf{x}) = t$. Από εδώ και στο εξής θα αναφερόμαστε σε μία παρατήρηση με \mathbf{x} , όπου \mathbf{x} θα είναι το διάνυσμα αυτής της παρατήρησης, και με t στην αντίστοιχη επισημείωσή της. Η συνάρτηση y , περιέχει παραμέτρους \mathbf{w} , τις οποίες θέλουμε να βρούμε ώστε λύσουμε το παραπάνω πρόβλημα. Στην Επιβλεπόμενη Μάθηση, οι παράμετροι \mathbf{w} βρίσκονται ελαχιστοποιώντας μία συνάρτηση κόστους $L(\mathbf{w})$ ως προς \mathbf{w} .

2.2.1 Μηχανές Διανυσμάτων Υποστήριξης

Στην υποενότητα αυτή θα μιλήσουμε για έναν πολύ διαδεδομένο κλασικό αλγόριθμο Μηχανικής Μάθησης, τις Μηχανές Διανυσμάτων Υποστήριξης (**Support Vector Machines** ή **SVM** [17]).

Ο SVM προσπαθεί να λύσει το πρόβλημα της ταξινόμησης μέσα από την έννοια του περιθωρίου (margin). Ως margin ορίζεται η μικρότερη απόσταση μεταξύ του περιθωρίου απόφασης (decision boundary) και οποιωνδήποτε σημείων των παρατηρήσεων. Το περιθώριο απόφασης αποτελεί την περιοχή στον χώρο των χαρακτηριστικών στην οποία το μοντέλο είναι διφορούμενο ως προς την πρόβλεψη της εισόδου. Έτσι, ο SVM προσπαθεί να βρει εκείνο το margin για τα δεδομένα εκπαίδευσής του, ώστε αυτό να είναι το μέγιστο δυνατό. Με άλλα λόγια, διαλέγει για περιθώριο απόφασης εκείνο που μεγιστοποιεί το margin.

Θεωρούμε την περίπτωση του προβλήματος της δυαδικής ταξινόμησης, ώστε για την πρώτη κλάση να ισχύει $t=1$ και για την δεύτερη $t=-1$. Έχουμε την περίπτωση γραμμικού μοντέλου της μορφής

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \quad (2.1)$$

όπου $\phi(\mathbf{x})$ είναι μία συνάρτηση μεταφοράς σε νέο χώρο χαρακτηριστικών και το b η παράμε-

τρος για την προκατάληψη του μοντέλου (bias). Η συνάρτηση ϕ μετατρέπει την είσοδο σε ένα διάνυσμα μεγαλύτερης διαστατικότητας. Έτσι, σημεία στο χώρο των χαρακτηριστικών ταξινομούνται με βάση το πρόσημο του y . Εάν είναι θετικό τότε ταξινομούνται στην κλάση που αντιστοιχεί στο $t=1$, αλλιώς στην κλάση για $t=-1$.

Η κάθετη απόσταση r ενός σημείου \mathbf{x} από το υπερεπίπεδο που ορίζει η εξίσωση $y(\mathbf{x}) = 0$ δίνεται από τον τύπο

$$r = \frac{|y(\mathbf{x})|}{\|\mathbf{w}\|} \quad (2.2)$$

Θεωρώντας πως θέλουμε μόνο εκείνες τις λύσεις που ταξινομούν σωστά όλα τα δείγματα εκπαίδευσης \mathbf{x}_n , έχουμε $t_n y(\mathbf{x}_n) > 0$ για όλες τις παρατηρήσεις του συνόλου εκπαίδευσης. Η απόσταση ενός σημείου \mathbf{x}_n από το περιθώριο απόφασης (decision boundary) είναι

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|} \quad (2.3)$$

Έτσι, το margin είναι η κάθετη απόσταση στο κοντινότερο \mathbf{x}_n . Άρα, σκοπός είναι να το μεγιστοποιήσουμε βελτιστοποιώντας τις παραμέτρους \mathbf{w} και b . Το μέγιστο margin δίνεται από την λύση της

$$\operatorname{argmax}_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|} \min_n [t_n (\mathbf{w}^T \phi(\mathbf{x}_n + b))] \quad (2.4)$$

Δεδομένου πως έχουμε θεωρήσει πως όλα τα σημεία εκπαίδευσης θα ταξινομούνται σωστά, δηλαδή θα είναι γραμμικά διαχωρίσιμα στον νέο χώρο χαρακτηριστικών μέσω της συνάρτησης ϕ , όλα τα σημεία θα τηρούν τον περιορισμό

$$t_n (\mathbf{w}^T (\mathbf{x}_n + b)) \geq 1 \quad (2.5)$$

για κάθε $n = 1, 2, \dots, N$, θεωρώντας πως η ισότητα ισχύει για εκείνα τα σημεία που βρίσκονται σε ελάχιστη απόσταση από την περιθώριο απόφασης. Από αυτό λαμβάνουμε ότι η ισότητα αυτή θα ισχύει πάντα για τουλάχιστον ένα σημείο, δηλαδή το πιο κοντινό. Την παραδοχή αυτήν μπορούσαμε και την πήραμε από το γεγονός ότι αν θέσουμε για \mathbf{w} το $\mathbf{k}\mathbf{w}$ και για b το $\mathbf{k}b$, τότε η απόσταση κάθε σημείου από το περιθώριο απόφασης παραμένει σταθερή.

Κατά αυτόν τον τρόπο, τελικά αυτό που θέλουμε, είναι να μεγιστοποιήσουμε το $\frac{1}{\|\mathbf{w}\|}$. Αυτό όμως συνεπάγεται να ελαχιστοποιήσουμε το $\|\mathbf{w}\|^2$. Ισοδύναμα, το πρόβλημά μας είναι να λύσουμε το

$$\operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.6)$$

Ωστόσο, αυτό είναι ένα τετραγωνικό πρόβλημα, δεδομένου πως έχουμε να ελαχιστοποιήσουμε μία τετραγωνική συνάρτηση ως προς γραμμικούς ανισοτικούς περιορισμούς. Έτσι, χρειαζόμαστε τους συντελεστές Lagrange $a_n \geq 0$, για κάθε περιορισμό της παραπάνω ανι-

σότητας. Με αυτόν τον τρόπο, προκύπτει η συνάρτηση

$$L(\mathbf{w}, b, a) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n [t_n \mathbf{w}^T \phi(\mathbf{x}_n + b) - 1] \quad (2.7)$$

Παραγωγίζοντας την αυτήν ως προς \mathbf{w} και ως προς b παίρνουμε αντίστοιχα

$$\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad (2.8)$$

$$0 = \sum_{n=1}^N a_n t_n \quad (2.9)$$

Έτσι, προκύπτει η διυκή (dual) αναπαράσταση και ανάγουμαι το πρόβλημα στο να μεγιστοποιήσουμε την

$$l(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (2.10)$$

όπου η k είναι η συνάρτηση πυρήνα (kernel function) και $\mathbf{a} = \{a_1, \dots, a_n\}$. Η συνάρτηση πυρήνα ορίζεται ως εξής

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \quad (2.11)$$

Με αυτόν τον τρόπο, η συνάρτηση του $y(\mathbf{x})$ μετασχηματίζεται ανάλογα χρησιμοποιώντας την συνάρτηση πυρήνα και τους συντελεστές Lagrange, και δίνεται από την σχέση

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \quad (2.12)$$

Μία συνάρτηση πυρήνα που θα μας απασχολήσει είναι ο πυρήνας RBF, του οποίου η συνάρτηση δίνεται από την σχέση

$$k(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x} - \mathbf{y}\|^2} \quad (2.13)$$

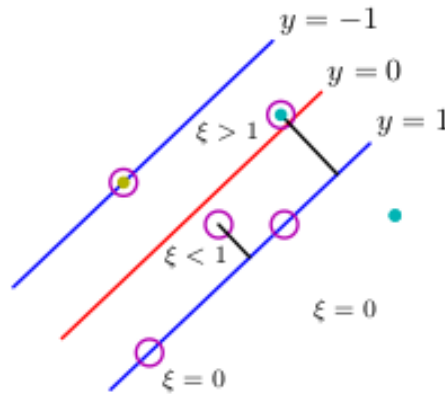
όπου το γ είναι κάποιος θετικός αριθμός.

Επιπλέον, προκύπτει ότι ικανοποιούνται οι συνθήκες Karush-Kuhn-Tucker (KKT), όπου στην περίπτωσή μας ένα από τα πράγματα που απαιτούν είναι τα $a_n \geq 0$. Τα σημεία για τα οποία τα a_n είναι μηδενικά είναι τα διανύσματα υποστήριξης (support vectors). Έτσι, η συνάρτηση του y τελικά καθορίζεται μόνο από τα διανύσματα υποστήριξης. Όσοσο, για την έρευνα των διανυσμάτων υποστήριξης έπαιξαν ρόλο όλα τα σημεία εκπαίδευσης.

Η χρήση του πυρήνα είναι πολύ σημαντική και για την χρονική πολυπλοκότητα του SVM, λόγω του περίφημου τεχνάσματος του πυρήνα (kernel trick). Με αυτό υπολογίζουμε ένα κριτήριο ομοιότητας, μεταξύ του σημείο εισόδου και ενός διανύσματος υποστήριξης, που ισούται με το εσωτερικό γινόμενο των αντίστοιχων $\phi(\mathbf{x})$ τους, χωρίς ωστόσο να υπολογίζουμε

τις ίδιες τις $\phi(\mathbf{x})$. Έτσι, δεν βρίσκουμε τις συντεταγμένες των σημείων αυτών στην διάσταση της συνάρτησης ϕ . Αντίθετα, χρησιμοποιούμε τις συντεταγμένες τους όπως είναι στην διάσταση εισόδου.

Ωστόσο, έχουμε υποθέσει πως το μοντέλο θα ταξινομεί πάντα σωστά τα σημεία εισόδου. Κάτι τέτοιο ωστόσο θα είχε ως αποτέλεσμα την υπερεκπαίδευση (overfitting), δηλαδή την αδυναμία του μοντέλου να γενικεύσει την μάθηση σε νέα σημεία. Για τον λόγο αυτό, επεκτείνουμε την προηγούμενη μοντελοποίηση, προσθέτοντας τις λεγόμενες μεταβλητές χαλάρωσης (slack variables) $\xi_n \geq 0$, έχοντας μια μεταβλητή χαλάρωσης για κάθε σημείο εκπαίδευσης. Οι μεταβλητές αυτές ορίζονται ως εξής: $\xi_n = 0$ εάν τα σημεία είναι πάνω ή εσωτερικά της σωστής περιοχής του margin, $0 < \xi_n \leq 1$ εάν τα σημεία είναι μέσα αλλά στη σωστή περιοχή του margin και $\xi_n > 1$ εάν τα σημεία βρίσκονται στην λάθος μεριά του περιθωρίου απόφασης. Στην Εικόνα 2.1, αναπαριστούμε τις μεταβλητές χαλάρωσης και τη σημασία τους στο margin.



Εικόνα 2.1: Αναπαράσταση των slack variables. Πηγή [6]

Έτσι, πλέον θέλουμε να ελαχιστοποιήσουμε το

$$C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.14)$$

όπου η υπερπαράμετρος C καθορίζει τον βαθμό που θα τιμωρούμε το λάθος ταξινόμησης ενός σημείου εκπαίδευσης. Εύκολα αντιλαμβανόμαστε ότι όσο μεγαλώνει το C τόσο πιο πολύ τιμωρούμε το λάθος με αποτέλεσμα να οδηγούμαστε στην προηγούμενη προσέγγιση και άρα στην υπερεκπαίδευση. Έχοντας πλέον ορίσει το πρόβλημα μπορούμε πάλι να ορίσουμε την συνάρτηση Lagrange L ως εξής

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n [t_n y(\mathbf{x}_n) - 1 + \xi_n] - \sum_{n=1}^N m_n \xi_n \quad (2.15)$$

όπου a_n, m_n είναι συντελεστές Lagrange.

Κάνοντας την ίδια διαδικασία με την προηγούμενη προσέγγιση καταλήγουμε στη διυική (dual) αναπαράσταση. Έτσι, μπορούμε να βρούμε τους συντελεστές Lagrange και το b

τηρώντας και πάλι τις ΚΚΤ συνθήκες που προκύπτουν. Όλες οι ενδιάμεσες πράξεις, αποδείξεις και οι συνθήκες υπάρχουν αναλυτικά στο C.M.Bishop[6]. Και πάλι η συνάρτηση $y(\mathbf{x})$ γράφεται ως εξής

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \quad (2.16)$$

Σημειώνουμε πως ο SVM, προκειμένου να βρει τις βέλτιστες παραμέτρους \mathbf{w} , επιλύει ένα πρόβλημα κυρτής βελτιστοποίησης. Έτσι, κάθε λύση του προβλήματος είναι και ολικό ακρότατο.

Στην περίπτωση ταξινόμησης με $K > 2$ κλάσεις, μπορούμε να διαχειριστούμε το πρόβλημα ως εξής: (α) Τρέχουμε K δυαδικές ταξινομήσεις ένας εναντίον όλων (one-vs-all) SVMs ή (β) τρέχουμε $\frac{K(K-1)}{2}$ δυαδικές ταξινομήσεις ένας εναντίον ενός (one-vs-one) SVMs για όλα τα δυνατά διαφορετικά ζευγάρια. Η τελευταία μέθοδος μπορεί να βελτιστοποιηθεί με την χρήση του DAGSVM, με την οποία για να ταξινομηθεί ένα νέο σημείο λαμβάνονται υπόψιν μόνο $K-1$ ζευγάρια ταξινομητών SVM [6]. Σε όλες τις μεθοδολογίες παίρνουμε την πρόβλεψη εκείνου του SVM που αντιστοιχεί στην μεγαλύτερη απόσταση από το περιθώριο απόφασης.

2.2.2 Αλγόριθμος Perceptron

Ένα γραμμικό μοντέλο, που αποτελεί την βάση των νευρωνικών δικτύων σήμερα, είναι ο αλγόριθμος Perceptron. Τον αλγόριθμο αυτόν τον δημιούργησε ο Rosenblatt το 1957 [18]. Αναφέρεται σε ένα πρόβλημα δυαδικής ταξινόμησης, στο οποίο από ένα διάνυσμα εισόδου \mathbf{x} μέσω ενός γραμμικού μοντέλου παράγεται η έξοδος y σύμφωνα με την σχέση

$$y = f(\mathbf{w}^T \mathbf{x}) \quad (2.17)$$

Όπου η συνάρτηση f λέγεται συνάρτηση ενεργοποίησης (activation function) και είναι η συνάρτηση προσήμου $\text{sgn}(x)$.

Θεωρούμε και πάλι την περίπτωση που η επισημείωση για την πρώτη κλάση αντιστοιχεί σε $t = 1$ και για την δεύτερη σε $t = -1$. Ο αλγόριθμος ταξινομεί την είσοδο στην πρώτη κλάση όταν $y > 0$ ενώ στην δεύτερη όταν $y < 0$. Ο αλγόριθμος Perceptron κάθε φορά "τιμωρεί" τις εισόδους που δεν ταξινομούνται σωστά προσπαθώντας να ελαχιστοποιήσει το $-\mathbf{w}^T \mathbf{x} t$. Έτσι, το κριτήριο του Perceptron ορίζεται ως

$$L(\mathbf{w}) = - \sum_{n=1}^M \mathbf{w}^T \mathbf{x}_n t_n \quad (2.18)$$

όπου το M είναι το πλήθος των εισόδων που δεν ταξινομήθηκαν σωστά.

Έτσι, προκειμένου να ενημερώσουμε τις παραμέτρους \mathbf{w} , εκτελούμε στοχαστικό αλγόριθμο κλίσης μετάβασης (stochastic gradient descent ή SGD) με τον ακόλουθο τρόπο

$$\mathbf{w}^{(s+1)} = \mathbf{w}^{(s)} - a \nabla L(\mathbf{w}) = \mathbf{w}^{(s)} + a \mathbf{x}_n t_n \quad (2.19)$$

όπου a είναι ο ρυθμός μάθησης (learning rate) και s είναι ο αριθμός που δηλώνει το βήμα

που βρίσκεται ο αλγόριθμος. Σημειώνουμε πως ο SGD είναι ένας βελτιστοποιητής (optimizer). Για τους optimizers θα συζητήσουμε σε επόμενη υποενότητα του κεφαλαίου.

Αξίζει να παρατηρήσουμε πως ο αλγόριθμος Perceptron εγγυάται πως θα συγκλίνει ταξινομώντας όλα τα δεδομένα εισόδου σωστά, αρκεί αυτά να είναι γραμμικώς διαχωρίσιμα. Σε αντίθετη περίπτωση, είναι σίγουρο πως δεν θα βρει ποτέ μια τέτοια λύση και άρα δεν θα συγκλίνει. Ωστόσο, όπως είδαμε και στον SVM θα μπορούσαμε να αντικαταστήσουμε την είσοδο \mathbf{x} με το $\phi(\mathbf{x})$ μεταφέροντας την σε έναν επαυξημένο χώρο διαστάσεων. Στον νέο χώρο διαστάσεων τα δεδομένα εισόδου μπορεί πλέον να είναι γραμμικώς διαχωρίσιμα και συνεπώς να συγκλίνει ο αλγόριθμος.

2.2.3 Τεχνητά Νευρωνικά Δίκτυα

Όπως έχουμε ήδη δει, οι Μηχανές Διανύσματος Υποστήριξης (SVM) επιλύουν ένα πρόβλημα βελτιστοποίησης μιας κυρτής συνάρτησης, ώστε να βρουν τα διανύσματα υποστήριξης. Ωστόσο, τα διανύσματα υποστήριξης είναι σχετικά λίγα σε σχέση με το αρχικό σύνολο εκπαίδευσης, αλλά αυξάνουν σημαντικά όσο αυτό γίνεται μεγαλύτερο.

Μια εναλλακτική μέθοδος είναι τα Τεχνητά Νευρωνικά Δίκτυα (**Artificial Neural Networks**), τα οποία χρησιμοποιούν σταθερό αριθμό μη γραμμικών συναρτήσεων, με την ιδιότητα ωστόσο να μεταβάλλουν τις παραμέτρους τους κατά τη διάρκεια της μάθησης. Στον αλγόριθμο του SVM ο αριθμός των μη γραμμικών συναρτήσεων μέσω των διανυσμάτων υποστήριξης δεν αποτελεί υπερπαραμέτρο αλλά μαθαίνεται κατά τη διάρκεια της εκπαίδευσης. Αντίθετα στο νευρωνικό δίκτυο ο αριθμός αυτός είναι σταθερός εξ αρχής.

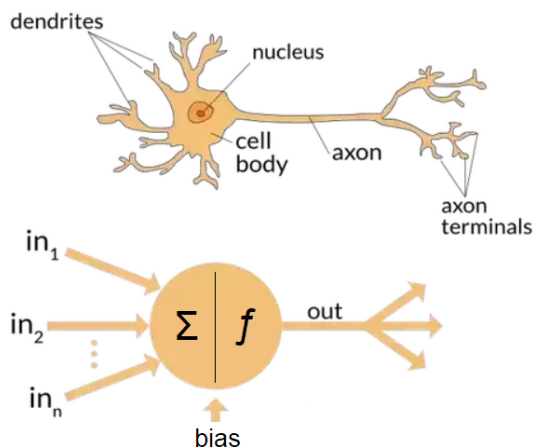
Αξίζει να σημειώσουμε πως μπορούμε να θεωρήσουμε τον SVM σαν ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο με ένα κρυφό επίπεδο νευρώνων. Αυτό έχει ως μέγεθος εισόδου την διάσταση των σημείων εκπαίδευσης, ως μέγεθος του κρυφού επιπέδου το πλήθος των διανυσμάτων υποστήριξης και ως έξοδο την συνάρτηση y . Τα βάρη από την είσοδο προς το κρυφό επίπεδο είναι ίσα με 1, ενώ τα βάρη από το κρυφό επίπεδο προς την έξοδο είναι ίσα με τους αντίστοιχους συντελεστές Lagrange. Σαν συνάρτηση ενεργοποίησης των νευρώνων του κρυφού επιπέδου χρησιμοποιείται η συνάρτηση πυρήνα. Τέλος στον νευρώνα εξόδου προστίθεται και η παράμετρος της προκατάληψης (b).

Τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) αποτελούν υπολογιστικά μοντέλα που η λειτουργία τους θυμίζει την λειτουργικότητα των νευρώνων στον άνθρωπο. Έτσι, τα κύρια συστατικά των Τεχνητών Νευρωνικών Δικτύων είναι οι *νευρώνες* που σε γενικές γραμμές προσομοιώνουν τους νευρώνες του ανθρώπινου μυαλού.

Στον άνθρωπο υπάρχουν περισσότεροι από 10 δισεκατομμύρια νευρώνες, καθένας από τους οποίους συνδέεται με κάποιες χιλιάδες άλλους νευρώνες. Το κυτταρικό σώμα του νευρώνα επεξεργάζεται τις εισερχόμενες ενεργοποιήσεις και εξάγει νέες. Οι διαφορετικοί νευρώνες στέλνουν και λαμβάνουν ενεργοποιήσεις μεταξύ τους μέσα από τους άξονες και τους δενδρίτες αντίστοιχα. Επίσης, μέσα από τις συνάψεις μπορούν και εκπέμπουν σήματα προς όλους τους άλλους νευρώνες με τους οποίους συνδέονται. Μια σύναψη ανάμεσα σε δύο νευρώνες δημιουργείται ανάμεσα στον δενδρίτη του ενός και στο τερματικό του άξονα του άλλου.

Από την άλλη, στα Τεχνητά Νευρωνικά Δίκτυα οι νευρώνες λαμβάνουν εισόδους - ενεργο-

ποιήσεις - από άλλους νευρώνες. Οι ενεργοποιήσεις πολλαπλασιάζονται με συναπτικά βάρη. Στην συνέχεια, οι σταθμισμένες ενεργοποιήσεις προστίθενται μεταξύ τους διαμορφώνοντας την συσσωρευτική μέθοδο που προσομοιάζει στην ανάλογή της στο κυτταρικό σώμα του ανθρώπινου νευρώνα. Η παράμετρος προκατάληψης προστίθεται σε αυτό το άθροισμα. Τέλος, το συνολικό άθροισμα εισχωρεί σε μια συνάρτηση ενεργοποίησης. Στην Εικόνα 2.2, αναπαριστούμε τις διαφορές του ανθρώπινου με τον τεχνητό νευρώνα.



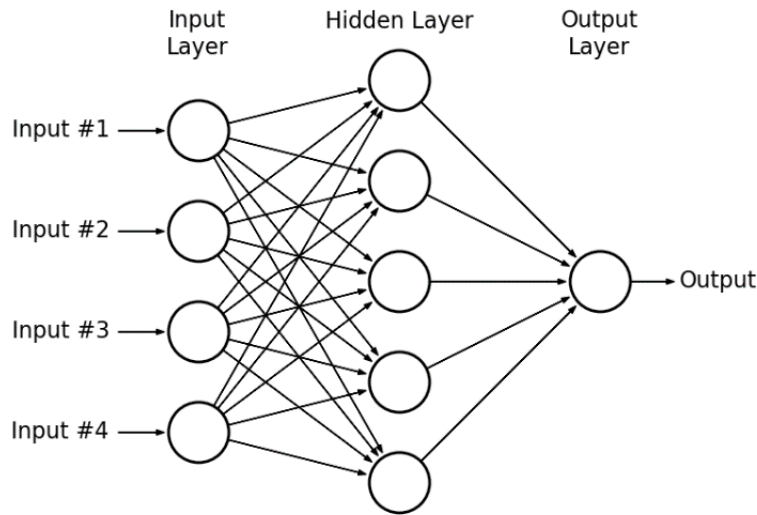
Εικόνα 2.2: Διαφορές ανθρώπινου και τεχνητού νευρώνα

Τα Τεχνητά Νευρωνικά Δίκτυα είναι κατευθυνόμενοι γράφοι που ως κόμβους έχουν νευρώνες και ως ακμές βάρη. Κάθε νευρώνας περιέχει μια συνάρτηση ενεργοποίησης και μία παράμετρο προκατάληψης. Τα βάρη και οι παράμετροι προκατάληψης αποτελούν τις παραμέτρους που το νευρωνικό δίκτυο μαθαίνει κατά την διάρκεια της εκπαίδευσης. Η διαδικασία της μάθησης απαιτεί μια συνάρτηση κόστους, την οποία προσπαθεί να βελτιστοποιήσει το νευρωνικό δίκτυο ως προς τις παραμέτρους μάθησης. Τα Τεχνητά Νευρωνικά Δίκτυα χρησιμοποιούν δύο μηχανισμούς με τους οποίους επιτυγχάνεται η Επιβλεπόμενη Μάθηση - το πρόσθιο πέρασμα (**forward propagation**) και το οπίσθιο πέρασμα (**backpropagation**) [19]. Στο πρόσθιο πέρασμα υπολογίζονται οι εξοδοί των συναρτήσεων ενεργοποίησης όλων νευρώνων του δικτύου ξεκινώντας από τους νευρώνες εισόδου και καταλήγοντας στους νευρώνες εξόδου. Από την άλλη, στο οπίσθιο πέρασμα επιτυγχάνεται η ανανέωση των παραμέτρων μάθησης βελτιστοποιώντας την συνάρτηση κόστους ως προς την αντίστοιχη παράμετρο. Οι ανανεώσεις των παραμέτρων συμβαίνουν από την αντίθετη κατεύθυνση, δηλαδή από τους νευρώνες εξόδου προς τους νευρώνες εισόδου.

2.2.3.1 Πολυεπίπεδο Perceptron

Μία από τις πιο διαδεδομένες αρχιτεκτονικές ενός Τεχνητού Νευρωνικού Δικτύου είναι το Πολυεπίπεδο Perceptron (**Multilayer Perceptron** ή **MLP**). Αυτό αποτελείται από το επίπεδο εισόδου, το επίπεδο εξόδου και τα ενδιάμεσα κρυφά επίπεδα. Στο Σχήμα 2.3, απεικονίζουμε ένα Πολυεπίπεδο Perceptron με ένα κρυφό επίπεδο. Τυπικά, εάν ένα MLP προσεγγίζει μία συνάρτηση $f = f^{(n)}(f^{(n-1)}(\dots(f^{(1)}(\mathbf{x})))$, τότε η $f^{(1)}$ είναι η συνάρτηση εισόδου, η $f^{(n-1)}$ η συνάρτηση του $(n-1)$ -στου κρυφού επιπέδου και η $f^{(n)}$ η συνάρτηση του τελευταίου επιπέδου - εξόδου. Επίσης ως βάθος του νευρωνικού δικτύου ορίζεται το n . Μάλιστα, με την

Βαθιά Μάθηση επιτυγχάνεται η αύξηση του βάθους των νευρωνικών δικτύων προκειμένου να αυξηθούν τα κρυφά επίπεδα και άρα να προκύπτουν πιο πολύπλοκες συναρτήσεις.



Σχήμα 2.3: Πολυεπίπεδο Perceptron (MLP) με ένα κρυφό επίπεδο

Σε κάθε νευρώνα i ενός επιπέδου k , η έξοδος z_i είναι το

$$z_i^{(k)} = (\mathbf{w}_i^{(k-1)})^T \mathbf{a}^{(k-1)} \quad (2.20)$$

όπου το $\mathbf{w}_i^{(k-1)}$ είναι το διάνυσμα των παραμέτρων μάθησης ανάμεσα στα επίπεδα $k-1$ και στον νευρώνα i του επιπέδου k , και το $\mathbf{a}^{(k-1)}$ είναι το διάνυσμα των εξόδων των συναρτήσεων ενεργοποίησης του επιπέδου $k-1$. Σημειώνουμε ότι στο $\mathbf{w}_i^{(k-1)}$ είναι ενσωματωμένη και η αντίστοιχη παράμετρος της προκατάληψης, όπως επίσης και το $\mathbf{a}^{(k-1)}$ είναι επαυξημένο ανάλογα έχοντας ένα επιπλέον χαρακτηριστικό που είναι πάντα 1. Η έξοδος της συνάρτησης ενεργοποίησης ενός νευρώνα i επιπέδου k είναι

$$\alpha_i^{(k)} = h(z_i^{(k)}) \quad (2.21)$$

όπου η $h(\cdot)$ είναι συνάρτηση ενεργοποίησης του νευρώνα.

2.2.3.2 Συναρτήσεις Ενεργοποίησης

Στο σημείο αυτό θα αναφερθούμε σε χρήσιμες συναρτήσεις ενεργοποίησης που θα συνηθίσουμε στην συνέχεια. Η επιλογή της σωστής συνάρτησης ενεργοποίησης είναι ιδιαίτερα σημαντική για το νευρωνικό δίκτυο. Πολλές φορές χρησιμοποιούμε διαφορετικές συναρτήσεις ενεργοποίησης για το επίπεδο εξόδου από τα άλλα επίπεδα, προκειμένου να εκφράσουμε την έξοδο με μορφή πιθανότητας.

Μία κλασική συνάρτηση ενεργοποίησης είναι η σιγμοειδής (**sigmoid**). Εκφράζεται ως

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.22)$$

Η σιγμοειδής συνάρτηση λαμβάνει ως είσοδο οποιονδήποτε πραγματικό αριθμό και εξάγει

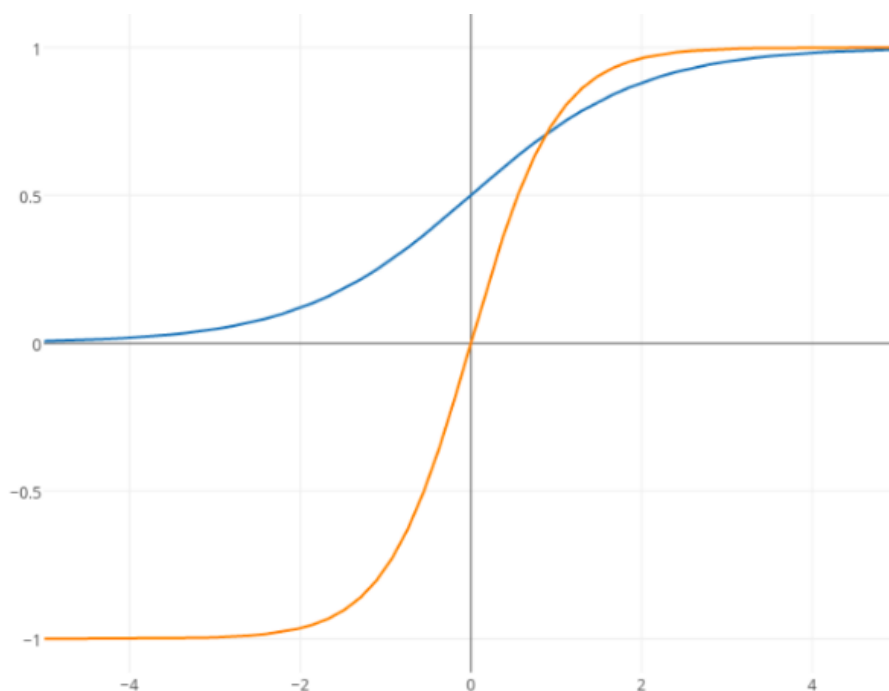
έναν αριθμό που ανήκει στο $[0, 1]$. Ένα μοντέλο μπορεί να χρησιμοποιήσει την σιγμοειδή ως συνάρτηση ενεργοποίησης στο επίπεδο εξόδου του στην περίπτωση της δυαδικής ταξινόμησης. Αυτό γίνεται με τρόπο ώστε εάν η έξοδος της είναι μεγαλύτερη από 0.5 τότε η είσοδος ταξινομείται στην πρώτη κλάση, αλλιώς στην δεύτερη.

Ωστόσο, η σιγμοειδής δεν θεωρείται σήμερα αρκετά καλή επιλογή ως συνάρτηση ενεργοποίησης των κρυφών επιπέδων. Σύμφωνα με τον Le Cun και άλλους [20] το νευρωνικό δίκτυο συγκλίνει πιο γρήγορα όταν η μέση τιμή των χαρακτηριστικών εισόδου τείνει στο 0. Αυτό ωστόσο επεκτείνεται και για τα επόμενα επίπεδα. Κάτι τέτοιο, όμως, δεν μπορεί να γίνει με την σιγμοειδή αφού η μέση τιμή θα είναι κοντά στο 0.5 και όχι στο 1.

Μία συνάρτηση που προσπερνά αυτό το εμπόδιο είναι η υπερβολική εφαπτομένη ***tanh***, η οποία αποτελεί μετατοπισμένη σιγμοειδή. Δίνεται από την σχέση

$$\tanh(x) = \frac{e^x - e^{-x}}{e^{-x} + e^x} \quad (2.23)$$

Ισχύει ότι $|\tanh(x)| \leq 1$ Επίσης, σε περιπτώσεις που έχουμε πολλά δεδομένα και θέλουμε πιο γρήγορα να φθάσουμε σε σύγκλιση, προτιμάται η *tanh* γιατί έχει κάπως μεγαλύτερη κλίση (gradient) από την σιγμοειδή.



Σχήμα 2.4: Αναπαράσταση sigmoid με *tanh* σε κοινό διάγραμμα. Με μπλε αναπαρίσταται η sigmoid, ενώ με πορτοκαλί η *tanh*.

Ωστόσο, ένας παράγοντας που καθιστά κάποιες φορές την σιγμοειδή και την *tanh* απαγορευτικές ως συναρτήσεις ενεργοποίησης στη Βαθιά Μάθηση είναι πως οι παράγωγοί τους έχουν μικρές τιμές. Ιδιαίτερα, όπως είπαμε, αυτό ισχύει για την σιγμοειδή. Κατά την διάρκεια του backpropagation, όταν πολλά κρυφά επίπεδα χρησιμοποιούν την σιγμοειδή σαν συνάρτηση ενεργοποίησης τότε τόσες μικρές τιμές παραγώγων (gradients) πολλαπλασιάζονται μαζί. Έτσι, το gradient μικραίνει εκθετικά καθώς προχωράμε πίσω στα πρώτα επίπεδα.

Μάλιστα, το *gradient* μικραίνει τόσο πολύ που φθάνουμε σε σημεία να σταματά η μάθηση. Αυτό είναι το φαινόμενο της εξαφανιζόμενης κλίσης (**vanishing gradient**).

Μία συνάρτηση ενεργοποίησης που επιλύει αυτό το πρόβλημα είναι η **ReLU** [21], η οποία ορίζεται ως εξής

$$\text{ReLU}(x) = \max(x, 0) \quad (2.24)$$

Η συνάρτηση αυτή έχει ως παράγωγο 1 για θετική είσοδο και το 0 για αρνητική. Για είσοδο 0 η παράγωγος δεν ορίζεται, αν και από σύμβαση συνήθως επιλέγεται να εξάγει και πάλι το 0, ώστε το αντίστοιχο διάνυσμα να είναι ακόμα πιο αραιό.

Όσον αφορά τη συναρτήση ενεργοποίησης του επιπέδου εξόδου για περιπτώσεις ενός προβλήματος ταξινόμησης με περισσότερες από δύο (M) κλάσεις, τότε αντί της σιγμοειδούς, συνήθως χρησιμοποιούμε την **Softmax**, η οποία δίνεται από την σχέση

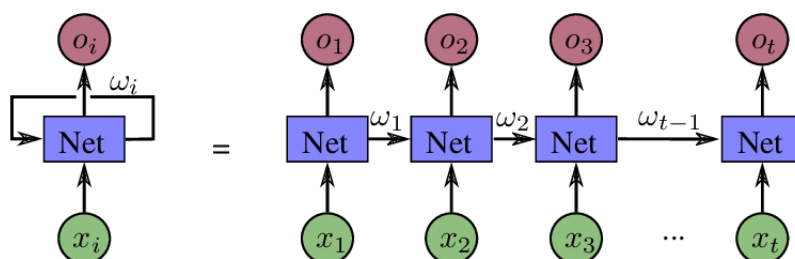
$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^M e^{x_j}} \quad (2.25)$$

Με την συνάρτηση αυτή, κανονικοποιείται η είσοδος και το νευρωνικό δίκτυο εξάγει σε κάθε έξοδο μια τιμή πιθανότητας. Έτσι, η έξοδος με την μεγαλύτερη τιμή πιθανότητας ταξινομεί την είσοδο του νευρωνικού στην αντίστοιχη κλάση.

Τέλος, στην περίπτωση της παλινδρόμησης (regression), θέλουμε να εξάγουμε μια συνεχή τιμή. Έτσι, η συνάρτηση ενεργοποίησης του τελικού νευρώνα πρέπει να είναι η ταυτοτική, δηλαδή η $f(x) = x$.

2.2.3.3 Αναδρομικά Νευρωνικά Δίκτυα

Μία μεγάλη υποκατηγορία των Τεχνητών Νευρωνικών Δικτύων αποτελούν τα Αναδρομικά Νευρωνικά Δίκτυα (**Recurrent Neural Networks** ή **RNNs**) [22]. Σε αντίθεση με ένα MLP, ένα RNN δεν έχει σταθερό αριθμό νευρώνων εισόδων αλλά λαμβάνει ως είσοδο μια ολόκληρη αλληλουχία μη σταθερού μεγέθους. Αυτό γίνεται με τη χρήση μιας εσωτερικής κατάστασης που δουλεύει ως μνήμη του δικτύου. Το RNN, βασιζόμενο και πάλι στο forward και backpropagation, μπορεί και διαμορφώνει μια προσωρινή δυναμική συμπεριφορά με το να "ξετυλίγει" την αλληλουχία εισόδου, ενημερώνοντας κάθε φορά την εσωτερική του μνήμη. Ένα τέτοιο δίκτυο με αλληλουχία πεπερασμένου μήκους μπορεί να ξετυλιχτεί σε ένα Κατευθυνόμενο Ακυκλικό Γράφο (DAG) μέσω ενός πρόσθιου νευρωνικού δικτύου (feedforward neural network), διατηρώντας, ωστόσο, την δυναμική του συμπεριφορά.



Σχήμα 2.5: Ξετύλιγμα RNN

Μια αλληλουχία σαν είσοδος μπορεί να είναι μια τυπική χρονοσειρά δεδομένων. Έτσι, ορίζεται η έννοια του χρονικού βήματος (timestep), που αφορά την στιγμή κατά την οποία το RNN διαβάζει την αντίστοιχη είσοδο της ακολουθίας. Πολλές εφαρμογές μπορούμε να τις εκφράσουμε έμμεσα σαν χρονοσειρές για παράδειγμα να θεωρήσουμε ως χρονικά βήματα τις λέξεις ενός κειμένου ή τα φωνήματα κατά την επεξεργασία της φωνής. Με την άνθηση της Βαθιάς Μάθησης, τα RNNs έχουν γνωρίσει φοβερή αναγνωρισιμότητα σε ανάλογα προβλήματα, όπως η ταξινόμηση κειμένων, η αυτόματη μετάφραση φράσεων, η ανάκτηση πληροφορίας κ.ά.

Το RNN είναι αναδρομικό, καθώς η έξοδος της κατάστασής του σε ένα χρονικό βήμα αποτελεί την είσοδο της κατάστασής του στο αμέσως επόμενο. Αυτή η κατάσταση - μνήμη - αποκαλείται *hidden state* το οποίο παράγεται από το λεγόμενο hidden layer. Το ξετύλιγμα του RNN περιγράφεται στο Σχήμα 2.5. Το hidden state και η έξοδος σε κάθε χρονικό βήμα t υπολογίζονται από τις σχέσεις

$$\mathbf{h}_t = f(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{U}_h \mathbf{x}_t + \mathbf{b}_h) \quad (2.26)$$

$$\mathbf{y}_t = f(\mathbf{W}_y \mathbf{h}_t + \mathbf{b}_y) \quad (2.27)$$

όπου η f είναι η element-wise συνάρτηση ενεργοποίησης και τα \mathbf{h}_t , \mathbf{W}_h , \mathbf{U}_h , \mathbf{y}_t , \mathbf{b}_h και \mathbf{b}_y είναι οι παράμετροι μάθησης του δικτύου.

Έναν από τους πιο διαδεδομένους τύπους RNN αποτελεί το **Long Short-Term Memory** ή **LSTM** [23]. Αυτό το δίκτυο φημίζεται για την διαχείριση εισόδων σε αλληλουχίες, καθώς διατηρεί μακροχρόνιες μνήμες. Η αρχιτεκτονική ενός LSTM αναπαρίσταται στο Σχήμα 2.6. Όπως βλέπουμε, αποτελείται από ένα *cell state* και τρεις θύρες: το *input gate*, το *forget gate* και το *output gate*. Καθένα από αυτά τα συστατικά καλύπτει μια σημαντική λειτουργικότητα του LSTM. Πιο αναλυτικά, τα συστατικά αυτά έχουν τις εξής λειτουργίες:

- **Input gate:** Ελέγχει σε ποιον βαθμό η είσοδος του τρέχοντος χρονικού βήματος θα επηρεάσει το cell.
- **Forget gate:** Αποφασίζει ποιες πληροφορίες την μνήμης και σε ποιο βαθμό θα κρατη-

θούν ή θα διαγραφούν.

- Output gate: Ελέγχει ποιες πληροφορίες θα περάσουν σαν έξοδος στο τρέχον χρονικό βήμα.
- Cell state: Διαρκώς ενημερώνεται αλληλεπιδρώντας με τις τρεις θύρες.

Τυπικά, οι μαθηματικές σχέσεις που διέπουν ένα LSTM είναι οι ακόλουθες

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (2.28)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (2.29)$$

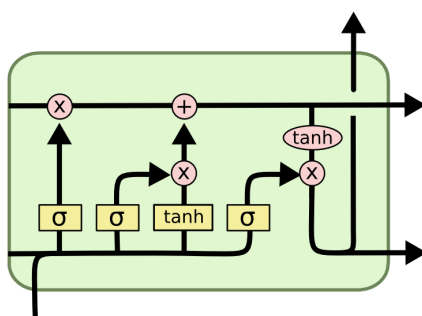
$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (2.30)$$

$$\mathbf{u}_t = \tanh(\mathbf{W}_u \mathbf{x}_t + \mathbf{U}_u \mathbf{h}_{t-1} + \mathbf{b}_u) \quad (2.31)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{u}_t \quad (2.32)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (2.33)$$

όπου \mathbf{W} , \mathbf{U} αποτελούν τα βάρη και οι δείκτες f, i, o αποτελούν τις αντίστοιχες θύρες.



Σχήμα 2.6: Long short-term memory

2.2.3.4 Υπερπαράμετροι Μάθησης

Στο σημείο αυτό, θα αναφέρθουμε στις υπερπαραμέτρους της μάθησης ενός νευρωνικού δικτύου. Στην φάση της μάθησης, η περίοδος κατά την οποία το νευρωνικό δίκτυο εκπαιδεύεται πάνω σε όλο το σύνολο εκπαίδευσης έχοντας δει κάθε παρατήρηση ακριβώς μία φορά ονομάζεται **εποχή** (epoch). Όταν λέμε πως το νευρωνικό δίκτυο εκπαιδεύεται, εννοούμε ότι ανανεώνει τις παραμέτρους μάθησης, δηλαδή τα βάρη και τις προκαταλήψεις. Ωστόσο, επειδή τις περισσότερες φορές το σύνολο εκπαίδευσης είναι αρκετά μεγάλο, μέσα σε μία εποχή ανανεώνουμε αρκετές φορές (M) τις παραμέτρους μάθησης σπάζοντας το σύνολο εκπαίδευσης σε ίσα M αυστηρώς διαφορετικά κομμάτια. Κάθε τέτοιο κομμάτι ονομάζεται παρτίδα (**batch**).

Ωστόσο, με ποιον τρόπο θα ανανεώνουμε τις παραμέτρους μάθησης; Η απάντηση είναι με τους βελτιστοποιητές (**optimizers**). Ήδη έχουμε αναφερθεί στον ΣΓΔ με τον οποίο λύναμε το πρόβλημα ταξινόμησης με τον αλγόριθμο Perceptron. Σημειώνουμε πως όλοι οι optimizers χρειάζονται τον ρυθμό μάθησης (**learning rate**) να μειώνεται με την πάροδο των εποχών. Αυτό γιατί με ένα μεγάλο ή σταθερό learning rate, το μοντέλο είναι αρκετά επιρρεπές να κολλήσει σε τοπικά βέλτιστα. Το αρνητικό με τον SGD είναι ότι χρειάζεται πολύ χρόνο για να συγκλίνει. Αντ' αυτού άλλοι optimizers συχνά χρησιμοποιούνται, όπως ο RMSProp [24] ή ο Adam [25].

2.2.3.5 Το Πρόβλημα της Υπερεκπαίδευσης

Η υπερεκπαίδευση (**overfitting**) αποτελεί την περίπτωση στην οποία ένα μοντέλο Μηχανικής Μάθησης μαθαίνει υπέρ του δέοντος τα δεδομένα εκπαίδευσης, με αποτέλεσμα αυτό να μην μπορεί να γενικεύσει την μάθησή του σε νέες παρατηρήσεις. Με άλλα λόγια το μοντέλο δεν μαθαίνει, αλλά απομνημονεύει το σύνολο εκπαίδευσης [26].

Αυτό μπορεί να συμβεί όταν το μοντέλο είναι πάρα πολύ πολύπλοκο, με αποτέλεσμα να προσεγγίζει μια πολύπλοκη μη γραμμική συνάρτηση που επιλύει πολύ καλά το πρόβλημα. Επίσης μια άλλη αιτία υπερεκπαίδευσης είναι οι πολλές εποχές μάθησης χωρίς να ελαττώνεται με κάποιο ρυθμό το learning rate. Συνήθως, η μάθηση γίνεται στο σύνολο εκπαίδευσης αλλά παράλληλα στο τέλος κάθε εποχής το μοντέλο δοκιμάζεται και σε ένα ανεξάρτητο σύνολο - σύνολο επαλήθευσης (validation set). Συνήθως, η ένδειξη ότι εμφανίζεται υπερεκπαίδευση του μοντέλου είναι ότι η επίδοση στο σύνολο εκπαίδευσης συνεχίζει να βελτιώνεται, αλλά στο σύνολο επαλήθευσης μειώνεται ή ταλαντώνεται.

Δύο μέθοδοι που αντιμετωπίζουν το φαινόμενο αυτό είναι το πρόωρο σταμάτημα της μάθησης (early stopping) [27] και το ορόσημο (checkpoint) της μάθησης την εποχή εκείνη που είχε παρατηρηθεί η καλύτερη επίδοση στο validation set. Μία ακόμα μέθοδος είναι το να αγνοείς με τυχαίο τρόπο νευρώνες από το forward και backpropagation. Με άλλα λόγια συγκεκριμένοι νευρώνες σε κάθε ενημέρωση των παραμέτρων με πιθανότητα p αγνοούνται και με $1 - p$ παράμενον ενεργοί. Αυτό έχει ως αποτέλεσμα την προσθήκη θορύβου κατά τη διάρκεια της μάθησης. Έτσι, το μοντέλο αναγκάζεται να μάθει πιο σημαντικά χαρακτηριστικά χρησιμοποιώντας κάθε φορά ένα υποσύνολο των νευρώνων του. Η παραπάνω τεχνική ονομάζεται απόρριψη (dropout) [28].

2.3 Ενεργός Μάθηση

Η Ενεργός Μάθηση (**Active Learning**) αποτελεί έναν τύπο της Μηχανικής Μάθησης. Κύρια υπόθεση της Ενεργού Μάθησης είναι εάν ένας αλγόριθμος μάθησης μπορεί να καταλάβει ποιες παρατηρήσεις είναι πιο σημαντικές, να μειώσει τον συνολικό αριθμό των δεδομένων εκπαίδευσης. Αυτό, ωστόσο, πρέπει να γίνεται με τρόπο ώστε να φθάνει και πάλι σε σύγκλιση. Εμείς θα περιοριστούμε στην Επιβλεπόμενη Ενεργό Μάθηση.

