Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Παρακολούθηση Διαλογικής Κατάστασης με αρχιτεκτονική Encoder-Decoder και χρήση Pointer - Generator Δικτύου

Διπλωματικη Εργασια

του

ΧΡΗΣΤΟΥ ΜΙΑΜΗ

Επιβλέπων: Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εργαστηριο Ετφτων Συστηματων
Αθήνα, Νοέμβριος 2019

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων

Παρακολούθηση Διαλογικής Κατάστασης με αρχιτεκτονική Encoder-Decoder και χρήση Pointer - Generator Δικτύου

Διπλωματικη Εργασια

του

ΧΡΗΣΤΟΥ ΜΙΑΜΗ

Επιβλέπων: Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Νοεμβρίου 2019.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ανδρέας Σταφυλοπάτης	Παναγιώτης Τσανάκας	Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2019

(Υπογραφή)

.....

ΜΙΑΜΗΣ ΧΡΗΣΤΟΣ

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

Περίληψη

Το αντικείμενο της παρούσας διπλωματικής είναι η ανάπτυξη ενός μοντέλου παρακολούθησης της διαλογικής κατάστασης (Dialog State Tracking) εφεξής DST . Το DST είναι ένα ανοιχτό πρόβλημα στην ερευνητική κοινότητα με συνεχώς αναπτυσσόμενες μεθόδους και αποτελεί μέρος μιας ευρύτερης ροής εργασίας στον τομέα της *Επεξεργασίας Φυσικής Γλώσσας*. Πιο αναλυτικά, η παρακολούθηση της κατάστασης (state tracking ή αλλιώς belief tracking) εντοπίζεται στην εκτίμηση του στόχου του χρήστη κατά την (χρονική) εξέλιξη του διαλόγου. Η ακριβής εκτίμηση της κατάστασης κατά την εξέλιξη του διαλόγου είναι επιθυμητή καθώς παρέχει ευρωστία σε σφάλματα που εμφανίζονται σε προστάδια της επεξεργασίας γλώσσας όπως σφάλματα στην αναγνώριση ομιλίας ή σφάλματα εξαιτίας της ασάφειας [9] που είναι ένα εγγενές χαρακτηριστικό της γλώσσας. Πρόσφατα, έχουν προταθεί πολλά μοντέλα για την παρακολούθηση της κατάστασης διαλόγου. Ωστόσο, οι συγκρίσεις μεταξύ μοντέλων είναι σπάνιες και διαφορετικές ερευνητικές ομάδες χρησιμοποιούν διαφορετικά δεδομένα από διαφορετικούς τομείς. Επιπλέον, δεν υπάρχει επί του παρόντος κοινό σύνολο δεδομένων που να επιτρέπει πειράματα πάνω στην παρακολούθηση κατάστασης διαλόγου, με αποτέλεσμα να πρέπει να γίνει συλλογή και προεπεξεργασία τέτοιων δεδομένων από διαφορετικές πηγές, κάτι το οποίο είναι δαπανηρό και χρονοβόρο. Όλα αυτά τα ζητήματα εμποδίζουν την ανάπτυξη ενός state-of-the-art συστήματος. Η παρούσα διπλωματική αναπτύσσει ένα DST μοντέλο, η αρχιτεκτονική του οποίου επιλέχθηκε ύστερα από εκτενή βιβλιογραφική έρευνα και επιτυγχάνει επί του παρόντος state-of-the-art αποτελέσματα σε ένα ευρέως χρησιμοποιούμενο και γνωστό σύνολο δεδομένων. Αρχικά, θα γίνει μια εισαγωγή στις βασικές ενότητες που θα συναντηθούν στην παρούσα διπλωματική. Θα παρουσιαστεί εκτενώς το απαραίτητο θεωρητικό υπόβαθρο εντός των θεωρητικών αναγκών της διπλωματικής. Στη συνέχεια, θα αναφερθούν τεχνικές επεξεργασίας κειμένου και αρχιτεκτονικές που χρησιμοποιούνται στην ανάπτυξη του μοντέλου. Έπειτα θα ακολουθήσει εκτενής ανάλυση της βασικής αρχιτεκτονικής του μοντέλου και της πειραματικής διαδικασίας και συνέχεια θα γίνει εμβάθυνση στα εξαγόμενα αποτελέσματα. Τέλος, θα προταθούν επιπλέον τροποποιήσεις στο σχεδιασμό της αρχιτεκτονικής με σκοπό την περαιτέρω βελτίωση των τελικών αποτελεσμάτων.

Λέξεις Κλειδιά

Παρακολούθηση διαλογικής κατάστασης, παρακολούθηση πεποιθήσεων, επεξεργασία φυσικής γλώσσας, μηχανική μάθηση, νευρωνικά δίκτυα.

Abstract

The purpose of this thesis is the development of a Dialog State Tracking model (DST). DST is an open problem in the constantly evolving research community and is part of a a broader workflow in the field of Natural Language Processing. More specifically, in dialog systems, state tracking (also called belief tracking) refers to the evaluation of the user's goal as the dialogue progresses (over time). Accurate assessment of the dialog state is desirable as it provides robustness to errors that occur during the stages of natural language processing such as speech recognition errors or ambiguity errors that are inherent characteristic of the language. Recently, many models have been proposed for Dialog State Tracking. However, comparisons between models are rare and different research groups use different data from different domains. In addition, there is currently no common data set that allows experiments on dialog state tracking, as a result, such data must be collected and pre-processed from different sources, which is expensive and time consuming as there is no common line in the representation of such data. All of these issues hinder development of a state-of-the-art system.

The current thesis develops a DST model whose architecture was selected after extensive bibliographic research and it currently achieves state-of-the-art results in a widely-used and well-known dataset. First, there will be an introductory step into the key sections mentioned in the current thesis. Then, the essential theoretical background within the theoretical needs of this thesis will be described thoroughly. Next, we will explore various word processing techniques and key architectures used in the development of the thesis model. Afterwards, there will be an extensive analysis of the basic model architecture and the experimental process, and the inspection of the results will follow. Finally, additional design modifications will be proposed in order to further improve the results of the model.

Keywords

Dialog State Tracking, belief tracking, natural language processing, machine learning, neural networks.

Ευχαριστίες

Ευχαριστώ τον καθηγητή Ανδρέα Σταφυλοπάτη και τα μέλη του εργαστηρίου Ευφυών Συστημάτων για την ευκαιρία που μου δόθηκε να εργαστώ στο συγκεκριμένο θέμα της διπλωματικής μου.

Ευχαριστώ ιδιαίτερα τον κύριο Δρ. Γεώργιο Σιόλα, ο οποίος μου προσέφερε την κάθε δυνατή βοήθεια και σωστή καθοδήγηση κατά την εκπόνηση της διπλωματικής μου εργασίας όπως επίσης και στο μάθημα Νευρωνικών Δικτύων που αποτέλεσε βασικό γνωσιακό υπόβαθρο για την εκτέλεση της παρούσας εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τους καθηγητές Γεώργιο Στάμου και Παναγιώτη Τσανάκα που συμμετέχουν στην τριμελή επιτροπή.

Ευχαριστώ τους φίλους μου για τη συνεχή υποστήριξη που μου προσέφεραν καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας και την οικογένειά μου που ήταν πάντα δίπλα μου.

"Trying is the first step towards failure"

Homer Simpson

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η ανάπτυξη μεθόδων για διαλογική επικοινωνία μεταξύ ανθρώπου και υπολογιστή έχει πολύ σημαντικά οφέλη. Προσωπικοί βοηθοί, αυτόματοι εξυπηρετητές, general purpose chatbots είναι τομείς οι οποίοι συνεχώς εξελίσονται και η διαλογική επικοινωνία είναι ένα απαραίτητο λειτουργικό τους στοιχείο. Μεγάλες εταιρίες μπορούν να μειώσουν το κόστος τους ως προς το ανθρώπινο δυναμικό χρησιμοποιώντας αυτόματους εξυπηρετητές στον τομέα της εξυπηρέτησης πελατών. Προσωπικοί βοηθοί μπορούν να κάνουν πιο εύκολες και γρήγορες απλές λειτουργίες που όμως θα ήταν χρονοβόρες για ανθρώπους λιγότερο εξοικειωμένους με την τεχνολογία ή για ανθρώπους με ειδικές ανάγκες. Κάτι τέτοιο όμως απαιτεί ομαλή και άψογη λειτουργικότητα από την πλευρά του υπολογιστή καθώς καλείται να αντιμετωπίσει πολλά προβλήματα της φυσικής γλώσσας όπως η ασάφεια, ο συμπερασμός από τα συμφραζόμενα, η μη αυστηρή χρήση κανόνων, το τεράστιο οντολογικό πεδίο, τα αφηρημένα νοήματα κ.α [61], [25], [32]. Η παρακολούθηση της διαλογικής κατάστασης αποτελεί ένα κομμάτι της ευρύτερης επεξεργασίας της φυσικής γλώσσας το οποίο έχει σκοπό να αποσαφηνίσει τους στόχους του χρήστη ώστε να οδηγήσει το σύστημα σε πιο στοχευμένου τύπου διαλόγου. Ένα διαλογικό σύστημα επικοινωνίας προσανατολισμένο σε έναν στόχο περιλαμβάνει μια συνεχή αλληλεπίδραση ανάμεσα σε μια μηχανή agent και τον άνθρωπο ο οποίος επιθυμεί να πετύχει το στόχο του μέσα από την ομιλία. Σε γενικές γραμμές, τέτοιου είδους συστήματα αποτελούνται από τέσσερα μέρη, την αυτόματη αναγνώριση ομιλίας (Automatic Speech Recognition) [10], την κατανόηση της φυσικής γλώσσας (Natural Language Understanding) [3], την παραγωγή κειμένου και ομιλίας (Text Generation and Text to Speech) [24] και έναν διαχειριστή (Dialogue Manager) [34]. Το dialog state tracking αποτελεί κομμάτι του δεύτερου μέρους, δηλαδή ανήκει στην κατανόηση της φυσικής γλώσσας. Γενικότερος σκοπός είναι ανάγκη να αναπτυχθούν σωστές μέθοδοι που θα προσφέρουν μια πιο ανθρώπινη φύση στην διαλογική επικοινωνία σε κάθε κομμάτι του συστήματος και ειδικός σκοπός της διπλωματικής είναι η ανάπτυξη τεχνικών στον τομέα του Dialog State Tracking .

1.1 Μηχανική Μάθηση

Η μηχανική μάθηση είναι ένας κλάδος της επιστήμης των υπολογιστών ο οποίος έχει ως αντικείμενο την ανάπτυξη τεχνικών εκμάθησης χρησιμοποιώντας κατά κύριο λόγο στατιστικές μεθόδους σε μεγάλου όγκου δεδομένα. Όταν αναφερόμαστε στον όρο (εκ)μάθηση εννοούμε τη δυνατότητα των υπολογιστών να βελτιώνουν σταδιακά μια μετρική αξιολόγησης μέσα από την επεξεργασία δεδομένων. Η ιστορία της μηχανικής μάθησης μπορούμε να πούμε ότι ξεκινάει περίπου στα μέσα του $18^{ου}$ αιώνα από το περίφημο θεώρημα του Άγγλου στατιστικολόγου Thomas Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

το οποίο εκφράζει την πιθανότητα να συμβεί ένα ενδεχόμενο A δεδομένου ότι συνέβη ένα ενδεχόμενο B δίνοντας έτσι την μαθηματική ιδέα εξαγωγής πληροφορίας μέσα από κάποιο δεδομένο. Ο $20^{ος}$ αιώνας ήταν γεμάτος από εφευρέσεις μοντέλων και μεθόδων όπως το Perceptron, ο αλγόριθμος Backpropagation με ιστορικό σημείο το 1997 όπου για πρώτη φορά ο υπολογιστής νικάει τον άνθρωπο στο σκάκι (IBM Deep Blue vs Garry Kasparov). Η μηχανική μάθηση όμως γνώρισε τεράστια εξέλιξη τον $21^{ο}$ αιώνα καθώς ο όγκος των δεδομένων είναι πρωτοφανής όπως επίσης και η τεχνολογική εξέλιξη η οποία δίνει τη δυνατότητα επεξεργασίας μεγάλης κλίμακας σε υψηλές ταχύτητες κάτι το οποίο αποτελούσε εμπόδιο παλιότερα στην εφαρμογή και ανάπτυξη των μοντέλων. Επομένως η μηχανική μάθηση είναι ένας τομέας άρρηκτα συνδεδεμένος με την τωρινή καθημερινότητα και ο συνδυασμός των συνεχώς αναπτυσσόμενων μεθόδων και τεχνολογικών επιτευγμάτων θέτουν νέες, απαιτητικότερες ανάγκες στη ζωή του ανθρώπου.

1.2 Χρήσιμες ορολογίες

Σε αυτήν την ενότητα θα περιγράψουμε συντόμως τις πιο συχνές ορολογίες που συναντώνται στα πλαίσια της διαλογικής κατάστασης (Dialog State Tracking) και που θα χρησιμοποιούνται εκτενώς στο υπόλοιπο της εργασίας προκειμένου ο αναγνώστης να έχει καλύτερη κατανόηση κι ευρύτερη εποπτεία. Δεν γίνεται αυστηρή τοποθέτηση στους ορισμούς γιατί ο σκοπός είναι περισσότερο η αποσαφήνιση αυτών των εννοιών.

- **Παρακολούθηση διαλογικής κατάστασης**: Είναι η κύρια θεματολογία και αντικείμενο της διπλωματικής. Ορίζουμε τη διαλογική κατάσταση ως ένα ¨χώρο μνήμης' ο οποίος διατηρεί πληροφορία σχετικά με την ιστορία του διαλόγου που αναπτύσσεται μεταξύ δυο μερών. Η παρακολούθηση αναφέρεται στη διαδικασία που εποπτεύει και αλλάζει αυτό το χώρο μνήμης βάσει του διαλόγου.

- **Έκφραση**: Η έκφραση (utterance) είναι μια πλήρης δήλωση που γίνεται από τα δυο μέρη που συμμετέχουν στο διάλογο.

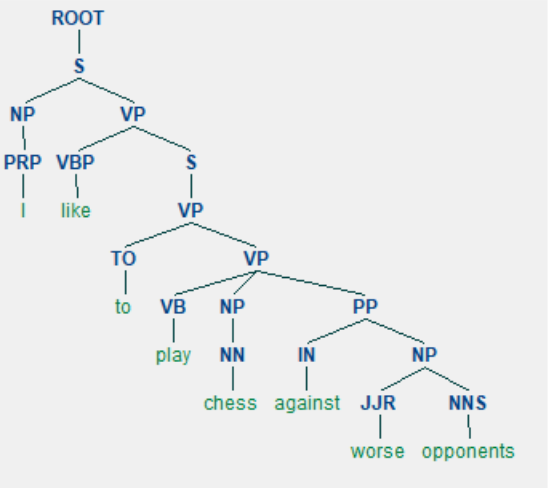
- **Στροφή**: Η στροφή (turn) ορίζεται ως το σημείο όπου έχουν συμβεί δυο συνεχόμενες εκφράσεις, μία από το κάθε μέρος, δηλαδή μια απάντηση και μια ανταπάντηση.

- **Θεματική ενότητα**: Η θεματική ενότητα (domain) ορίζεται ως το συσχετιζόμενο θέμα για το οποίο συμβαίνει ο διάλογος. Ένας διάλογος μπορεί να αποτελείται από μία ή πολλαπλές (multi-domain) θεματικές ενότητες.

- **Πεδίο-Τιμή**: Το πεδίο-τιμή (slot-value) είναι ένα ζευγάρι (pair) το οποίο διατηρεί πληροφορία για τη διαλογική κατάσταση (dialog state). Η διαλογική κατάσταση είναι ένα σύνολο από τέτοια ζευγάρια. Το πεδίο αναφέρεται στο είδος της πληροφορίας και η τιμή εκφράζει ποιοτικά ή ποσοτικά αυτό το είδος.

1.3 Ορισμός προβλήματος

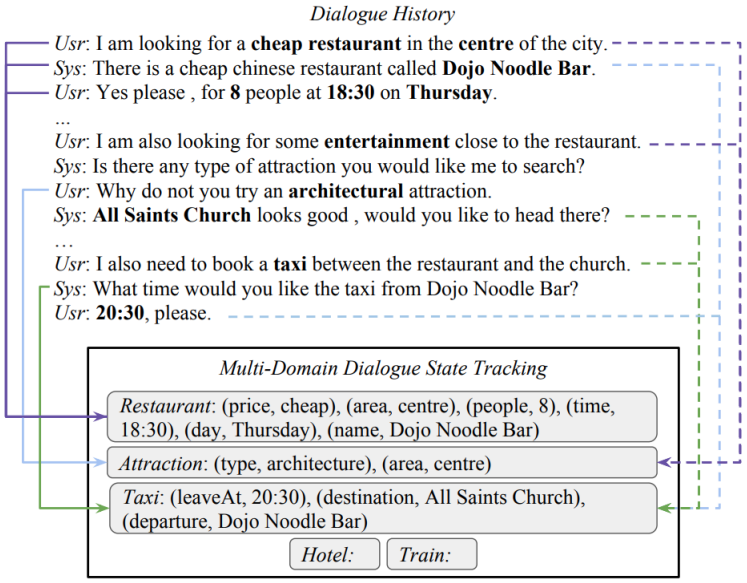
Το πρόβλημα παρακολούθησης της διαλογικής κατάστασης είναι ένα ανοιχτό πρόβλημα με αποτελέσμα να μην μπορεί να δωθεί ένας αυστηρός ή απόλυτος ορισμός. Προς το παρόν οι βασικές μετρικές αξιολόγησης δεν ξεπερνούν το 50% στην επιτυχία πρόβλεψης [73], [45], [17], [31] ,[69] κάτι το οποίο επιβεβαιώνει πόσο δύσκολο και απαιτητικό πρόβλημα είναι. Υπάρχουν σαφώς ορισμένες μοντελοποιήσεις με πιο συχνά χρησιμοποιούμενη την συμπλήρωση πεδίων (slot filling) [55], [66], [71]. Αναλυτικότερα, θα μπορούσαμε να ορίσουμε το πρόβλημα ως δοσμένου ενός διαλόγου που εξελίσσεται μεταξύ δυο ανθρώπων ή μεταξύ ανθρώπου μηχανής εν προκειμένω, σκοπός είναι η συμπλήρωση πεδίων συσχετισμένων με την εκάστοτε θεματική ενότητα. Παραδείγματος χάρει, τελείται ένας διάλογος που αφορά την κράτηση μιας κλίνης ξενοδοχείου και στόχος είναι να γίνει στοχευμένη συμπλήρωση στο πεδίο του αριθμού των ατόμων που απαιτεί αυτή η κράτηση. Σε ένα τέτοιο πρόβλημα η κλίμακα δυσκολίας είναι μεγάλη καθώς κατά τη διάρκεια που τελείται ο διάλογος υπάρχουν παρελθοντικές αναφορές, αναιρέσεις ή αλλαγές προτιμήσεων όπως επίσης και νοηματικές προκλήσεις. Το τελευταίο περιγραφεται καλύτερα με τη χρήση ενός παραδείγματος, όπου ο χρήστης ζητάει ένα δωμάτιο για τον ίδιο και τα δυο παιδιά του και το σύστημα πρέπει να αποφανθεί ότι ο συνολικός αριθμός των ατόμων είναι τρία.



S: Simple declarative clause
NP: Proper noun
PRP: Personal pronoun Phrase
VP: Verb Phrase
VBP: Verb, non-3rd person singular present
VB: Verb
NN: Noun, singular or mass
PP: Preposition Phrase
JJR: Adjective, comparative
NNS: Proper noun, plural

Σχήμα 1.1: Παράδειγμα συντακτικού δένδρου (Syntactic tree example)

Στα πλαίσια ανάλυσης της φυσικής γλώσσας η προαναφερθείσα δυσκολία αποτελεί άμεση συνέπεια καθώς η γλώσσα ως γενικότερο αντικείμενο μελέτης είναι ελάχιστα εξερευνημένο καθότι είναι άρρηκτα συνδεδεμένη με τον ανθρώπινο εγκέφαλο ο οποίος είναι εξίσου ανεξερεύνητο πεδίο. Η εκφραστική δύναμη της γλώσσας και η αδυναμία πλήρης μοντελοποίησής της μας οδηγεί στην επιστράτευση στατιστικών μεθόδων ή όπως ήδη αναφέρθηκε στο προηγούμενο κεφάλαιο, στη μηχανική μάθηση. Οι ρητές προγραμματιστικές μέθοδοι έχουν αποδειχθεί δύσκαμπτες και δύσκολα κλιμακούμενες καθώς δεν έχουν μεγάλη ανοχή σε σφάλματα τα οποία εμφανίζονται συχνά στη γλώσσα κι επιπλέον δεν μπορούν να συλληφθούν όλοι οι κανόνες πάνω στους οποίους εκδηλώνεται η γλώσσα. Ως αποτέλεσμα, αποφεύγουμε ρητές προγραμματιστικές μεθόδους (όπως γραμματικές, συντακτικές και σημασιολογικές αναλύσεις) και καταφεύγουμε σε μεθόδους μηχανικής μάθησης οι οποίες δίνουν τη δυνατότητα συνεχούς βελτίωσης μέσα από παραδείγματα. Στο σχήμα 1.1 ένα παράδειγμα συντακτικής ανάλυσης[1] της γλώσσας.



Σχήμα 1.2: Παράδειγμα διαλόγου με πολλούς τομείς (Example of a multi-domain conversation)

1.4 Αντικείμενο της διπλωματικής

Το αντικείμενο της διπλωματικής είναι η μελέτη και ανάπτυξη μηχανιστικών μεθόδων εκμάθησης που αντιμετωπίζουν το πρόβλημα της παρακολούθησης της διαλογικής κατάστασης όπως ορίστηκε σε προηγούμενο κεφάλαιο. Ο λόγος για τον οποίο γίνεται χρήση μεθόδων που ανήκουν στον τομέα της μηχανικής μάθησης έγγυται στη φύση του προβλήματος που όπως ήδη αναφέρθηκε δεν μπορεί να οριστεί αυστηρώς και με ντιτερμινιστικό τρόπο. Η παρακολούθηση της διαλογικής κατάστασης Dialog State Tracking είναι ένα βασικό στοιχείο των

[1] https://stanfordnlp.github.io/CoreNLP/

διαλογικών συστημάτων που προσανατολίζονται σε ένα συγκεκριμένο έργο όπως η κράτηση σε ένα εστιατόριο ή κράτηση εισητηρίων. Σκοπός του DST είναι να εξάγει τις προθέσεις του χρήστη που εκφράζονται κατά τη διάρκεια της συζήτησης και να τις κωδικοποιήσει σε μια μορφή συνόλου από πεδία και τιμές. Η σωστή πρόβλεψη των τιμών μπορεί να αποβεί κρίσιμη για την δράση που θα λάβει το σύστημα και πώς θα χειριστεί γενικά το διάλογο για να υπάρξει μια ομαλή μετάβαση σε επόμενο βήμα. Ένα παράδειγμα διαλόγου με την αντίστοιχη κατάσταση φαίνονται στο σχήμα 1.2

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη (Artificial Intelligence-AI) είναι ένας τομέας της επιστήμης των υπολογιστών που όπως υποδηλώνει το όνομά του ασχολείται με την προσομοίωση νοήμον συμπεριφοράς από τους υπολογιστές. Δηλαδή, οποιοδήποτε πρόγραμμα είναι σε θέση να μιμείται σε έναν βαθμό την ανθρώπινη συμπεριφορά, μπορούμε να πούμε πως ανήκει στον τομέα της τεχνητής νοημοσύνης. Η τεχνητή νοημοσύνη επεκτείνεται σε πολλά παρακλάδια όπως ο επαγωγικός λογικός προγραμματισμός, οι ευριστικοί αλγόριθμοι, οι πιθανοτικοί μέθοδοι κ.ά. Επειδή δεν υπάρχουν αυστηρά όρια μπορούμε να πούμε με επιείκια ότι στη μηχανική μάθηση (machine learning) ανήκουν όσα παρακλάδια έχουν τη φιλοσοφία της ανάπτυξης προγραμμάτων που έχουν την ικανότητα να τροποποιούν τον εαυτό τους.

2.2 Μηχανική Μάθηση

Η μηχανική μάθηση (Machine Learning-ML) όπως ήδη αναφέρθηκε αποτελεί τομέα της τεχνητής νοημοσύνης με αντικείμενο την ανάπτυξη προγραμμάτων / μηχανιστικών μεθόδων που έχουν την ικανότητα να μαθαίνουν. Συγκεντρώνει ένα μεγάλο πεδίο γνώσης από πολλές επιστήμες μέσα στις οποίες ανήκουν η στατιστική, ο μαθηματικός λογισμός, η φιλοσοφία, η γνωστική επιστήμη κ.ά. Σχετικά με τον ορισμό της μάθησης από προγράμματα ο Tom M. Mitchell έθεσε πιο επίσημα πως ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από μια εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης Π, αν η επίδοσή του στις εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E (A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E)[43, p. 2]. Βάσει του τρόπου εκμάθησης, η μηχανική μάθηση χωρίζεται σε τρεις πολύ βασικές κατηγορίες, την επιβλεπόμενη μάθηση (επιτηρούμενη μάθηση, μάθηση με επίβλεψη, Supervised learning), την μη επιβλεπόμενη μάθηση (μάθηση χωρίς επίβλεψη, Unsupervised learning) και την ενισχυμένη μάθηση (Reinforcement learning) οι οποίες περιγράφονται αναλυτικότερα στη συνέχεια.

2.2.1 Επιβλεπόμενη Μάθηση

Στην επιβλεπόμενη μάθηση (supervised learning) τα δεδομένα (dataset) με τα οποία τροφοδοτείται ο αλγόριθμος είναι σημειωμένα (labeled) με μια τιμή. Ο σκοπός του αλγορίθμου είναι μέσω της εκπαίδευσης από τα δεδομένα να προσεγγίσει μια συνάρτηση f η οποία αντιστοιχεί κάθε παράδειγμα δεδομένου x σε μια ετικέτα (label) y. Η συνήθης διαδικάσια που ακολουθείται σε αυτό το είδος μάθησης είναι ο χωρισμός του συνόλου δεδομένων σε δεδομένα εκπαίδευσης (Training Data) και δεδομένα δοκιμής (Test Dataset). Με τα πρώτα γίνεται η εκπαίδευση του αλγορίθμου, ενώ με τα δεύτερα αξιολογούμε την επιτυχία του αλγορίθμου, δηλαδή την ικανότητά του να γενικεύσει την αντοιστιχία από τα δεδομένα στις ετικέτες. Η τιμή της ετικέτας του εκάστοτε δεδομένου είναι αυτό που μαθαίνει στον αλγόριθμο πώς πρέπει να αντοιστοιχηθεί το συγκεκριμένο δεδομένο εξού και το όνομα ¨επιβλεπόμενη' μάθηση. Οι ετικέτες μπορούν να έχουν είτε ποιοτικές είτε ποσοτικές τιμές. Γι΄ αυτό υπάρχουν δύο είδη επιβλεπόμενης μάθησης, η κατηγοριοποίηση (Classification) και η παλινδρόμηση (Regression).

- Κατηγοριοποίηση (Classification): Στην κατηγοριοποίηση, η ετικέτα στα δεδομένα εκφράζει ένα ποιοτικό χαρακτηριστικό. Στόχος του αλγορίθμου είναι να ταξινομήσει τα δεδομένα στην κατηγορία την οποία ανήκουν. Ο διαχωρισμός καλλιτεχνικών πινάκων σε ¨Μανιερισμό¨, Ἰμπρεσιονισμό¨, ¨Ἐξπρεσιονισμό¨, ή ο διαχωρισμός ενός e-mail σε spam ή not-spam ανήκουν σε προβήματα κατηγοριοποίησης.

- Παλινδρόμηση (Regression): Στην παλινδρόμηση, η ετικέτα στα δεδομένα εκφράζει ένα ποσοτικό χαρακτηριστικό. Σε αυτήν την περίπτωση, η τιμή είναι συνεχής και ο στόχος του αλγορίθμου δεν είναι να ταξινομήσει το δεδομένο εισόδου σε κάποια κλάση, αλλά να προβλέψει μια τιμή η οποία εκδηλώνει ποσότητα. Η πρόβλεψη της τιμής πώλησης ενός προϊόντος ή η πρόβλεψη του φορολογικού δείκτη αποτελούν προβήματα παλινδρόμησης.

Θα ήταν σωστό να αναφέρουμε πως τα δύο είδη της επιβλεπόμενης μάθησης δεν είναι αμοιβαίως αποκλειόμενα καθώς είναι εφικτό να μετατρέψουμε ένα πρόβλημα παλινδρόμησης σε κατηγοριοποίηση και αντίστροφα. Για παράδειγμα εάν έχουμε μια τιμή προϊόντος, μπορούμε να δημιουργήσουμε τις κλάσεις ¨ψθηνό¨, ¨κανονικό' και ¨ακριβό¨. Μερικές φορές όμως μπορεί να μην είναι εφικτό να γίνει μια τέτοια μετατροπή.

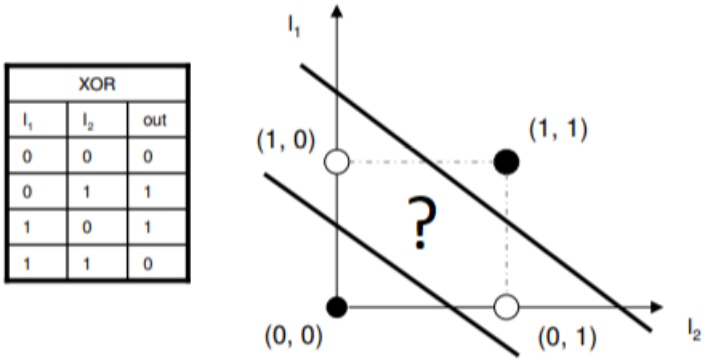
2.2.2 Μη Επιβλεπόμενη Μάθηση

Στην μη επιβλεπόμενη μάθηση (unsupervised learning), σε αντίθεση με την επιβλεπόμενη, τα δεδομένα δεν έχουν κάποια ετικέτα. Σε αυτήν την περίπτωση χρησιμοποιείται η τεχνική της ομαδοποίησης (clustering). Σκοπός είναι ο αλγόριθμος να εκπαιδευτεί στα δεδομένα με στόχο να τα διαχωρίσει σε ομάδες βάσει κοινών χαρακτηριστικών. Αυτό είναι ένα ιδιαίτερα δύσκολο πρόβλημα καθώς πολλές φορές ο αριθμός των ομάδων δεν είναι γνωστός εκ των προτέρων. Για παράδειγμα έστω ότι έχουμε εικόνες από πρόσωπα ανθρώπων και θέλουμε να τις διαχωρίσουμε έτσι ώστε η κάθε ομάδα να έχει εικόνες από τον ίδιο άνθρωπο. Δε γνωρίζουμε εκ των προτέρων πόσοι άνθρωποι υπάρχουν. Ο διαχωρισμός μορίων σε ¨φάρμακο'

ή ¨όχι φάρμακο' είναι παράδειγμα όπου γνωρίζουμε τις κλάσεις διαχωρισμού αλλά και πάλι εκλείπει η ετικέτα για κάθε μόριο, επομένως ο αλγόριθμος αποσκοπεί στο να ανακαλύψει αυτήν την ποιοτική διαφορά που καθιστά ένα μόριο ως φάρμακο ή μη.