Όπως γνωρίζουμε, τα μοντέλα Επιβλεπόμενης Μάθησης συνήθως χρειάζονται αρκετά δεδομένα εκπαίδευσης, ώστε να φθάσουν σε σύγκλιση [3]. Μερικές φορές, σε πραγματικού χρόνου προβλήματα, οι επισημειώσεις των παρατηρήσεων είναι ελάχιστα ή καθόλου διαθέσιμες, όπως στην περίπτωση του email spamming. Σε τέτοιες περιπτώσεις, θα θέλαμε να εκπαιδεύσουμε μοντέλα Επιβλεπόμενης Μάθησης χρησιμοποιώντας όσο το δυνατόν ελάχιστες επισημειώσεις, λαμβάνοντας υπόψιν πως η διαδικασία απόκτησης όλων των επισημειώσεων μπορεί να είναι αρκετά χρονοβόρα, δύσκολη ή/και ακριβή. Σημειώνουμε πως σε αυτές τις περιπτώσεις το σύνολο δεδομένων μπορεί να είναι πολύ μεγάλο, χωρίς ωστόσο να έχουμε πολλές επισημειώσεις.

Έτσι, η Ενεργός Μάθηση ρωτά (query) έναν ειδικό (*Oracle*), για παράδειγμα έναν άνθρωπο επισημειωτή, ώστε να αποκτήσει επισημειώσεις για έναν περιορισμένο αριθμό παρατηρήσεων υποθέτοντας πως θα είναι οι λιγότερες απαραίτητες για υψηλές επιδόσεις. Επίσης, η χρήση της Ενεργού Μάθησης συνηθίζεται αρκετά σε περιπτώσεις που δεν έχουμε καθόλου σύνολο δεδομένων ή έχουμε ένα αρχικό αλλά αρκετά περιορισμένου μεγέθους. Σε τέτοιες περιπτώσεις, θέλουμε να αποκτήσουμε νέα επισημειωμένα δεδομένα έχοντας τη δυνατότητα της επιλογής ανάμεσα σε ορισμένα διαθέσιμα μη επισημειωμένα.

Η τυπική διαδικασία της Ενεργού Μάθησης είναι η εξής: το μοντέλο εκπαιδεύεται πάνω στα επισημειωμένα δεδομένα του. Στην συνέχεια ρωτάει τον Oracle, βάσει κάποιου επιλεγμένου κριτηρίου, έναν συγκεκριμένο αριθμό παρατηρήσεων για επισημειώσεις. Ονομάζουμε την διαδικασία αυτή ως *γύρο* της Ενεργού Μάθησης. Ένας νέος γύρος θα συμβαίνει μέχρις ότου το μοντέλο να φθάσει σε σύγκλιση ή αυτό να ξεπεράσει ένα προκαθορισμένο μέγιστο όριο ερωτήσεων που έχει οριστεί. Ονομάζουμε *batch* της Ενεργού Μάθησης τον αριθμό των ερωτήσεων που συμβαίνουν σε κάθε γύρο.

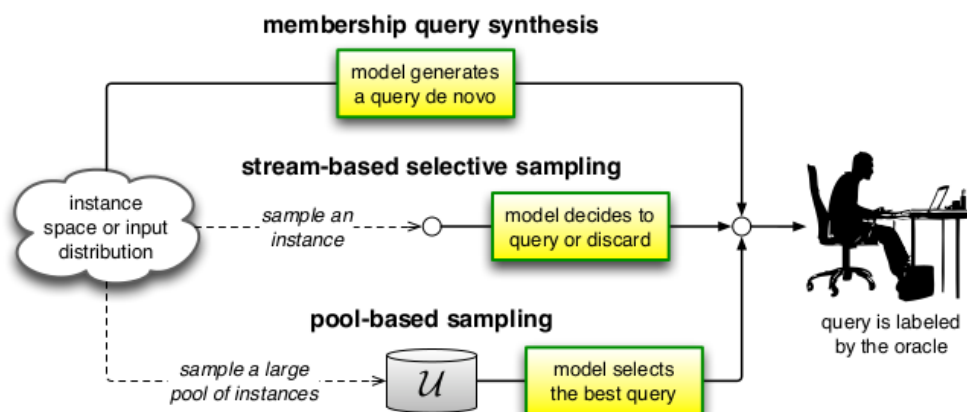
2.3.1 Διαφορές Με Ημιεπιβλεπόμενη Μάθηση

Σε γενικές γραμμές, η Ενεργός Μάθηση προσπαθεί να λύσει ένα κοινό πρόβλημα με την Ημιεπιβλεπόμενη Μάθηση (**Semi-Supervised Learning**). Το πρόβλημα της Ημιεπιβλεπόμενης Μάθησης σχετίζεται με την εύρεση ενός τρόπου για επαυξήση του αρχικού συνόλου δεδομένων εκπαίδευσης - το οποίο περιέχει λίγες επισημειωμένες παρατηρήσεις - ώστε το μοντέλο να φθάσει με βέλτιστο τρόπο σε σύγκλιση. Η διαδικασία της μάθησης εδώ δεν περιλαμβάνει τη χρήση κάποιου Oracle. Ωστόσο, κάθε φορά που το μοντέλο εκπαιδεύεται, τότε επαυξάνεται το σύνολο εκπαίδευσης με εκείνες τις παρατηρήσεις που το μοντέλο είναι πιο σίγουρο για τις αντίστοιχες επισημειώσεις τους. Η διαδικασία αυτή, όπως και στην Ενεργό Μάθηση, διαρκεί μέχρι το μοντέλο να φθάσει σε σύγκλιση. Παρ' όλα αυτά, επειδή δεν διεξάγεται κάποια διαδικασία ανθρώπινης επισημείωσης, μπορεί η μάθηση να συνεχιστεί μέχρι

το μοντέλο να επισημειώσει όλες τις διαθέσιμες παρατηρήσεις.

2.3.2 Κατηγορίες Ενεργού Μάθησης

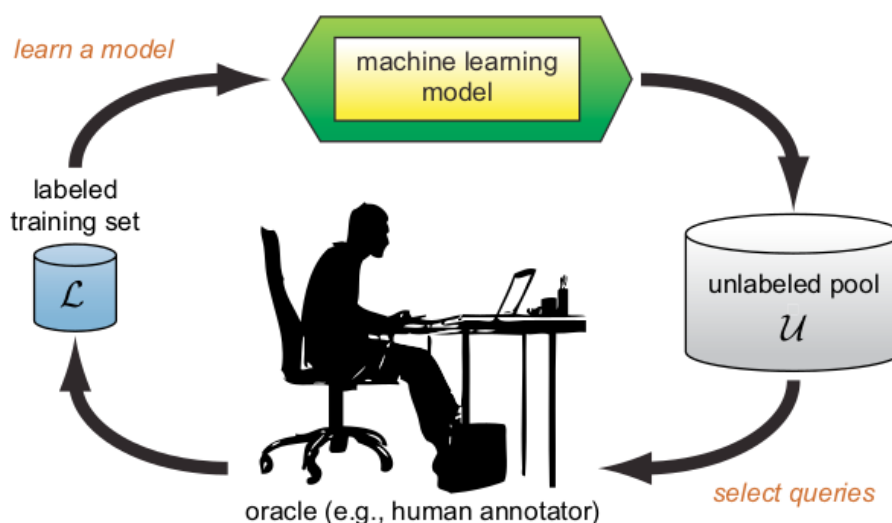
Οι βασικές κατηγορίες - σενάρια - Ενεργού Μάθησης είναι οι εξής: **Membership Query Synthesis**, **Stream-Based Selective Sampling** και **Pool-Based Sampling**, οι οποίες αναπαρίστανται συγκεντρωτικά στο Σχήμα 2.7.



Σχήμα 2.7: Active Learning. Πηγή [3]

- Membership Query Synthesis:** Στην κατηγορία αυτή, το μοντέλο ρωτάει για επισημειώσεις πάνω σε παρατηρήσεις που συνθέτει το ίδιο εκ νέου, χρησιμοποιώντας μια δοσμένη κατανομή. Παρ' όλα αυτά οι παρατηρήσεις που συνθέτονται μερικές φορές είναι διφορούμενες για την διαδικασία επισημείωσης από τον Oracle, όπως στην περίπτωση της αναγνώρισης χειρόγραφου. Επιπλέον, υπάρχουν περιπτώσεις που τα συνθετικά δεδομένα δεν αντικατοπτρίζουν πραγματικές παρατηρήσεις, πχ. όταν το πρόβλημα αφορά κείμενα.
- Stream-Based Selective Sampling:** Εδώ τα δεδομένα έρχονται σειριακά σε μορφή ροής (stream), ένα-ένα, υποθέτοντας ότι η απόκτηση των μη επισημειωμένων δεδομένων περιλαμβάνει μια μη ακριβή διαδικασία. Κάθε φορά που το κανάλι φέρνει μια νέα παρατήρηση, το μοντέλο εκτιμά, κατά πόσο αξίζει να δοθεί σε αυτήν η αντίστοιχη επισημείωση, και αποφασίζει ανάλογα. Η stream-based Ενεργός Μάθηση συνηθίζεται να χρησιμοποιείται σε εφαρμογές πραγματικού χρόνου, όπως στην ανάκτηση πληροφορίας [29] ή το POS Tagging [16].
- Pool-Based Sampling:** Η κατηγορία αυτή προϋποθέτει την ύπαρξη ενός μεγάλου μη επισημειωμένου συνόλου δεδομένων, που ονομάζεται *pool*, και ενός αρχικού μικρού μεγέθους επισημειωμένου συνόλου εκπαίδευσης. Το μοντέλο εκπαιδεύεται σε κάθε γύρο πάνω στο σύνολο εκπαίδευσης και εκτιμά όλες τις παρατηρήσεις του pool ρωτώντας με άπληστο συνήθως τρόπο τον Oracle για εκείνες που θεωρεί πιο σημαντικές. Στο Σχήμα 2.8 αναπαρίσταται η παραπάνω διαδικασία. Η pool-based Ενεργός Μάθηση έχει επίσης χρησιμοποιηθεί σε πολλές εφαρμογές πραγματικού χρόνου, συμπερι-

λαμβανομένων προβλημάτων ταξινόμησης κειμένων [30], ανάκτησης πληροφορίας [31] κά.



Σχήμα 2.8: Pool-Based Active Learning. Πηγή [3]

2.3.3 Στρατηγικές Ενεργού Μάθησης

Αφού έχουμε περιγράψει το τι προσπαθεί να επιλύσει η Ενεργός Μάθηση μέσα από τα διαφορετικά της σενάρια, στο σημείο αυτό θα συζητήσουμε για τις στρατηγικές ερωτημάτων (**Querying Strategies**) με τις οποίες επιλέγονται σε κάθε γύρο οι ερωτήσεις προς τον Oracle.

2.3.3.1 Στρατηγικές Βασισμένες στην Ετερογένεια

Οι στρατηγικές στην κατηγορία αυτή αφορούν μοντέλα με τα οποία επιλέγονται για ερωτήσεις οι παρατηρήσεις εκείνες που είναι πιο ετερογενείς σε σχέση με αυτές που υπάρχουν στο σύνολο εκπαίδευσης - **Heterogeneity-based models** [32]. Αυτό μπορεί να θεωρηθεί ως η αβεβαιότητα σε ένα πρόβλημα ταξινόμησης ή, εναλλακτικά, ως η διαφωνία ανάμεσα σε ένα σετ διαφορετικών ταξινομητών.

Η πιο διαδομένη μέθοδος είναι η Δειγματοληψία Αβεβαιότητας (**Uncertainty Sampling**), κατά την οποία οι ερωτήσεις προς τον Oracle επιλέγονται με βάση την αβεβαιότητα του μοντέλου προς τις διαθέσιμες παρατηρήσεις. Σε προβλήματα ταξινόμησης, οι τρεις κύριες μέθοδοι Uncertainty Sampling είναι οι εξής: με το να επιλέγονται οι παρατηρήσεις που είναι λιγότερο αισιόδοξες (*least confident*), με το να επιλέγονται οι παρατηρήσεις μέσω Περιθώριας Δειγματοληψίας (*Margin Sampling*) του μοντέλου ή με το να επιλέγονται οι παρατηρήσεις που εκφράζουν την μεγαλύτερη εντροπία (*entropy*) [3]. Σημειώνουμε πως προκειμένου ένα μοντέλο Επιβλεπόμενης Μάθησης να μπορεί να εκπαιδευτεί με Ενεργό Μάθηση βασισμένη στο Uncertainty Sampling, δεν πρέπει κατά ανάγκη αυτό να είναι μπεύσιανό. Πρακτικά, αρκεί το μοντέλο να εξάγει για κάθε έξοδο μια τιμή που είναι κανονικοποιημένη και μπορεί να εκφραστεί ως τιμή πιθανότητας.

Μία άλλη μέθοδος είναι η Ερώτηση Επιλεγμένη Από Επιτροπή (**Query-by-Committee**). Εδώ υπάρχει ένα σετ από ταξινομητές, οι οποίοι σε κάθε γύρο προβλέπουν την ταξινόμηση όλων των διαθέσιμων μη επισημειωμένων παρατηρήσεων. Έτσι, επιλέγονται για ερώτηση εκείνες οι παρατηρήσεις με τις περισσότερες διαφωνίες των ταξινομητών.

Τέλος, μια άλλη μέθοδος είναι η Αναμενόμενη Αλλαγή Μοντέλου (**Expected Model Change**). Στην περίπτωση αυτή, επιλέγονται εκείνες οι παρατηρήσεις για τις οποίες εάν γνωρίζαμε τις επισημειώσεις τους, θα αναμενόταν η μεγαλύτερη αλλαγή στην κλίση (gradient) της συνάρτησης κόστους του μοντέλου. Έτσι, επιλέγουμε κάθε φορά τις παρατηρήσεις που ενδέχεται να επηρεάσουν περισσότερο το μοντέλο.

2.3.3.2 Στρατηγικές Βασισμένες στην Επίδοση

Οι στρατηγικές βασισμένες στην επίδοση αφορούν μοντέλα (**Performance-based models**) που επιλέγουν να ρωτάνε μη επισημειωμένες παρατηρήσεις με βάση τις προβλέψεις τους σε αυτές. Μία τέτοια στρατηγική είναι η Αναμενόμενη Μείωση Λάθους (**Expected Error Reduction**). Σε αντίθεση με το Uncertainty Sampling, εδώ επιλέγονται παρατηρήσεις που όταν προστεθούν στο σύνολο εκπαίδευσης αναμένεται να μειωθεί μέγιστα η αβεβαιότητα του μοντέλου για αυτές.

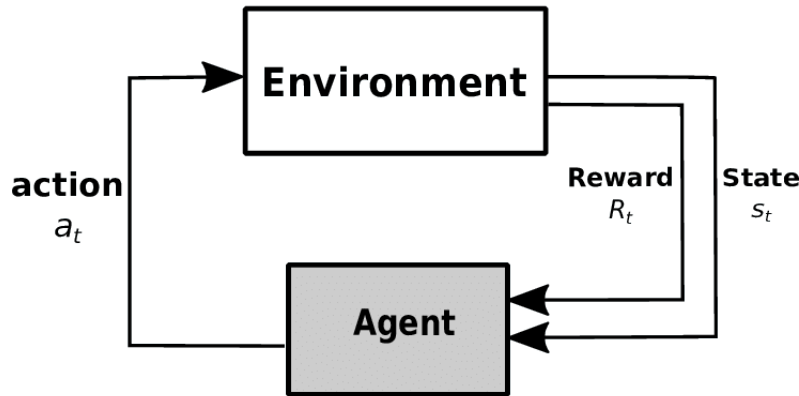
2.3.3.3 Στρατηγικές Βασισμένες στην Αντιπροσωπευτικότητα

Οι στρατηγικές αυτές αφορούν μοντέλα που προσπαθούν να δημιουργήσουν όσο το δυνατόν πιο αντιπροσωπευτικές παρατηρήσεις με αυτές που βρίσκονται στο σύνολο εκπαίδευσης. Τέτοια μοντέλα λέγονται βασισμένα στην αντιπροσωπευτικότητα (**Representativeness-based models**) [32]. Ένα παράδειγμα αυτών αποτελούν τα μοντέλα πυκνότητας (**Density-based models**) που χρησιμοποιούν ένα κριτήριο ετερογένειας σε συνδυασμό με ένα κριτήριο αντιπροσωπευτικότητας των παρατηρήσεων.

2.4 Ενισχυτική Μάθηση

Στην ενότητα αυτή, θα συζητήσουμε για κάποιες βασικές έννοιες της Ενισχυτικής Μάθησης (Reinforcement Learning ή RL) που χρειάζονται για τις απαιτήσεις της παρούσας διπλωματικής εργασίας.

Όπως αναφέραμε και στην ενότητα 2.1.3, το πρόβλημα της Ενισχυτικής Μάθησης περιλαμβάνει έναν πράκτορα (**agent**) που δραστηριοποιείται σε ένα περιβάλλον (**environment**). Το περιβάλλον αυτό παρέχει καταστάσεις **states**. Σε κάθε χρονικό βήμα $t + 1$, ο πράκτορας μεταπηδά από ένα state S_t προς κάποιο άλλο S_{t+1} επιλέγοντας κάποια διαθέσιμη δράση (**action**) A_t . Όταν ο πράκτορας μεταπηδά στο νέο state τότε το περιβάλλον του παρέχει μια επιβράβευση (**reward**) για το πόσο καλό είναι το state αυτό. Η διαδικασία αυτή περιγράφεται από το Σχήμα 2.9.



Σχήμα 2.9: Ενισχυτική Μάθηση

Στην περίπτωση αυτή, το περιβάλλον παρέχει στον πράκτορα μια ολοκληρωμένη εικόνα της πραγματικότητας, και για αυτό μιλάμε για καταστάσεις (states). Ένα τέτοιο περιβάλλον λέγεται πλήρως παρατηρήσιμο (*fully-observable*). Αντίθετα, σε περιπτώσεις όπου το περιβάλλον παρέχει μια πιο περιορισμένη εικόνα, ορίζεται η έννοια της παρατήρησης (*observation*) αντί της κατάστασης και το περιβάλλον αυτό καλείται μερικώς παρατηρήσιμο (*partially observable*).

Σκοπός του πράκτορα είναι να μεγιστοποιήσει τη μελλοντική συσσωρευτική επιβράβευση (cumulative reward). Έτσι, ο πράκτορας προσπαθεί να βρει μια βέλτιστη στρατηγική, ή πολιτική, (**policy**), ώστε να πετύχει αυτόν το στόχο. Αξίζει να σημειώσουμε πως στην παραπάνω περιγραφή του περιβάλλοντος, θεωρούμε πως το reward δίνεται στον πράκτορα την στιγμή που αυτός μεταβαίνει σε κάποιο άλλο state. Από την άλλη, υπάρχουν αρκετές περιπτώσεις που αυτό δεν είναι δυνατό να συμβεί ή δεν είναι χρήσιμο να συμβεί. Ωστόσο, στη δική μας πειραματική μελέτη ο πράκτορας θα λαμβάνει το reward τη στιγμή που επισκέπτεται το νέο state.

Ένα πρόβλημα Ενισχυτικής Μάθησης λέμε ότι έχει άπειρο ορίζοντα (**infinite horizon**) όταν μεταβαίνει καταστάσεις επί άπειρον, χωρίς να προβλέπεται ένα τερματικό state. Αντίθετα, όταν το πρόβλημα περιέχει τερματικά states τότε, στο πρόβλημα, συνήθως ορίζεται η έννοια του επεισοδίου (*episode, trial* ή *trajectory*). Τέτοιες περιπτώσεις λέμε ότι έχουν πεπερασμένο ορίζοντα (**finite horizon**). Οι περιπτώσεις στις οποίες δεν ορίζονται επεισόδια λέγονται **non episodic tasks**.

2.4.1 Επιβράβευση και Ανταμοιβή

Όπως αναφεράμε παραπάνω, σκοπός του πράκτορα είναι να βρει μια βέλτιστη στρατηγική (optimal policy) προκειμένου να μεγιστοποιήσει το συσσωρευτικό reward - επιβράβευση. Αυτό ορίζεται ως

$$r_t = R(S_t, A_t, S_{t+1}) \quad (2.34)$$

όπου η R είναι η συνάρτηση της επιβράβευσης (reward function). Επίσης συχνά στη βιβλιογραφία το reward έχει ως δείκτη το $t + 1$, καθώς ο πράκτορας τυπικά το λαμβάνει σε εκείνο το χρονικό βήμα. Ο ορισμός της reward function είναι καθοριστικής σημασίας, καθώς εάν ο

πράκτορας λαμβάνει θορυβώδη rewards μπορεί να μην φθάσει ποτέ σε βέλτιστη στρατηγική. Με την έννοια θορυβώδη, αναφερόμαστε σε rewards τα οποία δεν ανταποκρίνονται ορθά σε μια μετάβαση του πράκτορα ή δεν τον βοηθούν να καταλάβει τους στόχους του.

Όταν μιλάμε για συσσωρευτική επιβράβευση, αναφερόμαστε στον όρο της *ανταμοιβής* (return). Η ανταμοιβή μπορεί να είναι πεπερασμένη είτε άπειρη. Σε περιπτώσεις που είναι πεπερασμένη, δηλαδή όταν ορίζεται κάποιο τερματικό state, συχνά χρησιμοποιείται ο παρακάτω τύπος

$$R_t = \sum_{k=0}^T r_{t+k+1} \quad (2.35)$$

όπου T είναι ο πεπερασμένος αριθμός χρονικών βημάτων. Όσο το T προσεγγίζει το άπειρο τότε θεωρητικά το άνω όριο του R_t είναι το άπειρο. Έτσι, για περιπτώσεις μη πεπερασμένου αριθμού χρονικών βημάτων ορίζουμε την ανταμοιβή G_t ως εξής

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (2.36)$$

όπου $\gamma \in [0, 1]$ είναι ο εκπτώτικος παράγοντας (discounted factor). Το G_t λέγεται ανταμοιβή με έκπτωση (discounted return).

Το γ είναι πολύ σημαντικό για τον καθορισμό της ανταμοιβής του πράκτορα. Παρατηρούμε πως εάν $\gamma = 0$, τότε ο πράκτορας είναι μυωπικός, καθώς ενδιαφέρεται να μεγιστοποιήσει μόνο το reward που λαμβάνει άμεσα, δηλαδή το r_{t+1} . Όταν το $\gamma < 1$, το G_t είναι πεπερασμένο εφόσον η ακολουθία των r_t είναι φραγμένη. Από την άλλη, όσο το γ προσεγγίζει το 1, τόσο περισσότερο ο πράκτορας θα λαμβάνει υπόψιν τα μελλοντικά rewards. Επίσης, το G_t συχνά προτιμάται και για περιπτώσεις finite horizon, δηλαδή με τερματικά states, καθώς το γ για μελλοντικά rewards θα είναι όλο και μικρότερο. Με αυτόν τον τρόπο, δίνεται μεγαλύτερη σημασία στα πιο άμεσα rewards, χωρίς ωστόσο να μην λαμβάνονται υπόψιν και τα επόμενα. Έτσι, διαισθητικά, το γ μπορεί να μοντελοποιήσει το γεγονός πως ο πράκτορας δεν γνωρίζει πότε το επεισόδιο θα τερματιστεί μέσω του τερματικού state.

2.4.2 Μαρκοβιανή Διαδικασία Λήψης Αποφάσεων (MDP)

Ένα περιβάλλον Ενισχυτικής Μάθησης μπορεί να περιγραφεί ως μια Μαρκοβιανή Διαδικασία Λήψης Αποφάσεων (**Markov Decision Process** ή **MDP**). Τα MDPs στηρίζονται στην Μαρκοβιανή Ιδιότητα, η οποία στηρίζεται στο ότι "το μέλλον είναι ανεξάρτητο του παρελθόντος δεδομένου του παρόντος". Τυπικά, θεωρούμε

$$P[S_{t+1} | S_1, \dots, S_t] = P[S_{t+1} | S_t] \quad (2.37)$$

Με αυτόν τον τρόπο, μπορούμε να ορίσουμε τις πιθανότητες μετάβασης (**transition probabilities** ή **state-transition probabilities**) μεταξύ ενός state σε ένα άλλο. Μάλιστα, δεδομένου πως από ένα state S_t ο πράκτορας μπορεί να μεταβεί στο S_{t+1} μέσω ενός action A_t , η αντίστοιχη πιθανότητα μετάβασης είναι η $P[S_{t+1} | S_t, A_t]$. Με το P συμβολίζουμε πιθανότητα.

Σε ένα πεπερασμένο MDP, το σύνολο των states S , των actions A και των rewards R

είναι όλα πεπερασμένα. Στην περίπτωση αυτή, οι τυχαίες μεταβλητές r και s έχουν διακριτές κατανομές πιθανότητας εξαρτώμενες από το προηγούμενο state και action. Έτσι, ορίζεται η δυναμική (*dynamics*) του MDP με την συνάρτηση πιθανότητας

$$p(s', r | s, a) = P[S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a] \quad (2.38)$$

για όλα τα $s', s \in S, r \in R$ και $a \in A$. Έτσι, οι πιθανότητες μετάβασης ισούνται με

$$P[S_{t+1} | S_t, A_t] = p(s' | s, a) = \sum_{r \in R} p(s', r | s, a) \quad (2.39)$$

2.4.2.1 Στρατηγικές και Συναρτήσεις Αξίας

Όλοι σχεδόν οι αλγόριθμοι Ενισχυτικής Μάθησης κάνουν λόγο για συναρτήσεις αξίας (**value functions**). Πρόκειται για συναρτήσεις του state ή του ζευγαριού state-action. Οι συναρτήσεις αυτές εκτιμούν το πόσο καλό είναι για τον πράκτορα να βρίσκονται στο δεδομένο state ή πόσο καλό είναι να βρίσκονται στο δεδομένο state και να επιλέγουν το δεδομένο action. Σημειώνουμε πως τα rewards, που θα λαμβάνει μελλοντικά ο πράκτορας, εξαρτώνται από τις επιλογές των actions του. Έτσι, χρειάζεται να ορίσουμε τις στρατηγικές (*policies*) με τις οποίες ο πράκτορας επιλέγει τα actions. Τυπικά, ένα **policy** π είναι μια αντιστοίχιση από τα states προς τα διαθέσιμα actions [7]. Εάν ο πράκτορας ακολουθεί ένα policy π την χρονική στιγμή t , τότε το $\pi(a|s)$ είναι η πιθανότητα $A_t = a$ εάν $S_t = s$.

Η value function ενός state s , χρησιμοποιώντας το policy π , $u_\pi(s)$, είναι η αναμενόμενη ανταμοιβή δεδομένου πως ο πράκτορας βρίσκεται στο s και ακολουθεί το π από εδώ και στο εξής. Τυπικά, η state-value function ορίζεται ως εξής

$$u_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \quad (2.40)$$

Από την άλλη, η action-value function ορίζεται ως εξής

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \quad (2.41)$$

Για πεπερασμένα MDPs μπορούμε να ορίσουμε με ακρίβεια ένα βέλτιστο policy θεωρώντας πως οι value functions έχουν μια μερική διάταξη πάνω στα policies. Ένα policy π είναι καλύτερο από ένα άλλο π' , εάν η αναμενόμενη ανταμοιβή του είναι μεγαλύτερη για όλα τα $s \in S$. Ορίζουμε το βέλτιστο policy με π_* και αντιστοίχως τα u_* και q_* . Τυπικά γράφουμε

$$u_*(s) = \max_{\pi} u_\pi(s) \quad (2.42)$$

και

$$q_*(s, a) = \max_{\pi} q_\pi(s, a) \quad (2.43)$$

για όλα τα $s \in S$ και τα $a \in A$, αντίστοιχα.

Ορίζουμε τις εξισώσεις **Bellman** βελτιστότητας, λαμβάνοντας ότι η αξία ενός state με

βέλτιστο policy πρέπει να ισούται με την αναμενόμενη ανταμοιβή για το καλύτερο διαθέσιμο action του state αυτού.

$$u_*(s) = \max_{a \in A(s)} q_{\pi_*}(s, a) = \max_a \sum_{s', r} p(s', r | s, a) [r + \gamma u_*(s')] \quad (2.44)$$

και

$$\begin{aligned} q_*(s, a) &= \mathbb{E} \left[r_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a \right] \\ &= \sum_{s', r} p(s', r | s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right] \end{aligned} \quad (2.45)$$

Σε περιπτώσεις πεπερασμένων MDPs, όταν είναι γνωστά τα δυναμικά του περιβάλλοντος, η Ενισχυτική Μάθηση μπορεί να επιτευχθεί με αλγόριθμους **δυναμικού προγραμματισμού**. Οι αλγόριθμοι δυναμικού προγραμματισμού περιλαμβάνουν τους Policy Evaluation, Value Iteration [7] κá. Σε αντίθετες περιπτώσεις, χρησιμοποιούνται μέθοδοι που περιλαμβάνουν *Temporal Difference* ή *Monte Carlo* αλγόριθμους.

2.4.2.2 Κατηγοριοποίηση Στρατηγικών

Οι μεθοδολογίες για τις στρατηγικές (policies) χωρίζονται σε δύο κατηγορίες: τις **on-policy** και τις **off-policy**. Οι on-policy μέθοδοι προσπαθούν να αξιολογούν και να βελτιώνουν ένα υπάρχον policy, με το οποίο ο πράκτορας λαμβάνει αποφάσεις. Αντίθετα, οι off-policy μέθοδοι αξιολογούν και βελτιώνουν κάποιο policy διαφορετικό από εκείνο που χρησιμοποιείται για να παράξει τα δεδομένα [7]. Στις off-policy μεθόδους, το policy που θέλουμε να βελτιστοποιήσουμε λέγεται target, ενώ εκείνο με το οποίο παράγεται η εμπειρία λέγεται behavior.

2.4.3 Εξερεύνηση vs Εκμετάλλευση

Στην Ενισχυτική Μάθηση, συχνά θίγεται η συζήτησης περί εξερεύνησης και εκμετάλλευσης. Προκειμένου να λάβουμε μεγάλο συσσωρευτικό reward, ο πράκτορας πρέπει να επιλέγει τα καλύτερα δυνατά actions. Ωστόσο, για να μάθει ποια είναι τα καλύτερα actions, πρέπει πρώτα να έχει μάθει να τα αναγνωρίζει. Έτσι, για να μάθει κάτι τέτοιο, είναι αναγκαίο να επιλέγει και actions που δεν έχει ξαναεπιλέξει. Κατά αυτόν τον τρόπο, ο πράκτορας μπορεί να εκμεταλλεύεται τα καλύτερα του actions αλλά και να εξερευνεί νέα, τα οποία ενδέχεται να συμβάλουν μακροπρόθεσμα στο να φθάσει σε βέλτιστο policy. Αυτό είναι και το λεγόμενο αντιστάθμισμα (**trade-off**) ανάμεσα στην εξερεύνηση και την εκμετάλλευση. Έτσι, ο πράκτορας χρειάζεται να βρίσκει μια ισορροπία ανάμεσα σε αυτά τα δύο, ώστε να αντιμετωπίζει πιο αποτελεσματικά το εκάστοτε πρόβλημα.

2.4.4 Μέθοδοι Monte Carlo

Οι μέθοδοι Monte Carlo χρησιμοποιούνται, όταν έχουμε ως δεδομένα δείγματα επεισοδίων, χωρίς a priori γνώση του περιβάλλοντος. Σκοπός του πράκτορα φυσικά είναι να βρει ένα βέλτιστο policy. Έτσι ένα μοντέλο Monte Carlo χρειάζεται απλά να παράγει μετα-

βάσεις του πράκτορα και όχι ολόκληρες τις κατανομές πιθανοτήτων, που απαιτούνται στους αλγόριθμους δυναμικού προγραμματισμού.

Η ιδέα των μεθόδων Monte Carlo είναι πως ο πράκτορας κάθε φορά παράγει την εμπειρία με τις μεταβάσεις των states για ένα ολόκληρο επεισόδιο - ή μέχρι μία καθορισμένη τερματική κατάσταση. Με αυτόν τον τρόπο, εκπαιδεύεται επαναληπτικά για κάθε μετάβασή του, χρησιμοποιώντας την *πραγματική ανταμοιβή* που έλαβε κατά τη διάρκεια του επεισοδίου αυτού. Ένα βασικό πλεονέκτημα των μεθόδων αυτών είναι πως οι συναρτήσεις αξίας για κάθε state είναι ανεξάρτητες, υπό την έννοια ότι η εκτίμηση αυτή δεν βασίζεται σε κάποια προηγούμενως υπολογισμένη εκτίμηση κάποιου άλλου state. Με άλλα λόγια, οι Monte Carlo μέθοδοι δεν κάνουν bootstrapping [7].

2.4.5 Μέθοδοι Temporal Difference

Το Temporal Difference συνδυάζει τη δειγματοληψία (sampling) των μεθόδων Monte Carlo - και το bootstrapping - του δυναμικού προγραμματισμού. Η ανανέωση του value function $V(s)$ γίνεται με τον εξής τρόπο

$$V(S_t) = V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \quad (2.46)$$

όπου α είναι το step size της μάθησης. Η διαφορά $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ λέγεται Temporal Difference Error (*TD Error*).

Υπάρχουν δύο βασικοί αλγόριθμοι που βασίζονται στο Temporal Difference: ο Sarsa και ο Q-learning. Ο Sarsa είναι on-policy, ενώ ο Q-learning είναι off-policy. Δεδομένου πως στην μελέτη μας, θα ασχοληθούμε με αλγόριθμους βασισμένους στο Q-learning, θεωρούμε πως άλλοι αλγόριθμοι Temporal Difference είναι εκτός του ενδιαφέροντος της παρούσης δουλειάς και παραπέμπουμε τον αναγνώστη στη σχετική βιβλιογραφία.

2.4.5.1 Q-learning

Ο αλγόριθμος του Q-learning ανανεώνει την value function $Q(S,A)$ ως εξής

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \quad (2.47)$$

Η Q λέγεται και Q-function, ενώ οι τιμές της ονομάζονται Q-values. Παρακάτω, παραθέτουμε τον αλγόριθμο του Q-learning. Αυτός χρησιμοποιεί ως target policy την άπληστη επιλογή και ως behavior policy κάποια, η οποία λαμβάνει υπόψιν το trade-off της εξερεύνησης-εκμετάλλευσης. Ένα τέτοιο policy μπορεί να είναι το ϵ -greedy, για το οποίο θα συζητήσουμε και στο πειραματικό μέρος της εργασίας μας. Για σύγκλιση, δηλαδή για να φθάσουμε σε βέλτιστο policy, αποτελεί απαίτηση το γεγονός πως για όλα τα ζευγάρια S,A πρέπει να συμβαίνει τακτικά η ανανέωση της $Q(S,A)$. Παράλληλα, χρειάζεται να μειώνεται με κάποιο ρυθμό και το step size α , που έχει παρόμοια σημασία με το learning rate στα τεχνητά νευρωνικά δίκτυα.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:
Initialize S
Loop for each step of episode:
Choose A from S using policy derived from Q (e.g., ε -greedy)
Take action A , observe R, S'
 $Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$
 $S \leftarrow S'$
until S is terminal

Εικόνα 2.3: Ο Αλγόριθμος Q-learning. Πηγή [7]

2.4.5.2 Double Q-learning

Το πρόβλημα που προκύπτει με το απλό Q-learning - όπως και με το Sarsa - είναι πως συχνά συμβαίνει υπερεκτίμηση των εκτιμώμενων τιμών που επιλέγονται από τον τελεστή \max κατά την ενημέρωση της Q . Έτσι, το μοντέλο οδηγείται σε σημαντικά μεγάλη θετική προκατάληψη (bias). Διαισθητικά, ένας τρόπος να το δούμε είναι πως το μοντέλο που επιλέγει το καλύτερο ζευγάρι είναι και εκείνο που υπολογίζει τις αντίστοιχες τιμές της Q για όλα τα ζευγάρια.

Για αυτόν τον λόγο, προτείνεται η χρήση δύο μοντέλων, Q_1 και Q_2 - ίδιας αρχιτεκτονικής - ώστε το ένα να επιλέγει το καλύτερο ζευγάρι (S, A) - κατά τον τελεστή του \max - και το άλλο να εκτιμά τα Q-values. Έτσι, η ενημέρωση γίνεται ως εξής

$$Q_1(S_t, A_t) = Q_1(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q_2 \left(S_{t+1}, \underset{a}{\operatorname{argmax}} Q_1(S_{t+1}, a) \right) - Q_1(S_t, A_t) \right] \quad (2.48)$$

Μάλιστα, προτείνεται τα δύο μοντέλα να εναλλάσσουν τη λειτουργία τους με τυχαίο τρόπο - πιθανότητα ίση με 0.5 - κατά την ενημέρωση. Παρακάτω, απεικονίζουμε τον αλγόριθμο του Double Q-learning.

Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:
Initialize S
Loop for each step of episode:
Choose A from S using the policy ε -greedy in $Q_1 + Q_2$
Take action A , observe R, S'
With 0.5 probability:
 $Q_1(S, A) \leftarrow Q_1(S, A) + \alpha (R + \gamma Q_2(S', \underset{a}{\operatorname{argmax}} Q_1(S', a)) - Q_1(S, A))$
else:
 $Q_2(S, A) \leftarrow Q_2(S, A) + \alpha (R + \gamma Q_1(S', \underset{a}{\operatorname{argmax}} Q_2(S', a)) - Q_2(S, A))$
 $S \leftarrow S'$
until S is terminal

Εικόνα 2.4: Ο Αλγόριθμος Double Q-learning. Πηγή [7]

2.4.5.3 Deep Q-learning

Μέχρι στιγμής έχουμε δει περιπτώσεις στις οποίες ο αριθμός των states ή/και των actions ήταν πεπερασμένος. Αυτό σημαίνει πως μπορούμε όλες τις εκτιμήσεις της value function να τις αποθηκεύουμε σε κάποια δομή, πχ. πίνακα (tabular methods). Επίσης, έχουμε αναφέρει πως σε τέτοιες περιπτώσεις επέρχεται σύγκλιση, και άρα το βέλτιστο policy, κάτω από ορισμένες συνθήκες.

Ωστόσο, τι γίνεται στην περίπτωση που δεν μπορούμε να σώσουμε τις εκτιμήσεις σε πίνακα στην περίπτωση που ο αριθμός των states ή/και των actions είναι πολύ μεγάλος ή/και μη πεπερασμένος - άρα έχουμε μη πεπερασμένο (infinite) MDP; Την απάντηση σε αυτό το ερώτημα δίνουν οι μέθοδοι προσεγγιστικών συναρτήσεων (function approximation methods). Στην εποχή της Βαθιάς Μάθησης, η πιο συχνή προσεγγιστική συνάρτηση είναι ένα νευρωνικό δίκτυο. Όταν η Ενισχυτική Μάθηση συνδυάζεται με την Βαθιά Μάθηση, τότε συχνά χρησιμοποιείται ο όρος της Βαθιάς Ενισχυτικής Μάθησης (Deep Reinforcement Learning ή Deep RL).

Έτσι, ο Mnih και οι συνεργάτες του [8] πρότειναν το **Deep Q-learning** μέσα από τη χρήση δύο νευρωνικών δικτύων που λέγονται **Deep Q-Networks (DQN)**. Ο αλγόριθμος αυτός αναπαρίσταται παρακάτω.

Algorithm 1: deep Q-learning with experience replay.

```

Initialize replay memory  $D$  to capacity  $N$ 
Initialize action-value function  $Q$  with random weights  $\theta$ 
Initialize target action-value function  $\hat{Q}$  with weights  $\theta^- = \theta$ 
For episode = 1,  $M$  do
  Initialize sequence  $s_1 = \{x_1\}$  and preprocessed sequence  $\phi_1 = \phi(s_1)$ 
  For  $t = 1, T$  do
    With probability  $\epsilon$  select a random action  $a_t$ 
    otherwise select  $a_t = \operatorname{argmax}_a Q(\phi(s_t), a; \theta)$ 
    Execute action  $a_t$  in emulator and observe reward  $r_t$  and image  $x_{t+1}$ 
    Set  $s_{t+1} = s_t, a_t, x_{t+1}$  and preprocess  $\phi_{t+1} = \phi(s_{t+1})$ 
    Store transition  $(\phi_t, a_t, r_t, \phi_{t+1})$  in  $D$ 
    Sample random minibatch of transitions  $(\phi_j, a_j, r_j, \phi_{j+1})$  from  $D$ 
    Set  $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^-) & \text{otherwise} \end{cases}$ 
    Perform a gradient descent step on  $(y_j - Q(\phi_j, a_j; \theta))^2$  with respect to the network parameters  $\theta$ 
    Every  $C$  steps reset  $\hat{Q} = Q$ 
  End For
End For

```

Εικόνα 2.5: Ο Αλγόριθμος Deep Q-learning. Πηγή [8]

Σημειώνουμε τη χρήση μιας δομής που αποθηκεύει τις εμπειρίες που λαμβάνει ο πράκτορας. Η δομή αυτή λέγεται *Experience Replay* και αποθηκεύει εγγραφές της μορφής

$$\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle$$

Στον παραπάνω αλγόριθμο, το S_t περνάει από μια μη γραμμική συνάρτηση ϕ , οπότε οι εγγραφές είναι της μορφής

$$\langle \phi_t, A_t, R_{t+1}, \phi_{t+1} \rangle$$

Ακόμη τονίζουμε τη χρήση των δύο Q-Networks, από τα οποία, το ένα λειτουργεί ως στόχος κατά το TD Error. Αυτό το νευρωνικό δίκτυο - στόχος - λέγεται Target Q-Network. Ο αλγόριθμος μετά από έναν προκαθορισμένο αριθμό χρονικών βημάτων ανανεώνει επαναληπτικά τις παραμέτρους $\bar{\theta}$ του Target Q-Network με τις αντίστοιχες (θ) του άλλου.

Η χρήση του Target Q-Network είναι καθοριστικής σημασίας για το DQN. Με το Deep Q-learning, έχουμε αντικαταστήσει τις πραγματικές τιμές της value function με μία προσεγγιστική συνάρτηση. Ωστόσο, ενημερώνουμε τις τιμές για κάθε action, χωρίς να έχουμε βεβαίωση ότι φθάνουμε σε σύγκλιση ή ότι η μάθηση είναι σταθερή. Συχνά, στο Deep Q-learning βλέπουμε ο πράκτορας να αρχίζει να μαθαίνει καλά το πρόβλημα, παίρνοντας σωστές αποφάσεις. Ωστόσο μετά από κάποιο σημείο παρατηρούμε τρομέρη μείωση των σωστών αποφάσεων, καθώς το step size ελαττώνεται. Αυτό το φαινόμενο λέγεται καταστροφική λήθη (**catastrophic forgetting**). Έτσι, η χρήση του Target Q-Network εξασφαλίζει την σταθερή μάθηση. Διαισθητικά, ο πράκτορας κάθε φορά στοχεύει να μάθει έναν στόχο, που αντιστοιχεί σε ένα βάθος, που εκφράζει το πόσο καλά αυτός έχει μάθει το πρόβλημα. Όταν φθάσει σε αυτό το βάθος της μάθησης, τότε αλλάζει το στόχο, ώστε να εμβαθύνει ακόμα περισσότερο στο πρόβλημα. Με άλλα λόγια, "ο πράκτορας προσπαθεί να κυνηγήσει έναν κινούμενο στόχο", ώστε με κάθε ανανέωση των παραμέτρων του στόχου να σκοπεύουμε σε ένα πιο σταθερό μοντέλο.

2.4.5.4 Double Deep Q-learning

Το Deep Q-learning, ωστόσο, έχει το ίδιο πρόβλημα υπερεκτίμησης των τιμών που είδαμε και στο Q-learning. Για το λόγο αυτό, ο van Hasselt και άλλοι [33] πρότειναν τον αλγόριθμο του **Double Deep Q-learning** χρησιμοποιώντας και πάλι δύο Q-Networks. Τα νευρωνικά δίκτυα αυτά λέγονται **Double Deep Q-Network (DDQN)**. Ομοίως με το Double Q-learning, ο στόχος κατά την ανανέωση της Q γράφεται ως εξής

$$y = r_{t+1} + \gamma Q \left(\phi_{t+1}, \underset{a}{\operatorname{argmax}} Q(\phi_{t+1}, a; \bar{\theta}) ; \bar{\theta} \right) \quad (2.49)$$

Κατά τα άλλα, ο αλγόριθμος του Deep Q-learning παραμένει ως έχει.

2.5 Διαχείριση Κειμένων και Επεξεργασία Φυσικής Γλώσσας

Στην ενότητα αυτή θα συζητήσουμε σχετικά με τη διαχείριση των κειμένων, μέσα από μεθόδους της Επεξεργασίας Φυσικής Γλώσσας (**Natural Language Processing** ή **NLP**). Στο NLP, είναι απαραίτητο να μετατρέπουμε ένα κείμενο σε μια μορφή, η οποία θα μπορεί να χρησιμοποιηθεί άμεσα από αλγορίθμους Μηχανικής Μάθησης.

Στην Υποενότητα 2.5.1 θα περιγράψουμε μεθόδους με τις οποίες προεπεξεργαζόμαστε ένα κείμενο, στην Υποενότητα 2.5.2 θα συζητήσουμε για το μοντέλο ανάκτησης πληροφορίας με λογική Μπουλ, ενώ στην Υποενότητα 2.5.3 θα μιλήσουμε για αναπαραστάσεις των λέξεων

- ή των κειμένων - με τη μορφή διανύσματων.

2.5.1 Προεπεξεργασία Κειμένου

Προκειμένου ένα κείμενο να χρησιμοποιηθεί από κάποιον αλγόριθμο, συνήθως χρειάζεται να υποστεί μια αρχική προεπεξεργασία (**Text Preprocessing**). Αρχικά, χρειάζεται να μετασχηματίσουμε το κείμενο σε κομμάτια (**tokenization**), ώστε κάθε κομμάτι (token) να αποτελεί μια λέξη ή κάποιο σύμβολο. Στην συνέχεια, συνήθως πρέπει να αφαιρέσουμε τα tokens που περιέχουν σύμβολα, τα οποία δεν προσδίδουν άμεση πληροφορία σχετικά με το σημασιολογικό περιεχόμενο του κειμένου. Τέτοια σύμβολα μπορεί να είναι σημεία στίξης, αριθμοί, μαθηματικά σύμβολα κ.ά. Βεβαίως τα σύμβολα που αφαιρούμε μπορούν να ποικίλλουν, αναλόγα με τις απαιτήσεις του εκάστοτε προβλήματος. Μία άλλη σημαντική ενέργεια είναι η απαλοιφή των tokens που περιέχουν κάποια λέξη η οποία εμφανίζεται αρκετά συχνά στα κείμενα και δεν προσδίδει σημασιολογική πληροφορία. Τέτοιες λέξεις (**stopwords**) επιλέγονται ως υπερπαραμέτροι από τον ίδιο τον προγραμματιστή και μπορεί να είναι άρθρα, αντωνυμίες, προσδιορισμοί, ρήματα κατάστασης κ.ά. Η αφαίρεση των λέξεων αυτών είναι συχνή σε εφαρμογές, όπου δεν απαιτείται η πλήρης αλληλουχία των λέξεων του κειμένου.

Επιπλέον, σε αρκετές περιπτώσεις συνήθίζεται να επιλέγονται δύο εναλλακτικές διαδικασίες: το **stemming** και το **lemmatization**. Το stemming είναι η διαδικασία με την οποία οι λέξεις αντικαθίστανται από τις ρίζες τους. Με τον τρόπο αυτό, απαλοψουμε χαρακτήρες και καταλήξεις λέξεων, που δεν προσφέρουν παραπάνω σημασιολογικό περιεχόμενο από την ρίζα της λέξης. Από την άλλη, το lemmatization είναι η διαδικασία με την οποία ομαδοποιούμε μαζί τα παράγωγα μίας λέξης, ώστε αυτά να αποτελούν το ίδιο token. Με αυτόν τον τρόπο, αντιστοιχίζονται όλες οι μορφές μιας λέξης με το λήμμα της δηλαδή με τη μορφή με την οποία θα τη βρήκαμε σε ένα λεξικό. Ωστόσο, για να γίνει αυτή η διαδικασία, χρειάζεται εκ των προτέρων να είναι γνωστό το λεξικό με τις λέξεις των κειμένων.

2.5.2 Ανάκτηση Πληροφορίας με το Μοντέλο Λογικής Μπουλ

Στην Ανάκτηση Πληροφορίας, ανέκαθεν, σκοπός είναι με χρήση ενός ερωτήματος (query) να εξάγονται κείμενα από μια συλλογή δεδομένων. Τα κείμενα αυτά πρέπει να ικανοποιούν ορισμένες ιδιότητες που περιέχονται μέσα στο query.

Το μοντέλο λογικής Μπουλ (**Boolean Model** [34]) είναι το πρώτο μοντέλο, που αναπτύχθηκε ποτέ, για Ανάκτηση Πληροφορίας. Βασίζεται στην λογική Μπουλ και την κλασική θεωρία συνόλων, κατά τρόπο, ώστε τόσο το κείμενο όσο και το query να αναπαρίστανται ως σύνολα όρων. Ένας όρος μπορεί να είναι μία λέξη ή μια φράση. Ένα κείμενο μπορεί να αναπαρίσταται ως σύνολο όρων, χωρίς ωστόσο κατά ανάγκη κάθε λέξη του κειμένου να περιέχεται σε κάποιο όρο. Η ανάκτηση των κειμένων συμβαίνει, εάν τα κείμενα αυτά περιέχουν όλους τους όρους που υπάρχουν στην αναπαράσταση του query. Έτσι, για κάθε query μπορεί να ανακτώνται ακόμα και περισσότερα από ένα κείμενα της συλλογής. Δεδομένου πως δεν θα χρησιμοποιήσουμε στη μελέτη μας το Boolean Model, παραπέμπουμε τον αναγνώστη για περισσότερες πληροφορίες στο [34].

2.5.3 Διανυσματικές Αναπαραστάσεις Λέξεων και Κειμένων

Στην υποενότητα αυτή, θα μιλήσουμε για αναπαραστάσεις των λέξεων ενός κειμένου σε μορφή διανύσματος σταθερού μεγέθους. Ανέκαθεν, οι αναπαραστάσεις αυτές έχουν βοηθήσει σημαντικά στην επίλυση πολλών διαφορετικών προβλημάτων της Επεξεργασίας Φυσικής Γλώσσας. Με τον όρο διάνυσμα λέξεως αναφερόμαστε στην αντικατάσταση κάθε λέξης του λεξικού των κειμένων με ένα διάνυσμα σταθερής διάστασης. Σκοπός μίας τέτοιας μεθοδολογίας είναι η κάθε λέξη να αναπαρίσταται στο χώρο και, άρα, να ορίζεται σε έναν βαθμό η ομοιότητα ή διαφορετικότητα των λέξεων, μέσα από τα διανύσματα αναπαράστασης. Έτσι, μπορούμε να μετράμε την ομοιότητα δύο λέξεων μέσα από μαθηματικές μετρικές, όπως το εσωτερικό γινόμενο και οι ομοιότητες συνημιτόνου και Jaccard, ή η Ευκλείδεια απόσταση.

Από εδώ και στο εξής, θα αναφερόμαστε στις διανυσματικές αναπαραστάσεις των λέξεων με τον ορό διανύσματα λέξεων (**word embeddings**). Δεδομένου πως κάθε λέξη θα αντιστοιχίζεται σε ένα word embedding, δηλαδή σε ένα διάνυσμα σταθερής διάστασης, ιδανικά θα θέλαμε η κάθε διάσταση να περιγράφει διαφορετική σημασιολογική ή/και συντακτική ιδιότητα από τις υπόλοιπες. Επιπλέον, κάθε κείμενο μπορεί να αναπαρίσταται ως ένας πίνακας (matrix) με γραμμές τα word embeddings των αντίστοιχων λέξεών του. Έτσι, εάν υποθέσουμε πως η διάσταση των word embeddings είναι D και το κείμενο περιέχει N λέξεις, τότε το κείμενο μπορεί να έχει πίνακα αναπαράστασης μεγέθους $N \times D$.

Αξίζει να αναφέρουμε το μοντέλο διανυσματικού χώρου (**Vector Space Model [35]**). Σε αντίθεση με το Boolean Model, με το μοντέλο αυτό, μπορούμε αλγεβρικά να αναπαριστούμε κείμενα ως διανύσματα, τα οποία αποτελούνται από αναγνωριστικά χαρακτηριστικά, όπως δεικτοδότησης ή/και σημασιολογίας.

Μέχρι το τέλος της Ενότητας αυτής, θα συζητήσουμε για διαφορετικές μεθοδολογίες και αλγόριθμους που αντικαθιστούν τις λέξεις ενός κειμένου, είτε το ίδιο το κείμενο, με διανύσματα. Επιγραμματικά θα αναφερθούμε στα εξής: Δείκτες του λεξικού (2.5.2.1), One-Hot Vectors (2.5.2.2), Διανύσματα με χρήση tf-idf, word embeddings με τον επαναληπτικό αλγόριθμο word2vec (2.5.2.3) και word embeddings με τον επαναληπτικό αλγόριθμο Glove (2.5.2.4). Σημειώνουμε πως οι μέθοδοι One-Hot Vectors, tf-idf, word2vec και Glove χρησιμοποιούν ως βάση το Vector Space Model.

2.5.3.1 Δείκτες του λεξικού

Μία αναποτελεσματική μέθοδος για την αντικατάσταση των λέξεων με διανύσματα είναι η χρήση δεικτών με τις αντίστοιχες εγγραφές τους στο λεξικό. Μία τέτοια ενέργεια δεν παρέχει σε κανένα βαθμό σημασιολογική ή/και συντακτική πληροφορία. Επίσης, δεδομένου πως κάθε λέξη είναι ένα βαθμωτό μέγεθος - και πιο συγκεκριμένα ένας αύξοντας αριθμός - οι λέξεις μεταξύ τους επιδέχονται διάταξη. Ωστόσο, αυτή η διατάξη δεν ανταποκρίνεται προφανώς στην πραγματική συσχέτιση των λέξεων και προκαλεί θόρυβο. Όσον αφορά το ολόκληρο κείμενο, αυτό αναπαριστάται ως ένα διάνυσμα με χαρακτηριστικά τους δείκτες των αντίστοιχων λέξεών του.

2.5.3.2 One-Hot Vectors

Μία εναλλακτική, αλλά και πάλι, αρκετά απλή μεθοδολογία είναι η χρήση διακεκριμένων διανυσμάτων για κάθε λέξη του λεξικού, με την χρήση **One-Hot Vectors**. Τα διανύσματα αυτά είναι one-hot, υπό την έννοια πως υπάρχουν τόσες στάθμες - διαστάσεις - όσες είναι οι διαφορετικές λέξεις του λεξικού. Κάθε στάθμη είναι ενεργή - ίση με 1 - μόνο για την λέξη που της αντιστοιχεί. Με αυτόν τον τρόπο, κάθε λέξη αντιστοιχεί σε ένα διάνυσμα, το οποίο σε όλες τις διαστάσεις του, εκτός από μία, είναι μηδενικό. Όσες περισσότερες λέξεις αποτελούν το λεξικό, τόσες θα είναι και οι διαστάσεις ενός διανύσματος. Έτσι, δημιουργούμε διανύσματα πολύ μεγάλης διάστασης, αλλά και πάρα πολύ αραιά, χωρίς, ωστόσο, αυτά να παρέχουν πληροφορίες για το σημασιολογικό και συντακτικό περιεχόμενο των λέξεων. Από την άλλη, είναι χρήσιμα σε περιπτώσεις, όπως η αναπαράσταση ολόκληρου του κειμένου. Κάθε κείμενο αναπαρίσταται ως ένας πίνακας, του οποίου οι γραμμές είναι οι one-hot αναπαραστάσεις των λέξεων του. Τονίζουμε πως οι γραμμές του πίνακα τηρούν τη σειρά των λέξεων, όπως αυτές εμφανίζονται στο κείμενο.

2.5.3.3 Διάνυσμα κειμένου με χρήση tfidf

Μία στατιστική μέθοδος αναπαράστασης των λέξεων σε διανύσματα είναι μέσω της χρήσης του term frequency-inverse document frequency (**tfidf**). Το tfidf είναι μια στατιστική μετρική, που σκοπεύει να αναδείξει την σημαντικότητα μιας λέξης δεδομένου ενός συνόλου κειμένων.

Αρχικά, δημιουργούμε το λεξικό των λέξεων, που υπάρχουν σε όλα τα κείμενα του συνόλου. Θα αναπαραστήσουμε κάθε λέξη με έναν πραγματικό αριθμό, όπως και στην περίπτωση με τους δείκτες. Ωστόσο, στην περίπτωση αυτή, κάθε κείμενο αντιστοιχίζεται με ένα διάνυσμα, το οποίο έχει διάσταση ίση με το μέγεθος του λεξικού.

Για κάθε κείμενο i του συνόλου εργαζόμαστε με τον ακόλουθο τρόπο. Βρίσκουμε για κάθε λέξη του j το term frequency (tf), το οποίο ορίζεται ως εξής

$$tf_{i,j} = \frac{n_{i,j}}{\sum_{j=1}^{N_W} n_{i,j}} \quad (2.50)$$

όπου το N_W είναι το πλήθος των διαφορετικών λέξεων του κειμένου j και το $n_{i,j}$ είναι το πλήθος των φορών που εμφανίζεται η λέξη i στο j .

Στη συνέχεια, για κάθε λέξη i του λεξικού βρίσκουμε το inverse document frequency (idf) που ορίζεται ως εξής

$$idf(i) = \log \frac{N}{n} \quad (2.51)$$

όπου το N είναι το πλήθος όλων των κειμένων του συνόλου και το n είναι το πλήθος των κειμένων που περιέχουν τη λέξη i .

Έτσι, για κάθε λέξη ενός κειμένου, προκύπτει η τιμή του $tfidf = tf \cdot idf$. Με αυτόν τον τρόπο, κάθε κείμενο αναπαρίσταται ως διάνυσμα μεγέθους W , όπου το W είναι το πλήθος των διαφορετικών λέξεων στο σύνολο κειμένων. Σημειώνουμε πως ούτε με το $tfidf$ το κείμενο αναπαρίσταται σημασιολογικά ή συντακτικά. Από την άλλη, επιτυγχάνεται μια

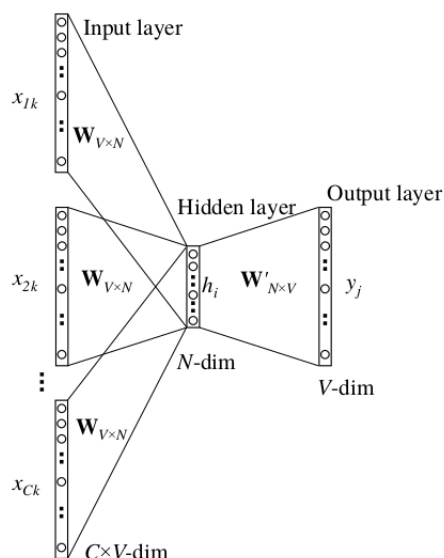
στατιστική ένδειξη της σημασίας των λέξεων πάνω στα δεδομένα κείμενα. Με αυτόν τον τρόπο, κάποιες λέξεις κρίνονται σημαντικές σε σχέση με κάποιες άλλες. Αναφέρουμε πως η χρήση του tfidf συνήθως καθιστά αναγκαία την προεπεξεργασία του κειμένου, που περιγράψαμε παραπάνω. Ειδικότερα, τις περισσότερες φορές απαιτείται το stemming/lemmatization, αφού οι αναπαραστάσεις δεν παρέχουν σημασιολογική ερμηνεία των λέξεων, με αποτέλεσμα παρόμοιες σημασιολογικά λέξεις τελικά να διαφέρουν σημαντικά.

2.5.3.4 Το word2vec μοντέλο

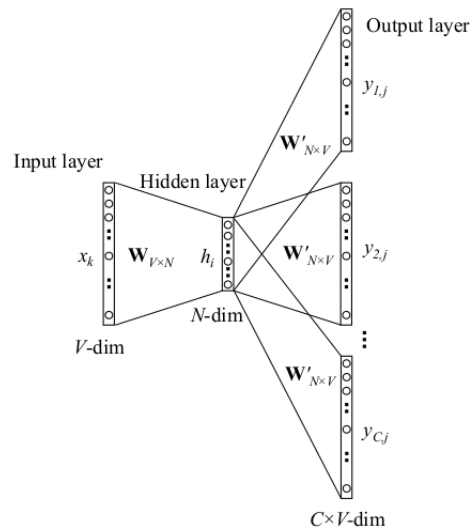
Στο σημείο αυτό, θα μιλήσουμε για έναν επαναληπτικό αλγόριθμο που στοχεύει στην δημιουργία word embeddings, τα οποία παρέχουν σημασιολογική και συντακτική πληροφορία των λέξεων. Ο αλγόριθμος **word2vec** προσπαθεί, με επαναληπτικό τρόπο, να εντοπίσει τις συσχετίσεις μεταξύ των λέξεων ενός κειμένου. Υπολογίζει την πιθανότητα εμφάνιση μιας λέξης, δεδομένου του περιεχομένου (*context*) ενός κυλιόμενου παραθύρου στο οποίο βρίσκεται κάθε φορά. Στην επιστημονική βιβλιογραφία, έχουν παρατηρηθεί αρκετές προσπάθειες [36, 37], με εκείνη του Mikolov και άλλων [38] να έχει εδραιωθεί ευρέως στο χώρο της Επεξεργασίας Φυσικής Γλώσσας.

Στο word2vec προτείνουν δύο βασικούς αλγόριθμους: το **Continuous Bag-of-words (CBOW)** και το **skip-gram**. Οι δύο αυτοί αλγόριθμοι στοχεύουν στον ίδιο σκοπό, αλλά με εναλλακτικό τρόπο. Στο CBOW, Σχήμα 2.10, είναι δεδομένες οι context λέξεις και το μοντέλο προσπαθεί να προβλέψει την κεντρική. Αντίθετα, στο skip-gram, Σχήμα 2.11 είναι δεδομένη η κεντρική λέξη και το μοντέλο προσπαθεί να προβλέψει τις υπόλοιπες λέξεις του context.

Επιπλέον, έχουν προταθεί δύο μεθοδολογίες βελτιστοποίησης πόρων: μέσω *hierarchical softmax* και μέσω *negative sampling*. Στην πρώτη μέθοδο, χρησιμοποιείται ένα δέντρο, προκειμένου να υπολογίζονται αποδοτικά οι πιθανότητες των λέξεων του λεξικού. Αντίθετα, στη δεύτερη μέθοδο, προτείνεται η δειγματοληψία των αρνητικών προτύπων [4].



Σχήμα 2.10: Continuous Bag-of-words. Πηγή [4]

Σχήμα 2.11: *skip-gram*. Πηγή [4]

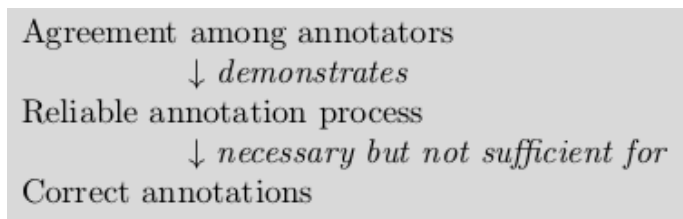
2.5.3.5 Το Glove μοντέλο

Μία εναλλακτική μέθοδος είναι μέσω του **Glove** μοντέλου [39]. Στον αλγόριθμο αυτόν, το μοντέλο προσπαθεί να προβλέψει την πιθανότητα εμφάνισης μιας λέξης στο context μιας άλλης λέξης. Η μάθηση πραγματοποιείται μέσω της βελτιστοποίησης μιας συνάρτησης ελαχίστων τετραγώνων. Σκοπός είναι να δημιουργηθεί ένας χώρος αναπαράστασης, ο οποίος θυμάται όλη τη σημαντική σημασιολογική πληροφορία των λέξεων. Ωστόσο, δεδομένου πως δεν θα συναντήσουμε το Glove μοντέλο στη συνέχεια της μελέτης μας, προτρέπουμε τον αναγνώστη στο [39].

2.6 Συμφωνία Επισημειωτών

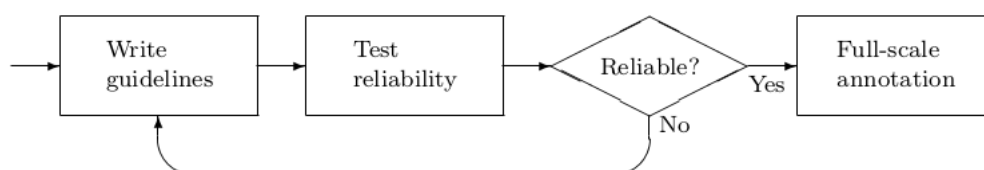
Προκειμένου να προχωρήσουμε στη διαδικασία επισημείωσης ενός συνόλου παρατηρήσεων, πρέπει αρχικά να ελέγξουμε κατά πόσο οι επισημειωτές συμφωνούν μεταξύ τους. Έτσι, στην ενότητα αυτή, θα μελετήσουμε την συμφωνία των επισημειωτών (**Inter-annotator agreement**).

Το βασικό πρόβλημα, στην περίπτωση που έχουμε παραπάνω από δύο επισημειωτές, είναι πως εάν δεν προκύψει ικανοποιητική συμφωνία μεταξύ τους, τότε πιθανόν οι τελικές επισημειώσεις να είναι λανθασμένες. Σύμφωνα με την Εικόνα 2.6, η ένδειξη συμφωνίας μεταξύ των επισημειωτών υποδηλώνει την αξιοπιστία των επισημειώσεων, που αποτελεί αναγκαία αλλά όχι ικανή συνθήκη για σωστές επισημειώσεις.



Εικόνα 2.6: Το σχήμα του *Inter-annotator agreement*. Πηγή [5]

Οι επισημειωτές πρέπει να αποτελούν γνώστες του μελετούμενου θέματος. Κατά την διαδικασία του *Inter-annotator agreement*, ο καθένας πρέπει να δίνει επισημειώσεις με τέτοιο τρόπο, ώστε να μην γνωρίζει τις αντίστοιχες επισημειώσεις των υπολοίπων. Η διαδικασία αυτή είναι επαναληπτική. Αρχικά, επιλέγεται ένα αντιπροσωπευτικό δείγμα του συνόλου δεδομένων. Οι επισημειωτές ορίζουν τις οδηγίες, με τις οποίες αυτοί θα δίνουν τις αντίστοιχες επισημειώσεις. Στη συνέχεια, δίνουν επισημειώσεις πάνω στο δείγμα των παρατηρήσεων, που επιλέχθηκε. Εάν προκύψει η επιθυμητή συμφωνία, τότε συνεχίζουν με τη διαδικασία επισημείωσης όλου του συνόλου δεδομένων. Εάν όχι, τότε ξαναορίζονται οι οδηγίες επισημειώσεων και ξαναεπισημειώνουν το δείγμα παρατηρήσεων - πάλι ανεξάρτητα μεταξύ τους - μέχρις ότου να ικανοποιηθεί η επιθυμητή συμφωνία. Η παραπάνω διαδικασία απεικονίζεται στο Σχήμα 2.12.



Σχήμα 2.12: Επαναληπτική διαδικασία *Inter-annotator agreement*. Πηγή [5]

Θα μελετήσουμε τρεις μετρικές που υποδηλώνουν την συμφωνία των επισημειωτών: το Raw Agreement, το Cohen's kappa και το Krippendorff's alpha.

2.6.1 Raw Agreement

Το **Raw Agreement** (ή Observed Agreement) είναι μια βασική μετρική της συμφωνίας των επισημειωτών, και αποτελεί, ίσως, την πιο συχνή μέθοδο για μέτρηση της συμφωνίας [40]. Δίνεται από την σχέση

$$\text{Raw Agreement} = \frac{\text{Number of Agreed Annotations}}{\text{Number of All Annotations}} \quad (2.52)$$

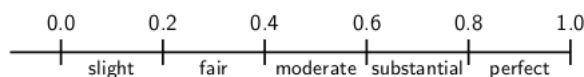
Το βασικό αρνητικό με το Raw Agreement είναι ότι δεν λαμβάνει καθόλου υπόψη την πιθανότητα ορισμένες συμφωνημένες επισημειώσεις να προέκυψαν τυχαία. Επίσης, δεν προσμετράει τις διαφορές μεταξύ των μη συμφωνηθέντων επισημειώσεων. Παρ' όλα αυτά, αποτελεί μια ένδειξη συμφωνίας.

2.6.2 Cohen's kappa

Ο πλέον αποδεκτός τρόπος για να δεχτούμε την συμφωνία ανάμεσα στους επισημειωτές είναι οι μετρικές της οικογένειας των kappa/alpha [5]. Το **Cohen's kappa** [41] αποτελεί μια κλασική στατιστική μετρική, που μετράει το ποσό της συμφωνίας που διαπιστώθηκε πάνω από το επίπεδο της αναμενόμενης τυχαίοτητας. Το Cohen's kappa δίνεται από την σχέση

$$\kappa = \frac{A_0 - A_e}{1 - A_e} \quad (2.53)$$

όπου A_0 είναι το πραγματικό ποσοστό συμφωνίας και το A_e είναι το αναμενόμενο. Έτσι το κ είναι ένας συντελεστής φραγμένος στο $[0, 1]$. Σε περίπτωση τέλει συμφωνίας, αυτό είναι ίσο με 1, ενώ σε περίπτωση παντελούς διαφωνίας ίσο με 0. Στο Σχήμα 2.13 αναπαριστούμε τις ενδεικτικές κλίμακες αξιοπιστίας των τιμών για το κ .



Σχήμα 2.13: Κλίμακες αξιοπιστίας των τιμών του Cohen's kappa

2.6.3 Krippendorff's alpha

Μια μετρική της οικογένειας των kappa/alpha που εκφράζεται στα πλαίσια της διαφωνίας των επισημειωτών παρά της συμφωνίας τους είναι το **Krippendorff's alpha** [42]. Η μετρική αυτή λαμβάνει υποψήν τις αποστάσεις μεταξύ των όλων των επισημειώσεων, που δόθηκαν για μια παρατήρηση, θεωρώντας πως οι δυνατές διαφορετικές επισημειώσεις λαμβάνουν διάταξη (ordered annotations). Δίνεται από την σχέση

$$\alpha = 1 - \frac{D_0}{D_e} \quad (2.54)$$

όπου

$$D_0 = \frac{1}{Ic(c-1)} \sum_{i \in I} \sum_{k_1 \in K} \sum_{k_2 \in K} n_{ik_1} n_{ik_2} d_{k_1 k_2} \quad (2.55)$$

με I να είναι το πλήθος των παρατηρήσεων προς επισημείωση, c ο αριθμός των επισημειωτών, K το πλήθος των διακεκριμένων επισημειώσεων (labels), n_{ik} ο αριθμός των επισημειωτών που έθεσαν για την παρατήρηση i την επισημείωση k και $d_{k_i k_j} = (r(k_i) - r(k_j))^2$, όπου $r(k_i)$ είναι η βαθμίδα της επισημείωσης i ,

και

$$D_e = \frac{1}{N(N-1)} \sum_{k_1 \in K} \sum_{k_2 \in K} n_{k_1} n_{k_2} d_{k_1 k_2} \quad (2.56)$$

με N το συνολικό πλήθος των επισημειώσεων που δόθηκαν και n_k το συνολικό πλήθος των επισημειώσεων k που δόθηκαν από όλους τους επισημειωτές.

Στο Σχήμα 2.14 αναπαριστούμε τις κλίμακες αξιοπιστίας των τιμών του Krippendorff's alpha. Σημειώνουμε ότι το Krippendorff's alpha δεν θεωρεί ότι η διαφωνία είναι αυστηρά το

συμπληρωματικό της συμφωνίας. Για να ισχύει αυτό, τότε πρέπει $a = κ$, δηλαδή θέτοντας $D_0 = 1 - A_0$ και $D_e = 1 - A_e$.



Σχήμα 2.14: Κλίμακες αξιοπιστίας των τιμών του Krippendorff's alpha

2.7 Στατιστικός Έλεγχος

Στο σημείο αυτό, θα συζητήσουμε για τον στατιστικό έλεγχο των αποτελεσμάτων. Στην Επιβλεπόμενη Μάθηση, υπάρχουν πολλές περιπτώσεις στις οποίες για ένα πρόβλημα δοκίμαζουμε αρκετά διαφορετικά μοντέλα μάθησης. Τις περισσότερες φορές, προτού αρχίσει η μάθηση, καταγράφουμε τις μετρικές, με τις οποίες θα ελέγξουμε την επίδοση των μοντέλων αυτών. Έτσι, συχνά επιλέγουμε να κρατήσουμε ως καλύτερο αυτό με τις καλύτερες μετρικές επιδόσεις. Το πρόβλημα, το οποίο αναφέρεται στην επιλογή του καλύτερου μοντέλου μάθησης ανάμεσα σε μοντέλα που έχουν εκπαιδευτεί πάνω στο ίδιο πρόβλημα, συχνά αποκαλείται **model selection**.

Προκειμένου να συγκρίνουμε αποτελεσματικά τις επιδόσεις ανάμεσα σε διαφορετικά μοντέλα, είναι σημαντικό να προβούμε σε κάποιο στατιστικό έλεγχο. Έαν η επίδοση ενός μοντέλου A είναι καλύτερη αριθμητικά από ένα άλλο μοντέλο B, και κάποιο στατιστικό τεστ έχει δείξει στατιστική σημαντικότητα (**statistical significance**) - λαμβάνοντας τα αναλυτικά αποτελέσματά τους - τότε θα θεωρούμε πως το A είναι καλύτερο στατιστικά από το B, σε κάποιο προκαθορισμένο επίπεδο αβεβαιότητας. Ωστόσο, προκειμένου τα αποτελέσματα των στατιστικών τεστ να είναι ορθά, πρέπει να μην καταπατώνται οι εκάστοτε απαραίτητες συνθήκες τους.

Η καταγραφή της στατιστικής σημαντικότητας, σχετικά με τα αποτελέσματα των μοντέλων μάθησης, παρέχει τα εξής θετικά στοιχεία σε ένα πείραμα:

- εγγύηση για *επαναληψιμότητα* των αποτελεσμάτων
- εγγύηση για *ευρωστία* του μοντέλου

Στις υποενότητες που ακολουθούν, θα μιλήσουμε για στατιστικά τεστ, που θα χρησιμοποιήσουμε στα πειράματά μας. Θα συζητήσουμε για το paired-t-test (2.7.1), για το Shapiro-Wilk test (2.7.2) και το Wilcoxon signed-rank test (2.7.3)

2.7.1 paired-t-test

Το *paired-t-test* [43] είναι ένα στατιστικό τεστ, που παίρνει σαν είσοδο δύο σειρές αποτελεσμάτων από διαφορετικές μετρήσεις, και ελέγχει εάν η μέση διαφορά των ζευγαριών των μετρήσεων είναι ίση με το 0 ή όχι. Το τεστ αυτό εκτελείται, όταν τα αποτελέσματα των δυο μετρήσεων μπορούν να παραταχθούν *σε ζεύγη* μεταξύ τους. Για να διεξαχθεί αυτό το τεστ πρέπει να ικανοποιούνται κάποιες συνθήκες:

- Η εξαρτημένη μεταβλητή των διαφορών των αποτελεσμάτων των δυο μετρήσεων πρέπει να ακολουθεί την κανονική κατανομή.

- Οι παρατηρήσεις των δύο μετρήσεων πρέπει να είναι στατιστικά ανεξάρτητες.
- Η εξαρτημένη μεταβλητή πρέπει να είναι συνεχής.
- Η εξαρτημένη μεταβλητή δεν πρέπει να περιέχει απομακρυσμένες τιμές (outliers).

Το paired-t-test έχει μηδενική (null) υπόθεση ότι οι διαφορές των δύο μετρήσεων έχουν μηδενική κατανομή και άρα οι μετρήσεις προέρχονται από την ίδια κατανομή. Αν απορριφθεί η null υπόθεση, τότε θεωρούμε ότι οι μετρήσεις διαφέρουν με στατιστική σημαντικότητα.

Η διαδικασία του paired-t-test είναι η εξής: Υπολογίζουμε την μέση τιμή \bar{x}_d των διαφορών των μετρήσεων d . Στη συνέχεια, υπολογίζουμε το standard error (SE)

$$SE = \frac{s_d}{\sqrt{n}} \quad (2.57)$$

όπου το n είναι το πλήθος των παρατηρήσεων (ζευγαριών) και το s_d είναι η τυπική απόκλιση των διαφορών d . Υπολογίζουμε το

$$T = \frac{\bar{x}_d}{SE} \quad (2.58)$$

Λαμβάνοντας υπόψιν πως οι βαθμοί ελευθερίας είναι $df = n - 1$, κάτω από την null υπόθεση πρέπει αυτή η τιμή να ακολουθεί την t -κατανομή με $\alpha=0.05$ και $df = n - 1$. Κοιτώντας τους αντίστοιχους πίνακες, βγάζουμε το p-value. Στην παρούσα εργασία, θεωρούμε αυθαίρετα ότι αν το p-value < 0.05 , τότε μπορούμε να απορρίψουμε την null υπόθεση.

Κατά αυτόν τον τρόπο, στο πρόβλημα του model selection, εάν οι μετρήσεις των πειραμάτων μας τηρούν τις συνθήκες του paired-t-test, μπορούμε να προτιμήσουμε ένα μοντέλο σε σχέση με κάποιο άλλο. Αρκεί να έχει καλύτερη επίδοση και το αντίστοιχο p-value τους να είναι μικρότερο από το 0.05.

2.7.2 Shapiro-Wilk test

Όπως αναφέραμε παραπάνω, βασική απαίτηση για να διεξάγουμε το paired-t-test είναι η τήρηση της συνθήκης για κανονική κατανομή. Ωστόσο πώς θα καταλάβουμε εάν οι παρατηρήσεις ακολουθούν την κανονική κατανομή;

Λύση σε αυτό το πρόβλημα έρχεται να δώσει ένα άλλο στατιστικό τεστ, το *Shapiro-Wilk test* [44], το οποίο θεωρεί σαν null υπόθεση ότι οι παρατηρήσεις ακολουθούν την κανονική κατανομή.

Η διαδικασία του τεστ αυτού είναι η ακόλουθη. Θεωρώντας πως έχουμε παρατηρήσεις x_i , με $i \in [1, N]$, όπου N το πλήθος των παρατηρήσεων, βρίσκουμε το άθροισμα των διαφορών των παρατηρήσεων από την μέση τιμή \bar{x} και συμβολίζουμε με SS

$$SS = \sum_{i=1}^N (x_i - \bar{x})^2 \quad (2.59)$$

Στη συνέχεια, ταξινομούμε τις παρατηρήσεις σε αύξουσα σειρά και βρίσκουμε το δείκτη

(index) της διαμέσου αυτών, έστω m . Υπολογίζουμε το παρακάτω άθροισμα b ,

$$b = \sum_{i=1}^m a_i(x_{N+1-i} - x_i) \quad (2.60)$$

όπου a_i [45] είναι βάρη που αντιστοιχούν στην τιμή του N . Τελός, βρίσκουμε την τιμή του W ,

$$W = \frac{b^2}{SS} \quad (2.61)$$

και δεδομένου και του N , υπολογίζουμε την αντίστοιχη τιμή του p-value [45].

Όπως και προηγουμένως, εάν το p-value είναι μικρότερο από το 0.05, θεωρούμε ότι απορρίπτεται η null υπόθεση. Με άλλα λόγια, θεωρούμε ότι οι παρατηρήσεις δεν ακολουθούν την κανονική κατανομή. Έτσι, για να διεξάγουμε το paired-t-test πρέπει τα αντίστοιχα p-values να είναι μεγαλύτερα από 0.05.

2.7.3 Wilcoxon signed-rank test

Εάν το paired-t-test δεν μπορεί να πραγματοποιηθεί στην περίπτωση, που η συνθήκη για κανονική κατανομή έχει καταπατηθεί, τότε χρειάζεται να διεξάγουμε κάποιο μη παραμετρικό τεστ. Μη παραμετρικά τεστς είναι εκείνα για τα οποία δεν μας ενδιαφέρει εξ αρχής η ακριβής κατανομή των παρατηρήσεων των μετρήσεων. Για αυτό το λόγο, λέγονται και *distribution free tests*. Ένα τέτοιο τεστ είναι το *Wilcoxon signed-rank test*. Για το πρόβλημα του model selection, το τεστ αυτό μας ενδιαφέρει για παρατηρήσεις που αντιστοιχούν σε ζεύγη.

Η null υπόθεση είναι ίδια με το paired-t-test, θεωρώντας πως οι διαφορές των μετρήσεων πρέπει να ακολουθούν την συμμετρική κατανομή γύρω από το 0 - δηλαδή να έχουν μηδενική μέση τιμή. Το τεστ αυτό προτιμάται σε σχέση με το paired-t-test, όταν κάποιες συνθήκες του δεύτερου καταπατώνται [46].

Η διαδικασία διεξαγωγής του Wilcoxon signed-rank test είναι η ακόλουθη. Αρχικά παίρνουμε όλες τις αντίστοιχες διαφορές d_i των μετρήσεων, με $i \in [1, N]$, όπου N το πλήθος των παρατηρήσεων σε κάθε μέτρηση. Στη συνέχεια, ταξινομούμε σε ranks, αγνοώντας το πρόσημο, ώστε το πρώτο rank να περιέχει την μικρότερη διαφορά $|d_i|$ κ.ο.κ. Ύστερα, επισημειώνουμε το κάθε rank με το πρόσημο της αντίστοιχης διαφοράς d_i . Υπολογίζουμε τα W^+ και W^- , όπου το πρώτο είναι το άθροισμα όλων των ranks με θετικό πρόσημο και το δεύτερο είναι το άθροισμα όλων των ranks με αρνητικό πρόσημο.

Θεωρώντας πως βρισκόμαστε στην null υπόθεση, υποθέτουμε πως η κατανομή των θετικών και αρνητικών ranks είναι κατανεμήμενη με τυχαίο τρόπο ανάμεσα στις πιθανές τιμές όλων των ranks. Έτσι, επιλέγουμε $W = \min(W^+, W^-)$. Τέλος, με βάση τους πίνακες για το Wilcoxon signed-rank test βρίσκουμε το p-value.

Και πάλι, θεωρούμε πως αν το p-value είναι μικρότερο από 0.05, τότε η null υπόθεση καταρρίπτεται. Άρα, σε μια τέτοια περίπτωση θεωρούμε πως οι παρατηρήσεις των δύο μετρήσεων δεν ανήκουν στην ίδια κατανομή με στατιστική σημαντικότητα.

Κεφάλαιο **3**

Focused Crawler - Εστιασμένη Διαδικτυακή Ανίχνευση

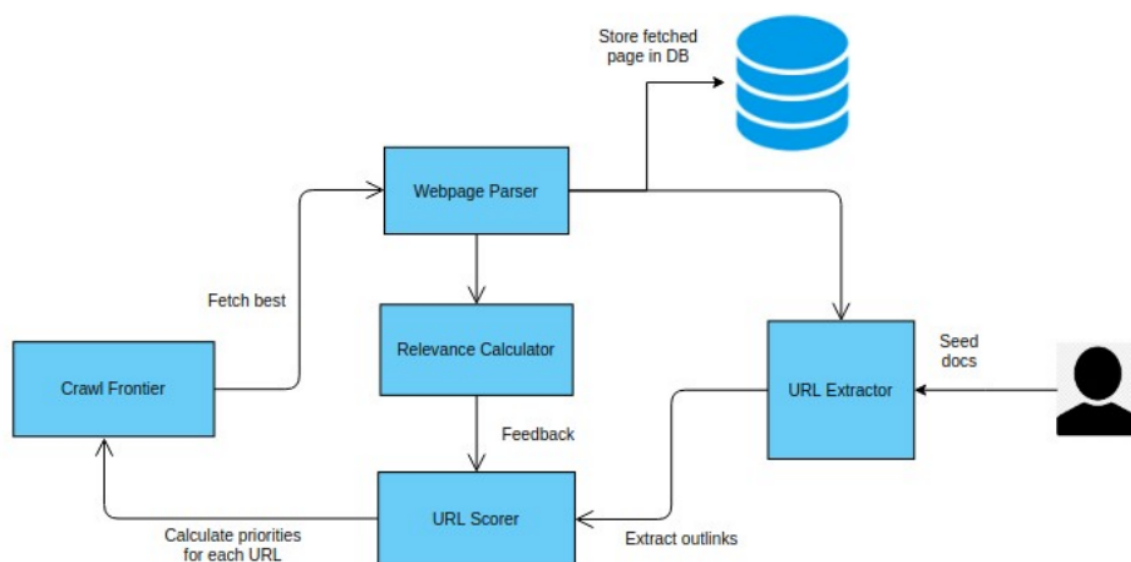
3.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζουμε την βασική θεωρητική ανασκόπηση σχετικά με την αναζήτηση κειμένων συγκεκριμένης θεματολογίας στο Διαδίκτυο. Η παραπάνω διαδικασία αποτελείται από τη δημιουργία ενός ευφυούς συστήματος αναζήτησης, που εστιάζει σε συγκεκριμένα θέματα μελέτης και προσπαθεί να βρίσκει όσο το δυνατόν περισσότερες σχετικές ιστοσελίδες στο Διαδίκτυο. Ένα τέτοιο σύστημα συχνά εμφανίζεται στην βιβλιογραφία με τον τίτλο του εστιασμένου διαδικτυακού ανιχνευτή (*focused crawler*), του Ρομπότ (Robot) ή της Αράχνης (Spider) [47]. Εμείς από εδώ και πέρα θα χρησιμοποιούμε τον όρο του *focused crawler*. Ακόμη, θα αναφερόμαστε στην συνάφεια μιας ιστοσελίδας, με το επιθυμητό θέμα αναζήτησης, απλά με τη λέξη “*συνάφεια*”.

Οι *focused crawlers* λειτουργούν ως αυτόνομοι πράκτορες σχεδιασμένοι με τέτοιο τρόπο, ώστε να συλλέγουν ιστοσελίδες, σχετικές με ένα προκαθορισμένο θέμα που ορίζει ο χρήστης. Δοθεισών μίας ή περισσότερων αρχικών σελίδων, που ονομάζονται σπόροι (**seeds**), ο *focused crawler* βρίσκει σχετικές ιστοσελίδες. Αυτό γίνεται με το να αξιοποιεί τους υπερσυνδέσμους (hyperlinks) και τους Ενιαίους Εντοπιστές Πόρων (Uniform Resource Locators ή URLs), που περιέχονται στις ιστοσελίδες αυτών που έχει ήδη επισκεφθεί. Τις περισσότερες φορές, οι *focused crawlers* χρησιμοποιούν μια δομή, η οποία ονομάζεται μέτωπο αναζήτησης (**crawl frontier** ή **frontier**) που χρειάζεται για την προσωρινή αποθήκευση των προς επίσκεψη (unvisited) URLs, μέχρις ότου αυτά να προσπελαστούν. Σε κάθε νέο URL που εξάγεται από την τρέχουσα ανακτημένη (visited) σελίδα, προκειμένου να εισαχθεί στο μέτωπο αναζήτησης μαζί με τα υπόλοιπα unvisited URLs, του ανατίθεται μια τιμή προτεραιότητας που αφορά την εκτίμηση του *focused crawler* για την συνάφειά του. Ο *focused crawler* επιλέγει να επισκεφτεί κάθε φορά εκείνο το URL από το μέτωπο αναζήτησης με την μεγαλύτερη τιμή προτεραιότητας. Παράλληλα, το αποθηκεύει - πχ. σε μια βάση δεδομένων - και το εξάγει στον χρήστη. Όταν το σύστημα επισκέφτεται μια ιστοσελίδα, υπολογίζει και την αντίστοιχη τιμή συνάφειας της. Έτσι, λαμβάνοντας υπόψιν αυτήν την τιμή, ο *focused crawler* καλείται να προσαρμόζει την συμπεριφορά του, βελτιώνοντας τον τρόπο που υπολογίζει τις προτεραιότητες των URLs.

Επιπλέον, πολύ συχνά γίνεται χρήση ενός είδους ταξονομίας. Από αυτή, παρέχονται στον

σύστημα λέξεις-κλειδιά (keywords), τα οποία σχετίζονται με πιθανά διαφορετικά επιθυμητά θέματα ή/και υποθέματα. Τα θέματα αυτά μπορούν να οργανωθούν σε ένα δέντρο ταξινόμησης θεμάτων ταξινόμησης των κειμένων. Η χρήση ταξινόμησης αποσκοπεί στην ύπαρξη προϋπάρχουσας γνώσης, σχετικά με την ιεραρχία των θεμάτων. Με αυτόν τον τρόπο, συχνά, γίνεται μία ταξινόμηση των κειμένων ως προς τα επιθυμητά θέματα, με βάση την ύπαρξη μιας λέξης-κλειδιού ή ενός συνδυασμού αυτών.



Σχήμα 3.1: Αρχιτεκτονική γενικού μοντέλου *focused crawler*

Η διαδικασία του *focused crawling* συνεχίζεται διαρκώς, προσπευλαύνοντας κάθε φορά το πιο υποσχόμενο URL από το μέτωπο αναζήτησης· δηλαδή εκείνο με την μεγαλύτερη τιμή προτεραιότητας. Αυτή συνήθως τερματίζεται, όταν έχει προσπελαστεί ένας επαρκής αριθμός σελίδων, είτε όταν εξαντληθούν σε μεγάλο βαθμό τα αποθέματα μνήμης [47]. Στο Σχήμα 3.1, απεικονίζεται η αρχιτεκτονική του **γενικού *focused crawler*** μοντέλου, πάνω στο οποίο θα στηρίξουμε και εμείς την δική μας προσέγγιση, που θα περιγράψουμε εκτενώς στις επόμενες ενότητες. Στο σχήμα αυτό, κύρια συστατικά του συστήματος αποτελούν το μέτωπο αναζήτησης (Crawl Frontier), ο εκτιμητής συνάφειας (Relevance Calculator), καθώς και ο εκτιμητής των τιμών προτεραιότητας των URLs (URL Scorer). Επιπλέον, προσθέτουμε και ένα κλειστό σύνολο (Closure), στο οποίο αποθηκεύονται τα URLs που επιστρέφονται από το σύστημα. Με το κλειστό σύνολο εξασφαλίζουμε ότι όλα τα URLs που θα φέρει το σύστημα είναι μεταξύ τους διαφορετικά. Εκτενής περιγραφή του γενικού μοντέλου παρουσιάζεται στον Αλγόριθμο 3.1.

Αξίζει να σημειώσουμε πως ο απώτερος σκοπός ενός *focused crawler* είναι να βρει όσες περισσότερες σχετικές, με το θέμα αναζήτησης, ιστοσελίδες, αλλά και όσες πιο λίγες μη σχετικές. Επιπλέον, εξίσου σημαντικό θεωρείται να μπορεί να ανταποκριθεί και στις λειτουργικές απαιτήσεις των χρηστών, που αφορούν τη χρονική και χωρική βελτιστοποίηση της αναζήτησης. Στην παρακάτω ενότητα, ακολουθεί μία συγκεντρωτική περιγραφή του μοντέλου του *focused crawler* από την σχετική βιβλιογραφία.

ΑΛΓΟΡΙΘΜΟΣ 3.1: Γενικός Αλγόριθμος Focused Crawler

Input: *seeds* URLs, *TIMESTEPS*, URLScorer and RelevanceCalculator
 Initialize empty crawl frontier *F* and closure *C*
while *seeds* not empty **do**
 Pop a URL *url* from *seeds*
 Add *url* to *C*
 Fetch webpage *w url*
 L := Extract all outlinks URLs from *w* that are not in *C*
 for *l* in *L* **do**
 score := URLScorer(*l*)
 Add tuple (*l, score*) to *F*
 end for
end while
timestep := 1
while not *timestep* > *TIMESTEPS* **do**
 Select the (*url, score*) from *F* with the best score
 Observe relevance := RelevanceCalculator(*url*)
 Give feedback to URLScorer with respect to the observed relevance
 Add *url* to *C*
 Delete all tuples from *F* that have as URL the *url*
 L' := Extract all outlinks from *url* that are not in *C*
 for *l* in *L'* **do**
 score := URLScorer(*l*)
 Add tuple (*l, score*) to *F*
 end for
 timestep := *timestep* + 1
end while
Output: *C*

3.2 Σχετική Βιβλιογραφία και Προηγούμενες Μελέτες

Μερικές κλασικές προσεγγίσεις [47, 48] για focused crawling περιλάμβαναν την προσαρμογή της συμπεριφοράς μιας υπάρχουσας μηχανής αναζήτησης στα προαπαιτούμενα του χρήστη. Πιο αναλυτικά, τα προαπαιτούμενα του χρήστη μπορούσαν να εκφραστούν με έναν από τους εξής τρόπους: (α) με ένα ερώτημα (query) αποτελούμενο από λέξεις κλειδιά, (β) με ένα αντιπροσωπευτικό κείμενο σχετικό με εκείνα που ενδιαφέρεται ο χρήστης, ή (γ) με ένα σύνολο κειμένων σχετικών με τα ενδιαφέροντα του χρήστη. Στις περιπτώσεις αυτές, το σύστημα χρησιμοποιεί ερωτήματα (queries) πάνω σε μία συγκεκριμένη μηχανή αναζήτησης, προκειμένου να κατασκευάσει μία βάση δεδομένων από κείμενα. Αυτά λαμβάνονται ως πιθανώς σχετικά με το επιθυμητό θέμα. Εναλλακτικά, το σύστημα ξεκινά την αναζήτηση από ένα σύνολο κεντρικών σημείων - ιστοσελίδων - στο διαδίκτυο, προκαθορισμένα από τον ίδιο το χρήστη. Η σύγκριση των κειμένων, προκειμένου να αποφανθεί το κατά πόσο είναι σχετική μία ιστοσελίδα, υπολογίζεται με τη χρήση ενός μοντέλου της θεωρίας ανάκτησης πληροφορίας (Information Retrieval)· όπως το Boolean Model ή το Vector Space Model.

Ο Chakrabarti κ.ά [49] πρώτοι εισήγαγαν την έννοια του focused crawler, για να αναζητούν επιλεκτικά σελίδες σχετικές ως προς ένα προκαθορισμένο σύνολο θεμάτων. Χρη-

σιμοποίησαν έναν ταξινομητή και ένα διυλιστή (distiller), ώστε να καθοδηγούν το crawling αναγνωρίζοντας hubs - κομβικές ιστοσελίδες που εξάγουν αρκετά σχετικά URLs, και αξιολογώντας τις προτεραιότητες επίσκεψης των ιστοσελίδων, σύμφωνα με ένα μέτρο κεντρικότητας. Ο intelligent crawler [15], που προτάθηκε από τον Aggarwal κ.ά., χρησιμοποιεί ένα στατιστικό μοντέλο, με βάση το οποίο ο focused crawler δίνει προτεραιότητες στα URLs που βρίσκονται στο μέτωπο αναζήτησης. Και πάλι ο Chakrabarti κ.ά [50] δημιούργησαν τον enhanced focused crawler που αποτελείται για πρώτη φορά από δυο ταξινομητές, τον baseline [49, 51] και έναν απλοποιημένο ενισχυμένο ταξινομητή, που είχε τον ρόλο του μαθητή (apprentice): αμφότεροι είναι Rainbow Naive Bayes [52] ταξινομητές. Ο apprentice δίνει τις προτεραιότητες για τα προς επίσκεψη URLs αξιοποιώντας τα Document Object Model (DOM) [53] χαρακτηριστικά της πηγαίας ιστοσελίδας, ενώ παράλληλα δέχεται online ανατροφοδότηση και παραδειγμάτα προς εκπαίδευση από τον baseline. Ωστόσο, μόνο όταν ο apprentice λάβει έναν επαρκή αριθμό παραδειγμάτων, ο focused crawler θα τον θέσει εν δράσει, ώστε να δίνει αυτός, και όχι ο baseline, τις προτεραιότητες στα URLs του μετώπου αναζήτησης.