2.2.3 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση (reinforcement learning) χρησιμοποιεί τεχνικές οι οποίες λόγω της γενικότητάς τους χρησιμοποιούνται και σε τομείς εκτός της μηχανικής μάθησης, όπως η θεωρία παιγνίων, συστήματα πολλαπλών πρακτόρων κ.α [57], [1]. Η βασική τεχνική που χρησιμοποιεί είναι η ανταμοιβή και η τιμωρία (reward and punishment). Δηλαδή ένας πράκτορας ανταμοίβεται ή τιμωρείται από το περιβάλλον του για τις κινήσεις που επιλέγει με στόχο να επιτελέσει μια διαδικασία. Διαφέρει ως μέθοδος από τις προηγούμενες κατηγορίες μάθησης. Στην ενισχυτική μάθηση δεν απαιτείται τα δεδομένα να έχουν ετικέτες. Βασικός στόχος είναι μια ισορροπία ανάμεσα στην εξερεύνηση (exploration) του περιβάλλοντος και την εκμετάλλευση (exploitation) της ήδη υπάρχουσας γνώσης. Η ενισχυτική μάθηση χρησιμοποιήθηκε ως μέθοδος από την ομάδα Deepmind για να εκπαιδευτεί από το μηδέν ο αλγόριθμος AlphaGo Zero ο οποίος ήταν ο πρώτος που κατάφερε να νικήσει τον άνθρωπο στο περίφημο παιχνίδι Go.[1]



Σχήμα 2.1: Μη γραμμικά διαχωρίσημο πρόβλημα XOR

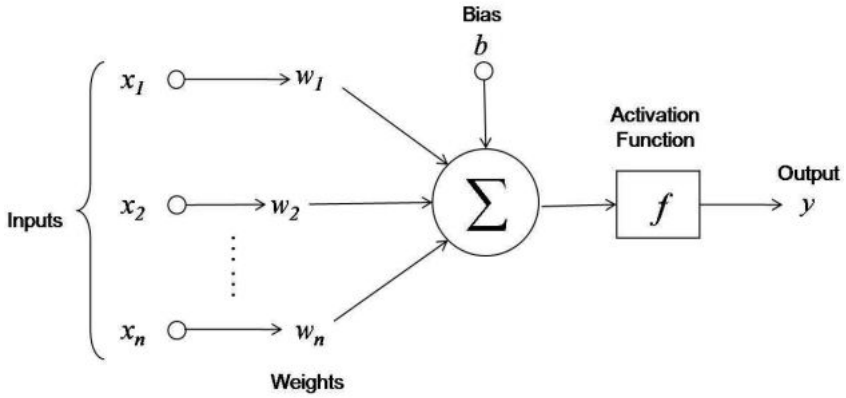
2.3 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks) είναι δίκτυα εμπνευσμένα από τα βιολογικά νευρωνικά δίκτυα του εγκεφάλου. Είναι γνωστό ότι οι συνάψεις των νευρώνων σε έναν εγκέφαλο είναι ιδιαίτερα πολλές και περίπλοκες επομένως τα τεχνητά νευρωνικά δίκτυα δεν προσπαθούν να αντιγράψουν το βιολογικό δίκτυο αλλά να αποτελέσουν μια απλουστευμένη μοντελοποίηση αυτού.

Η ιστορία της δημιουργίας του πρώτου τεχνητού νευρώνα ξεκινάει το 1943[39] όπου οι Warren McCulloch και Walter Pitts έδειξαν ότι νευρώνες με δυαδική συνάρτηση ενεργοποίη-

[1] https://deepmind.com/blog/article/alphago-zero-starting-scratch

σης ήταν ανάλογο με προτάσεις λογικής πρώτης τάξης. Μία από τις δυσκολίες με τον νευρώνα McCulloch-Pitts ήταν η απλότητα του. Επέτρπε μόνο δυαδικές εισόδους και εξόδους, χρησιμοποιούσε μόνο τη βηματική συνάρτηση ενεργοποίησης και δεν ενσωμάτωνε τη στάθμιση των διαφορετικών εισόδων.



Σχήμα 2.2: Η δομή ενός νευρώνα Perceptron

Στη συνέχεια, το 1958 ο Frank Rosenblatt εφήυρε τον πρώτο τεχνητό νεωρώνα και δημιούργησε τα πρώτα δίκτυα. Η δεκαετία του ′70 ήταν γεμάτες ανακαλύψεις με αποκορύφωμα το 1969, όπου οι Minsky και Papert στο περίφημο βιβλίο τους [42] απέδειξαν ότι οι νευρώνες perceptrons μπορούν να εκπαιδευτούν μόνο για την επίλυση γραμμικών διαχωριζόμενων προβλημάτων. Για παράδειγμα, ένα από τα πιο καταδικαστικά παραδείγματα μη γραμμικών διαχωριζόμενων προβλημάτων είναι το αποκλειστικό OR (XOR)[2]. Προκειμένου το δίκτυο να επιλύσει επιτυχώς το πρόβλημα XOR, η έξοδός του μπορεί να είναι αληθής μόνο αν μία από τις εισόδους του είναι αληθής, αλλά όχι και οι δύο. Αυτό αποτέλεσε ένα μεγάλο πλήγμα στην ανάπτυξη των τεχνητών νευρώνων. Οι Minsky και Papert γνώριζαν ότι πολλά στρώματα θα μπορούσαν να λύσουν αυτό το πρόβλημα, αλλά δεν υπήρχε αλγόριθμος για την εκπαίδευση ενός τέτοιου δικτύου. Χρειάστηκαν 17 χρόνια μέχρι να επινοηθεί ένας τέτοιος αλγόριθμος, γνωστός ως backpropagation. Σήμερα τα τεχνητά νευρωνικά δίκτυα έρχονται να δόσουν λύση σε όσα προβλήματα δεν μπορεί να χρησιμοποιηθεί ο κλασικός ρητός προγραμματισμός.

2.3.1 Ο αλγόριθμος Perceptron

Ο αλγόριθμος ή νευρώνας Perceptron είναι το δομικό στοιχείο των νευρωνικών δικτύων. Ως αλγόριθμος χρησιμοποιείται για την εκπαίδευση ενός δυαδικού ταξινομητή ή αλλιώς μιας συνάρτησης κατωφλιού (threshold function). Αυτή η συνάρτηση αντιστοιχεί κάθε είσοδοδιάνυσμα $\boldsymbol{x} = [x_1, x_2, ..., x_n] \in \mathbb{R}^n$ σε μια τιμή εξόδου $f(\boldsymbol{x})$. Η έξοδος του νευρώνα υπολογίζεται από τη σχέση:

$$y = f(\boldsymbol{w} \cdot \boldsymbol{x} + b) \tag{2.1}$$

όπου το $\boldsymbol{w} \cdot \boldsymbol{x} = \sum_{i=1}^{n} x_i w_i$ είναι το εσωτερικό γινόμενο (dot product) του διανύσματος εισόδου $\boldsymbol{x}$ με το διάνυσμα βαρών $\boldsymbol{w} = [w_1, w_2, ..., w_n] \in \mathbb{R}^n$ του νευρώνα και b είναι η πόλωση

[2] https://en.wikipedia.org/wiki/XOR_gate

(bias). Η τιμή της εξόδου y εξαρτάται από τη συνάρτηση ενεργοποίησης f. Στη γενική περίπτωση $y \in \mathbb{R}$ αλλά μπορεί επίσης και $y \in [0,1]$ ή ακόμα $y \in \{0,1\}$.

Ο νευρώνας Perceptron ανήκει στην κατηγορία των γραμμικών ταξινομητών (linear classifier) δηλάδη διαχωρίζει την είσοδο γραμμικά σε δυο κλάσεις. Θα δούμε στη συνέχεια ότι ένας νευρώνας δεν επαρκεί για να λύσει μη γραμμικά διαχωρίσιμα προβλήματα όπως αναφέρθηκε προηγούμενως με το πρόβλημα της XOR όπως φαίνεται και στο σχήμα 2.1. Όμως ένα πολυ-επίπεδο τεχνητό νευρωνικό δίκτυο είναι εφικτό. Στο σχήμα 2.2 παρουσιάζουμε τη δομή που έχει ένας τεχνητός νευρώνας έτσι όπως χρησιμοποιείται επί του παρόντος.

2.3.2 Θεώρημα Καθολικής Προσέγγισης Universal Approximation Theorem

Το θεώρημα καθολικής προσέγγισης είναι ένα θεώρημα που αποδεικνύει την 'δύναμη' που έχουν τα εμπρόστια τροφοδοτούμενα δίκτυα (feedforward networks) δηλαδή δίκτυα που αποτελούνται από επίπεδα νευρώνων και η πληροφορία 'ρέει' από το ένα επίπεδο στο επόμενο προς μία κατεύθυνση. Αυτό που δηλώνει το θεώρημα είναι ότι ένα τέτοιο δίκτυο με ένα μόνο κρυφό επίπεδο (πέρα από το επίπεδο εισόδου και το επίπεδο εξόδου) αποτελούμενο από πεπερασμένο αριθμό νευρώνων είναι ικανό να προσεγγίσει με όσο μικρό σφάλμα επιθυμούμε οποιαδήποτε συνεχή συνάρτηση ορισμένη σε ένα κλειστό σύνολο των πραγματικών αριθμών. Εκφράζοντάς το πιο επίσημα χρησιμοποιώντας μαθηματικούς όρους το θεώρημα καθολικής σύγκλισης δηλώνει ότι:

Έστω μια συνάρτηση $\varphi : \mathbb{R} \to \mathbb{R}$ η οποία είναι μη σταθερή, συνεχής και φραγμένη, γνωστή ως συνάρτηση ενεργοποίησης. Επίσης, συμβολίζουμε με I_m τον m-διάστατο υπερκύβο στο χώρο $[0,1]^m$ και με $C(I_m)$ την κλάση όλων των πραγματικών συνεχών συναρτήσεων στο I_m. Τότε, δοσμένου οποιοδήποτε αριθμού $\varepsilon > 0$ και οποιαδήποτε συνάρτηση $f \in I_m$, υπάρχει ένας ακέραιος N, σταθερές α_i, $\beta_i \in \mathbb{R}$ και διανύσματα $\boldsymbol{w_i} \in \mathbb{R}^m$ για $i = 1, 2, ..., N$ τέτοια ώστε να ορίσουμε τη συνάρτηση:

$$F(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \varphi(\boldsymbol{w_i}^T \boldsymbol{x} + \beta_i) \tag{2.2}$$

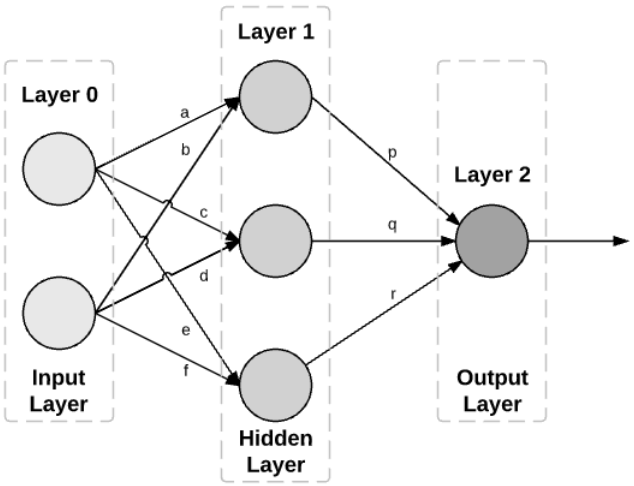
ως μια προσέγγιση της συνάρτησης f καθώς για όλα τα $\boldsymbol{x} \in I_m$ ισχύει ότι:

$$|F(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon \tag{2.3}$$

Δηλαδή αρκεί ένα κρυφό επίπεδο (hidden layer) και ένας νεωρώνας στο στάδιο εξόδου όπως φαίνεται στο σχήμα 2.3 ώστε να προσεγγίσουμε οποιαδήποτε συνάρτηση.

2.3.3 Πολυεπίπεδα Νευρωνικά Δίκτυα Multilayer Perceptrons (MLP)

Στο κεφάλαιο 2.3.2 έγινε μια εισαγωγή στα πολυεπίπεδα νευρωνικά δίκτυα (multilayer perceptrons). Όπως είδαμε ένας απλός νευρώνας Perceptron δεν είναι σε θέση να επιλύσει μη γραμμικά προβλήματα αλλά ένα πολυεπίπεδο νευρωνικό δίκτυο μπορεί να προσεγγίσει οποιαδήποτε μη γραμμική συνάρτηση. Τα πολυεπίπεδα δίκτυα όπως δηλώνει και το όνομά τους είναι δίκτυα που αποτελούνται από μια ακολουθία επιπέδων Perceptrons. Κάθε ενδιάμεσο ή κρυφό



Σχήμα 2.3: Feedforward Network with a single hidden layer

επίπεδο (hidden layer) έχει ως είσοδο τις εξόδους του προηγούμενου επιπέδου και τροφοδοτεί με την έξοδό του το επόμενο επίπεδο. Επιπλέον το πρώτο επίπεδο είναι το επίπεδο εισόδου, δηλαδή το διάνυσμα $\boldsymbol{x}$ και το τελευταίο επίπεδο είναι το επίπεδο εξόδου $\boldsymbol{y}$. Το τελευταίο επίπεδο μπορεί να έχει μία ή περισσότερες εξόδους ανάλογα το είδος του προβλήματος. Για παράδειγμα όπως αναφέρθηκε στο κεφάλαιο 2.2.1 μπορεί ένα νευρωνικό δίκτυο να χρειαστεί να ταξινομήσει μια είσοδο σε κλάσεις με τιμές 'μπλε', 'κόκκινο', 'πράσινο', επομένως οι έξοδοι του δικτύου πρέπει να είναι τρεις, με την τιμή της κάθε μίας να εκφράζει κατά πόσο η είσοδος ανήκει στην εκάστοτε κατηγορία. Οι νευρώνες που ανήκουν στο ίδιο επίπεδο δεν έχουν καμία μορφή σύνδεσης μεταξύ τους. Όπως αναφέρθηκε, η είσοδος ενός νευρώνα Perceptron ενός επιπέδου είναι η έξοδος όλων των νευρώνων του προηγούμενου επιπέδου. Η ροή της πληροφορίας αρχίζει από το πρώτο επίπεδο που είναι η είσοδος και διαχέεται σειριακά από το ένα επίπεδο στο επόμενο. Γι' αυτό το λόγο τα πολυεπίπεδα δίκτυα ανήκουν στην κατηγορία των πρόσθια τροφοδοτούμενων δικτύων (feedforward network).

Όταν τα κρυφά (ενδιάμεσα) επίπεδα είναι παραπάνω από ένα τότε το δίκτυο ανήκει στην κατηγορία των βαθιά νευρωνικών δικτύων (deep neural networks). Τα βαθιά νευρωνικά δίκτυα έχουν αποδείξει μεγάλη επιτυχία σε πολλούς τομείς προβλημάτων. Η αρχιτεκτονική τους επιτρέπει να εξάγουν χαρακτηριστικά στοιχεία από την είσοδο σε πολύ αφηρημένους βαθμούς. Δηλαδή, τα βάρη κάθε επιπέδου εκφράζουν και ένα διαφορετικό στοιχείο της εισόδου. Για παράδειγμα, σε ένα σύνολο από εικόνες γραμμάτων, ένα επίπεδο μπορεί να εκφράζει τις ευθείες γραμμές, ένα άλλο τις γωνίες ή την πυκνότητα σε κάποιο σημείο του σχήματος ενός γράμματος.

2.3.4 Βασικές Συναρτήσεις και Αλγόριθμοι Εκπαίδευσης

2.3.4.1 Συνάρτηση Ενεργοποίησης Activation function

Η συνάρτηση ενεργοποίησης [44] είναι η συνάρτηση όπου εφαρμόζεται στην έξοδο του νευρώνα. Η χρήση αυτής της συνάρτησης εξυπηρετεί δύο σκοπούς. Ο πρώτος είναι ότι

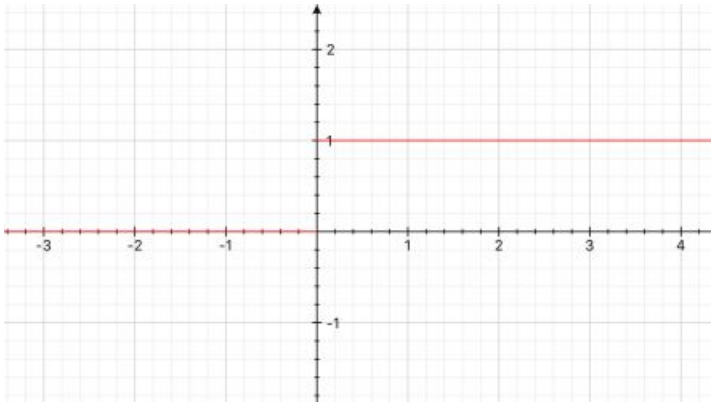
αντιστοιχεί τις τιμές εξόδου του νευρώνα σε ένα κλειστό σύνολο. Εάν δεν υπήρχε καθόλου η συνάρτηση αυτή τότε η έξοδος του νευρώνα από τη σχέση 2.1 θα ήταν απλά:

$$y = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{2.4}$$

Παρατηρούμε λοιπόν στη σχέση 2.4 ότι η τιμή y μπορεί να πάρει τιμές στο διάστημα $(-\infty, +\infty)$. Επομένως, δεν γίνεται αντιληπτό πότε ο νευρώνας θα πρέπει να 'πυροδοτήσει' δηλαδή να 'δώσει' σημασία στην έξοδο ή να την αποκόψει. Χρησιμοποιώντας λοιπόν μια συνάρτηση ενεργοποίησης $f : \mathbb{R} \to [0, 1]$ μεταφέρουμε την έξοδο σε ένα κλειστό σύνολο με αποτέλεσμα πλέον να γίνεται κατανοητό ότι τιμές κοντά στο μηδέν σημαίνουν πως δεν έχουν σημασία, ενώ τιμές κοντά στο ένα εκδηλώνουν την πυροδότηση του νευρώνα.

Ο δεύτερος σκοπός που επιτελεί η συνάρτηση ενεργοποίησης (και βάσει του κεφαλαίου 2.3.2 είναι απαραίτητος) είναι η εισαγωγή μη γραμμικότητας στο σύστημα. Η μη γραμμικότητα είναι αναγκαία προκειμένου το νευρωνικό δίκτυο να προσεγγίζει μη γραμμικές συναρτήσεις. Παρακάτω, παρουσιάζονται παραδείγματα συναρτήσεων ενεργοποίησης. Για τις επόμενες εξισώσεις που παρουσιάζονται εντός του κεφαλαίου, η μεταβλητή x θεωρείται η έξοδος του νευρώνα πριν εφαρμοστεί η συνάρτηση ενεργοποίησης όπως εκφράζεται από τη σχέση 2.4:

- *Βηματική Συνάρτηση (Step function)*:



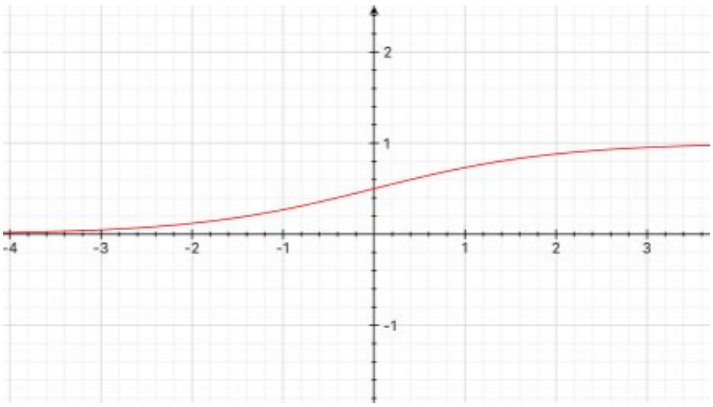
Σχήμα 2.4: Βηματική συνάρτηση Step function

$$f(x) = \begin{cases} 1, & x > 0 \\ 0, & otherwise \end{cases} \tag{2.5}$$

Η βηματική συνάρτηση δε χρησιμοποιείται ως συνάρτηση ενεργοποίησης. Η εκπαίδευση ενός νευρωνικού δικτύου απαιτεί τον υπολογισμό παραγώγων συναρτήσεων. Η βηματική συνάρτηση όπως φαίνεται και στο σχήμα 2.4 δεν είναι παραγωγίσημη στο σημείο μηδέν. Γίνεται όμως αναφορά σε αυτήν καθώς αποτελεί τη βασική ιδέα που πρέπει να πληρεί μια συνάρτηση ενεργοποίησης, δηλαδή να αντιστοιχεί τις τιμές του $\mathbb{R}$ σε ένα κλειστό πεδίο εν προκειμένου το $[0, 1]$ και να προσδίδει στο σύστημα μη γραμμικότητα. Όπως θα φανεί και στη συνέχεια, οι περισσότερες συναρτήσεις μοιάζουν ποιοτικά στη βηματική με τη μόνη διαφορά ότι είναι παντού παραγωγίσιμες.

- *Σιγμοειδής Συνάρτηση (Sigmoid function)*:

  Όπως φαίνεται και στο σχήμα 2.5 η σιγμοειδής συνάρτηση έχει ένα 'S' σχήμα. Το πεδίο τιμών της είναι το $(0, 1)$. Η σιγμοειδής συνάρτηση εμφανίζει κάποια προβήματα όπως η εξαφάνιση της κλίσης (Vanishing gradient problem). Επίσης, το πεδίο τιμών δεν είναι κεντραρισμένο στο μηδέν για να υπάρχει συμμετρία. Τέλος, η χρήση σιγμοιδής συνάρτησης ως συνάρτηση ενεργοποίησης οδηγεί σε πιο αργή σύγκλιση (slow convergence) κατά τη διάρκεια της εκπαίδευσης.

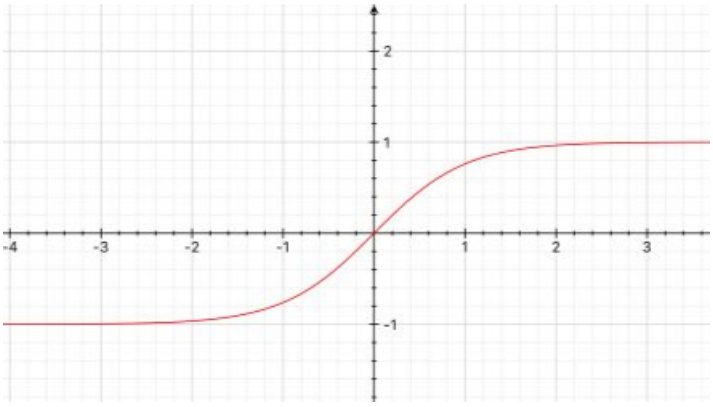


Σχήμα 2.5: Σιγμοειδής Συνάρτηση (Sigmoid function)

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.6}$$

- *Υπερβολική εφαπτομένη (Hyperbolic tangent)*:

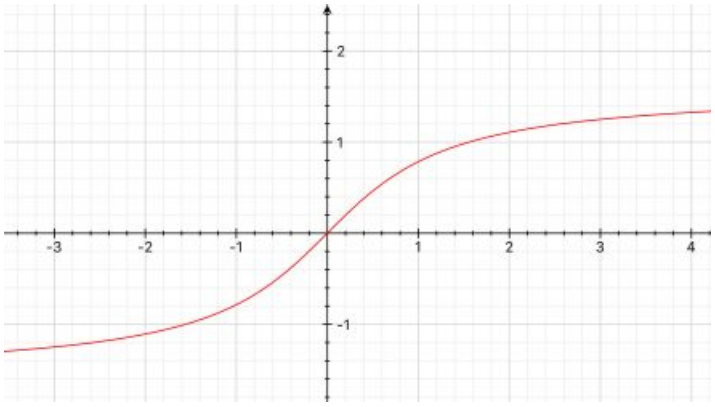
  Η υπερβολική εφαπτομένη είναι μια βελτίωση της σιγμοειδής συνάρτησης. Το πεδίο τιμών της είναι το $(-1, 1)$ και είναι κεντραρισμένο στο μηδέν. Προτιμάται από τη σιγμοειδή κυρίως ευκολότερης βελτιστοποίησης αλλά υστερεί και αυτή από το πρόβλημα της εξαφάνισης κλίσης.



Σχήμα 2.6: Υπερβολική εφαπτομένη Tanh function

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 = 2\sigma(2x) - 1 \tag{2.7}$$

- *Αντίστροφη Εφαπτομένη (Arctan)*:

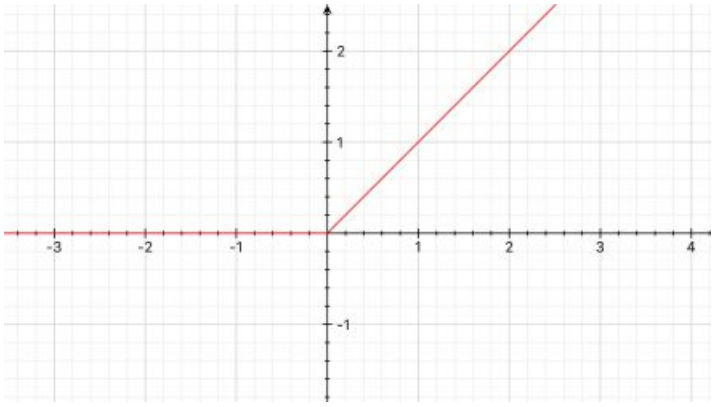


Σχήμα 2.7: Αντίστροφη Εφαπτομένη (Arctan)

$$f(x) = \arctan(x) \tag{2.8}$$

- *Rectified Linear Unit (ReLU)*:

  Το πρόβλημα στην εξαφάνιση κλίσης έρχεται να δώσει η συνάρτηση ReLU η οποία φαίνεται στο σχήμα 2.8. Πλέον, σχεδόν όλα τα μοντέλα βαθιάς μάθησης χρησιμοποιούν την ReLU καθώς έχει αποδειχθεί να είναι αρκετά πιο αποδοτική. Ο περιορισμός είναι ότι μπορεί να χρησιμοποιηθεί μόνο στα ενδιάμεσα επίπεδα νευρώνων (hidden layers). Στο τελικό επίπεδο εξόδου ανάλογα με το είδος του προβλήματος μπορεί να χρησιμοποιηθεί η συνάρτηση SoftMax για πρόβλημα κατηγοριοποίησης ή μια απλή γραμμική συνάρτηση για πρόβλημα παλινδρόμησης. (βλ. κεφάλαιο 2.2.1).



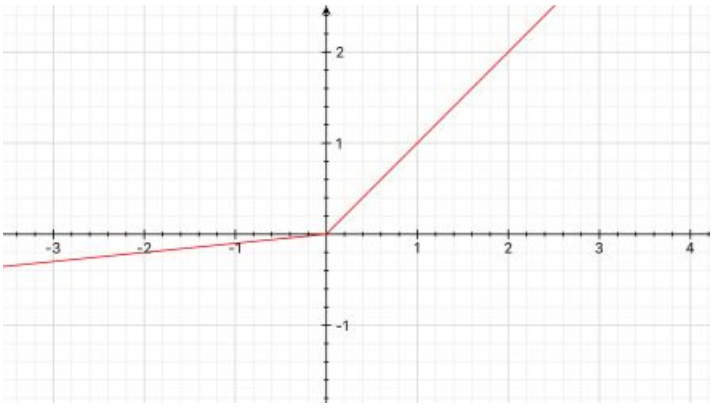
Σχήμα 2.8: Rectified Linear Unit (ReLU)

$$f(x) = max(0, x) = \begin{cases} x, & x > 0 \\ 0, & otherwise \end{cases} \tag{2.9}$$

Ένα άλλο πρόβλημα με τη ReLu είναι ότι μερικές κλίσεις μπορεί να είναι 'ασταθείς' κατά τη διάρκεια της εκπαίδευσης και να οδηγηθούν σε τελμάτωση. Μπορεί να προκαλέσει κάποια ενημέρωση βάρους το οποίο να μην ενεργοποιήσει ποτέ ξανά την έξοδο σε κανένα δεδομένου εισόδου. Αυτό το φαινόμενο ονομάζεται Νεκροί Νευρώνες (Dead Neurons, Dying ReLU). Τη λύση σε αυτό το πρόβλημα το δίνει μια βελτιωμένη παραλλαγή της ReLU, η Leaky ReLU.

- *Leaky Rectified Linear Unit (Leaky ReLU)*:

  Σε αντίθεση με τη ReLU, η Leaky ReLU οπως φαίνεται στο σχήμα 2.9 δεν στέλνει όλες τις αρνητικές τιμές στο μηδέν, αλλά διατηρεί μια μικρή κλίση α συνήθως $\alpha = 0.01$ ώστε να βοηθήσει νεκρούς νευρώνες να ανακτηθούν και πάλι.

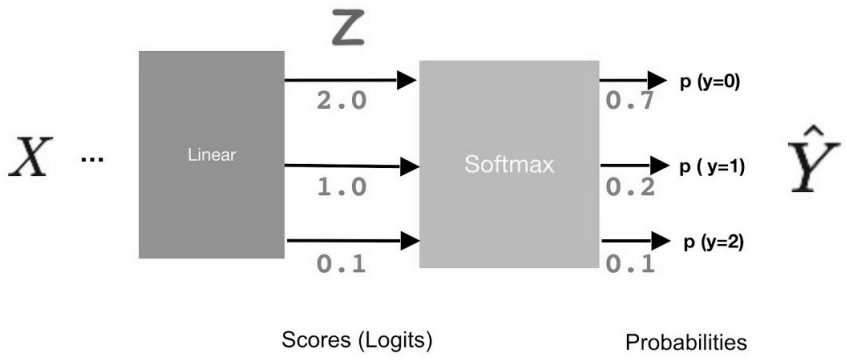


Σχήμα 2.9: Leaky ReLU

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases} \tag{2.10}$$

- *Συνάρτηση Softmax*: Η συνάρτηση Softmax [51] δεν είναι αυστηρά συνάρτηση ενεργοποίησης. Μπορεί να χρησιμοποιηθεί ανεξάρτητα από κάποια συνάρτηση ενεργοποίησης στο τελικό στάδιο της εξόδου του νευρωνικού δικτύου.