Οι σημασιολογικοί (semantic) focused crawlers εκτιμούν τις ομοιότητες, ανάμεσα σε ιστοσελίδες και έννοιες οντολογιών - προκαθορισμένων από τις προτιμήσεις του χρήστη - ταξινομώντας τα URLs βάσει οντολογίας [54]. Ο Tane κ.ά [55] πρότειναν έναν βασισμένο στην οντολογία (ontology-based) focused crawler, με τον οποίο ο χρήστης αρχικά δίνει ο ίδιος τα βάρη στις έννοιες (topics) μιας οντολογίας. Στη συνέχεια, τα βάρη άλλων εννοιών μπορούν να υπολογιστούν χάρη στις αλληλοσυσχετίσεις τους μέσα στην οντολογία. Με αυτό τον τρόπο, ταίριαζαν την τρέχουσα ιστοσελίδα και τα URLs, που αυτή εξάγει, με τα βάρη των εννοιών. Έτσι, τα URLs μπορούσαν να κατατάσσονται και να συσταδοποιούνται σύμφωνα με τις έννοιες που προκύπτουν. Ο Lokman [56], που χρησιμοποίησαν οι Can και Baykal, συλλέγει URLs σχετικά με την ιατρική, υπολογίζοντας τη συνάφεια της τρέχουσας σελίδας και των URLs αναφορικά με έννοιες από το Unified Medical Language System (UMLS) [57]. Χρησιμοποιούν τις συχνότητες και τα βάρη των εννοιών, καθώς και ένα μέτωπο αναζήτησης για την αποθήκευση των προς επίσκεψη URLs.

Παρά την ήδη 20 χρόνων ιστορία του, το focused crawling ακόμα λαμβάνει σημαντική προσοχή, προσπαθώντας να είναι ακόμα πιο πολύ αποτελεσματικό. Οι García και Carriegos [58] προσπάθησαν να τρέξουν πολλά παράλληλα ερωτήματα ώστε να αυξήσουν την ταχύτητα της ανάκτησης πληροφορίας, ενώ ο Πίου και άλλοι [59] πρότειναν μία προσαρμοστική μέθοδο για online focused crawling με εφαρμογή στον Ισλαμικό εξτρεμισμό.

Αρκετές προηγούμενες μελέτες έχουν εφαρμόσει Ενισχυτική Μάθηση (Reinforcement Learning) στο focused crawling. Οι Rennie και McCallum [60] εισήγαγαν τη χρήση της Ενισχυτικής Μάθησης στο Web crawling, με μία μέθοδο που χρειαζόταν offline εκπαίδευση με χρήση tabular μεθόδων. Εφάρμοσαν bag-of-words στο κείμενο - της ιστοσελίδας που εξάγει - το οποίο βρίσκεται στη γειτονιά των κειμένων των URLs. Αφού δημιούργησαν ζεύγη με τα URLs και τις αντίστοιχες τιμές της value function, κατασκεύασαν μία γενική αντιστοίχιση ανάμεσα στα κείμενα και τις τιμές αυτές, χρησιμοποιώντας Naive Bayes ταξινομητή. Οι Partalas κ.ά [61] πρότειναν την RLwCS μέθοδο, κατά την οποία ο focused crawler διατηρεί μια συλλογή ταξινομητών από την WEKA machine learning library [62], οι οποίοι μπορούν να χρησιμοποιηθούν για την εκτίμηση της συνάφειας μιας ιστοσελίδας. Η μέθοδος αυτή

αναζητά μια στρατηγική (policy) για την επιλογή του καλύτερου ταξινομητή εξ' αυτών για κάθε ιστοσελίδα, ενώ παράλληλα οι ταξινομητές εκπαιδεύονται online καθόλη τη φάση της μάθησης. Ο Wei κ.ά [63] εστίασαν στο πώς θα κατατάσσουν κείμενα τα οποία έχουν συλλεχθεί από τον Παγκόσμιο Ιστό, αντιμετωπίζοντας έτσι ένα πρόβλημα κατάταξης (ranking). Στο MDPRank, που πρότειναν, μοντελοποίησαν το περιβάλλον ως πεπερασμένο Markov Decision Process (MDP). Χρησιμοποίησαν την REINFORCE, ως μέθοδο για την εκπαίδευση των παραμέτρων του μοντέλου, και το Normalized Discounted Cumulative Gain (NDCG) [64], αξιολογώντας για κάθε θέση (ranking position) τις επιδόσεις όλων των επόμενων. Ο Han και άλλοι [65] πρότειναν την χρήση Ενισχυτικής Μάθησης στο focused crawling λαμβάνοντας ως state την ιστοσελίδα και ως actions τα εξερχόμενα URLs από αυτήν. Πρότειναν μια κοινή αναπαράσταση μεταξύ του state και του action λαμβάνοντας υπόψιν στατιστικά χαρακτηριστικά, προκειμένου ο πράκτορας να είναι σε θέση να αντιλαμβάνεται hubs ιστοσελίδες. Ακόμα, πρότειναν μια μέθοδο διακριτοποίησης των χαρακτηριστικών αυτών, ώστε να μειώσουν τον χώρο των states και των actions - μοντελοποιώντας ένα MDP με διακριτούς χώρους states και actions. Επιπλέον, θεώρησαν πως είναι ανάγκη να γίνεται μη διαδοχική μετάβαση των states, θεωρώντας πως θα επιλέγεται σε κάθε χρονικό βήμα εκείνο το URL από το μέτωπο αναζήτησης του οποίου η αναπαράσταση έχει το μεγαλύτερο Q-value· το οποίο προκύπτει από μία προσεγγιστικής συνάρτησης της βέλτιστης action-value function.

Στην μελέτη μας, θα προσεγγίσουμε την υλοποίηση του focused crawler με συνδυασμό της χρήσης Ενεργού Μάθησης (Active Learning) και Ενισχυτικής Μάθησης (Reinforcement Learning). Θα μελετήσουμε έναν focused crawler δύο σταδίων με βάση το πρότυπο ερευνητή-μαθητή, που προτείνουμε στην επόμενη ενότητα. Το σύστημα μας παράλληλα θα περιλαμβάνει τα βασικά συστατικά του γενικού μοντέλου focused crawler, που περιγράψαμε (βλ. Σχήμα 3.1), δηλαδή το μέτωπο αναζήτησης (Crawl Frontier), τον εκτιμητή συνάφειας (Relevance Calculator) καθώς και τον εκτιμητή των τιμών προτεραιότητας των URLs (URL Scorer). Παράλληλα, όπως θα δούμε και στη συνέχεια, στην περίπτωση ενός προβλήματος εξειδικευμένης - ειδικής - αναζήτησης, θα χρησιμοποιήσουμε ένα σύστημα ταξονομίας από λέξεις-κλειδιά (keywords). Αυτό θα συμβάλλει στην αποτελεσματικότερη εύρεση και αναγνώριση σχετικών κειμένων, καθώς και στην άμεση αξιολόγησή τους. Ο focused crawler μας θα εξειδικεύεται στην αναζήτηση σχετικών κειμένων ιστοσελίδων - που αποτελούν βιβλιογραφικές αναφορές - και θα υποστηρίζει την ερευνητική διαδικασία του επιστήμονα, δίνοντας του την δυνατότητα να δίνει άμεση ανατροφοδότηση (feedback) στο σύστημα· όταν αυτό είναι αναγκαίο.

3.3 Καθορισμός του προβλήματος

Στην σημείο αυτό, θα ορίσουμε το πρόβλημα που έχουμε να αντιμετωπίσουμε. Θέλουμε να φτιάξουμε ένα focused crawler σύστημα, το οποίο θα αναζητά στον Παγκόσμιο Ιστό εθνοφαρμακολογικές βιβλιογραφικές αναφορές (δημοσιεύσεις). Το σύστημά μας θέλουμε να αποτελέσει ένα υποστηρικτικό, αλλά και απαραίτητο, εργαλείο για τον ερευνητή προσφέροντας καινοτόμες λειτουργικότητες με χρήση της Μηχανικής Μάθησης.

Θα θεωρήσουμε πως έχουμε να αντιμετωπίσουμε δύο διαφορετικά προβλήματα αναζήτησης - ένα γενικό και ένα ειδικό. Στο γενικό πρόβλημα, ο ερευνητής θέλει να συλλέξει ένα

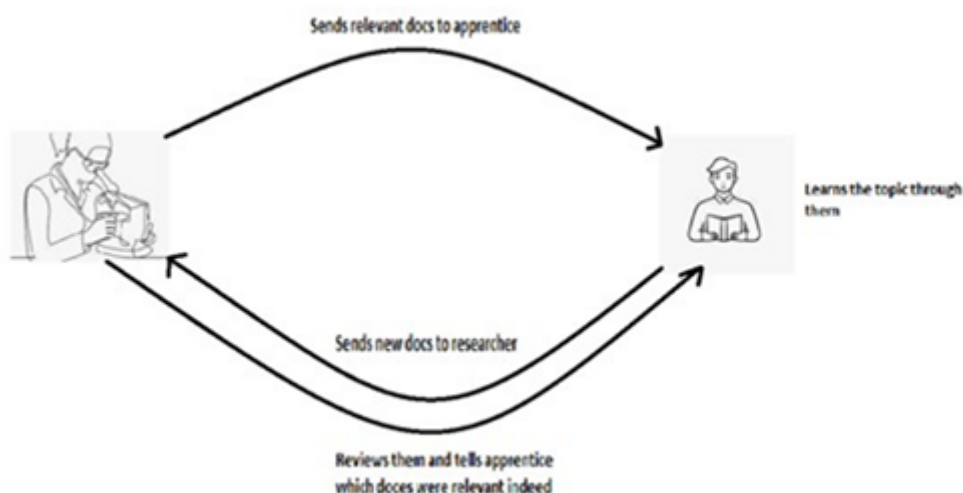
μεγάλο αριθμό αναφορών, οι οποίες σχετίζονται με την εθνοφαρμακολογία, χωρίς ωστόσο να υπάρχει κάποιος άλλος περιορισμός. Στο ειδικό πρόβλημα, θεωρούμε πως ο ερευνητής θέλει να αναζητεί βιβλιογραφικές αναφορές, που σχετίζονται με την εθνοφαρμακολογική παράδοση και γνώση. Ωστόσο, στο πρόβλημα αυτό, θέλουμε, ακόμα, οι αναφορές αυτές να σχετίζονται με συγκεκριμένα γεωγραφικά πλαίσια, και ιδανικά να αφορούν καθορισμένες οικογένειες και κατηγορίες φυτών.

Λαμβάνοντας όλα αυτά υπόψιν, εμείς θα στηρίξουμε την εθνοφαρμακολογική αναζήτηση με την ανάπτυξη ενός ευφυούς *focused crawler*, που θα δέχεται από τον ερευνητή σαν είσοδο λίγες αρχικές *seeds* αναφορές, σε μορφή URLs, τις οποίες, εκείνος θεωρεί πως είναι πολύ ενδιαφέρουσες για την έρευνα του. Στην περίπτωση του ειδικού προβλήματος, θα εξετάσουμε και την περίπτωση που η είσοδος του συστήματος θα περιλαμβάνει και σχετικές λέξεις-κλειδιά (*keywords*). Οι λέξεις-κλειδιά αυτές θα ομαδοποιούνται από τον ερευνητή, σε εκείνες που αφορούν γεωγραφικούς περιορισμούς - όπως χώρες, εθνικότητες, τοποθεσίες κ.ά. - και σε εκείνες που αφορούν συγκεκριμένες οικογένειες και κατηγορίες φυτών. Καθόλη την έρευνά μας στο ειδικό πρόβλημα αναζήτησης, θεωρούμε πως το σύστημα θα λαμβάνει τις λέξεις-κλειδιά από μία άλλη πηγή λογισμικού, η οποία ντετερμινιστικά θα παράγει αυτόματα την ταξινόμηση των λέξεων αυτών.

Ο *focused crawler*, από τη μεριά του, πρέπει να εξάγει στον ερευνητή έναν καθορισμένο αριθμό αναφορών που θεωρεί σχετικές. Παράλληλα, όποτε το θεωρεί αναγκαίο, κατά τη διάρκεια της μάθησης, μπορεί να ζητάει ανατροφοδότηση (*feedback*) από τον ερευνητή, προκειμένου το σύστημα να κατανοεί καλύτερα το θέμα της αναζήτησης. Σημειώνουμε πως είναι σημαντικό ο *focused crawler* να ρωτάει τον χρήστη όσο γίνεται πιο σπάνια, χωρίς αυτό να σημαίνει πως θα φέρνει λιγότερο σχετικά αποτελέσματα. Ακόμα, εξίσου απαραίτητη είναι και η εξασφάλιση μιας αποδοτικής μεθόδου αναζήτησης κειμένων, ώστε ο ερευνητής να μπορεί να χρησιμοποιήσει το σύστημα σε πραγματικό χρόνο.

Διαισθητικά, φανταζόμαστε τον *focused crawler* μας σαν έναν μαθητή (*apprentice*) και τον χρήστη που διεξάγει την αναζήτηση σαν τον ερευνητή (*researcher*)· που έχει αναθέσει στο μαθητή το πρόβλημα της έρευνας των σχετικών κειμένων. Ο ερευνητής του υποδεικνύει, αρχικά, λίγα ορισμένα κείμενα, τα οποία είναι πολύ σχετικά με το θέμα. Ο μαθητής διαβάσει τα κείμενα αυτά και προσπαθεί να κατανοήσει το επιθυμητό θέμα. Στην συνέχεια, αυτός βρίσκει κάποια κείμενα - που θεωρεί πως είναι σχετικά - και ο καθηγητής ελέγχει ορισμένα από αυτά επισημαίνοντας ποια ήταν όντως σχετικά. Ο μαθητής εκπαιδεύεται μέσα από τις υποδείξεις του ερευνητή και φέρνει νέα κείμενα που θεωρεί σχετικά. Ο ερευνητής θα συνεχίζει να διακόπτει και να επισημαίνει κείμενα - που του φέρνει ο μαθητής του - μέχρις ότου να θεωρήσει ότι αυτός έχει μάθει σε ικανοποιητικό βαθμό το σχετικό θέμα της αναζήτησης. Παράλληλα, ο μαθητής θα συνεχίσει την αναζήτησή του, ακόμα και αν ο ερευνητής δεν τον διακόπτει πλέον με επισημειώσεις, μέχρις ότου να συλλέξει έναν επιθυμητό αριθμό κειμένων. Το παραπάνω **πρότυπο ερευνητή-μαθητή**, που προτείνουμε, για τον *focused crawler* απεικονίζεται στο Σχήμα 3.2.

Η κύρια μετρική που θέλουμε να βελτιστοποιήσουμε είναι το ποσοστό επιτυχίας της συλλογής (***harvest rate***) [15], η οποία εκφράζει το ποσοστό των πραγματικά σχετικών κειμένων, που μας επέστρεψε ο *focused crawler*, σε σχέση με όλα τα κείμενα που μας επέστρεψε. Ωστόσο, είναι σημαντικό να μελετήσουμε όλη την εξέλιξη του *harvest rate* κατά την διάρκεια



Σχήμα 3.2: Πρότυπο Ερευνητή-Μαθητή

του τρεξίματος του focused crawler. Το harvest rate δίνεται από την παρακάτω σχέση,

$$\text{Harvest Rate} = \frac{\text{NUMBER OF RELEVANT PAGES CRAWLED}}{\text{NUMBER OF ALL PAGES CRAWLED}}$$

Σε πρακτικό επίπεδο, η βελτιστοποίηση του harvest rate σημαίνει πως το σύστημα θα εξάγει όσα περισσότερα κείμενα ενδιαφέρουν αρκετά τον ερευνητή. Στην περίπτωση του γενικού προβλήματος, υψηλό harvest rate θα σημαίνει την εύρεση πολλών εθνοφαρμακολογικών αναφορών, ενώ στην περίπτωση του ειδικού, υψηλό harvest rate θα σημαίνει την εύρεση πολλών εθνοφαρμακολογικών αναφορών, που σχετίζονται με τα ενδιαφερόμενα γεωγραφικά πλαίσια, καθώς και με τις αντίστοιχες οικογένειες φυτών.

Κεφάλαιο 4

Προτεινόμενες μέθοδοι

Στο κεφάλαιο αυτό, θα παρουσιάσουμε αναλυτικά όλες τις μεθοδολογίες, που θα ακολουθήσουμε, ώστε να λύσουμε το πρόβλημα της αναζήτησης των σχετικών κειμένων. Στην Ενότητα 4.1, θα περιγράψουμε τον Focused Crawler Δύο Σταδίων, που αναφέραμε και στο Κεφάλαιο 3. Ο focused crawler αυτός χρησιμοποιεί Ενεργό και Ενισχυτική Μάθηση αξιοποιώντας το πρότυπο ερευνητή-μαθητή. Στην Ενότητα 4.2, θα συζητήσουμε σχετικά με έναν ντετερμινιστικό focused crawler που επιλέγει κείμενα, με βάση άπλειστα κριτήρια. Ο ντετερμινιστικός focused crawler δεν χρησιμοποιεί καθόλου Μηχανική Μάθηση ή κάποιο είδος Τεχνητής Νοημοσύνης. Έτσι, θεωρούμε πως αξίζει να μελετήσουμε κατά πόσο αποτελεσματικός είναι σε σχέση με τον πρώτο.

4.1 Focused Crawler Δύο Σταδίων

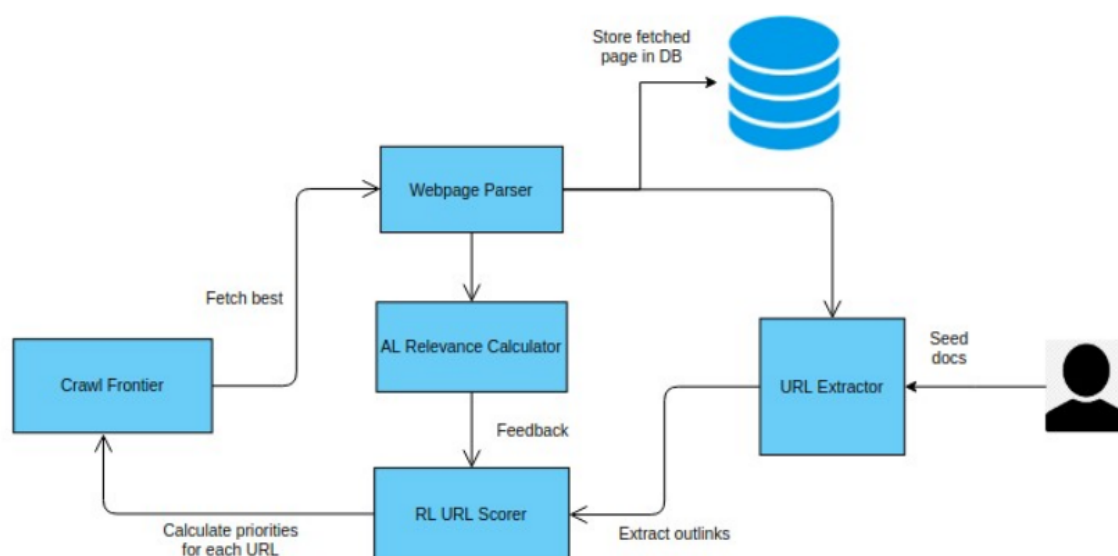
Στην δική μας προσέγγιση για την επίλυση του προβλήματος αναζήτησης σχετικών ιστοσελίδων στο Διαδίκτυο, προτείνουμε ένα σύστημα focused crawler το οποίο περιέχει δύο στάδια ανάπτυξης. Το συστημά μας βασίζεται στο πρότυπο ερευνητή-μαθητή που προτείναμε στο Κεφάλαιο 3.

Στο πρώτο στάδιο, ο ερευνητής εκπαιδεύει τον μαθητή του, προκειμένου αυτός να είναι σε θέση να ταξινομή μια δημοσίευση σε μία από τις ορισμένες κατηγορίες, που έχει ορίσει ο ερευνητής. Προτείνουμε την εκπαίδευση του μαθητή με χρήση αλγορίθμων Ενεργού Μάθησης (**Active Learning**), όπως θα δούμε στην υποενότητα 4.4.1. Παράλληλα, το σύστημα διαθέτει τα κύρια συστατικά του γενικού μοντέλου focused crawler, βλ. Σχήμα 3.1· δηλαδή το μέτωπο αναζήτησης (Crawl Frontier), τον εκτιμητή συνάφειας (Relevance Calculator) καθώς και τον εκτιμητή των τιμών προτεραιότητας των URLs (URL Scorer). Ταυτόχρονα, παίρνει ως είσοδο μερικά πολύ σχετικά κείμενα (seeds) που του δίνει ο χρήστης.

Όταν πλέον ο μαθητής έχει εκπαιδευτεί επαρκώς πάνω στο παραπάνω πρόβλημα ταξινόμησης, τότε ξεκινάει το δεύτερο στάδιο του focused crawler. Εδώ, εφόσον πλέον ο μαθητής έχει μάθει να αναγνωρίζει τι είναι σχετικό και ενδιαφέρον για τον ερευνητή, τώρα εκπαιδεύεται για να ψάχνει και να φέρνει τα πιο σχετικά ή πιο σημαντικά για την αναζήτηση του κείμενα. Για την διαδικασία αυτή θεωρούμε τον μαθητή ως έναν ευφυή πράκτορα που φέρνει κάθε χρονική στιγμή την ιστοσελίδα εκείνη που θεωρεί πως είναι η τρέχουσα καλύτερη του επιλογή, μακροχρόνια, από όλες τις διαθέσιμες ιστοσελίδες που βρίσκονται στο μέτωπο αναζήτησης. Προτείνουμε την εκπαίδευση του ευφύη πράκτορα-μαθητή με την ανάπτυξη

αλγορίθμων Ενισχυτικής Μάθησης (**Reinforcement Learning**), θεωρώντας ότι ο πράκτορας “ζει” σε ένα σε περιβάλλον Markov Decision Process (MDP), όπως θα περιγράψουμε αναλυτικά στην υποενότητα 4.4.2.

Αξίζει να σημειώσουμε πως μπορούμε να ακολουθήσουμε δύο μεθοδολογίες για την αλληλεπίδραση και συνεργασία των δύο σταδίων: (α) η εκπαίδευση των δυο σταδίων να γίνει αυστηρώς σειριακά, δηλαδή με την ολοκλήρωση του πρώτου να αρχίσει το δεύτερο, (β) το πρώτο στάδιο να εξελίσσεται μαζί με το δεύτερο· δηλαδή ο μαθητής να μαθαίνει ταυτόχρονα το πρόβλημα ταξινόμησης μαζί με το πρόβλημα εύρεσης της καλύτερης λύσης. Εμείς ακολουθούμε την (α) μεθοδολογία, δεδομένου πως έχουμε περιορισμένο αριθμό ιστοσελίδων που πρέπει να εξάγουμε, καθώς και περιορισμένες ερωτοαπαντήσεις με τον ερευνητή. Έτσι, ακολουθώντας την μεθοδολογία αυτή, ουσιαστικά χρησιμοποιούμε την Ενεργό Μάθηση

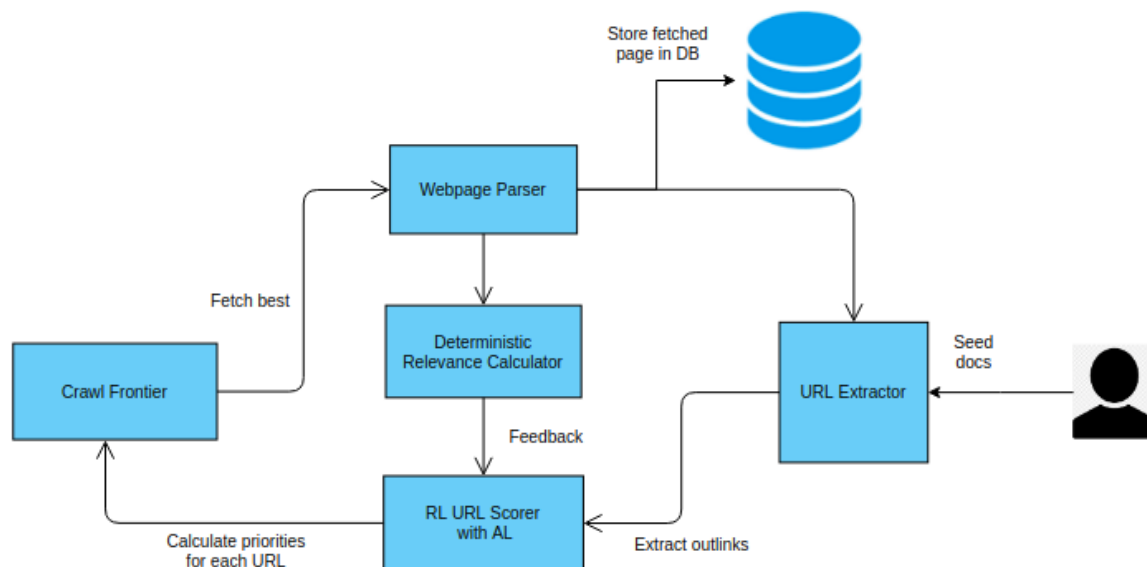


Σχήμα 4.1: *Focused Crawler Δύο Σταδίων*

(Active Learning) για τον εκτιμητή συνάφειας (AL Relevance Calculator) και την Ενισχυτική Μάθηση (Reinforcement Learning για τον εκτιμητή των τιμών προτεραιότητας των URLs (RL URL Scorer), όπως φαίνεται και στο Σχήμα 4.1. Ωστόσο, σε πιο ειδικά προβλήματα αναζήτησης που είναι σημαντική η ύπαρξη ταξινόμησης των λέξεων-κλειδιών για τη διαπίστωση της συνάφειας, χρησιμοποιούμε και τα δύο στάδια στον URL Scorer, ενώ ο Relevance Calculator χρησιμοποιεί νετερμινιστικά κριτήρια απόφασης, όπως φαίνεται στο Σχήμα 4.2.

4.1.1 Πρώτο Στάδιο Focused Crawler: Ενεργός Μάθηση

Στην υποενότητα αυτή, θα περιγράψουμε λεπτομερώς το πρώτο στάδιο εκπαίδευσης του συστήματος focused crawler. Δεδομένου πως τα δύο στάδια του συστήματος θα εκτελούνται, και στις δύο περιπτώσεις του focused crawler, αυστηρώς σειριακά, η εκπαίδευση του μαθητή για το πρόβλημα της ταξινόμησης με Ενεργό Μάθηση (Active Learning) απαιτεί ένα σύνολο δεδομένων βιβλιογραφικών αναφορών εκ των προτέρων (a priori). Αυτό συμβαίνει, καθώς η βελτιστοποίηση του μαθητή για την εύρεση σχετικών κειμένων προϋποθέτει την ικανότητα



Σχήμα 4.2: Εναλλακτικός Focused Crawler Δύο Σταδίων

του να αναγνωρίζει τι είναι σχετικό. Έτσι, έχουμε να αντιμετωπίσουμε ένα πρόβλημα Ενεργού Μάθησης βασισμένο σε συλλογή δεδομένων (**pool-based Active Learning**).

Το σύστημα μας αρχικά διαθέτει μόνο τις αρχικές seeds ιστοσελίδες, που έδωσε ο ερευνητής ως είσοδο, οι οποίες γνωρίζουμε πως αποτελούν πολύ σχετικά δεδομένα. Προκειμένου να δημιουργήσουμε τη συλλογή δεδομένων (pool), χρειάζεται να εκτελέσουμε ένα - ή/και περισσότερα - **τυχαίο πιλοτικό τρέξιμο** (Random Crawling), βλ. Αλγόριθμος 4.1, χωρίς παραμέτρους και χωρίς κάποιο μοντέλο μηχανικής μάθησης, πέρα από τις seeds ιστοσελίδες. Με αυτόν τον τρόπο, μπορούμε να φέρουμε έναν χρονικά εφικτό αριθμό URLs, τα οποία αποθηκεύουμε στο pool. Λαμβάνοντας υπόψιν πως ένας τυχαίος μηχανισμός εύρεσης ιστοσελίδων πιθανόν να οδηγήσει στο γεγονός πως το pool θα αποτελείται κυρίως από μη σχετικές ιστοσελίδες, μπορούμε να αποθηκεύουμε σε αυτό επίσης τα URLs, που εξάγονται και περιέχονται από τα seeds URLs. Τα URLs που συλλέγουμε με τον τελευταίο τρόπο αναμένουμε να αποτελούν κατά πλειοψηφία σχετικές ιστοσελίδες, υποθέτοντας πως μία ιστοσελίδα είναι πιθανό να συνδέεται με ιστοσελίδες παρόμοιου περιεχομένου στο γράφο του Παγκόσμιου Ιστού. Κατά αυτόν τον τρόπο, εξισορροπούμε σε ένα βαθμό την αναλογία των σχετικών και μη σχετικών δεδομένων στο pool, με αποτέλεσμα να είμαστε σε θέση να ξεκινήσουμε την βασική λειτουργία του πρώτου σταδίου· δηλαδή την εκπαίδευση του μαθητή στο πρόβλημα της ταξινόμησης με Ενεργό Μάθηση.

Έτσι, έχοντας πλέον συλλέξει έναν επαρκή αριθμό δεδομένων που αποτελούν το pool, θα περιγράψουμε τους αλγορίθμους Ενεργού Μάθησης (Active Learning) που θα χρησιμοποιήσουμε για να εκπαιδεύσουμε τον μαθητή. Τονίζουμε πως ο κύριος λόγος που χρησιμοποιούμε Ενεργό Μάθηση είναι πως δε διαθέτουμε επισημειώσεις (annotations) στα δεδομένα του pool και, επιπλέον, χρειάζεται να καταβάλλουμε την ελάχιστη προσπάθεια στη διαδικασία των ερωταποκρίσεων με τον ερευνητή. Αυτό σημαίνει πως ο μαθητής είναι αναγκαίο να κάνει όσο δυνατόν λιγότερες ερωτήσεις για επισημειώσεις των ιστοσελίδων του pool στον ερευνητή. Ωστόσο, σημειώνουμε και πάλι πως παρά τον περιορισμένο αριθμό επισημειωμένων δεδο-

μένων, που θα λάβουμε, θέλουμε το σύστημα να είναι σε θέση να ταξινομεί αποτελεσματικά τα URLs ανάλογα με την συνάφειά τους.

Όπως γνωρίζουμε, με την Ενεργό Μάθηση επιτυγχάνεται ακριβώς αυτό το ζητούμενο. Ένα μοντέλο, που εκπαιδεύεται με Ενεργό Μάθηση, αρχικά έχει πολύ λίγα δεδομένα εκπαίδευσης και σε κάθε γύρο ρωτάει ένα συγκεκριμένο αριθμό επισημειώσεων, αυξάνοντας έτσι συνεχώς τα δεδομένα του συνόλου εκπαίδευσης. Ωστόσο, με την Ενεργό Μάθηση φθάνουμε γρήγορα σε υψηλή επίδοση, ρωτώντας κάθε φορά, όχι τυχαία δείγματα από το pool, αλλά εκείνα που το μοντέλο είτε πιστεύει πως θα του παρέχουν την περισσότερη πληροφορία, είτε είναι πιο αβεθαίο για τις αληθινές επισημειώσεις. Σημειώνουμε πως σε όλες τις μεθόδους που θα μελετήσουμε, ο αριθμός των ερωτοαποκρίσεων για επισημειώσεις θα είναι σταθερός.

Θα εξετάσουμε τις εξής στρατηγικές ερωτήσεων με Ενεργό Μάθηση: (α) Τυχαία Δειγματοληψία (Random Sampling), (β) Δειγματοληψία με Αβεβαιότητα (Uncertainty Sampling) με Περιθώρια Δειγματοληψία (Margin Sampling), (γ) Δειγματοληψία με Αβεβαιότητα (Uncertainty Sampling) με Softmax. Στα (α), (β) θα εξετάσουμε Μηχανές Διανυσματικής Υποστήριξης (Support Vector Machines - SVM) που ακόμη και σήμερα, παρά την πρόοδο της Βαθιάς Μάθησης (Deep Learning), θεωρούνται αρκετά αξιόπιστη μέθοδος για Ενεργό Μάθηση και ερευνάται αρκετά η χρήση τους [66]. Στην (γ) περίπτωση, θα εξετάσουμε κατά πόσο αποτελεσματικά μπορεί να συνδυαστεί η Ενεργός Μάθηση με την Βαθιά Μάθηση στο πρόβλημά μας, κάνοντας χρήση τεχνητών νευρωνικών δικτύων πολυεπίπεδων Perceptron (MLP), καθώς και αναδρομικών νευρωνικών δικτύων (RNN), όπως Long short-term memory (LSTM).

Ο focused crawler μας εξετάζει το επίπεδο συνάφειας μίας βιβλιογραφικής δημοσίευσης στο Διαδίκτυο, αξιοποιώντας μόνο την περίληψη (abstract) αυτής. Και για τις τρεις μεθόδους Ενεργού Μάθησης που θα μελετήσουμε, τονίζουμε πως έχουμε να αντιμετωπίσουμε το ίδιο πρόβλημα ταξινόμησης, με κλάσεις τα επίπεδα συνάφειας του προβλήματος αναζήτησης. Επιπλέον, θα χρησιμοποιήσουμε word2vec word embeddings, τα οποία θα επεξεργαστούμε κατάλληλα σε κάθε περίπτωση για την αναπαράσταση των κειμένων. Θεωρούμε ότι τα word embeddings που θα χρησιμοποιήσουμε, θα είναι προεκπαιδευμένα πάνω στο πρόβλημά μας. Έτσι, δε θα χρειαστεί να τα εκπαιδεύσουμε περαιτέρω. Στις υποενότητες που ακολουθούν, θα περιγράψουμε αναλυτικά τις διαφορετικές μεθόδους Ενεργού Μάθησης που θα μελετήσουμε μαζί με τις αντίστοιχες αναπαραστάσεις των κειμένων των δημοσιεύσεων.

4.1.1.1 Random Sampling

Στην υποενότητα αυτή, θα περιγράψουμε την βασική (baseline) μέθοδο που θα μελετήσουμε, δηλαδή την Ενεργό Μάθηση με Τυχαία Δειγματοληψία ερωτήσεων. Θα χρησιμοποιήσουμε ένα μοντέλο Μηχανής Διανυσματικής Υποστήριξης (SVM), το οποίο πρώτα θα εκπαιδευτεί στο αρχικό μικρό σύνολο δεδομένων εκπαίδευσης που διαθέτουμε. Στη συνέχεια, σε κάθε γύρο της Ενεργού Μάθησης, θα ρωτάει τον ερευνητή ένα συγκεκριμένο αριθμό (k) δεδομένων από το pool, επιλέγοντάς τα με τυχαίο τρόπο. Τα δεδομένα αυτά μαζί με τις αντίστοιχες επισημειώσεις που θα λάβει, θα προσθέτονται στο σύνολο εκπαίδευσης και το μοντέλο θα επανεκπαιδεύεται. Το SVM μοντέλο που θα μελετήσουμε με Τυχαία Δειγματοληψία θα το αποκαλούμε από εδώ και πέρα *Random SVM*.

ΑΛΓΟΡΙΘΜΟΣ 4.1: Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο

Input: *seeds* URLs, *TIMESTEPS*
 Initialize empty crawl frontier *F* and closure *C*
 Initialize empty *buffer*
while *seeds* not empty **do**
 Pop a URL *url* from *seeds*
 Add *url* to *C*
 Fetch webpage *w url*
 L := Extract all outlinks URLs from *w* that are not in *C*
 for *l* in *L* **do**
 Add *l* to *F*
 end for
end while
timestep := 1
while not *timestep* > *TIMESTEPS* **do**
 Select a *url* from *F* at random
 Add *url* to *buffer*
 Add *url* to *C*
 Delete all records from *F* that have as URL the *url*
 L' := Extract all outlinks from *url* that are not in *C*
 for *l* in *L'* **do**
 Add *l* to *F*
 end for
 timestep := *timestep* + 1
end while
Output: *buffer*

Τυπικά, κάθε σημείο \mathbf{x} του pool έχει πιθανότητα να επιλέχθει για επισημείωση ίση με

$$p(\mathbf{x}) = \frac{1}{\|\text{pool}\|} \quad (4.1)$$

όπου $\|\text{pool}\|$ είναι το πλήθος των δεδομένων που βρίσκονται μέσα στο pool. Σημειώνουμε πως ο αριθμός αυτός προφανώς θα μειώνεται, καθώς παίρνουμε τις αντίστοιχες επισημειώσεις από τον ερευνητή, και άρα τα δεδομένα αυτά θα μεταφέρονται στο σύνολο εκπαίδευσης.

Όσον αφορά το πιο τεχνικό μέρος, το Random SVM θα λαμβάνει ως είσοδο την περιληψη (abstract) της εκάστοτε δημοσίευσης με την μορφή των word embeddings. Για κάθε κείμενο θα χρησιμοποιούμε ένα max pooling layer και ένα mean pooling layer των word embeddings, οι έξοδοι των οποίων θα είναι ενωμένες σε ένα διάνυσμα (vector), που θα αποτελεί την είσοδο του SVM.

4.1.1.2 Uncertainty Sampling με Margin Sampling

Μια άλλη μεθοδολογία που θα δοκιμάσουμε αφορά την Περιθώρια Δειγματοληψία με την εκπαίδευση και πάλι ενός SVM μοντέλου. Το μοντέλο αυτό θα το αποκαλούμε με τον όρο *Margin SVM* από εδώ και στο εξής. Η είσοδος του μοντέλου θα είναι ακριβώς ίδια με εκείνη του Random SVM, έχοντας και πάλι ενιαίο διάνυσμα (vector) για τις ενωμένες εξόδους

των max pooling layer και mean pooling layer των word embeddings. Ωστόσο, η μέθοδος δειγματοληψίας πλέον δεν θα στηρίζεται στην τυχαιότητα.

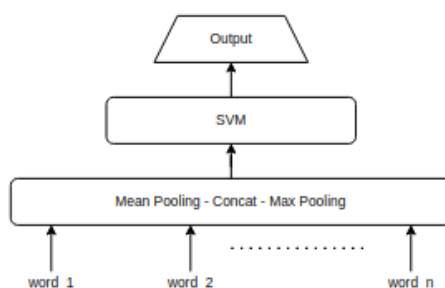
Τυπικά, συμβολίζουμε με \mathbf{X}_{train} το σύνολο δεδομένων εκπαίδευσης και με \mathbf{y}_{train} τις αντίστοιχες επισημειώσεις τους. Ακόμα, συμβολίζουμε με \mathbf{X}_{pool} και \mathbf{y}_{pool} τα αντίστοιχα δεδομένα και επισημειώσεις του pool. Δεδομένων k ερωτοαποκρίσεων ανά γύρο Ενεργού Μάθησης, ο Margin SVM ρωτάει σε κάθε γύρο τα k δεδομένα του pool που έχουν την μικρότερη περιθώρια απόσταση. Στην περίπτωση ενός SVM με δυαδική ταξινόμηση (binary classification), η περιθώρια απόσταση μεταφράζεται ως η απόσταση ενός σημείου στον χώρο από το κατώθλι απόφασης (decision boundary). Έτσι, η περιθώρια δειγματοληψία ρωτάει για επισημειώσεις τα k δεδομένα του pool με

$$\operatorname{argmin}_{\mathbf{x}} d(\mathbf{x}) \quad (4.2)$$

όπου $d(\mathbf{x})$, η απόσταση ενός σημείο \mathbf{x} του pool από το περιθώριο απόφασης (decision boundary). Από την άλλη, στην περίπτωση περισσότερων κλάσεων, ρωτάμε σε κάθε γύρο τα k δεδομένα του pool με

$$\operatorname{argmin}_{\mathbf{x}} p_{y_1}(\mathbf{x}) - p_{y_2}(\mathbf{x}) \quad (4.3)$$

όπου το y_1 είναι η πιο πιθανή κλάση, η y_2 είναι η δεύτερη πιο πιθανή κλάση και το p δηλώνει την αντίστοιχη πιθανότητα. Στο σχήμα 4.3, αναπαριστούμε το μοντέλο SVM που θα χρησιμοποιήσουμε.



Σχήμα 4.3: *Random/Margin SVM*

4.1.1.3 Uncertainty Sampling με Softmax

Ωστόσο, αν και το SVM είναι πολύ χρήσιμο σε πάρα πολλές περιπτώσεις ταξινόμησης με χρήση Ενεργού Μάθησης, είναι αρκετά ευάλωτο σε περιπτώσεις απομακρυσμένων σημείων (outliers). Παράλληλα, συχνά, αδυνατεί να κατανοήσει την επιλογή χαρακτηριστικών (feature selection) [67]. Έτσι, θα δοκιμάσουμε και μεθόδους Βαθιάς Μάθησης σε συνδυασμό με Ενεργό Μάθηση.

Για όλα τα μοντέλα που θα μελετήσουμε, θα χρησιμοποιήσουμε ως δειγματοληπτική μέθοδο την Softmax, μέσω της οποίας μπορούμε να κανονικοποιούμε με την μορφή πιθανότητας τις εξόδους ενός νευρωνικού δικτύου. Θυμίζουμε, η συνάρτηση της Softmax είναι η

παρακάτω

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^M e^{x_j}} \quad (4.4)$$

όπου \mathbf{x} είναι ένα διάνυσμα μήκους M . Έτσι, τώρα σε κάθε γύρου Ενεργού Μάθησης θα ρωτάμε τα k δεδομένα του pool με

$$\underset{\mathbf{x}}{\operatorname{argmin}} \max_i \text{Softmax}(\mathbf{x})_i \quad (4.5)$$

δηλαδή επιλέγοντας κάθε φορά εκείνο το σημείο \mathbf{x} του pool για το οποίο είναι μικρότερη η Softmax πιθανότητα για την πιο πιθανή του κλάση.

Στο σημείο αυτό θα περιγράψουμε τα διαφορετικά μοντέλα που θα μελετήσουμε με την παραπάνω μεθοδολογία δειγματοληψίας της Ενεργού Μάθησης. Θα δοκιμάσουμε το LSTM, το οποίο θα έχει ένα LSTM παίρνει ως είσοδο τα word embeddings του κειμένου ως χρονοσειρά και θα εξάγει την κρυφή κατάσταση του τελευταίου χρονικού βήματος (last timestep). Η έξοδος αυτή θα αποτελεί είσοδο ενός MLP, που θα προβλέπει στην έξοδο την πιο πιθανή κλάση. Το AverageLSTM θα έχει ίδια αρχιτεκτονική με το προηγούμενο, με την διαφορά πως το LSTM δεν θα δίνει το last timestep για είσοδο στο MLP, αλλά θα μεσολαβεί ένα mean pooling layer πάνω στις εξόδους όλων των timesteps. Επιπλέον, το DoubleLSTM θα είναι όμοιο με το AverageLSTM, με την διαφορά ότι θα υπάρχει και δεύτερο LSTM σε σειρά μετά από το πρώτο. Μία τέτοια μεθοδολογία έδειξε ότι μπορεί να έχει καλύτερα αποτελέσματα σε προβλήματα ταξινόμησης κειμένων με Ενεργό Μάθηση από το SVM [67]. Τέλος, το MeanMaxMLP θα αποτελείται από ένα MLP που θα έχει ως είσοδο την είσοδο που χρησιμοποιήσαμε και στα Random SVM και Margin SVM. Σημειώνουμε πως ανάμεσα στα διάφορα layers θα χρησιμοποιούμε dropout layers, προκειμένου να προσθέτουμε θόρυβο κατά τη διάρκεια της εκπαίδευσης και να αποφεύγουμε την υπερεκπαίδευση (overfitting). Για όλα τα παραπάνω μοντέλα θα χρησιμοποιήσουμε ως συνάρτηση κόστους την cross entropy, η οποία δίνεται από την σχέση :

$$L_{CE} = - \sum_{i=0}^K t_i \log p(i) \quad (4.6)$$

όπου το K είναι αριθμός των διαφορετικών κλάσεων της ταξινόμησης, το t_i είναι η αληθινή τιμή της κλάσης και λαμβάνει τιμές 0 και 1, και το $p(i)$ είναι η Softmax πιθανότητα για την i -οστή κλάση. Ως συναρτήσεις ενεργοποίησης (activation function) των νευρώνων των MLP, χρησιμοποιούμε την ReLU που, θυμίζουμε, ορίζεται ως εξής

$$\text{ReLU}(x) = \max(0, x) \quad (4.7)$$

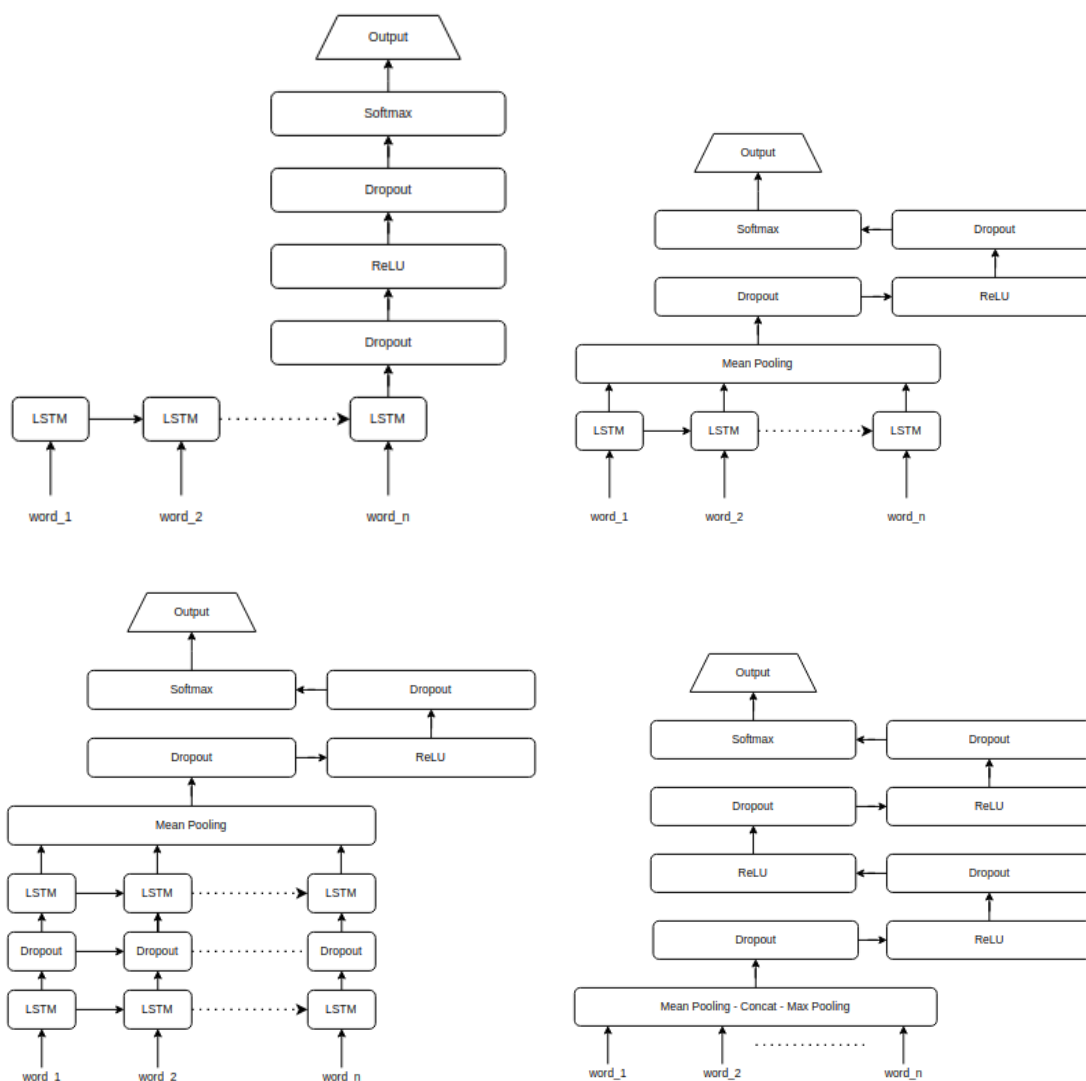
Στους νευρώνες τελευταίου επιπέδου, χρησιμοποιούμε την Softmax. Τα μοντέλα που περιγράψαμε απεικονίζονται στο Σχήμα 4.4.

Επιπλέον, σε περιπτώσεις όπου το σύνολο δεδομένων του pool είναι ανισόρροπο (imbalanced dataset), προτείνουμε την χρήση θετικών βαρών στην εκάστοτε συνάρτηση κόστους. Έτσι, "τιμωρούμε" περισσότερο τις περιπτώσεις, όπου το μοντέλο δεν προβλέπει σωστά κάποια

παρατήρηση, που ανήκει στη λιγότερο συχνή κλάση. Με αυτόν τον τρόπο, θέλουμε να ελαχιστοποιήσουμε την σταθμισμένη cross entropy που ορίζεται ως εξής:

$$L_{WCE} = - \sum_{i=1}^K w_i t_i \log p(i) \quad (4.8)$$

με $w_i > 0$ για $i = 1, 2, \dots, K$.



Σχήμα 4.4: Μοντέλα Ενεργού Μάθησης με χρήση Uncertainty Sampling με Softmax. Πάνω αριστερά: LSTM, Πάνω δεξιά: AverageLSTM, Κάτω αριστερά: DoubleLSTM, Κάτω δεξιά: MeanMaxMLP

4.1.2 Δεύτερο Στάδιο Focused Crawler: Ενισχυτική Μάθηση

Στην υποενότητα αυτή, θα μελετήσουμε το δεύτερο στάδιο του focused crawler μας, το οποίο στηρίζεται στην χρήση Ενισχυτικής Μάθησης (Reinforcement Learning). Διαισθητικά, στο στάδιο αυτό, σκοπό αποτελεί η εκμάθηση του μαθητή, ώστε αυτός να βρίσκει δημοσιεύσεις στο Διαδίκτυο, με αποτελεσματικό και αποδοτικό τρόπο, έχοντας πλέον κατανοήσει σε

βάθος τι ενδιαφέρει τον ερευνητή.

Για να πετύχουμε τον κύριο σκοπό του focused crawler - δηλαδή να επιστρέφουμε όσες περισσότερες σχετικές ιστοσελίδες - πρέπει να επιλέγουμε κάθε φορά εκείνο το URL που είναι το πιο υποσχόμενο, στο να οδηγήσει σε μία σχετική ιστοσελίδα. Οι focused crawlers γενικώς στηρίζονται στην τοπικότητα, υπό την έννοια ότι οι ιστοσελίδες είναι αρκετά πιθανόν να συνδέονται με ιστοσελίδες παρόμοιου περιεχομένου στον γράφο του Παγκόσμιου Ιστού. Μάλιστα, είναι πιθανόν γειτονικές ιστοσελίδες να παρουσιάζουν μεγάλη ομοιότητα ως προς το κείμενο τους [68]. Μολονότι μια ιστοσελίδα μπορεί να μην φαίνεται να είναι ιδιαίτερα σχετική, με το θέμα αναζήτησης, ωστόσο μπορεί να οδηγήσει σε άλλες σχετικές, μακροπρόθεσμα. Εάν ο focused crawler πράκτορας είναι σε θέση να κατανοήσει ένα τέτοιο URL, τότε ενδεχομένως να αξίζει να το επισκεφτεί. Από εδώ και στο εξής, θα αναφερόμαστε με τον όρο κέντρο (**hub** [49, 61]), όταν μία ιστοσελίδα, ανεξάρτητα με το αν είναι σχετική, αποτελεί κόμβο, ο οποίος συνδέεται με ακμή με κόμβους σχετικών ιστοσελίδων στο γράφο του Παγκόσμιου Ιστού. Επιπλέον, τονίζουμε πως είναι σημαντικό ο πράκτορας να εκτιμάει τα URLs του μετώπου αναζήτησης ως προς το πόσο οφέλιμα αυτά θα αποθούν μακροπρόθεσμα, καθόλη την διάρκεια του focused crawling.

Γράφουμε τις συναρτήσεις $Url: W \rightarrow U$ και $Website: U \rightarrow W$, όπου το W είναι το σύνολο των ιστοσελίδων του Παγκόσμιου Ιστού και U το σύνολο των αντίστοιχων URLs. Θεωρούμε ότι ο **γράφος του Παγκόσμιου Ιστού** είναι κατευθυνόμενος και έχει ως κόμβους ιστοσελίδες W . Θεωρούμε πως ένας κόμβος-ιστοσελίδα v_i , με αντίστοιχο URL το $u_i = Url(v_i)$, συνδέεται με τον v_j , με αντίστοιχο URL το $u_j = Url(v_j)$, αν και μόνο αν η ιστοσελίδα του v_i εξάγει το URL $Url(v_j)$, δηλαδή αν περιέχει σύνδεσμο προς το u_j , και συμβολίζουμε την αντίστοιχη ακμή με $u_i u_j$ ή $v_i v_j$. Και οι δύο συμβολισμοί είναι αποδεκτοί, αφού μια ιστοσελίδα περιγράφεται από μοναδικό URL και αντίστροφα.

Υποθέτουμε ότι ο focused crawler, όταν επισκέπτεται μια ιστοσελίδα, τότε επεκτείνει έναν κατευθυνόμενο υπογράφο του Παγκόσμιου Ιστού. Συμβολίζουμε με G τον κατευθυνόμενο γράφο του Παγκόσμιου Ιστού και με g_t τον **κατευθυνόμενο υπογράφο** του G που έχει διανύσει ο focused crawler, μέχρι και το χρονικό βήμα t . Για λόγους ευκολίας, στον υπογράφο g_t , όταν συμβολίζουμε έναν κόμβο-ιστοσελίδα v_i , θεωρούμε πως ο focused crawler επισκέφτηκε αυτήν την ιστοσελίδα το χρονικό βήμα t . Εναλλακτικά, την ίδια ιστοσελίδα μπορούμε να συμβολίσουμε με $w_i \in W$, δηλαδή $v_i \equiv w_i$.

Όταν ο focused crawler επισκέπτεται μία ιστοσελίδα w_{t+1} , με αντίστοιχο URL το $u_{t+1} = Url(w_{t+1})$, την χρονική στιγμή $t + 1$, τότε προκύπτει ο g_{t+1} επεκτείνοντας τον g_t κατά τον κόμβο v_{t+1} και ακμή $u_{t_k} u_{t+1}$ όπου το $u_{t_k} = Url(w_k)$ είναι το URL της ιστοσελίδας w_k , που επισκέφτηκε ο focused crawler την χρονική στιγμή t_k - με $0 < t_k < t + 1$ - και εξήγαγε το u_{t+1} .

Θεωρούμε τα εξής σύνολα: το C_t - ως το **κλειστό σύνολο** των URLs που έχει επισκεφτεί ο focused crawler μέχρι το χρονικό βήμα t - το V_t - ως το **σύνολο των κόμβων** ιστοσελίδων του g_t - και το E_t - ως το **σύνολο των κατευθυνόμενων ακμών** του g_t . Σημειώνουμε πως ένα URL $u \notin C_t$ μπορεί να έχει εξαχθεί από πολλές ιστοσελίδες του V_t . Ωστόσο, υποθέτουμε πως όταν ο focused crawler επισκέπτεται την ιστοσελίδα w_t του u_t , γνωρίζει και την ιστοσελίδα w_k με URL το u_k , όπου η εξαγωγή του u_t από την w ήταν και ο λόγος της επίσκεψης στην w_t . Με αυτό τον τρόπο, πέρα από τον κόμβο v_k , ο focused crawler γνωρίζει και ποια ακμή

να επεκτείνει· δηλαδή την $u_k u_t$. Στο σημείο αυτό, θα δώσουμε δύο χρήσιμους ορισμούς, που θα χρησιμοποιούμε αρκετά από εδώ και στο εξής.

Ορισμός 4.1. *Μεταξύ μιας ακμής $u_i u_j$ του υπογράφου g_t που συνδέει τα ιστοσελίδες $w_i = \text{Website}(u_i)$ και $w_j = \text{Website}(u_j)$, ορίζουμε ως **ιστοσελίδα γονέα** της w_j , την w_i , και συμβολίζουμε με $p(w_j) = w_i$.*

Ορισμός 4.2. *Ως **μονοπάτι** $\text{path}(t) = \text{path}_{t_0 \rightarrow t}$ της ιστοσελίδας w_t που επισκέπτεται ο *focused crawler* το χρονικό βήμα t , θεωρούμε το μέγιστο κατευθυνόμενο μονοπάτι του g_t , μήκους n , με κόμβους $v^{(1)}, v^{(2)}, \dots, v^{(n-1)}, v^{(n)} \in V_t$, ώστε $t_0 \leq t$, ο αρχικός κόμβος του μονοπατιού $v^{(1)} \equiv w_{t_0}$, ο τελικός κόμβος $v^{(n)} \equiv w_t$ και $p(v^{(n)}) = v^{(n-1)}, \dots, p(v^{(2)}) = v^{(1)}$.*

Εάν υποθέσουμε ότι ο *focused crawler* επισκέπτεται μία ιστοσελίδα το πολύ μία φορά, τότε άμεσα συμπεραίνουμε πως για κάθε ιστοσελίδα που επισκέπτεται υπάρχει μοναδικό μονοπάτι ιστοσελίδας.

Η Ενισχυτική Μάθηση (Reinforcement Learning) αναζητά μια βέλτιστη στρατηγική (policy), δεδομένων των καταστάσεων (states) και των δράσεων (actions), με στόχο να μεγιστοποιήσει τη συνολική συσσωρευτική επιβράβευση - ανταμοιβή. Σκοπεύουμε να χρησιμοποιήσουμε την Ενισχυτική Μάθηση, προκειμένου ο εκπαιδευόμενος πράκτορας να είναι σε θέση να εκτιμά τα προς επίσκεψη URLs, ανακαλύπτοντας, με αποδοτικό και αποτελεσματικό τρόπο, όσα περισσότερα σχετικά κείμενα. Ο πράκτορας αυτός ουσιαστικά αποτελεί τον URL Scorer του *focused crawler*. Στόχος του είναι να επιλέγει εκείνα τα URLs που θα του αποφέρουν το μέγιστο συσσωρευτικό reward. Σημειώνουμε πως ο *focused crawler* πράκτορας - μαθητής - δεν γνωρίζει a priori το πως αναπαριστάται ο γράφος του Παγκόσμιου Ιστού αλλά τον ανακαλύπτει καθόλη τη διάρκεια της εκτέλεσής του.

4.1.2.1 Ο Focused Crawler ως Markov Decision Process (MDP)

Στο σημείο αυτό, θα μοντελοποιήσουμε το περιβάλλον του *focused crawler* σε ένα MDP $\langle S, A, R, T \rangle$, έχοντας ως βάση μία παλαιότερη μελέτη [65]. Θεωρούμε S (states) τις ιστοσελίδες - που επισκέπτεται ο πράκτορας - και A (actions) τα URLs που εξάγονται από την εκάστοτε τρέχουσα προοπτελασμένη ιστοσελίδα. Όταν ο πράκτορας βρίσκεται σε μία ιστοσελίδα S_i και επιλέγει ένα URL, που εξήχθη από αυτή, έστω A_j , τότε συμβαίνει μία μετάβαση T_i , και ο πράκτορας επισκεφτείται την ιστοσελίδα του URL A_j , έστω S_{i+1} . Τότε, αυτός παρατηρεί ένα reward $r \in \mathbb{R}$, το οποίο αφορά τη συνάφεια της ιστοσελίδας. Έτσι, η συνάρτηση μετάβασης T αποτελεί την πιθανότητα μετάβασης από την τρέχουσα ιστοσελίδα προς μια άλλη, αν επιλεγεί ένα συγκεκριμένο URL που έχει εξαχθεί από αυτήν.

Ωστόσο, όπως έχουμε αναφέρει, ο *focused crawler* όταν επισκέπτεται μια ιστοσελίδα, δεν επιλέγει κατά ανάγκη στη συνέχεια να επισκεφτεί κάποιο URL που έχει εξαχθεί από αυτήν, αλλά εκείνο που έχει την μεγαλύτερη τιμή προτεραιότητας στο μέτωπο αναζήτησης (crawl frontier). Αυτό πρακτικά σημαίνει πως μπορεί να επιλέγει συνεχώς το πιο υποσχόμενο URL (action), ακόμα και χωρίς το τρέχον state του να το διαθέτει αυτό. Τονίζουμε πως εάν επιλέγαμε πάντοτε μεταβάσεις διαδοχικών states, τότε αυτό θα είχε ως αποτέλεσμα ο *focused crawler* να εξερευνούσε αποκλειστικά ένα μονοπάτι του γράφου και να έπεφε σε

τοπικά ακρότητα. Για αυτό και είναι πολύ σημαντικό το μέτωπο αναζήτησης για τον focused crawler [65].

Αξίζει να τονίσουμε ότι ο πράκτορας, όταν επιλέγει να επισκεφτεί ένα URL, μπορεί επίσης να λαμβάνει υπόψιν και πληροφορίες τόσο της ιστοσελίδας όσο και του μονοπατιού αυτής. Σημειώνουμε πως δεδομένου ότι θα χρησιμοποιήσουμε model-free Ενισχυτική Μάθηση [7] βασισμένη στο Q-learning, η συνάρτηση μετάβασης T δεν θα μας απασχολήσει άμεσα, αφού δεν θα γνωρίζουμε τη δυναμική (dynamics) του περιβάλλοντος. Με έναν Q-learning-τύπου αλγόριθμο, σε κάθε state ο πράκτορας θα υπολογίζει ένα Q-value για κάθε action.

Παίρνοντας υπόψιν την ανάγκη τόσο για μη διαδοχικές μεταβάσεις των states όσο και για αξιοποίηση πληροφορίας προηγούμενων ιστοσελίδων, που έχει επισκεφτεί ο πράκτορας, πρέπει να τροποποιήσουμε την μοντελοποίηση του MDP που δώσαμε παραπάνω. Έτσι, θεωρούμε ότι το **state** s_t το χρονικό βήμα t είναι ο υπογράφος g_t που περιέχει όλα τα μονοπάτια ιστοσελίδων $path(t_i)$, για $0 < t_i \leq t$, τις οποίες έχει επισκεφτεί ο focused crawler πράκτορας μέχρι το χρονικό βήμα t . Σημειώνουμε πως το πλήθος των διαφορετικών states ταυτίζεται με το πλήθος των διαφορετικών κατευθυνόμενων υπογράφων του G , δεδομένων ωστόσο των αρχικών seeds που δίνει ο χρήστης.

Στο σημείο αυτό, θα γράψουμε φαρμαλιστικά το **μέτωπο αναζήτησης** (crawl frontier). Θεωρούμε μέτωπο αναζήτησης F_t το σύνολο που περιέχει όλα τα ζευγάρια της μορφής $(path(t_i), u^k)$, όπου $0 < t_i \leq t$, $1 < k \leq M_t$, με M_t το πλήθος των στοιχείων του F_t , $path(t_i)$ το μονοπάτι της ιστοσελίδας w_{t_i} , που ο πράκτορας επισκέφτηκε το χρονικό βήμα t_i , και u^k το URL που αντιστοιχεί σε μια ιστοσελίδα w^k που ο πράκτορας μπορεί να επισκεφτεί σε επόμενα χρονικά βήματα. Διαισθητικά, τα ζευγάρια του F_t αποτελούν τους πιθανούς νέους κόμβους και τις αντίστοιχες ακμές που μπορούν να προστεθούν στο g_t , λαμβάνοντας ότι ένας πιθανός νέος κόμβος είναι ο $Website(u^k)$ με αντίστοιχη ακμή την $Url(w_{t_i})u^k$.

Ως **actions** A_{s_t} του s_t λαμβάνουμε όλες τις διαθέσιμες ακμές της μορφής $\langle Url(w_{t_i})u^k \rangle$, που μπορούμε να επεκτείνουμε τον g_t σε g_{t+1} . Αυτό, ωστόσο, με την προϋπόθεση ότι $u^k \notin C_t$, για κάθε $k \in [1, M_t]$. Με την επιλογή ενός action a_t , ο πράκτορας επισκέπτεται μια νέα ιστοσελίδα w_{t+1} , επεκτείνοντας το μονοπάτι της ιστοσελίδας γονέα της, δηλαδή της $p(w_{t+1})$.

Ως **πιθανότητα μετάβασης**, από ένα state s' σε κάποιο άλλο s'' , θεωρούμε την πιθανότητα επιλογής του ζευγαριού στο F_t που δίνει, έμμεσα, εκείνη την ακμή, η οποία οδηγεί στον υπογράφο του s'' .

Όσον αφορά τη **reward function**, αυτή παραμένει η ίδια με την πρώτη μας προσέγγιση. Για σχετική ιστοσελίδα, το reward είναι ίσο με 1, ενώ 0 στην αντίθετη περίπτωση. Δηλαδή όταν ο πράκτορας μεταβαίνει από το s_t στο s_{t+1} , μέσω ενός action a_t που περιείχε την πληροφορία για το URL της ιστοσελίδας w_{t+1} , το περιβάλλον ενημερώνει τον πράκτορα σχετικά με την συνάφειά αυτής.

Σημειώνουμε πως όταν ο πράκτορας μεταβαίνει από το s_t στο s_{t+1} , μέσω του action a_t , εξάγει νέα actions, τα $a_{s_{t+1}}$ ακμές που οδηγούν σε κόμβους u^k με $Url(u^k) \notin C_t$. Θεωρώντας ότι με το a_t επέλεξε έναν κόμβο w_{t+1} , αυτός επανεκτιμά τα Q-values όλων των διαθέσιμων actions του $A_{s_{t+1}} = a_{s_{t+1}} \cup [A_{s_t} - DEL(A_{s_t})]$, όπου $DEL(A_{s_t}) = \{u^k w_{t+1}\}$ είναι το σύνολο που περιέχει όλες τις ακμές για κάθε $u^k \in V_t$. Για αυτό μας χρειάζεται και το μέτωπο αναζήτησης, προκειμένου πέρα από τα διαθέσιμα ζευγάρια, να διατηρεί και να υπολογίζει ξανά τα αντίστοιχα Q-values αυτών, ώστε σε λογαριθμικό χρόνο να εξάγει το καλύτερο ζευγάρι και

έμμεσα το πιο υποσχόμενο action.

Έτσι, **επεκτείνουμε τον ορισμό του μετώπου αναζήτησης F** , ώστε να αποτελεί ένα σύνολο με στοιχεία της μορφής $\langle (path(t_i), u^k), qvalue^{(k)} \rangle$, με το $qvalue^{(k)}$ να αποτελεί την τιμή του Q-value του κ-στού στοιχείου του F . Με αυτόν τον τρόπο, το κάθε ζευγάρι, που περιέχεται σε κάθε στοιχείο του μετώπου αναζήτησης, εκτός του ότι υποδηλώνει το αντίστοιχο action, περιέχει και πληροφορίες για την προτεραιότητα επιλογής του, μέσω της αντίστοιχης Q-value.

Όπως ήδη έχουμε δει, κάθε τέτοιο ζευγάρι περιέχει πληροφορίες για το URL u^k - που θα επισκεφτεί πιθανώς στο επόμενο βήμα ο πράκτορας - αλλά και πληροφορίες για το μονοπάτι $path(t_i)$ της ιστοσελίδας που εξήγαγε το u^k την χρονική στιγμή t_i . Θεωρώντας πως είμαστε στο χρονικό βήμα t , κάθε ζευγάρι περιέχει πληροφορίες, σχετικές τόσο με το τρέχον state g_t - όσο και με το αντίστοιχο action.

Όσον αφορά το action, ο τρόπος αναπαράστασης είναι αρκετά άμεσος, αφού σε αυτό περιέχεται η πληροφορία του URL του επόμενου πιθανού προς επίσκεψη κόμβου w_{t+1} . Όσον αφορά το state, ωστόσο, θεωρούμε πως από όλο τον υπογράφο χρειαζόμαστε μόνο εκείνους τους κόμβους $w_{t_j} \in path(t_i)$ για $t_j \leq t_i$. Διαισθητικά, μπορούμε να σκεφτούμε ότι για να μετρήσουμε την συνεισφορά του τρέχοντος state g_t πάνω σε ένα ζευγάρι ενός στοιχείου του μετώπου αναζήτησης, συσσωρεύουμε (aggregate) όλη την πληροφορία του g_t , θεωρώντας ότι οι κόμβοι που δεν ανήκουν στο μονοπάτι της ιστοσελίδας γονέα της προς επίσκεψη ιστοσελίδας, δεν συμμετέχουν καθόλου στην συνεισφορά αυτή.

Παρόλα αυτά, όπως αναφέραμε και προηγουμένως, τονίζουμε την δράση του κλειστού συνόλου (closure) C_t , με το οποίο τα actions A_{s_t} για το s_t , κατά το χρονικό βήμα t , δεν τα έχει ξαναεπιλέξει ποτέ προηγουμένως ο focused crawler πράκτορας. Διαισθητικά, αυτό σημαίνει πως ο πράκτορας μπορεί να επισκεφτεί μια ιστοσελίδα το πολύ μία φορά, καταχωρώνοντας ότι δεν έχει νόημα στο focused crawling να επισκεφτόμαστε ξανά ίδιες ιστοσελίδες [65]. Από αυτό απορρέει πως ο g_t πάντοτε θα είναι Ακυκλικός Κατευθυνόμενος Γράφος (DAG).

Ακόμα, αναφέρουμε ότι ο επαναυπολογισμός των Q-values των στοιχείων του μετώπου αναζήτησης που αντιστοιχούν στα παρελθοντικά actions, στα πλαίσια του focused crawler μεταφράζεται διαισθητικά στο εξής. Όταν επισκεφτόμαστε μία νέα ιστοσελίδα, υπολογίζονται οι εκτιμήσεις για το πόσο υποσχόμενα είναι τα άμεσα εξαγόμενα URLs, τα οποία αποθηκεύονται στο μέτωπο αναζήτησης, αφού προηγουμένως ο πράκτορας έχει ενημερώσει τις αντίστοιχες τιμές για τα διαθέσιμα, πλέον, URLs που βρίσκονταν από πριν εκεί. Μία τέτοια ενημέρωση του μετώπου αναζήτησης λέγεται **σύγχρονη (synchronous update)** [65]. Έτσι, η σύγχρονη ενημέρωση περιλαμβάνει τον υπολογισμό των Q-values για τα $A_{s_{t+1}} = a_{s_{t+1}} \cup [A_{s_t} - DEL(A_{s_t})]$ για τα οποία μιλήσαμε παραπάνω.

Αξίζει να σημειώσουμε πως το αρχικό state θα είναι ο αρχικός υπογράφος g_0 , που δεδομένων των seeds ιστοσελίδων του focused crawler, δεν είναι ασθενώς-συνεκτικός (weakly-connected graph). Δηλαδή, θεωρούμε πως πρώτου ξεκινήσει το πρώτο χρονικό βήμα, το αρχικό state $s_0 - g_0$ - αποτελούνταν μόνο από τους $|S|$ κόμβους των seeds ιστοσελίδων, όπου $|S|$ ο αριθμός αυτών. Παράλληλα, δεδομένου πως ο πράκτορας πάντοτε επιλέγει actions που αντιστοιχίζονται σε URLs ιστοσελίδων, που δεν έχει επισκεφτεί ξανά σε παρελθοντικά χρονικά βήματα, ο υπογράφος των states του μονίμως δεν θα είναι ασθενώς-συνεκτικός.

Τέλος, τονίζουμε πως το E_0 είναι το κενό σύνολο. Από την αλλαγή, τερματικό state δεν

ορίζεται, αφού η όλη διαδικασία είναι συνεχής, χωρίς να υπάρχουν δηλαδή επεισόδια - non episodic - ενώ ολοκληρώνεται όταν ο πράκτορας έχει επισκεφτεί ένα συγκεκριμένο αριθμό ιστοσελίδων. Άρα, εάν υποθέσουμε ότι ο πράκτορας πρέπει να επισκεφτεί T διαφορετικές ιστοσελίδες, τότε το τελευταίο state θα είναι ο υπογράφο g_T .

Συμπερασματικά, η δική μας μοντελοποίηση του MDP για το περιβάλλον του focused crawler συνοπτικά είναι η εξής: Ως state s_t ορίζουμε τον κατευθυνόμενο υπογράφο g_t του χρονικού βήματος t , κατά το οποίο ο πράκτορας επισκέφτηκε την ιστοσελίδα w_t . Ως action a_t , ορίζουμε την επιλογή εκείνης της ακμής - έμμεσα - από το μέτωπο αναζήτησης, με το οποίο θα μεταβούμε από το s_t στο state s_{t+1} και θα επισκεφτούμε την ιστοσελίδα w_{t+1} . Τα διαθέσιμα actions $A_{s_{t+1}}$ για το s_{t+1} αποτελούνται από τις ακμές που αντιστοιχίζονται στα URLs που εξάγονται από την w_{t+1} , τα οποία δεν τα έχει ξαναεπισκεφτεί ο πράκτορας, και τα συμβολίζουμε με $a_{s_{t+1}}$, *συυ* τα actions του s_t , χωρίς ωστόσο εκείνα που αντιστοιχίζονται σε URLs ιστοσελίδων που έχει ξαναεπισκεφτεί ο πράκτορας, δηλαδή τα $A_{s_t} - DEL(A_{s_t})$: όπου $DEL(A_{s_t}) = \{u^k w_{t+1}\}$ είναι το σύνολο που περιέχει όλες τις ακμές για κάθε $u^k \in V_t$. Ως reward R_{t+1} ορίζουμε την συνάφεια της ιστοσελίδας w_{t+1} με το θέμα αναζήτησης, που ενδέχεται να παρέχεται από έναν Oracle. Τέλος, λαμβάνουμε ως πιθανότητα μετάβασης από το state s^r προς το state s^r' για το χρονικό βήμα $t + 1$ την πιθανότητα επιλογής εκείνου του ζευγαριού από το μέτωπο αναζήτησης F , που αντιστοιχεί στην ακμή του γράφου G για την επέκταση του υπογράφου g_t στον υπογράφο g_{t+1} .

4.1.2.2 Επιβράβευση - The Reward function

Ως reward θεωρούμε πως θα είναι 1 εάν η ιστοσελίδα του νέου τρέχοντος state είναι σχετική αλλιώς 0. Άρα, έχουμε την εξής συνάρτηση (reward function):

$$R(\text{webpage}) = \begin{cases} 1, & \text{if webpage is relevant} \\ 0, & \text{if webpage is irrelevant} \end{cases}$$

Προκειμένου να καταλάβουμε, εάν μια ιστοσελίδα είναι όντως σχετική, προτείνουμε τη χρήση της εκμάθησης του θέματος από τον μαθητή που συνέβη στο πρώτο στάδιο του focused crawler. Με άλλα λόγια, την reward function θα αποτελεί το μοντέλο Ενεργού Μάθησης - που εκπαιδεύτηκε στο προηγούμενο στάδιο - το οποίο θα λαμβάνει στην είσοδο τα word embeddings της περίληψης (abstract) της δημοσίευσης. Έτσι, η reward function της Ε-νοσχυτικής Μάθησης είναι ο AL Relevance Calculator του focused crawler, βλ. Σχήμα 4.1. Ωστόσο, σε ιδιαίτερες περιπτώσεις αναζήτησης, όπως θα μελετήσουμε και παρακάτω, προτείνουμε διαφορετική χρήση του Relevance Calculator, βλ. Σχήμα 4.2, με την reward function ωστόσο να παραμένει η ίδια.

4.1.2.3 Κοινές Αναπαραστάσεις state και action

Αφού έχουμε μοντελοποιήσει το πρόβλημά μας ως MDP, θα αναφερθούμε σχετικά με την αναπαράσταση του state και του action. Δεδομένου πως ένα URL, προκειμένου να εισχωρηθεί στο μέτωπο αναζήτησης, πρέπει πρωτίστως να εκτιμηθεί ως προς την προτεραιότητα επίσκεψής του, δεδομένης και της ιστοσελίδας που το εξήγαγε, είναι ανάγκη η διαδικασία

αυτή να γίνεται ξεχωριστά για κάθε URL που εξάγεται από αυτήν.

Όπως αναφέραμε και στην προηγούμενη υποενότητα, ενά ζευγάρι ενός στοιχείου του μετώπου αναζήτησης περιέχει πληροφορίες σχετικές τόσο με το τρέχον state - g_t - όσο και με το αντίστοιχο action για ένα συγκεκριμένο χρονικό βήμα t . Μάλιστα έχουμε σημειώσει πως το state συσσωρεύει μόνο την πληροφορία που περιέχεται στους κόμβους που ανήκουν μέσα στο μονοπάτι της ιστοσελίδας - που εξάγει το αντίστοιχο URL - και όχι όλους τους κόμβους του. Αυτό σημαίνει πως για κάθε state πρέπει να εκτιμάται ξεχωριστά κάθε action του.

Διαισθητικά ένα URL χρειάζεται την ιστοσελίδα γονέα του και πιθανώς και ολόκληρο το μονοπάτι αυτής, ώστε να εκτιμηθεί το πόσο υποσχόμενο είναι, για να το επισκεφτεί ο focused crawler πράκτορας. Αυτό μας οδηγεί στο να έχουμε κοινή αναπαράσταση μεταξύ της συνεισφοράς του state και του επιλεγμένου action στο πρόβλημά μας. Έλλωστε μια παρόμοια μέθοδος έχει ξαναχρησιμοποιηθεί σε ένα focused crawler σύστημα [65]. Έτσι, δεδομένου ενός state s και ενός διαθέσιμου action a , ορίζουμε την κοινή τους αναπαράσταση $\mathbf{x}(s, a)$. Σημειώνουμε ότι το $\mathbf{x}(s, a)$ δεν σχετίζεται άμεσα με τους ορισμούς του state και του action που δώσαμε, αλλά αποτελεί το διάνυσμα της κοινής τους αναπαράστασης που θα χρησιμοποιήσουμε για να λύσουμε το πρόβλημα.

Θεωρητικά, το MDP μας είναι πεπερασμένο, εάν τα χαρακτηριστικά της κοινής αναπαράστασης είναι διακριτά. Σε αντίθετη περίπτωση, τυπικά είναι μη πεπερασμένο, δεδομένων των συνεχών χώρων των states και actions. Πρακτικά, όμως, γνωρίζουμε ότι υπάρχουν πεπερασμένου αριθμού διαφορετικοί κατευθυνόμενοι υπογράφοι g , αφού υπάρχουν πεπερασμένες διαφορετικές ιστοσελίδες. Έτσι, θεωρούμε ότι το MDP μας είναι πεπερασμένο, αλλά εξαιρετικά μεγάλο (large MDP).

Στο σημείο αυτό, θα περιγράψουμε αναλυτικά τα χαρακτηριστικά που αφορούν το state (S) και το action (A). Οι διαφορετικές αναπαραστάσεις αυτών που θα μελετήσουμε είναι οι εξής: (α) General Representation, (β) Keyword Representation και (γ) Only NLP Representation. Οι (α) και (γ) αναπαραστάσεις προτείνονται μόνο για το Σχήμα 4.1, ενώ η (β) κυρίως για το Σχήμα 4.2 του focused crawler. Σε ένα γενικό πλαίσιο, θα θέλαμε η αναπαράσταση του state να γίνει με τέτοιο τρόπο ώστε να περιέχει πληροφορίες για την ίδια την ιστοσελίδα, δηλαδή σχετικά με το περιεχόμενο της, αλλά και πληροφορίες σχετικά με το μονοπάτι που έχει ακολουθήσει ο focused crawler πράκτορας μέχρι αυτήν.

Εάν η αναπαράσταση του state παρέχει στοιχεία της συνάφειας των προγόνων της ιστοσελίδας, μακροχρόνια ο πράκτορας θα θέλαμε να αναγνωρίζει σχετικά μονοπάτια, και άρα τις ιστοσελίδες που αποτελούν hubs. Όσον αφορά τα actions, θα θέλαμε να περιέχουν πληροφορίες σχετικά με το περιεχόμενο των αντίστοιχων τίτλων των δημοσιεύσεων, προκειμένου ο πράκτορας να είναι σε θέση να ανιχνεύει πιθανώς σχετικούς τίτλους με το θέμα αναζήτησης. Σημειώνουμε πως ο τίτλος μιας δημοσίευσης είναι διαθέσιμος - ως anchor text μαζί με το URL της - την στιγμή που αυτό εξάγεται από μια ιστοσελίδα, με αποτέλεσμα να μπορούμε να τον αξιοποιήσουμε για την αναπαράσταση του action.

Τονίζουμε πως και οι τρεις αναπαραστάσεις χρησιμοποιούν word embeddings για την αναπαράσταση του state ή/και του action, κάτι που έχει ξαναχρησιμοποιηθεί και σε άλλες μελέτες [69]. Ακόμα, οι (α), (β) συσσωρεύουν την πληροφορία για το state, παίρνοντας υπόψιν τους κόμβους που ανήκουν στο μονοπάτι της ιστοσελίδας w , που εξήγαγε το αντίστοιχο action. Από την άλλη, η (γ) χρησιμοποιεί από αυτό το μονοπάτι μόνο την ιστοσελίδα w .

Ξεκινάμε να περιγράψουμε την (α) *General Representation* αναπαράσταση. Για την αναπαράσταση του state θα πάρουμε τα εξής χαρακτηριστικά: Συνάφεια της w , που ταυτίζεται με το αντίστοιχο reward της, Αριθμός Σχετικών Κόμβων του Μονοπατιού και Αριθμός Μη Σχετικών Κόμβων του Μονοπατιού. Για το action θα χρησιμοποιήσουμε τα word embeddings του τίτλου του αντίστοιχου URL, και ειδικότερα το διάνυσμα του μέσου όρου τους. Με αυτόν τον τρόπο, βλέπουμε πως έχουμε εν δυνάμει άπειρο αριθμό διαφορετικών states και άπειρο αριθμό διαφορετικών actions, αφού τόσο τα χαρακτηριστικά του state όσο και τα word embeddings του action μπορούν να λάβουν μη πεπερασμένες συνεχείς τιμές.

Η αναπαράσταση (β) *Keyword Representation* απευθύνεται σε περιπτώσεις που έχουμε a priori γνώση για τις λέξεις-κλειδιά των κειμένων των δημοσιεύσεων του συγκεκριμένου προβλήματος αναζήτησης. Λαμβάνει το state με τα εξής χαρακτηριστικά: Συνάφεια της w , Ποσοστό Συνάφειας Κόμβων του Μονοπατιού και Απόσταση από τον Τελευταίο Σχετικό Κόμβο του Μονοπατιού. Ως Ποσοστό Συνάφειας Κόμβων του Μονοπατιού ορίζουμε το πηλίκο των σχετικών κόμβων του μονοπατιού προς όλους τους κόμβους του μονοπατιού. Ως Απόσταση από τον Τελευταίο Σχετικό Κόμβο του Μονοπατιού ορίζουμε το $\frac{1}{d}$, όπου d η απόσταση από τον τελευταίο σχετικό κόμβο του μονοπατιού συμπεριλαμβανομένου και του w . Το τελευταίο χαρακτηριστικό δίνει περισσότερες πληροφορίες για την ύπαρξη του hub, έχοντας υποθέσει πως όσο μεγαλώνει η απόσταση d τόσο λιγότερο πιθανόν είναι να βρισκόμαστε σε hub. Όσον αφορά το action λαμβάνουμε τα εξής: Εύρεση λέξεων-κλειδιών κατηγορίας-1, Εύρεση λέξεων-κλειδιών κατηγορίας-2, ..., Εύρεση λέξεων-κλειδιών κατηγορίας- K , όπου K το πλήθος των διαφορετικών κατηγοριών λέξεων κλειδιών, και Πιθανότητα Συνάφειας Τίτλου. Το καθένα από τα παραπάνω χαρακτηριστικά εύρεσης των λέξεων-κλειδιών είναι μία δυαδική μεταβλητές: δηλαδή θα λαμβάνει 1, εάν βρέθηκε κάποια λέξη-κλειδί της αντίστοιχης κατηγορίας, αλλιώς 0. Εδώ έχουμε υποθέσει πως οι λέξεις-κλειδιά ανήκουν σε διαφορετικές κατηγορίες, οι οποίες είναι όλες γνώστες. Για την Πιθανότητα Συνάφειας Τίτλου προτείνουμε την χρήση ενός νευρωνικού δικτύου που θα δέχεται ως είσοδο τα word embeddings του τίτλου και θα εξάγει σε μορφή πιθανότητας την πρόβλεψή του, για το εάν ο τίτλος αυτός είναι σχετικός.

Σε πιο δύσκολα προβλήματα αναζήτησης, που θα μελετήσουμε, ωστόσο, προτείνουμε τη διακριτοποίηση των χαρακτηριστικών. Για παράδειγμα, όταν η ύπαρξη συγκεκριμένων λέξεων-κλειδιών είναι απαραίτητη, προκειμένου να αναγνωριστούν οι ιστοσελίδες ως σχετικές από το Gold Standard, προτείνουμε τη διακριτοποίηση του Ποσοστού Συνάφειας Κόμβων του Μονοπατιού και της Απόστασης από τον Τελευταίο Σχετικό Κόμβο του Μονοπατιού της αναπαράστασης του state. Με αυτόν τον τρόπο, θέλουμε να γίνει πιο εύκολη η μάθηση του πράκτορα και να φθάσει πιο γρήγορα σε βέλτιστη στρατηγική (optimal policy). Μία τέτοια μέθοδος έχει ξαναχρησιμοποιηθεί άλλωστε στο focused crawling [65]. Έτσι, πετυχαίνουμε να έχουμε πεπερασμένο αριθμό διαφορετικών αναπαραστάσεων των states, δεδομένης και της reward function που ορίσαμε, και μη πεπερασμένο αριθμό διαφορετικών αναπαραστάσεων των actions, γνωρίζοντας ότι η Πιθανότητα Συνάφειας Τίτλου δεν λαμβάνει διακριτοποίηση.

Προτείνουμε η διακριτοποίηση να γίνει ως εξής: εάν η τιμή του χαρακτηριστικού εμπεριέχεται στο εύρος τιμών ενός κάδου (bucket), τότε λαμβάνει την τιμή του κάδου. Ορίζουμε, αυθαίρετα, τους κάδους, με τα εξής εύρη τιμών: $[0 - 0.1)$, $[0.1 - 0.2)$, $[0.2 - 0.3)$, $[0.3 - 0.4)$, $[0.4 - 0.5)$, $[0.5 - 0.6)$, $[0.6 - 0.7)$, $[0.7 - 0.8)$, $[0.8 - 0.9)$ και $[0.9 - 1.0)$. Οι αντίστοιχες τιμές των κάδων αυτών είναι οι εξής: 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. Εάν

το χαρακτηριστικό έχει εξαρχής τιμή ίση με 1.0 τότε παραμένει ως έχει. Η διακριτοποίηση ήταν και ο λόγος που προσθέσαμε αυτά τα χαρακτηριστικά αφαιρώντας τα δύο τελευταία της (α) αναπαράστασης για το state, δεδομένου πως συνεχώς αυξανόμενα από την μονάδα χαρακτηριστικά δεν μπορούν να ομαδοποιηθούν εύκολα σε κάδους.

Επιπλέον, όσον αφορά το χαρακτηριστικό της Πιθανότητας Συνάφειας Τίτλου, μία βολική λύση θα ήταν η μεταφορά μάθησης (Transfer Learning) από κάποιο ήδη προεκπαιδευμένο νευρωνικό δίκτυο εστιασμένο στο πρόβλημα μας. Αυτό θα έχει ως αποτέλεσμα την επιτάχυνση της online μάθησης του πράκτορα, αλλά με κόστος την προεκπαίδευση και την εξάρτηση από αυτό το πρότερο μοντέλο. Σε αυτήν την περίπτωση, προτείνουμε η μεταφορά μάθησης να γίνει από το πρώτο στάδιο του focused crawler, όπου ο μαθητής έχει εκπαιδευτεί στο να κατανοήσει το θέμα αναζήτησης.

Έχοντας περιγράψει τις πρώτες δύο μεθοδολογίες αναπαράστασης, μένει να μιλήσουμε για την (γ) *Only NLP Representation*. Εδώ οι αναπαραστάσεις του state και του action είναι παρόμοιες, με το state να διαθέτει τα word embeddings της περίληψης της δημοσίευσης, που περιέχει η ιστοσελίδα w , και το action τα word embeddings του τίτλου που του αντιστοιχεί. Συγκεκριμένα, λαμβάνουμε την συνένωση του Mean Pooling - Max Pooling, όπως χρησιμοποιήσαμε και στην Ενεργό Μάθηση. Δεδομένου πως η αναπαράσταση αυτή δεν παρέχει καθόλου πληροφορίες για το υπόλοιπο state - και κυρίως για το μονοπάτι της ιστοσελίδας w - αλλά ούτε άμεσες πληροφορίες για το ίδιο το w , θα την χρησιμοποιήσουμε ως baseline.

4.1.2.4 Focused Crawler Πράκτορες: DQN και DDQN

Δεδομένου πως οι χώροι των αναπαραστάσεων των states και των actions είναι μη πεπερασμένοι, προκειμένου να λύσουμε το πρόβλημά μας χρειαζόμαστε κάποια προσεγγιστική συνάρτηση (function approximation), και όχι το κλασικό πινακοειδές (tabular) Q-learning για να προσεγγίσουμε την βέλτιστη Q^* [7]. Έτσι, θα χρησιμοποιήσουμε τους αλγόριθμους Deep Q-learning και Double Deep Q-learning, κάνοντας ωστόσο ορισμένες τροποποιήσεις λόγω της κοινής αναπαράστασης του state και του action. Πιο αναλυτικά, ο πράκτορας DQN - και ο DDQN - αντιμετωπίζει ένα πολλαπλό πρόβλημα παλινδρόμησης (regression), όπου δεδομένου του state και των διαθέσιμων actions, με ένα πρόσθιο πέρασμα (forward propagation) υπολογίζει τα Q-values όλων των actions τους, έχοντας ως είσοδο μόνο το state.

Στην περίπτωσή μας, κάτι τέτοιο δεν είναι εφικτό, αφού αν και έχουμε να αντιμετωπίσουμε και πάλι ένα πρόβλημα παλινδρόμησης, η κοινή αναπαράσταση $\mathbf{x}(state, action)$ μας αναγκάζει να χρειαζόμαστε ξεχωριστό πρόσθιο πέρασμα για κάθε action, με είσοδο το $\mathbf{x}(state, action)$. Αναπόφευκτα, τροποποιούνται και οι αντίστοιχοι μαθηματικοί συμβολισμοί. Η Q-function για ένα state s , ένα action a και παραμέτρους θ γράφεται $Q(\mathbf{x}(s, a); \theta)$. Επιπλέον, στο Experience Replay πλέον αποθηκεύουμε τριάδες $\langle \mathbf{x}(s, a), r, s' \rangle$, όπου r το αντίστοιχο reward και s' το νέο state λόγω της επιλογής του action a . Παράλληλα, στο μέτωπο αναζήτησης, για πρακτικούς λόγους υλοποίησης, κρατάμε και το URL, έστω url , το οποίο αντιστοιχίζεται με το a . Έτσι, στο μέτωπο αναζήτησης αποθηκεύουμε εγγραφές τύπου $\langle url, \mathbf{x}(s, a), qvalue(a) \rangle$, όπου το $qvalue(a)$ είναι η τιμή του Q-value για το a .

Σημειώνουμε ότι αν και οι αλγόριθμοι Deep Q-learning και Double Deep Q-learning

έχουν σχεδιαστεί για περιπτώσεις όπου μόνο ο χώρος των states είναι πολύ μεγάλος ή μη πεπερασμένος, εμείς παρακάμπτουμε αυτό το εμπόδιο, αφού αν και ο χώρος αναπαράστασης των actions είναι πολύ μεγάλος, εκείνα που είναι διαθέσιμα σε κάθε state είναι πάντα πεπερασμένου αριθμού.

Όσον αφορά τις αναπαραστάσεις (α) και (β), που αναλύσαμε παραπάνω, δεδομένου πως η κύρια μετρική μας είναι το Harvest Rate και πως τα seeds URLs είναι a priori σχετικά με το θέμα, αποθηκεύουμε εξαρχής στο Experience Replay τις αντίστοιχες τριάδες $\langle \mathbf{x}(s, a), r, s' \rangle$ με $r = 1$, θεωρώντας ότι στο διάνυσμα $\mathbf{x}(s, a)$ τα χαρακτηριστικά της αναπαράστασης του state ήταν όλα 0. Μάλιστα, για λόγους ευκολίας, υποθέτουμε ότι τα actions που αντιστοιχούσαν στα seeds προήλθαν από ένα μηδενικού διανύσματος null state s_{-1} .

Οι πράκτορες Ενισχυτικής Μάθησης που βασίζονται στο Q-learning χρειάζονται έναν παράγοντα εξερεύνησης του χώρου των states, έχοντας επίγνωση του trade-off μεταξύ της εξερεύνησης (exploration) και της εκμετάλλευσης (exploitation), όπως περιγράψαμε αναλυτικά στο Κεφάλαιο 2. Ιδιαίτερα ο DQN-τύπου πράκτορας χρειάζεται αρκετή εξερεύνηση, πρώτου να αρχίσει να εκπαιδεύεται [8]. Για αυτό τον λόγο, χρειάζεται για έναν καθορισμένο αριθμό χρονικών βημάτων, ο focused crawler πράκτορας να επιλέγει με τυχαίο τρόπο actions από το μέτωπο αναζήτησης. Αυτό ωστόσο πρέπει να γίνει με προσοχή, αφού θέλουμε ο πράκτορας να φθάσει σε σύγκλιση σχετικά γρήγορα, δεδομένου πως έχουμε και καθορισμένο αριθμό προσπελάσεων ιστοσελίδων. Επιπλέον, είναι χρήσιμο να διατηρήσουμε τη δυνατότητα για εξερεύνηση ακόμα όταν έχει αρχίσει η εκπαίδευση. Αυτό γιατί θέλουμε συνεχώς να επεκτείνουμε μονοπάτια στο γράφο του Παγκόσμιου Ιστού, που μπορεί να φανούν αρκετά χρήσιμα μακροπρόθεσμα, ακόμα και αν άμεσα δεν φαίνονται τόσο υποσχόμενα. Κάτι τέτοιο πρέπει να γίνεται με μικρή πιθανότητα και μάλιστα η πιθανότητα αυτή να φθίνει συνεχώς, μέχρι μια ελάχιστη τιμή καθόλη την διάρκεια της μάθησης του πράκτορα.

Έτσι, θα ακολουθήσουμε στρατηγική (off-policy) ϵ -greedy [7], με την οποία θα επιλέγεται το καλύτερο ζευγάρι $\mathbf{x}(state, action)$ από το μέτωπο αναζήτησης με πιθανότητα $1 - \epsilon$. Από την άλλη, με πιθανότητα ϵ θα επιλέγεται κάποιος με τυχαίο τρόπο. Ορίζουμε ακόμα $\lambda \in [0, 1)$ ώστε σε κάθε διαφορετικό χρονικό βήμα (timestep) να ανανεώνουμε το ϵ ως εξής:

$$\epsilon = \max\{\lambda \cdot \epsilon, \epsilon_0\} \quad (4.9)$$

όπου το $\epsilon_0 \in [0, 1]$ η ελάχιστη τιμή που μπορεί να πάρει το ϵ .