  Η έξοδος του δικτύου δίνει μια ποσοτική τιμή η οποία εκφράζει το κατά πόσο η είσοδος ανήκει στην κλάση που αντιπροσωπεύει αυτή η έξοδος. Για παράδειγμα όπως φαίνεται στο σχήμα 2.10 η έξοδος του δικτύου πριν εφαρμοστεί η Softmax είναι $0.1, 1.0, 2.0$. Δηλαδή η κλάση που αντιστοιχεί καλύτερα η είσοδος $\boldsymbol{X}$ είναι αυτή που αντιστοιχεί στην έξοδο 2.0. Το πρόβλημα είναι όμως ότι δε θέλουμε να κρατάμε μόνο τη μεγαλύτερη τιμή (αλλιώς θα χρησιμοποιούσαμε απλώς μια $max()$ συνάρτηση) αλλά να έχουμε μια γενική εικόνα της ποσοστιαίας αντιστοίχισης της εισόδου σε κάποια κλάση. Οι τιμές



Σχήμα 2.10: Softmax

όμως της εξόδου του κάθε νευρώνα μπορούν να κυμαίνονται σε ένα πολύ μεγάλο έυρος. Επομένως μέσω της Softmax καταφέρνουμε να κανονικοποιήσουμε τις τιμές και πλέον οι τιμές $[y_1, y_2, ..., y_j]$ της εξόδου να έχουν την ιδιότητα:

$$\sum_{k=1}^{K} y_k = \sum_{k=1}^{K} Softmax(\boldsymbol{z})_k = 1 \tag{2.11}$$

Επομένως αυτή η σχέση δίνει τη δυνατότητα οι έξοδοι να χρησιμοποιούνται ως πιθανότητες. Δηλαδή, η κάθε τιμή εκφράζει την πιθανότητα η είσοδος να ανήκει στην αντίστοιχη κλάση.

$$Softmax(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2.12}$$

2.3.4.2 Συνάρτηση Κόστους Cost Function

Στη βιβλιογραφία οι συναρτήσεις Loss function και Cost function χρησιμοποιούνται εναλλακτικά ως όροι και συσχετίζονται. Γενικά, ο σκοπός που εξυπηρετούν είναι η μέτρηση ποινής (penalty). Πιο συγκεκριμένα, η συνάρτηση Loss function ορίζεται για ένα δεδομένο (data point), εξαρτάται από την έξοδο του νευρωνικού δικτύου (output) και την πρόβλεψη (prediction) και μετρά μια ποινή η οποία εκφράζει την απόσταση της προβλεπόμενης τιμής και της πραγματικής εξόδου. Για παράδειγμα η συνάρτηση τετραγωνικού σφάλματος square loss στη σχέση 2.13 μετρά το τετραγωνικό σφάλμα μεταξύ της εξόδου $f(x_i|\vartheta)$ (η οποία έχει είσοδο το δεδομένο x_i και παραμέτρους/βάρη ϑ) και της αναμενόμενης τιμής y_i.

$$l(f(x_i|\vartheta), y_i) = (f(x_i|\vartheta) - y_i)^2 \tag{2.13}$$

Η συνάρτηση Cost function είναι πιο γενική από την Loss function. Μπορεί να οριστεί σε όλο ή ένα μέρος του συνόλου δεδομένων. Για παράδειγμα στη σχέση 2.14 ορίζεται η συνάρτηση μέσου τετραγωνικού σφάλματος σε N δεδομένα εισόδου.

$$MSE(\vartheta) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i|\vartheta) - y_i)^2 \tag{2.14}$$

Όπως θα δούμε στο κεφάλαιο 2.3.4.3, στόχος της εκπαίδευσης είναι να ελαττώσει όσον το δυνατό περισσότερο τη Loss (Cost) function, δηλαδή οι προβλεπόμενες και πραγματικές τιμές να είναι όσο πιο κοντά μπορούν. Σε αυτό το σημείο αξίζει να αναφέρουμε το εξής. Επιτυχία ενός νευρωνικού δικτύου είναι η γενίκευση. Δηλαδή, δοσμένου ενός συνόλου δεδομένων εκπαίδευσης (train data) το δίκτυο θα πρέπει να έχει καλή επιτυχία πρόβλεψης και στο σύνολο δοκιμής (test data). Όσο εκπαιδεύεται το νευρωνικό δίκτυο, η Loss function πάντα θα μειώνεται. Αλλά πάντα πρέπει να υπάρχει προσοχή για φαινόμενα υπερ-προσαρμογής (over-fitting). Σκοπός δεν είναι να προσεγγίσουμε τη συνάρτηση που διέπει τα δεδομένα εκπαίδευσης, αλλά μια συνάρτηση που έχει την ικανότητα να γενικεύσει από αυτά και να μπορεί να προβλέψει δεδομένα τα οποία δεν έχει 'ξανασυναντήσει'.

Μια από τις πιο ευρέως χρησιμοποιούμενες συναρτήσεις κόστους είναι η Κοινή-Εντροπία (Cross-entropy loss function). Η εντροπία είναι ένας όρος που χρησιμοποιείται στη Φυσική και τη Θεωρία Πληροφορίας. Με απλά λόγια εκδηλώνει την αταξία ενός στοιχείου ή το κατά πόσο πιθανό είναι να συμβεί κάτι. Η από κοινού εντροπία δίνει μια μέτρηση του κατά πόσο δύο διαφορετικές πηγές έχουν την ίδια 'ποσότητα πληροφορίας'. Για παράδειγμα μια ακολουθία αριθμών $1, 1, 1, 1, 1, 1, 1, ..., 1, 1$ έχει μηδενική εντροπία καθώς επικρατεί πλήρης τάξη. Μια τυχαία όμως ακολουθία αριθμών $0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1...$ έχει μεγάλη αταξία και συνεπώς μεγάλη εντροπία. Επομένως η από κοινή εντροπία αυτών των δύο ακολουθιών θα είναι μεγάλη καθώς είναι πολύ ανόμοιες. Η cross-entropy function ορίζεται στη σχέση 2.15 ως η μέση τιμή των cross-entropies N δεδομένων.

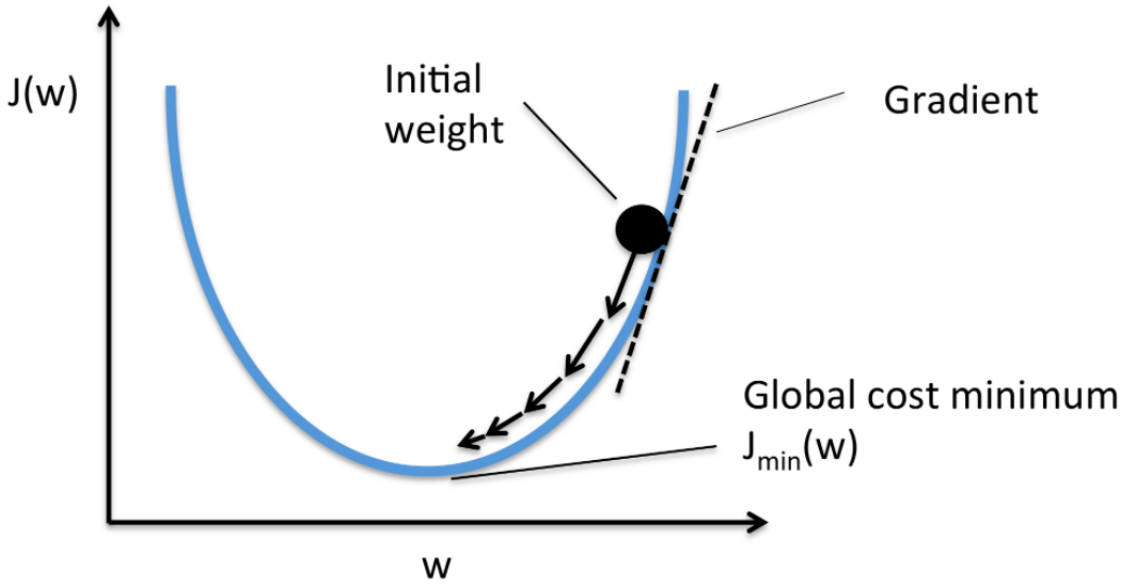
$$J(\vartheta) = \frac{1}{N}\sum_{n=1}^{N}H(p_n, q_n) = -\frac{1}{N}\sum_{n=1}^{N}f(x_n|\vartheta)logy_n + (1 - f(x_n|\vartheta))log(1 - y_n) \tag{2.15}$$

2.3.4.3 Αλγόριθμος Κατάβασης Κλίσης Gradient Descent Algorithm

Ο αλγόριθμος κατάβασης κλίσης (Gradient Descent Algorithm) σκοπεύει στην ελαχιστοποίηση της συνάρτησης κόστους $J(\boldsymbol{\vartheta})$ όπου $\boldsymbol{\vartheta}$ είναι διάνυσμα και αντιπροσωπεύει τις παραμέτρους του δικτύου. Για παράδειγμα το $\boldsymbol{\vartheta}$ στα πρόσθια τροφοδοτούμενα δίκτυα είναι τα βάρη (weights) και οι πολώσεις (biases) των νευρώνων. Η βασική ιδέα που ακολουθείται είναι το κάθε βάρος του νευρώνα να επαναϋπολογιστεί αφαιρώντας μια μικρή ποσότητα ανάλογη της επιρροής που έχει στην συνάρτηση κόστους. Αυτή η μικρή ποσότητα είναι ουσιαστικά η μερική παράγωγος της συνάρτησης κόστους ως προς την παράμετρο του εκάστοτε βάρους πολλαπλασιασμένη με μια παράμετρο α η οποία ονομάζεται ρυθμός μάθησης (learning rate). Στη σχέση 2.16 φαίνεται η νέα τιμή τη χρονική στιγμή $t + 1$ της παραμέτρου θ_j.

$$\vartheta_{j,t+1} = \vartheta_{j,t} - \alpha\frac{\partial}{\partial\vartheta_{j,t}}J(\boldsymbol{\vartheta}) \tag{2.16}$$

Η ιδέα αυτή υπήρχε ήδη στα μαθηματικά από πολύ παλιότερα (βλ. Newton method). Στην εικόνα 2.11 φαίνεται μια απεικόνιση για το πώς αλλάζει η τιμή του βάρους w προκειμένου η συνάρτηση κόστους J να φτάσει στο ελάχιστο σημείο της.

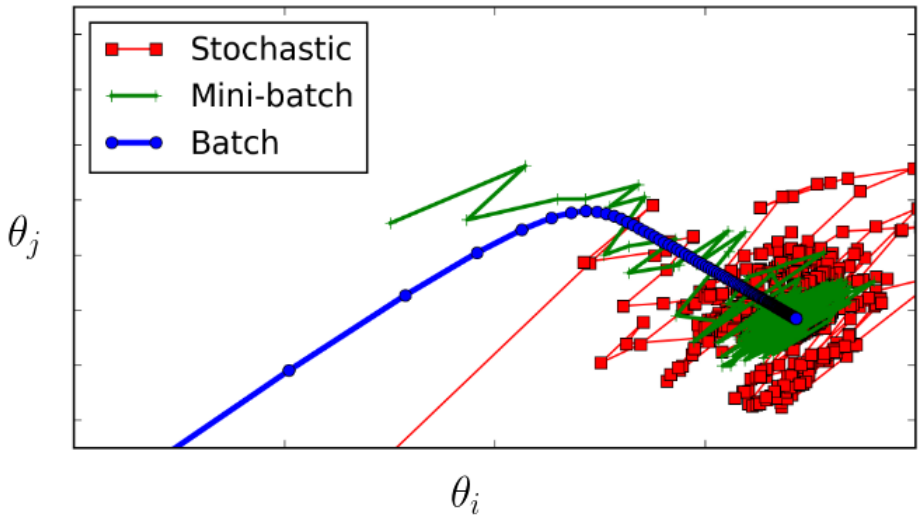


Σχήμα 2.11: Gradient Descend Visualization

Στη γενική περίπτωση ο αλγόριθμος κλίσης κατάβασης εκτελείται αφού έχει γίνει μια πρόσθια τροφοδότηση όλου του συνόλου των δεδομένων στο δίκτυο. Πολλές φορές όμως, είτε λόγο περιορισμών υλικού (π.χ μνήμη), είτε επειδή χρειάζεται το δίκτυο να εκπαιδευτεί πιο γρήγορα, δε χρησιμοποιείται όλο το σύνολο των δεδομένων αλλά χωρίζεται σε τεμάχια batches και πλέον η ανανέωση των βαρών του δικτύου γίνεται έπειτα από την πρόσθια τροφοδότηση ενός τέτοιου τεμαχίου. Δηλαδή η συνάρτηση κόστους υπολογίζεται μόνο στα δεδομένα που ανήκουν στο τεμάχιο.

Βάσει του μεγέθους του τεμαχίου χωρίζουμε τον αλγόριθμο κατάβασης κλίσης σε τρεις κατηγορίες, Batch Gradient Descent, Mini-batch Gradient Descent και Stochastic Gradient Descent. Στη βιβλιογραφία μια συνηθισμένη τιμή για το μέγεθος του mini-batch είναι 16 ή 32 δεδομένα από το σύνολο δεδομένων. Ο στοχαστικός Stochastic είναι ειδική περίπτωση του mini-batch όπου χρησιμοποιείται μόνο ένα δεδομένο για τροφοδότηση προτού ανανεωθούν τα βάρη.

Ο κάθε ένας έχει τα πλεονεκτήματα και τα μειονεκτήματά του. Το πλεονέκτημα χρήσης τεμαχίων έναντι των συνολικών δεδομένων για τροφοδότηση προτού ανανεωθούν τα βάρη είναι οι μικρότερες απαιτήσεις μνήμης του συτήματος και οι ταχύτερες εκπαιδεύσεις. Το μειονέκτημα είναι ο λιγότερο ακριβής υπολογισμός της κλίσης που γίνεται προκειμένου ο αλγόριθμος να συγκλίνει σε ελάχιστο. Στην εικόνα 2.12 φαίνεται πόσο περισσότερο ταλαντώνεται ο Stochastic Gradient Descent έναντι του Batch Gradient Descent προκειμένου δύο τυχαίες παράμετροι του δικτύου ϑ_i, ϑ_j να αποκτήσουν τις τελικές τους τιμές.



Σχήμα 2.12: Convergence of Gradient Descent with various batch sizes

2.3.4.4 Βελτιστοποίηση Optimization

Η βελτιστοποίηση (Optimization) είναι η τεχνική που ακολουθείται προκειμένου να ελαχιστοποιηθεί (minimize) ή να μεγιστοποιηθεί maximize μια 'συνάρτηση - στόχος' (objective function). Για παράδειγμα ο αλγόριθμος κατάβασης κλίσης που ορίστηκε στο κεφάλαιο 2.3.4.3 είναι ένας αλγόριθμος που ελαχιστοποιεί τη συνάρτηση κόστους του δικτύου που αναφέρθηκε στο κεφάλαιο 2.3.4.2. Οι εσωτερικές παράμετροι ενός μοντέλου παίζουν πολύ σημαντικό ρόλο στην αποτελεσματική και αποδοτική εκπαίδευσή του και στην παραγωγή ακριβή αποτελεσμάτων. Αυτός είναι ο λόγος που χρησιμοποιούμε διάφορες στρατηγικές και αλγόριθμους βελτιστοποίησης για την ενημέρωση και τον υπολογισμό των κατάλληλων και βέλτιστων τιμών των παραμέτρων ενός τέτοιου μοντέλου που επηρεάζουν τη διαδικασία εκμάθησης του μοντέλου και την έξοδό του.

Οι αλγόριθμοι βελτιστοποίησης χωρίζονται σε δυο βασικές κατηγορίες:

- *Πρώτης Τάξης (First Order Optimization Algorithms)* Αυτοί οι αλγόριθμοι ελαχιστοποιούν ή μεγιστοποιούν μια συνάρτηση απώλειας Loss function χρησιμοποιώντας τις τιμές της Κλίσης Gradient σε σχέση με τις παραμέτρους. Ο πιο ευρέως χρησιμοποιούμενος αλγόριθμος βελτιστοποίησης πρώτης τάξης όπως αναφέρθηκε στο κεφάλαιο 2.3.4.3 είναι ο αλγόριθμος κατάβασης κλίσης Gradient Descent Algorithm. Η παράγωγος πρώτης τάξης εκδηλώνει τη μείωση ή την αύξηση της συνάρτησης σε ένα συγκεκριμένο σημείο. Ουσιαστικά αντιπροσωπεύει μια γραμμή η οποία είναι εφαπτόμενη σε ένα σημείο στην επιφάνεια του σφάλματος.

- *Δεύτερης Τάξης (Second Order Optimization Algorithms)* Οι μέθοδοι δεύτερης τάξης [4] χρησιμοποιούν την παράγωγο δεύτερης τάξης που ονομάζεται επίσης Hessian για να ελαχιστοποιήσουν ή μεγιστοποιήσουν τη συνάρτηση απώλειας Loss function. Ο Hessian πίνακας περιέχει τις μερικές παραγώγους δεύτερης τάξης Second Order Par-

tial Derivatives. Το μειονέκτημα αυτής της μεθόδου είναι ότι η δεύτερη παράγωγος είναι υπολογιστικά δαπανηρή και γι΄ αυτό το λόγο δε χρησιμοποιείται πολύ. Η δεύτερη παράγωγος εκφράζει την αύξηση ή τη μείωση της πρώτης παραγώγου, πράγμα που υποδηλώνει οτι λαμβάνει υπόψιν την καμπυλότητα της συνάρτησης. Το πλεονέκτημα αυτής της μεθόδου είναι ότι είναι αποτελεσματικότερη σε κάθε βηματικό υπολογισμό.

Το πρόβλημα με τον αλγόριθμο κατάβασης κλίσης είναι ότι δημιουργούνται ταλαντώσεις μεγάλης διακύμανσης καθιστώντας δύσκολη την επίτευξη σύγκλισης. Τη λύση σε αυτό δίνει μια τεχνική που ονομάζεται 'Ορμή' Momentum η οποία επιταχύνει τη διαδικασία πλοηγώντας τις παραμέτρους κατά μήκος της σχετικής κατεύθυνσης και ελαττώνοντας τις ταλαντώσεις σε άσχετες κατευθύνσεις. Η ιδέα είναι ότι πρώτα γίνεται ένα μεγάλο άλμα με βάση την προηγούμενη ορμή, στη συνέχεια υπολογίζεται η κλίση και έπειτα γίνεται η ενημέρωση των παραμέτρων. Είναι μια μέθοδος look-ahead, δηλαδή, αντί να υπολογίζεται η κλίση συναρτήσει των παραμέτρων, υπολογίζεται συναρτήσει μιας προσέγγισης της μέλλουσας τιμής τους λαμβάνοντας υπόψιν την ορμή. Αυτή η ιδέα βοηθά τον αλγόριθμο να μη χάσει κάποιο ελάχιστο λόγο μεγάλης αλλαγής βαρών (overshoot) και να ανταποκρίνεται καλύτερα στις αλλαγές.

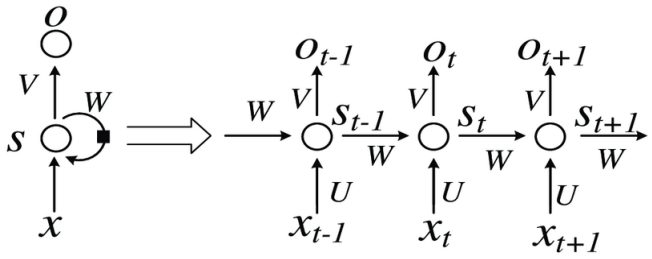
Υπάρχουν τρεις βασικοί αλγόριθμοι οι οποίοι χρησιμοποιούν την παραπάνω ιδέα και αποτελούν παραλλαγές του αλγορίθμου Gradient Descent, ο Adagrad [37], ο AdaDelta [70] και ο Adam (Adaptive Moment Estimation) [29]. Ο Adagrad χρησιμοποιεί διαφορετικό ρυθμό εκμάθησης (learning rate) για κάθε παράμετρο ϑ σε κάθε χρονικό βήμα με βάση τις παλιότερες κλίσεις που υπολογίστηκαν για αυτήν την παράμετρο. Το πλεονέτκημα είναι ότι δε χρειάζεται να οριστεί εξαρχής ένας απόλυτος ρυθμός εκμάθησης καθώς έχει την ικανότητα να τον προσαρμόζει κατά τη διάρκεια της εκπαίδευσης. Ένα μειονέκτημα είναι ότι ο ρυθμός εκμάθησης συνεχώς μειώνεται κάνοντας το μοντέλο δύσκαμπτο σε καινούρια δεδομένα. Ο AdaDelta είναι μια επέκταση του Adagrad. Αφαιρεί το πρόβλημα της συνεχούς μείωσης του ρυθμού εκμάθησης (vanishing learning rate) χρησιμοποιώντας μια πιο σύνθετη τεχνική για την εξιοποίηση των προηγούμενων κλίσεων. Τέλος, ο Adam (Adaptive Moment Estimation) χρησιμοποιεί και την πρώτη και την δεύτερη ροπή των κλίσεων (First and Second Moment of Gradient), όπου η πρώτη είναι ο μέσος όρος (Mean) και η δεύτερη είναι η διακύμανση (Variance). Ο Adam λειτουργεί καλά στην πράξη και συγκρινόμενος με τους άλλους αλγόριθμους προσαρμοζόμενου ρυθμού μάθησης είναι προτιμότερος καθώς συγκλίνει πολύ γρήγορα. Επίσης, διορθώνει τα προβλήματα που υπάρχουν στις άλλες τεχνικές βελτιστοποίησης, όπως η αργή σύγκλιση ή η μεγάλη διακύμανση στις ενημερώσεις παραμέτρων που οδηγεί σε ταλαντώσεις.

2.3.5 Επαναλαμβανόμενα Νευρωνικά Δίκτυα Recurrent Neural Networks (RNN)

Στο κεφάλαιο 2.3.3 παρουσιάσαμε τα πολυ-επίπεδα νευρωνικά δίκτυα. Αυτά τα δίκτυα είναι κατάλληλα για ταξινόμηση δεδομένων με στατικό χαρακτήρα. Τέτοια δεδομένα είναι οι εικόνες ή η εγγραφή μιας βάσης δεδομένων για παράδειγμα τα στοιχεία ενός πελάτη μιας εταιρίας. Πολλές φορές όμως, τα δεδομένα χαρακτηρίζονται από χρονική/ακολουθιακή μεταβολή, όπως ένα βίντεο, οι τιμές των μετοχών του χρηματιστηρίου ή στην παρούσα διπλωματική, η ακολουθία

των λέξεων ενός κειμένου. Η ανάγκη επεξεργασίας τέτοιου είδους δεδομένων οδήγησε στην ανάπτυξη νευρωνικών δικτύων τα οποία λαμβάνουν υπόψιν τους αυτή την ακολουθιακή φύση των δεδομένων.

Τα επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks) [28] είναι δίκτυα που ενσωματώνουν τον ακολουθιακό χαρακτήρα στην ίδια τους τη δομή. Το βασικό τους στοιχείο είναι η μνήμη, η οποία διαμορφώνεται βάσει της ακολουθιακής ιστορίας των δεδομένων που τροφοδοτούνται ακολουθιακά στο δίκτυο. Η εσωτερική μνήμη που διαθέτουν επιτρέπει να επεξεργάζονται κάθε νέα είσοδο, η οποία αποτελεί μέρος της ακολουθίας των δεδομένων, λαμβάνοντας υπόψιν και όσα δεδομένα προηγήθηκαν προκειμένου να προβλέψουν την επόμενη κατάσταση. Στην εικόνα 2.13 φαίνεται το 'ξεδίπλωμα' ενός RNN σε αντίστοιχο ακολουθιακό μοντέλο.



Σχήμα 2.13: RNN Unfold Equivalent

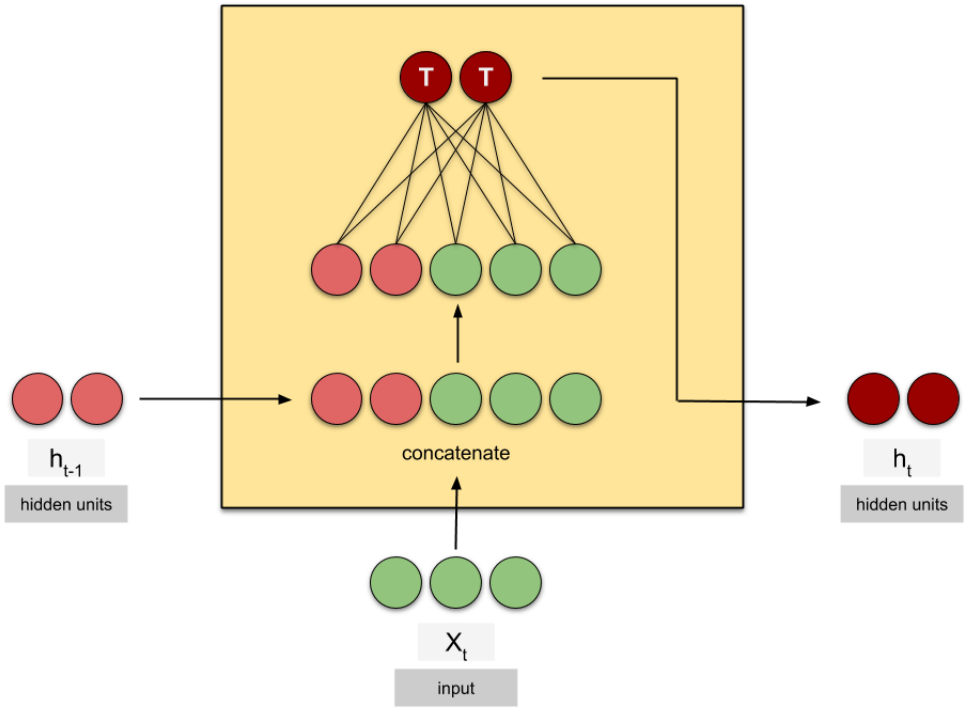
2.3.5.1 Πρότυπο RNN (Vanilla RNN)

Το πρότυπο RNN είναι η πιο απλή μορφή ενός επαναλαμβανόμενου νευρωνικού δικτύου. Στην εικόνα 2.14 φαίνεται η δομή του Vanilla RNN.

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.17}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{2.18}$$

Στις σχέσεις 2.17 και 2.18 φαίνεται ο υπολογισμός της επόμενης κρυφής κατάστασης (hidden state) h_t και η έξοδος του κυττάρου y_t αντίστοιχα. Οι μεταβλητές W, U είναι πίνακες παραμέτρων και οι διαστάσεις τους εξαρτώνται από τη διάσταση της εισόδου x και της κρυφής κατάστασης h. Οι συναρτήσεις σ_h, σ_y είναι συναρτήσεις ενεργοποίησης. Είναι σημαντικό να αναφέρουμε ότι το μέγεθος του διανύσματος h συναντάται με διαφορετικές ονομασίες σε διάφορες τεχνικές υλοποιήσεις όπως state size, hidden size, number of units αλλά ουσιαστικά εκφράζει το μέγεθος της εσωτερικής μνήμης του κυττάρου. Όσο πιο μεγάλη είναι η διάστασή του τόσο πιο σύνθετο είναι το μοντέλο.



Σχήμα 2.14: Vanilla RNN Cell

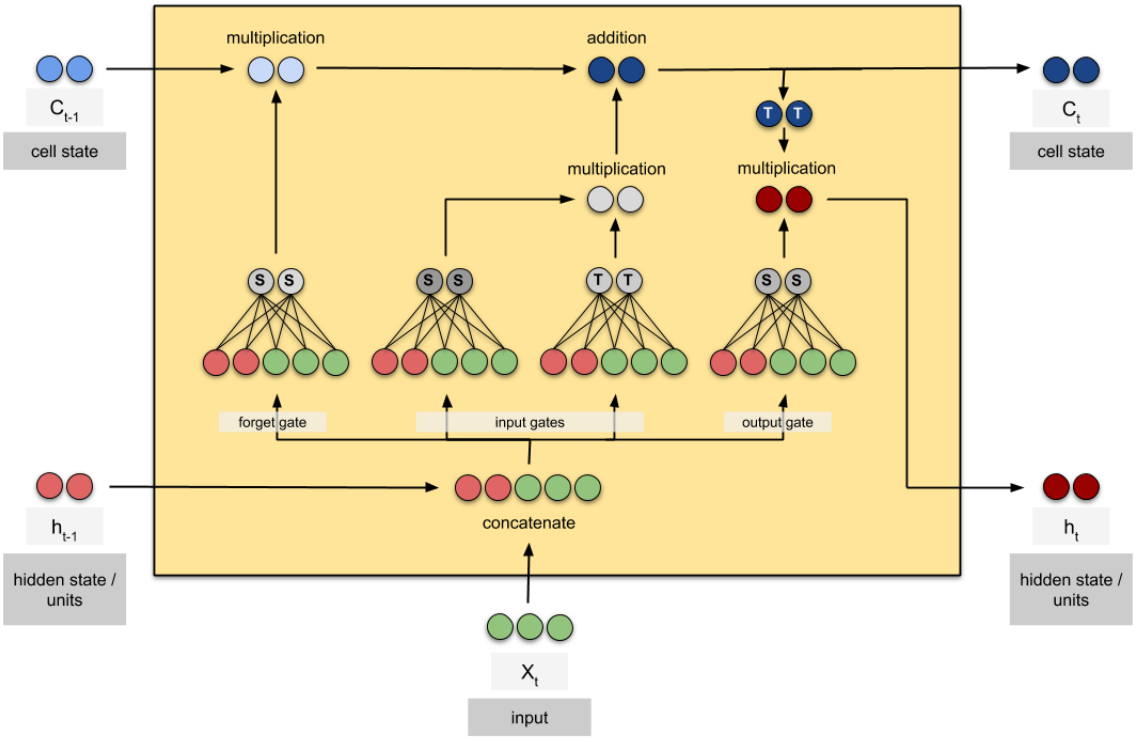
2.3.5.2 Κύτταρο Μακράς Βραχυπρόθεσμης Μνήμης Long Short-Term Memory Cell (LSTM)

Τα LSTM [28], [56] είναι πολύ πιο σύνθετα μοντέλα από τα απλά RNN. Βασική διαφορά είναι ότι τα LSTM πέρα από την κρυφή κατάσταση hidden state έχουν επίσης και την κατάσταση κυττάρου cell state. Επιπλέον, διαθέτουν μια πύλη λήθης (forget gate) η οποία έχει τη δυνατότητα να αποκόβει κάποιες τιμές δίνοντας καθ' αυτόν τον τρόπο λύση στο πρόβλημα της εξαφανιζόμενης κλίσης (vanishing gradient) το οποίο εμφανίζεται στα απλά RNN. Στην εικόνα 2.15 φαίνεται η δομή ενός κυττάρου LSTM.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2.19}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2.20}$$
$$i'_t = \sigma_{i'}(W_{i'} x_t + U_{i'} h_{t-1} + b_{i'}) \tag{2.21}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.22}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ i'_t \tag{2.23}$$
$$h_t = o_t \circ \sigma_h(c_t) \tag{2.24}$$

Μεταβλητές των σχέσεων [2.19-2.24]

- $x_t \in \mathbb{R}^d$: Το διάνυσμα εισόδου στο κύτταρο LSTM.