Όπως γνωρίζουμε οι πράκτορες DQN και DDQN χρησιμοποιούν δύο νευρωνικά δίκτυα - ένα για δίκτυο πρόβλεψης (Q-Network), με παραμέτρους $\boldsymbol{\theta}$, και ένα δίκτυο στόχο (Target Q-Network), με παραμέτρους $\boldsymbol{\theta}^-$, τα οποία υποχρεωτικά έχουν την ίδια αρχιτεκτονική. Θα χρησιμοποιήσουμε Multilayer-Perceptron (MLP) για την εκτίμηση της συνάρτησης (function approximation) της Q-function, το οποίο θα λαμβάνει ως είσοδο το διάνυσμα $\mathbf{x}(state, action)$ και θα εξάγει το αντίστοιχο Q-value. Ο αριθμός των κρυφών επιπέδων εξαρτάται από την επιλογή της αναπαράστασης της εισόδου. Στο σημείο αυτό, αξίζει να γράψουμε τις σχέσεις για τους αλγόριθμους που θα χρησιμοποιήσουμε. Για το DQN με Experience Replay, ο στόχος για κάθε $Q(\mathbf{x}(s_t, a); \boldsymbol{\theta}^-)$ ενημερώνεται ως εξής:

$$y = r_{t+1} + \gamma \max_{a \in \mathcal{A}_{s_{t+1}}} Q(\mathbf{x}(s_{t+1}, a); \boldsymbol{\theta}^-) \quad (4.10)$$

Ενώ για το DDQN με Experience Replay ο στόχος για κάθε $Q(\mathbf{x}(s_{t+1}, a); \boldsymbol{\theta}^-)$ ενημερώνεται ως εξής:

$$y = r_{t+1} + \gamma Q\left(\mathbf{x}\left(s_{t+1}, \underset{a \in a_{s_{t+1}}}{\operatorname{argmax}} Q(\mathbf{x}(s_{t+1}, a); \boldsymbol{\theta})\right); \boldsymbol{\theta}^-\right) \quad (4.11)$$

όπου γ είναι ο εκπτώτικος παράγοντας (discount factor). Σημειώνουμε πως δεδομένου πως έχουμε να αντιμετωπίσουμε ένα συνεχές πρόβλημα χωρίς επεισόδια - αλλά με ένα μόνο τερματικό state, πρέπει να θέσουμε το γ περίπου ίσο με το 1. Επιπλέον, αξίζει να υπογραμμίσουμε πως στον υπολογισμό του y χρησιμοποιούμε τα actions $a \in a_{s_{t+1}}$, δηλαδή μόνο εκείνα που αντιστοιχούν στα URLs που εξάγονται από την ιστοσελίδα-κόμβο w_{t+1} , και δεν περιέχονται στο κλειστό σύνολο C_{t+1} . Διαισθητικά, με αυτόν τον τρόπο εξετάζουμε άμεσα κατά πόσο το νέο state είχε κάποιο καλύτερο action από το προηγούμενό του. Προκειμένου να γίνει αυτό, χρειάζεται στην υλοποίησή μας να κρατάμε όλα τα στιγμιότυπα του κλειστού συνόλου C_{t_k} , δηλαδή για κάθε $t_k \in [1, T]$, όπου T ο συνολικός αριθμός των χρονικών βημάτων.

Η συνάρτηση που θέλουμε να ελαχιστοποιήσουμε είναι η Mean Squared Error της εξίσωσης Bellman, όπου ο βέλτιστος στόχος $r + \gamma Q^*(\mathbf{x}(s, a))$ προσεγγίζεται με το y , χρησιμοποιώντας τις παραμέτρους $\boldsymbol{\theta}^-$, από κάποια προηγούμενη επανάληψη [8].

$$L(\boldsymbol{\theta}) = [y - Q(\mathbf{x}(s_t, a_t); \boldsymbol{\theta})]^2 \quad (4.12)$$

Στα πειράματά μας θα αναφερθούμε στις υπερπαραμέτρους των μοντέλων που θα δοκιμάσουμε. Εφόσον έχουμε δει αναλυτικά όλο το δεύτερο στάδιο του focused crawler πράκτορα, θα δείξουμε αναλυτικά τα βήματα του αλγορίθμου (Αλγόριθμος 4.2) του σταδίου αυτού.

ΑΛΓΟΡΙΘΜΟΣ 4.2: Δεύτερο Στάδιο Focused Crawler: Ενισχυτική Μάθηση

Input: *seeds* URLs, *TIMESTEPS*, *RelevanceCalculator*, representation \mathbf{x} , URLScorer \mathcal{Q} -Network \mathcal{Q} , Target \mathcal{Q} -Network \mathcal{Q}^- , Target Update Period *TUP*, Batch Size *batch*, Replay Start Size *start*, *epsilon* γ , ϵ , *eps_decay* and *eps_min*

Initialize s_0 from *seeds*, s_{-1} as null state, empty crawl frontier F_0 , closure C_0 , Experience Replay *ER* and $t := 1$

while *seeds* not empty **do**

 Pop one *url* from *seeds* with respective *action* := action of l

 From *url* fetch webpage w

 Store to *ER*: $\langle \mathbf{x}(s_{-1}, action), 1, s_0 \rangle$ and add *url* to C_0

$L :=$ Extract all outlinks from w that are not in C_0

for l in L **do**

$a :=$ action corresponding to l and $qvalue := \mathcal{Q}(\mathbf{x}(s_0, a); \mathcal{Q})$

 Store $\langle l, \mathbf{x}(s_0, a), qvalue \rangle$ to F_0

end for

end while

while not $t > TIMESTEPS$ **do**

if $t \text{ MOD } TUP = 0$ **then**

 Update $\mathcal{Q}^- = \mathcal{Q}$

end if

if $t < start$ **then**

 Pop at random a record from F : $\langle url, \mathbf{x}(s_{t_k}, a), qvalue \rangle$

else

if with probability ϵ **then**

 Pop at random a record from F : $\langle url, \mathbf{x}(s_{t_k}, a), qvalue \rangle$

else

 Pop the record with the highest \mathcal{Q} -value from F : $\langle url, \mathbf{x}(s_{t_k}, a), qvalue \rangle$

 Update $\epsilon := \max\{\epsilon \cdot eps_decay, eps_min\}$

end if

end if

$F_t := F_{t-1}$ without records corresponding to $DEL(A_{s_{t-1}})$ and $C_t := C_{t-1} \cup \{url\}$

 From *url* fetch webpage w_t and get transition from s_{t-1} to s_t

 Observe reward $r_t := \text{RelevanceCalculator}(w_t)$

 Store to *ER*: $\langle \mathbf{x}(s_{t-1}, a), r, s_t \rangle$

if $t > start$ **then**

 Sample from *ER* a random *batch* of records $\langle \mathbf{x}(s_{i-1}, a_{i-1}), r_i, s_i \rangle$

 Set $y := r_i + \gamma \max_{a \in a_{s_i}} \mathcal{Q}(\mathbf{x}(s_i, a); \mathcal{Q}^-)$ // DQN

 Set $y := r_i + \gamma \mathcal{Q}(\mathbf{x}(s_i, \arg\max_{a \in a_{s_i}} \mathcal{Q}(\mathbf{x}(s_i, a); \mathcal{Q}^-)); \mathcal{Q}^-)$ // DDQN

 Perform gradient descent step on $(y - \mathcal{Q}(\mathbf{x}(s_{i-1}, a_{i-1}); \mathcal{Q}))^2$

end if

 Update the q-values of all records in F_t using \mathcal{Q}

$L :=$ Extract all outlinks from w_t that are not in C_t

for l in L **do**

$a' :=$ action corresponding to l and $qvalue := \mathcal{Q}(\mathbf{x}(s_t, a'); \mathcal{Q})$

 Store to F_t : $\langle l, \mathbf{x}(s_t, a'), qvalue \rangle$

end for

$t := t + 1$

end while

Output: *HarvestRate*, C_T

4.2 Ντετερμινιστικός Focused Crawler: Greedy Επιλογή

Στην ενότητα 4.1, μελετήσαμε την περίπτωση ενός focused crawler που αποτελείται από δύο στάδια μάθησης, με τα οποία το σύστημα μπορεί να κατανοεί τα ενδιαφέροντα θέματα αναζήτησης αλλά και τους τρόπους με τους οποίους είναι σε θέση να βρίσκει τα σχετικά URLs. Στο σημείο αυτό, θα περιγράψουμε μία περίπτωση ενός ντετερμινιστικού focused crawler, ο οποίος δεν διαθέτει κανένα μοντέλο Μηχανικής Μάθησης.

Ο ντετερμινιστικός focused crawler μας (**DetFC**) ακολουθεί το γενικό μοντέλο (Σχ.3.1), που περιγράψαμε στο Κεφαλαίο 3, λαμβάνοντας ως είσοδο λίγα σε αριθμό seeds URLs αλλά και λέξεις-κλειδιά (keywords), που περιγράφουν το θέμα αναζήτησης. Θεωρούμε πως οι λέξεις-κλειδιά στην είσοδο του συστήματος είναι οργανωμένες σε επιμέρους κατηγορίες λέξεων-κλειδιών. Ο *DetFC* θα εφαρμόζεται σε δύσκολα (ειδικά) προβλήματα αναζήτησης, τα οποία θα χαρακτηρίζονται από τον συνδυασμό λέξεων-κλειδιών από όλες τις καθορισμένες κατηγορίες τους.

Τυπικά, συμβολίζουμε με T το υπερσύνολο θέμα της αναζήτησης και με t το γνήσιο υποσύνολό του, το οποίο αποτελεί το ενδιαφέρον θέμα του χρήστη. Επιπλέον, συμβολίζουμε με K_1, K_2, \dots, K_{N_K} τις διαφορετικές μεταξύ τους ως προς το νόημα κατηγορίες K και με KW_i όλες τις λέξεις κλειδιά KW για $i < N_{kw}$, όπου N_{kw} το πλήθος των λέξεων-κλειδιών. Θεωρούμε πως $KW_i \neq KW_j$ για όλα τα $i \neq j$ και ορίζουμε την συνάρτηση $F_K: KW \rightarrow K$, με την οποία αντιστοιχίζουμε μονοσήμαντα κάθε λέξη-κλειδί με μία και μόνο κατηγορία. Τέλος, θεωρούμε ότι κάθε κατηγορία K_i διαθέτει μια τιμή προτεραιότητας P_i για $i < N_{kw}$, και ορίζουμε την συνάρτηση $F_P: K \rightarrow P$, όπου το $P > 0$ είναι κάποιος θετικός αριθμός. Κατά αυτόν τον τρόπο, έχουμε P_1, P_2, \dots, P_K προτεραιότητες που επιδέχονται διάταξη μεταξύ τους.

Προκειμένου ένα κείμενο να ανήκει στο t , πρέπει να περιέχεται σε αυτό τουλάχιστον μία λέξη-κλειδί από όλες τις κατηγορίες K . Αν δεν τηρείται το παραπάνω, τότε θεωρούμε ότι το κείμενο είναι αρκετά πιθανόν να ανήκει στο T , δεδομένου πως τα seeds ανήκουν σίγουρα στο t και άρα και στο T . Όταν ένα κείμενο περιέχει τουλάχιστον μία λέξη-κλειδί μιας κατηγορίας K_i , τότε λέμε πως το κείμενο έχει (has) την P_i . Ορίζουμε τις μεταβλητές r_i τέτοιες ώστε:

$$r_i(\text{text}) = \begin{cases} 1, & \text{if text has } P_i \\ 0, & \text{otherwise} \end{cases} \quad (4.13)$$

Όσον αφορά την αναζήτηση δημοσιεύσεων, ο *DetFC* θα λαμβάνει υπόψιν τον τίτλο της δημοσίευσης για κάθε URL που εξάγεται και τοποθετείται στο μέτωπο αναζήτησης, ενώ για τον υπολογισμό της συνάφειας μιας δημοσίευσης θα λαμβάνει υπόψιν και τον τίτλο και την περίληψη (abstract) της. Πιο αναλυτικά, ο URL Scorer θα είναι η παρακάτω συνάρτηση

$$URLScorer(\text{title}) = \sum_{i=1}^{N_K} r_i P_i \quad (4.14)$$

Παράλληλα, ορίζουμε την συνάρτηση του Relevant Calculator ως εξής:

$$RelevanceCalculator(publication) = \begin{cases} 1, & \text{if abstract has } P_1, \dots, P_{N_K} \\ 1, & \text{if title has } P_1, \dots, P_{N_K} \\ 0, & \text{otherwise} \end{cases} \quad (4.15)$$

Δεδομένων των δυο βασικών συναρτήσεων του γενικού μοντέλου του focused crawler, ο DetFC επιλέγει να επισκεφτεί πάντα εκείνο το URL στο μέτωπο αναζήτησης που έχει την μεγαλύτερη τιμή της συνάρτησης URLScorer. Αυτό σημαίνει πως αν υπάρχει κάποιο URL, που στον τίτλο του περιέχονται λέξεις-κλειδιά από όλες τις κατηγορίες K , τότε ο DetFC θα το επισκεφτεί σίγουρα νωρίτερα από τα υπόλοιπα για τα οποία δεν ισχύει αυτό. Σε πιο γενικό πλαίσιο, ο DetFC εξασφαλίζει πως θα επιλέγονται πάντα διαθέσιμα URLs του μετώπου αναζήτησης που έχουν τις μεγαλύτερες αθροιστικά προτεραιότητες.

ΑΛΓΟΡΙΘΜΟΣ 4.3: Άπληστος Ντετερμινιστικός Αλγόριθμος Focused Crawler: DetFC

Input: seeds URLs, $\{K_i\}$, $\{KW_i\}$, $\{P_i\}$, TIMESTEPS
Initialize empty crawl frontier F and closure C
Build URLScorer and RelevanceCalculator with $\{K_i\}$, $\{KW_i\}$, $\{P_i\}$
while seeds not empty **do**
 Pop a URL url from seeds
 Add url to C
 Fetch webpage w url
 $L :=$ Extract all outlinks URLs from w that are not in C
 for l in L **do**
 $title :=$ Extract anchor text from l
 $score :=$ URLScorer($title$)
 Add tuple $(l, score)$ to F
 end for
end while
 $timestep := 1$
while not $timestep > TIMESTEPS$ **do**
 Select greedily a $(url, score)$ from F
 Observe relevance := RelevanceCalculator(url)
 Add url to C
 Delete all tuples from F that have as URL the url
 $L' :=$ Extract all outlinks from url that are not in C
 for l in L' **do**
 $title :=$ Extract anchor text from l
 $score :=$ URLScorer($title$)
 Add tuple $(l, score)$ to F
 end for
 $timestep := timestep + 1$
end while
Output: HarvestRate, C

Στην ειδική περίπτωση που οι διαφορετικές κατηγορίες λέξεων-κλειδιών είναι δύο, δηλ.

$N_K = 2$, αυτό σημαίνει πως ο DetFC θα επισκέπτεται με πρώτη προτεραιότητα τίτλους που περιέχουν λέξεις-κλειδιά και από τις δυο κατηγορίες, με δεύτερη προτεραιότητα τίτλους που περιέχουν λέξεις-κλειδιά από την πιο σημαντική κατηγορία και με τρίτη προτεραιότητα τίτλους που προέρχονται από την λιγότερη σημαντική κατηγορία. Έτσι, ο αλγόριθμος του DetFC είναι άπληστος (greedy). Τονίζουμε πως στις περιπτώσεις που υπάρχουν πολλά URLs που ισοβαθούν και η τιμή τους είναι μέγιστη στο μέτωπο αναζήτησης, επιλέγουμε κάποιο με τυχαίο τρόπο. Ο αλγόριθμός του DetFC παρουσιάζεται αναλυτικά στον Αλγόριθμο 4.3.

Κεφάλαιο 5

Πειραματική Ανάλυση και Αποτελέσματα

Στο κεφάλαιο αυτό θα περιγράψουμε την όλη πειραματική διαδικασία που ακολουθήσαμε, καθώς και την εξαγωγή των αποτελεσμάτων από τα πειράματα που αξιοποίησαν τα εναλλακτικά συστήματα *focused crawler* των μεθοδολογιών που είδαμε στο προηγούμενο κεφάλαιο. Για την οργάνωση της πειραματικής διαδικασίας, αλλά και για την μάθηση του συστήματος, υπενθυμίζουμε πως στη διάθεσή μας είχαμε δύο καταξιωμένους εμπειρογνώμονες εθνοφαρμακολογίας (*domain experts*), οι οποίοι, στα πειράματά μας, ανέλαβαν τον ρόλο του ερευνητή για τον *focused crawler* δύο σταδίων. Στις υποενότητες που ακολουθούν θα περιγράψουμε τα δύο προβλήματα αναζήτησης που αναφέραμε - ένα γενικό και ένα ειδικό - και θα αναλύσουμε τις προσεγγίσεις μας εξάγοντας τα αντίστοιχα συμπεράσματα.

Και για τα δύο προβλήματα αναζήτησης, η διαδικασία του *focused crawling* πραγματοποιείται πάνω στο PubMed [13] και την βάση δεδομένων MEDLINE η οποία παρέχει πρόσβαση στην πλειονότητα των δημοσιεύσεων ιατροφαρμακευτικού περιεχομένου. Σημειώνουμε πως χρησιμοποιούμε προεκπαιδευμένα *word2vec word embeddings* [70], μεγέθους 4GB. Τα *word2vec word embeddings* αυτά έχουν εκπαιδευτεί πάνω σε όλο το PubMed καθώς και σε αντίστοιχα κείμενα από την Wikipedia [71]. Τονίζουμε πως κάθε λέξη, μέσω του *word embedding*, αντιστοιχίζεται σε ένα διάνυσμα 200 διαστάσεων. Υπενθυμίζουμε πως ο κύριος σκοπός μας είναι να μεγιστοποιήσουμε το τελικό *harvest rate* του *focused crawler*. *Στα διαγράμματα harvest rate που ακολουθούν, αναπαριστούμε το συσσωρευτικό harvest rate για όλα τα χρονικά βήματα της εκτέλεσης του focused crawler.*

5.1 Γενικό Πρόβλημα Αναζήτησης

Στο γενικό πρόβλημα, ορίζουμε ως σχετικό θέμα αναζήτησης οτιδήποτε έχει να κάνει με εθνοφαρμακολογία, χωρίς κανέναν άλλον περιορισμό. Αυτό σημαίνει πως οι αναφορές που συλλέγει το σύστημα μπορεί να ανήκουν σε δύο κατηγορίες: E και E^C , όπου η πρώτη αποτελείται από τα κείμενα που είναι σχετικά με την εθνοφαρμακολογία, ενώ η δεύτερη είναι η συμπληρωματική της πρώτης.

5.1.1 Προετοιμασία Πειράματος

Αφού πλέον ορίστηκε το θέμα αναζήτησης, οι εμπειρογνώμονες εθνοφαρμακολογίας διαλέγουν 25 URLs δημοσιεύσεων, που θεώρησαν πως ανήκουν στην κατηγορία E . Η διαδικασία

αυτή, ωστόσο, πραγματοποιείται με τρόπο, ώστε ο κάθε εμπειρογνώμονας να μας δώσει περίπου τον ίδιο αριθμό δημοσιεύσεων, χωρίς να γνωρίζει τα κείμενα που μας έδωσε ο άλλος. Τα 25 URLs αυτά αποτελούν την είσοδο και, άρα, τις seeds ιστοσελίδες του focused crawler. Από αυτά παίρνουμε 427 URLs, με τα οποία seeds συνδέονταν άμεσα. Στη συνέχεια, εκτελούμε τυχαίο πιλοτικό τρέξιμο (Random Crawling) με είσοδο τα seeds και φέρνουμε 400 URLs δημοσιεύσεων. Αφαιρώντας όλα τα διπλά αποτελέσματα (duplicates), φθάνουμε πολύ εύκολα σε ένα σύνολο δεδομένων 813 URLs, το οποίο, ωστόσο, είναι μη επισημειωμένο, αφού δεν περιέχει annotations.

Στη συνέχεια, δεδομένου πως διαθέτουμε δύο επισημειωτές, πρέπει να ελέγξουμε την συμφωνία των επισημειωτών (Interannotator Agreement), προκειμένου να μελετήσουμε την αξιοπιστία των επισημειώσεων, που θα λαμβάνει το σύστημα στο πρώτο στάδιο της Ενεργού Μάθησης (Active Learning). Διαλέγοντας τα περισσότερα από τα seed URLs και τυχαία δείγματα από το σύνολο δεδομένων μας, ρωτάμε τους επισημειωτές για 50 συνολικά URLs και μελετάμε τρεις μετρικές: το Raw Agreement, το Cohen's kappa και το Krippendorff's alpha. Στους πίνακες 5.1 και 5.2 παρουσιάζονται τα αποτελέσματα των επισημειώσεων (confusion matrix) αλλά και των τιμών των μετρικών, που προέκυψαν αντίστοιχα. Και για τις τρεις μετρικές βλέπουμε πως έχουμε σχεδόν άριστη συμφωνία δεδομένου πως τόσο το Cohen's kappa όσο και το Krippendorff's alpha δείχνουν άριστη συμφωνία όταν οι τιμές τους είναι μεγαλύτερες από 0.8.

Raw Agreement	0.96
Cohen's Kappa	0.92
Krippendorff's alpha	0.92

Πίνακας 5.1: *Μετρικές Interannotator Agreement Γενικού Προβλήματος*

		Annotator 2	
		E	E^C
Annotator 1	E	24	2
	E^C	0	24

Πίνακας 5.2: *Confusion Matrix Επισημειώσεων Γενικού Προβλήματος*

Δεδομένης της μεγάλης συμφωνίας των επισημειωτών, προχωράμε στην απόκτηση του χρυσού κανόνα (Gold Standard) όλων των επισημειώσεων T των 800 URLs δεδομένων, που συλλέξαμε με την παραπάνω διαδικασία. Κατά αυτόν τον τρόπο, δημιουργούμε ένα σύνολο δεδομένων (URL, T) , που περιέχει 813 URLs με επισημειώσεις ως προς την συνάφεια στο θέμα της εθνοφαρμακολογίας. Σημειώνουμε πως το σύνολο δεδομένων αυτό περιέχει κείμενα της κατηγορίας E σε ποσοστό 58%. Από αυτό κρατάμε τα 25 seeds URLs για το αρχικό σύνολο δεδομένων.

5.1.2 Ενεργός Μάθηση

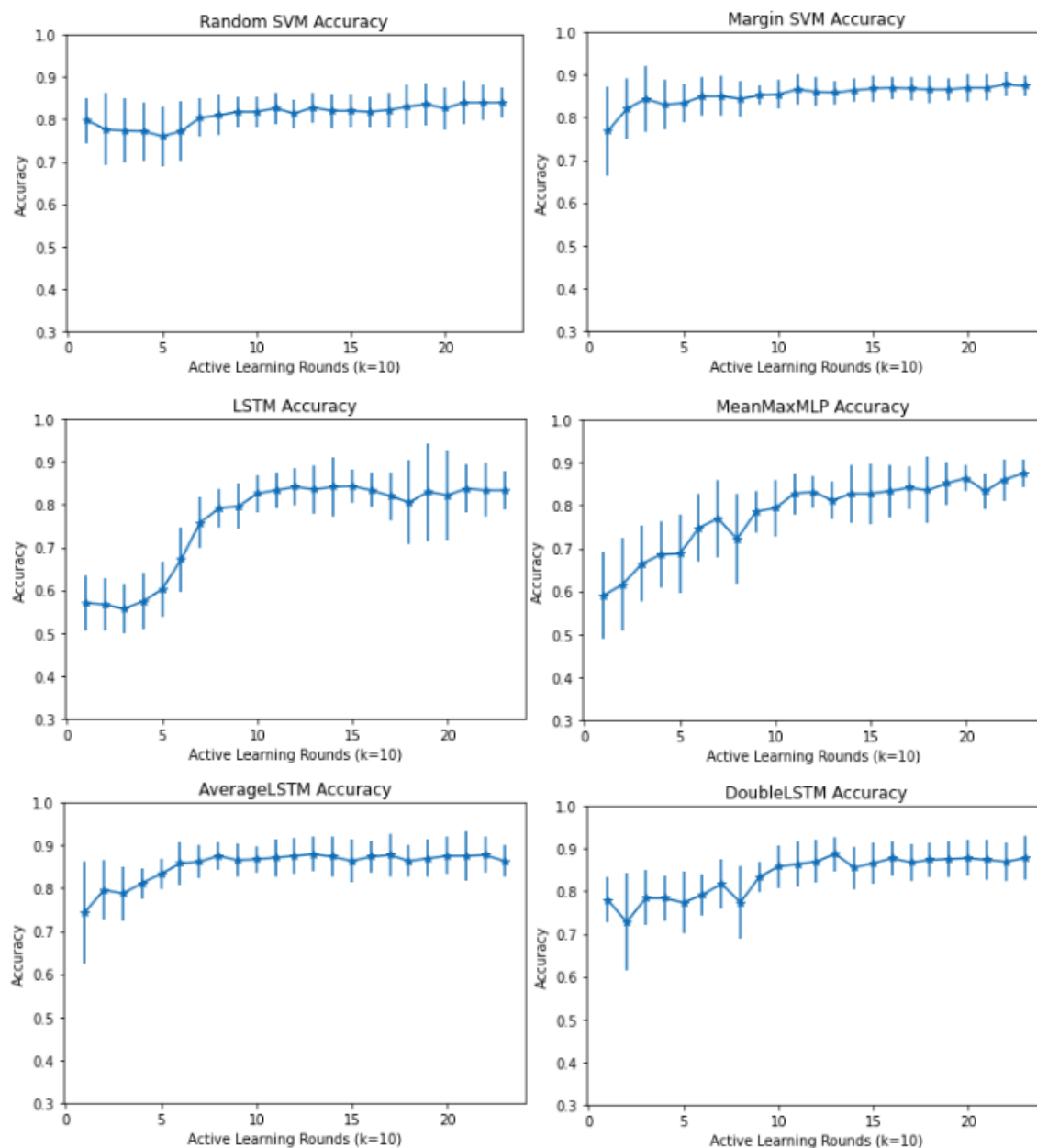
Στη συνέχεια, θέλουμε να φτιάξουμε τρία αρχικά σύνολα δεδομένων για την είσοδο της Ενεργού Μάθησης - το σύνολο εκπαίδευσης 50 δεδομένων με το 58% να ανήκουν στην κατηγορία E, το σύνολο τεστ (test set) 50 δεδομένων με το 50% να ανήκουν και πάλι στην κατηγορία E και το pool με τα υπόλοιπα 763 δεδομένα. Κάνουμε 10-fold-cross-validation [72], ώστε να δημιουργήσουμε 10 διαφορετικές τριάδες των τριών παραπάνω συνόλων δεδομένων της Ενεργού Μάθησης, με την ιδιότητα ότι το σύνολο τέστινγκ θα είναι πάντα ξένο με οποιοδήποτε σύνολο τέστινγκ άλλης τριάδας. Το ίδιο ισχύει και για τα 25 δεδομένα της E^C του αρχικού συνόλου εκπαίδευσης. Με αυτόν τον τρόπο, θέλουμε να δημιουργήσουμε "ασφαλή" σύνολα δεδομένων, ώστε με στατιστικά τεστ να προβούμε στην επιλογή του καλύτερου μοντέλου. Ωστόσο, εάν επιλέγαμε δεδομένα από το σύνολο των 813 δεδομένων με τυχαίο τρόπο, ώστε να φτιάχναμε τις 10 τριάδες δεδομένων, τότε είναι σχεδόν βέβαιο πως θα υπήρχαν επικαλύψεις. Μια τέτοια μέθοδος δεν συνίσταται να συμβαίνει σε καμία περίπτωση για την επιλογή του καλύτερου μοντέλου [73].

Προκειμένου να προχωρήσουμε στην εκπαίδευση των μοντέλων Ενεργού Μάθησης, θέτουμε ως μέγιστο όριο (budget) ερωτοαποκρίσεων με το Gold Standard, δηλαδή τον ερευνητή, τις 250. Θεωρούμε ως βασικό (baseline) μοντέλο τον Random SVM, που χρησιμοποιεί τυχαία δειματοληψία (Random Sampling) και εξετάζουμε ακόμα τα μοντέλα LSTM, AverageLSTM, DoubleLSTM και MeanMaxMLP, που περιγράψαμε αναλυτικά στο προηγούμενο κεφάλαιο. Για όλα τα μοντέλα, χρησιμοποιούμε $k = 10$, δηλαδή σε κάθε γύρο (batch) της Ενεργού Μάθησης επαυξάνουμε το σύνολο εκπαίδευσης κατά 10 παρατηρήσεις από το pool. Με δεδομένο το budget που θέσαμε παραπάνω, ορίζονται συνολικά 25 γύροι εκπαίδευσης για κάθε μοντέλο. Στο σχήμα 5.1 απεικονίζουμε τις γραφικές παραστάσεις του μέσου accuracy των μοντέλων που εκπαιδεύτηκαν με Ενεργό Μάθηση (Active Learning).

Για να μπορέσουμε να βγάλουμε συμπεράσματα σχετικά με το ποιο είναι το καλύτερο μοντέλο, προκειμένου να προχωρήσουμε στο δεύτερο στάδιο του focused crawler, λαμβάνουμε υπόψιν τις εκπαιδεύσεις των μοντέλων από τον 10ο γύρο και μετά, όπου αυτά αρχίζουν και συγκλίνουν. Ξεκινώντας από τον 10ο γύρο, θα επιλέξουμε εκείνο το μοντέλο που έχει μεγαλύτερη αντίστοιχη μέση τιμή του accuracy και παρουσιάζει στατιστικά σημαντική διαφορά (statistical significance) έναντι των άλλων, λαμβάνοντας υπόψιν την ανάγκη για ευρωστία και επαναπαραγωγιμότητα των αποτελεσμάτων. Τα αποτελέσματα υποδεικνύουν ότι αυτή η διαφορά αναδεικνύεται στον **13ο γύρο**.

Χρησιμοποιούμε paired-t-test για να ελέγξουμε την στατιστική σημαντικότητα, το οποίο αποτελεί πολύ συχνή μέθοδο για την επιλογή του καλύτερου μοντέλου ταξινομήτη [73]. Στο τεστ αυτό, θυμίζουμε ότι η null υπόθεση - οι παρατηρήσεις δεν ανήκουν στην ίδια κατανομή - απορρίπτεται όταν $p - value < 0.05$. Αρχικά, ελέγχουμε να δούμε εάν οι κατανομές των μέσων τιμών όλων των μοντέλων ακολουθούν κανονική κατανομή, ώστε να τηρείται απόλυτα ο βασικός περιορισμός για το paired-t-test μέσω Shapiro-Wilk test. Θυμίζουμε ότι το τεστ αυτό έχει ως null υπόθεση ότι η κατανομή είναι κανονική την οποία δεχόμαστε εάν το $p - value > 0.05$. Σε αντίθετη περίπτωση την απορρίπτουμε και θεωρούμε πως η κατανομή δεν είναι κανονική. Στο γύρο αυτό, για όλα τα μοντέλα δεν απορρίπτεται η null υπόθεση, εκτός από το AverageLSTM. Δεδομένου πως οι παρατηρήσεις είναι λίγες σε αριθμό, είμα-

σε αυστηροί με την κανονικότητα των παρατηρήσεων και δεν εκτελούμε το τεστ πάνω στις διαφορές τους. Έτσι, τρέχουμε το paired-t-test για όλα τα ζευγάρια, εκτός από εκείνα που περιέχουν το AverageLSTM. Τα αποτελέσματα έδειξαν πως το p-value < 0.05, σε όλα τα τεστ που συμμετείχε η κατανομή του DoubleLSTM. Το ίδιο ισχύει και για το Margin SVM, εκτός όταν εξετάσθηκε μαζί με το LSTM. Έτσι, από όλα τα μοντέλα που εξετάσθηκαν με αυτόν τον τρόπο, προτιμάμε στατιστικά το DoubleLSTM.



Σχήμα 5.1: Διαγράμματα Ενεργού Μάθησης πάνω στο Μέσο Accuracy με Τυπική Απόκλιση συν το χρόνο.

Μένει να δούμε μεταξύ του AverageLSTM και του DoubleLSTM, αν διαφέρει κάποιο στατιστικά για αυτόν τον γύρο. Θα εκτελέσουμε το Wilcoxon signed-rank test, το οποίο είναι η μη παραμετρική έκδοση του paired-t-test που έχει ωστόσο λιγότερη στατιστική δύναμη. Θυμίζουμε ότι η null υπόθεση είναι ίδια με του paired-t-test. Το p-value που προέκυψε

μεταξύ του **AverageLSTM** και του **DoubleLSTM** ήταν $0.53 > 0.05$, με αποτέλεσμα να μην απορρίπτεται η null υπόθεση ότι οι τιμές τους προέρχονται από την ίδια κατανομή. Ωστόσο, το $p\text{-value} < 0.05$ για όλα τα άλλα ζευγάρια που περιείχαν το AverageLSTM.

Στον πίνακα 5.3 απεικονίζουμε τις μέσες τιμές και τις τυπικές αποκλίσεις των μοντέλων στο γύρο 13, ενώ παράλληλα δείχνουμε και την ομαδοποίηση των μοντέλων ως προς την στατιστική σημαντικότητά τους. Δύο μοντέλα αντιστοιχούν στο ίδιο γράμμα, όταν στατιστικά είναι το ίδιο αποτελεσματικά, όπως τα AverageLSTM και DoubleLSTM. Επίσης, σε ένα μοντέλο αντιστοιχίζονται δύο γράμματα, όταν είναι το ίδιο αποτελεσματικό, στατιστικά, με όλα τα μοντέλα και των δύο γραμμάτων. Για παράδειγμα, βλέπουμε πως το LSTM αντιστοιχίζεται στις ομάδες B και C, επειδή δε διαφέρει στατιστικά από το MarginSVM - ομάδα B - και από το RandomSVM - ομάδα C, μολονότι τα MarginSVM και RandomSVM διαφέρουν.

Classifier	Mean Accuracy	Standard Deviation	Group
Random SVM	0.828	0.034	C
Margin SVM	0.858	0.026	B
LSTM	0.836	0.057	B C
AverageLSTM	0.880	0.041	A
DoubleLSTM	0.880	0.040	A
MeanMaxMLP	0.812	0.044	D

Πίνακας 5.3: Μέσες τιμές Accuracy, Τυπικές Αποκλίσεις για τον Γύρο 13 με ακρίβεια τριών μηδενικών και ομαδοποίηση βάσει στατιστικής σημαντικότητας. Η ομάδα A περιέχει τα καλύτερα μοντέλα, ενώ η ομάδα D τα χειρότερα.

Όπως βλέπουμε, μπορούμε να επιλέξουμε κάποιο από τα AverageLSTM και DoubleLSTM, αφού έχουν ίδιες μέσες τιμές με ακρίβεια τριών μηδενικών και στατιστικά μπορούμε να υποθέσουμε πως ανήκουν στην ίδια κατανομή οι επιμέρους επιδόσεις τους. Συμπεραίνουμε πως το παραπάνω δυαδικό πρόβλημα ταξινόμησης σε καμία περίπτωση δεν απαιτούσε όλα τα κείμενα που συλλέχθηκαν στο pool, αφού χρειαστήκαμε τελικά 130 επισημειώσεις έναντι των 763. Από την αλλή είδαμε πως η Ενεργός Μάθηση έπαιξε σημαντικό ρόλο σε αυτό, καθώς μείωσε κατά **83%** τις ερωτήσεις που έγιναν τελικά στον ερευνητή (Gold Standard).

Σημειώνουμε, επίσης, πως σημαντική επίδραση είχε και ο αριθμός των ερωτήσεων που έκανε το σύστημα σε κάθε γύρο της Ενεργού Μάθησης. Παρατηρήσαμε πως τα μοντέλα SVM δεν επηρεάζονταν σε επίδοση ακόμα και με σχετικά μικρό αριθμό (5) ερωτήσεων ανά γύρο προκειμένου να φθάσουν σύντομα σε σύγκλιση. Σε αντίθεση, τα τεχνητά νευρωνικά δίκτυα χρειάζονταν μεγαλύτερο αριθμό (10). Το γεγονός αυτό πηγάζει από το ότι τα νευρωνικά δίκτυα, όπως τα αναδρομικά (RNN), χρειάζονται περισσότερα δεδομένα εκπαίδευσης λόγω των περισσότερων παραμέτρων τους, και αυτός είναι και ο κύριος λόγος που δεν χρησιμοποιούνται κατά πολύ στην Ενεργό Μάθηση [74]. Ωστόσο, στην περίπτωσή μας, βλέπουμε πως τα μοντέλα Βαθιάς Μάθησης, που χρησιμοποιήσαμε, ξεπερνάνε τα μοντέλα κλασικής Μηχανικής Μάθησης, τόσο σε αποτελεσματικότητα όσο και αποδοτικότητα. Όσον αφορά την αποδοτικότητα, όπως βλέπουμε και στο Σχήμα 5.1, τα μοντέλα AverageLSTM και DoubleLSTM χρειάστηκαν λιγότερες ερωτήσεις/επισημειώσεις (130), προκειμένου να φθάσουν σε σύγκλιση, σε σχέση με το MarginSVM. Βλέπουμε πως το τελευταίο χρειάζεται περισσότερες ερωτήσεις, αφού η γραφική παράστασή του είναι γενικώς αύξουσα. Παρόλα

αυτά, σημειώνουμε ότι ακόμα και για περισσότερες ερωτήσεις το MarginSVM δεν ξεπερνά τις επίδοσεις του γύρου 13 των άλλων δύο μοντέλων.

5.1.3 Ενισχυτική Μάθηση

Έτσι, προχωράμε στο δεύτερο στάδιο του focused crawler, δηλαδή της Ενισχυτικής Μάθησης. Στο βήμα αυτό χρησιμοποιούμε το καλύτερο μοντέλο AverageLSTM που εκπαιδεύσαμε παραπάνω. Σε αυτό το γενικό πρόβλημα που περιγράψαμε, θα χρησιμοποιήσουμε δύο διαφορετικές μεθοδολογίες: (1) Focused Crawler με τυχαίο πιλοτικό τρέξιμο (Αλγόριθμος 4.1), και (2) Focused Crawler με General Representation αναπαράσταση. Ο baseline focused crawler, ο οποίος θυμίζουμε δεν χρησιμοποιεί κανένα μοντέλο Μηχανικής Μάθησης, λειτουργεί ως baseline για το γενικό πρόβλημα αναζήτησης. Θα τον τρέξουμε για είσοδο 25 seeds. Από την αλλαγή, θα εξετάσουμε την μέθοδό μας με 3 διαφορετικές εκδοχές: με είσοδο 25, 10 και 1 seed(s). Ως reward function, χρησιμοποιούμε στη μέθοδό μας το εκπαιδευμένο AverageLSTM.

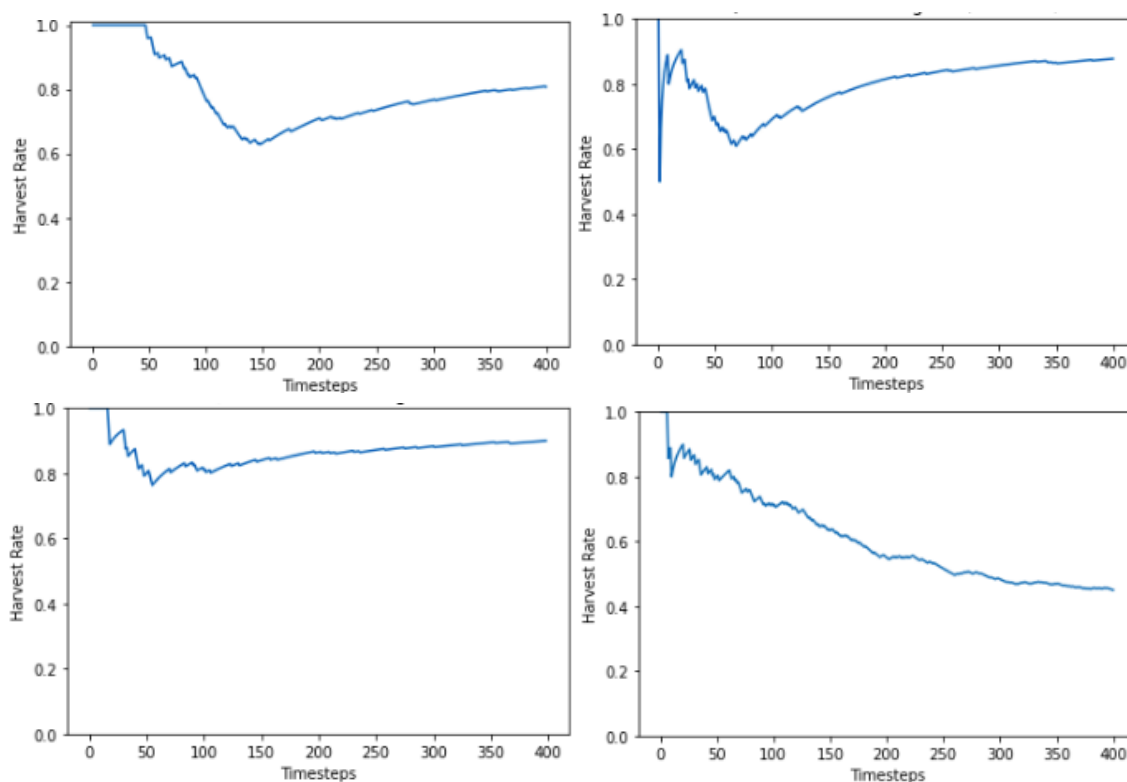
5.1.3.1 Σύγκριση μεθοδολογιών

Βασικός μας σκοπός σε αυτό το πρόβλημα είναι ακόμα και με είσοδο 1 seed να ξεπεράσουμε τον baseline. Τρέχουμε και τις δύο μεθοδολογίες με μέγιστο όριο χρονικών βημάτων ίσο με 400. Έτσι, φέρνουμε και στις δύο περιπτώσεις 400 κείμενα. Στο baseline το harvest rate είναι ίσο με **45%**. Από την άλλη, ο Focused Crawler με General Representation αναπαράσταση έχει τα εξής αποτελέσματα: για 25 seeds έχουμε harvest rate ίσο με **90%**, για 10 seeds έχουμε harvest rate ίσο με **87.8%** και για 1 seed έχουμε harvest rate ίσο με **80.8%**. Στον Πίνακα 5.4 παρουσιάζουμε συγκεντρωτικά τα αποτελέσματα για το γενικό πρόβλημα αναζήτησης.

Focused Crawler	Seeds	Harvest Rate	Timesteps
Random Crawling	25	45%	400
General Representation	25	90%	400
General Representation	10	87.8%	400
General Representation	1	80.8%	400

Πίνακας 5.4: Αποτελέσματα των focused crawlers για τον Γενικό Πρόβλημα αναζήτησης

Στο 5.2, παρουσιάζουμε τις γραφικές παραστάσεις των harvest rates των αποτελεσμάτων όλων των μεθοδολογιών.



Σχήμα 5.2: Διαγράμματα των συσσωρευτικών Harvest Rates του Γενικού Προβλήματος.

Πάνω Αριστερά: Focused Crawler με General Representation και **1 seed**

Πάνω Δεξιά: Focused Crawler με General Representation και **10 seeds**

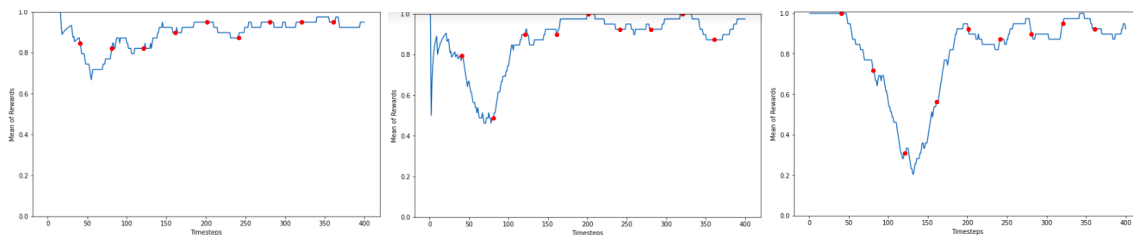
Κάτω Αριστερά: Focused Crawler με General Representation και **25 seeds**

Κάτω Δεξιά: Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο και **25 seeds**

5.1.3.2 Συμπεράσματα

Συμπεραίνουμε πως ο Focused Crawler με General Representation έλυσε αποτελεσματικά το πρόβλημα, αφού μας έφερε και στις τρεις περιπτώσεις παραπάνω από 320 (στα 400) σχετικά κείμενα. Βλέπουμε επίσης, πως ακόμα και με 1 seed καταφέρνει να είναι περίπου **1,8 φορές** πιο αποτελεσματικός από τον focused crawler με τυχαίο πιλοτικό τρέξιμο. Παρατηρούμε, πως στην περίπτωση του 1 seed, ο πράκτορας χρειάζεται περισσότερα χρονικά βήματα (περίπου 150) προκειμένου να μάθει να φάχνει αποτελεσματικά. Αντίθετα, οι άλλες δυο περιπτώσεις χρειάζονται λίγο παραπάνω από 50 χρονικά βήματα, όπου και συμβαίνει και η αφετηρία της μάθησης. Σημειώνουμε πως για τα πρώτα 40 χρονικά βήματα ο πράκτορας επιλέγει με τυχαίο τρόπο κείμενα. Δεδομένου πως σε κάθε χρονικό βήμα μετράμε το συσσωρευτικό harvest rate, δεν πρέπει να λαμβάνουμε υπόψιν τυχόν μεγάλες τιμές αυτού για τα αρχικά χρονικά βήματα.

Στο 5.3, βλέπουμε ότι και για τρεις διαφορετικές εισόδους ο πράκτορας μαθαίνει αρκετά αποτελεσματικά το πρόβλημα, με το μέσο reward να είναι σχεδόν στο 1.



Σχήμα 5.3: Διαγράμματα του μέσου reward με κυλιόμενο παράθυρο ίσο με 40 χρονικά βήματα συναρτήσει των χρονικών βημάτων.

Αριστερά: Focused Crawler με General Representation και **25 seeds**

Μέση: Focused Crawler με General Representation και **10 seeds**

Δεξιά: Focused Crawler με General Representation και **1 seed**

Οι κόκκινες βούλβες αναπαριστούν τα χρονικά βήματα στα οποία ενημερώνεται το Target Q-Network

Στον Πίνακα 5.5 εμφανίζονται οι υπερπαράμετροι που χρησιμοποιήσαμε για τον Focused Crawler με General Representation. Ως αρχικό batch size χρησιμοποιούμε το 16. Σημειώνουμε πως μεταβάλλουμε την τιμή του batch size με το πέρασμα των χρονικών βημάτων ως εξής:

$$batch_size = \max(batch_size, batch_size \cdot \log_{10} timestep) \quad (5.1)$$

Έτσι, όσο αυξάνονται οι εμπειρίες στο Experience Replay, θέλουμε να αυξάνεται ελάχιστα και το μέγεθος του batch, σε κάθε εκπαίδευση του πράκτορα. Με αυτόν τον τρόπο, υπάρχει αυξημένη πιθανότητα το μοντέλο να επανεκπαιδευτεί σε κάθε εμπειρία, που περιέχεται στο Experience Replay.

seeds	25, 10, 1
initial batch size	16
total timesteps	400
γ	0.99
replay start size	40
target update period (TUP)	40
initial epsilon value	0.15
epsilon decay	0.99
min epsilon value	0.08
clipvalue	1
optimizer	RMSProp
learning rate	0.0001
number of hidden layers	3
input dimension	403
hidden layer dimensions	[202, 101, 51]
hidden layer activation function	ReLU
output layer activation function	Identity function

Πίνακας 5.5: Υπερπαράμετροι Γενικού Προβλήματος Δεύτερου Σταδίου

Ακόμα, για τα Q-Networks χρησιμοποιούμε MLP με 4 hidden layers. Ως optimizer χρησι-

μποποιούμε τον RMSProp, με σταθερό learning rate ίσο με 0.0001. Επίσης, χρησιμοποιούμε clipvalue ίσο με 1, προκειμένου να κόβουμε τιμές των παραγώγων (gradients), των οποίων η απόλυτη τιμή τους είναι μεγαλύτερη από το 1. Αυτό συστήνεται για όλες τις εφαρμογές των Deep Q-Learning αλγορίθμων [8]. Όσον αφορά τις πιο τυπικές υπερπαραμέτρους της Ενισχυτικής Μάθησης, χρησιμοποιούμε αρχικό ϵ , του ϵ -greedy, ίσο με 0.15 το οποίο φθίνει έως την τιμή 0.08 με ρυθμό πτώσης 0.99. Ορίζουμε τον παράγοντα έκπτωσης γ ίσο με 0.99, δεδομένου πως έχουμε να αντιμετωπίσουμε ένα πρόβλημα χωρίς επεισόδια. Τέλος, η αφετηρία της εκπαίδευσης του πράκτορα συμβαίνει στο 40στο χρονικό βήμα, ενώ η ανανέωση του Target Q-Network συμβαίνει κάθε 40 χρονικά βήματα.

5.2 Ειδικό Πρόβλημα Αναζήτησης

Αφού έχουμε παρουσιάσει πλήρως τα αποτελέσματα του γενικού προβλήματος αναζήτησης, στο σημείο αυτό θα δείξουμε την επίδοση του focused crawler σε ένα πιο ειδικό θέμα. Οι εμπειρογνώμονες πεδίου (domain experts) μας όρισαν το πρόβλημα της αναζήτησης εθνοφαρμακολογικής μελέτης, που σχετίζεται με συγκεκριμένα γεωγραφικά πλαίσια, οικογένειες και κατηγορίες φυτών.