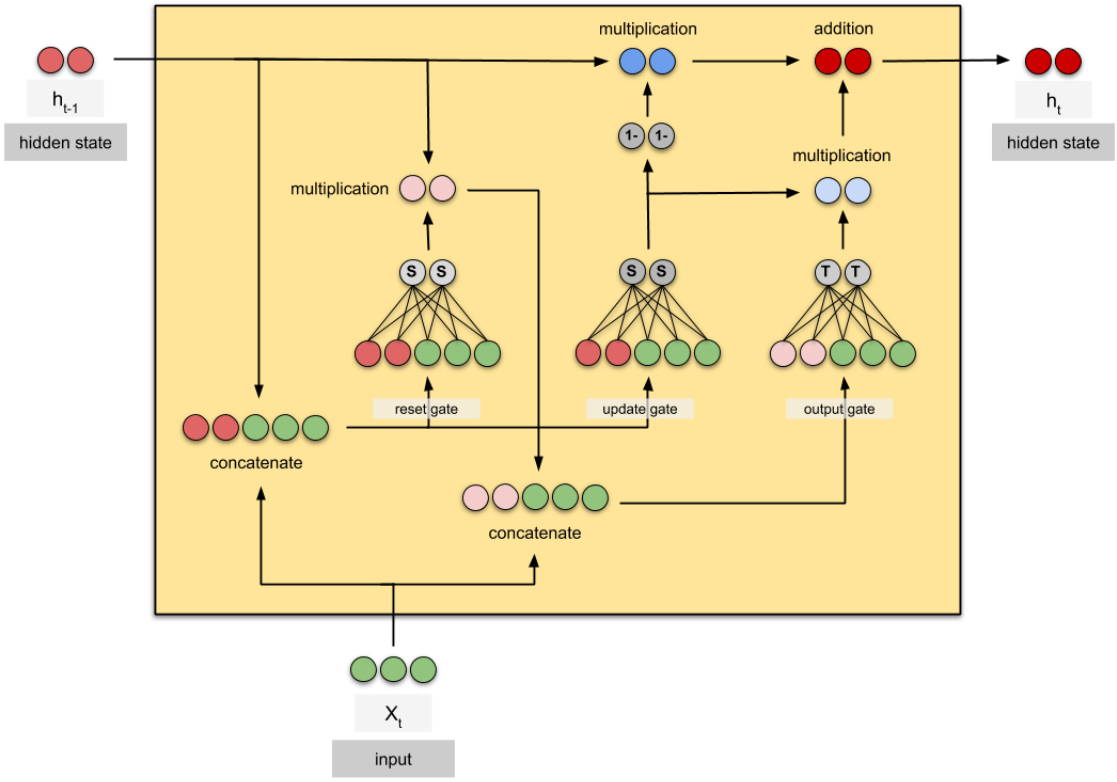
Σχήμα 2.15: LSTM Cell

- $f_t \in \mathbb{R}^h$: Το διάνυσμα ενεργοποίησης της πύλης λήθης (forget gate).
- $i_t \in \mathbb{R}^h$: Το διάνυσμα ενεργοποίησης της πύλης εισόδου.
- $i'_t \in \mathbb{R}^h$: Αυτή η μεταβλητή είναι ένα βοηθητικό διάνυσμα ενεργοποίησης της πύλης εισόδου. Διαφέρει από το i_t ως προς τη συνάρτηση ενεργοποίησης που χρησιμοποιείται.
- $o_t \in \mathbb{R}^h$: Το διάνυσμα ενεργοποίησης της πύλης εξόδου
- $h_t \in \mathbb{R}^h$: Η κρυφή κατάσταση (hidden state) του LSTM.
- $c_t \in \mathbb{R}^h$: Η κατάσταση κυττάρου (cell state).
- $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}, b \in \mathbb{R}^h$: Οι πίνακες βαρών και πόλωσης.
- σ_g: Η σιγμοειδής (sigmoid) συνάρτηση ενεργοποίησης.
- $\sigma_{i'}, \sigma_h$: Η υπερβολική εφαπτομένη (hyperbolic tangent) συνάρτηση ενεργοποίησης.
- *Τελεστής* $\circ$: Εφαρμόζει πολλαπλασιασμό ανά στοιχείο πίνακα.

2.3.5.3 Gated Recurrent Unit (GRU)

Τα GRU [11], [28], [15] είναι παρόμοια με τα LSTMs με τη μόνη διαφορά ότι δεν έχουν πύλη εξόδου. Αυτό τα κάνει πολύ πιο αποδοτικά καθώς χρησιμοποιούνται λιγότεροι παράμετροι

εσωτερικά του κυττάρου. Τα GRUs συστήθηκαν το 2014 από τον Kyunghyun Cho [11] και παρουσιάζουν πολύ καλή επίδοση σε μικρού όγκου δεδομένα. Στην εικόνα 2.16 φαίνεται η δομή ενός κυττάρου GRU.



Σχήμα 2.16: GRU Cell

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{2.25}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2.26}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1} + b_h)) \tag{2.27}$$

Μεταβλητές των σχέσεων [2.25-2.27]

- x_t: Το διάνυσμα εισόδου στο κύτταρο LSTM.
- h_t: Το διάνυσμα εξόδου ή η κρυφή κατάσταση (hidden).
- z_t: Το διάνυσμα της πύλης ανανέωσης (update gate vector).
- r_t: Το διάνυσμα της πύλης επαναφοράς (reset gate vector).
- W, U, b: Οι πίνακες βαρών και πόλωσης.
- σ_g: Η σιγμοειδής (sigmoid) συνάρτηση ενεργοποίησης.

- σ_h: Η υπερβολική εφαπτομένη (hyperbolic tangent) συνάρτηση ενεργοποίησης.
- *Τελεστής* $\circ$: Εφαρμόζει πολλαπλασιασμό ανά στοιχείο πίνακα.

2.4 Μετρικές Αξιολόγησης

Οι μετρικές αξιολόγησης μετρούν κάποιο μέγεθος του εκπαιδευμένου μοντέλου ως προς ένα συγκεκριμένο χαρακτηριστικό, όπως η ακρίβεια, η προκατάληψη κλπ. Είναι σημαντικό να ορίζονται μετρικές αξιολόγησης καθώς βάσει αυτών είναι εφικτή η σύγκριση μεταξύ διαφόρων μοντέλων που αντιμετωπίζουν το ίδιο πρόβλημα. Πρέπει να αναφερθεί επίσης πως δεν υπάρχει αυστηρή τοποθέτηση ως προς την καταλληλότητα επιλογής συγκεκριμένης μετρικής καθώς η αξιολόγηση του μοντέλου σε κάθε πρόβλημα εξαρτάται από τη φύση του προβήματος. Η πιο σημαντική και μετρική αξιολόγησης ενός μοντέλου είναι η ακρίβεια (Accuracy). Το μέτρο της εκφράζει το ποσοστό επιτυχίας του μοντέλου στην ταξινόμηση των δειγμάτων (του συνόλου δεδομένων ελέγχου (test set» στις σωστές κατηγορίες. Η ακρίβεια περιγράφεται από τη σχέση 2.28, δηλαδή είναι ο λόγος των σωστά ταξινομημένων δειγμάτων ως προς όλα τα δείγματα.

$$Accuracy = \frac{\text{Number of samples recognized correctly}}{\text{Number of Total samples}} \tag{2.28}$$

Η ακρίβεια δίνει μια άμεση και απλή αξιολόγηση της επιτυχίας του μοντέλου. Ωστόσο, η ακρίβεια (accuracy) δεν εκφράζει πληροφορίες σχετικά με την ταξινόμηση των δεδομένων ανά κατηγορία και σε ορισμένες περιπτώσεις δεν αποτελεί το κατάλληλο μέτρο για την αξιολόγηση ενός μοντέλου. Για παράδειμα, ένα μοντέλο που προβλέπει την ύπαρξη ή μη ασθένειας σε έναν άνθρωπο, μπορεί να έχει μεγάλη ακρίβεια η οποία οφείλεται στο γεγονός ότι οι περισσότεροι άνθρωποι είναι υγιείς και επομένως το μοντέλο είναι περισσότερο προκατειλλημένο στην πρόβλεψη μη ασθένειας. Για το συγκεκριμένο πρόβλημα, όμως, απασχολεί περισσότερο η πρόβλεψη ασθένειας σε έναν ασθενή παρά η πρόβλεψη μη ασθένειας σε έναν υγιή καθώς στην πρώτη περίπτωση η αστοχία του μοντέλου μπορεί να αποβεί μοιραία για έναν άνθρωπο ενώ στη δεύτερη περίπτωση όχι. Επομένως, ορίζονται δύο καινούριες μετρικές, precision (ακρίβεια) και recall (ανάκληση). Η διαφορά του accuracy με το precision είναι ότι το πρώτο αξιολογεί την απόσταση των μετρήσεων από μια επιθυμητή τιμή ενώ το δεύτερο αξιολογεί την απόσταση των μετρήσεων ανά μεταξύ τους. Πριν οριστούν αυτές οι δυο μετρικές αξιολόγησης, ορίζονται πρώτα κάποιες κλάσεις βάσει της προβλεπόμενης και της πραγματικής κατηγορίας που ανήκουν τα δείγματα.

- **True Positive (TP):** Το σύνολο των δειγμάτων για τα οποία η πρόβλεψη είναι σωστή και η προβλεπόμενη κατηγορία είναι η positive
- **True Negative (TN):** Το σύνολο των δειγμάτων για τα οποία η πρόβλεψη είναι σωστή και η προβλεπόμενη κατηγορία είναι η negative
- **False Positive (FP):** Το σύνολο των δειγμάτων για τα οποία η πρόβλεψη είναι λανθασμένη και η προβλεπόμενη κατηγορία είναι η positive (ενώ η πραγματική κατηγορία είναι η negative)

- **False Negative (FN):** Το σύνολο των δειγμάτων για τα οποία η πρόβλεψη είναι λανθασμένη και η προβλεπόμενη κατηγορία είναι η negative (ενώ η πραγματική κατηγορία είναι η positive)

Βάσει αυτών των κλάσεων ορίζονται οι μετρικές Precision και Recall στις σχέσεις 2.29 και 2.30 [20] αντίστοιχα.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.29}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.30}$$

Επανέρχοντας στο προηγούμενο παράδειγμα, αν θέσουμε ως positive κατηγορία τους ασθενείς, τότε η μετρική που απασχολεί περισσότερο είναι η ανάκληση (recall) καθώς είναι επιθυμητό να μειωθεί ο όρος FN δηλαδή δείγματα που λανθασμένα προβλέπονται ως υγιή άτομα (ενώ είναι ασθενείς). Βέβαια δε θέλουμε και κάθε υγιής άνθρωπος να προβλέπεται ως ασθενής (δηλαδή χαμηλό precision) καθώς κάτι τέτοιο θα οδηγούσε σε σύγχυση. Επομένως είναι σημαντικό να υπάρχει μια ισορροπία σε αυτές τις δυο μετρικές αξιολόγησης γιατί συνήθως αύξηση της μιας οδηγεί σε μείωση της άλλης. Γι΄ αυτό το λόγο ορίζεται μια επιπλέον μετρική που ονομάζεται F_1 Score και είναι ο αρμονικός μέσος του Precision και του Recall όπως φαίνεται στη σχέση 2.31.

$$\text{F}_1\text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.31}$$

Ένα σημαντικό κομμάτι της διπλωματικής, όπως θα αναφερθεί σε επόμενο κεφάλαιο, είναι η μετατροπή των λέξεων κειμένου σε διανύσματα από αριθμητικές τιμές ώστε να είναι επεξεργάσιμα από ένα υπολογιστικό σύστημα. Τα διανύσματα αυτά είναι σημεία στο χώρο $\mathbb{R}^n$ και πάνω σε αυτά ορίζονται μετρικές όπως η ευκλίδεια απόσταση και η Ομοιότητα Συνημιτόνου (Cosine Similarity) [23]. Η Ομοιότητα Συνημιτόνου (Cosine Similarity) είναι μια σημαντική μετρική η οποία εκφράζει τη μεταξύ γωνία δύο διανυσμάτων. Δύο διανύσματα που είναι παράλληλα εκφράζουν μεγάλη συσχέτιση μεταξύ τους επομένως το συνημίτο της μεταξύ τους γωνίας (0^o) είναι 1. Αντίστοιχα, δύο διανύσματα που είναι κάθετα εκφράζουν πλήρη ουδετερότητα και το συνημίτονο της μεταξύ τους γωνίας (90^o) είναι 0. Η μετρική αυτή για δύο διανύσματα $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^n$ διατυπώνεται στη σχέση 2.32.

$$\text{Cosine Similarity} = cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.32}$$

Κεφάλαιο 3

Επεξεργασία Κειμένου

Τα νευρωνικά μοντέλα βασίζονται σε υπολογιστικές πράξεις μεταξύ αριθμών. Όσο σύνθετο κι αν είναι ένα μοντέλο, στον πυρήνα του εκτελούνται μαθηματικές πράξεις πρόσθεσης και πολλαπλασιασμού. Πολλά δεδομένα, όπως οι τιμές των μετοχών στο χρηματηστήριο ή ένα σύνολο από εικόνες αναπαρίστανται ήδη με τη μορφή αριθμού ή συνόλου από αριθμούς, επομένως η τροφοδότηση ενός τέτοιου συνόλου δεδομένων στο δίκτυο είναι άμεσα εφικτή ή έστω έπειτα από μια προεπεξεργασία. Τα μοντέλα όμως που επεξεργάζονται κείμενο δεν έχουν αυτή την πολυτέλεια καθώς οι λέξεις που αποτελούν το κείμενο ή οι χαρακτήρες που αποτελούν τις λέξεις δεν εμφανίζουν κάποια άμεση αριθμητική αντιστοιχία. Στα επόμενα υποκεφάλαια θα παρουσιαστούν γνωστές μεθοδολογίες που αντιμετωπίζουν αυτό το πρόβλημα και τα αντίστοιχα πλεονεκτήματα και μειονεκήματά τους.

3.1 Μέθοδος Bag of Words (BoW)

Η τεχνική Bag of Words [72] είναι η πιο απλή τεχνική που μετατρέπει κείμενο σε διάνυσμα. Δοσμένου ενός λεξιλογίου $V = [w_1, w_2, ..., w_v]$ με πλήθος λέξεων v ($|V| = v$) και ενός κειμένου D με πλήθος λέξεων d ($|D| = d \leq v$), ο αλγόριθμος Bag Of Words κατασκευάζει ένα διάνυσμα $\boldsymbol{s} = [s_1, s_2, ..., s_v] \in \mathbb{R}^v$ όπου το $s_i = occurrences(w_i, D)$ όπου $w_i \in V$ και $occurrences(w, D)$ είναι ο αριθμός που εμφανίζεται η λέξη w στο κείμενο D. Για παράδειγμα, έστω ότι το λεξιλόγιο είναι $V = [duck, ducks, little, bowling, quack, go]$ με $|V| = 6$ και το κείμενο είναι η πρόταση 'Little ducks go quack quack quack.' Τότε το παραγώμενο διάνυσμα θα είναι το $\boldsymbol{s} = [0, 1, 1, 0, 3, 1]$ καθώς οι λέξεις 'duck' και 'bowling' δεν εμφανίζονται καμία φορά στην πρόταση, οι λέξεις 'little','ducks' και 'go' εμφανίζονται από μία φορά και η λέξη 'quack' εμφανίζεται τρεις.

Παρά την τεράστια απλότητα του αλγορίθμου, αφού η μόνη διατήρηση πληροφορίας από το αρχικό κείμενο είναι το πλήθος εμφάνισης των λέξεων και η δυσκολία κλιμάκωσης λόγο των περισσότερο πλούσιων λεξιλογίων, αυτή η τεχνική χρησιμοποιούταν σχεδόν αποκλειστικά και με μεγάλη επιτυχία για ένα τεράστιο φάσμα εργασιών στον τομέα επεξεργασίας φυσικής γλώσσας εδώ και δεκαετίες. Ακόμη και με τη σημαντική πρόοδο στη διανυσματική αναπαράσταση κειμένου τα τελευταία χρόνια, μικρές παραλλαγές αυτής της μεθόδου εξακολουθούν να χρησι-

μοποιούνται και σήμερα. Αυτή η μέθοδος αποτελεί τη βασική γραμμή (baseline) σύγκρισης με άλλους πιο σύνθετους αλγορίθμους. Το προφανές μειονέκτημα αυτής της μεθόδου είναι ότι δε λαμβάνει καθόλου υπόψιν τη συντακτική δομή μιας πρότασης. Για παράδειγμα, οι προτάσεις 'Make love not war' και 'Make war not love' αναπαρίστανται με το ίδιο διάνυσμα παρ΄ ότι νοηματικά είναι αντίθετες.

3.2 Μέθοδος Bag of n-grams

Αυτή η μέθοδος [33] είναι μια παραλλαγή της μεθόδου BoW η οποία αξιοποιεί σε έναν βαθμό τη σειρά εμφάνισης των λέξεων. Εκτός από τις λέξεις, δημιουργούνται επιπλέον συνδυασμοί λέξεων που βρίσκονται σε σειρά. Για παράδειγμα η πρόταση 'This is something else' έχει 2-grams: "This is", "is something", "something else" και 3-grams: "This is something", "is something else". Το πρόβλημα που εμφανίζεται σε αυτή τη μέθοδο είναι ότι το μέγεθος του λεξιλογίου αυξάνεται με ρυθμό $O(v^n)$ όπου v το πλήθος των αρχικών λέξεων και n το μήκος των ακολουθιών των λέξεων. Για παράδειγμα, εάν το λεξιλόγιο έχει 100 λέξεις και χρησιμοποιηθούν όλα τα 2-grams και 3-grams τότε το νέο μέγεθος του λεξιλογίου θα είναι $100 + 100^2 + 100^3 \approx 1000000$ κάτι το οποίο δεν είναι διαχειρίσιμο.

3.3 Η τεχνική TF-IDF

Η τεχνική TF-IDF [50] χωρίζεται σε δυο μέρη, τη συχνότητα όρου (Term Frequency (TF) και την αντίστροφη συχνότητα εγγράφων (Inverse Document Frequency (IDF). Το πρώτο μέρος ορίζεται ως οι φορές που εμφανίζεται μια λέξη σε μια δοσμένη πρόταση. Το δεύτερο μέρος ορίζεται ως ο φυσικός λογάριθμος του λόγου των συνολικών εγγράφων ως προς το πλήθος των εγγράφων όπου εμφανίζεται η λέξη. Το σύνολο των εγγράφων προέρχεται από μια μεγάλη συλλογή (corpus) η οποία περιέχει δισεκατομύρια λέξεις.

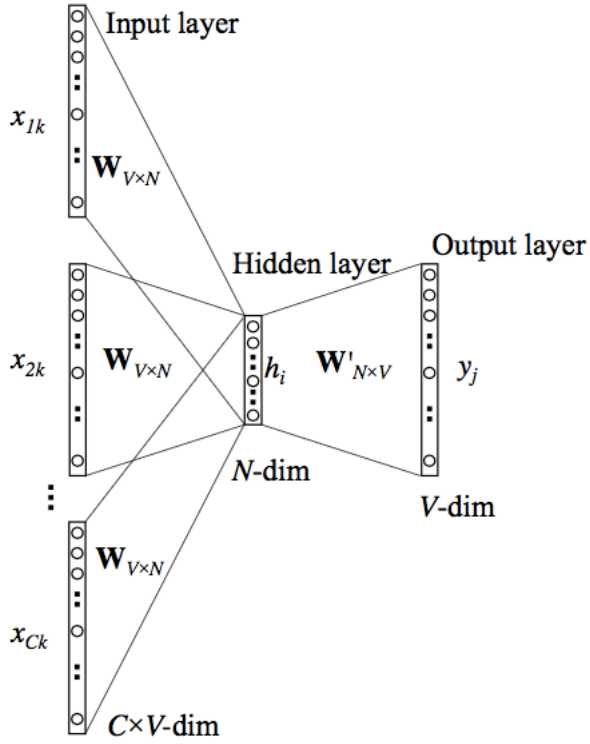
$$TF{-}IDF(w_i) = TF{*}IDF = TermFreq(w_i){*}log(\frac{\text{number of total documents}}{\text{number of documents word w}_i appears}) \tag{3.1}$$

Η σχέση 3.1 μπορεί να τροποποιηθεί επιτρέποντας σταθερούς όρους για κανονικοποίση ή μετατόπιση.Το πλεονέκτημα της τεχνικής TF-IDF έναντι της BoW είναι ότι εξετάζει τη σημασία των λέξεων απέναντι σε ένα μεγάλο πλήθος κειμένων. Για παράδειγμα, λέξεις όπως "is", "the" εμφανίζονται παντού, που σημαίνει ότι δεν εμπεριέχουν ιδιαίτερη πληροφορία, επομένως ο όρος IDF μηδενίζει καθώς ο λόγος εσωτερικά του λογαρίθμου προσεγγίζει τη μονάδα. Αντιθέτως, πιο σπάνιες λέξεις όπως η λέξη "Ubiquitous" αποκτάνε μεγαλύτερο βάρος καθώς καθορίζουν πιο έντονα το περιεχόμενο μιας πρότασης. Ωστόσο, η τεχνική αυτή συναντάει τις ίδιες αδυναμίες με την Bow αφού δε λαμβάνει υπόψιν τη σειρά εμφάνισης των λέξεων κι επίσης χρειάζονται πολύ μεγάλου όγκου δεδομένα προκειμένου να εφαρμοστεί επιτυχώς.

3.4 Μετατροπή Λέξης σε Διάνυσμα (Word2Vec)

Στην τεχνική Word2Vec [41], [19] ανήκουν δυο βασικά μοντέλα.

- **CBOW (Continuous Bag of Words)**: Το νευρωνικό δίκτυο δέχεται ως είσοδο γειτονικές λέξεις μιας λέξης-στόχου σε μια πρόταση και προβλέπει τη λέξη-στόχο (target word) η οποία βρίσκεται ανάμεσα.
- **Skip-grams**: Το νευρωνικό δίκτυο δέχεται ως είσοδο μια λέξη και προσπαθεί να προβλέψει τις γειτονικές λέξεις. Ουσιαστικά, αποτελεί αντίστροφη αρχιτεκτονική του μοντέλου CBOW.



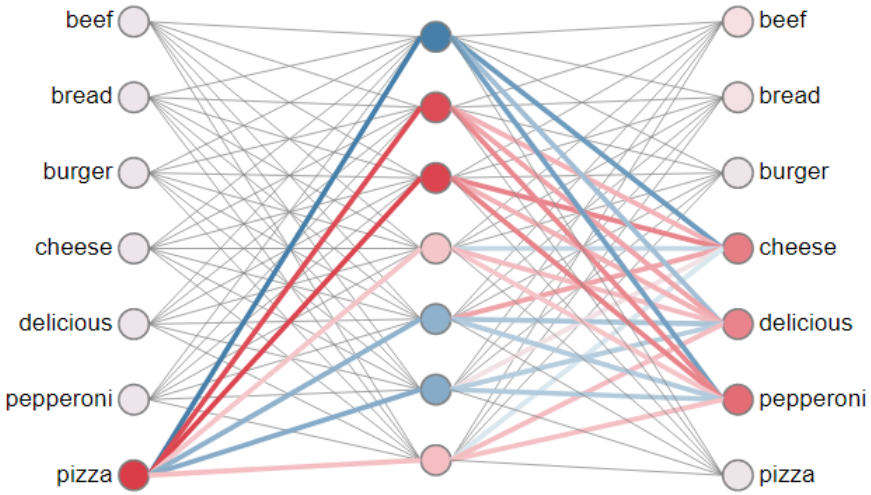
Σχήμα 3.1: Δομή Δικτύου CBOW

Στην εικόνα 3.1 φαίνεται η δομή ενός δικτύου CBOW. Το μοντέλο αυτό [35] λαμβάνει ως είσοδο C λέξεις που ανήκουν στα συμφραζόμενα (context) [16] της λέξης που έχει στόχο να προβλέψει. Οι λέξεις είναι κωδικοποιημένες με μορφή One-hot encoding, δηλαδή το διάνυσμα $\boldsymbol{w}$ που αντιπροσωπεύει μια λέξη έχει άσσο στον αντίστοιχο δείκτη που βρίσκεται η λέξη στο λεξιλόγιο και μηδέν αλλού ($\boldsymbol{w} \in \mathbb{R}^v$).

$$onehot(w_i) = \begin{cases} 1, & if w_i = v_i \in V \\ 0, & otherwise \end{cases} \tag{3.2}$$

Η ιδέα είναι ότι λέξεις που μοιράζονται τα ίδια συμφραζόμενα [49], θα έχουν κατά πάσα πιθανότητα παρόμοια σημασία. Στην εικόνα 3.2 φαίνεται ένα παράδειγμα δικτύου CBOW

όπου δείχνει μετά την εκπαίδευση τη σχέση που έχει η λέξη pizza με άλλες λέξεις βάσει των συμφραζομένων που συναντήθηκαν κατά την εκπαίδευση.



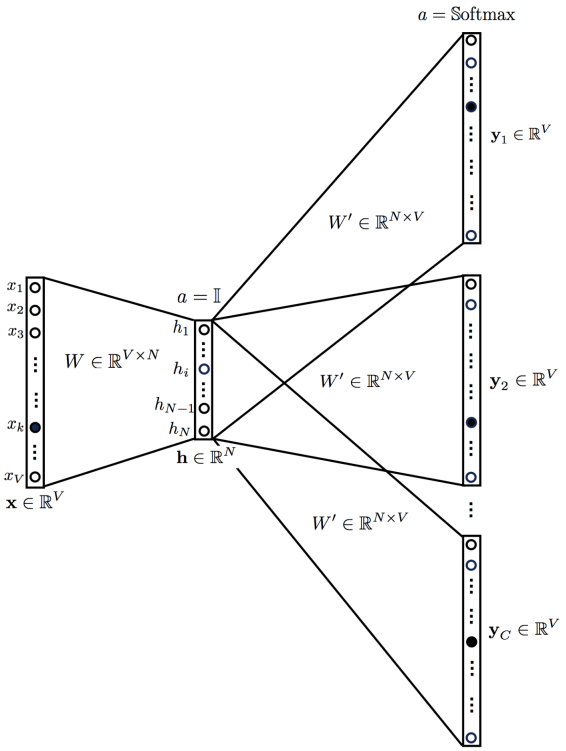
Σχήμα 3.2: Παράδειγμα CBOW με $Context = 1word$

Στην εικόνα 3.3 φαίνεται η δομή ενός μοντέλου Skip-gram. Το μοντέλο [40] δέχεται ως είσοδο μια λέξη και παράγει ένα σύνολο από C κατανομές V πιθανοτήτων.

3.5 GloVe (Global Vectors)

Η τεχνική word2vec που περιγράφηκε στο κεφάλαιο 3.4 είναι σίγουρα πιο αποδοτική από τις προηγούμενες τεχνικές BoW, TF-IDF αλλά κι αυτή παρουσιάζει μειονεκτήματα. Συγκεκριμένα, είναι σχεδιασμένη κατά τέτοιο τρόπο ώστε ο υπολογισμός των μεταξύ εμφανίσεων λέξεων να συμβαίνει τοπικά (local) [26]. Για παράδειγμα, στην πρόταση 'The cat ate the mouse.' παρατηρούμε ότι η λέξη 'the' θα εμφανιστεί στα συμφραζόμενα της λέξης 'cat' και 'mouse' όμως δεν προσδίδει κάποιο ιδιαίτερο εννοιολογικό περιεχόμενο καθώς εμφανίζεται σε όλα τα ουσιαστικά της αγγλικής γλώσσας. Ανήκει δηλαδή σε μια κατηγορία λέξεων που ονομάζεται stop words και στην οποία ανήκουν όσες λέξεις γενικά δεν προσθέτουν εννοιολογική πληροφορία όπως τα άρθρα 'a', 'an' και άλλες λέξεις όπως οι 'and', 'but' και 'or'.

Το GloVe [48], [38], [6] αντιμετωπίζει το παραπάνω πρόβλημα καθώς είναι σχεδιασμένο κατά τρόπο τέτοιο να υπολογίζει την συνεμφάνιση (co-occurrence) λέξεων μέσα σε ένα μεγάλο πλήθος κειμένων (corpus) σε καθολικό βαθμό (global). Το μοντέλο GloVe εκπαιδεύεται στις μη μηδενικές καταχωρίσεις ενός πίνακα συνεμφάνισης λέξεων, ο οποίος καταγράφει τη συχνότητα που οι λέξεις συνυπάρχουν μεταξύ τους σε ένα κείμενο. Η συμπλήρωση ενός τέτοιου πίνακα απαιτεί ένα μόνο πέρασμα του κειμένου για τη συλλογή των στατιστικών στοιχείων. Βέβαια ένα τέτοιο πέρασμα μπορεί να είναι υπολογιστικά δαπανηρό και χρονοβόρο, αλλά είναι ένα κόστος που συμβαίνει μια φορά. Οι επόμενες επαναλήψεις κατά τη διάρκεια της εκπαίδευσης είναι πολύ πιο γρήγορες καθώς ο αριθμός των μη μηδενικών στοιχείων του πίνακα είναι



Σχήμα 3.3: Δομή Δικτύου Skip-gram

συνήθως πολύ μικρότερος από τον συνολικό αριθμό των λέξεων στο κείμενο.

Το αποτέλεσμα είναι ότι τα διανύσματα των λέξεων με παρόμοιο περιεχόμενο βρίσκονται 'κοντά' [58] στο χώρο. Το GloVe δημιουργεί ουσιαστικά μια συνάρτηση $\phi(\cdot) : Words \rightarrow \mathbb{R}^n$ η οποία δέχεται μια λέξη και επιστρέφει το διάνυσμα αυτής της λέξης και μετατρέπει την 'πρόσθεση' και 'αφαίρεση' εννοιολογικού περιεχομένου των λέξεων στις αντίστοιχες αριθμητικές πράξεις. Στις σχέσεις 3.3 και 3.4 φαίνονται δύο τέτοια παραδείγματα.

$$\phi(\text{“queen”}) = \phi(\text{“king”}) - \phi(\text{“man”}) + \phi(\text{“woman”}) \tag{3.3}$$

$$\phi(\text{“berlin”}) = \phi(\text{“athens”}) - \phi(\text{“greece”}) + \phi(\text{“germany”}) \tag{3.4}$$

Κεφάλαιο 4

Βασικές Αρχιτεκτονικές

Σε αυτό το κεφάλαιο θα μελετηθούν οι βασικές αρχιτεκτονικές και σχεδιαστικές υλοποιήσεις των μοντέλων που αντιμετωπίζουν το πρόβλημα του Dialog State Tracking .

4.1 Αμφίδρομα RNN (bi-Directional RNN)

Όπως ήδη έχει αναφερθεί στα κεφάλαια 2.3.5.1, 2.3.5.2 και 2.3.5.3, τα RNN, LSTM και GRU είναι ακολουθιακά/επαναλαμβανόμενα (recurrent) δίκτυα των οποίων κοινό χαρακτηριστικό είναι ότι σε κάθε βήμα (timestep) η έξοδος υπολογίζεται από την τρέχουσα είσοδο και την προηγούμενη κατάσταση. Αυτή η λογική υποδεικνύει πως η κάθε έξοδος εξαρτάται από όλες τις προηγούμενες εισόδους και η τελική κατάσταση του δικτύου εμπεριέχει πληροφορία απ΄ όλες τις εισόδους. Ουσιαστικά ένα τέτοιο δίκτυο σε κάθε βήμα προσπαθεί να προβλέψει στην έξοδό του την επόμενη τιμή βάσει της ακολουθίας που έχει προηγηθεί μέχρι στιγμής. Ένα τέτοιο μονοκατευθυντικό δίκτυο δηλαδή έχει πρόσβαση μόνο στο παρελθόν. Πολλές φορές όμως, η γνώση του μέλλοντος βοηθάει περισσότερο στην πρόβλεψη μιας τιμής. Για παράδειγμα, έστω ότι ένα τέτοιο δίκτυο προσπαθεί να προβλέψει την επόμενη λέξη σε μια πρόταση όπως η 'I want to ...'. Προφανώς το σύνολο των λέξεων που ακολουθούν μετά από αυτήν την πρόταση είναι μεγάλο. Αν όμως το δίκτυο τροφοδοτηθεί και με τις λέξεις που έπονται '... free.' τότε είναι πιο εύκολο να προβλέψει την ενδιάμεση λέξη "break". Γι΄ αυτό το λόγο κατά τη διαδικασία της εκπαίδευσης το δίκτυο τροφοδοτείται από την αρχική ακολουθία και στη συνέχεια τροφοδοτείται με την αντίστροφη ακολουθία (reversed) [53], [46], [62]. Οι έξοδοι και οι καταστάσεις του δικτύου υπολογίζονται συνδυαστικά από τα δύο περάσματα. Οι δυο βασικότεροι συνδυασμοί είναι η πρόσθεση και η συνένωση (concatenation). Η πρώτη διατηρεί το μέγεθος των διανυσμάτων αλλά εμφανίζει μεγαλύτερη απώλεια πληροφορίας. Η δεύτερος συνδυασμός διατηρεί ανέπαφη την πληροφορία από τα δύο περάσματα της ακολουθίας αλλά διπλασιάζει το διανυσματικό μέγεθος.

4.2 Βαθιά RNN (Deep/Stacked RNN)

Στο κεφάλαιο 4.1 το δίκτυο αποτελείται από μόνο ένα στρώμα (layer) RNN. Το στρώμα αυτό αποτελείται από πολλά RNN, δέχεται τις εισόδους (την ακολουθία) και παράγει τις εξόδους. Τα βαθιά δίκτυα ακολουθούν την ίδια λογική [47], [64] μόνο μου αποτελούνται από περισσότερα στρώματα 'στοιβαγμένα' (stacked) το ένα πάνω στο άλλο. Σε αυτήν την περίπτωση το αρχικό στρώμα δέχεται τις εισόδους και με τις εξόδους που παράγει τροφοδοτεί το επόμενο στρώμα κοκ. Το τελικό στρώμα παράγει τις τελικές εξόδους του δικτύου. Πρόσθετα κρυφά στρώματα μπορούν να προστεθούν σε ένα νευρωνικό δίκτυο πολλαπλών στρώσεων για να το καταστήσουν πιο βαθύ. Κάθε στρώμα είναι σε θέση να συνδυάσει επιπλέον την αναπαράσταση της γνώσης από τα προηγούμενα στρώματα και να δημιουργήσει νέες αναπαραστάσεις σε υψηλότερα επίπεδα αφαίρεσης. Σε μια ακολουθία λέξεων, όπως ένα κείμενο, κάθε επίπεδο μπορεί να συνδυάσει εξαρτήσεις μεταξύ λέξεων που απέχουν διαφορετικά βήματα (timesteps) μεταξύ τους και να κατανοήσει πιο πολύπλοκες σχέσεις που υπάρχουν στα γλωσσικά δεδομένα. Στην εργασία Speech Recognition with Deep Recurrent Neural Networks [21] φαίνεται πως η απόδοση του μοντέλου εξαρτάται περισσότερο από το βάθος των στρωμάτων και όχι τόσο από το μέγεθος της κρυφής κατάστασης.

4.3 Seq2Seq Model (Encoder - Decoder)

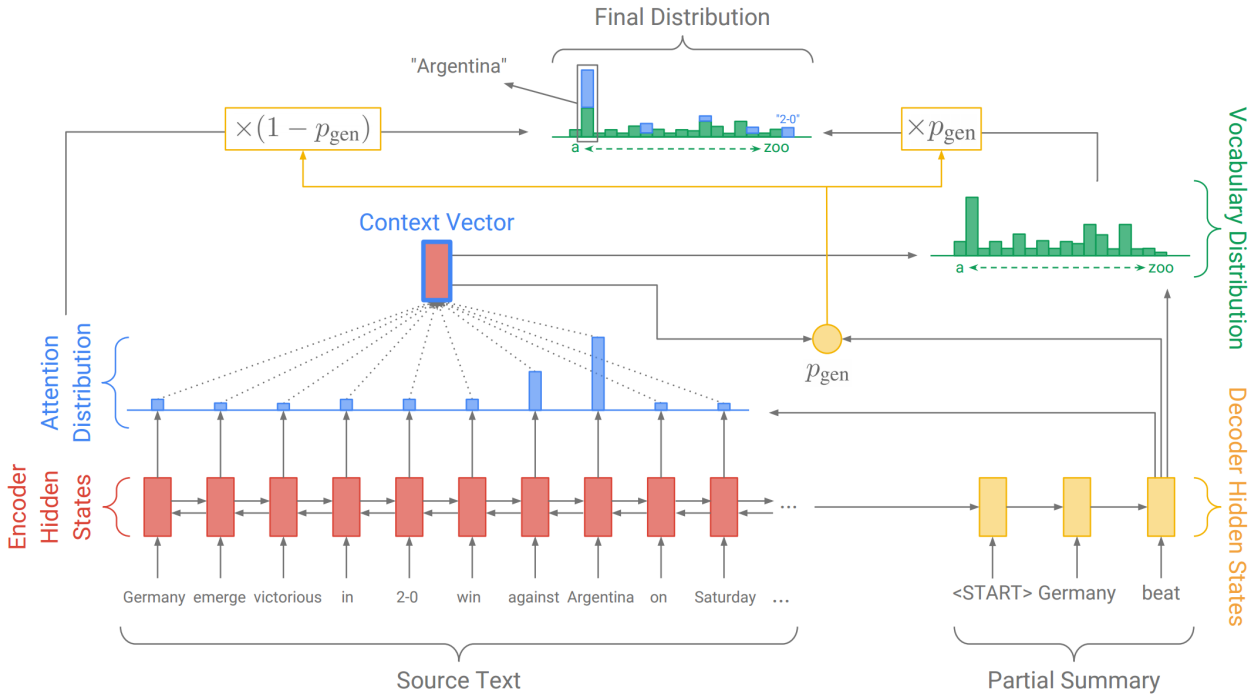
Η ανάγκη αντιστοίχισης ακολουθιακών δεδομένων οδήγησε στην ανάπτυξη των μοντέλων Sequence to Sequence (Seq2Seq) [59], [13]. Η αντιστοίχιση ακολουθιακών δεδομένων εμφανίζεται σε διάφορους τομείς όπως η μηχανική μετάφραση (machine translation), οι ερωταπαντήσεις (question answering), η περιγραφή ενός βίντεο (video captioning) κλπ. Το κοινό στοιχείο αυτών είναι ότι μια ακολουθία εισόδου αντιστοιχίζεται σε μια ακολουθία εξόδου με διαφορετικό μήκος και ίσως διαφορετικού τύπου δεδομένα. Για παράδειγμα, ένα βίντεο μπορεί να απεικονίζει ένα μπαλόνι που αιωρείται και η περιγραφή να είναι 'A balloon is flying.'. Παρατηρούμε ότι η είσοδος είναι η ακολουθία των εικόνων frames του βίντεο ενώ η έξοδος είναι μια ακολουθία από λέξεις. Προφανώς αυτές οι δύο ακολουθίες έχουν διαφορετικό μήκος καθώς το βίντεο μπορεί να αποτελείται από 100 frames ενώ η έξοδος αποτελείται από 4 λέξεις/διανύσματα. Επιπλέον τα δεδομένα είναι διαφορετικής φύσης (εικόνες-λέξεις).

Τα μοντέλα Seq2Seq [60] χρησιμοποιούν την αρχιτεκτονική του κωδικοποιητή - αποκωδικοποιητή (encoder - decoder) [27] για να αντιμετωπίσουν τέτοιου είδους προβλήματα. Ο κωδικοποιητής (encoder) είναι ένα RNN (LSTM, GRU) δίκτυο ενός ή πολλαπλών στρωμάτων ο οποίος δέχεται ακολουθιακά κάθε δεδομένο της εισόδου, εξάγει πληροφορία και διαδίδει την έξοδό του. Η τελική έξοδος είναι η τελική κρυφή κατάσταση που παράγεται από τον κωδικοποιητή του μοντέλου. Αυτό το διάνυσμα στοχεύει στην ενσωμάτωση της συνολικής πληροφορίας απ' όλα τα στοιχεία εισόδου προκειμένου να βοηθήσει τον αποκωδικοποιητή να κάνει ακριβείς προβλέψεις. Σημαντικό μέρος είναι το γεγονός ότι η τελική κατάσταση του κωδικοποιητή ορίζεται ως η αρχική κρυφή κατάσταση του αποκωδικοποιητή του μοντέλου. Κατά την διαδικασία εκπαίδευσης, είναι σημαντικό ο κωδικοποιητής να δημιουργεί τελικές

καταστάσεις οι οποίες 'μοιάζουν' όταν αντίστοιχα τα δεδομένα εισόδου είναι παρόμοια. Για παράδειγμα, εάν ένα βίντεο δείχνει ένα μπαλόνι να αιωρείται για 100 frames και ένα άλλο για 1000 frames θα πρέπει η τελική κατάσταση του κωδικοποιητή και για τις δύο ακολουθίες να είναι όμοια. Με αυτόν τον τρόπο διευκολύνεται στη συνέχεια ο αποκωδικοποιητής καθώς η αρχική κατάσταση η οποία του ορίζεται θα είναι όμοια και στις δυο περιπτώσεις προκειμένου να παράξει μια ακριβή περιγραφή. Ο αποκωδικοποιητής στη συνέχεια αφού αρχικοποιηθεί με την τελική κατάσταση του κωδικοποιητή, δέχεται ως είσοδο συνήθως κάποιο ειδικό token όπως "<START>","<GO>","<SOS>" και σε κάθε βήμα παράγει μια έξοδο η οποία θα είναι είσοδος στο επόμενο βήμα κοκ. Η αποκωδικοποίηση ολοκληρώνεται όταν ο αποκωδικοποιητής παράξει ένα ειδικό token όπως "<END>","<EOS>" ή σταματάει βίαια όταν ξεπεραστεί ένα προκαθορισμένο μήκος. Δεν είναι απαραίτητο ο αποκωδικοποιητής να δεχτεί κάποιο ειδικό token ως πρώτη είσοδο, αλλά είναι μια τεχνική η οποία δίνει ένα ξεκάθαρο όριο στην παραγόμενη ακολουθία του αποκωδικοποιητή.

4.4 Μηχανισμός Προσοχής και Διάνυσμα Συμφραζομένων (Attention Mechanism and Context Vector)

Τα μοντέλα Seq2Seq και γενικότερα τα RNN δίκτυα έχουν δύο βασικούς περιορισμούς ως προς την εκφραστικότητά τους. Αυτοί οι περιορισμοί είναι η ικανότητά τους να αποθηκεύουν πληροφορία στις εσωτερικές τους παραμέτρους σχετικά με την εργασία που εκπαιδεύονται και να ενσωματώνουν στην κρυφή κατάσταση την πληροφορία των εισόδων (information bottleneck). Όπως αναφέρθηκε στο κεφάλαιο 4.3, ο κωδικοποιητής παράγει μια τελική κατάσταση η οποία εμπεριέχει την πληροφορία όλων των εισόδων. Είναι λογικό όμως βάσει σχεδιασμού, η τελική κατάσταση να εξαρτάται περισσότερο από τις τελικές εισόδους και λιγότερο από τις αρχικές καθώς όσο αυξάνεται το μήκος της ακολουθίας, οι αριθμητικές τιμές του διανύσματος κρυφής κατάστασης που αντιπροσωπεύουν παρελθοντικότερες εισόδους επηρεάζονται και αλλάζουν περισσότερο απ' ότι όσες αντιπροσωπεύουν πιο πρόσφατες εισόδους στο μοντέλο. Είναι επιθυμητό λοιπόν ο αποκωδικοποιητής, κατά τη διαδικασία αποκωδικοποίησης σε κάθε βήμα, να δίνει σημασία (προσοχή) σε όλο το μήκος της ακολουθίας εισόδου και όχι μόνο να γνωρίζει την τελική κωδικοποιημένη κατάσταση για να προβλέψει. Ο μηχανισμός προσοχής (attention mechanism) [65], [14], [36] είναι μια σχεδιαστική υλοποίηση η οποία εφαρμόζεται στα μοντέλα Seq2Seq και βοηθά τον αποκωδικοποιητή να δώσει προσοχή σε όλες τις ενδιάμεσες καταστάσεις του κωδικοποιητή. Αναλυτικότερα, αντί να παραβλέπονται οι ενδιάμεσες καταστάσεις του κωδικοποιητή και να χρησιμοποιείται μόνο η τελική κατάσταση για την αρχικοποίηση του αποκωδικοποιητή, στην περίπτωση του μηχανισμού προσοχής όλες οι ενδιάμεσες καταστάσεις του κωδικοποιητή αποθηκεύονται σε έναν προσωρινό πίνακα ο οποίος χρησιμοποιείται σε κάθε βήμα (time step) αποκωδικοποίησης για την παραγωγή του διανύσματος συμφραζομένων (context vector). Αρχικά, κάθε ενδιάμεση κατάσταση του κωδικοποιητή συνδυάζεται γραμμικά με την κατάσταση του αποκωδικοποιητή και υπολογίζεται ένα σκορ (βάρος προσοχής). Στη συνέχεια όλες οι ενδιάμεσες καταστάσεις πολλαπλασιάζονται η κάθε μία με το αντίστοιχο βάρος προσοχής τους και προστίθονται παράγοντας κατ'



Σχήμα 4.1: Pointer - Generator Network (with Attention Mechanism)

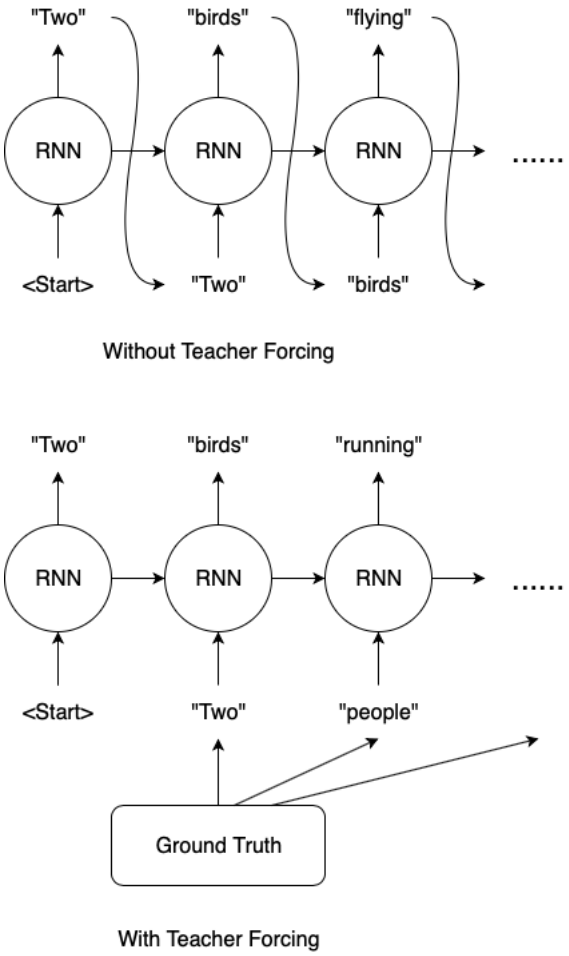
αυτόν τον τρόπο το διάνυσμα συμφραζομένων. Δηλαδή, το διάνυσμα συμφραζομένων είναι ένα διάνυσμα που εξαρτάται έμμεσα από το διάνυσμα κατάστασης του αποκωδικοποιητή καθώς αυτό συμμετέχει στον υπολογισμό των βαρών προσοχής και επιπλέον εμπεριέχει όλες τις ενδιάμεσες καταστάσεις του κωδικοποιητή σταθμισμένες ως προς αυτά τα βάρη (σκορ). Το διάνυσμα συμφραζομένων μπορεί να χρησιμοποιηθεί με διάφορους τρόπους. Ο πιο συχνός είναι ο συνδυασμός του διανύσματος συμφραζομένων είτε με την προηγούμενη κατάσταση του αποκωδικοποιητή είτε με την είσοδό του προκειμένου να υπολογιστεί η κρυφή κατάσταση ή η νέα είσοδος αντίστοιχα για το επόμενο βήμα. Κατ΄ αυτόν τον τρόπο, η πρόβλεψη του αποκωδικοποιητή εξαρτάται από ένα διάνυσμα που εμπεριέχει όλες τις ενδιάμεσες καταστάσεις της κωδικοποιημένης ακολουθίας. Επομένως, είναι περισσότερο πιθανό να δώσει σημασία σε συγκεκριμένα μέρη της ακολουθίας για να πετύχει μια πιθανότερη πρόβλεψη. Το διάνυσμα συμφραζομένων μπορεί επίσης να χρησιμοποιηθεί και σε άλλου τύπου μοντέλα. Για παράδειγμα, στο πρόβλημα question answering, ο κωδικοποιητής δέχεται ως είσοδο ένα κείμενο και ο αποκωδικοποιητής προβλέπει την απάντηση σε μια ερώτηση δεδομένου του κειμένου. Είναι πιθανό όμως η ερώτηση να μην επιδέχεται κάποια απάντηση στο πλαίσιο του κειμένου. Σε αυτήν την περίπτωση, χρησιμοποιείται ένας δυαδικός ταξινομητής, ο οποίος δέχεται ως είσοδο το διάνυσμα συμφραζομένων (συνήθως αυτό που παράγεται στην πρώτο βήμα αποκωδικοποίησης) και προβλέπει εάν πρέπει να δοθεί κάποια απάντηση ή όχι, δηλαδή εάν θα πρέπει να ληφθεί υπόψιν η έξοδος του αποκωδικοποιητή ως απάντηση ή δεν πρέπει να υπολογιστεί καθόλου καθώς δεν υπάρχει απάντηση.

4.5 Τεχνικές σύνοψης κειμένου και Pointer - Generator Network

Υπάρχουν δυο βασικές κατηγορίες μεθόδων για τη σύνοψη κειμένου, η μέθοδος της εξόρυξης (extractive method) και η αφαιρετική μέθοδος (abstractive method). Οι τεχνικές εξόρυξης για τη σύνοψη κειμένου [54], [2] επιλέγουν σχετικές φράσεις από το κείμενο εισόδου και τις αναδιατάσσουν ώστε να σχηματίσουν προτάσεις. Αυτές οι μέθοδοι χρησιμοποιούνται συχνά στη βιομηχανία, καθώς είναι πολύ εύκολο να εφαρμοστούν. Χρησιμοποιούν υπάρχουσες φράσεις φυσικής γλώσσας και έχουν αρκετά καλή ακρίβεια κυρίως σε μικρά κείμενα όπως e-mails ή κριτικές πελατών για προϊόντα. Επιπλέον, επειδή είναι τεχνικές χωρίς επίβλεψη, η εκπαίδευση τέτοιων μοντέλων είναι αρκετά γρήγορη. Από την άλλη πλευρά όμως, καθώς αυτές οι τεχνικές παίζουν απλά με τη σειρά των προτάσεων στο κείμενο για να συνοψίσουν, δεν έχουν τόσο καλή απόδοση όσο οι άνθρωποι, ειδικά σε μεγάλα κείμενα που απαιτείται μεγαλύτερο βάθος αφαίρεσης για την παραγωγή μιας σύνοψης και γι΄ αυτό το λόγο χρησιμοποιούνται αφαιρετικές μέθοδοι.

Οι αφαιρετικές μέθοδοι στηρίζονται στο μοντέλο Encoder - Decoder όπως περιγράφηκε στο κεφάλαιο 4.3, δηλαδή κωδικοποιούν ένα κείμενο εισόδου και αποκωδικοποιούν προτάσεις που περιγράφουν συνοπτικά αυτό το κείμενο. Αυτό έχει ένα τεράστιο πλεονέκτημα ότι αυτά τα μοντέλα δεν περιορίζονται μόνο σε λέξεις ή φράσεις που υπάρχουν στο κείμενο όπως οι μέθοδοι εξόρυξης, αλλά έχουν τη δυατότητα να παράξουν νέες προτάσεις μέσω της χρήσης ενός λεξιλογίου πάνω στο οποίο έχουν εκπαιδευτεί. Αν και η σύνοψη ενός κειμένου χρησιμοποιώντας αφαιρετικές μεθόδους είναι πιο διαισθητικά κοντά στον άνθρωπο, έχουν κι αυτές κάποια βασικά μειονεκτήματα. Πρώτον, η εκπαίδευση του μοντέλου απαιτεί πολλά δεδομένα και επομένως χρόνο. Επίσης, τα δεδομένα αυτά πρέπει να έχουν αναλυθεί από ανθρώπους, δηλαδή κάθε κείμενο εισόδου πρέπει να συνοδεύεται και από την αντίστοιχη σύνοψή του καθώς η μάθηση είναι επιβλεπόμενη. Ένα ακόμα βασικό μειονέκτημα είναι ότι πολλές λέξεις ή φράσεις μπορούν να επαναληφθούν σε μεγάλο βαθμό κατά τη διαδικασία πραγωγής της σύνοψης από τον αποκωδικοποιητή. Επιπλέον, ένα εγγενές πρόβλημα που συναντάται στις αφαιρετικές μεθόδους είναι ότι σημαντικές λεπτομέρειες που αναφέρονται στο κείμενο μπορεί αναπαράγονται λανθασμένα. Για παράδειγμα, στο κείμενο μπορεί να αναφέρεται ότι η Γερμανία νίκησε την Αργεντινή 3-2 και ο αποκωδικοποιητής να αναπαράξει το σκορ ως 3-3 ή 2-2 το οποίο θα ήταν και νοηματικά λάθος. Αυτό συμβαίνει διότι όπως αναφέρθηκε ο αποκωδικοποιητής έχει τη δυνατότητα να αναπαράξει λέξεις από το λεξιλόγιο του και επομένως να προβλέψει λανθασμένα τη λέξη ‘2’ αντί της λέξης 3. Αυτό το φαινόμενο οδηγεί στην ανάγκη της χρήσης λέξεων ή φράσεων αυτούσιων από το κείμενο εισόδου.

Η τεχνική Pointer - Generator [54] καταφέρνει να συνδυάσει τις δυο προαναφερόμενες μεθόδους, δηλαδή το μοντέλο για κάθε πρόβλεψη επόμενης λέξης έχει τη δυνατότητα είτε να επιλέξει λέξεις αυτούσιες (αντιγραφή) μέσα από το κείμενο εισόδου (Pointer), είτε να αναπαράξει λέξεις από το λεξιλόγιο πάνω στο οποίο έχει εκπαιδευτεί (Generator). Σε σύγκριση με τα μοντέλα Seq2Seq με Μηχανισμό Προσοχής (Attention Mechanism), το μοντέλο Pointer - Generator αποδίδει καλύτερα κυρίως κατά την αντιγραφή λέξεων από το κείμενο εισόδου. Η



Σχήμα 4.2: With and Without Teacher Forcing

αντιγραφή λέξεων από το κείμενο δίνει ένα μεγάλο πλεονέκτημα στο μοντέλο καθώς μπορεί να διαχειριστεί λέξεις οι οποίες είναι εκτός λεξιλογίου (Out of Vocabulary words). Στα πλαίσια του μοντέλου της διπλωματικής, η χρήση της τεχνικής Pointer - Generator είναι αναγκαία αφού ο αποκωδικοποιητής πρέπει να είναι σε θέση να αναπαράξει για παράδειγμα το όνομα ενός ξενοδοχείου ή ενός προορισμού που όμως δεν υπήρχε στα δεδομένα εκπαίδευσης και είναι εκτός του λεξιλογίου. Στην εικόνα 4.1 φαίνεται το μοντέλο Pointer - Generator. Ουσιαστικά, γίνεται ένας υπολογισμός μιας τελικής κατανομής πιθανοτήτων για την πρόβλεψη της επόμενης λέξης, η οποία προκύπτει ως σταθμισμένος μέσος των κατανομών που υπολογίζονται πάνω στο κείμενο εισόδου και στο σύνολο των λέξεων του λεξιλογίου. Μια σημαντική υποσημείωση, όπως φαίνεται και στην εικόνα, το διάνυσμα συμφραζομένων (Context Vector) που περιγράφηκε στο κεφάλαιο 4.4, χρησιμοποιείται στο μοντέλο για τον υπολογισμό της κατανομής πιθανότητας στο σύνολο των λέξεων του λεξιλογίου.

4.6 Teacher Forcing Method

Κατά τη διαδικασία εκπαίδευσης ενός Encoder - Decoder μοντέλου, ο αποκωδικοποιητής (decoder) προβλέπει την επόμενη λέξη λαμβάνοντας ως είσοδο την λέξη από την πρόβλεψη του προηγούμενου βήματος και την κρυφή κατάσταση που ενσωματώνει συνολικά την πληροφορία της ακολουθίας λέξεων. Πολλές φορές, ο αποκωδικοποιητής δεν προβλέπει την επιθημητή λέξη με αποτέλεσμα να εκπαιδεύεται σε λανθασμένες ακολουθίες. Για να αποφευχθεί αυτό το φαινόμενο, χρησιμοποιείται η διαδικασία teacher forcing κατά την οποία, αντί ο αποκωδικοποιητής να λαμβάνει ως είσοδο την λέξη από την πρόβλεψη του προηγούμενου βήματος, λαμβάνει ως είσοδο την επιθυμητή λέξη, δηλαδή αυτή που θα έπρεπε να έχει προβλέψει. Η διαδικασία αυτή διδάσκει (Teaching) ουσιαστικά τον αποκωδικοποιητή τη σωστή πρόβλεψη εξαναγκάζοντας (forcing) την είσοδο σε κάθε επόμενο βήμα ώστε να προκύψει η επιθυμητή ακολουθία (Ground Truth). Η εκπαίδευση με τη χρήση της μεθόδου Teacher Forcing [5], [30], [63] κάνει το μοντέλο να συγκλίνει ταχύτερα. Στα πρώτα στάδια της εκπαίδευσης, οι προβλέψεις του μοντέλου είναι άστοχες με αποτέλεσμα οι κρυφές καταστάσεις του μοντέλου να ενημερώνονται από μια σειρά λανθασμένων προβλέψεων. Η συσσώσερυση τέτοιων σφαλμάτων κάνει ακόμα πιο δύσκολη την εκπαίδευση του μοντέλου επομένως η χρήση της μεθόδου καθίσταται αναγκαία. Ωστόσο, κατά τη αξιολόγηση του μοντέλου, δεδομένου ότι δεν υπάρχει διαθέσιμη κάποια επιθυμητή ακολουθία, ο αποκωδικοποιητής θα χρειαστεί να τροφοδοτήσει τη δική του προηγούμενη πρόβλεψη ως είσοδο στον εαυτό του για την επόμενη πρόβλεψη. Αυτή η ασυμφωνία ανάμεσα στην εκπαίδευση του μοντέλου και της αξιολόγησής του μπορεί να οδηγήσει σε κακή απόδοση και αστάθεια. Γι΄ αυτό το λόγο στα αρχικά στάδια εκπαίδευσης του μοντέλου γίνεται αποκλειστική χρήση της μεθόδου Teacher Forcing προκειμένου να δοθεί μια αρχική βοήθεια στο μοντέλο και όσο η εκπαίδευση εκτυλίσσεται μειώνεται το ποσοστό που χρησιμοποιείται ώστε σταδικά να καλυφθεί η ασυμφωνία των δύο τρόπων. Στην εικόνα 4.2 φαίνεται η διαφορά ανάμεσα στη χρήση ή μη της τεχνικής Teacher Forcing. Η επιθυμητή ακολουθία είναι η 'Two people running'. Παρατηρούμε πως το μοντέλο χωρίς τη χρήση της μεθόδου μετά τη λέξη 'Two' προβλέπει λανθασμένα τη λέξη 'birds' και το σφάλμα διαδίδεται και στις επόμενες λέξεις ('flying' αντί για 'running').

Κεφάλαιο 5

Παρακολούθηση Κατάστασης Διαλόγου στο MultiWOZ Dataset

Στο παρόν κεφάλαιο παρουσιάζεται αρχικά το σύνολο των δεδομένων που χρησιμοποιήθηκε στο πλαίσιο εκπαίδευσης του βασικού μοντέλου (base model). Στη συνέχεια αναπτύσσεται διεξοδικά ο σχεδιασμός του μοντέλου ο οποίος βασίζεται στις αρχιτεκτονικές που αναφέρθηκαν στο κεφάλαιο 4. Ακολουθεί η πειραματική διαδικασία στην οποία εφαρμόζονται τροποποιήσεις παραμέτρων και τεχνικών του μοντέλου με σκοπό τη βελτίωση της επίδοσής του. Συμπληρωματικά εξετάζονται τα σφάλματα των προβλέψεων με σκοπό τη βαθύτερη κατανόηση των αδυναμιών του μοντέλου για πιθανές μελλοντικές τροποποιήσεις. Στο τέλος του κεφαλαίου, γίνεται μια σύντομη παρουσίαση παρεμφερών εργασιών και τελική σύκριση των αποτελεσμάτων.

5.1 Δεδομένα (Multi-WOZ Dataset)

Η μηχανική μάθηση έχει κάνει μεγάλα άλματα στον ερευνητικό τομέα της παρακαλούθησης διαλογικής κατάστασης (Dialog State Tracking). Ωστόσο, η ανάπτυξη μεθόδων εμποδίζεται από το μικρό πλήθος δεδομένων που είναι διαθέσιμα. Η παρούσα διπλωματική εργασία στηρίχθηκε πάνω σε ένα μεγάλο σύνολο δεδομένων, το MultiWOZ (Multi-Domain Wizard-of-Oz) [7]. Το παρόν σύνολο δεδομένων αποτελεί μια μεγάλη συλλογή γραπτών συζητήσεων που έχουν διατελεστεί μεταξύ ανθρώπων. Τα δεδομένα είναι πλήρως σημειωμένα με ετικέτες (labeled data) και οι διάλογοι εκτείνονται σε ένα ευρύ φάσμα θεματικών ενοτήτων (multi-domain dialogues).

Επί του παρόντος, το MultiWOZ είναι τουλάχιστον μια τάξη μεγέθους μεγαλύτερο απ΄ όλα τα προηγούμενα προσημειωμένα σύνολα δεδομένων και αυτός είναι ο λόγος για τον οποίο χρησιμοποιήθηκε στην παρούσα διπλωματική. Στον πίνακα 5.1 φαίνεται η σύγκριση του MultiWOZ με παρόμοια σύνολα δεδομένων πάνω σε στατιστικά στοιχεία που αφορούν το Dialog State Tracking . Οι αριθμητικές τιμές με bold δηλώνουν τη μέγιστη τιμή στην αντίστοιχη σειρά του πίνακα.

Metric	DSTC2	SFX	WOZ2.0	FRAMES	KVRET	M2M	MultiWOZ
# Dialogues	1,612	1,006	600	1,369	2,425	1,500	**8,438**
Total # turns	23,354	12,396	4,472	19,986	12,732	14,796	**113,556**
Total # tokens	199,431	108,975	50,264	251,867	102,077	121,977	**1,490,615**
Avg. turns per dialogue	14.49	12.32	7.45	**14.60**	5.25	9.86	13.46
Avg. tokens per turn	8.54	8.79	11.24	12.60	8.02	8.24	**13.13**
Total unique tokens	986	1,473	2,142	12,043	2,842	1,008	**23689**
# Slots	8	14	4	**61**	13	14	24
# Values	212	1847	99	3871	1363	138	**4510**

Πίνακας 5.1: Σύγκριση του MultiWOZ Dataset με παρόμοια datasets

5.1.1 Ανάλυση Δεδομένων

Όπως ήδη αναφέρθηκε, το MultiWOZ είναι επί του παρόντος το μεγαλύτερο σύνολο κειμένων από διαλόγους μεταξύ ανθρώπων. Οι διάλογοι εκτείνονται πάνω σε 7 διαφορετικές θεματικές ενότητες, οι οποίες είναι 'Ξενοδοχείο' (Hotel), 'Τραίνο' (Train), 'Αξιοθέατο' (Attraction), 'Εστιατόριο' (Restaurant), 'Ταξί' (Taxi), 'Νοσοκομείο' (Hospital) και 'Αστυνομία' (Police). Περιέχει 8438 διαλόγους με 13 στροφές κατά μέσο όρο, δηλαδή 26 εναλλαγές μεταξύ των δυο συνομιλητών. Επίσης, διαθέτει 30 ζευγάρια από θεματικές ενότητες και πεδία (domain-slot pairs), το εύρος τιμών (οντολογία) των οποίων είναι 4500. Είναι σημαντικό να αναφερθεί ότι οι θεματικές ενότητες 'Νοσοκομείο' (Hospital) και 'Αστυνομία' (Police) δε λήφθηκαν υπόψιν για την εκπαίδευση και την αξιολόγηση καθώς αποτελούν ένα πολύ μικρό ποσοστό των συνολικών διαλόγων με αποτέλεσμα οι διάλογοι που αντιστοιχούν σε αυτές της κατηγορίες να μην επαρκούν για μια ουσιώδη εκπαίδευση. Στον πίνακα 5.2 φαίνονται όλα τα πεδία χωρισμένα ανά θεματική ενότητα. Επίσης, στις τελευταίες τρεις σειρές του πίνακα παρουσιάζεται το πλήθος των δειγμάτων που χρησιμοποιήθηκαν για τη δημιουργία του Train Set, του Validation Set και του Test Set.