Πιο συγκεκριμένα, θεωρούμε πως το σχετικό θέμα μελέτης αφορά εθνοφαρμακολογική μελέτη στην Ελλάδα, τα Βαλκάνια, την Κύπρο, την Τουρκία, το Ιράν και το Ιράκ. Οι επιθυμητές οικογένειες φυτών περιλάμβαναν αλφαθητικά τις εξής: *Alliaceae, Anacardiaceae, Apiaceae, Asparagaceae, Asphodelaceae, Asteraceae, Boraginaceae, Brassicaceae, Cactaceae, Capparaceae, Cannabaceae, Caryophyllaceae, Cistaceae, Fabaceae, Fagaceae, Gentianaceae, Honey, Hypericaceae, Lamiaceae, Liliaceae, Malvaceae, Mastic Gum, Mastic Resin, Mediterenean Diet, Moraceae, Oleaceae, Paeoniaceae, Pine Resin, Platanaceae, Royal Jelly, Rosaceae, Salicaceae, Scrophulariaceae, Solanaceae, Urticaceae, Valerianaceae, Vitaceae*. Πρόκειται για σημαντικές ερευνητικά οικογένειες και κατηγορίες φυτών ως προς την εθνοφαρμακολογική τους χρήση στον Ελλαδικό και τον ευρύτερο Βαλκανικό χώρο.

Ωστόσο, προκειμένου να έχουμε μια πιο σφαιρική εικόνα του προβλήματος αναζήτησης, οι εμπειρογνώμονες πεδίου μας όρισαν επιμέρους επίπεδα συνάφειας των βιβλιογραφικών αναφορών τα οποία ήταν τα εξής: (1) πολύ σχετικά (Highly Relevant), δημοσιεύσεις που αφορούν την εθνοφαρμακολογία στην Ελλάδα, τα Βαλκάνια, την Τουρκία κλπ, και σχετίζονται με τις επιθυμητές οικογένειες/κατηγορίες φυτών, (2) σχετικά (Relevant), δημοσιεύσεις που αφορούν την εθνοφαρμακολογία στην Ελλάδα, τα Βαλκάνια και την Τουρκία, αλλά δεν σχετίζονται με τις επιθυμητές οικογένειες φυτών, (3) μερικώς σχετικά (Partially Relevant), δημοσιεύσεις που αφορούν την εθνοφαρμακολογία γενικά, και (4) μη σχετικά (Irrelevant), δημοσιεύσεις που δεν αφορούν καθόλου την εθνοφαρμακολογία. Πρακτικά, από τη μεριά του focused crawling, θεωρήσαμε τα επίπεδα (1), (2) ως σχετικά (relevant) και τα (3), (4) ως μη σχετικά (irrelevant).

5.2.1 Προετοιμασία Πειράματος

Αφού πλέον ορίσαμε τα θέματα της αναζήτησης, οι εμπειρογνώμονες εθνοφαρμακολογίας διαλέγουν 25 URLs δημοσιεύσεων, που θεωρούν πως ανήκουν στην Highly Relevant κατηγορία συνάφειας. Η διαδικασία αυτή, ωστόσο, πραγματοποιείται με τρόπο, ώστε ο κάθε εμπειρογνώμονας να μας δώσει περίπου τον ίδιο αριθμό δημοσιεύσεων, χωρίς να γνωρίζει τα κείμενα που μας έδωσε ο άλλος. Σημειώνουμε πως τα URLs αυτά συμπίπτουν με το προηγούμενο γενικό πρόβλημα αναζήτησης.

Στη συνέχεια, πρέπει να ελέγξουμε και πάλι την συμφωνία των επισημειωτών (Interannotator Agreement) στο νέο πρόβλημα, προκειμένου να μελετήσουμε την αξιοπιστία των επισημειώσεων που θα λαμβάνει το σύστημα στο πρώτο στάδιο της Ενεργού Μάθησης. Διαλέγοντας όλα τα seeds URLs και τυχαία δείγματα από το σύνολο δεδομένων μας, ρωτάμε τους επισημειωτές και πάλι για 50 συνολικά URLs και μελετάμε ξανά τις τρεις μετρικές: το Raw Agreement, το Cohen's kappa και το Krippendorff's alpha. Στους πίνακες 5.6 και 5.9 παρουσιάζονται τα αναλυτικά αποτελέσματα των επισημειώσεων αλλά και των τιμών των μετρικών, που προέκυψαν αντίστοιχα. Στο πρόβλημα αυτό, και για τις τρεις μετρικές βλέπουμε πως έχουμε σχεδόν ικανοποιητική συμφωνία δεδομένου πως το Cohen's kappa θεωρείται ουσιώδες (substantial) για τιμές μεγαλύτερες από 0.6 και το Krippendorff's alpha άριστο για τιμές μεγαλύτερες από 0.8. Έτσι, μπορούμε να συνεχίσουμε με τις επισημειώσεις από τους ειδικούς.

Raw Agreement	0.82
Cohen's Kappa	0.71
Krippendorff's alpha	0.92

Πίνακας 5.6: *Μετρικές Interannotator Agreement Ειδικού Προβλήματος*

		Annotator 2			
		HR	R	PR	IR
Annotator 1	HR	19	1	0	0
	R	4	0	0	2
	PR	0	0	3	0
	IR	0	0	2	19

Πίνακας 5.7: *Confusion Matrix Επισημειώσεων Ειδικού Προβλήματος*

Εκτελούμε παρόμοια διαδικασία με το γενικό πρόβλημα ακολουθώντας τα εξής βήματα: Παίρνουμε τα 427 λινκς που εξήχθησαν από τα seeds URLs. Από αυτά είναι 154 σχετικά, 253 μερικώς και 20 μη σχετικά. Διεξάγουμε δυο τυχαία πιλοτικά τρεξίματα (Αλγόριθμος 4.1). Στο πρώτο φέρνουμε: 35 σχετικά, 150 μερικώς και 215 μη σχετικά. Στο δεύτερο: 32 σχετικά, 161 μερικώς και 207 μη σχετικά. Αφαιρώντας τα διπλότυπα (duplicates) φθάνουμε σε συνολικό σύνολο δεδομένων (χωρίς να μετράμε τα 25 seeds URLs): 202 σχετικά, 474 μερικώς σχετικά και 336 μη σχετικά. Σημειώνουμε πως εφόσον θεωρήσαμε δύο κατηγορίες με σχετικές κλάσεις, τα παραπάνω 202 σχετικά κείμενα που φέραμε είναι είτε πολύ σχετικά

ή απλώς σχετικά με το θέμα αναζήτησης. Έτσι, ουσιαστικά έχουμε θέσει σαν πρώτη προτεραιότητα της αναζήτησης την εύρεση κειμένων με βάση τα προαναφερθέντα γεωγραφικά πλαίσια και σαν δεύτερη προτεραιότητα τις οικογένειες και κατηγορίες φυτών.

Αφού έχουμε ολοκληρώσει την προεργασία του *focused crawling*, είμαστε έτοιμοι να ορίσουμε τις διαφορετικές μεθοδολογίες που θα μελετήσουμε. Αρχικά θα δοκιμάσουμε έναν *Focused Crawler* με *Only NLP Representation*, ο οποίος θα λειτουργήσει ως *baseline* μεθοδολογία. Στη συνέχεια, δεδομένου πως το πρόβλημά μας έχει εξ ορισμού λέξεις-κλειδιά, θα εξετάσουμε την περίπτωση του *DetFC* πάνω στο θέμα αναζήτησής μας. Τέλος, θα εξετάσουμε την περίπτωση του *Focused Crawler* με *Keyword Representation*. Όλες οι μεθοδολογίες θα εξεταστούν με είσοδο 25 *seeds URLs*, ενώ οι δύο με τις καλύτερες επιδόσεις θα μελετηθούν και με 10 και 1 *seeds URLs* για είσοδο αντίστοιχα.

5.2.2 Ενεργός Μάθηση

Ξεκινάμε με το πρώτο στάδιο του *focused crawler*. Έχουμε να αντιμετωπίσουμε ένα πρόβλημα ταξινόμησης με τρεις κλάσεις μέσω Ενεργού Μάθησης. Θα χρησιμοποιήσουμε τα μοντέλα που είδαμε και στο γενικό πρόβλημα.

Ωστόσο, δεδομένου πως μας ενδιαφέρουν σε μεγάλο βαθμό οι λέξεις που αντιπροσωπεύουν οι χώρες, για την ταξινόμηση, αντικαθιστούμε σε όλα τα κείμενα όλες τις λέξεις χωρών που δεν περιέχονται στους γεωγραφικούς περιορισμούς του προβλήματος με την λέξη "Togo". Αντίθετα, αντικαθιστούμε όλες τις λέξεις χωρών που περιέχονται στα επιθυμητά γεωγραφικά πλαίσια με την λέξη "Greece". Αυτή η αντικατάσταση προϋποθέτει πλήρη γνώση της λίστας των λέξεων που υποδεικνύουν σχετικότητα (βάσει χώρας). Άρα, αξιοποιούμε εξωτερική γνώση στην προεπεξεργασία του πειράματος. Αυτό το κάνουμε γιατί αν και οι λέξεις των χωρών παίζουν μεγάλη σημασία στο παραπάνω πρόβλημα ταξινόμησης, οι χώρες που ανήκουν - ή δεν ανήκουν - στα συγκεκριμένα γεωγραφικά πλαίσια δεν διαφέρουν μεταξύ τους όσον αφορά την ταξινόμηση αυτή. Το ίδιο ακριβώς κάνουμε και με τις εθνικότητες, μόνο όσον αφορά τις επιθυμητές χώρες, αντικαθιστώντας με την λέξη "Greek". Η χρήση της λέξης "Togo" προήλθε από το γεγονός πως το αντίστοιχο *word embedding* της έχει αρκετά μικρό *cosine similarity* με εκείνο της λέξης "Greece".

Στο σημείο αυτό, θα δοκιμάσουμε να λύσουμε το πρόβλημα ταξινόμησης με Ενεργό Μάθηση, έχοντας μέγιστο όριο (*budget*) ερωτήσεων και πάλι τις 250 επισημειώσεις. Θα ακολουθήσουμε δύο μεθοδολογίες. Αρχικά, θα εξετάσουμε την περίπτωση η ταξινόμηση να περιέχει 3 διαφορετικές κλάσεις: (α) σχετικά κείμενα, (β) μερικώς σχετικά κείμενα και (γ) μη σχετικά κείμενα. Η (α) περιλαμβάνει τα *Highly Relevant* και τα *Relevant* που αναφέραμε παραπάνω. Στην άλλη περίπτωση, θα μελετήσουμε το πρόβλημα σαν δυαδική ταξινόμηση, με κλάσεις: (α) σχετικά κείμενα και (β) μη σχετικά. Τα τελευταία περιλαμβάνουν και τα μερικώς σχετικά κείμενα.

Και στις δύο αυτές μεθοδολογίες ακολουθούμε την ίδια λογική με το γενικό πρόβλημα αναζήτησης. Εκτελούμε *k-fold-cross-validation* ώστε να δημιουργήσουμε *k* τριάδες συνόλων δεδομένων. Θέλουμε το αρχικό σύνολο εκπαίδευσης να περιέχει 50 δεδομένα σε σύνολο, από τα οποία τα μισά να είναι σχετικά. Ωστόσο, λόγω περιορισμένου αριθμού διαθέσιμων Σχετικών κειμένων χρησιμοποιούμε $k = 4$.

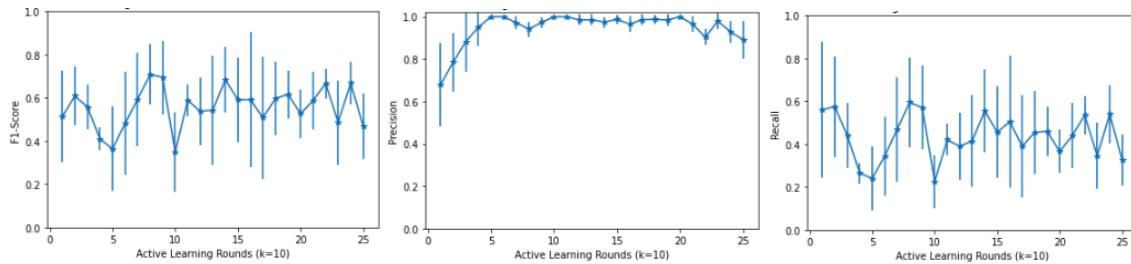
Αξίζει να τονίσουμε πως έχουμε ανισορροπία του αρχικού συνόλου δεδομένων, με τα σχετικά κείμενα να εμφανίζονται σε ποσοστό περίπου 20%. Έτσι, για όλα τα μοντέλα θα χρησιμοποιήσουμε θετικά βάρη για κάθε κλάση, στις αντίστοιχες συναρτήσεις κόστους, με βάση τις αναλογίες κάθε χρονικής στιγμής στο τρέχον σύνολο εκπαίδευσης. Πιο αναλυτικά, θα χρησιμοποιήσουμε την τετραγωνική ρίζα του πηλίκου του μεγέθους του συνόλου εκπαίδευσης με το πλήθος των δεδομένων της αντίστοιχης κλάσης. Για παράδειγμα, η κλάση των σχετικών κειμένων, θα χρησιμοποιεί για τον k -στο γύρο της Ενεργού Μάθησης το θετικό βάρος: $w = \sqrt{\frac{\text{Number of training data in } k^{\text{th}} \text{ AL round}}{\text{Number of relevant training data in } k^{\text{th}} \text{ AL round}}}$. Με τον ίδιο τρόπο υπολογίζονται και τα βάρη των υπόλοιπων κλάσεων.

Επιπλέον, λαμβάνοντας υπόψιν αυτήν την ανισορροπία του συνόλου εκπαίδευσης, για όλα τα μοντέλα θα εξετάσουμε τις μετρικές Precision, Recall και F₁-score για την κλάση των σχετικών κειμένων. Αυτό γιατί στην πραγματικότητα μόνο αυτή μας ενδιαφέρει ως προς την αποτελεσματικότητα (effectiveness) της Ενεργού Μάθησης. Δεδομένου πως το πρώτο στάδιο σχεδιάζουμε να το χρησιμοποιήσουμε ως reward function, είναι αναγκαίο τόσο το precision όσο και το recall να είναι όσο καλύτερα μπορούμε. Θεωρούμε πως το precision πρέπει να είναι κατά ελάχιστο 90%, ώστε η πιθανότητα των False Positives στην πρόβλεψη σχετικών κειμένων να είναι περίπου 1 στα 10 κείμενα. Ακόμα θέλουμε το recall να είναι πάνω από 50%, ώστε από 10 σχετικά κείμενα να μπορούμε να αναγνωρίσουμε τουλάχιστον τα 5. Υπογραμμίζουμε πως το precision σε αυτό το πρόβλημα θέλουμε να είναι αρκετά υψηλό, καθώς σε αντίθετη περίπτωση ο πράκτορας της Ενισχυτικής Μάθησης θα μαθαίνει να επισκέφεται ιστοσελίδες που θεωρεί πως είναι σχετικές, ενώ στην πραγματικότητα δεν είναι. Με άλλα λόγια, εάν κρατήσουμε ένα μοντέλο με πιο χαμηλό precision, τότε ο focused crawler θα λαμβάνει ως σχετικά πολύ περισσότερα κείμενα από ότι πρέπει και, άρα, θα καταλήξουμε τελικά σε ένα πλασματικά μεγάλο harvest rate.

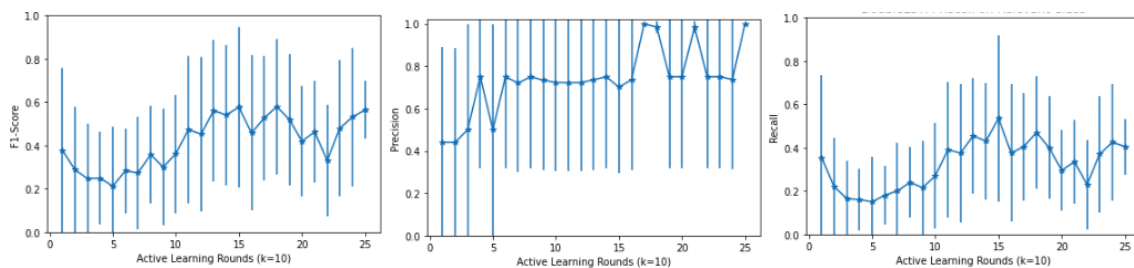
5.2.2.1 Ταξινόμηση με 3 κλάσεις

Στην περίπτωση που έχουμε 3 κλάσεις, θέλουμε το αρχικό σύνολο εκπαίδευσης να περιέχει 12 μερικώς σχετικά κείμενα και 13 μη σχετικά. Το test set περιέχει 100 κείμενα από τα οποία είναι 50 σχετικά, 25 μερικώς σχετικά και 25 μη σχετικά. Τα υπόλοιπα κείμενα, σε κάθε fold, ανήκουν στο αρχικό pool. Σε αυτήν την περίπτωση, εξετάζουμε τα καλύτερα μοντέλα του γενικού προβλήματος, δηλαδή τα AverageLSTM και DoubleLSTM. Σημειώνουμε πως η χρήση του Margin SVM εδώ αναμένουμε να μην έχει καθολού ενθαρρυντικά αποτελέσματα, δεδομένου πως στα περισσότερα σχετικά κείμενα περιμένουμε η ύπαρξη συνδυασμών των λέξεων κλειδιών να είναι σημαντική. Αυτή η πληροφορία όμως χάνεται με την χρήση του Mean Pooling - Max Pooling.

Τα αποτελέσματα από την εκπαίδευση των μοντέλων αυτών με cross validation φαίνονται στα Σχήματα 5.4 και 5.5. Όπως βλέπουμε κανένα από τα δύο μοντέλα δεν καταφέρνει να δώσει ενθαρρυντικά αποτελέσματα κυρίως ως προς το recall. Ακόμα, παρατηρούμε, από την γραφική παράσταση του recall και του F1-score, πως κανένα μοντέλο δεν συγκλίνει (ταλαντώνεται) ακόμα και όταν έχουν εξαντλήσει όλες τις διαθέσιμες ερωτήσεις προς τον Oracle. Έτσι θα δοκιμάσουμε την ταξινόμηση με χρήση 2 κλάσεων.



Σχήμα 5.4: Διαγράμματα *AverageLSTM* πάνω στο μέσο *F1-score* (αριστερά), στο μέσο *Precision* (μέση) και στο μέσο *Recall* (δεξιά) με *Τυπική Απόκλιση* για την σχετική κλάση, συν τω χρόνω.



Σχήμα 5.5: Διαγράμματα *DoubleLSTM* πάνω στο μέσο *F1-score* (αριστερά), στο μέσο *Precision* (μέση) και στο μέσο *Recall* (δεξιά) με *Τυπική Απόκλιση* για την σχετική κλάση, συν τω χρόνω.

Παρ' όλα αυτά, αξίζει να υπολογίσουμε τα εκτιμώμενα precision, recall και *F1-Score* στην περίπτωση ενός τυχαίου (**stratified dummy**) ταξινομητή που προβλέπει βάσει την a priori πιθανότητα κάθε κλάσης. Οι a priori πιθανότητες, που προκύπτουν από το pool, είναι: περίπου 20% για την κλάση των σχετικών (R), περίπου 47% για την κλάση των μερικώς σχετικών (PR), και περίπου 33% για την κλάση των μη σχετικών (IR). Κατασκευάζουμε, έτσι, τον confusion matrix, Πίνακας 5.8, με τα αποτελέσματα που θα παίρναμε, εάν δοκιμάζαμε τον stratified dummy ταξινομητή πάνω στο pool.

		Actual Values		
		R	PR	IR
Predicted Values	R	41	94	67
	PR	94	222	158
	IR	67	158	111

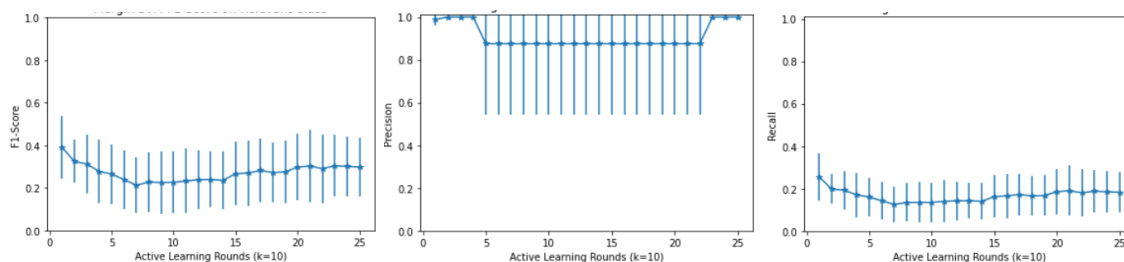
Πίνακας 5.8: *Confusion Matrix* πάνω στο pool για 3 κλάσεις

Από τα αποτελέσματα αυτά, για την κλάση των σχετικών κειμένων (R), έχουμε: **precision=recall=F1-Score≈20%**. Έτσι, βλέπουμε πως οι παραπάνω μέθοδοι Ενεργού Μάθησης, αν και δεν έφτασαν τα επιθυμητά επίπεδα που είχαμε ορίσει, ξεπερνούν κατά πολύ τον τυχαίο (stratified dummy) ταξινομητή από τους πρώτους 10 γύρους.

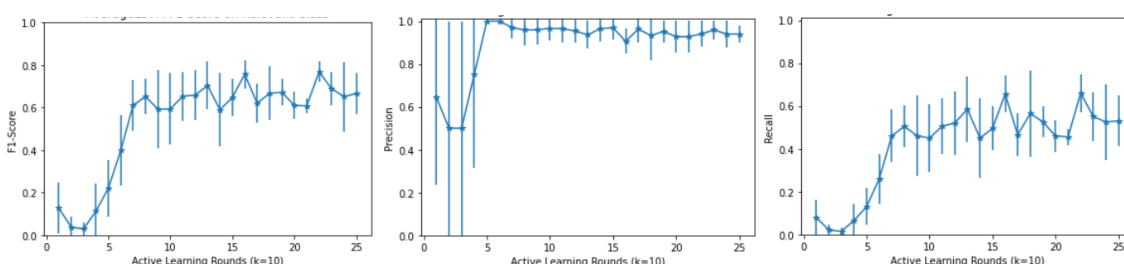
5.2.2.2 Ταξινόμηση με 2 κλάσεις

Στην περίπτωση με 2 κλάσεις, διατηρούμε τις αναλογίες στα σύνολα δεδομένων που έχουμε και παραπάνω, με την διαφορά πως τα μερικώς σχετικά και τα μη σχετικά κείμενα

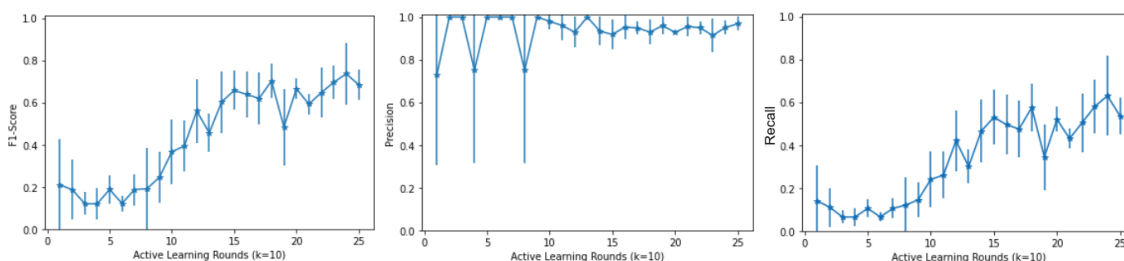
ανήκουν πλέον στην ίδια κλάση. Εδώ δοκιμάζουμε και τον Margin SVM. Στα Σχήματα 5.6, 5.7, 5.8, απεικονίζουμε τις επιδόσεις των μοντέλων στις τρεις μετρικές.



Σχήμα 5.6: Διαγράμματα Margin SVM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση για την σχετική κλάση



Σχήμα 5.7: Διαγράμματα AverageLSTM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση, συν το χρόνο, για την σχετική κλάση.



Σχήμα 5.8: Διαγράμματα DoubleLSTM πάνω στο μέσο F1-score (αριστερά), στο μέσο Precision (μέση) και στο μέσο Recall (δεξιά) με Τυπική Απόκλιση, συν το χρόνο, για την σχετική κλάση.

Λαμβάνουμε ότι το Margin SVM δεν αποδίδει καθόλου, καθώς το μέσο recall είναι σε όλους τους γύρους κάτω από 40%. Όσον αφορά το AverageLSTM βλέπουμε πως τις καλύτερες επιδόσεις τις έχει στον **22ο γύρο**. Από την άλλη το DoubleLSTM τις έχει στον **24ο γύρο**. Έτσι, βλέπουμε πως από άποψη χρήσης λιγότερων κειμένων το AverageLSTM χρησιμοποίησε 20 κείμενα λιγότερα από το DoubleLSTM. Στον Πίνακα 5.9 απεικονίζουμε αναλυτικά τις τιμές τους για τους γύρους αυτούς.

Classifier	Best AL Batch	Mean F ₁ -score	Mean Precision	Mean Recall
AverageLSTM	22	0.769	0.940	0.660
DoubleLSTM	24	0.736	0.950	0.630

Πίνακας 5.9: Αποτελέσματα Ενεργού Μάθησης στο Ειδικό Πρόβλημα με 2 κλάσεις (ακρίβεια τριών μηδενικών)

Συγκρίνοντας αυτά τα αποτελέσματα, παρατηρούμε πως το AverageLSTM είναι καλύτερο σε όλα εκτός από το precision. Παρ' όλα αυτά, η διαφορά τους είναι ελάχιστη της τάξης του 1%. Σημειώνουμε πως επειδή δεν έχουμε πολλές παρατηρήσεις λόγω του 4-fold-cross-validation δεν προτείνεται να κάνουμε τα στατιστικά τεστς που είδαμε στο γενικό πρόβλημα.

Ωστόσο, αν και τα αποτελέσματα φαίνονται αρκετά ενθαρρυντικά, δεδομένου του εξειδικευμένου ορισμού των σχετικών κείμενων στο πρόβλημα, μελετάμε την επίδοση του καλύτερου AverageLSTM πάνω στο αρχικό σύνολο δεδομένων επαυξημένου με τα πιο αντιπροσωπευτικά seeds. Τα τελευταία, αυθαίρετα, θεωρήσαμε πως είναι εκείνα, των οποίων οι τίτλοι ή/και η περίληψη περιείχαν τουλάχιστον μία λέξη-κλειδί. Τα αποτελέσματα που πήραμε ήταν τα εξής: **Precision = 0.46, Recall = 0.70** και **F₁-score = 0.56** για την κλάση των σχετικών κειμένων. Η αποτυχία αυτή οφείλεται επίσης στο γεγονός πως είχαμε αρκετά λίγα κείμενα για ένα δύσκολο πρόβλημα ταξινόμησης. Βλέποντας πως το precision είναι αρκετά χαμηλότερο από 90%, δεν θα χρησιμοποιήσουμε τελικά κανένα από αυτά τα μοντέλα σαν reward function του δεύτερου σταδίου της Ενισχυτικής Μάθησης.

5.2.3 Καθορισμός του Relevance Calculator

Θυμίζουμε ότι αρχική μας ιδέα ήταν να χρησιμοποιούσαμε ένα μοντέλο εκπαιδευμένο με Ενεργό Μάθηση ως AL Relevance Calculator (βλ. Σχήμα 4.1). Αντ' αυτού, ως Relevance Calculator χρησιμοποιούμε εκείνον του DetFC, έχοντας ως προτεραιότητες τις εξής: $P_1 = 10$, εάν ένα κείμενο περιέχει τουλάχιστον μία από τις λέξεις-κλειδιά των χωρών και εθνικοτήτων, $P_2 = 5$, εάν ένα κείμενο περιέχει τουλάχιστον μία από τις λέξεις-κλειδιά των οικογενειών και κατηγοριών φυτών ή τουλάχιστον μία από τις λέξεις-κλειδιά ορισμένων γενικών όρων του προβλήματος αναζήτησης. Τις τελευταίες τις αποκτήσαμε ρωτώντας τους ειδικούς για το τι λέξεις-κλειδιά θα έδιναν σε μια μηχανή αναζήτησης. Αυτές οι λέξεις-κλειδιά είναι οι εξής: *Ethnopharmacology, Ethnobotany, herb, medical plant*.

Επιπλέον, θεωρώντας πως διαθέτουμε μία εξωτερική πηγή που παίρνει ως είσοδο μία λέξη - ουσιαστικό - και εξάγει αυτόματα τα άμεσα επίθετα παράγωγά της, προσθέτουμε και τις λέξεις-κλειδιά: *ethnopharmacological, ethnobotanical* και *herbal*. Στην περίπτωση μας, πήραμε ως εξωτερική πηγή, για την εξαγωγή των επιθέτων, το WordWeb Online [75]. Όσον αφορά τις λέξεις-κλειδιά των οικογενειών και κατηγοριών φυτών, τις επεκτείνουμε κι άλλο με τις λέξεις που αντιστοιχούν στα διάφορα είδη των κατηγοριών αυτών, χρησιμοποιώντας τις πληροφορίες που παρέχει η ιστοσελίδα Flora of Greece [76]. Σημειώνουμε πως τον σύνδεσμο αυτής μας τον σύστησαν οι εμπειρογνώμονες πεδίου.

Τέλος, όσον αφορά τις λέξεις-κλειδιά των χωρών, είχαμε λάβει ως είσοδο τις λέξεις *Greece, Balkan Countries, Turkey, Iran, Iraq, Cyprus* και από την μεριά μας προσθέσαμε τις λέξεις-κλειδιά *Greek, Serbia, Serbian, Romania, Romanian, Croatia, Croatian, Bosnia and Herzegovina, Bosnian, Slovenia, Slovenian, Macedonia, Macedonian, Albania, Albanian, Turkish, Cypriot, Iraqi, Iranian, Kosovo* αντλώντας αυτόματα πληροφορίες από το English Club [77] και το Wikitravel [78]. Κατά αυτόν τον τρόπο, θεωρούμε ότι έχουμε αρκετές λέξεις-κλειδιά, ώστε να περιγράψουμε ντετερμινιστικά εάν ένα κείμενο είναι σχετικό με το θέμα αναζήτησης. Από εδώ και στο εξής θα αναφερόμαστε σε αυτόν τον Relevance Calculator ως Ethnopharmacological Relevance Calculator ή **ERC**.

Το μόνο που μένει είναι να τεστάρουμε την επίδοση του ERC. Για αυτόν τον λόγο, τον δοκιμάζουμε πάνω στο αρχικό σύνολο των 1012 δεδομένων και στα αντιπροσωπευτικά seeds, όπως και προηγουμένως με την Ενεργό Μάθηση. Τα αποτελέσματα αυτού είναι: **Precision = 0.95, Recall = 0.61** και **F₁-score = 0.75**. Βλέπουμε πως έχουμε precision και recall αρκετά υψηλότερα από τα ελάχιστα όρια που είχαμε θέσει, δηλαδή 90% και 50% αντίστοιχα.

Τονίζουμε πως αν χρησιμοποιήσουμε τον ERC για reward function, τότε στο τέλος του τρεξίματος του focused crawler, το harvest rate αναμένουμε να είναι μια ελάχιστη εκτίμηση της πραγματικής του τιμής. Αυτό γιατί, όπως είδαμε έχουμε πολύ υψηλό precision και μέτριο recall. Έτσι, προτιμούμε να έχουμε έναν λιγότερο αισιόδοξο εκτιμητή της συνάφειας παρά έναν πολύ αισιόδοξο. Συγκεντρωτικά, τα αποτελέσματα μας τα αναπαριστούμε στον Πίνακα 5.10.

Relevance Calculator	Mean F ₁ -score	Mean Precision	Mean Recall
AverageLSTM with AL	0.56	0.46	0.70
ERC	0.75	0.95	0.61

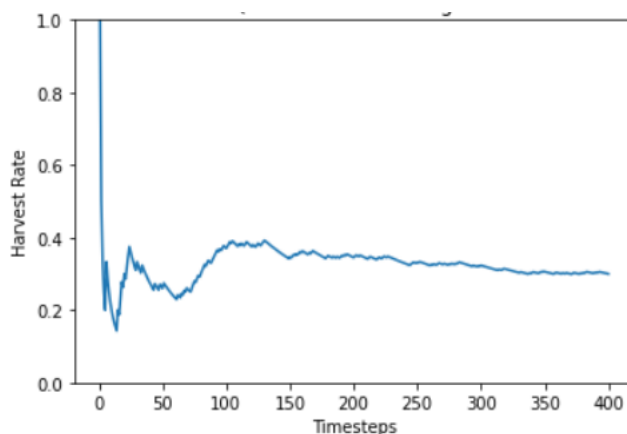
Πίνακας 5.10: Σύγκριση Καλύτερου Μοντέλου Ενεργού Μάθησης με τον ERC πάνω σε όλο το σύνολο δεδομένων

5.2.4 Ενισχυτική Μάθηση

5.2.4.1 Focused Crawler με Only NLP Representation - Baseline

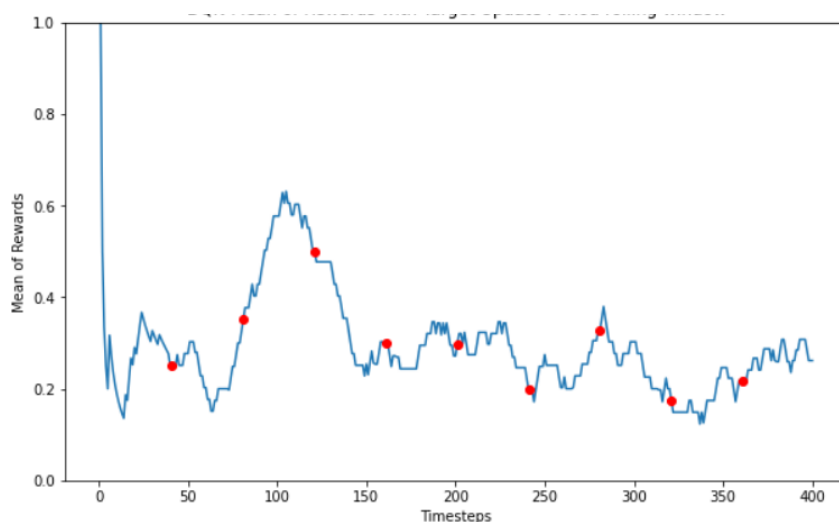
Τρέχοντας τον baseline για είσοδο 25 seeds φέρνουμε 400 κείμενα στην έξοδο. Με άλλα λόγια χρησιμοποιούμε 400 χρονικά βήματα. Χρησιμοποιούμε τον πράκτορα DQN, έχοντας σαν Q-Network ένα MLP. Δεδομένου πως κάθε word embedding είναι ένα διάνυσμα 200 διαστάσεων, τόσο το state όσο και το action χρειάζονται 400 διαστάσεις. Ο αριθμός αυτός προέκυψε γιατί χρησιμοποιήσαμε για την αναπαράσταση mean-max pooling. Άρα κάθε \mathbf{x} είσοδος του MLP είναι ένα διάνυσμα 800 διαστάσεων. Από τα 400 κείμενα που φέραμε ο ERC έδειξε πως μόνο 120 από αυτά ήταν σχετικά, δηλαδή το harvest rate ήταν **30%**.

Όσον αφορά το DQN χρησιμοποιούμε περίοδο ανανέωσης του νευρωνικού στόχου ίση με 40 χρονικά βήματα. Παράλληλα ξεκινάμε την εκπαίδευση του Q-Network μετά από τα πρώτα 40 χρονικά βήματα. Σημειώνουμε πως ο αριθμός των συνολικών χρονικών βημάτων (400) που θέσαμε για το τρέξιμο του baseline είναι σχετικά μικρός σε σχέση με τις υπόλοιπες μεθοδολογίες που θα δούμε. Από την άλλη, όσο αυξάνει ο αριθμός αυτός, περιμένουμε να φθίνει η επίδοση. Με άλλα λόγια, θεωρούμε πως με την μεθοδολογία αυτή το harvest rate δεν θα αυξάνει σημαντικά μετά τα 150 πρώτα χρονικά βήματα, κάτι το οποίο παρατηρούμε και στο Σχήμα 5.9.



Σχήμα 5.9: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Only NLP Representation

Επίσης, στο Σχήμα 5.10 βλέπουμε την γραφική παράσταση των rewards που δέχεται ο πράκτορας, καθόλη την διάρκεια του focused crawling, με κυλιόμενο παράθυρο ίσο με την περίοδο ανανέωσης του νευρωνικού στόχου (Target Q-Network). Στο σχήμα αυτό, βλέπουμε πως ο πράκτορας στην αρχή φαίνεται να φέρνει αρκετά σχετικά κείμενα. Ωστόσο, θεωρούμε πως κάτι τέτοιο συμβαίνει, λαμβάνοντας υπόψιν πως τα διαθέσιμα URLs του στο μέτωπο αναζήτησης προέρχονται σε μεγάλο βαθμό από τα seeds, με αποτέλεσμα αυτά πιθανόν να είναι όντως σχετικά. Από την άλλη, όσο αυξάνονται τα χρονικά βήματα βλέπουμε να έχουμε περίπου σταθερό ρυθμό εύρεσης σχετικών κειμένων, της τάξης του 1 σχετικού στα 5 κείμενα. Αυτό πιστεύουμε ότι γίνεται, διότι ο πράκτορας δυσκολεύεται να αντιληφθεί εάν ένα κείμενο είναι σχετικό (Highly Relevant / Relevant) ή μερικώς σχετικό (Partially Relevant). Έτσι, είναι αναγκαίο να χρησιμοποιήσουμε άλλες αναπαραστάσεις του \mathbf{x} , ώστε να κατευθύνουμε άμεσα τον πράκτορα προκειμένου αυτός να φθάσει στην βέλτιστη στρατηγική, και άρα να κατανοεί καλύτερα το τι πρέπει να ψάχνει.

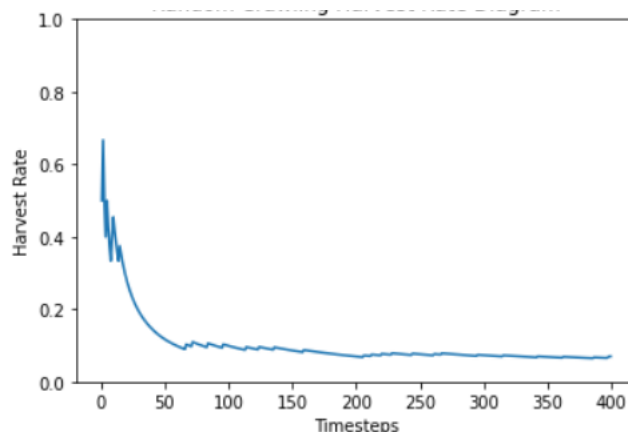


Σχήμα 5.10: Το διάγραμμα του Focused Crawler με Only NLP Representation με κυλιόμενο παράθυρο ίσο με 40 χρονικά βήματα. Οι κόκκινες βούλες συμβολίζουν τα χρονικά βήματα στα οποία έγινε ενημέρωση των βαρών του νευρωνικού στόχου (Target Q-Network).

5.2.4.2 Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο

Αξίζει να δείξουμε πως η baseline μέθοδος έχει σημαντικές διαφορές σε σχέση με έναν πολύ απλό focused crawler χωρίς μάθηση. Για τον σκοπό αυτό, εξετάζουμε τα αντίστοιχα αποτελέσματα ενός focused crawler με τυχαίο πιλοτικό τρέξιμο, χρησιμοποιώντας και πάλι τον ERC για reward function.

Στο Σχήμα 5.11, βλέπουμε το harvest rate με τυχαίο πιλοτικό τρέξιμο. Στα 400 χρονικά βήματα βλέπουμε πως το harvest rate είναι μόλις **7%**. Άρα ο baseline έχει κατά **23%** καλύτερη απόδοση από τον focused crawler με τυχαίο πιλοτικό τρέξιμο.



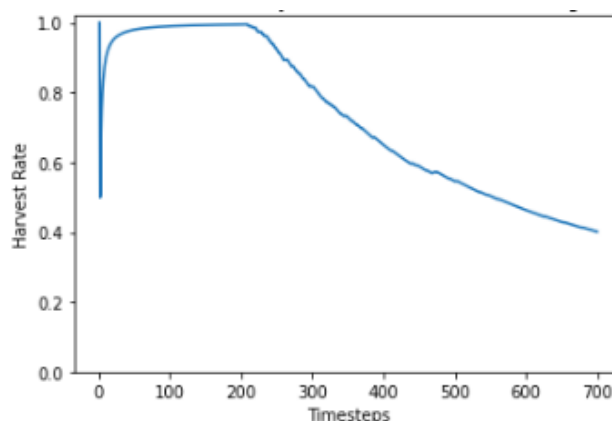
Σχήμα 5.11: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Τυχαίο Πιλοτικό Τρέξιμο

5.2.4.3 Ντετερμινιστικός Greedy Focused Crawler: DetFC

Προχωράμε με την περίπτωση του ντετερμινιστικού focused crawler DetFC. Όπως και προηγούμενος, και εδώ χρησιμοποιούμε τις ίδιες προτεραιότητες των κατηγοριών των λέξεων-κλειδιών. Θυμίζουμε ότι έχουμε $P_1 = 10$, εάν ένα κείμενο περιέχει τουλάχιστον μία από τις λέξεις-κλειδιά των χωρών και εθνικοτήτων, $P_2 = 5$, εάν ένα κείμενο περιέχει τουλάχιστον μία από τις λέξεις-κλειδιά των οικογενειών και κατηγοριών φυτών ή τουλάχιστον μία από τις λέξεις-κλειδιά ορισμένων γενικών όρων εθνοφαρμακολογίας. Οι λέξεις-κλειδιά για όλες τις προτεραιότητες είναι οι ίδιες που ορίστηκαν παραπάνω.

Τρέχουμε τον DetFC με μέγιστο όριο τα 700 χρονικά βήματα και το τελικό harvest rate είναι **40.14%**. Αυτό σημαίνει ότι από τα 700 κείμενα που έφερε τα 336 ήταν σχετικά. Συγκρίνοντάς τον με τον baseline, παρατηρούμε πως το harvest rate στα 400 χρονικά βήματα ήταν **64.83%** φέρνοντας 321 σχετικά κείμενα. Έτσι, ο DetFC βελτιώνει τον baseline κατά 2,161 φορές.

Σημειώνουμε πως δεν έχει νόημα να τρέξουμε τον DetFC για λιγότερα αρχικά seeds, λόγω της μειωμένης δυνατότητάς του να αναγνωρίσει hubs ιστοσελίδες. Λαμβάνουμε υπόψη πως ο DetFC αναγνωρίζει σχετικούς τίτλους που περιέχουν τους κατάλληλους συνδυασμούς από λέξεις-κλειδιά. Άρα, με λιγότερα seeds ο DetFC θα έχει και λιγότερους διαθέσιμους σχετικούς τίτλους να επισκεφτεί, θεωρώντας πως ενδεχομένως να μην ανακαλύψει ποτέ τα URLs των υπολοίπων seeds. Στο Σχήμα 5.12, αναπαριστούμε το διάγραμμα του harvest rate για τον DetFC.



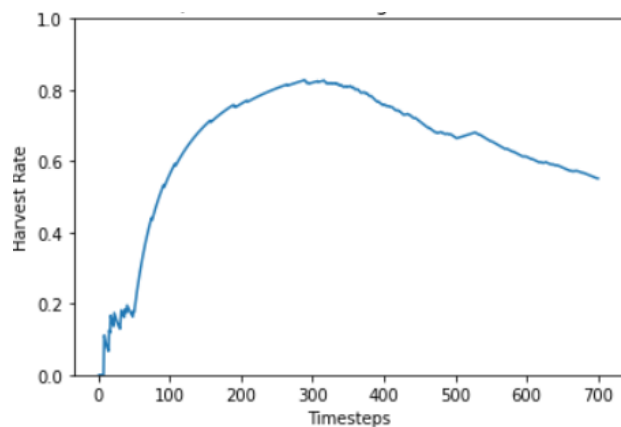
Σχήμα 5.12: Το διάγραμμα του συσσωρευτικού Harvest Rate του DetFC

5.2.4.4 Focused Crawler με Keyword Representation

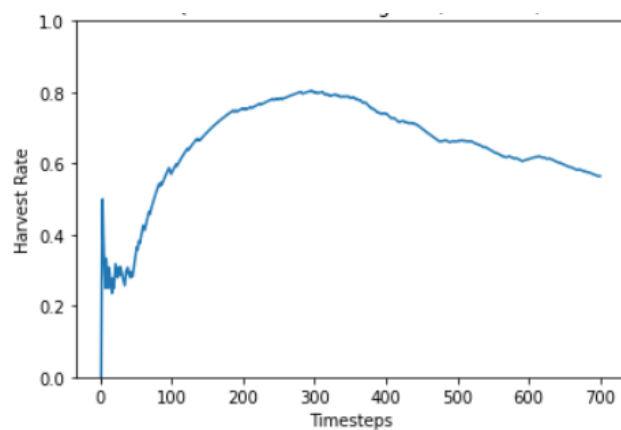
Στο σημείο, αυτό θα μελετήσουμε την περίπτωση του Focused Crawler με Keyword Representation. Χρησιμοποιούμε και πάλι τον ERC ως reward function. Παρ' όλα αυτά χρειαζόμαστε το πρώτο στάδιο του focused crawler, δηλαδή της Ενεργού Μάθησης, για το χαρακτηριστικό της Πιθανότητας Συνάφειας Τίτλου. Δεδομένης της μη ενθαρρυντικής επίδοσης της Ενεργού Μάθησης στο ειδικό πρόβλημα, χρησιμοποιούμε Ενεργό Μάθηση πάνω στο γενικό θέμα αναζήτησης, δηλαδή, εδώ, της Εθνοφαρμακολογίας. Ωστόσο, λαμβάνοντας υπόψιν ότι αυτό το πρόβλημα το έχουμε ήδη λύσει στο γενικό θέμα αναζήτησης της προηγούμενης ενότητας, θα μεταφέρουμε την μάθηση (*transfer learning*) του καλύτερου μας μοντέλο AverageLSTM.

Και σε αυτήν την περίπτωση, θέτουμε το μέγιστο όριο χρονικών βημάτων ίσο με 700. Πειραματιζόμαστε με δύο πράκτορες - DQN και DDQN - όπου και οι δύο έχουν ίδια αρχικοποίηση στα νευρωνικά τους δίκτυα. Όπως αναφέραμε και στην αρχή τρέχουμε για είσοδο: 25, 10 και 1 seed(s). Βασικός μας σκόπος είναι να δείξουμε πως οι πράκτορες μας με αυτήν την μεθοδολογία ξεπερνάνε τον DetFC ακόμα και με 1 seed για είσοδο. Σημειώνουμε πως και για τα δύο μοντέλα χρησιμοποιούμε τις ίδιες υπερπαραμέτρους, ώστε να μελετησουμε πραγματικά αν κάποιο είναι συστηματικά πιο αποδοτικό από το άλλο.

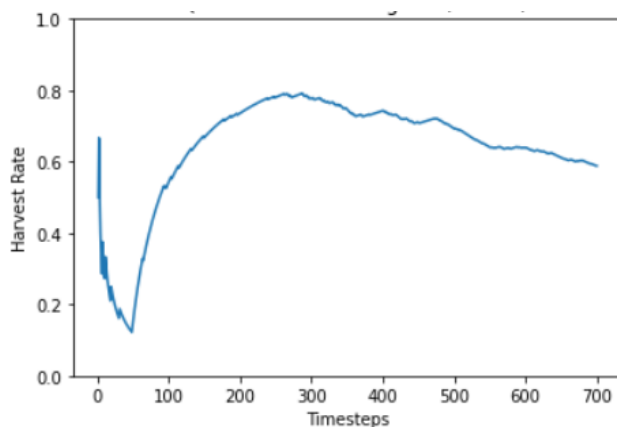
Τα αποτελέσματα του harvest rate για το DQN όλων των περιπτώσεων seeds τα αναπαριστούμε στα Σχήματα 5.13, 5.14 και 5.15.