	Hotel	Train	Attraction	Restaurant	Taxi
Slots	price type parking stay day people area stars internet name	destination departure day arrive by leave at people	area name type	food price area name time day people	destination departure arrive by leave at
Train Set	3381	3103	2717	3813	1654
Validation Set	416	484	401	438	207
Test Set	394	494	395	437	195

Table 5.2: MultiWOZ - Dataset Information

Παρακάτω ακολουθεί ένα δείγμα σε απλοποιημένη μορφή από το σύνολο δεδομένων.

- **Dialogue-ID:** PMUL1635
  - **Turn Number:** 0
    - **User Utterance:** "I need to book a hotel in the east that has 4 stars"
    - **System Utterance:** "I can help you with that. What is your price range ?"
    - **Dialogue State:** {hotel-area: east, hotel-stars: 4}
  - **Turn Number:** 1
    - **User Utterance:** "That does not matter as long as it has free wifi and parking"
    - **System Utterance:** "..."
    - **Dialogue State:** {hotel-area: east, hotel-stars: 4, hotel-parking: yes, hotel-internet: yes}

Η κάθε στροφή turn περιέχει την έκφραση του χρήστη user utterance και την απάντηση από το σύστημα system utterance όπως επίσης και τη διαλογική κατάσταση Dialog State. Η διαλογική κατάσταση είναι ένα σύνολο από ζευγάρια θεματικής ενότητας-πεδίου και τιμής domain-slot value pairs. Σκοπός του μοντέλου είναι βάσει της ιστορίας του διαλόγου που έχει προηγηθεί μεταξύ του χρήστη και του συστήματος και της τρέχουσας στροφής να προβλέψει την τρέχουσα κατάσταση διαλόγου, δηλαδή αρχικά να αναγνωρίσει σωστά τα ζευγάρια τα οποία 'ενεργοποιούνται' από το διάλογο και στη συνέχεια να αποδώσει σε αυτά τις προβλεπόμενες τιμές τους.

Το MultiWOZ Dataset περιέχει επιπλέον μια μεγάλη βάση δεδομένων για τις αντίστοιχες θεματικές ενότητες που αναφέρθηκαν. Η βάση περιέχει πληροφορίες σχετικά με ξενοδοχεία, λεωφορεία, ταξί κλπ. Παρακάτω παρουσιάζονται δυο παραδείγματα από τη βάση των ξενοδοχείων και των τρένων αντίστοιχα.

- **Hotel DB:**
  - **address** "124 tenison road"
  - **area** "east"
  - **internet** "yes"
  - **parking** "no"
  - **name** "a and b guest house"
  - **phone** "01223315702"
  - **postcode** "cb12dp"
  - **pricerange** "moderate"
  - **stars** "4"
  - **type** "guesthouse"
- **Train DB:**
  - **arriveBy** "05:51"
  - **day** "monday"
  - **departure** "cambridge"
  - **destination** "london kings cross"
  - **duration** "51 minutes"
  - **leaveAt** "05:00"
  - **price** "23.60 pounds"
  - **trainID** "TR7075"

5.1.2 Επεξεργασία Δεδομένων

Το MultiWOZ Dataset παρέχει τα δεδομένα των διαλόγων και των βάσεων σε αρχεία τύπου ".json". Το πρώτο βήμα είναι η ανάλυση των κειμένων (parsing) και των βάσεων προκειμένου να παραχθούν οι διάλογοι και η οντολογία. Οι διάλογοι είναι ένα σύνολο από στροφές (turns), δηλαδή ερωταπαντήσεις (a user utterance followed by a system utterance) ανάμεσα στον χρήστη και το σύστημα. Κάθε διάλογος συνοδεύεται και με τα αντίστοιχα σημειώματα labels τα οποία είναι η κατάσταση του διαλόγου μετά από κάθε στροφή και περιέχουν τις τιμές που πρέπει να λάβουν τα πεδία των θεματικών ενοτήτητων μετά τη στροφή. Για κάθε

διάλογο δημιουργούνται όλοι οι δυνατοί 'υποδιάλογοι'. Για παράδειγμα, αν ένας διάλογος αποτελείται από 5 στροφές (και αντίστοιχα 5 ετικέτες διαλογικής κατάστασης για κάθε στροφή) τότε θα δημιουργηθούν 5 διάλογοι με τον πρώτο να περιέχει μόνο την πρώτη στροφή, ο δεύτερος να περιέχει την πρώτη και τη δεύτερη, ο τρίτος να περιέχει τις τρεις πρώτες κοκ και ο καθένας έχει ως προσημειωμένη ετικέτα (label) τη διαλογική κατάσταση (dialogue state) που αντιστοιχεί στην τελευταία του στροφή. Στη συνέχεια, σε κάθε έκφραση (utterance) ακολουθεί η διαδικασία tokenization προκειμένου να διαχωρίσουμε τις λέξεις κι επιπλέον να αντιστοιχίσουμε (mapping) λέξεις όπως "it's", "didn't", "we've", "dinners" στις συστατικές τους λέξεις "it is", "did not", "we have", "dinner -s" αντίστοιχα. Επιπλέον, δημιουργούνται ειδικές λέξεις Special Tokens που εξυπηρετούν συγκεκριμένο σκοπό για τη διευκόλυνση της εκπαίδευσης του μοντέλου. Οι ειδικές λέξεις "<SOS>" (Start of Sentence), "<EOS>" (End of Sentence) μπαίνουν εμβόλιμα στην αρχή και στο τέλος αντίστοιχα κάθε έκφρασης (utterance). Η εκπαίδευση του μοντέλου απαιτεί τα δεδομένα εισόδου (εν προκειμένω οι ακολουθίες των λέξεων) να είναι σταθερού μήκους. Όμως κάθε διάλογος αποτελείται από διαφορετικό πλήθος λέξεων. Γι΄ αυτό το λόγο χρησιμοποιείται η ειδική λέξη "<PAD>" (Padding) η οποία μπαίνει εμβόλιμα στο τέλος (έπειτα του <EOS>) σε διαλόγους με πλήθος λέξεων μικρότερο από τον μεγαλύτερο διάλογο προκειμένου να μετατραπούν όλοι σε ίσου μήκους ακολουθίες. Τέλος, χρησιμοποείται η ειδική λέξη "<UNK>" (Unknown) για να αντιμετωπιστούν λέξεις εκτός λεξιλογίου (Out of Vocabulary (OOV) words).

5.1.3 Διαχωρισμός Δεδομένων

Το τελικό στάδιο επεξεργασίας των δεδομένων είναι ο διαχωρισμός του αρχικού συνόλου στα σύνολα εκπαίδευσης (Train Set), επικύρωσης (Validation Set) και ελέγχου (Test Set). Το Train Set όπως υποδηλώνει και το όνομά του χρησιμοποιείται για την εκπαίδευση του μοντέλου. Το Validation Set χρησιμοποιείται για την αξιολόγηση των μετρικών κατά τη διάρκεια της εκπαίδευσης. Τέλος, το Test Set χρησιμοποείται για την τελική αξιολόγηση του μοντέλου αφού έχει ολοκληρωθεί η διαδικασία της εκπαίδευσης. Σε ευτό το σημείο είναι σημαντικό να αναφέρουμε ότι το μοντέλο είναι προκατειλημμένο (biased) στα δύο πρώτα σύνολα δεδομένων καθώς κατά τη διαδικασία της εκπαίδευσης ενσωματώνει το Train Set και το καλύτερο μοντέλο έχει αξιολογηθεί βάσει του Validation Set. Επομένως, οι τελικές μετρικές που αξιολογούν την απόδοση του μοντέλου είναι αυτές που έχουν μετρηθεί πάνω στο Test Set, δηλαδή ένα σύνολο που είναι άγνωστο στο μοντέλο. Τα ποσοστά διαχωρισμού του αρχικού συνόλου δεδομένων στα τρία υποσύνολα είναι 80%, 10% και 10% αντίστοιχα.

5.1.4 Λεξιλόγιο

Το λεξιλόγιο είναι μια δομή λεξικού (dictionary) το οποίο αντιστοιχεί έναν δείκτη i σε μια λέξη w ($w_i = vocab(i), i \in [0, |V_{size}|]$). Όπως θα δούμε στη συνέχεια, το μοντέλο προβλέπει πιθανότητες πάνω στους δείκτες του λεξιλογίου που σημαίνει ότι το μέγεθος του λεξιλογίου πρέπει να παραμένει σταθερό καθ΄ όλη τη διάρκεια εκπαίδευσης και τελικής χρήσης (inference mode) του μοντέλου. Επειδή, κατά τη διάρκεια χρήσης μπορεί να εμφανιστούν άγνωστες

λέξεις (out of vocabulary), υπάρχει εξαρχής μια καταχώριση στο λεξιλόγιο της ειδικής λέξης "<UNK>" προκειμένου να αντιμετωπιστεί το πρόβλημα της επέκτασης του λεξιλογίου. Στη συνέχεια, γίνεται αναλυτικότερη παρουσίαση στη διαχείριση τέτοιων λέξεων. Εδώ αξίζει να αφερθεί ότι το λεξιλόγιο δημιουργήθηκε χρησιμοποιώντας τις λέξεις που ανήκουν μόνο στο train set χωρίς να ληφθούν υπόψιν αυτές που ανήκουν στο test set.

5.2 Αρχιτεκτονική Μοντέλου

Το μοντέλο αποτελείται από τρία μέρη, των κωδικοποιητή εκφράσεων (utterance encoder), την πύλη πεδίου (slot gate) και τον αποκωδικοποιητή κατάστασης (state decoder). Ο κωδικοποιητής είναι ένα δίκτυο που κωδικοποιεί τις εκφράσεις του διαλόγου. Δέχεται δηλαδή σε κάθε βήμα timestep ως είσοδο την τρέχουσα λέξη και παράγει ως έξοδο την επόμενη κρυφή κατάσταση (hidden state) η οποία θα χρησιμοποιηθεί στο επόμενο βήμα με είσοδο την επόμενη λέξη κοκ. Σκοπός του κωδικοποιητή είναι αφού τροφοδοτηθεί με όλες τις λέξεις του διαλόγου, να παράξει ένα τελικό διάνυσμα κατάστασης το οποίο εμπεριέχει όλη την πληροφορία του διαλόγου. Ο αποκωδικοποιητής κατάστασης (state decoder) αποκωδικοποιεί ανεξάρτητα για όλα τα ζευγάρια θεματική ενότητα-πεδίο (domain-slot), δηλαδή δέχεται ως αρχική είσοδο ένα ζευγάρι και παράγει ως έξοδο την προβλεπόμενη τιμή που έχει αυτό το ζευγάρι. Επειδή όμως πολλά ζευγάρια δεν πρέπει να πάρουν κάποια τιμή καθώς δε γίνεται αναφορά σε αυτά στον διάλογο, χρησιμοποιείται η πύλη πεδίου (slot gate) η οποία αποφασίζει αν η τιμή που προβλέπει ο αποκωδικοποιητής πρέπει να χρησιμοποιηθεί ή αν το ζευγάρι δεν αναφέρεται στον διάλογο οπότε δεν πρέπει να αποδοθεί κάποια τιμή σε αυτό.

Ορίζεται ως $X = \{(U_1, R_1), (U_2, R_2), ..., (U_T, R_T)$ το σύνολο στροφών turns T που αποτελούν το διάλογο, όπου η κάθε στροφή αποτελείται από την έκφραση του χρήστη user utterance U ακολουθούμενη από την έκφραση του συστήματος system response R. Επιπλέον, ορίζεται ως $B = \{B_1, B_2, ..., B_T\}$ ως το σύνολο των διαλογικών καταστάσεων dialog state / belief state για κάθε στροφή. Κάθε διαλογική κατάσταση B_i είναι μια τριπλέτα (tuple) που αποτελείται από τη θεματική ενότητα (domain), το πεδίο (slot) και την τιμή (value), $B_i = (D_n, S_m, Y_j)$ όπου $n \in [1, N], m \in [1, M], j \in [1, J]$ και N, M, J είναι το πλήθος των θεματικών ενοτήτων, το πλήθος των πεδίων και το πλήθος των τιμών αντίστοιχα. Όπως φαίνεται και στον πίνακα 5.2 κάθε θεματική ενότητα έχει διαφορετικό πλήθος πεδίων. Τα συνολικά ζευγάρια που σχηματίζονται είναι 30 στο σύνολο. Δηλαδή $N = 5, M = M(n) = \{10, 6, 3, 7, 4\}, J = 30$. Για παράδειγμα μια διαλογική κατάσταση είναι η $B_{example} = \{(hotel, area, east), (restaurant, food, italian)\}$. Επίσης, ένα παράδειγμα όπου φαίνεται η εναλλαγή τιμής μιας τριπλέτας κατά τη διάρκεια του διαλόγου είναι $B_{example} = \{(hotel, day, monday), (hotel, day, tuesday)\}$ όπου στην πρώτη στροφή ο χρήστης ήθελε να κάνει μια κράτηση σε ένα ξενοδοχείο τη Δευτέρα (Monday) αλλά στη δεύτερη στροφή, είτε λόγο μη διαθεσιμότητας είτε αλλαγής προτίμησης ο χρήστης αποφάσισε να κάνει την κράτηση την Τρίτη (Tuesday).

5.2.1 Κωδικοποιητής Εκφράσεων Utterance Encoder

Ο κωδικοποιητής εκφράσεων είναι ένα RNN δίκτυο. Η είσοδος του κωδικοποιητή είναι η ιστορία του διαλόγου X μέχρι και τη χρονική στιγμή t, $X_t = [U_1, R_1, U_2, R_2..., U_t, R_t]$ δηλαδή η ακολουθία όλων των λέξεων. Επίσης, $X_t \in \mathbb{R}^{|X_t| \times d_{emb}}$ όπου $|X_t|$ το πλήθος των λέξεων του διαλόγου και d_{emb} το μήκος του embedding διανύσματος της λέξης. Ο κωδικοποιημένος διάλογος συμβολίζεται ως $H_t = [h_1^{enc}, h_2^{enc}, ..., h_{|X_t|}^{enc}] \in \mathbb{R}^{|X_t| \times d_{hdd}}$ όπου το $h_t^{enc} \in \mathbb{R}^{d_{hdd}}$ είναι η κρυφή κατάσταση του δικτύου τη χρονική στιγμή t και d_{hdd} είναι το μήκος της κρυφής κατάστασης. Ουσιαστικά, το h_1^{enc} περιέχει με κωδικοποιημένη μορφή την πρώτη λέξη, το h_2^{enc} περιέχει τις δυο πρώτες λέξεις και τελικά το το h_t^{enc} εμπεριέχει κωδικοποιημένη όλη την πληροφορία του διαλόγου. Όπως ανφέρεται και στο κεφάλαιο 4.1, η χρήση αμφίδρομων δικτύων είναι καταλληλότερη επιλογή από τη χρήση μονοκατευθυντικών. Επομένως το δίκτυο του κωδικοποιητή είναι αμφίδρομο (bidirectional) που σημαίνει ότι τροφοδοτείται με την αρχική ακολουθία και με την αντίστροφή της. Επομένως, το τελικό κωδικοποιημένο διάνυσμα κρυφής κατάστασης προκύπτει ως το άθροισμα των κρυφών καταστάσεων σε κανονική και αντίστροφη τροφοδότηση, $h_i^{enc} = \overrightarrow{h_i^{enc,normal}} + \overleftarrow{h_i^{enc,reverse}}$. Ο σκοπός που διατηρείται όλη η ιστορία του διαλόγου στο διάνυσμα H_t, όπως θα δούμε στη συνέχεια, εξυπηρετεί στο να μπορεί το μοντέλο να προβλέψει την τιμή ενός πεδίου μιας θεματικής ενότητας βάσει του πεδίου μιας άλλης. Για παράδειγμα, εάν ένα χρήστης ζητήσει ένα ξενοδοχείο στο κέντρο μιας πόλης και στη συνέχεια ζητήσει να κλείσει ένα ταξί, τότε θα πρέπει η τοποθεσία αναχώρησης για το ταξί να αποδοθεί από την τοποθεσία του ξενοδοχείου. Δηλαδή σε δυο στροφές του διαλόγου η διαλογική κατάσταση θα είναι: ..., (hotel, area, centre), ... , (taxi, departure, centre), ..., επομένως το μοντέλο θα πρέπει να αξιοποιεί την πληροφορία παρελθοντικού διαλόγου προκειμένου να επιλύσει αυτό το πρόβλημα. Αυτός ο Μηχανισμός Προσοχής περιγράφθηκε στο κεφάλαιο 4.4.

5.2.2 Αποκωδικοποιητής Καταστάσεων State Decoder

Ο αποκωδικοποιητής καταστάσεων είναι ένα μονοκατευθυντικό δίκτυο RNN. Η κρυφή κατάσταση του αποκωδικοποιητή αρχικοποιείται από την τελική έξοδο $h_{|X_t|}^{enc}$ του κωδικοποιητή όπως συμβαίνει συνήθως στα μοντέλα encoder/decoder. Η πρώτη είσοδος του αποκωδικοποιητή είναι το άθροισμα των embedding vectors της θεματικής ενότητας domain και του πεδίου slot. Ο αποκωδικοποιητής αποκωδικοποιεί J φορές ανεξάρτητα για όλα τα ζευγάρια $domain - slot_j$. Σε κάθε βήμα s (time step) για το j domain-slot ο αποκωδικοποιητής δέχεται ως είσοδο μια λέξη-embedding vector w_{js} και παράγει την επόμενη κρυφή κατάσταση h_{js}^{dec}. Στη συνέχεια, αυτή η κρυφή κατάσταση τροφοδοτείται σε ένα Linear-Softmax δίκτυο, δηλαδή ένα επίπεδο νευρώνων με συνάρτηση ενεργοποίησης την Softmax ώστε από το χώρο του διανύσματος $h_{js}^{dec} \in \mathbb{R}^{d_{hdd}}$ να περάσουμε σε μια κατανομή πιθανότητας στο χώρο του λεξιλογίου $\mathbb{R}^{|V|}$.

$$P_{js}^{vocab} = Softmax(E \cdot (h_{js}^{dec})^T) \in \mathbb{R}^{|V|}, E \in \mathbb{R}^{|V| \times d_{hdd}} \tag{5.1}$$

Επιπλέον, υπάρχει η ανάγκη διαχείρισης λέξεων οι οποίες δεν βρίσκονται εντός του λεξιλογίου V (out of vocabulary words) επομένως το μοντέλο θα πρέπει να αναφέρεται σε αυτές

τις λέξεις από το αρχικό κείμενο [22], [67], [68]. Γι' αυτό το λόγο χρησιμοποιείται ένας μηχανισμός αντιγραφής (copy mechanism). Δηλαδή, εκτός της κατανομής P_{js}^{vocab}, δημιουργείται και μια κατανομή $P^{history}$ πάνω στην κωδικοποιημένη διαλογική ιστορία H_t.

$$P_{js}^{history} = Softmax(H_t \cdot (h_{js}^{dec})^T) = Softmax(h_1^{enc} \cdot (h_{js}^{dec})^T, h_2^{enc} \cdot (h_{js}^{dec})^T, ..., h_{|X_t|}^{enc} \cdot (h_{js}^{dec})^T) \in \mathbb{R}^{|X_t|} \tag{5.2}$$

Επίσης, αυτή η κατανομή χρησιμοποιείται και ως μηχανισμός προσοχής (attention mechanism) καθώς λαμβάνονται υπόψιν όλες οι προηγούμενες καταστάσεις του διαλόγου και δίνεται περισσότερο σημασία σε κάθε μία από αυτές. Βάσει της κατανομής $P_{js}^{history}$ δημιουργείται το διάνυσμα συμφραζομένων (context vector) c_{js} το οποίο προκύπτει από την σχέση 5.3.

$$c_{js} = P_{js}^{history} \cdot H_t \in \mathbb{R}^{d_{hdd}} \tag{5.3}$$

Πρόκειται για ένα διάνυσμα το οποίο δίνει βάρος σε κάθε κωδικοποιημένη κατάσταση του διαλόγου βάσει της κατανομής πιθανότητας $P_{js}^{history}$ και εμπεριέχει όλη την πληροφορία των κωδικοποιημένων καταστάσεων. Επιπλέον, δημιουργείται ένας αριθμός p_{gen} (στη βιβλιογραφία αναφέρεται ως generation probability) ο οποίος υπολογίζεται στη σχέση 5.4 κι εξαρτάται από την εκάστοτε είσοδο w_{js} και κρυφή κατάσταση h_{js}^{dec} του αποκωδικοποιητή στο βήμα s, όπως επίσης και από το διάνυσμα συμφραζομένων c_{js}.

$$p_{js}^{gen} = Sigmoid(W \cdot [h_{js}^{dec}; w_{js}; c_{js}] \in (0, 1) \tag{5.4}$$

Αυτός ο αριθμός χρησιμοποιείται για να αποδόσει βάρος ανάμεσα στην επιλογή μιας λέξης από το λεξιλόγιο ή από το κείμενο εισόδου του κωδικοποιητή όπως αναφέρεται και στο κεφάλαιο 4.5. Επειδή οι δυο κατανομές πιθανοτήτων $P^{vocab} \in \mathbb{R}^{|V|}$ και $P^{history} \in \mathbb{R}^{|X_t|}$ έχουν διαφορετικό μέγεθος, χρησιμοποιείται η συνάρτηση $Scatter : |X_t| \rightarrow |V|$ προκειμένου κάθε $\alpha_i \in P^{history}$ να αντιστοιχηθεί στο σωστό δείκτη k του λεξιλογίου βάσει της λέξης w_i που είναι η είσοδος του κωδικοποιητή τη στιγμη i.

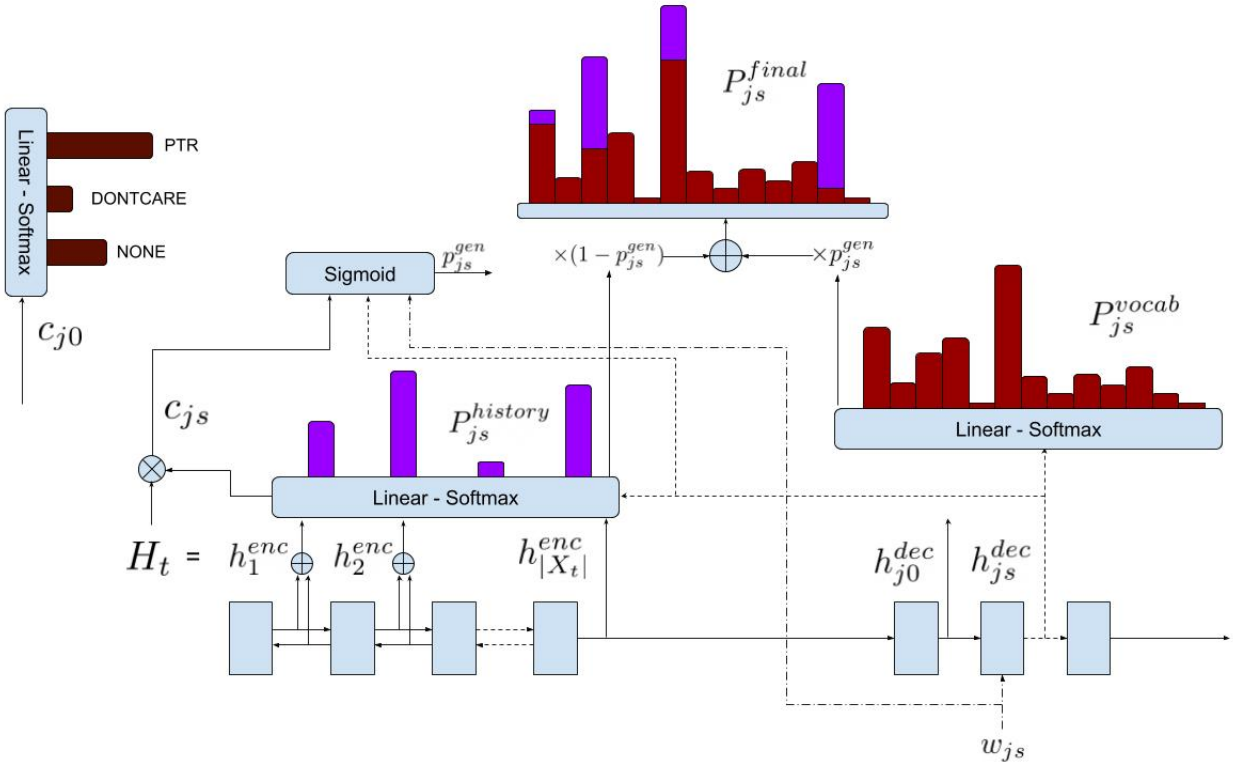
$$P_{new}^{history}[k] = \begin{cases} P_{old}^{history}[i], \text{if vocab[k]} = w_i, \text{for i } \in \{0, .., |X_t|\}, \text{for k } \in \{0, .., |V|\} \\ 0, otherwise \end{cases} \tag{5.5}$$

Στην περίπτωση όπου μια λέξη w_i είναι εκτός του λεξιλογίου (OOV) τότε ο δείκτης $k = 0$, δηλαδή αντιστοιχίζεται σε μια προκαθορισμένη θέση στο λεξιλόγιο στην οποία ανήκει η ειδική λέξη "<UNK>". Πλέον, οι δυο κατανομές πιθανοτήτων $P^{vocab} \in \mathbb{R}^{|V|}$ και $P^{history} \in \mathbb{R}^{|V|}$ έχουν το ίδιο μέγεθος και μπορεί να οριστεί η τελική κατανομή πιθανότητας P^{final} συναρτήσει του αριθμού p_{gen} όπως φαίνεται στη σχέση 5.6

$$P_{js}^{final} = p_{js}^{gen} \times P_{js}^{vocab} + (1 - p_{js}^{gen}) \times P_{js}^{history} \in \mathbb{R}^{|V|} \tag{5.6}$$

Χρησιμοποιώντας την συνολική κατανομή πιθανότητας P_{js}^{final} από τη σχέση 5.6 μπορεί να υπολογιστεί η πρόβλεψη για την επόμενη λέξη στο βήμα s του αποκωδικοποιητή για το

$domain - slot_j$, επιλέγοντας τον αντίστοιχο δείκτη της λέξης με την μεγαλύτερη πιθανότητα (argmax). Αυτή η λέξη θα αποτελέσει την επόμενη είσοδο $w_{j,s+1}$ του αποκωδικοποιητή στο βήμα $s+1$ και όλη η διαδικασία επαναλαμβάνεται. Η επανάληψη της διαδικασίας είναι απαραίτητη καθώς κάποια τιμή μπορεί να αποτελείται από περισσότερες λέξεις, όπως το όνομα ενός ξενοδοχείου. Η διαδικασία αποκωδικοποίησης σταματάει όταν ο αποκωδικοποιητής προβλέψει στην έξοδό του τη λέξη "<EOS>" η οποία υποδηλώνει το τέλος της πρότασης ή όταν ξεπεραστούν 10 βήματα. Η επιλογή του αριθμού 10 είναι αυθαίρετη. Οι περισσότερες τιμές domain-slot values δεν ξεπερνούν το πλήθος των 4 λέξεων, επομένως εάν δεν έχει παραχθεί η λέξη "<EOS>" εντός 10 επαναλήψεων τότε η αποκωδικοποίηση τερματίζεται.



Σχήμα 5.1: Αρχιτεκτονική μοντέλου

5.2.3 Πύλη Πεδίου Slot Gate

Η πύλη πεδίου (slot gate) είναι ένας βοηθητικός ταξινομητής ο οποίος προβλέπει βάσει του διανύσματος συμφραζομένων (context vector) c_{j0} που παράγεται στο πρώτο βήμα αποκωδικοποίησης ($s=0$) εάν το αντίστοιχο domain-slot$_j$ θα πρέπει να ενεργοποιηθεί ή όχι. Οι κλάσεις που προβλέπει είναι τρεις:

- **PTR**: Αυτή η κλάση αντιπροσωπεύει ότι το domain-slot$_j$ ενεργοποιείται από το διάνυσμα συμφραζομένων (context vector) επομένως η τιμή που παράγει ο αποκωδικοποιητής λαμβάνεται ως έγκυρη τιμή Y_j για το domain-slot$_j$.
- **DONTCARE**: Αυτή η κλάση αντιπροσωπεύει περιπτώσεις κατά τις οποίες ο ίδιος ο χρήστης δηλώνει ότι δεν ενδιαφέρεται για κάποια τιμή. Δηλαδή, το σύστημα μπορεί να προχωρήσει σε ερώτηση "What is your preferred pricerange for the hotel ?" και ο χρήστης να απαντήσει "I don't care as long as there is a free parking". Επομένως σε αυτή την περίπτωση δε μας απασχολεί η τιμή για το ζευγάρι domain-slot = hotel-price.
- **NONE**: Αυτή η κλάση αντιπροσωπεύει ότι το domain-slot$_j$ δεν αναφέρεται καθόλου εντός του διαλόγου επομένως παρόμοια με την κλάση DONTCARE η τιμή από τον αποκωδικοποιητή παραλείπεται.

Παρόμοια με την έξοδο του αποκωδικοποιητή, χρησιμοποιείται κι εδώ ένα Linear-Softmax δίκτυο ώστε από τον χώρο του context vector $c_{j0} \in \mathbb{R}^{d_{hdd}}$ να μεταφερθούμε στο χώρο των τριών κλάσεων $\mathbb{R}^3$. Χρησιμοποιείται δηλαδή ένας πίνακας βαρών $W_G \in \mathbb{R}^{3 \times d_{hdd}}$ και η συνάρτηση ενεργοποίησης $Softmax$. Η έξοδος / κατανομή πιθανότητας της πύλης πεδίου G_j για το $domain - slot_j$ υπολογίζεται από τη σχέση 5.7.

$$G_j = Softmax(W_G \cdot (c_{j0})^T) \in \mathbb{R}^3 \tag{5.7}$$

5.2.4 Συνάρτηση Κόστους Loss Function

Για τη διαδικασία της εκπαίδευσης ορίζονται οι συναρτήσεις κόστους για την πύλη πεδίου (slot gate) και για τον αποκωδικοποιητή decoder. Για την πύλη πεδίου ορίζεται ως συνάρτηση κόστους L_{gate} η διασταυρούμενη εντροπία cross entropy loss ανάμεσα στην προβλεπόμενη τιμή G_j^{target} και την πραγματική τιμή y_j^{gate} για όλα τα ζευγάρια domain-slot όπως φαίνεται στη σχέση 5.8.