Σχήμα 5.13: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DQN και 25 seeds

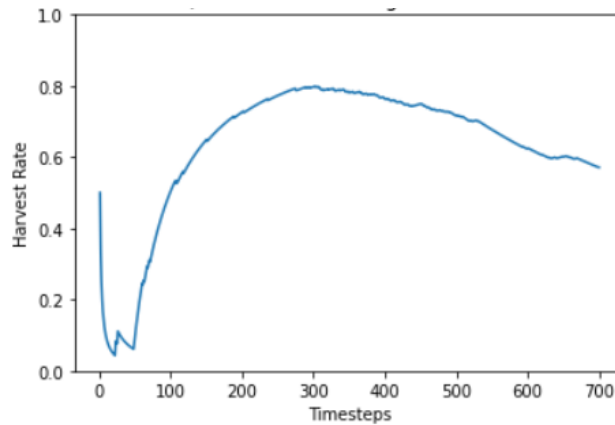


Σχήμα 5.14: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DQN και 10 seeds

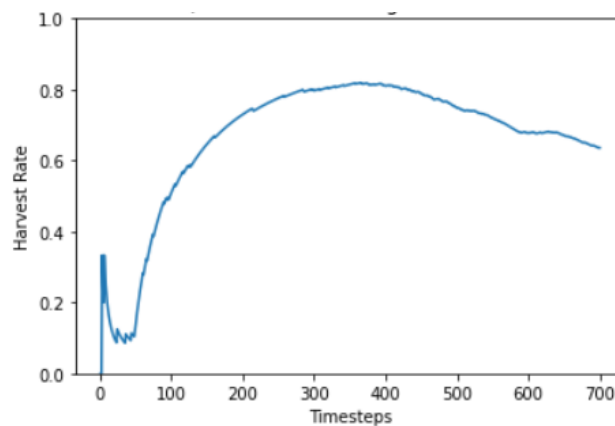


Σχήμα 5.15: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DQN και 1 seed

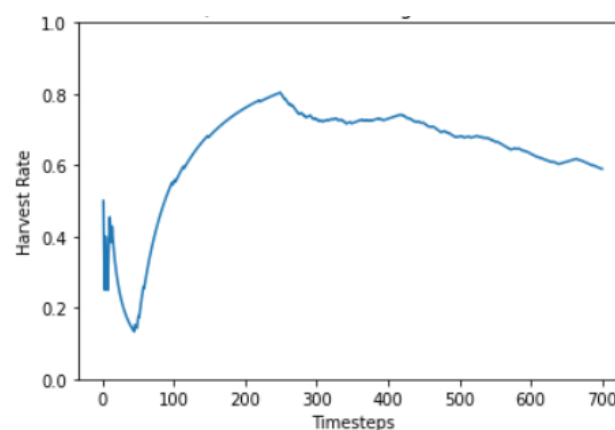
Τα αντίστοιχα γραφήματα για το DDQN απεικονίζονται στα Σχήματα 5.16, 5.17 και 5.18.



Σχήμα 5.16: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DDQN και 25 seeds



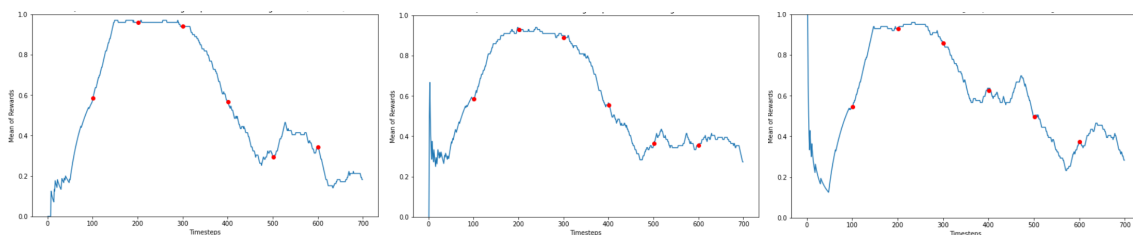
Σχήμα 5.17: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DDQN και 10 seeds



Σχήμα 5.18: Το διάγραμμα του συσσωρευτικού Harvest Rate του Focused Crawler με Keyword Representation με DDQN και 1 seed

Σημειώνουμε ότι το τελικό harvest rate του DDQN είναι μονίμως καλύτερο από το DQN και για τις τρεις περιπτώσεις, έχοντας καλύτερη επίδοση στα 700 κείμενα περίπου από 1% έως και 7%. Το καλύτερο harvest rate - **63.57%** - σημειώθηκε από το DDQN με 10 seeds.

Επίσης, παρατηρούμε πως και οι δύο πράκτορες έχουν πολύ καλύτερη επίδοση από τον DetFC ακόμα και με τη χρήση μόνο 1 seed. Πρακτικά, και οι δύο πράκτορες είναι τουλάχιστον **1,38 φορές** καλύτεροι από τον DetFC ανεξάρτητα από τον αριθμό των seeds. Στο 5.19, αναπαριστούμε, και εδώ, τις μεταβολές του μέσου reward για το DQN, σε κυλιόμενο παράθυρο ίσο με 100 χρονικά βήματα. Στο 5.20, αναπαριστούμε τα αντίστοιχα γραφήματα για το DDQN. Στο τέλος του κεφαλαίου, στα Σχήματα 5.21 και 5.22, αναπαριστούμε συγκεντρωτικά τα αποτελέσματα για τα DQN και DDQN.



Σχήμα 5.19: Διαγράμματα του μέσου reward για το DQN με κυλιόμενο παράθυρο ίσο με 100 χρονικά βήματα συναρτήσει των χρονικών βημάτων.

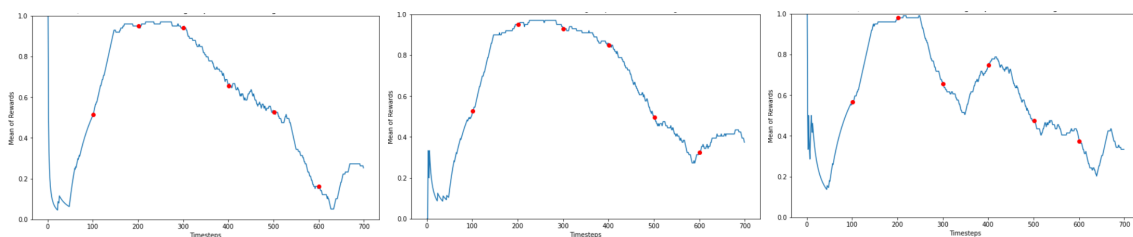
Αριστερά: Focused Crawler με Keyword Representation με DQN και **25 seeds**

Μέση: Focused Crawler με Keyword Representation με DQN και **10 seeds**

Δεξιά: Focused Crawler με Keyword Representation με DQN και **1 seed**

Οι κόκκινες βούλες αναπαριστούν τα χρονικά βήματα στα οποία ενημερώνεται το Target Q-Network

Παρατηρούμε πως και στα δύο μοντέλα, και στις τρεις περιπτώσεις των seeds, ο πράκτορας μαθαίνει να επιλέγει σχετικές ιστοσελίδες μετά τα 50 χρονικά βήματα. Σημειώνουμε πως έχουμε ορίσει την αφετηρία της εκπαίδευσης στο 50ο χρονικό βήμα. Αυτό σημαίνει πως το μοντέλο μετά την πρώτη του εκπαίδευση, κατάφερε να μάθει το σχετικό θέμα αναζήτησης. Εδώ, εκτιμούμε πως καθοριστική σημασία έπαιξε το γεγονός ότι το Experience Replay το είχαμε αρχικοποιήσει με τις αναπαραστάσεις των seeds.



Σχήμα 5.20: Διαγράμματα του μέσου reward για το DDQN με κυλιόμενο παράθυρο ίσο με 100 χρονικά βήματα συναρτήσει των χρονικών βημάτων.

Αριστερά: Focused Crawler με Keyword Representation με DDQN και **25 seeds**

Μέση: Focused Crawler με Keyword Representation με DDQN και **10 seeds**

Δεξιά: Focused Crawler με Keyword Representation με DDQN και **1 seed**

Οι κόκκινες βούλες αναπαριστούν τα χρονικά βήματα στα οποία ενημερώνεται το Target Q-Network

Στον Πίνακα 5.11 εμφανίζονται οι υπερπαραμέτροι που χρησιμοποιήσαμε για τον Focused Crawler με Keyword Representation. Ως αρχικό batch size χρησιμοποιούμε το 25. Σημειώνουμε πως μεταβάλλουμε, και εδώ, την τιμή του batch size με το πέρασμα των χρο-

νικών βημάτων ως εξής:

$$\text{batch_size} = \max(\text{batch_size}, \text{batch_size} \cdot \log_{10} \text{timestep}) \quad (5.2)$$

Με αυτόν τον τρόπο, όσο αυξάνονται οι εμπειρίες στο Experience Replay, θέλουμε, και πάλι, να αυξάνεται και το μέγεθος του batch. Δεδομένου πως οι εμπειρίες μετά από έναν καθορισμένο αριθμό χρονικών βημάτων διαφέρουν αρκετά μεταξύ τους, κατά το χρονικό βήμα που αντιστοιχεί στο μισό των συνολικών χρονικών βημάτων διπλασιάζουμε αυθαίρετα το batch size.

seeds	25, 10, 1
initial batch size	25
total timesteps	700
γ	0.99
replay start size	50
target update period (TUP)	100
initial epsilon value	0.15
epsilon decay	0.99
min epsilon value	0.05
clipvalue	1
optimizer	RMSProp
learning rate	0.001
number of hidden layers	3
input dimension	6
hidden layer dimensions	[25, 5]
hidden layer activation function	ReLU
output layer activation function	Identity function

Πίνακας 5.11: Υπερπαραμέτροι της *Keyword Representation* για το Δεύτερο Στάδιο του Ειδικού Προβλήματος

Αναφερόμαστε στο χρονικό σημείο κατά το οποίο, σε γενικές γραμμές, αρχίζει και πέφτει το μέσο reward του πράκτορα. Θα συζητήσουμε σχετικά με αυτό στην Συμπεράσματα (5.2.4.5). Με αυτόν τον τρόπο, θέλουμε το μοντέλο να εκπαιδεύεται πιο συχνά σε εμπειρίες που αντιστοιχούν σε πιο πρόσφατα μηδενικά rewards.

Ακόμα, για τα Q-Networks χρησιμοποιούμε MLP με 2 hidden layers. Ως optimizer χρησιμοποιούμε τον RMSProp, με σταθερό learning rate ίσο με 0.001. Επίσης, χρησιμοποιούμε και πάλι clipvalue ίσο με 1. Όσον αφορά τις πιο τυπικές υπερπαραμέτρους της Ενισχυτικής Μάθησης, χρησιμοποιούμε αρχικό ϵ ίσο με 0.15 το οποίο φθίνει εως την τιμή 0.05 με ρυθμό πώσης 0.99. Ορίζουμε το γ ίσο με 0.99, δεδομένου πως έχουμε να αντιμετωπίσουμε και πάλι ένα πρόβλημα χωρίς επεισόδια. Τέλος, η αφετηρία της εκπαίδευσης του πράκτορα συμβαίνει στο 50στο χρονικό βήμα, ενώ η ανανέωση του Target Q-Network συμβαίνει κάθε 100 χρονικά βήματα.

5.2.4.5 Συμπεράσματα

Βλέπουμε πως και στις τρεις περιπτώσεις των seeds, και τα δύο μοντέλα απολαμβάνουν μέγιστο μέσο reward για μεγάλο χρονικό διάστημα, της τάξης των 150 χρονικών βημάτων.

Αυτό ωστόσο αποκλίνει στην περίπτωση του DDQN με 10 seeds, στην οποία κρατάει για περίπου 250 χρονικά βήματα.

Θεωρούμε πως στην περίπτωση αυτή επιλέχθηκε κατά τη διάρκεια του τρεξίματος κάποιο-/α URL(s) τυχαία, που αν και δεν φαινόταν πολύ υποσχόμενο/α, τελικά ήταν πολύ σημαντικό/ά hub(s). Εδώ φαίνεται και η αξία της αρχικής εξερεύνησης στην αρχή του τρεξίματος. Προφανώς εάν αφήναμε και άλλο την εξερεύνηση, χωρίς τη μάθηση του μοντέλου, είναι δυνατόν ο πράκτορας να ανακάλυπτε και άλλα μονοπάτια που θα οδηγούσαν σε σχετικές ιστοσελίδες. Ωστόσο, μια τέτοια ενέργεια είναι χρήσιμη για αρκετά μακροπρόθεσμα. Παράλληλα, κάτι τέτοιο, εδώ, δεν το προτιμούμε, διότι μπορεί να λειτουργήσει εις βάρος του τελικού harvest rate, φέρνοντας με τυχαίο τρόπο κυρίως μη σχετικά κείμενα.

Στην συνέχεια, βλέπουμε το μέσο reward να πέφτει. Θεωρούμε πως στο σημείο αυτό ο focused crawler με Keyword Representation έχει μάθει το θέμα αναζήτησης κατ'ελάχιστο καλά όσο το γνωρίζει ο DetFC. Με άλλα λόγια, μόλις δει κάποιον τίτλο με τις επιθυμητές λέξεις-κλειδιά, έχει μάθει να επιλέγει το αντίστοιχο URL του. Από εκείνο το σημείο και ύστερα κρίνουμε την ικανότητα του μοντέλου να αναγνωρίζει hubs, που θα το οδηγούν σε σχετικά κείμενα. Σε όλες τις περιπτώσεις βλέπουμε να συμβαίνει αυτό, παρατηρώντας απότομα "όρη" στα γραφήματα του μέσου reward. Αυτή είναι και η διαφορά με τον DetFC. Ο νιτερμινιστικός focused crawler δεν μπορεί να αναγνωρίσει μη σχετικούς τίτλους, που σχετίζονται ωστόσο με το ευρύτερο θέμα της εθνοφαρμακολογίας. Επίσης, ο DetFC δεν μπορεί προφανώς να αναγνωρίζει μοτίβα, ώστε να διαλέγει ιστοσελίδες που πιθανόν είναι hubs.

Στον Πίνακα 5.12, αναπαριστούμε συγκεντρωτικά τις επιδόσεις όλων των μοντέλων μας πάνω στο Ειδικό Πρόβλημα. Πιο αναλυτικά, εξετάζουμε αρχικά τις περιπτώσεις με 400 και 700 χρονικά βήματα. Στα 400 χρονικά βήματα, η επίδοση για το Keyword Representation αφορά τη μέση τιμή των harvest rates όλων των αντίστοιχων μοντέλων. Από την άλλη, στην περίπτωση των 700 χρονικών βημάτων δείχνουμε αναλυτικά όλες τις επιδόσεις των μοντέλων.

Focused Crawler	Seeds	Harvest Rate	Timesteps
Random Crawling	25	7.00%	400
Only NLP Representation	25	30.00%	400
DetFC	25	64.83%	400
Keyword Representation	[25, 10, 1]	75.68%	400
DetFC	25	40.14%	700
Keyword Representation with DQN	25	55.14%	700
Keyword Representation with DDQN	25	57.14%	700
Keyword Representation with DQN	10	56.42%	700
Keyword Representation with DDQN	10	63.57%	700
Keyword Representation with DQN	1	58.14%	700
Keyword Representation with DDQN	1	59.00%	700

Πίνακας 5.12: Αποτελέσματα των focused crawlers για το Ειδικό Πρόβλημα αναζήτησης. Σημειώνουμε πως το harvest rate (75.68%) με Keyword Representation για τα 400 χρονικά βήματα αποτελεί την μέση τιμή των τιμών που προέκυψαν για 25, 10 και 1 seed(s).

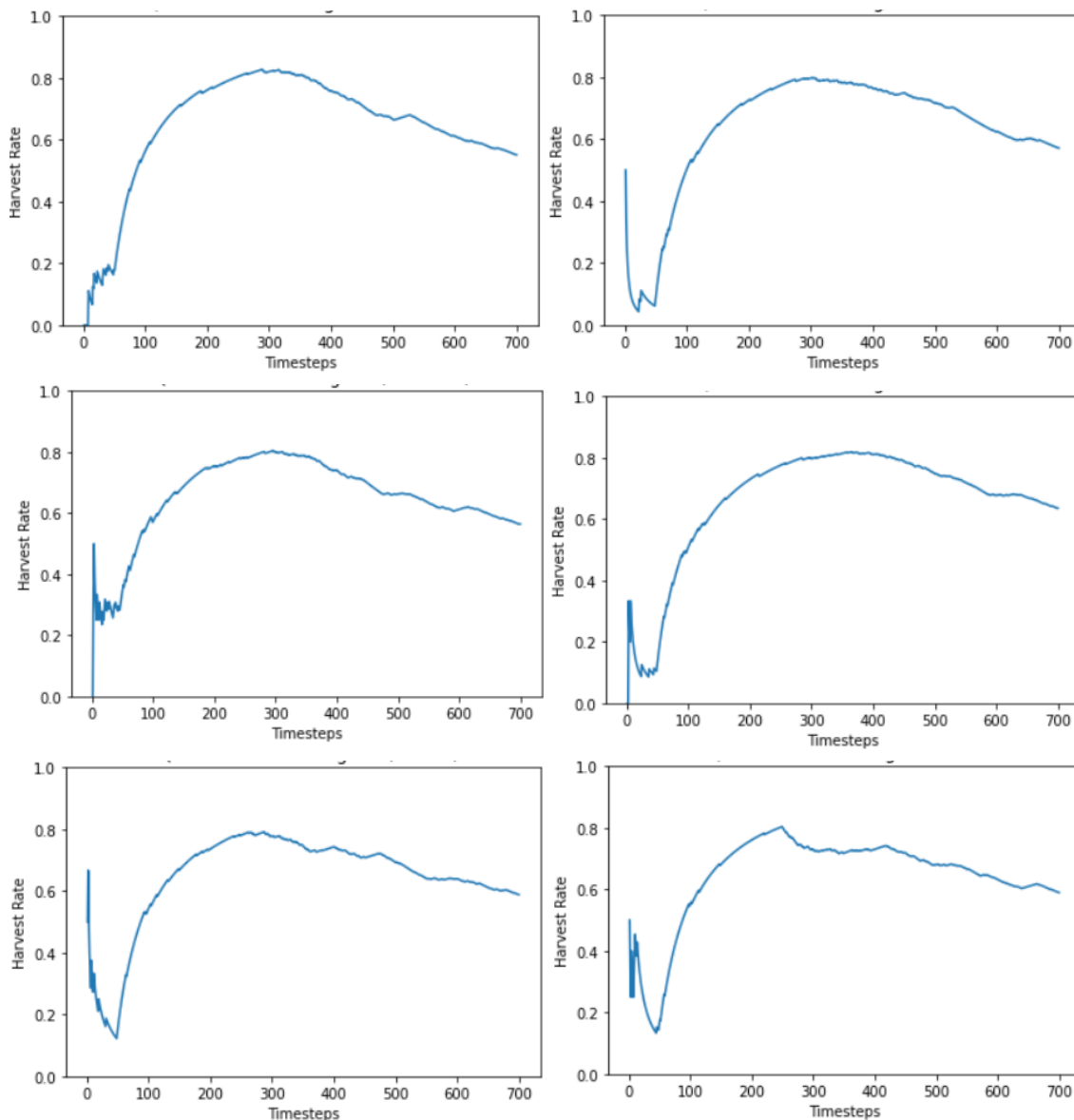
Παρατηρούμε πως και στα 400 και στα 700 χρονικά βήματα η Keyword Representation υπερτερεί σημαντικά έναντι όλων των άλλων αναπαραστάσεων. Στα 400, η μέση επίδοση

είναι καλύτερη 1,167 φορές από τον DetFC, 2,522 φορές από την Only NLP Representation και 10,811 φορές από το Τυχαίο Πιλοτικό Τρέξιμο (Random Crawling). Παράλληλα, στα 700 χρονικά βήματα ο καλύτερος DDQN - των 10 χρονικών βημάτων - είναι **1,583 φορές** καλύτερος από τον DetFC. Σημειώνουμε πως η τιμή αυτή, όσο αυξάνονται τα χρονικά βήματα, περιμένουμε πως θα μεγαλώνει, δεδομένου πως αυξάνεται συνεχώς το μέγεθος του μετώπου αναζήτησης.

5.2.5 Σύγκριση με ανθρώπινη προσωμείωση

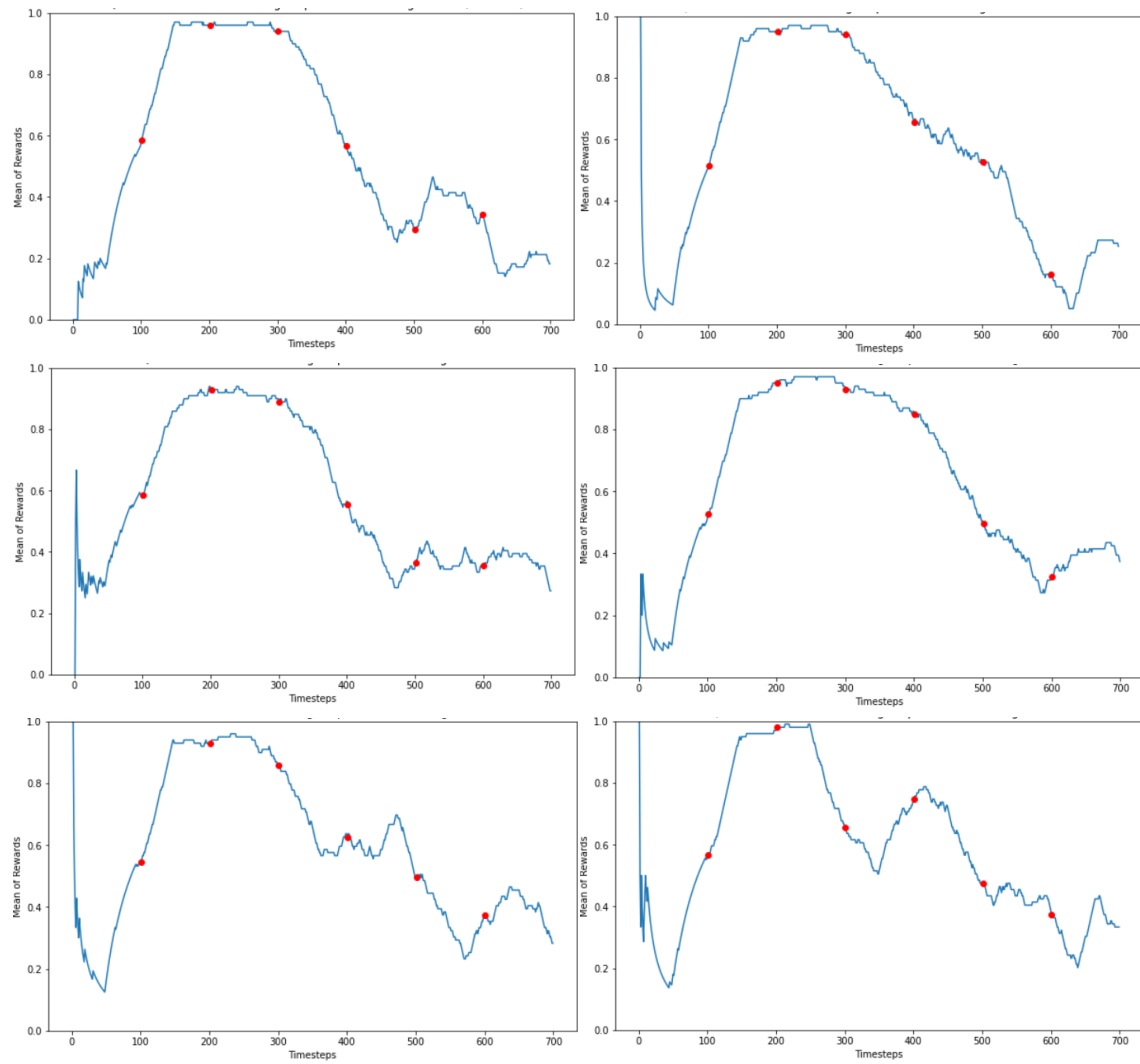
Το συνολικό τρέξιμο του δεύτερου σταδίου διήρκησε περίπου για 7 ώρες. Μέσα σε αυτές τις ώρες, το σύστημα έφερε 700 κείμενα, από τα οποία, στην καλύτερη περίπτωση, τα 447 ήταν σχετικά με το θέμα αναζήτησης. Από την άλλη, εκτιμήσαμε πως ένας ειδικός, που γνωρίζει καλά το θέμα αναζήτησης, χρειάζεται περίπου 36 ώρες για να φέρει τον ίδιο αριθμό (427) σχετικά κείμενα, έχοντας την ίδια αφειτηρία (seeds) με τον focused crawler μας. Έτσι, βελτιώσαμε περίπου κατά **5,14 φορές** την αποδοτικότητα (**efficiency**) σε σχέση με τον ειδικό.

Όσον αφορά την αποτελεσματικότητα, εκτιμήσαμε πως για 50 αντιπροσωπευτικά κείμενα το ποσοστό των σχετικών κειμένων, που εξάγονται από αυτά, είναι οπτιμιστικά περίπου ίσο με 19%. Εμείς στην καλύτερη μας προσπάθεια φέραμε σχετικά κείμενα σε ποσοστό περίπου 63%. Άρα, βελτιώσαμε την αποτελεσματικότητα (**effectiveness**) τουλάχιστον **3,31 φορές** σε σχέση με τον ειδικό.



Σχήμα 5.21: Συγκεντρωτικά τα διαγράμματα των συσσωρευτικών Harvest Rates με Keyword Representation (DQN-DDQN).

Αριστερά απεικονίζουμε για το DQN και δεξιά για το DDQN. Τα δύο πρώτα διαγράμματα (της πρώτης γραμμής) αφορούν για 25 seeds, τα επόμενα δύο για 10 seeds και τα τελευταία δύο για 1 seed



Σχήμα 5.22: Συγκριτικά τα διαγράμματα του μέσου reward με κυλιόμενο παραθύρο ίσο με 100 χρονικά βήματα, με χρήση Keyword Representation (DQN-DDQN). Αριστερά απεικονίζουμε για το DQN και δεξιά για το DDQN. Τα δύο πρώτα διαγράμματα (της πρώτης γραμμής) αφορούν για 25 seeds, τα επόμενα δύο για 10 seeds και τα τελευταία δύο για 1 seed

Κεφάλαιο **6**

Επίλογος

6.1 Συμπεράσματα και Συνεισφορές

Εν κατακλείδι, στην διπλωματική αυτή μελετήσαμε μεθόδους *focused crawling*, με στόχο να στηρίζουμε την έρευνα των ειδικών εθνοφαρμακολογίας στο πρόβλημα αναζήτησης σχετικών κειμένων. Δημιουργήσαμε έναν *focused crawler* δύο σταδίων ακολουθώντας το Πρότυπο Μαθητή-Ερευνητή που ορίσαμε. Στο πρώτο στάδιο, χρησιμοποιήσαμε μεθόδους Ενεργού Μάθησης (Active Learning) και στο δεύτερο αλγορίθμους Βαθιάς Ενισχυτικής Μάθησης (Deep Reinforcement Learning). Σημειώνουμε πως δεν έχει ξαναχρησιμοποιηθεί ο συνδυασμός αυτών στο *focused crawling*.

Ορίσαμε το περιβάλλον του *focused crawler* σαν ένα MDP, το οποίο διαφοροποιείται από αντίστοιχες απόπειρες στη βιβλιογραφία. Το δικό μας MDP είναι πολύ μεγάλο - με συνεχείς χώρους των *states* και των *actions* - και εδραιώνει τη χρήση της Βαθιάς Ενισχυτικής Μάθησης στο *focused crawling*. Επίσης, φαίνεται ότι η κοινή αναπαράσταση ανάμεσα στο *state* και το *action* είναι αρκετά αποδοτική, όταν το *focused crawling* συνδυάζεται με Ενισχυτική Μάθηση.

Ακόμα, λάβαμε υπόψιν διαφορετικές αναπαραστάσεις - τις *General Representation*, *Only NLP Representation* και *Keyword Representation*. Αυτές τις μελετήσαμε σε δύο διαφορετικά προβλήματα αναζήτησης πάνω στο PubMed - ένα γενικό πρόβλημα και ένα πιο ειδικό με λέξεις-κλειδιά. Συμπεραίνουμε πως η Ενεργός Μάθηση μπορεί να κατανοήσει αρκετά αποτελεσματικά το γενικό πρόβλημα αναζήτησης, σημειώνοντας μέσο *accuracy* ίσο με 88.0% με χρήση 130 μόνο επιπλέον επισημειωμένων δεδομένων. Από την άλλη, παρατηρούμε πως η Ενεργός Μάθηση δεν μπορεί να λύσει αποτελεσματικά, με αποκλειστική χρήση *word2vec embeddings*, ένα πρόβλημα ταξινόμησης, στο οποίο παίζουν πολύ σημαντικό ρόλο συγκεκριμένοι συνδυασμοί λέξεων-κλειδιών. Αντίθετα, αυτό το πρόβλημα λύνεται ικανοποιητικά με χρήση ενός ντετερμινιστικού ταξινομητή, σημειώνοντας 95% *precision* και 61% *recall*.

Για το γενικό πρόβλημα βλέπουμε ότι ο *focused crawler* με *General Representation* βελτιώνει τη *baseline* μέθοδο κατά 1.8 φορές. Από την άλλη, στο ειδικό πρόβλημα ο *focused crawler* με *Keyword Representation* υπερνικά τον άπληστο ντετερμινιστικό *focused crawler* (DetFC) κατά τουλάχιστον 1.583 φορές. Η βελτίωση αυτή ρεαλιστικά μπορεί να είναι αρκετά μεγαλύτερη, θεωρώντας πως το σύστημά μας καταλαβαίνει ως σχετικά περίπου το 61% όλων των σχετικών κειμένων.

Ακόμα, συγκρίναμε την μέθοδό μας με μία ρεαλιστική ατομική απόπειρα του ειδικού

για εύρεση των σχετικών κειμένων. Βλέπουμε πως η μέθοδος μας βελτιώνει την ανθρώπινη προσπάθεια έχοντας 5.14 φορές καλύτερη αποδοτικότητα και 3.31 φορές καλύτερη αποτελεσματικότητα.

Στην περίπτωση της Keyword Representation, το πρώτο στάδιο του focused crawler, εκτός από την αναπαράσταση των χαρακτηριστικών του action, μπορούμε να το χρησιμοποιούμε και σαν reward function του δεύτερου σταδίου. Ωστόσο, εάν το πρόβλημα αναζήτησης είναι αρκετά δύσκολο για το πρόβλημα ταξινόμησης του πρώτου σταδίου, με αποτέλεσμα μια θορυβώδη reward function, προτείνουμε την αντικατάστασή της από κάποια εναλλακτική μέθοδο, όπως αυτή του ERC. Από την άλλη, προκειμένου να είναι αποτελεσματική η μεταφορά μάθησης στην αναπαράσταση του action, προτείνουμε η Ενεργός Μάθηση του πρώτου σταδίου να πραγματοποιηθεί πάνω σε ένα υπερσύνολο πιο γενικό θέμα αναζήτησης - στην περίπτωσή μας ήταν η εθνοφαρμακολογία - ώστε να δώσει την πληροφορία της συνάφειας του action σχετικά με το ευρύτερο θέμα.

Τέλος, σημειώνουμε πως η μελέτη μας συνέβαλε στη δημιουργία δύο συνόλων δεδομένων με 813 και 1012 URLs αναφορών αντίστοιχα. Αυτά είναι επισημειώμενα με τις κλάσεις συνάφειας, που ορίσαμε στο γενικό και στο ειδικό πρόβλημα αναζήτησης κατά αντιστοιχία.

6.2 Μελλοντικές Επεκτάσεις

Στο σημείο αυτό, θα αναφερθούμε σε πιθανές μελλοντικές επεκτάσεις των ιδεών και μεθοδολογιών που μελετήσαμε στην διπλωματική εργασία αυτή. Ενδεικτικά αναφέρουμε τις παρακάτω:

- Δημιουργία ενός ολοκληρωμένου ανοικτού κώδικα (open source) λογισμικού. Στόχος μας θα είναι η παροχή, στους επιστήμονες εθνοφαρμακολογίας, ενός υποστηρικτικού εργαλείου για τη μελέτη τους.
- Επιπλέον ταξινόμηση των αποτελεσμάτων από τη μεριά της Εθνοφαρμακολογίας/Φαρμακογνωσίας μέσω Μηχανικής Μάθησης. Είναι ιδιαίτερα σημαντικό τα αποτελέσματα του τρεξίματος του focused crawler να κατηγοριοποιηθούν ως προς το είδος των ασθενειών ή των οργάνων του ανθρωπίνου σώματος στα οποία αναφέρονται.
- Αξιολόγηση των συστημάτων focused crawler, που μοντελοποιήσαμε, πάνω σε άλλα θέματα μελέτης πχ. αθλητικά νέα, ιατρικά κείμενα, χόμπυ κá. Επίσης, θα μπορούσαμε να δοκιμάσουμε και πάνω σε άλλες γνωστές βάσεις δεδομένων, πχ. το Google Scholar, η Wikipedia κá.
- Αυτόματος συμπερασμός των σημαντικών λέξεων-κλειδιών με τη βοήθεια της ανατροφοδότησης (feedback) από τον ερευνητή.
- Μελέτη διαφορετικών τεχνικών δειγματοληψίας στο κομμάτι της Ενεργού Μάθησης για σύνολα δεδομένων με μεγάλη ανισορροπία
- Απόπειρες για μάθηση του πρώτου σταδίου του focused crawler με One-Class Classification αντί για Ενεργό Μάθηση. Αυτό συνεπάγεται εντελώς διαφορετική αντιμετώπιση του προβλήματος, αφού δεν θα υπάρχει ανατροφοδότηση. Ωστόσο, σημαντικό θα είναι,

επίσης, εάν καταφέρουμε να συνδυάσουμε αυτές τις δύο μεθόδους και να συγκρίνουμε τα αποτελέσματα με τις υπάρχουσες μεθοδολογίες.

Βιβλιογραφία

- [1] A. Kontogiannis and E. Axiotis. *A Personalized Artificial-Intelligence-enabled Method for Efficient Research in Ethnopharmacology*. FOSSCOMM 2020, Thessaloniki, 2020.
- [2] *Reinforcement Learning Applications: A Brief Guide on How to Get Business Value from RL*. <https://perfectial.com/blog/reinforcement-learning-applications/>.
- [3] Burr Settles. *Active learning literature survey*. Τεχνική Αναφορά με αριθμό, 2010.
- [4] Xin Rong. *word2vec Parameter Learning Explained*, 2016.
- [5] Ron Artstein. *Inter-annotator Agreement*, σελίδες 297–313. Springer Netherlands, Dordrecht, 2017.
- [6] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [7] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.
- [8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg and Demis Hassabis. *Human-level control through deep reinforcement learning*. *Nature*, 518(7540):529–533, 2015.
- [9] Chattopadhyay Debprasad. *Ethnomedicine: A Source of Complementary Therapeutics*. 2019.
- [10] Evangelos Axiotis, Maria Halabalaki and Alexios Leandros Skaltsounis. *An Ethnobotanical Study of Medicinal Plants in the Greek Islands of North Aegean Region*. *Frontiers in Pharmacology*, 9, 2018.
- [11] Mohammad Uddin and Md. Hassan. *Determination of informant consensus factor of ethnomedicinal plants used in Kalenga forest, Bangladesh*. *Bangladesh Journal of Plant Taxonomy*, 21(1):83–91, 2014.
- [12] Jasmine Zenderland, R. Hart, R. Bussmann, Narel Y. Paniagua Zambrana, Shalva Sikharulidze, Zaal Kikvidze, D. Kikodze, David Tchelidze, M. Khutsishvili and K. Batsatsashvili. *The Use of “Use Value”: Quantifying Importance in Ethnobotany*. *Economic Botany*, 73:293 – 303, 2019.

- [13] *PubMed*®. <https://pubmed.ncbi.nlm.nih.gov/>.
- [14] *Google Scholar*. <https://scholar.google.com/>.
- [15] Charu C. Aggarwal, Fatima Al-Garawi and Philip S. Yu. *Intelligent Crawling on the World Wide Web with Arbitrary Predicates*. *Proceedings of the 10th International Conference on World Wide Web, WWW '01*, σελίδα 96–105, New York, NY, USA, 2001. Association for Computing Machinery.
- [16] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1η έκδοση, 1997.
- [17] Nello Cristianini and Elisa Ricci. *Support Vector Machines*, σελίδες 928–932. Springer US, Boston, MA, 2008.
- [18] F. Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957.
- [19] C. Van Der Malsburg. *Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. *Brain Theory* Günther Palm and Ad Aertsen, επιμελητές, σελίδες 245–248, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.
- [20] Yann LeCun, Léon Bottou, Genevieve B. Orr and Klaus Robert Müller. *Efficient BackProp*. *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, σελίδα 9–50, Berlin, Heidelberg, 1998. Springer-Verlag.
- [21] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*, 2019.
- [22] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, σελίδα 696–699. MIT Press, Cambridge, MA, USA, 1988.
- [23] Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term Memory*. *Neural Comput.*, 9(8):1735–1780, 1997.
- [24] Sebastian Ruder. *An overview of gradient descent optimization algorithms*, 2017.
- [25] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*, 2017.
- [26] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*. The MIT Press, 2016.
- [27] Lutz Prechelt. *Early Stopping – But When?*, σελίδες 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

- [29] Hwanjo Yu. *SVM Selective Sampling for Ranking with Application to Data Retrieval*. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, σελίδα 354–363, New York, NY, USA, 2005. Association for Computing Machinery.
- [30] Steven C. H. Hoi, Rong Jin and Michael R. Lyu. *Large-Scale Text Categorization by Batch Mode Active Learning*. *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, σελίδα 633–642, New York, NY, USA, 2006. Association for Computing Machinery.
- [31] Burr Settles and Mark Craven. *An Analysis of Active Learning Strategies for Sequence Labeling Tasks*. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, σελίδα 1070–1079, USA, 2008. Association for Computational Linguistics.
- [32] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han and Philip S. Yu. *Active learning: A survey*, σελίδες 571–605. CRC Press, 2014.
- [33] Hado van Hasselt, Arthur Guez and David Silver. *Deep Reinforcement Learning with Double Q-Learning*. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, σελίδα 2094–2100. AAAI Press, 2016.
- [34] F. Wilfrid Lancaster and Emily Gallup Fayen. *Information retrieval: on-line [by] F. W. Lancaster and E. G. Fayen*. Melville Pub. Co Los Angeles, 1973.
- [35] G. Salton, A. Wong and C. S. Yang. *A vector space model for automatic indexing*. *Communications of the ACM*, 1975.
- [36] Yoshua Bengio, Réjean Ducharme, Pascal Vincent and Christian Janvin. *A Neural Probabilistic Language Model*. *J. Mach. Learn. Res.*, 3(νύλλ):1137–1155, 2003.
- [37] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu and Pavel Kuksa. *Natural Language Processing (Almost) from Scratch*. *J. Mach. Learn. Res.*, 12(νύλλ):2493–2537, 2011.
- [38] Tomas Mikolov, G.s Corrado, Kai Chen and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. σελίδες 1–12, 2013.
- [39] Jeffrey Pennington, Richard Socher and Christopher Manning. *GloVe: Global Vectors for Word Representation*. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, σελίδες 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.
- [40] Petra Saskia Bayerl and Karsten Ingmar Paul. *What Determines Inter-Coder Agreement in Manual Annotations? A Meta-Analytic Investigation*. *Computational Linguistics*, 37(4):699–725, 2011.
- [41] Jacob Cohen. *A Coefficient of Agreement for Nominal Scales*. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

- [42] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications, 2004.
- [43] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye. *Probability & statistics for engineers and scientists*. Pearson Education, Upper Saddle River, 8την έκδοση, 2007.
- [44] S. S. Shapiro and M. B. Wilk. *An Analysis of Variance Test for Normality (Complete Samples)*. *Biometrika*, 52(3/4):591–611, 1965.
- [45] *Real Statistics Using Excel*. <https://www.real-statistics.com/statistics-tables/shapiro-wilk-table/>.
- [46] Janez Demšar. *Statistical Comparisons of Classifiers over Multiple Data Sets*. *J. Mach. Learn. Res.*, 7:1–30, 2006.
- [47] Sotiris Batsakis, Euripides G. M. Petrakis and Evangelos Milios. *Improving the Performance of Focused Web Crawlers*. *Data Knowl. Eng.*, 68(10):1001–1013, 2009.
- [48] Filippo Menczer, Gautam Pant and Padmini Srinivasan. *Topical Web Crawlers: Evaluating Adaptive Algorithms*. *ACM Trans. Internet Technol.*, 4(4):378–419, 2004.
- [49] Soumen Chakrabarti, Martin van den Berg and Byron Dom. *Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery*. *Comput. Netw.*, 31(11–16):1623–1640, 1999.
- [50] Soumen Chakrabarti, Kunal Punera and Mallela Subramanyam. *Accelerated Focused Crawling through Online Relevance Feedback*. *Proceedings of the 11th International Conference on World Wide Web, WWW '02*, σελίδα 148–159, New York, NY, USA, 2002. Association for Computing Machinery.
- [51] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles and Marco Gori. *Focused Crawling Using Context Graphs*. *Proceedings of the 26th International Conference on Very Large Data Bases, VLDB '00*, σελίδα 527–534, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [52] Andrew K. McCallum. *BOW: A toolkit for statistical language modeling, text retrieval, classification and clustering*. <http://www.cs.cmu.edu/~mccallum/bow/>, 1996.
- [53] *The HTML DOM (Document Object Model)*. https://www.w3schools.com/js/js_htmlDOM.asp.
- [54] Hai Dong, Farookh Khadeer Hussain and Elizabeth Chang. *State of the Art in Semantic Focused Crawlers*. *Proceedings of the International Conference on Computational Science and Its Applications: Part II, ICCSA '09*, σελίδα 910–924, Berlin, Heidelberg, 2009. Springer-Verlag.

- [55] Julien Tane, Christoph Schmitz and Gerd Stumme. *Semantic Resource Management for the Web: An e-Learning Application. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers Posters*, WWW Alt. '04, σελίδα 1–10, New York, NY, USA, 2004. Association for Computing Machinery.
- [56] Aysu Betin Can and Nazife Baykal. *MedicoPort: A medical search engine for all. Computer Methods and Programs in Biomedicine*, 86(1):73 – 86, 2007.
- [57] *Unified medical language system. National Library of Medicine*, 2008.
- [58] Juan F. García and Miguel V. Carriegos. *Mirkwood: An Online Parallel Crawler. International Joint Conference: 12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on European Transnational Education (ICEUTE 2019)* Francisco Martínez Álvarez, Alicia Troncoso Lora, José Antonio Sáez Muñoz, Héctor Quintián and Emilio Corchado, επιμελητές, σελίδες 47–56, Cham, 2020. Springer International Publishing.
- [59] *Adaptive focused crawling using online learning: A study on content related to Islamic extremism. Zenodo*, 2018.
- [60] Jason Rennie and Andrew McCallum. *Using Reinforcement Learning to Spider the Web Efficiently. Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, σελίδα 335–343, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
- [61] Ioannis Partalas, Georgios Paliouras and I. Vlahavas. *Reinforcement Learning with Classifier Selection for Focused Crawling. σελίδες 759–760*, 2008.
- [62] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Elsevier, 2005.
- [63] Zeng Wei, Jun Xu, Yanyan Lan, Jiafeng Guo and Xueqi Cheng. *Reinforcement Learning to Rank with Markov Decision Process. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, σελίδα 945–948, New York, NY, USA, 2017. Association for Computing Machinery.
- [64] Y. Wang, L. Wang, Y. Li, D. He, W. Chen and T. Y. Liu. *A Theoretical Analysis of NDCG Ranking Measures. Journal of Machine Learning Research*, 2013.
- [65] Miyoung Han, Pierre Henri Wuillemin and Pierre Senellart. *Focused Crawling through Reinforcement Learning. 18th International Conference on Web Engineering (ICWE 2018)* Tommi Mikkonen, Ralf Klamka and Juan Hernández, επιμελητές, τόμος 10845 στο *Lecture Notes in Computer Science*, σελίδες 261–278, Cáceres, Spain, 2018. Springer.
- [66] M. Bloodgood. *Support Vector Machine Active Learning Algorithms with Query-by-Committee Versus Closest-to-Hyperplane Selection. 2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, σελίδες 148–155, 2018.

- [67] Bang An, Wenjun Wu and Huimin Han. *Deep Active Learning for Text Classification. Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, ICVISP 2018, New York, NY, USA, 2018. Association for Computing Machinery.
- [68] Brian D. Davison. *Topical Locality in the Web. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, σελίδα 272–279, New York, NY, USA, 2000. Association for Computing Machinery.
- [69] Karthik Narasimhan, Tejas Kulkarni and Regina Barzilay. *Language Understanding for Text-based Games using Deep Reinforcement Learning. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, σελίδες 1–11, Lisbon, Portugal, 2015. Association for Computational Linguistics.
- [70] *Biomedical natural language processing*. <https://bio.nlplab.org/>.
- [71] *Wikipedia, the free encyclopedia*. <https://www.wikipedia.org/>.
- [72] M. Stone. *Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.
- [73] T. G. Dietterich. *Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation*, 10(7):1895–1923, 1998.
- [74] C. Schröder and A. Niekler. *A Survey of Active Learning for Text Classification using Deep Neural Networks*, 2020.
- [75] *WordWeb Online*. <https://www.wordwebonline.com/>.
- [76] *Flora of Greece Web*. <http://portal.cybertaxonomy.org/flora-greece/content>.
- [77] *English Club*. <https://www.englishclub.com/vocabulary/world-countries-nationality.php>.
- [78] *Wikitravel - The Free Travel Guide*. https://wikitravel.org/en/Main_Page.

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

βλ.	βλέπε
κλπ	και λοιπά
κ.ο.κ	και ούτω καθεξής
κ.ά	και άλλοι/άλλες/άλλα
ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
RL	Reinforcement Learning
AL	Active Learning
SVM	Support Vector Machines
MDP	Markov Decision Process

Απόδοση ξενόγλωσσων όρων

Απόδοση

Ενεργός Μάθηση
συσσωρεύω
δυναμική ταξινόμηση
κάδος
ταξινόμηση
ταξινομητής
συσταδοποίηση
ερπετό
μπουσουλίμα
βάση δεδομένων
Βαθιά Μάθηση
Βαθιά Ενισχυτική Μάθηση
διακριτοποίηση
δυλιστής
εμπειρογνώμονας πεδίου
ανάδραση
πρόγραμμα εστιασμένης ανίχνευσης ιστού
μέτωπο αναζήτησης
Χρυσός Κανόνας Επισημειώσεων
υπερσύνδεσμοι
Ανάκτηση Πληροφορίας
σύνδεσμοι
Περιθώρια Δειγματοληψία
Μηχανική Μάθηση
Μαρκοβιανή Διαδικασία Λήψης Αποφάσεων
βασισμένο σε οντολογίες
Ενεργός Μάθηση βασισμένη σε συλλογή
ουρά προτεραιότητας
Τυχαία Δειγματοληψία
Ενισχυτική Μάθηση
συνάφεια
αρχικά κείμενα
σημασιολογικός
Επιβλεπόμενη Μάθηση

Ξενόγλωσσος όρος

Active Learning
aggregate
binary classification
bucket
classification
classifier
clustering
crawler
crawling
database
Deep Learning
Deep Reinforcement Learning
discretization
distiller
domain expert
feedback
focused crawler
frontier
Gold Standard
hyperlinks
Information Retrieval
links
Margin Sampling
Machine Learning
Markov Decision Process
ontology-based
pool-based Active Learning
priority queue
Random Sampling
Reinforcement Learning
relevance
seeds
semantic
Supervised Learning

Μηχανές Διανυσματικής Υποστήριξης
μεταφορά μάθησης

Δειγματοληψία με Αβεβαιότητα

Μη Επιβλεπόμενη Μάθηση

προς επίσκεψη στο μέλλον

προσπελασμένος

Παγκόσμιος Ιστός

πρόγραμμα ανίχνευσης ιστού

Support Vector Machines

transfer learning

Uncertainty Sampling

Unsupervised Learning

unvisited

visited

Web

web crawler