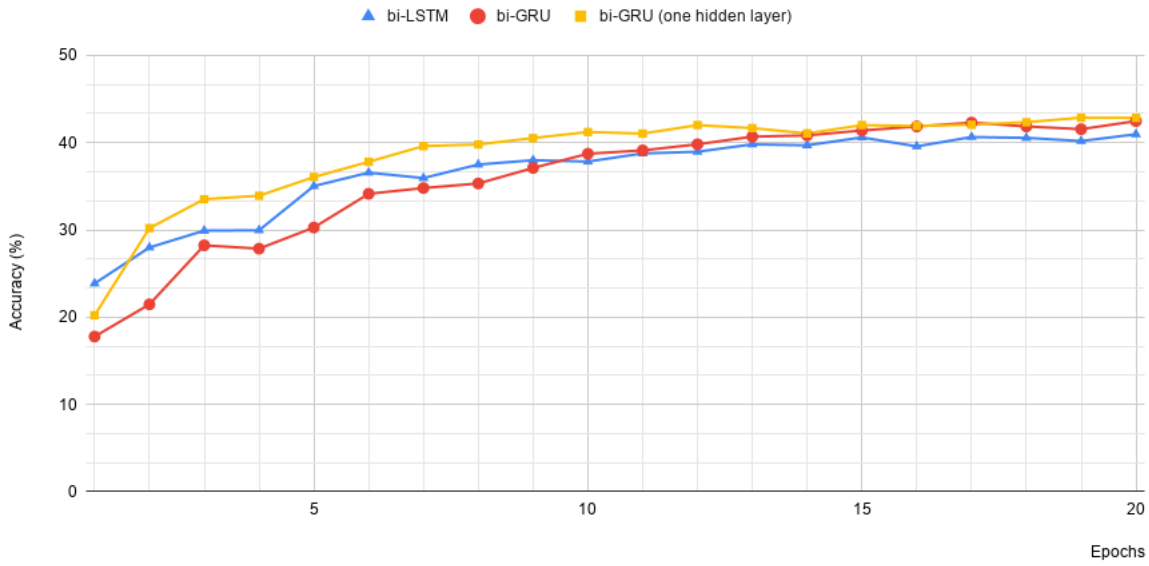
$$L_{gate} = \sum_{j=1}^{J} H(G_j, y_j^{target-gate}) = \sum_{j=1}^{J} -log(G_j \cdot (y_j^{target-gate})^T) \tag{5.8}$$

Για τον αποκωδικοποιητή ορίζεται ως συνάρτηση κόστους η L_{dec} ως η διασταυρούμενη εντροπία ανάμεσα στην τελική κατανομή πιθανότητας P_{js}^{final} που είναι η προβλεπόμενη τιμή και την πραγματική κατανομή $y_j^{target-word}$ υπολογισμένη για όλα τα ζευγάρια domain-slot και για όλα τα βήματα S (timesteps) του αποκωδικοποιητή. Η L_{dec} ορίζεται στη σχέση 5.9.

$$L_{dec} = \sum_{j=1}^{J} \sum_{s=1}^{S} -log(P_{js}^{final} \cdot (y_j^{target-word})^T) \tag{5.9}$$

Η συνολική συνάρτηση κόστους $Loss(avg)$ του μοντέλου ορίζεται ως το άθροισμα των δύο επιμέρους συναρτήσεων.

$$Loss_{avg} = L_{gate} + L_{dec} \tag{5.10}$$



Σχήμα 5.2: Accuracy of bi-LSTM, bi-GRU and bi-GRU (with one hidden layer) over 20 epochs

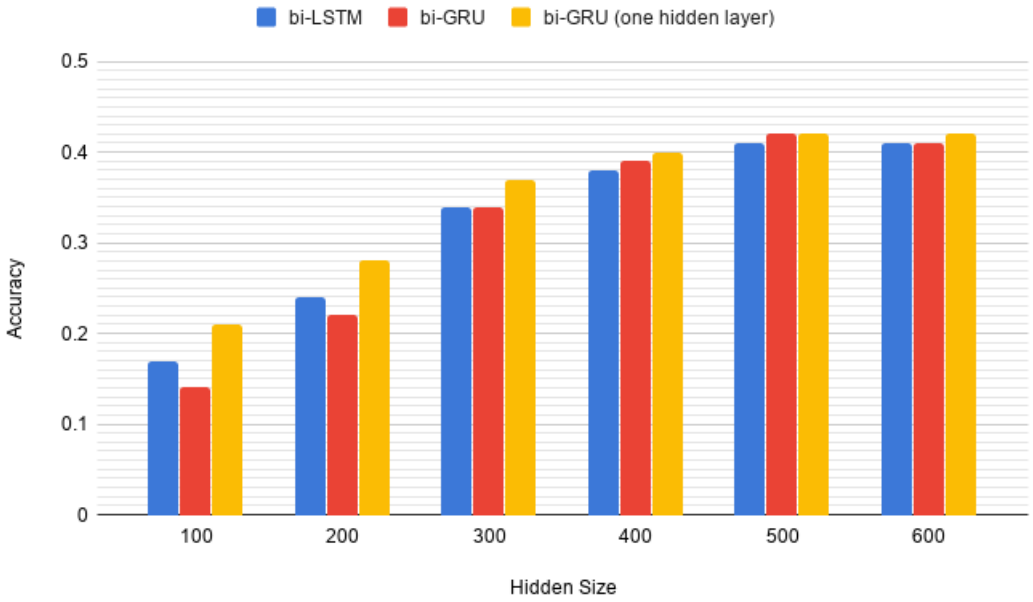
5.3 Πειραματική Διαδικασία

Σκοπός της πειραματικής διαδικασίας είναι η βαθύτερη κατανόηση των χαρακτηριστικών του μοντέλου και της αύξησης των μετρικών του απόδοσης. Οι παράμετροι του μοντέλου είναι αρκετές και κάποιες ίσως πολύ τεχνικές που ανήκουν στο κομμάτι της υλοποίησης (όπως η βιβλιοθήκη PyTorch και το hardware στην οποία αναπτύχθηκε και εκπαιδεύτηκε το μοντέλο). Οι παράμετροι που αναπτύσσονται στη συνέχεια είναι οι πιο βασικές και αυτές που επηρεάζουν σε μεγαλύτερο βαθμό την επίδοση του μοντέλου. Η ανάλυση αυτών των παραμέτρων παρέχει μια ουσιαστικότερη εποπτεία στην διαδικασία αξιολόγησης του μοντέλου.

Αρχικά, είναι σημαντικό να αποφασιστεί η βασική μονάδα επεξεργασίας του μοντέλου για τον κωδικοποιητή (encoder) και τον αποκωδικοποιητή (decoder), δηλαδή οι νευρώνες RNN, LSTM ή GRU. Η επιλογή αυτή είναι σημαντική καθώς κάθε νευρώνας έχει διαφορετική πολυπλοκότητα και επομένως επηρεάζει την επεκτασιμότητα του μοντέλου [12]. Στα πλαίσια της διπλωματικής χρησιμοποιήθηκαν LSTM, GRU και GRU με ένα επιπλέον επίπεδο (layer) καθώς αυτά χρησιμοποιούνται ευρέως στις ερευνητικές εργασίες. Για τον κωδικοποιητή χρησιμοποιούνται αμφίδρομα (bidirectional) δίκτυα ενώ για τον αποκωδικοποιητή χρησιμοποιούνται μονοκατευθυντικά (unidirectional).

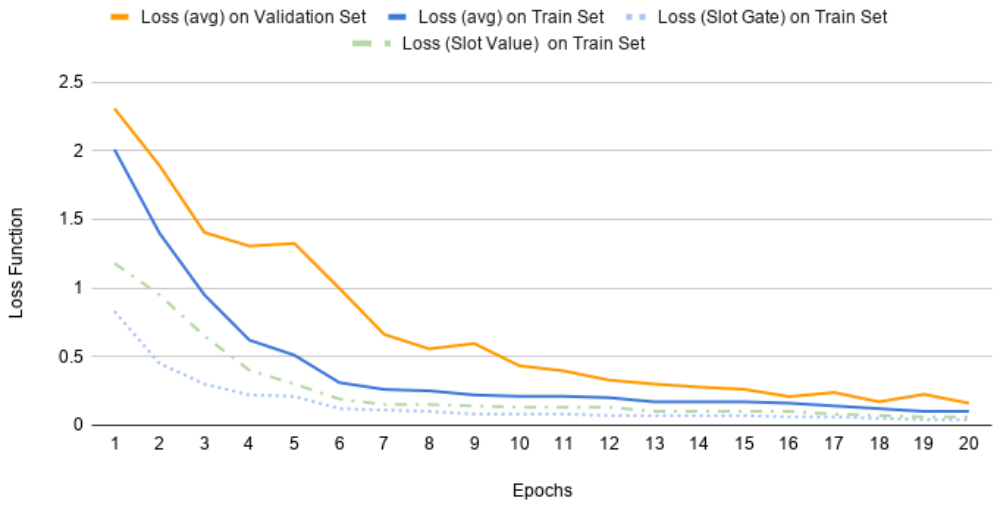
Στην εικόνα 5.2 φαίνεται η συμπεριφορά της ακρίβειας (Accuracy) των τριών δικτύων σε διάστημα 20 εποχών (epochs) υπολογισμένη στο validation set. Επίσης, στην εικόνα 5.3 φαίνεται η συμπεριφορά των τριών μοντέλων σε διαφορετικά μεγέθη της κρυφής κατάστασης. Παρατηρείται πως το bi-GRU (with one hidden layer) αποδίδει καλύτερα χωρίς όμως να αποκλίνει σημαντικά από τα άλλα δύο μοντέλα. Από τη στιγμη που η απόκλιση των άλλων δύο μοντέλων είναι αμελητέα, χρησιμοποιείται ως βασικό δίκτυο το απλό GRU καθώς είναι

λιγότερο πολύπλοκο από το LSTM και ταυτόχρονα έχει τις μισές απαιτήσεις στη μνήμη και μικρότερο χρόνο εκπαίδευσης από το GRU με ένα κρυφό επίπεδο [64]. Επιπλεόν, στην εικόνα 5.3 παρατηρούμε πως το μέγεθος της κρυφής κατάστασης επηρεάζει σημαντικά την ακρίβεια του μοντέλου. Αυτό είναι αναμενόμενο διότι κάθε λέξη ενσωματώνεται (word embedding) σε ένα διάνυσμα διάστασης 300 (embedding$(word) \in \mathbb{R}^{300}$) μέσω του GloVe. Επομένως, όσο μεγαλύτερη είναι η κρυφή κατάσταση τόσο μεγαλύτερη 'χωρητικότητα' [8] έχει το μοντέλο στην ενσωμάτωση της πληροφορίας μεγάλου μήκους ακολουθιών λέξεων στα διανύσματα κρυφών καταστάσεων.



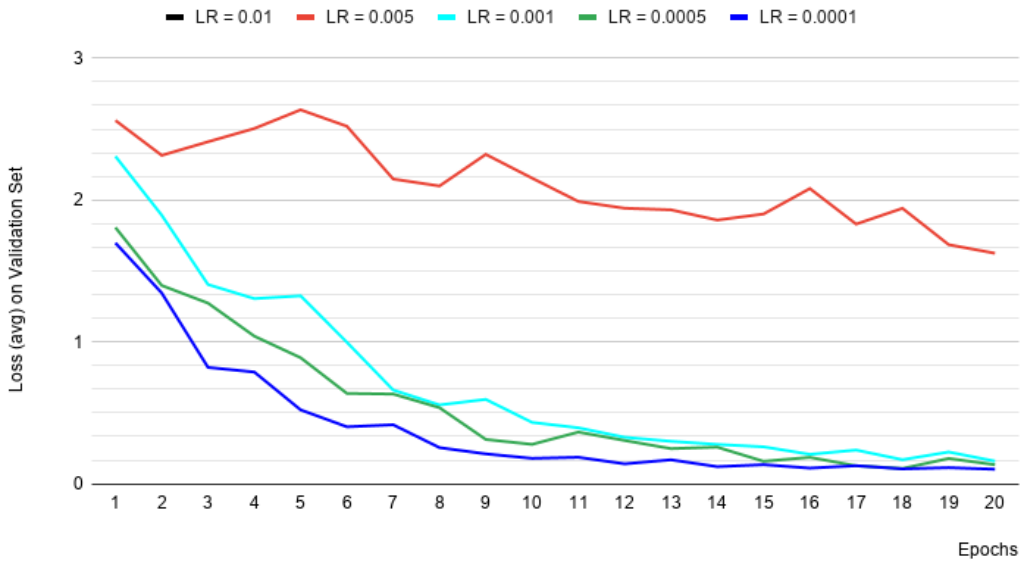
Σχήμα 5.3: Accuracies of bi-LSTM, bi-GRU and bi-GRU (with one hidden layer) over various Hidden Sizes

Επίσης είναι σημαντικό να κάνουμε επισκόπηση στη συμπεριφορά των συναρτήσεων κόστων που περιγράφονται στις σχέσεις 5.8 και 5.9 κατά τη διάρκεια εκπαίδευσης. Ορίζουμε ως συνολική συνάρτηση κόστους Loss (avg) το άθροισμα των δύο αυτών συναρτήσεων. Για τη διευκόλυνση του αναγνώστη αναφέρουμε πως οι δύο συναρτήσεις κόστους είναι η μέτρηση της διασταυρούμενης εντροπίας στην Πύλη Πεδίου (Slot Gate) και η μέτρηση της διασταυρούμενης εντροπίας στην τελική κατανομή πιθανότητας ορισμένη στο λεξιλόγιο και υπολογισμένη σε όλες τις παραγόμενες εξόδους του αποκωδικοποιητή για κάθε θεματική ενότητα-πεδίο (domain-slot) με τις πραγματικές τιμές (target values). Στην εικόνα 5.4 φαίνεται πως η συνάρτηση κόστους του validation set προσεγγίζει μετά από 20 εποχές αυτή του train set. Επομένως το μοντέλο δεν κάνει ούτε Underfit ούτε Overfit στα δεδομένα εκπαίδευσης.



Σχήμα 5.4: Loss Function on Train Set and Validation Set

Μια ακόμα σημαντική παράμετρος που επηρεάζει την σύγλιση του μοντέλου είναι ο ρυθμός μάθησης (Learning Rate). Ως αλγόριθμος βελτιστοποίησης χρησιμοποιήθηκε ο Adam Optimizer. Ο αλγόριθμος Adam εκτός από την παράμετρο Learning Rate, χρησιμοποιεί επιπλέον δύο παραμέτρους b_1, b_2 οι οποίες είναι συντελεστές για τα μεγέθη των τρέχων μέσων όρων και των τετραγώνων των βαρών του δικτύου. Αυτές οι παράμετροι παρέμειναν σταθερές στις προεπιλεγμένες τιμές της βιβλιοθήκης PyTorch, και οι τιμές τους ορίστηκαν ως $b_1 = 0.9, b_2 = 0.999$. Ο ρυθμός μάθησης χρησιμοποιεί τιμές στο διάστημα $[0.01, 0.0001]$. Στην εικόνα 5.5 φαίνεται η μείωση της Loss Function στο Validation Set σε διάστημα 20 εποχών για διαφορετικά μεγέθη ρυθμού μάθησης. Παρατηρούμε ότι για $lr = 0.01$ το μοντέλο αποτυχγάνει και δεν συγκλίνει επομένως δεν συμπεριλήφθηκε το αντίστοιχο διάγραμμα στην εικόνα.



Σχήμα 5.5: Loss Function on Validation Set computed on various Learning Rates (LR)

Επιπλέον, μια απαραίτητη τεχνική είναι να επιτρέψουμε στο μοντέλο κατά τη διάρκεια της εκπαίδευσης να εκπαιδεύσει τα διανύσματα των λέξεων (Word Embeddings). Συνήθως τα προ-εκπαιδευμένα (pre-trained) διανύσματα αποδίδουν καλύτερα από την αρχική δημιουργία τέτοιων διανυσμάτων χρησιμοποιώντας απλά το Train Set καθώς έχουν εκπαιδευτεί σε πολύ μεγαλύτερου όγκου κείμενα. Ωστόσο, η αρχικοποίηση των βαρών με τις προ-εκπαιδευμένες τιμές και η επίτρεψη επιπλέον εκπαίδευσης (fine-tuning) βοηθάει ακόμα περισσότερο το μοντέλο όπως φαίνεται και στον πίνακα 5.3. Αυτό οφείλεται στο γεγονός ότι ενώ λέξεις οι οποίες σε κείμενα γενικού περιεχομένου είναι ουδέτερες μεταξύ τους (πχ. “name”, “ashley”), σε διαλόγους με θεματική ενότητα το Ξενοδοχείο (hotel) αυτές οι δύο λέξεις έχουν περισσότερη συσχέτιση καθώς η λέξη ashley εμφανίζεται ως όνομα (name) ξενοδοχείου. Στον πίνακα 5.4 φαίνεται η μετρική Cosine Similarity για ζευγάρια λέξεων πριν και μετά την εκπαίδευση του μοντέλου. Το πιο ενδιαφέρον είναι το ζευγάρι king, queen το οποίο χάνει την προηγούμενη συσχέτιση που είχε καθώς η λέξη king αποτελούσε το όνομα ενός σταθμού τραίνου (London King ’s Cross) και χρησιμοποιούταν ευρύτερα ως προορισμός ή σημείο αναχώρησης.

	Fixed Embeddings	Trainable Embeddings
Accuracy % (on Validation Set)	28.75	39.44

Table 5.3: Fixed vs Trainable Embeddings

Word Pair	Before Training	After Training
ashley, name	0.1244	**0.2097**
price, cheap	0.5762	**0.6270**
hotel, stars	0.2649	**0.3092**
hotel, internet	0.3156	**0.3740**
king, queen	**0.7252**	0.3535

Table 5.4: Comparison of Cosine Similarity on Word Pairs Before and After Training

Ολοκληρώνοντας την πειρματική διαδικασία, μια τελευταία παράμετρος η οποία επηρεάζει το μοντέλο τόσο κατά τη διάρκεια της εκπαίδευσης όσο και στην επίδοσή του είναι η συχνότητα εφαρμογής της μεθόδου Teacher Forcing. Όπως αναφέρθηκε και στο κεφάλαιο 4.6 είναι σημαντικό κατά την εκπαίδευση να βοηθήσουμε το μοντέλο τροφοδοτώντας τον αποκωδικοποιητή κάποιες φορές με την ίδια του την έξοδο από το προηγούμενο βήμα και κάποιες φορές με την λέξη που έπρεπε να είχε προβλέψει. Αυτό πρέπει να συμβαίνει σε έναν ισορροπημένο βαθμό, επομένως ορίζεται ένας επιλογέας που αποφασίζει πιθανοτικά την επόμενη είσοδο στον αποκωδικοποιητή, είτε από την προηγούμενή του έξοδο, είτε από την λέξη που έπρεπε να προβλέψει. Στον πίνακα 5.5 φαίνεται πως η καταλληλότερη επιλογή είναι κοντά στο 40%, δηλαδή από τις 100 φορές, τις 40 ο αποκωδικοποιητής τροφοδοτείται με τη σωστή λέξη, ενώ τις 60 με τη λέξη που προέβλεψε ο ίδιος στο προηγούμενο βήμα. Παρ΄ ότι το μοντέλο μετα από 20 εποχές συγκλίνει πολύ πιο γρήγορα όσο αυξάνουμε το teacher forcing sample ratio, η ακρίβεια του μοντέλου μειώνεται στο validation set διότι το μοντέλο εξαναγκάζεται περισσότερο, παρά βελτιώνει τις εσωτερικές του παραμέτρους.

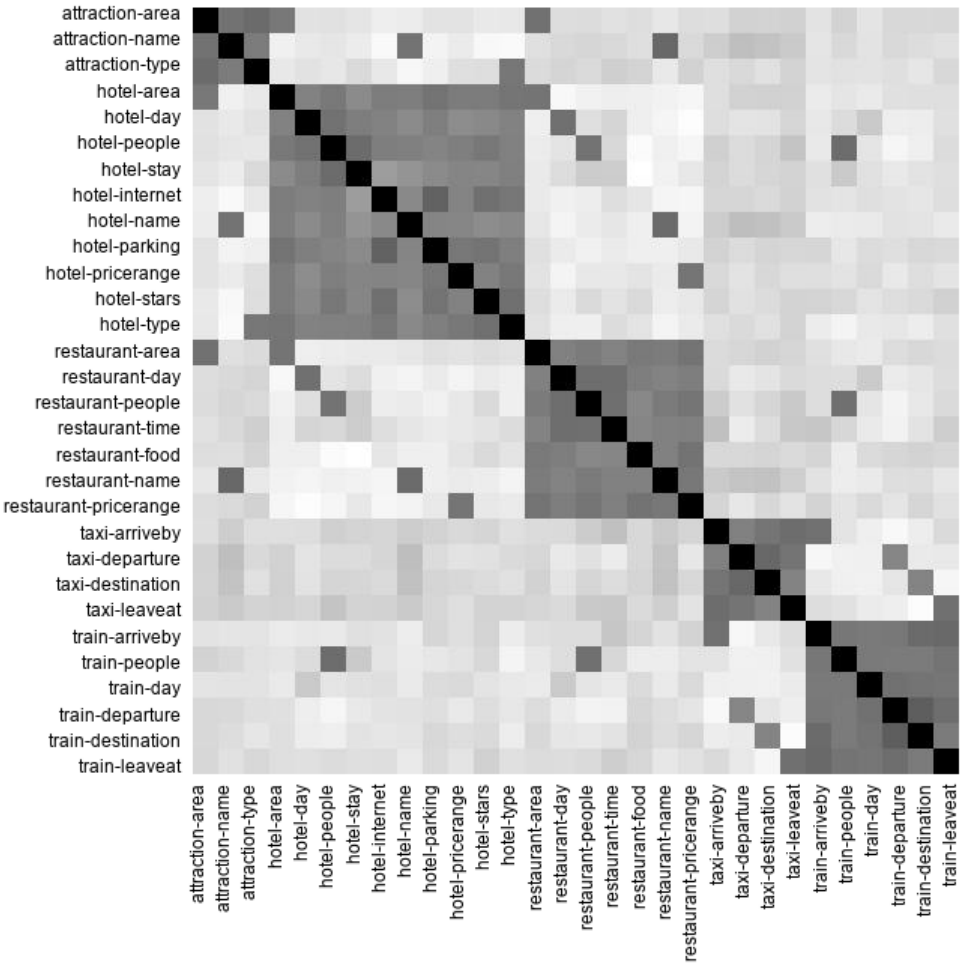
Teacher Forcing Sampling Ratio	Train Loss	Val Loss	Accuracy on Validation Set
20%	0.51	0.64	0.23
40%	**0.11**	**0.15**	**0.38**
60%	0.06	0.18	0.34
80%	0.03	1.29	0.12

Table 5.5: Accuracy and Loss in various Teacher Forcing Sampling Ratios

5.4 Ανάλυση Αποτελεσμάτων

Σε αυτό το κεφάλαιο θα αναλυθούν κάποια βασικά αποτελέσματα κυρίως ως προς την αστοχία του μοντέλου. Αυτό είναι βασικό γιατί προσφέρει μια ευρύτερη διορατικότητα σε πιθανές σχεδιαστικές τροποποιήσεις.

Στην εικόνα 5.6 παρουσιάζουμε τη ομοιότητα συνημιτόνου (cosine similarity) μεταξύ όλων των ζευγών θεματικής ενότητας - πεδίου (domain-slot). Σε αυτήν παρατηρούμε πέντε βασικά τετράγωνα όσα δηλαδή και τα domains (“Attraction”,“Hotel”,“Restaurant”,“Taxi”,“Train”).

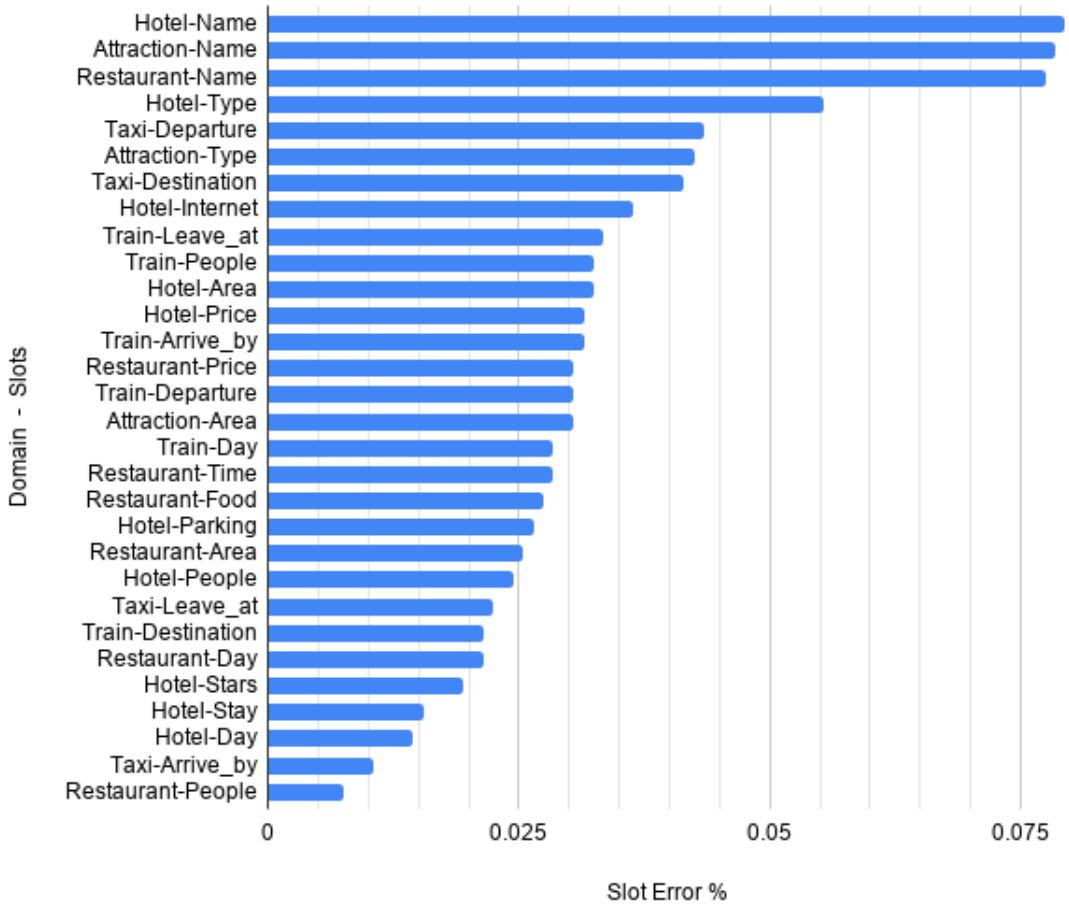


Σχήμα 5.6: Visualization of Cosine Similarity over all possible domain-slot pairs

Αυτό είναι λογικό καθώς στον αποκωδικοποιητή η πρώτη είσοδος που χρησιμοποιείται για την εκκίνηση της πρόβλεψης είναι το άθροισμα των διανυσμάτων (embeddings) domain και slot. Επομένως όλα τα slots που ανήκουν στην ίδια θεματική ενότητα μοιράζονται στο διάνυσμά τους και το διάνυσμα της θεματικής ενότητας. Επιπλέον, τα πιο αραιά γκρι τετραγωνάκια είναι η συσχέτιση μεταξύ πεδίων που μοιράζονται παρόμοιες τιμές. Για παράδειγμα τα ζεύγη (restaurant-name, attraction-name), (taxi-departure, train-departure) και (restaurant-people, hotel-people) έχουν παρόμοια οντολογία και εμφανίζουν μεγαλύτερη συσχέτιση απ΄ ότι για παράδειγμα το ζεύγος (restaurant-pricerange, hotel-area).

Στη συνέχεια, παρουσιάζουμε στην εικόνα 5.7 για κάθε ζεύγος domain-slot το ποσοστό σφάλματος.

Παρατηρούμε ότι στις πρώτες θέσεις βρίσκονται όσα ζεύγη περιέχουν το slot name. Αυτό συμβαίνει διότι το όνομα των ξενοδοχείων, των αξιοθέατων και των εστιατορίων είναι μεγαλύτερο της μίας λέξης με αποτέλεσμα ο αποκωδικοποιητής να μην το παράγει ολοκληρωμένο ή να παράγει περισσότερες λέξεις απ΄ όσες θα έπρεπε. Αντιθέτως, στις τελευταίες θέσεις βρίσκονται τα ζεύγη Hotel-Day, Taxi-Arrive by και Restaurant-People των οποίων το πεδίο

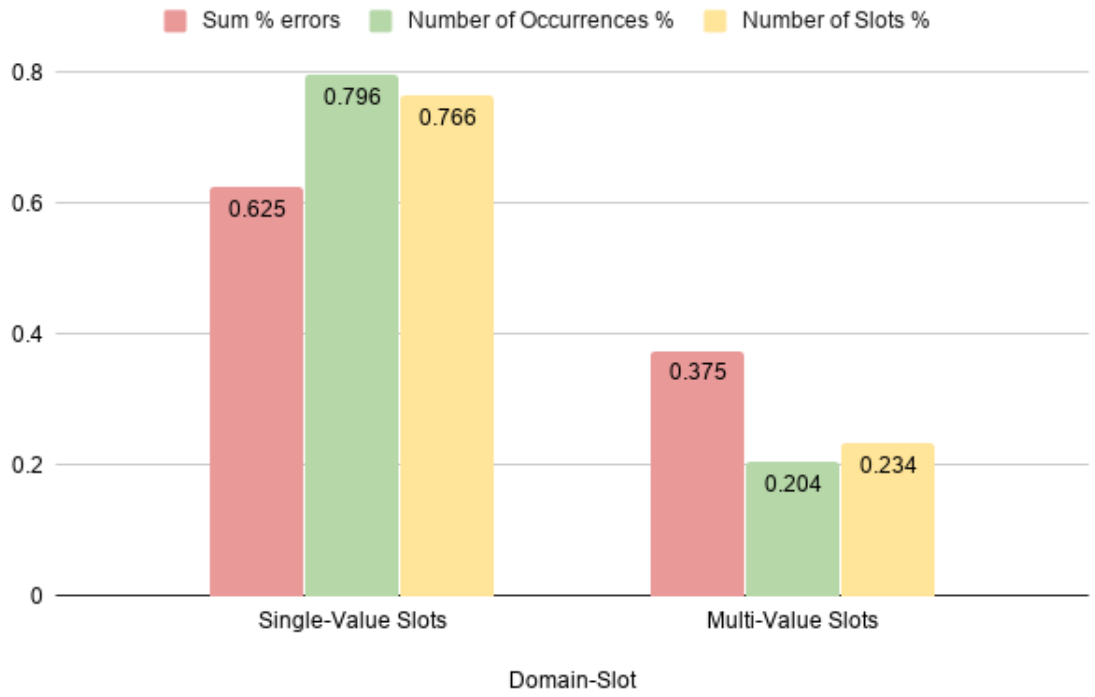


Σχήμα 5.7: Slot Error Rate

τιμών (οντολογία) είναι περιορισμένο και εύκολα προβλέψιμο.

Επίσης, στην εικόνα 5.8 παρουσιάζουμε μια γενικότερη ιδέα του προβλήματος των slots με τιμή μεγαλύτερη της μίας λέξης. Συγκρίνοντας τα single-value slots με τα multi-value slots παρατηρούμε ότι ενώ τα single-value αποτελούν το 76.6% όλων των slots (23/30) και εμφανίζονται σε παρόμοιο ποσοστό στο test set (79.6%) οφείλονται μόνο για το 62.5% των συνολικών σφαλμάτων το οποίο είναι δυσανάλογο. Ουσιαστικά το 37.5% των συνολικών σφαλμάτων οφείλεται στα multi-value slots τα οποία συμμετέχουν μόνο κατά 20.4% (1/5) στο test set.

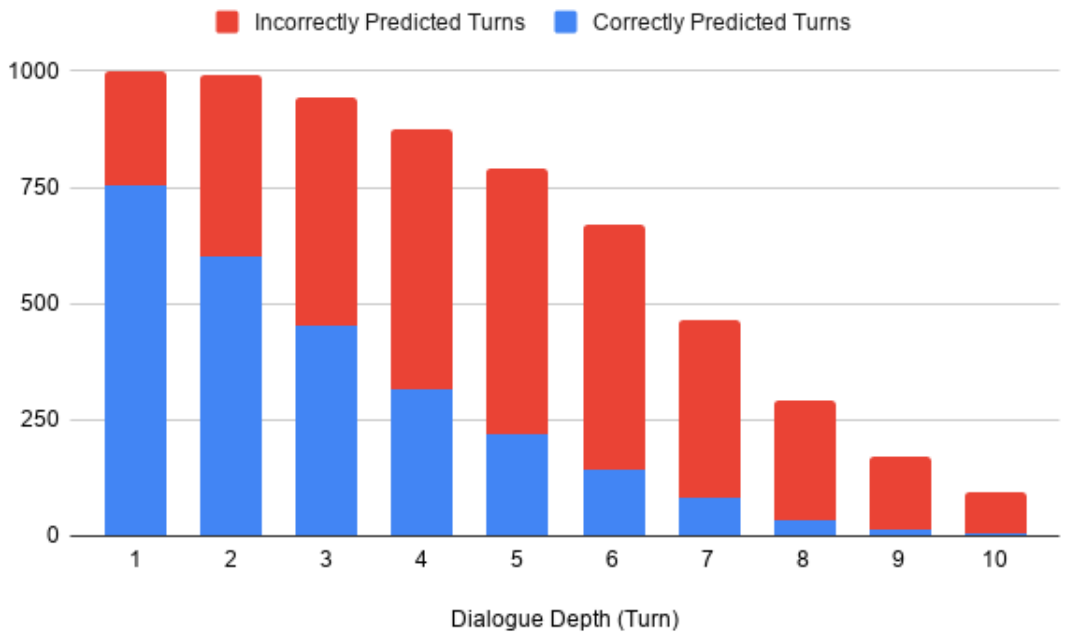
Τέλος, μια σημαντική ανάλυση είναι η επίδοση του μοντέλου καθώς το βάθος του διαλόγου ή αλλιώς το πλήθος των στροφών αυξάνεται. Στον πίνακα 5.6 και στην εικόνα 5.9 παρουσιάζονται τα ποσοστά αποτυχίας του μοντέλου για βάθος διαλόγου από 1 έως 10 στροφές. Παρατηρούμε ότι ήδη από την τρίτη στροφή το μοντέλο δυσκολεύεται και τα ποσοστά επιτυχίας είναι μικρότερα από 50%.



Σχήμα 5.8: Error Rate Between Single and Multi Value Slots

Dialogue Depth (# Turns)	Total Turns	Incorrect Turns %
1	1000	24.56
2	990	39.42
3	943	52.11
4	873	63.80
5	791	72.35
6	670	78.88
7	465	82.59
8	292	88.23
9	171	91.47
10	95	95.92

Table 5.6: Rates of Incorrectly Predicted Turns over Dialogue Depth



Σχήμα 5.9: Correct vs Incorrect Turns over Dialogue Depth

Error Category	Target	Prediction	Dialogue	%
Missing Slot (FP)	NONE	PTR	...	17.4
Missing Slot (FN)	PTR	NONE	...	56.2
Incorrect Resolution	'friday'	'thursday'	– "book it for 3 people and 5 nights starting from thursday ." – "your booking was unsuccessful . could you book another day or a shorter stay ?" – "let s try for **friday** instead . and i need the reference number please ."	22.1
Incorrect Boundary	'warkworth house'	'warkworth'	– "i am trying to find a hotel called **warkworth house**"	4.3

Table 5.7: Error Categorization with examples and Error Rates per Category

Επιπλέον, στον πίνακα 5.7 φαίνονται οι κατηγορίες που ανήκουν οι λανθασμένες προβλέψεις με αντίστοιχα παραδείγματα και ποσοστά συμμετοχής επί των συνολικών σφαλμάτων. Οι δύο βασικότερες είναι οι αστοχίες της Πύλης Πεδίου Slot Gate η οποία μαντεύει λάθος αν κάποιο Domain-Slot πρέπει να ενεργοποιηθεί μέσα από το Context του διαλόγου. Δυστυχώς

σε πολλά παραδείγματα το μοντέλο προβλέπει την ύπαρξη ενός Domain-Slot στην επόμενη στροφή turn από αυτήν που πρωτοεμφανίζεται με αποτέλεσμα να χάνει αρκετή ακρίβεια. Ουσιαστικά τα Slot mispredictions οφείλονται για το 73.6% (17.4% + 56.2%) των συνολικών σφαλμάτων με σημαντικότερη επίδραση την περίπτωση όπου το μοντέλο προβλέπει None για ένα Slot ενώ κανονικά θα έπρεπε να υπολογίσει κάποια τιμή (56.2%). Επιπλέον, το μοντέλο σε ορισμένες περιπτώσεις προβλέπει λάθος τιμή (Incorrect Resolution) η οποία έχει αναφερθεί στο διάλογο αλλά πρόκειται για μια ακυρωμένη κράτηση. Το μοντέλο δηλαδή δυσκολεύεται να προβλέψει την καινούρια τιμή που πρέπει να λάβει το αντίστοιχο domain-slot. Τέλος, ένα πολύ μικρό ποσοστό σφάλματος προκύπτει από λάθος όριο (Incorrect Boundary) στην τιμή του αποκωδικοποιητή όπως φαίνεται στο τελευταίο παράδειγμα του πίνακα.

Confusion Matrix		TARGET		PRECISION	RECALL	F1 SCORE
		PTR	NONE			
PREDICTION	PTR	17.1 %	1.8 %	90.47%	74.67%	81.81%
	NONE	**5.8%**	75.3 %			

Table 5.8: Confusion Matrix, Precision, Recall and F1 Score of Slot Gate

5.5 Παρεμφερείς Εργασίες

Σε αυτό το σημείο θα περιγράψουμε συνοπτικώς την αρχιτεκτονική μοντέλων που έχουν ακολουθήσει παρεμφερείς εργασίες στο Dialog State Tracking πάνω στο Multi-WOZ Dataset.

- **MDBT** [52] Η αρχιτεκτονική του MDBT χρησιμοποιεί ανεξάρτητους κωδικοποιητές για την έκφραση utterance του χρήστη User και του συστήματος System. Επιπλέον, στηρίζεται σε προκαθορισμένη οντολογία και για κάθε μια εξάξει ένα σκορ για το αν 'ενεργοποιείται' από το διάλογο ή όχι. Αν ενεργοποιείται τότε η τιμή της οντολογίας είναι και η τελική πρόβλεψη.

- **GLAD** [73] Η αρχιτεκτονική του GLAD χρησιμοποιεί έναν Global-locally self-attentive κωδικοποιητή. Για κάθε οντολογία το σύστημα δέχεται την τρέχουσα έκφραση από το χρήστη και τις προηγούμενες ενέργειες του συστήματος και αποφασίζει αν ενεργοποιείται κάποιο domain-slot. Αναλυτικότερα, για κάθε slot το σύστημα μαθαίνει να παράγει ένα διάνυσμα συμφραζομένων από την είσοδο του χρήστη και των προηγούμενων ενεργείων και παράγει ένα σκορ που εκφράζει την πιθανότητα να αναφέρεται στο διάλογο.

- **GCE** [45] Η αρχιτεκτονική του GCE είναι παρόμοια με αυτή του GLAD. Ο προτεινόμενος κωδικοποιητής που χρησιμοποιείται βασίζεται στη βελτίωση της καθυστέρησης και της ταχύτητας του μοντέλου κατά την τελική χρήση με την αφαίρεση αναποτελεσματικών rnn layers και μηχανισμών προσοχής, χωρίς να υποβαθμίζεται η απόδοση.

- **Neural Reading** [17] Η αρχιτεκτονική του Neural Reading μοντέλου είναι παρόμοια με αυτή της διπλωματικής καθώς χρησιμοποιεί pointer - generator μηχανισμό για την αντιγραφή copy λέξεων από το κείμενο εισόδου. Η διαφορά είναι ότι χρησιμοποιούνται δύο δείκτες, ένας "Start" κι ένας "End" που δηλώνουν την αρχή και το τέλος της απάντησης.

- **HyST** [18] Το μοντέλο Hyst σε κάθε στροφή παράγει ένα ανοιχτό σύνολο από υποψήφιες τιμές για κάθε domain-slot οι οποίες προκύπτουν από την τρέχουσα έκφραση και την ιστορία του διαλόγου. Η επιλογή της τελικής απάντησης από τις υποψήφιες τιμές εξαρτάται από το μήκος της απάντησης (μία ή περισσότερες λέξεις) για το κάθε domain-slot όπως επίσης και από το εάν αυτό ενεργοποιήθηκε από τα συμφραζόμενα του διαλόγου.

- **SUMBT** [31] Η αρχιτεκτονική του SUMBT (slot-utterance matching belief tracker μαθαίνει σχέσεις ανάμεσα στα διάφορα domains και slots και ανάμεσα στα slots και τις τιμές values. Χρησιμοποιεί τρεις κωδικοποιητές για την έκφραση του χρήστη user utterance, το αντίστοιχο domain-slot και την τελική τιμή target value. Ο αποκωδικοποιητής μαθαίνει από το διάνυσμα συμφραζομένων που εξαρτάται από το domain-slot και το διάλογο να παράγει την πραγματική τιμή.

- **TRADE** [69] Η αρχιτεκτονική του TRADE είναι και η αρχιτεκτονική που ακολουθήθηκε στην παρούσα διπλωματική. Συνδυάζει την ιδέα του Pointer - Generator [54] η οποία χρησιμοποιείται και σε τεχνικές περίληψης κειμένου text summarization κι επίσης την ιδέα του Slot Gate ως επιλογέα ενεργοποίησης για κάθε domain-slot.

Model	Joint Accuracy %	Slot Accuracy %
MDBT	15.57	89.53
GLAD	35.57	95.44
GCE	36.27	**98.42**
Neural Reading	41.10	-
HyST	44.24	-
SUMBT	46.649	96.44
TRADE	**48.62**	96.92
OUR MODEL	41.13	96.07

Table 5.9: Comparison of Joint and Slot accuracies between ours and related models.

Στον πίνακα 5.9 παρουσιάζονται οι μετρικές ακρίβειας του κάθε μοντέλου συμπεριλαμβανομένου και του δικού μας ως προς την συνολική ακρίβεια (Joint Accuracy) ως προς κάθε στροφή turn του διαλόγου και ανεξάρτητα ως προς τα Slots. Υπενθυμίζουμε ότι συνολική ακρίβεια σημαίνει επιτυχή πρόβλεψη σε όλο το σύνολο των domain-slots που αποτελούν την

κατάσταση του διαλόγου dialogue / belief state και συνοδεύουν κάθε στροφή turn. Η ακρίβεια για κάθε slot μετριέται ανεξάρτητα σε όλο το πλήθος των domain-slots. Το μοντέλο που αναπτύχθηκε στην παρούσα διπλωματική φαίνεται να έχει καλά αποτελέσματα ως προς τις άλλες εργασίες χωρίς όμως να πλησιάζει τόσο επιτυχώς τις μετρικές του TRADE στην αρχιτεκτονική του οποίου στηρίχθηκε. Αυτό μπορεί να οφείλεται κυρίως σε υπολογιστικούς πόρους καθώς στη βιβλιογραφία όσον αφορά το μέγεθος του batch size, δηλαδή το πλήθος των δειγμάτων που φορτώνονται στη μνήμη της GPU κατά τη διάρκεια της εκπαίδευσης, αναφέρονται ως συχνοί αριθμοί το 16, 32 και 64, ενώ κατά την εκπαίδευση του μοντέλου μας χρησιμοποιήθηκε batch size από 3 έως 8 για τις διάφορες παραμέτρους (learning rate, hidden size, number of layers) καθώς αυτό ήταν το διαθέσιμο όριο υποστήριξης της μνήμης. Αυτό είχε ως συνέπεια πιο αργή σύκλιση του μοντέλου και περισσότερο χρόνο ολοκλήρωσης μιας εποχής με αποτέλεσμα η συνολική διαδικασία της εκπαίδευσης να απαιτεί απαγορευτικούς χρόνους για κάθε πειραματικό γύρο.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σε αυτή τη διπλωματική παρουσιάσαμε την αρχιτεκτονική ενός μοντέλου που βασίζεται στη μέθοδο encoder - decoder για την κωδικοποίηση του διαλόγου και την παραγωγή τιμών για κάθε domain-slot με την προσθήκη μηχανισμού προσοχής attention-mechanism, μηχανισμού αντιγραφής Pointer - Generator και ενός επιλογέα απόφασης Slot Gate. Το σημαντικό χαρακτηριστικό του μοντέλου είναι ότι δε στηρίζεται σε μια προκαθορισμένη οντολογία τιμών για κάθε domain-slot όπως συμβαίνει στις δυο παρεμφερείς εργασίες MDBT και GLAD. Αυτό σημαίνει πως το μοντέλο μπορεί να παράξει λέξεις οι οποίες δεν υπάρχουν στο λεξιλόγιο Out of Vocabulary words καθώς εκτός της κατανομής πιθανότητας στο προκαθορισμένο λεξιλόγιο, το μοντέλο υπολογίζει μέσω του μηχανισμού προσοχής και μια κατανομή πάνω στο κωδικοποιημένο κείμενο εισόδου. Ο συνδυασμός αυτών των δύο κατανομών είναι και το κλειδί επιτυχίας της αρχιτεκτονικής. Επιπλέον, το Slot Gate λειτουργεί βοηθητικά στην απόφαση της ενεργοποίησης ή μη ενός domain-slot από τα συμφραζόμενα context του διαλόγου. Ωστόσο, το μοντέλο φαίνεται να δυσκολεύεται σε δυο σημαντικά σημεία. Το ένα είναι η επιτυχής πρόβλεψη τιμής μαγαλύτερης της μίας λέξης. Όπως φάνηκε και στο γράφημα 5.7, τα domain-slots με multi-value τιμές εμφανίζουν το μεγαλύτερο ποσοστό συμμετοχής στις αστοχίες του μοντέλου. Ουσιαστικά το μοντέλο παράγει μικρότερου μήκους απαντήσεις από τις επιθυμητές. Αυτό κατά πάσα πιθανότητα συμβαίνει διότι ο αποκωδικοποιητής είναι περισσότερο προκατειλλημένος στις απαντήσεις μιας λέξης καθώς τα single-value slots εμφανίζονται σε μεγαλύτερο ποσοστό στο train set. Το δεύτερο πρόβλημα είναι το Slot Gate που και αυτό αντίστοιχα όπως φαίνεται και στον πίνακα 5.8 εμφανίζει προκατάληψη ως προς την απόφαση της μη ενεργοποίησης κάποιου domain-slot. Αυτό είναι λογικό καθώς το μέσο μέγεθος του dialog-state σε κάθε στροφή είναι 5-6 domain-slots, ενώ τα στνολικά ζευγάρια είναι 30. Αυτό οδηγεί στην αστοχία πολλές φορές του μοντέλου να προβλέψει την ύπαρξη ενός domain-slot στη σωστή στροφή και να το προβλέπει στην επόμενη.

Αυτά τα δύο προβλήματα λειτουργούν συνδυαστικά. Αν για παράδειγμα αφαιρούσαμε πλήρως το Slot Gate για να αποφύγουμε την προκατάληψη όσον αφορά την ενεργοποίηση ενός υποψήφιου domain-slot και απλά χρησιμοποποιούσαμε τον αποκωδικοποιητή, τότε ο α-

ποκωδικοποιητής θα μάθαινε να παράγει σε πολλά δείγματα την ειδική λέξη "<EOS>", δηλαδή ότι δεν υπάρχει τιμή.

Μια πιθανή τροποποίηση είναι η επιπλέον χρήση ενός regression model πριν την αρχική είσοδο στον αποκωδικοποιητή το οποίο θα αποδίδει μια τιμή που θα εκφράζει το μήκος της υπολοιπόμενης απάντησης. Ουσιαστικά, θα χρειαζόταν ένα επιπλέον trainable-weight-matrix το οποίο θα είχε ως είσοδο ένα domain-slot$_j$ ($(embedding(domain_j) + embedding(slot_j)$ και θα παρήγαγε ένα σκορ $rest - length_j$. Αυτή η τιμή σκορ θα μπορούσε να χρησιμοποιηθεί συνδυαστικά με την τρέχουσα είσοδο του αποκωδικοποιητή η οποία με τη σειρά της θα επηρεάσει την πρόβλεψη της επόμενης λέξης. Πιο αναλυτικά, αντί να αρχικοποιείται ο αποκωδικοποιητής με την τιμή $embedding(domain) + embedding(slot)$ να αρχικοποιείται με την τιμή $(embedding(domain) + embedding(slot)); rest - length$, δηλαδή να επεκτείνεται το διάνυσμα εισόδου κατά μία τιμή η οποία εκφράζει και το υπόλοιπο αναμενόμενο μήκος. Με αυτόν τον τρόπο ο αποκωδικοποιητής θα είχε επιπλέον μια πληροφορία για την πρόβλεψη της επόμενης λέξης και επιπλέον σε κάθε domain-slot θα αντιστοιχούσε και μια αριθμητική τιμή που θα εξέφραζε το αναμενόμενο μήκος απάντησης γι΄ αυτό το domain-slot.

Μια επιπλέον προσθήκη που ενδεχομένως θα βοηθούσε στην αποφυγή προκατάληψης θα ήταν η χρήση δύο ή ακόμα και τριών ανεξάρτητων αποκωδικοποιητών (δηλαδή να μη μοιράζονται τις παραμέτρους τους) με τα αντίστοιχα Slot Gates οι οποίοι θα χρησιμοποιούνται ο καθένας σε διαφορετικά domain-slots ανάλογα του αναμενόμενου μήκους της απάντησης που εμφανίζεται συνήθως στο εκάστοτε domain-slot.

Μία ακόμη προτεινόμενη τροποποίηση όσον αφορά τον classifier Slot Gate είναι η αντικατάστασή του με τη χρήση μιας κατανομής πιθανότητας (ή απλά αριθμητικών σκορ) υπολογισμένη σε όλα τα domain-slots και την εκπαίδευση ενός βαθμωτού μεγέθους (scalar) που θα χρησιμοποιείται ως κατώφλι στην κατανομή για την επιλογή του συνόλου των domain-slots που πρέπει να ανήκουν στο εκάστοτε dialog-state. Το πλεονέκτημα αυτής της μεθόδου είναι ότι η εκπαίδευση ενός τέτοιου scalar μπορεί να γίνει ανεξάρτητα από τον αποκωδικοποιητή καθώς η είσοδός του αρκεί να είναι η τελική κωδικοποιημένη κατάσταση του κωδικοποιητή και το σκορ των domain-slots που ανήκουν στο dialog-state της προηγούμενης στροφής. Αυτή η επέκταση προσφέρει μια περισσότερο stateful αρχιτεκτονική καθώς η παραγωγή του επόμενου dialog-state εξαρτάται από το σκορ των domain-slots της προηγούμενης στροφής. Αντίστοιχα ο ίδιος μηχανισμός θα μπορούσε να χρησιμοποιηθεί για τη δημιουργία ενός δυαδικού classifier τύπου ("Skip", "Update"), όπου θα αποφάσιζε πιο στοχευμένα για κάθε domain-slot αν απλά παραμένει η τιμή του (συμπεριλαμβανομένης και της None) ως έχει στην επόμενη στροφή, ή πρέπει να ανανεωθεί (να δημιουργηθεί κάποια τιμή στην περίπτωση της None ή να αλλάξει μια ήδη υπάρχουσα τιμή σε None ή άλλη τιμή). Κατ΄ αυτόν τον τρόπο θα μειωνόταν και η χρήση του αποκωδικοποιητή μόνο στα ζευγάρια domain-slots που πρέπει να επανα-υπολογιστεί η τιμή τους. Στον πίνακα, 5.9 φαίνεται ότι όλα τα μοντέλα παρουσιάζουν πρόβλημα στο Joint Accuracy και όχι στο Slot Accuracy. Επομένως, αυτή η τροποποίηση θα έδινε παραπάνω σημασία στη σωστή δημιουργία του συνόλου domain-slots που απαρτίζουν το dialogue-state και ενδεχομένως να εμφάνιζε περισσότερες επιτυχίες στο Joint.

Κλείνοντας, ελπίζω κάποια στιγμή να εφαρμόσω τις προαναφερθείσες τροποποιήσεις κι

επίσης εύχομαι αυτή η διπλωματική να βοηθήσει στο μέλλον φοιτητές που έχουν σκοπό να ασχοληθούν με ένα παρόμοιο θέμα.

Βιβλιογραφία

[1] Mridul Agarwal and Vaneet Aggarwal. *A Reinforcement Learning Based Approach for Joint Multi-Agent Decision Making*. 2019. arXiv: 1909.02940 [cs.LG].

[2] Mehdi Allahyari et al. *Text Summarization Techniques: A Brief Survey*. 2017. arXiv: 1707.02268 [cs.CL].

[3] John S. Ball. *Using NLU in Context for Question Answering: Improving on Facebook's bAbI Tasks*. 2017. arXiv: 1709.04558 [cs.CL].

[4] Roberto Battiti. "First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method". In: *Neural Computation* 4 (Mar. 1992). DOI: 10.1162/neco.1992.4.2.141.

[5] Samy Bengio et al. *Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks*. 2015. arXiv: 1506.03099 [cs.LG].

[6] Robin Brochier, Adrien Guille, and Julien Velcin. "Global Vectors for Node Representations". In: *The World Wide Web Conference on - WWW '19* (2019). DOI: 10.1145/3308558.3313595. URL: http://dx.doi.org/10.1145/3308558.3313595.

[7] Paweł Budzianowski et al. *MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling*. 2018. arXiv: 1810.00278 [cs.CL].

[8] Christophe Cerisara, Pavel Král, and Ladislav Lenc. "On the Effects of Using word2vec Representations in Neural Networks for Dialogue Act Recognition". In: *Computer Speech Language* 47 (July 2017). DOI: 10.1016/j.csl.2017.07.009.

[9] Tongfei Chen et al. *Uncertain Natural Language Inference*. 2019. arXiv: 1909.03042 [cs.CL].

[10] Chung-Cheng Chiu et al. *State-of-the-art Speech Recognition With Sequence-to-Sequence Models*. 2017. arXiv: 1712.01769 [cs.CL].

[11] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[12] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[13] Angel Daza and Anette Frank. *Translate and Label! An Encoder-Decoder Approach for Cross-lingual Semantic Role Labeling*. 2019. arXiv: 1908.11326 [cs.CL].

[14] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. arXiv: 1810.04805 [cs.CL].

[15] Jeffrey L. Elman. "Finding Structure in Time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: 10.1207/s15516709cog1402_1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1. URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.

[16] Kawin Ethayarajh. *How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings*. 2019. arXiv: 1909.00512 [cs.CL].

[17] Shuyang Gao et al. *Dialog State Tracking: A Neural Reading Comprehension Approach*. 2019. arXiv: 1908.01946 [cs.CL].

[18] Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. *HyST: A Hybrid Approach for Flexible and Accurate Dialogue State Tracking*. 2019. arXiv: 1907.00883 [cs.CL].

[19] Yoav Goldberg and Omer Levy. *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. 2014. arXiv: 1402.3722 [cs.CL].

[20] Cyril Goutte and Éric Gaussier. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation". In: *ECIR*. 2005.

[21] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. *Speech Recognition with Deep Recurrent Neural Networks*. 2013. arXiv: 1303.5778 [cs.NE].

[22] Jiatao Gu et al. *Incorporating Copying Mechanism in Sequence-to-Sequence Learning*. 2016. arXiv: 1603.06393 [cs.CL].

[23] Dani Gunawan, C Sembiring, and Mohammad Budiman. "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents". In: *Journal of Physics: Conference Series* 978 (Mar. 2018), p. 012120. DOI: 10.1088/1742-6596/978/1/012120.

[24] Valerie Hajdik et al. *Neural Text Generation from Rich Semantic Representations*. 2019. arXiv: 1904.11564 [cs.CL].

[25] Sébastien Harispe et al. "Semantic Similarity from Natural Language and Ontology Analysis". In: *Synthesis Lectures on Human Language Technologies* 8.1 (May 2015), pp. 1–254. ISSN: 1947-4059. DOI: 10.2200/s00639ed1v01y201504hlt027. URL: http://dx.doi.org/10.2200/S00639ED1V01Y201504HLT027.

[26] Franziska Horn. *Context encoders as a simple but powerful extension of word2vec*. 2017. arXiv: 1706.02496 [stat.ML].

[27] Shaojie Jiang and Maarten de Rijke. "Why are Sequence-to-Sequence Models So Dull? Understanding the Low-Diversity Problem of Chatbots". In: *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 81–86. DOI: 10.18653/v1/W18-5712. URL: https://www.aclweb.org/anthology/W18-5712.

[28] Michael I. Jordan. "Chapter 25 - Serial Order: A Parallel Distributed Processing Approach". In: *Neural-Network Models of Cognition*. Ed. by John W. Donahoe and Vivian Packard Dorsel. Vol. 121. Advances in Psychology. North-Holland, 1997, pp. 471–495. DOI: https://doi.org/10.1016/S0166-4115(97)80111-2. URL: http://www.sciencedirect.com/science/article/pii/S0166411597801112.

[29] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[30] Alex Lamb et al. *Professor Forcing: A New Algorithm for Training Recurrent Networks*. 2016. arXiv: 1610.09038 [stat.ML].

[31] Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. "SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5478–5483. DOI: 10.18653/v1/P19-1546. URL: https://www.aclweb.org/anthology/P19-1546.

[32] Alexander Hanbo Li and Abhinav Sethy. *Knowledge Enhanced Attention for Robust Natural Language Inference*. 2019. arXiv: 1909.00102 [cs.CL].

[33] Bofang Li et al. "Weighted Neural Bag-of-n-grams Model: New Baselines for Text Classification". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 1591–1600. URL: https://www.aclweb.org/anthology/C16-1150.

[34] Xiujun Li et al. *End-to-End Task-Completion Neural Dialogue Systems*. 2017. arXiv: 1703.01008 [cs.CL].

[35] Qun Luo, Weiran Xu, and Jun Guo. "A Study on the CBOW Model's Overfitting and Stability". In: *International Conference on Information and Knowledge Management, Proceedings* 2014 (Nov. 2014), pp. 9–12. DOI: 10.1145/2663792.2663793.

[36] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. *Effective Approaches to Attention-based Neural Machine Translation*. 2015. arXiv: 1508.04025 [cs.CL].

[37] Agnes Lydia and Sagayaraj Francis. "Adagrad - An Optimizer for Stochastic Gradient Descent". In: (May 2019).

[38] Victor Makarenkov, Bracha Shapira, and Lior Rokach. *Language Models with Pre-Trained (GloVe) Word Embeddings*. 2016. arXiv: 1610.03759 [cs.CL].

[39] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: https://doi.org/10.1007/BF02478259.

[40] Tomas Mikolov et al. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv: 1310.4546 [cs.CL].

[41] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[42] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969.

[43] Thomas Mitchell. *Machine learning*. McGraw-Hill Pub. Co. (ISE Editions), 1997.

[44] Feiping Nie, Hu Zhanxuan, and Xuelong Li. "An investigation for loss functions widely used in machine learning". In: *Communications in Information and Systems* 18 (Jan. 2018), pp. 37–52. DOI: 10.4310/CIS.2018.v18.n1.a2.

[45] Elnaz Nouri and Ehsan Hosseini-Asl. *Toward Scalable Neural Dialogue State Tracking Model*. 2018. arXiv: 1812.00899 [cs.CL].

[46] Yannis Papanikolaou, Ian Roberts, and Andrea Pierleoni. *Deep Bidirectional Transformers for Relation Extraction without Supervision*. 2019. arXiv: 1911.00313 [cs.LG].

[47] Razvan Pascanu et al. *How to Construct Deep Recurrent Neural Networks*. 2013. arXiv: 1312.6026 [cs.NE].

[48] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[49] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: 1802.05365 [cs.CL].

[50] Shahzad Qaiser and Ramsha Ali. "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents". In: *International Journal of Computer Applications* 181 (July 2018). DOI: 10.5120/ijca2018917395.

[51] Zhenyue Qin and Dongwoo Kim. *Softmax Is Not an Artificial Trick: An Information-Theoretic View of Softmax in Neural Networks*. 2019. arXiv: 1910.02629 [cs.LG].

[52] Osman Ramadan, Paweł Budzianowski, and Milica Gašić. "Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 432–437. DOI: 10.18653/v1/P18-2069. URL: https://www.aclweb.org/anthology/P18-2069.

[53] Mike Schuster and Kuldip K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Trans. Signal Processing* 45 (1997), pp. 2673–2681.

[54] Abigail See, Peter J. Liu, and Christopher D. Manning. *Get To The Point: Summarization with Pointer-Generator Networks*. 2017. arXiv: 1704.04368 [cs.CL].

[55] Darsh J Shah et al. *Robust Zero-Shot Cross-Domain Slot Filling with Example Values*. 2019. arXiv: 1906.06870 [cs.CL].

[56] Alex Sherstinsky. *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. 2018. arXiv: 1808.03314 [cs.LG].

[57] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. arXiv: 1712.01815 [cs.AI].

[58] Pinky Sitikhu et al. *A Comparison of Semantic Similarity Methods for Maximum Human Interpretability*. 2019. arXiv: 1910.09129 [cs.IR].

[59] Daouda Sow et al. *A sequential guiding network with attention for image captioning*. 2018. arXiv: 1811.00228 [cs.CV].

[60] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

[61] Amir Vakili Tahami and Azadeh Shakery. *Enriching Conversation Context in Retrieval-based Chatbots*. 2019. arXiv: 1911.02290 [cs.CL].

[62] T. Thireou and M. Reczko. "Bidirectional Long Short-Term Memory Networks for Predicting the Subcellular Localization of Eukaryotic Proteins". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 4.3 (July 2007), pp. 441–446. DOI: 10.1109/tcbb.2007.1015.

[63] N. Toomarian and J. Barhen. "Fast temporal neural learning using teacher forcing". In: *IJCNN-91-Seattle International Joint Conference on Neural Networks*. Vol. i. July 1991, 817–822 vol.1. DOI: 10.1109/IJCNN.1991.155284.

[64] Javier S. Turek et al. *A single-layer RNN can approximate stacked and bidirectional RNNs, and topologies in between*. 2019. arXiv: 1909.00021 [cs.LG].

[65] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[66] Amir Pouran Ben Veyseh, Franck Dernonrcourt, and Thien Huu Nguyen. *Improving Slot Filling by Utilizing Contextual Information*. 2019. arXiv: 1911.01680 [cs.CL].

[67] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. *Pointer Networks*. 2015. arXiv: 1506.03134 [stat.ML].

[68] Chien-Sheng Wu. *Learning to Memorize in Neural Task-Oriented Dialogue Systems*. 2019. arXiv: 1905.07687 [cs.CL].

[69] Chien-Sheng Wu et al. *Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems*. 2019. arXiv: 1905.08743 [cs.CL].

[70] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. 2012. arXiv: 1212.5701 [cs.LG].

[71] Chenwei Zhang et al. *Joint Slot Filling and Intent Detection via Capsule Neural Networks*. 2018. arXiv: 1812.09471 [cs.CL].

[72] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. "Understanding bag-of-words model: A statistical framework". In: *International Journal of Machine Learning and Cybernetics* 1 (Dec. 2010), pp. 43–52. DOI: 10.1007/s13042-010-0001-0.

[73] Victor Zhong, Caiming Xiong, and Richard Socher. *Global-Locally Self-Attentive Dialogue State Tracker*. 2018. arXiv: 1805.09655 [cs.CL].

6.0.0.0.0.1